MultiBatch 10

Technical Guide

Intelligent
workload
management
for
HPE NonStop.

Published: February 2023
Edition: J-series or L-series RVUs

Version Control

Document Tracking

MultiBatch Version:	10.0
Issue Date:	FEB 2023

Document History

Date	Version	Details
NOV 2022	1.0	New product version and updated structure of the Technical Guide. Introduction of Define Classes BMON supports 2500 jobs Increase configuration limits, see 1.21 and A3 Remove UTCSV New section describing conditional parameters MBPARHK processing now processes all non-step related records On Demand Job functionality added Support for OSS processes has been reworked and improved Conditional Parameter values can now be up to 100 characters Both Conditional Parameter and Event Timer processing are now fully incorporated in the MultiBatch menu, security and auditing functionality
FEB 2023		New Ops GUI Server Add SWITCH BMON command

Company Information

Insider Technologies is a UK-based software company with over 30 years of experience creating mission-critical products across Banking, Financial Services, Telecoms, Government and Defence organisations.

For details about the full range of products and services available from Insider Technologies, please get in touch with us.

Insider Technologies Limited
2 City Approach
Albert Street
Manchester
M30 0BL
United Kingdom

+44 (0)161 876 6606
hello@insidertech.co.uk
www.insidertech.co.uk

Table of Contents

Chapter 1 MultiBatch Overview

1.1 Introduction

MultiBatch is a workload automation manager specifically designed for NonStop. It provides high-performance, parallel processing combined with advanced operational control across your NonStop nodes from within a single interface. MultiBatch delivers an enterprise-class capability for NonStop users with demanding workload requirements.

Key Benefits

- Sophisticated and integrated job definitions. Jobs are run directly as defined in the configuration database without the need to write separate control scripts.
- A unique Job/Segment/Unit structure that facilitates both parallel processing and sophisticated process dependencies.
- Deep support for OSS processes. Specific support is included in the user interface and scheduler equally to that provided for Guardian processes.
- Job dependency logic managed centrally in the scheduler. There is no requirement to write specific job release logic. Job dependencies are automatically adjusted depending on the job set selected.
- Support for on demand processing in addition to supporting your traditional regular daily schedules. Pre-defined job templates are used as the basis for ad-hoc job processing.
- An active roadmap ensures continuous investment and innovation.

Features

- Easy menu driven installation process, simple step-by-step guide and defaults for all key fields.
- Unique job structuring allows tasks to be broken out to facilitate grouping, parallel processing and dependencies.
- Job dependencies are centrally controlled and automatically adjusted to consider daily scheduling.
- Scheduling by job and within job by calendar, absolute dates and group logic.
- Create model jobs, which users can clone, modify and run on demand.
- Defines, assigns, parameters, file descriptors and environment variables can be grouped and re-used across jobs.
- The migration process allows central configuration to be installed across environments, complete with the necessary node, volume and subvolume conversions.
- Simple effective security with log on using Guardian user IDs and passwords; facilities are grouped into classes, which are allocated to users.
- A comprehensive status database is maintained and displayed in real time. Failed processing is highlighted, take action by drilling down to individual failed processes.
- Start time monitoring and alerts are raised if a job has not started by the prescribed time.
- Built-in DR Facility, the full monitor context can be replicated; workloads can resume from the point of failure.
- Optional tight integration with user jobs allows MultiBatch to monitor, maintain, and report job status and to facilitate restart after a failure.
- Deep parameterisation for key configuration fields where tokens are substituted for values when a schedule is loaded.
- Comprehensive time based scheduling with at, every and Unix-like CRON functionality.
- A full audit trail of user actions and task runtime errors is maintained.

1.2 System and Software Requirements

MultiBatch required an HPE NonStop system running under L19.03, J06.22 or later versions.
MultiBatch requires the following products.

On L-series OS

SKU	Description
BE359AC	HPE NonStop TS/MP (ACS)
BE221AC	HPE NonStop COBOL85 Run Time Library

On J-series OS

SKU	Description
QSR78	HPE NonStop TS/MP (ACS)
QSB81	HPE NonStop COBOL85 Run Time Library

1.3 Technical Guide Structure

The Technical Guide is functionally based and is designed to give the user an easy step by step guide to MultiBatch processing. The manual incorporates areas such as:

- A general introduction to MultiBatch.
- The steps required to set up and run a schedule. In each case the relevant screens and function keys are described and an example is provided to illustrate the steps.
- Details of the Utilities which can be used with MultiBatch.
- Details of the BCOM syntax.
- Details of the MultiBatch Migrator Facility.

The steps provided for the user to follow in order to configure and run MultiBatch are as follows:

- Security Management: This allows access to the various MultiBatch functions to be restricted to certain users.
- Calendar Maintenance: This allows the user to specify certain dates in the Calendar when schedules are to be run (e.g. for Quarterly reports) or days to be avoided (e.g. Public holidays).
- Configure MultiBatch: This allows the user to configure MultiBatch to suit their requirements.
- Define "Jobs": This allows the user to set up the jobs which are to be run. This involves defining the "Jobs" in a hierarchical way, breaking them down into Jobs, Segments and Units. Any dependencies are also set up at this time.
- Prepare Schedules: For a given date, the Prepare will report on jobs that will run and that will not run, together with the reasons why. The Prepare will also reconnect jobs whose dependent jobs are not scheduled to run and create a file of configuration commands to be used by a subsequent BCOM BUILD session.
- Run Jobs: Once the Jobs have all been scheduled, they can be run. Once running, the jobs will run in serial/parallel as defined by the Job Schedule.
- Monitor Job Status: The current status of individual jobs may be examined at any time and the job noted as "waiting", "running" or "failed", including the monitoring of Dependencies at Job and/or Segment level. More details are available at each stage if required.
- Error Resolution: MultiBatch provides a list of the most recent errors and an Operators Diary facility that allows the handover or the correction of the fault to be documented.

Some of these steps are merged into functional units and described as such in the manual. The following chapters describe functions such as how to:

- Logon to the MultiBatch system.
- Prepare and run Schedules.
- Monitor Job status.

- Run utilities.
- Recover a BMON.

The Technical Guide is designed to lead the user sequentially through all the steps required to use MultiBatch. It is divided into chapters and annexes on a "functional" basis (e.g. a chapter on monitoring) and the contents provide the user with "pointers" to general areas of interest. These are then sub-divided into smaller sections which allow the user to home in on a specific area of interest.

Each section of the Technical Guide contains a description of all relevant screens and fields and the functions which can be performed on each screen. A list of the MultiBatch function keys are provided at the end of Chapter 1.

1.4 MultiBatch Architecture

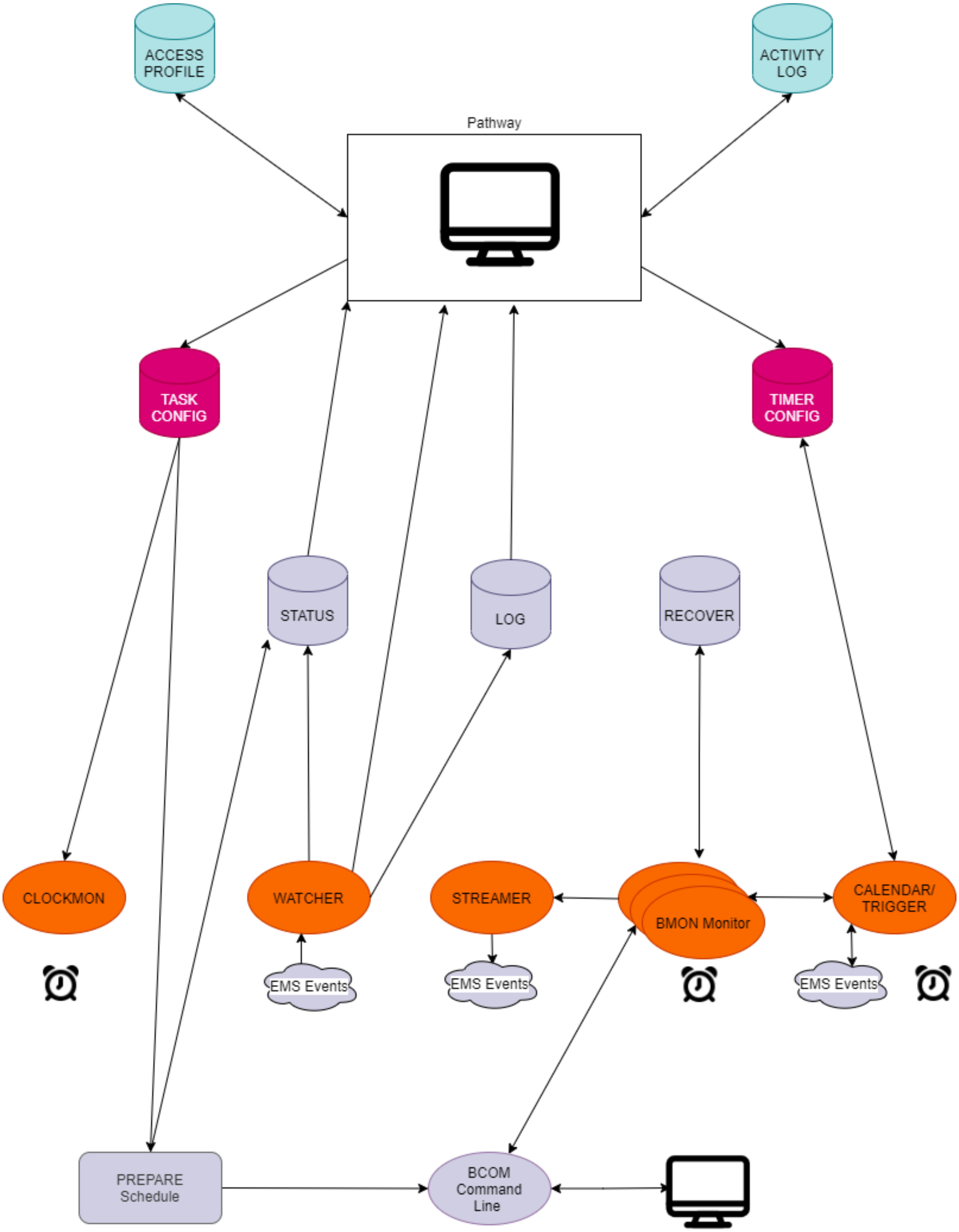


Figure 1.1 – MultiBatch Architecture

1.5 Architecture

Figure 1.1 shows the interaction between the major processes of a running MultiBatch system.

PROFILE and ACTIVITY logs form the access control and security elements of MultiBatch.

PROFILE contains the security profiles for authorised users where MultiBatch facility access is assigned to classes which are in turn assigned to users. Chapter 2 covers Security Management.

ACTIVITY LOG provides a log of changes that have been made your MultiBatch configuration. Chapter 7 covers Audit Logs.

STATUS, LOG and RECOVER are the dynamic run time MultiBatch databases.

STATUS provides information on the progress of schedules. Chapter 5 covers Status Monitoring.

LOG is written to, where a process does not complete successfully. Chapter 6 covers Error Resolution.

TASK CONFIG and TIMER CONFIG form your schedule configuration. Chapter 3 covers the configuration of Jobs, Segments and Units, Chapter 4 covers the configuration of schedules and Chapter 8 covers time based scheduling.

The orange processes are the MultiBatch run time process pairs:

CLOCKMON – start time monitor, see Chapter 5

WATCHER – processes run time events, see Chapter 5

STREAMER – the interface to EMS, see Chapter 5

BMON – see the next section

CALENDAR/TRIGGER – see Chapter 8

PREPARE – processes the schedule configuration for loading into BMON, see Chapter 4

BCOM – the BMON command line interface, see later in this chapter

1.6 BMON

BMON is a NonStop process which initiates processes and then continually monitors and logs their progress in a structured fashion.

BMON incorporates the facility for processes to be restarted from the point at which they stopped, with Database integrity intact, if the programs concerned have been coded to BMON's Application Interface standards.

Two object files are provided for BMON, namely, BMONL and BMONU.

The use of BMONL or BMONU is dependent on whether or not is it decided to use the "Set Step User ID" functionality within the CST screen. If this is required, then license and utilise the BMONL object code

BMONL can then be run as a "privileged program" for switch User ID, thereby allowing the BMON scheduler to run jobs as different User IDs, which is created via the CST "Set Step User ID" feature. If the "Set Step User ID" feature is not to be used, then rename BMONU to BMONL or BMON.

Optionally, simply refer to BMONL or BMONU object code.

Note: *BMON, the NonStop process which initiates, monitors and logs the progress of your job workload and has a number of interfaces. As with most HPE NonStop monitor processes this interface uses the message based $RECEIVE mechanism to pass Inter Process Messages (IPMs) from a command interface. The command interfaces to BMON are from:*

1) *BCOM – the Command Line Interface*

2) *TRIGGER – when processing jobs, segments and units set up in the Event-Timer scheduler*

3) *LIBUTIL – when using the programmatic interface to HOLD, RELEASE, START, RESTART, ABORT, RESTART UNIT, IFFAILED for Jobs, Segments or Units*

4) *STATSRV – provides BMON status details for the MultiBatch Screen COBOL Interfaces.*

When these command interfaces communicate with BMON an initial message is sent to ensure that further messages will be compatible with the BMON version.

BMON returns status 409 when the version of messages used by a command interface do not match the BMON version. When BCOM receives this 409 reply from BMON, it will issue the following message via the BCOM command line interface:

**** ERROR 515 *** Incompatible releases of BCOM & BMON*

The BMON interface is only updated for a major release, in this situation all of BCOM, TRIGGER, LIBUTIL and STATSRV are also part of the release. In this way the 409 status shows there is a configuration issue.

Status 409 is commonly generated where customer programs interfacing with BMON using LIBUTIL need to be modified to include the latest delivered LIBUTIL procedures.

1.7 Structure within BMON

A given Process within a BMON is assigned a "**UNIT**" identity which consists of TWO alpha, numeric, or alphanumeric characters.

Each Unit is assigned to a **SEGMENT** which is also identified by TWO alpha, numeric, or alphanumeric characters.

Each segment is assigned to a "**JOB**" whose identity consists of FOUR alpha or alphanumeric characters. A Job, Segment and Unit can also be allocated a unique alias name description, which forms part of the MultiBatch tokenised EMS event.

It can be seen from Figure 1.2 that BMON has a downward tree structure in its method of uniquely identifying a given Process. This is further extended by the fact that each Unit may consist of one or two "**STEP**'s". The use of this structure is extremely flexible, and any given environment can be tailored for specific requirements.

A simple summary of how the entities in the structure can be utilised is as follows (See Figure 1.2):

JOBs can be described as a set of functionally related programs to be run.

SEGMENTs can be used to represent group sets of Distributed processes, databases, regions, nodes, etc.

UNITs are the actual Programs which are to be run.

Note: *The two-step unit. Consider an example job where **FUP** needs to copy a file to a process.*

This process converts the file records into a new format and writes the converted data to a tape.

These two jobs are inextricably linked and together they are two programs that form one job.

*They should be started, restarted, aborted, held and released together and you can use **UNIT** commands to achieve this. However, the individual configurations need to be added and altered separately, e.g. change the priority of the **FUP** process - **STEP** commands are used to achieve this end.*

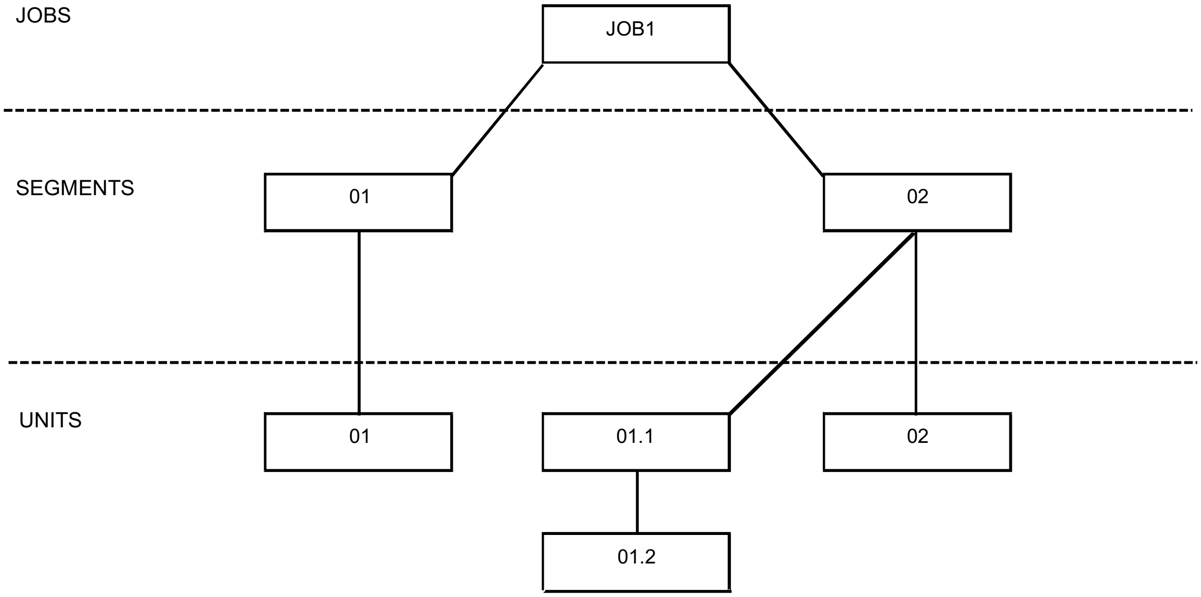


Figure 1.2 – Utilisation of MultiBatch Entities

1.8 Naming Conventions

For the rest of this description, the format <JOB.SEGMENT.UNIT.STEP> will be used to represent and uniquely identify any given Process which BMON is able to execute and monitor.

Figure 1.2:

Shows a JOB.

This job contains two SEGMENTs.

Segment 02 consists of two UNITs, one of which has two STEPs.

The second Step of the Two-Step Unit is represented as "JOB1.02.01.2".

A Single-Step Unit does not need any STEP qualification to identify it, and can be referred to simply using the form "JOB1.01.01".

This structure allows programs to be grouped together within BMON, enabling these groups to be initiates as a single entity, and monitored in a controlled, highly visible and easily identifiable fashion.

1.9 Protocol within BMON

BMON will deal with any Process executing on a NonStop System, recording start and finish times and whether the Process stopped normally, experienced an abnormal end ("abended"), or the Process's CPU failed.

This level of Process control is inadequate for any environment which demands total Data Integrity, to the extent that this does not guarantee that a "Stopped Process" actually completed successfully.

To this end BMON has the ability to "converse" with a user written program by providing a PROTOCOL by which, communication between the Monitor Process and user-written Application Programs may be conducted.

One aspect of this Protocol is that when a Process has completed successfully, BMON is informed and logs the event.

This facility is further enhanced by the concept that once a Process is started it should be able to resume processing from the point at which it stopped in the event of a failure.

Implementation of BMON Protocol within a user program provides a mechanism whereby Application Programs may be protected from both an accidental re-run and be presented with a request to restart when appropriate.

See Appendix G - BMON Protocol for a description of how the protocol can be implemented in end user programs.

1.10 Start Processes Running

BMON starts processes by utilising System Procedure Calls:

- "PROCESS_CREATE_" System Procedure call for Guardian Steps.
- "PROCESS_SPAWN_" System Procedure call for OSS Steps.

For two step units, STEP 1 will be started before STEP 2, so that STEP 2 can write to STEP 1.

PROTOCOL Based Units:

In the event of a Job being stopped prematurely, perhaps due to a CPU failure, it can only be "Restarted". The primary reason for this relates to MultiBatch Protocol based processes.

NON-PROTOCOL Based units:

For Non-Protocol Units, a "Restart" will effectively be a Rerun of the Program from the beginning, however BMON will log the number of "Restart" operations performed by a user for each individual Unit.

Note: *Although MultiBatch provides this PROTOCOL facility, users can implement their own restart strategies if necessary.*

1.11 Log Files

BMON logs events and these messages are sent to BMON's LOG1 and LOG2 files. The LOG1 and LOG2 files may be allocated or changed dynamically from within BCOM; optionally LOG1 may be the "OUT" file from the "RUN BMON" command.

Supplied with the MultiBatch environment is a 'STREAMER' Process which will translate text events from BMON into tokenised EMS events. Streamer will then forward these tokenised events either to the Primary EMS Collector, $0, or to an Alternate Collector process. To use this facility, the Streamer Process name is specified to BMON as the LOG1 or LOG2 file.

In the event that neither the user-supplied LOG Files nor the Streamer Process are available or accessible, then the messages will be sent to BMON's home terminal; if this is not available, the messages are sent as Text events to the Collector.

If a logging file is a Terminal and a "soft" error occurs the file will not be closed, otherwise the log file's error will be logged elsewhere and the log file closed.

A "soft" error can be described as a "temporarily unavailable" type of error such as when the terminal is being used for another purpose.

The following steps are taken when writing a log message:

1. Write to LOG1, if open.

2. Write to LOG2, if open.

3. If message is not written to LOG1 or LOG2, then write to BMON's Home Terminal (as specified in the command which started the BMON Process executing).

4. If not written to LOG1, LOG2 or Home Terminal, then write to $0 or the Alternate Collector.

5. If an error occurred whilst attempting to write to LOG1 or LOG2, then close relevant file(s) and repeat the above with appropriate error message.

1.12 NonStop Operation

The BMON Process can be configured to run as a NonStop Process pair in order to protect against a CPU failure; see CBM screen.

1.13 System Failure

BMON utilises a TMF Audited Recovery file, to which the status of the BMON scheduler will be continually recorded.

As the Recovery file is TMF protected, a user can optionally, replicate the file to a standby system, so that in the event of say a system failure, the BMON can be recovered on the standby node up to the point of failure.

Just two simple commands are all it takes to recover the BMON on the new node, as during recovery, the BMON contents such as Node name, Disk, Subvolume, limited OSS directory names can be automatically swapped to the new node values. This means that the recovered BMON is ready to resume processing with no manual reconfiguration.

When the BMON is recovered in this way, any jobs which were running at the time of the failure will be reported and the user can analyse the recovered BMON to determine the status of the schedule.

The recovered state of each Unit will then have to be individually assessed. The important decision will be to assess which Units stopped prematurely.

See Appendix H 'Recovering a BMON' for further details.

See also, 'Chapter 8 Event Timer' for guidance on managing and recovering Event Timer during site-swaps and/or system failures.

1.14 Status Flags

The history of each stage in every Process initiated and executed by BMON is recorded (for example, process stopped, Job complete) in an individual collection of Status Flags which are clustered into one "word" (16 bits).

A complete breakdown of the make-up of flags is provided in Appendix A.9 Miscellaneous Commands.

1.15 Running a BMON

To run BMON, assuming the Object code is held in "$SYSTEM.SYSTEM" which is typically referenced within the system's #PMSEARCHLIST built in variable, the following command could be used:

```
BMON / NAME $BMON, NOWAIT, CPU 2, HIGHPIN <ON/OFF> / ABORT^DELAY 10
```

This would start BMON with a process name of "$BMON". The BMON process can run as a HIGHPIN process if required.

There is a facility within MultiBatch to specify the maximum time that a Unit is to run. The user can either be informed by a warning message that the unit has reached maxtime, or alternatively the Unit can be aborted when its maxtime has been reached.

The ABORT^DELAY parameter may be used to specify how many minutes before abort and a warning message will appear in the log, e.g.:

```
2018-03-02 16:53:40 \LIVE.$MSTR:1509389657 MULTIBAT.1.0 055 AAAA.AA.01.*,{AAAA-UNIT}
ABORT will be forced at MAXTIME in 10 minutes, \LIVE.$BMON 02/03/18, 16:53.
```

This gives the user more time to turn the facility off by using the ALTER STEP <name> NOMAXTIME command. If the step was not altered in this way, then based on the example above, it will be aborted and an EMS event issued:

```
2018-03-02 17:03:41 \LIVE.$MSTR:1509389657 MULTIBAT.1.0 056 AAAA.AA.01.1,{AAAA-UNIT}
aborted at MAXTIME, \LIVE.$BMON 02/03/18, 17:03
```

BMON must be named to allow BCOM to access it; the name can be any valid NonStop Process name and must match the entry in the MultiBatch 'Maintain BMON Configuration' (CBM) screen.

1.16 Scheduling Jobs

Jobs and segments can be automatically started when pre-dependent JOBs and SEGMENTs have completed successfully.

Additionally, Jobs may be scheduled by a Start Time which may be used in conjunction with JOB DEPENDENCIES or on its own.

Event Timer can be used to automatically schedule the start of a job based on a database of Start Time and Calendar Periods.

See Chapter 8 Event Timer for further information.

It is also possible to create On Demand Jobs in BCOM from previously created Model Jobs, see Chapter 9 On Demand Jobs.

1.17 BCOM

BCOM is the Command Line Interface to an active BMON Process.

BCOM can be used to configure BMONs environment, inspect BMONs status and configuration, start processes running via BMON and obtain information about each of the entities to be monitored by BMON.

The BCOM command syntax is fully explained in Appendix A - BCOM Syntax

1.18 Running BCOM

During a BMON build process, an ASSIGN referencing the MultiBatch database DATACONF file is specified within user macros, e.g.:

```
ASSIGN DATACONF , FILE $LIVE.MBATDAT.DATACONF
```

This Assign name is included in the running BMON process and resides there until the BMON is shutdown.

Because of this, a user can execute the BCOM command against the running BMON process from any TACL session; there is no need to specify an ASSIGN within any user TACL macros or obey files.

Ensure that the ASSIGN is specified during subsequent BMON builds.

Then the command to run BCOM is as follows, assuming BCOM resides in $SYSTEM.SYSTEM which is typically referenced within the #PMSEARCHLIST variable:

```
BCOM / IN <infile>, OUT <outfile>, HIGHPIN <ON / OFF> / <BMON-name>; <command>
```

Where:

<infile> is an optional input command file.

<outfile> is an optional log file.

BMON can run as a HIGHPIN process if required.

<BMON-name> is the name of the BMON Process that BCOM is to communicate with.

<command> is an optional, valid BCOM command to be processed.

In the absence of an IN or an OUT file the current home terminal is used. If there is a command in the parameter field this will be used in place of an input file.

For interactive input files (Terminals and Processes) the BCOM prompt will be ">>".

In the event of no BMON Process name parameter being specified, the default name is "$BMON". If this is the wrong BMON and it is not currently running, then the name of the BMON process to be accessed can be dynamically allocated and changed anytime during the BCOM session by using the OPEN BMON command, e.g. OPEN BMON
$BMCL.

You may decide to create a file of frequently used commands, which may then be executed at the BCOM prompt as shown below:

```
OBEY <filename>
```

SYNTAX parameter may be used at run time to scan the configuration filename WITHOUT building the MultiBatch environment and to just check for any syntax errors.

EXTMEM may be used to calculate the size of the extended memory segment to be used; the SYNTAX check will be simultaneously performed.

Issue one of the following commands:

```
BCOM / in <infile> / SYNTAX

BCOM / in <infile>/ EXTMEM
```

1.19 MultiBatch Function Keys

The following is a list of standard function keys.

F1 List Exact
F2 Amend
F3 Delete
F4 Insert
F5 List Next
F8 Detail
F9 Reset
F13 Navigate Help
F14 Navigate
F15 Return
F16 Help

SF1 First Page
SF2 Next Page
SF3 Previous Page
SF4 Confirm
SF11 Explain
SF13 Show Function Keys (1)
SF14 Print Screen (2)
SF15 Display Previous Menu
SF16 Logoff

Next Page/Page Down

Prev Page/Page Up

Note (1): *All the valid function keys for any screen should be displayed. If there is no room on the screen for all the keys to be displayed at the same time, then SF13 will rotate through the options.*

Note (2): *The print location for the SF14 print function is determined by the PATHWAY configuration command:* ***SET PROGRAM PRINTER $S.#MBAT*** *or* ***SET TERM PRINTER $S.#MBAT*** *depending on whether your PATHWAY session was invoked by a RUN command (PROGRAM) or a START TERM command.*

1.20 Conditional Parameters

If there is an attribute that is common across a large number of records, e.g. a step's Home Terminal, then if the value of this attribute changes, this will require many updates within a BMON or BMONs configuration. To circumvent this, the conditional parameter facility can be used to provide a pointer into a parameter database, e.g.:

<HOME-TERM>, where "<" and ">" are the delimiters and HOME-TERM is the conditional parameter name.
In the Conditional Parameter database (PME), HOME-TERM could be set to $ZHOME, and this value will be used wherever <HOME-TERM> is referenced. If the value of home terminal is to be changed to $VHS, then just a single screen update is required within the Maintain Conditional Parameters (PME).

In addition to this, <HOME-TERM> can be declared as a global parameter to be used by all BMON environments, or it can be used by just a single environment when it is then known as a local parameter. When accessing a local parameter, a Shell Name is required.

The Shell Name for a BMON is set on the Maintain BMON Configuration (CBM) screen. When reconciling a Conditional Parameter this Shell Name is used together with the parameter name to look up a value for the BMON. The found value may indicate that the global value for all BMONs is to be used. See Appendix D Conditional Processing for a detailed description of the setting of Conditional Parameters and their values.

To aid clarity it is suggested that the Shell for a BMON is set to the BMON's process name without the dollar prefix. It is also possible to have the same Shell for all BMONs as Conditional Parameters are likely to have the same value across all BMONs.

A number of MultiBatch screens allow conditional values to set, these are reconciled when BCOM processes the values as created by the PREPARE program, see Prepare Phase (PRM) within Chapter 4 Scheduling. As stated above a value such as <HOME-TERM> can be used for a field. To see the value that will be substituted it is necessary to complete the screen's Explain Shell field, place the cursor on the field containing the parameter and then press SF11.

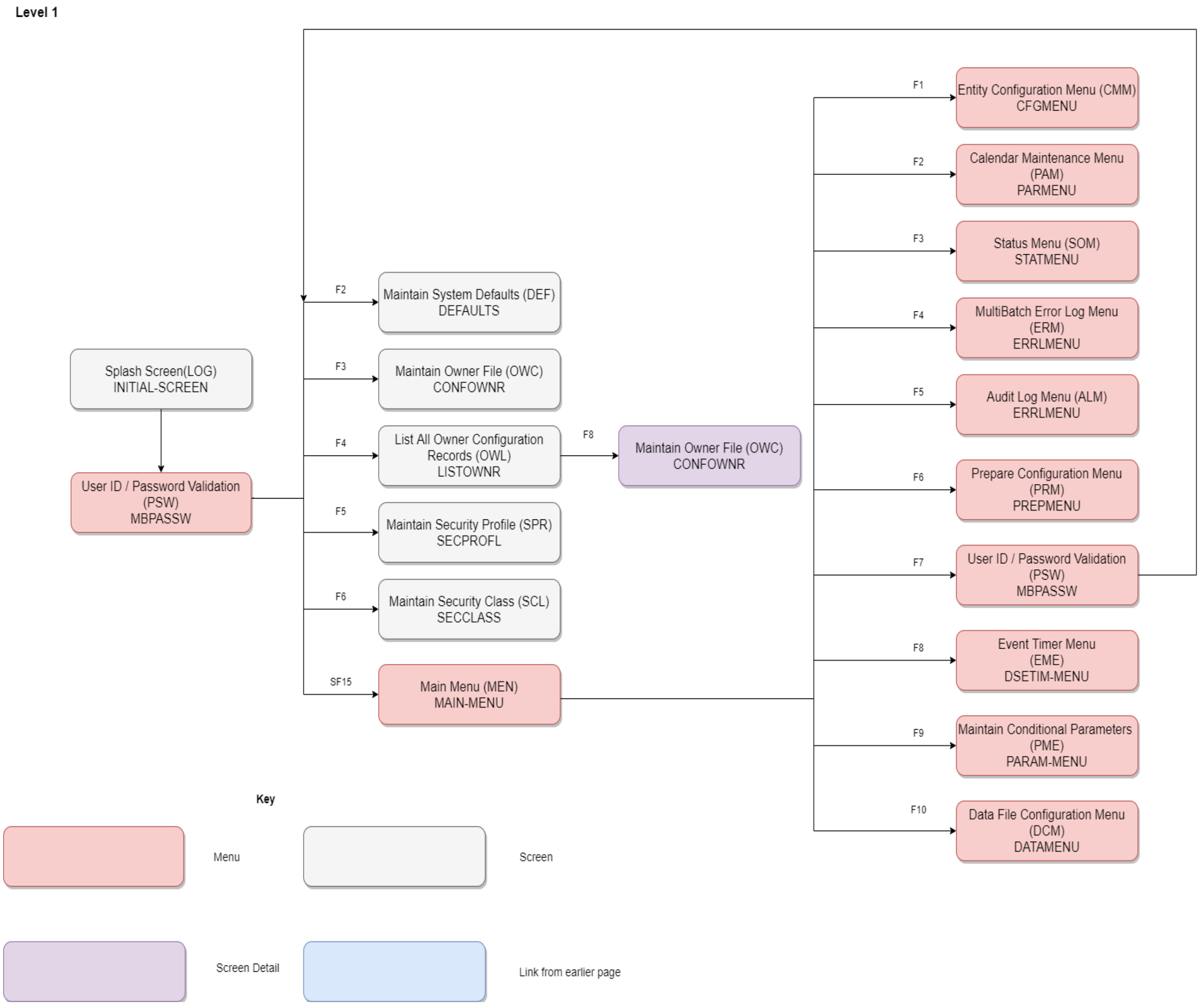
Typically, the Explain Shell field should be set to the BMON Shell Name from the CBM screen. The value can be preset for all screens where it appears by using the Maintain System Defaults (DEF) Screen, see Chapter 2 Security Management.

The default delimiters may be altered at the step level on the Maintain STEP Configuration (CST) and the Maintain OSS STEP Configuration (COS) screens, this is not usually necessary. Be aware this affects the processing of all Conditional Parameters when BCOM load the step, not just those directly defined for the step. For example, if the step uses Assign Classes, see the Assign Class Definition (CAS) screen, that have conditional parameters, these must also use the steps delimiter.

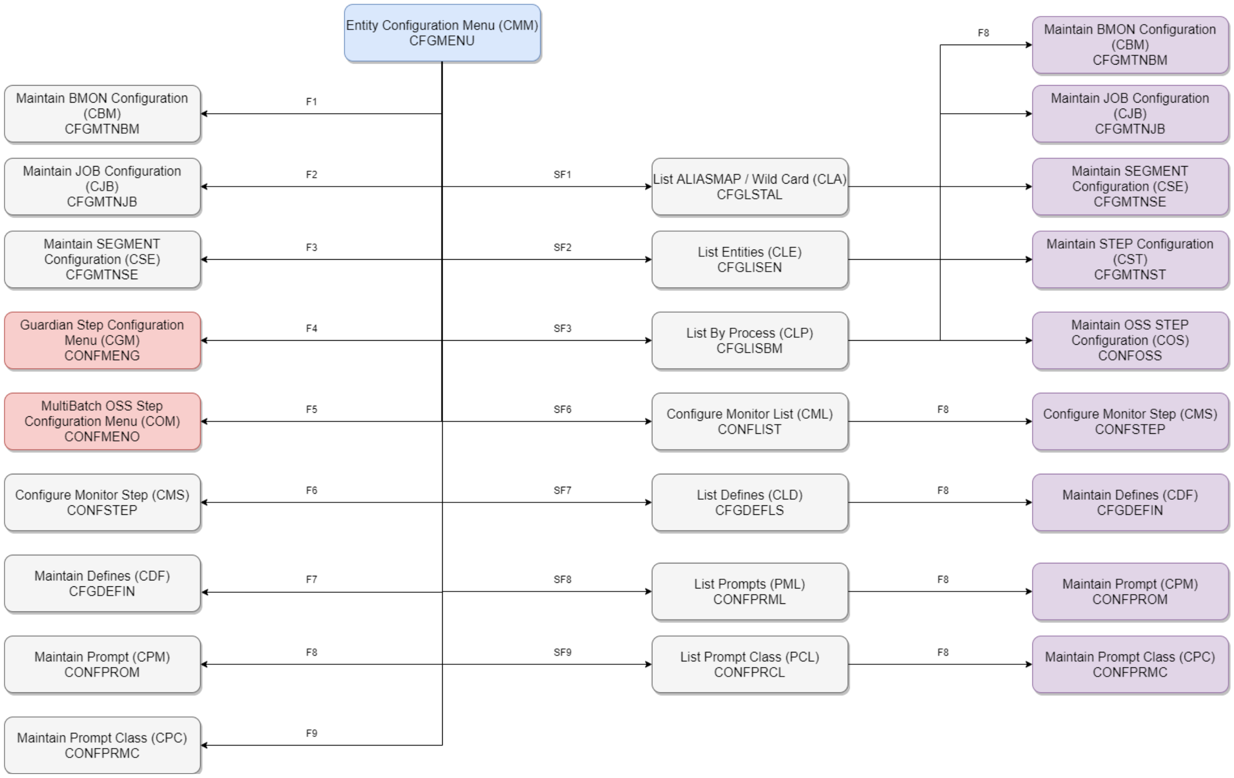
As described Conditional Parameters can be used on many record type/screens that are not step related, for example Define and Assign Classes. For these consider using global Conditional Parameters where several Shells have been defined but parameters have the same value.

1.21 Pathway Screen Navigation

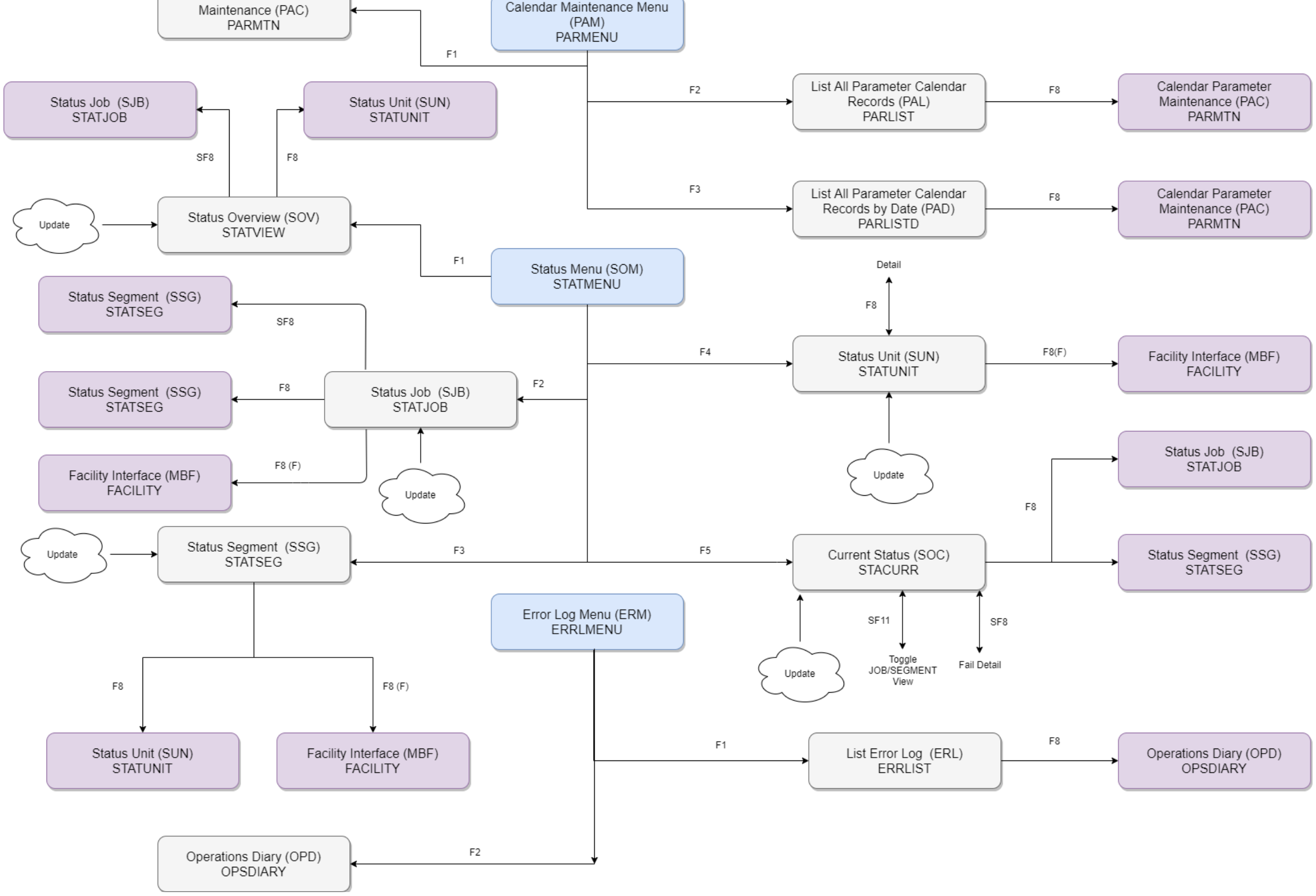
The diagrams below show all the screens in the Pathway Screen COBOL interface. They are intended as a reference to be used from the screen descriptions throughout this guide. As one screen leads to another the level or depth of nesting is never more than 3 screens.


Level 1
Splash Screen(LOG)
INITIAL-SCREEN
User ID / Password Validation
(PSW)
MBPASSW
F2
Maintain System Defaults (DEF)
DEFAULTS
F3
Maintain Owner File (OWC)
CONFOWNR
F4
List All Owner Configuration
Records (OWL)
LISTOWNR
F8
Maintain Owner File (OWC)
CONFOWNR
F5
Maintain Security Profile (SPR)
SECPROFL
F6
Maintain Security Class (SCL)
SECCLASS
SF15
Main Menu (MEN)
MAIN-MENU
F1
Entity Configuration Menu (CMM)
CFGMENU
F2
Calendar Maintenance Menu
(PAM)
PARMENU
F3
Status Menu (SOM)
STATMENU
F4
MultiBatch Error Log Menu
(ERM)
ERRLMENU
F5
Audit Log Menu (ALM)
ERRLMENU
F6
Prepare Configuration Menu
(PRM)
PREPMENU
F7
User ID / Password Validation
(PSW)
MBPASSW
F8
Event Timer Menu
(EME)
DSETIM-MENU
F9
Maintain Conditional Parameters
(PME)
PARAM-MENU
F10
Data File Configuration Menu
(DCM)
DATAMENU
Key
Menu
Screen
Screen Detail
Link from earlier page

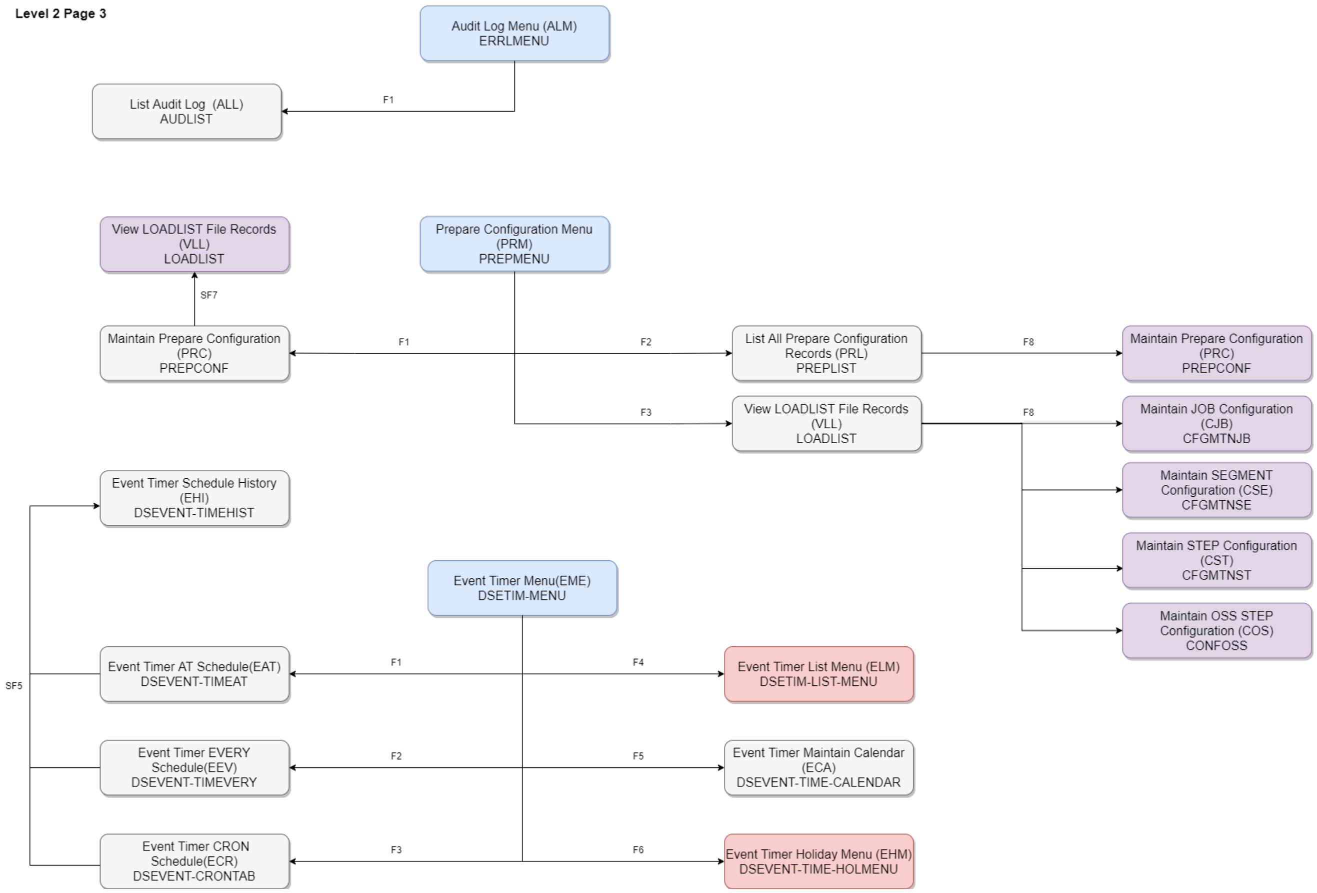
Level 2 Page 1


Entity Configuration Menu (CMM)
CFGMENU
F1
Maintain BMON Configuration (CBM) CFGMTNBM
F2
Maintain JOB Configuration (CJB) CFGMTNJB
F3
Maintain SEGMENT Configuration (CSE) CFGMTNSE
F4
Guardian Step Configuration Menu (CGM) CONFMENG
F5
MultiBatch OSS Step Configuration Menu (COM) CONFMENO
F6
Configure Monitor Step (CMS) CONFSTEP
F7
Maintain Defines (CDF) CFGDEFIN
F8
Maintain Prompt (CPM) CONFPROM
F9
Maintain Prompt Class (CPC) CONFPRMC
SF1
List ALIASMAP / Wild Card (CLA) CFGLSTAL
SF2
List Entities (CLE) CFGLISEN
SF3
List By Process (CLP) CFGLISBM
SF6
Configure Monitor List (CML) CONFLIST
SF7
List Defines (CLD) CFGDEFLS
SF8
List Prompts (PML) CONFPRML
SF9
List Prompt Class (PCL) CONFPRCL
F8
Maintain BMON Configuration (CBM) CFGMTNBM
Maintain JOB Configuration (CJB) CFGMTNJB
Maintain SEGMENT Configuration (CSE) CFGMTNSE
Maintain STEP Configuration (CST) CFGMTNST
Maintain OSS STEP Configuration (COS) CONFOSS
F8
Configure Monitor Step (CMS) CONFSTEP
F8
Maintain Defines (CDF) CFGDEFIN
F8
Maintain Prompt (CPM) CONFPROM
F8
Maintain Prompt Class (CPC) CONFPRMC

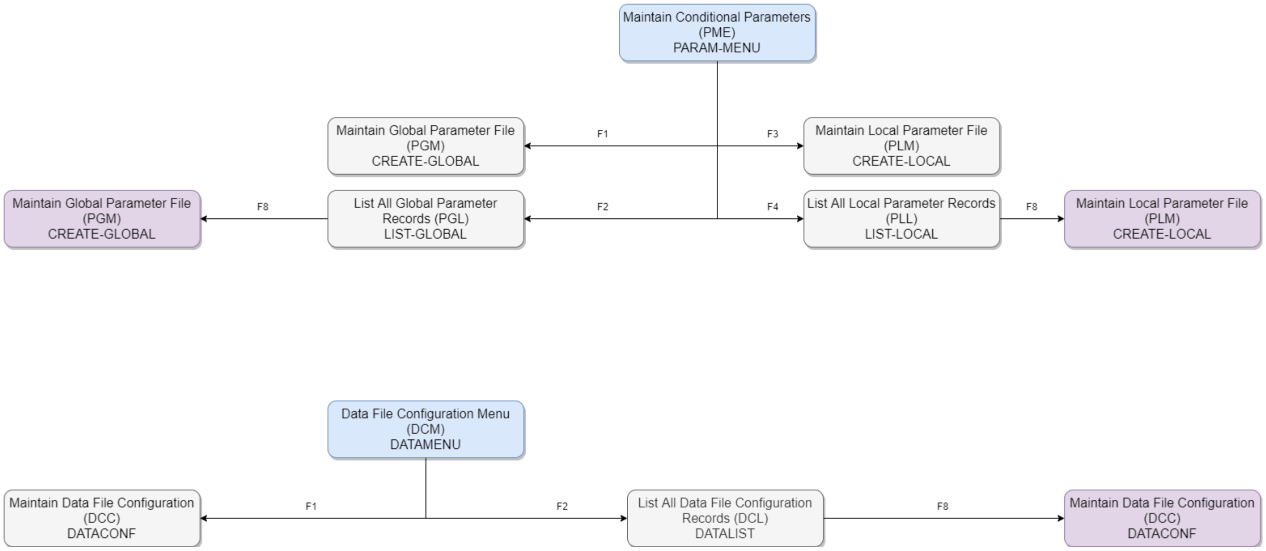
Level 2 Page 2


Calendar Parameter Maintenance (PAC) PARMTN
Calendar Maintenance Menu (PAM) PARMENU
F1
F2
List All Parameter Calendar Records (PAL) PARLIST
F8
Calendar Parameter Maintenance (PAC) PARMTN
F3
List All Parameter Calendar Records by Date (PAD) PARLISTD
F8
Calendar Parameter Maintenance (PAC) PARMTN
Status Job (SJB) STATJOB
Status Unit (SUN) STATUNIT
SF8
F8
Update
Status Overview (SOV) STATVIEW
F1
Status Menu (SOM) STATMENU
Status Segment (SSG) STATSEG
SF8
Status Segment (SSG) STATSEG
F8
Status Job (SJB) STATJOB
F2
Facility Interface (MBF) FACILITY
F8 (F)
Update
Detail
F8
F4
Status Unit (SUN) STATUNIT
F8(F)
Facility Interface (MBF) FACILITY
Update
Status Job (SJB) STATJOB
F8
Update
Status Segment (SSG) STATSEG
F3
F5
Current Status (SOC) STACURR
Status Segment (SSG) STATSEG
Error Log Menu (ERM) ERRLMENU
SF11
SF8
Update
Toggle JOB/SEGMENT View
Fail Detail
F8
F8 (F)
Status Unit (SUN) STATUNIT
Facility Interface (MBF) FACILITY
F1
List Error Log (ERL) ERRLIST
F8
Operations Diary (OPD) OPSDIARY
F2
Operations Diary (OPD) OPSDIARY

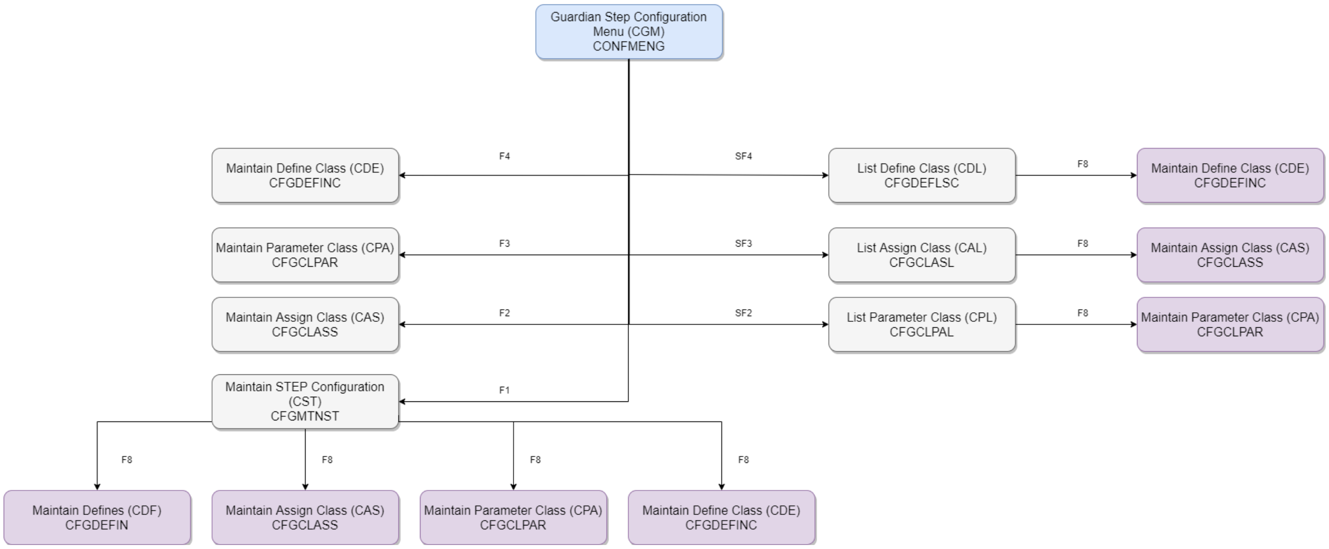
Level 2 Page 3



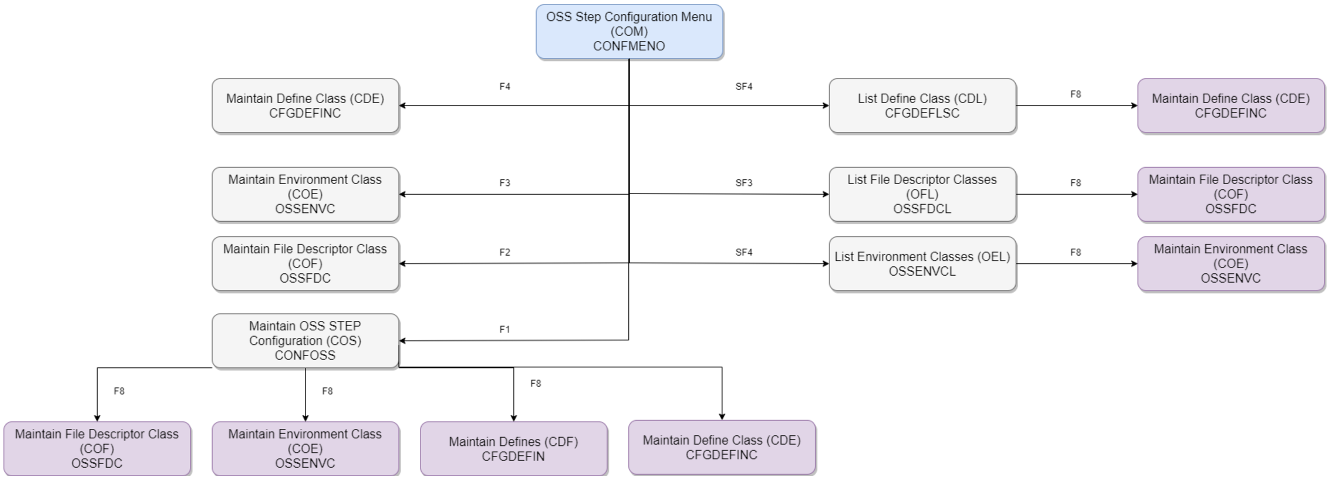
Level 2 Page 4


Maintain Conditional Parameters
(PME)
PARAM-MENU
Maintain Global Parameter File
(PGM)
CREATE-GLOBAL
F1
F3
Maintain Local Parameter File
(PLM)
CREATE-LOCAL
Maintain Global Parameter File
(PGM)
CREATE-GLOBAL
F8
List All Global Parameter
Records (PGL)
LIST-GLOBAL
F2
F4
List All Local Parameter Records
(PLL)
LIST-LOCAL
F8
Maintain Local Parameter File
(PLM)
CREATE-LOCAL
Data File Configuration Menu
(DCM)
DATAMENU
Maintain Data File Configuration
(DCC)
DATACONF
F1
F2
List All Data File Configuration
Records (DCL)
DATALIST
F8
Maintain Data File Configuration
(DCC)
DATACONF

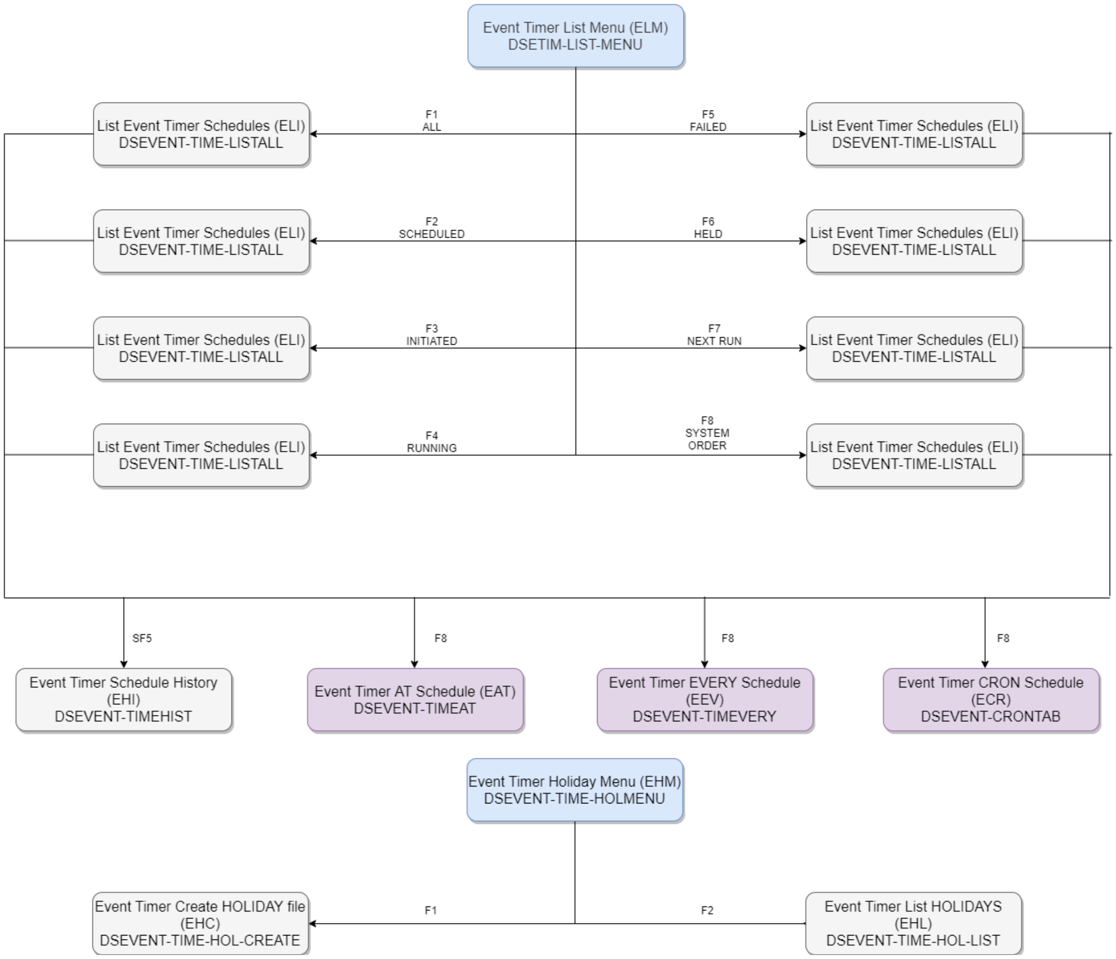
Level 3 Page 1


Guardian Step Configuration Menu (CGM)
CONFMENG
F4
SF4
Maintain Define Class (CDE)
CFGDEFINC
List Define Class (CDL)
CFGDEFLSC
F8
Maintain Define Class (CDE)
CFGDEFINC
F3
SF3
Maintain Parameter Class (CPA)
CFGCLPAR
List Assign Class (CAL)
CFGCLASL
F8
Maintain Assign Class (CAS)
CFGCLASS
F2
SF2
Maintain Assign Class (CAS)
CFGCLASS
List Parameter Class (CPL)
CFGCLPAL
F8
Maintain Parameter Class (CPA)
CFGCLPAR
Maintain STEP Configuration (CST)
CFGMTNST
F1
F8
F8
F8
F8
Maintain Defines (CDF)
CFGDEFIN
Maintain Assign Class (CAS)
CFGCLASS
Maintain Parameter Class (CPA)
CFGCLPAR
Maintain Define Class (CDE)
CFGDEFINC

Level 3 Page 2


OSS Step Configuration Menu (COM)
CONFMENO
F4
SF4
Maintain Define Class (CDE)
CFGDEFINC
List Define Class (CDL)
CFGDEFLSC
F8
Maintain Define Class (CDE)
CFGDEFINC
F3
SF3
Maintain Environment Class (COE)
OSSENVC
List File Descriptor Classes (OFL)
OSSFDCL
F8
Maintain File Descriptor Class (COF)
OSSFDC
F2
SF4
Maintain File Descriptor Class (COF)
OSSFDC
List Environment Classes (OEL)
OSSENVCL
F8
Maintain Environment Class (COE)
OSSENVC
Maintain OSS STEP Configuration (COS)
CONFOSS
F1
F8
F8
F8
Maintain File Descriptor Class (COF)
OSSFDC
Maintain Environment Class (COE)
OSSENVC
Maintain Defines (CDF)
CFGDEFIN
Maintain Define Class (CDE)
CFGDEFINC

Level 3 Page 3


Event Timer List Menu (ELM)
DSETIM-LIST-MENU
F1
ALL
F2
SCHEDULED
F3
INITIATED
F4
RUNNING
F5
FAILED
F6
HELD
F7
NEXT RUN
F8
SYSTEM
ORDER
List Event Timer Schedules (ELI)
DSEVENT-TIME-LISTALL
SF5
F8
F8
F8
Event Timer Schedule History (EHI)
DSEVENT-TIMEHIST
Event Timer AT Schedule (EAT)
DSEVENT-TIMEAT
Event Timer EVERY Schedule (EEV)
DSEVENT-TIMEVERY
Event Timer CRON Schedule (ECR)
DSEVENT-CRONTAB
Event Timer Holiday Menu (EHM)
DSEVENT-TIME-HOLMENU
F1
F2
Event Timer Create HOLIDAY file (EHC)
DSEVENT-TIME-HOL-CREATE
Event Timer List HOLIDAYS (EHL)
DSEVENT-TIME-HOL-LIST

1.22 Configuration Limits

Limit	Maximum
Number of BMON Processes	Unlimited
Number of Jobs per BMON Process	2500
Number of Segments per JOB	200
Number of Units per Segment	200
Number of Job to Job Dependencies	10 per Job
Number of Segment to Segment Dependencies	10 per Segment
Number of Assigns per Step	300
Number of Parameters per Step	100, or less if the total length of the parameter names + total length of the values + (2 * number of parameters) > 1024
Number of Defines per Step	300
Number of Environment Variables per Step	20
Number of File Descriptors per Step	10
Length of Startup Parameter String	80 characters
Number of Jobs Monitored by Watcher	20,000
Number of Global Parameters	Unlimited
Number of Local Parameters	Unlimited
Number of Jobs in the Event Timer Database	Unlimited

Chapter 2 Security Management

2.1 Introduction

This Chapter describes how to load the MultiBatch Pathway and how to access MultiBatch via the Password Validation Screen (PSW) – Figure 2.1.

This Chapter also:

- Provides an overview of the Security System.
- Describes how to Logon to MultiBatch.
- Provides an overview of how the Security Screens are linked/accessed and a brief description of their function.
- Describes how to add MultiBatch Owners to the system by first building Security Classes and then Security Profiles.

To conclude, the chapter details the On-Line Help Facility (F16 function key) and Navigation Facility (F13 function key) provided on the MultiBatch Screen COBOL interface.

While the emphasis is on giving worked examples to explain the screen and functions available, stand-alone screen/ function key descriptions are given where required.

2.2 Security Overview

The MultiBatch Security System provides an integrated environment within which users, System Managers, Support personnel and System Operators can configure a multi-node and multi-schedule environment.

You can configure your MultiBatch Security System in any way you wish, or you can follow the steps in this chapter to configure users such as MBAT.MANAGER, MBAT.SCHEDULE, etc., until you become familiar with the MultiBatch Security System configuration.

2.3 Add First User

Following a MultiBatch installation, the MultiBatch Security database does not contain any group.user ids. It does however contain a Security Class ‘ALLFACIL’, which provides full access to all of the MultiBatch Pathway facilities.

In order to add the first user, e.g. MBAT.CONFIG into the empty MultiBatch database, execute the UTADDUSR utility:

```
RUN <Object Svol>.UTADDUSR / IN <Database Svol>.DATACONF /A, B
```

where A = an existing Guardian Group.User name (not Alias name)

where B = an existing MultiBatch Security Class

e.g. `RUN MBATOBJ.UTADDUSR / MBATDAT.DATACONF / MBAT.CONFIG, ALLFACIL`

Info: The ALLFACIL Security Class (which controls access to the various MultiBatch facilities) is pre-configured following an installation, so by adding MBAT.CONFIG via UTADDUSR, this user id will have full access rights.

Alternatively, you may logon as super.super, or as a super.super alias at the PSW screen. The super.super user has global database update access capability and can, in addition, alter the FLAGS settings or force shutdown of any executing MultiBatch BMON environment.

Notes:

1. Each of the users to be added to MultiBatch must have been previously added on the NonStop system as Guardian USER IDs or Safeguard Aliases.

2. In order to logon to the MultiBatch system via an Alias id, the corresponding Safeguard group.user

3. Passwords are validated against the NonStop Safeguard security system and are not retained or stored within the MultiBatch database.

4. Any MultiBatch Registered user can obtain information from the MultiBatch Pathway System. The security features only control access to those facilities which would change any entry in the database, or would affect processing of any runtime element of the MultiBatch environment.

2.4 Startup and Logon

After MultiBatch has been installed and configured via the Installation Macro (see the MultiBatch Installation Guide document), volume to your MultiBatch object subvolume and load the Pathway system by entering the following command:

```
OBEY PWLOAD
```

Once the TACL prompt has returned, you can access the Pathway System by entering:

```
PATHCOM <Pathmon Process>; RUN M10
```

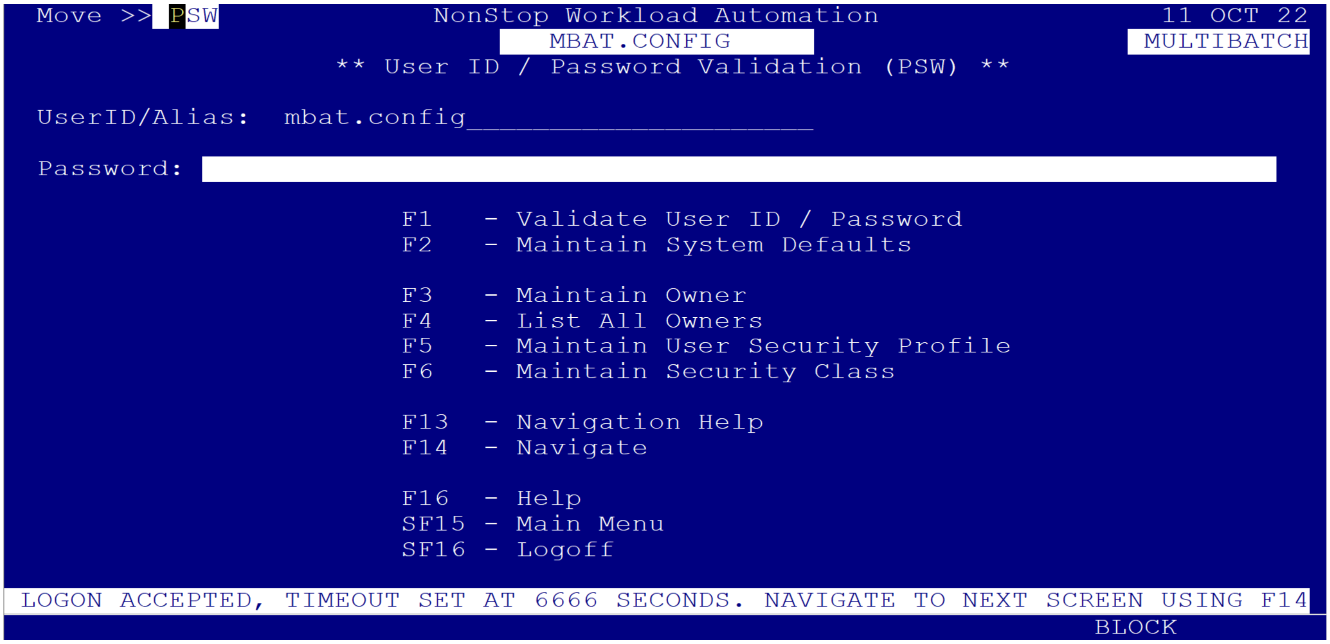
for example:

```
PATHCOM $MBPWY; RUN M10
```

You are then presented with the Password Validation (PSW) screen (Figure 2.1). In order to access the MultiBatch system, you must first logon by entering a valid user name / alias name and password at the Logon Screen and then pressing the F1 function key. If the logon attempt is successful, then the following message is displayed at the bottom of the screen.

```
LOGON ACCEPTED, TIMEOUT SET AT 6666 SECONDS. NAVIGATE TO NEXT SCREEN USING F14
```

This timeout value can be changed by altering the TIME-OUT parameter value for the SRV-ADMIN server class.



```
 Move >> PSW              NonStop Workload Automation                11 OCT 22
                             MBAT.CONFIG                          MULTIBATCH
                        ** User ID / Password Validation (PSW) **

 UserID/Alias:  mbat.config_____________________

 Password:

                      F1   - Validate User ID / Password
                      F2   - Maintain System Defaults

                      F3   - Maintain Owner
                      F4   - List All Owners
                      F5   - Maintain User Security Profile
                      F6   - Maintain Security Class

                      F13  - Navigation Help
                      F14  - Navigate

                      F16  - Help
                      SF15 - Main Menu
                      SF16 - Logoff

LOGON ACCEPTED, TIMEOUT SET AT 6666 SECONDS. NAVIGATE TO NEXT SCREEN USING F14
                                                                  BLOCK
```

Figure 2.1 Password Validation Screen (PSW)

Note: *The Pathway System will prevent you from logging on as one of the following users:*

a) The User Id of the SRV-ADMIN Server.

b) Group User if the SRV-ADMIN Server is running as Group Manager.

Additionally, the SRV-ADMIN server class must not run as SUPER.SUPER.

Note: *The User ID / Alias Name and Password are validated against the Safeguard security system. The password is not displayed when logging in and is not stored in the MultiBatch database.*

The user has two methods available to access the different screens, namely:

1. By using the function keys displayed at the bottom of each screen. These allow the user to perform various functions and move to associated screens.

2. By using the Navigation Facility. This allows the user to access any screen from the current screen by using the three-character identifier of the new screen. This is entered in the “Move” field at the top left-hand corner of the Pathway screen; press the F14 function key to move to the specified screen.

If you are unsure of the screen identifier, press F13 (Navigate Help) for a list of identifiers used to access all of the MultiBatch screens.

The screens accessible to / from the Password Validation Screen, (PSW) are described in Figure 2.2.

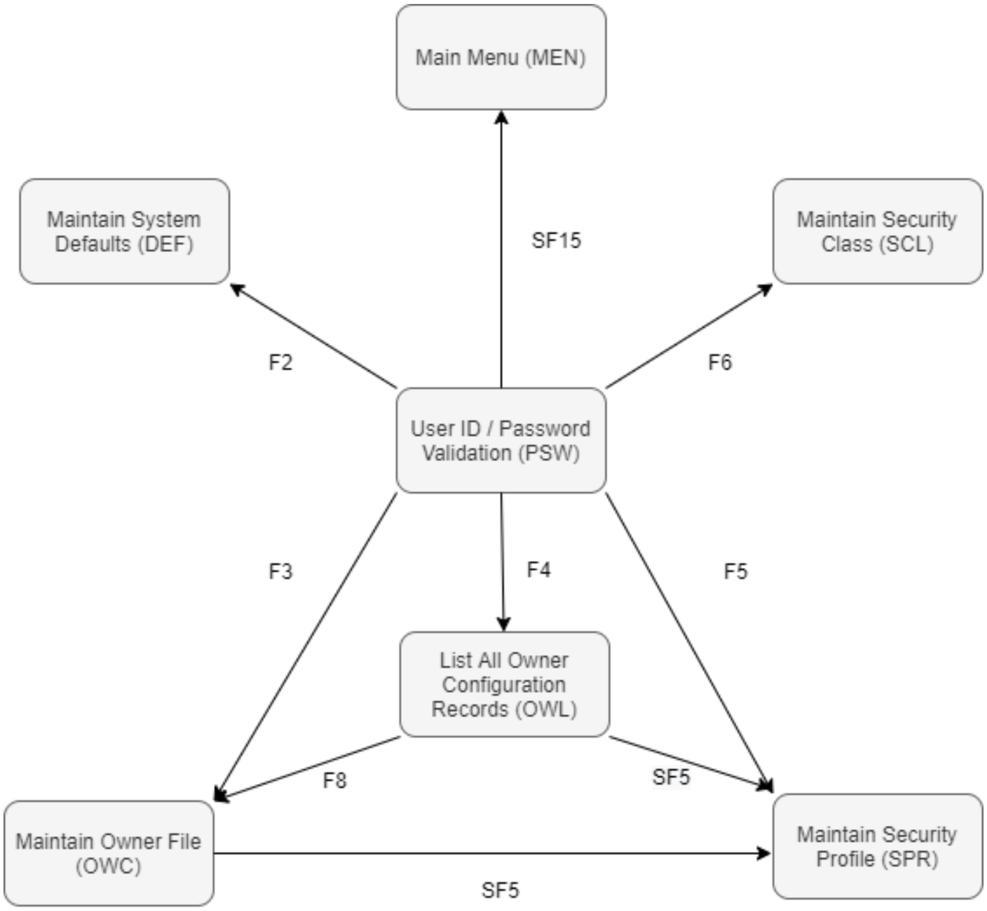
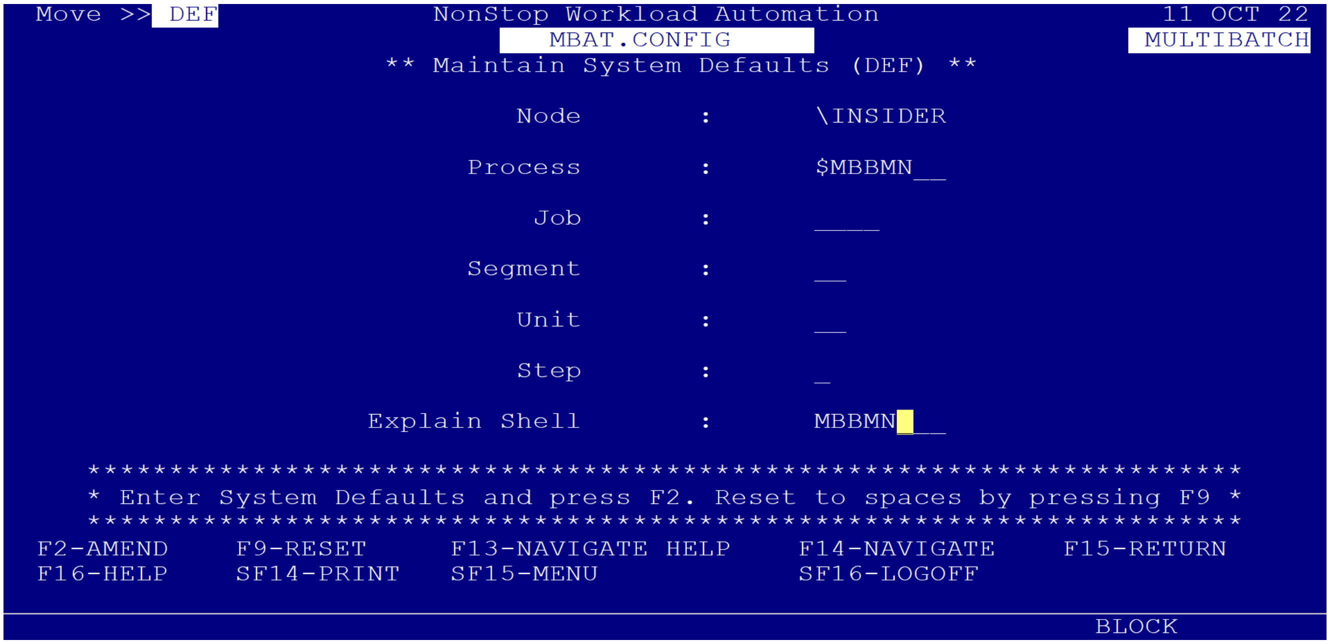

Main Menu (MEN)
Maintain System Defaults (DEF)
SF15
Maintain Security Class (SCL)
F2
F6
User ID / Password Validation (PSW)
F3
F4
F5
List All Owner Configuration Records (OWL)
F8
SF5
Maintain Owner File (OWC)
Maintain Security Profile (SPR)
SF5

Figure 2.2 - Relationships between Pathway Screens

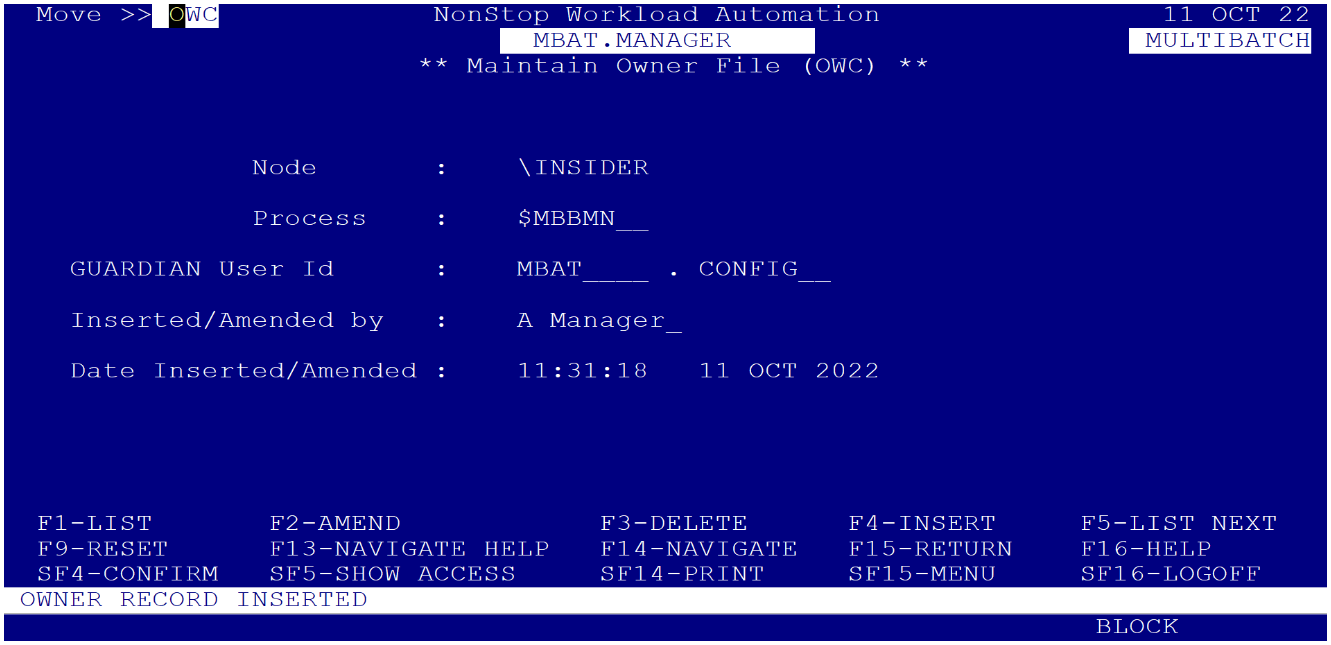
The function keys allow the user to access the following screens:

- Maintain System Defaults (DEF) Screen (Figure 2.3). This allows the user (if authorised) to enter data to be used as default for the current MultiBatch session. This means that you would not need to continually enter the NonStop Node, BMON process name, Job, Segment, Unit, Step names, or Explain Shell values when entering new records into the database; if the relevant fields are available in the selected screen.
- Maintain Owner File (OWC) Screen (Figure 2.4). This screen allows the user (if authorised) to amend a current Owner record or add/delete an Owner record.
- List All Owner Configuration Records (OWL) Screen . This displays a list of all MultiBatch Owners. The user (if authorised) can display the associated Owner File Screen and Security Profile Screen.
- Maintain Security Profile (SPR) Screen (Figure 2.5). This screen defines the Security Classes listed on



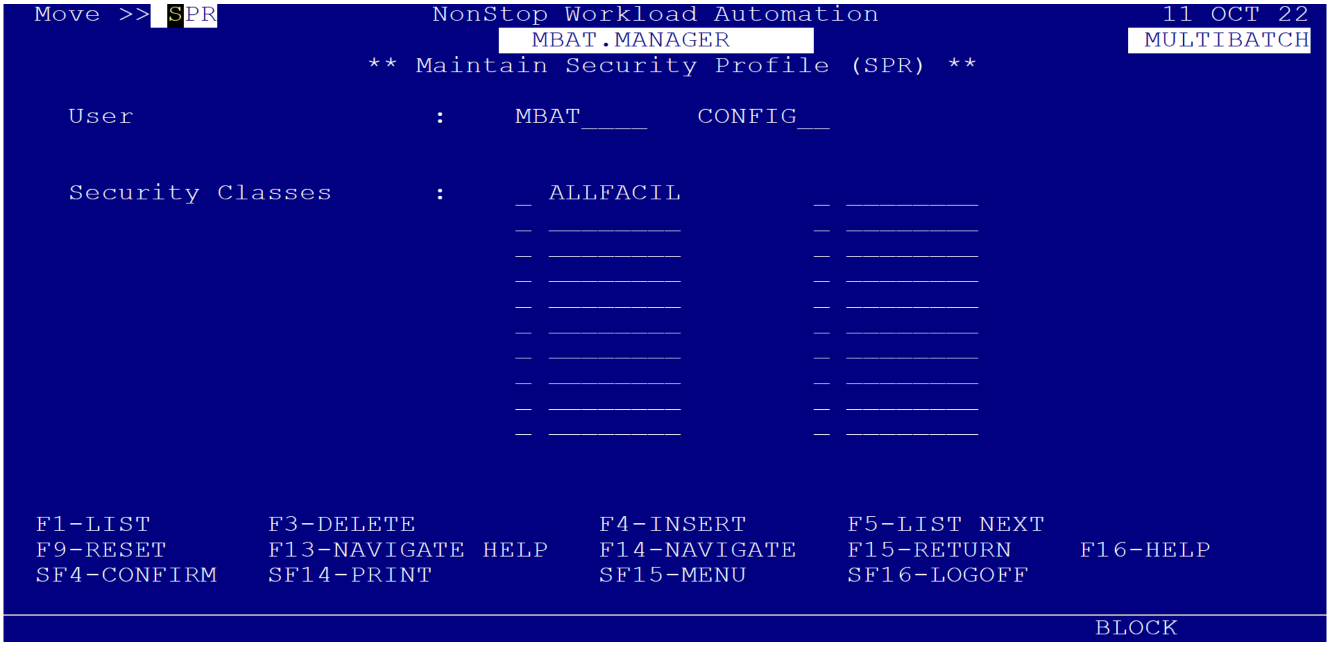
```
 Move >> DEF              NonStop Workload Automation                  11 OCT 22
                            MBAT.CONFIG                                MULTIBATCH
                       ** Maintain System Defaults (DEF) **

                             Node          :       \INSIDER

                          Process          :       $MBBMN__

                              Job          :       _____

                          Segment          :       __

                             Unit          :       __

                             Step          :       _

                    Explain Shell          :       MBBMN___

    *******************************************************************
    * Enter System Defaults and press F2. Reset to spaces by pressing F9 *
    *******************************************************************
 F2-AMEND    F9-RESET    F13-NAVIGATE HELP    F14-NAVIGATE    F15-RETURN
 F16-HELP    SF14-PRINT  SF15-MENU            SF16-LOGOFF

                                                                   BLOCK
```

Figure 2.3 – Maintain System Defaults screen (DEF)



```
 Move >> OWC              NonStop Workload Automation                  11 OCT 22
                            MBAT.MANAGER                               MULTIBATCH
                         ** Maintain Owner File (OWC) **


                  Node          :     \INSIDER

                  Process       :     $MBBMN__

    GUARDIAN User Id            :     MBAT____ . CONFIG__

    Inserted/Amended by         :     A Manager_

    Date Inserted/Amended :     11:31:18   11 OCT 2022



 F1-LIST        F2-AMEND              F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET       F13-NAVIGATE HELP     F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM    SF5-SHOW ACCESS       SF14-PRINT     SF15-MENU      SF16-LOGOFF
 OWNER RECORD INSERTED
                                                                   BLOCK
```

Figure 2.4 – Maintain Owner File screen (OWC)



```
 Move >> SPR              NonStop Workload Automation                  11 OCT 22
                              MBAT.MANAGER                          MULTIBATCH
                         ** Maintain Security Profile (SPR) **

   User                    :    MBAT____    CONFIG__


   Security Classes        :    _ ALLFACIL          _ ________
                                _ ________          _ ________
                                _ ________          _ ________
                                _ ________          _ ________
                                _ ________          _ ________
                                _ ________          _ ________
                                _ ________          _ ________
                                _ ________          _ ________
                                _ ________          _ ________
                                _ ________          _ ________


 F1-LIST        F3-DELETE           F4-INSERT      F5-LIST NEXT
 F9-RESET       F13-NAVIGATE HELP   F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM    SF14-PRINT          SF15-MENU      SF16-LOGOFF

                                                                BLOCK
```

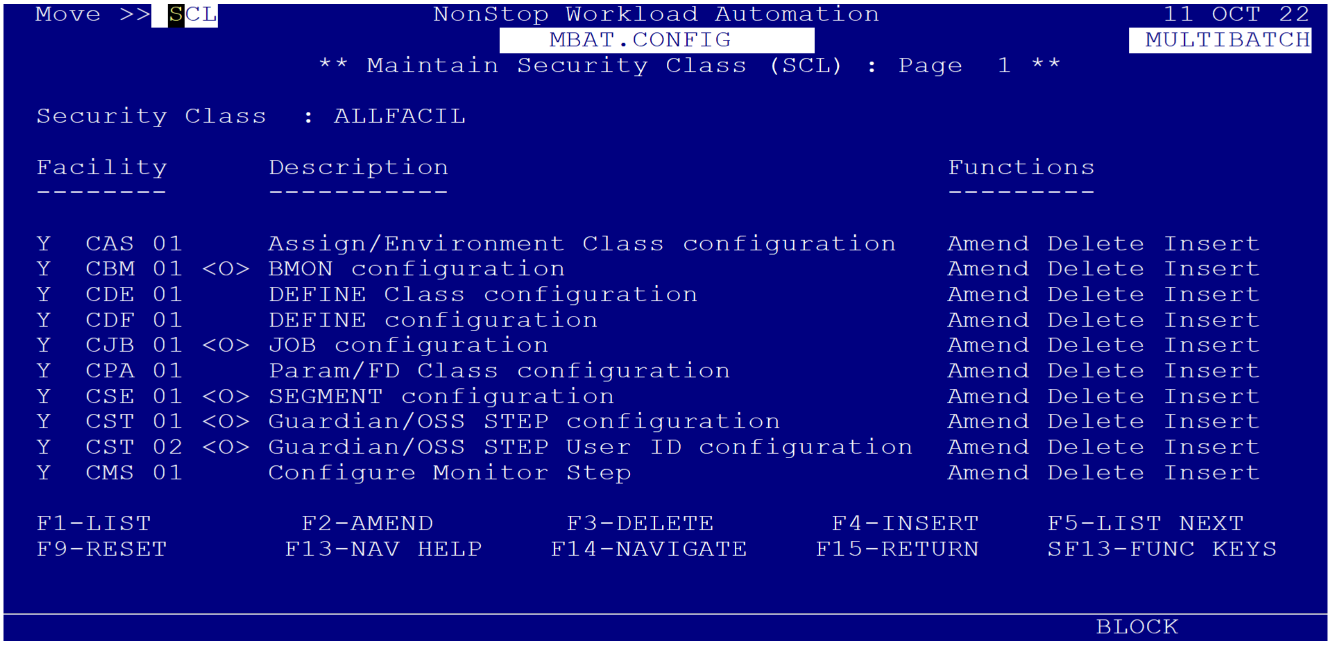
Figure 2.5 – Maintain Security Profile Screen (SPR)

2.5 Building Security Classes (SCL)

Maintain Security Class (SCL) Screen (Figure 2.6).

This screen defines the Security Classes utilised within the Security Profile screen.

The user (if authorised) can amend the access profile available within a class. There are multiple screens to SCL; use SF1, SF2, SF3 function keys to page up / down, or use Page Down / Page Up keys on the keyboard.



```
 Move >> SCL              NonStop Workload Automation                   11 OCT 22
                               MBAT.CONFIG                            MULTIBATCH
                    ** Maintain Security Class (SCL) : Page  1 **

 Security Class   : ALLFACIL

 Facility       Description                                 Functions
 --------       -----------                                 ---------

 Y  CAS 01      Assign/Environment Class configuration      Amend Delete Insert
 Y  CBM 01 <O>  BMON configuration                          Amend Delete Insert
 Y  CDE 01      DEFINE Class configuration                  Amend Delete Insert
 Y  CDF 01      DEFINE configuration                        Amend Delete Insert
 Y  CJB 01 <O>  JOB configuration                           Amend Delete Insert
 Y  CPA 01      Param/FD Class configuration                Amend Delete Insert
 Y  CSE 01 <O>  SEGMENT configuration                       Amend Delete Insert
 Y  CST 01 <O>  Guardian/OSS STEP configuration             Amend Delete Insert
 Y  CST 02 <O>  Guardian/OSS STEP User ID configuration     Amend Delete Insert
 Y  CMS 01      Configure Monitor Step                      Amend Delete Insert

 F1-LIST          F2-AMEND        F3-DELETE        F4-INSERT     F5-LIST NEXT
 F9-RESET        F13-NAV HELP    F14-NAVIGATE     F15-RETURN     SF13-FUNC KEYS


                                                                   BLOCK
```

Figure 2.6 – Maintain Security Class Screen (SCL)

Note: *For the **"CST 02"** entry in the Maintain Security Class screen, the facility can be designated:*

***Y** – allowed.*
***N** – not allowed.*
***P** – allowed with password.*

*The user can Add / Amend / Delete a Job depending on the User Id and the setting of **CST 02** as follows.*

CST 02	**Y**	**N**	**P**
User Id Value			
No User Id	Yes	Yes	Yes
Own User Id	Yes	No	Yes (but with Password)
Other User ID	Yes	No	Yes (but with Password)

*In all cases, changes are only allowed if the associated facility **CST 01** is set to Y.*

MBAT.CONFIG

This user has already been added via the UTADDUSR utility as described in section 2.3 'Add First User' and has full access rights by association with Security Class ALLFACIL.

The primary purpose of this user will be to add new MultiBatch Users / Owners into the Configuration database, and to control the setting of the auditing facility (See Chapter 7). The user will be able to perform any other action within MultiBatch, with the exception of altering the settings of a STEP's FLAGS within the run-time environment, or forcing the shutdown of an executing MultiBatch BMON process.

and all MultiBatch Owners. The user will be allowed to execute the PREPARE phase, but will not be allowed to execute the BUILD phase.

In the following sections you will learn how to configure additional group.user names and associated Security Classes for those such as below:

MBAT.SCHEDULE

The primary purpose of this user will be to run the Prepare and Build phases for any selected MultiBatch Process. The user will have update access to all Scheduling Parameters within the MultiBatch Configuration Database.

MBAT.OPS

The primary purpose of this user will be to monitor and control all run-time MultiBatch environments. This user will have the common set of capabilities to use START, RESTART, HOLD, ABORT and similar commands, as defined for the standard BCOM interface. The user will be able to alter settings of any STEPs FLAGS or to force the shutdown of an executing MultiBatch process.

APPL.OWNER1 / APPL.OWNER2

These users will be registered in the database by MBAT.CONFIG as MultiBatch owner Ids. They will be allowed to administer their own run-time MultiBatch environments by issuing START, ABORT, HOLD, RELEASE, RESTART commands using the PATHWAY interface to BCOM facilities.

Note that MBAT.OPS has additional BCOM command line level security as defined in the Maintain BMON Configuration Screen (CBM), see section 3.3 Configure BMON (CBM). Also see A.3 BMON Environment Commands for the implications of the BMON security setting values on the CBM setting.

Now that user Id MBAT.CONFIG has been added via the use of the UTADDUSR Utility, thereby associating this Guardian User Id with the ALLFACIL Security Class, other users and Security Classes are added via the screen interface.

Note: *You must ensure that a NonStop Guardian User Id exists for these additional MultiBatch users.*

Create a **SECURITY CLASS (SCL)** for a user as follows:

Access the Maintain Security Class Screen (**SCL**) (Figure 2.6) from the Logon screen (PSW) by pressing the **F6** function key.

Enter **SCHEDULE** in the Security Class field.

Enter a **Y** in the first column against **ALL** of the facilities (displayed on multiple pages) as users belonging to this Security Class will have access to all MultiBatch screens / identifiers.

Press the **F4** function key to insert the new class.

Press the **SF15** function key to return to the Logon screen (PSW).

Create a **SECURITY PROFILE (SPR)** for a user as follows:

Access the Maintain Security Profile Screen (**SPR**) (Figure 2.5) from the Logon screen (PSW) by pressing the **F5** function key.

Enter the user name **MBAT.SCHEDULE**

Enter the Security Class **SCHEDULE**

Enter any character by the Class name to mark it.

Press the **F4** function key to insert the new security profile.

Press the **SF15** function key to return to the Logon Screen (PSW).

Once you have BMONs configured, see Chapter 3 Configure MultiBatch, you can register a user as a **MULTIBATCH OWNER (OWC)** as follows:

Access the Maintain Owner File Screen (**OWC**) (Figure 2.4) from the Logon screen (PSW) by

Complete the relevant fields – Node, Process, User ID and inserted / amended by. Press the **F4**

Function key to insert the new owner.

Note: *A user need not be registered as an Owner but some functions accessing BMON related data are only accessible by registered Owners of the BMON.*

Press the **SF15** function key to return to the Logon Screen.

The other MultiBatch users can now be configured in a similar way, e.g., the Security Class **OPERATOR** will be defined for MBAT.OPS. This user will have limited capabilities within MultiBatch but for **all** BMON processes.

Two other users, APPL.OWNER1 and APPL.OWNER2 will be given the **OPERATOR** security class with limited capabilities to operate an environment, but they will only be allowed to make changes for the BMON process $BMN1 and $BMN2 respectively.

The restriction is enforced when registering users on the OWC screen, by entering the appropriate BMON process name in the Process field for each user. Where all BMONs are accessible then the Process on the OWC screen is set to "ALL".

You can now define the following Security Classes via the Maintain Security Class Screen (SCL) as follows:

MultiBatch User	**Security Class Name**
MBAT.CONFIG	CONFIG
MBAT.OPS	OPERATOR
MBAT.SCHEDULE	SCHEDULE
APPL.OWNER1	OPERATOR
APPL.OWNER2	OPERATOR

The "**<O>**" displayed by the name of some screens on the Maintain Security Class (SCL) screens denotes that these facilities can only be accessed by registered MultiBatch Owner, i.e. only the owner or a BMON (see OWC / OWL) can update any records for that BMON.

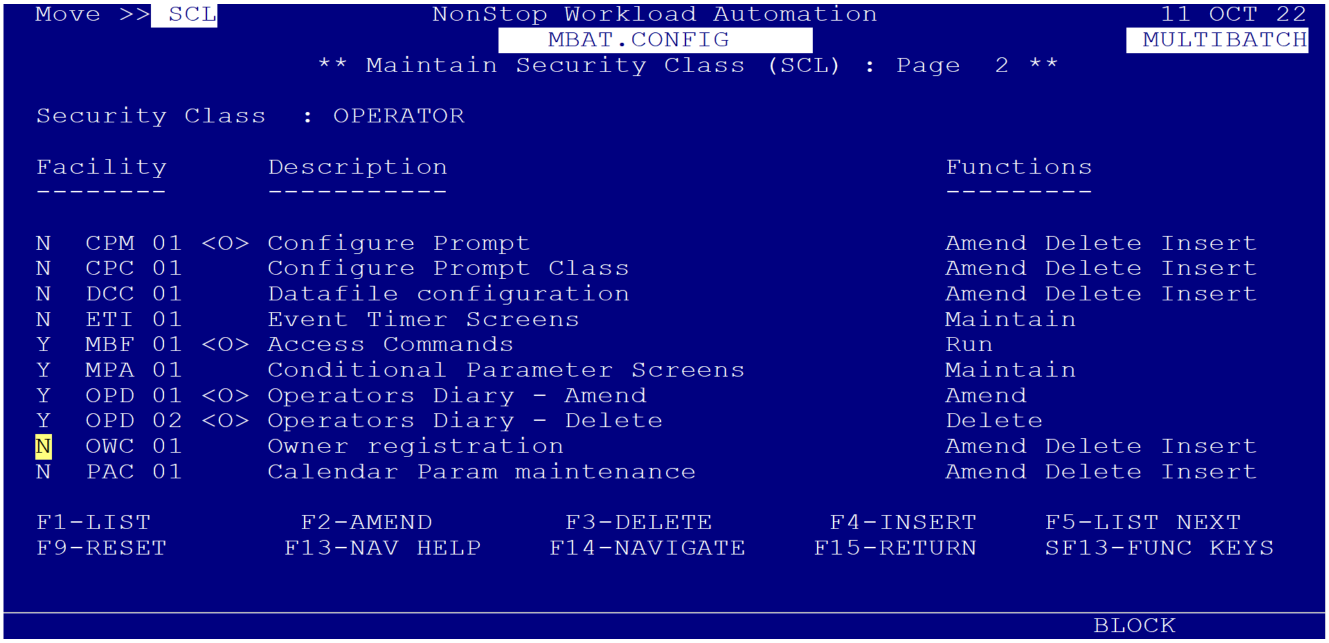
When setting up the Security Classes it is up to the user to determine what facilities are available to each user. Figure 2.7 shows

For example, the CONFIG security class will only be allowed to **AMEND**, **DELETE** or **INSERT** records at those screens with a **Y** in the first column and on the first SCL page. If any member of the CONFIG security class tries to perform anything other than a **READ** access to any other screen, then the following message is displayed at the bottom of the screen:

SECURITY VIOLATION - ACCESS DENIED

Add the OPERATOR security class in the same way as the CONFIG class was added.

The OPERATOR security class will only have access to MultiBatch Facility Interface (MBF) and the Operator's Diary (Amend Only) facilities. Make the relevant **Y** entries in column 1 (See Figure 2.7).



```
 Move >> SCL              NonStop Workload Automation               11 OCT 22
                                  MBAT.CONFIG                      MULTIBATCH
                    ** Maintain Security Class (SCL) : Page  2 **

 Security Class  : OPERATOR

 Facility       Description                              Functions
 --------       -----------                              ---------

 N  CPM 01 <O> Configure Prompt                          Amend Delete Insert
 N  CPC 01     Configure Prompt Class                    Amend Delete Insert
 N  DCC 01     Datafile configuration                    Amend Delete Insert
 N  ETI 01     Event Timer Screens                       Maintain
 Y  MBF 01 <O> Access Commands                           Run
 Y  MPA 01     Conditional Parameter Screens             Maintain
 Y  OPD 01 <O> Operators Diary - Amend                   Amend
 Y  OPD 02 <O> Operators Diary - Delete                  Delete
 N  OWC 01     Owner registration                        Amend Delete Insert
 N  PAC 01     Calendar Param maintenance                Amend Delete Insert

 F1-LIST          F2-AMEND        F3-DELETE       F4-INSERT     F5-LIST NEXT
 F9-RESET        F13-NAV HELP    F14-NAVIGATE    F15-RETURN     SF13-FUNC KEYS

                                                                   BLOCK
```

Figure 2.7 – Maintain Security Class – OPERATOR

2.6 Building Security Profiles (SPR)

At this stage, you should have configured the following Security Classes:

- CONFIG
- OPERATOR
- SCHEDULE
- ALLFACIL *

* **Reminder**: The ALLFACIL Security Class is pre-configured following an installation.

The next step is to link the Security Classes to MultiBatch users to form Security Profiles (a combination of a Security Class associated with a Guardian User ID, becomes the Security Profile).

This will give the System manager more control over a user's access to the various facilities within your MultiBatch environment. The restriction process for each Security Class is governed by the individual class capabilities as defined in the SCL screen.

You may add up to 20 Security Classes for each MultiBatch user. This feature can be used to aggregate together sub-divisions of security capabilities, such that you could combine say the CONFIG and SCHEDULE classes for a special purpose user id.

To set up Security Profiles, select the **F5 – Maintain User Security Profile** option at the Logon screen (PSW).

You can only Insert/Delete a record on this screen by placing a character in the first column to mark the selection and then using the appropriate function key displayed at the bottom of the screen.

2.7 Creating MultiBatch Owners (OWC)

Once the Security Class and Security Profile have been set up for each user they can be registered as MultiBatch Owners via the Maintain Owner File Screen (Figure 2.4). Enter the relevant details in the fields provided and use the appropriate function key to add the new owners.

If the user is to have access to all BMON processes the Node does not need to be specified - entering **ALL** in the Process field is sufficient.

If, however, a user is to have access to several BMON processes then the node must also be specified and a new record is required for each process.

Note: *If you try to add an owner (OWC) before creating a Security Profile for that owner, then the following message is displayed:*

USER NOT REGISTERED ON SECPROFL FILE

Once the users have been registered as Owners, the List All Owners Configuration Screen (See Figure 2.5) can be accessed to check that all have been successfully registered.

2.8 On-line Help Facility (F16)

MultiBatch has an extensive on-line HELP facility.

Information on any MultiBatch screen is obtained by pressing the **F16-HELP** function key on the relevant screen.

Press **F15-Return** to exit the HELP window.

If this is selected at a Menu Screen (e.g. Password Validation) the user is presented with a brief description of the facilities available. On other screens where data input is required, information such as "Valid Values", the number of "Overlays" and a brief description of the fields where data is required is presented.

2.9 Navigation Facility

The Navigation facility is provided to allow the user to move quickly between screens without having to step through the screen via the function keys. This is achieved by entering the appropriate screen identifier in the "MOVE" field at the top left-hand corner of the screen and pressing the **F14** function.

Use of the **F13 Navigate Help** function displays a list of all valid screen identifiers and the associated screen titles, alternatively, entering **NAV** in the "MOVE" field and pressing **F14** function key. The valid identifiers are given in Table 2.1 below.

From now on, the term "Navigate" is used to indicate that a move to the specified screen is to be carried out using the Navigate facility or the appropriate function keys.

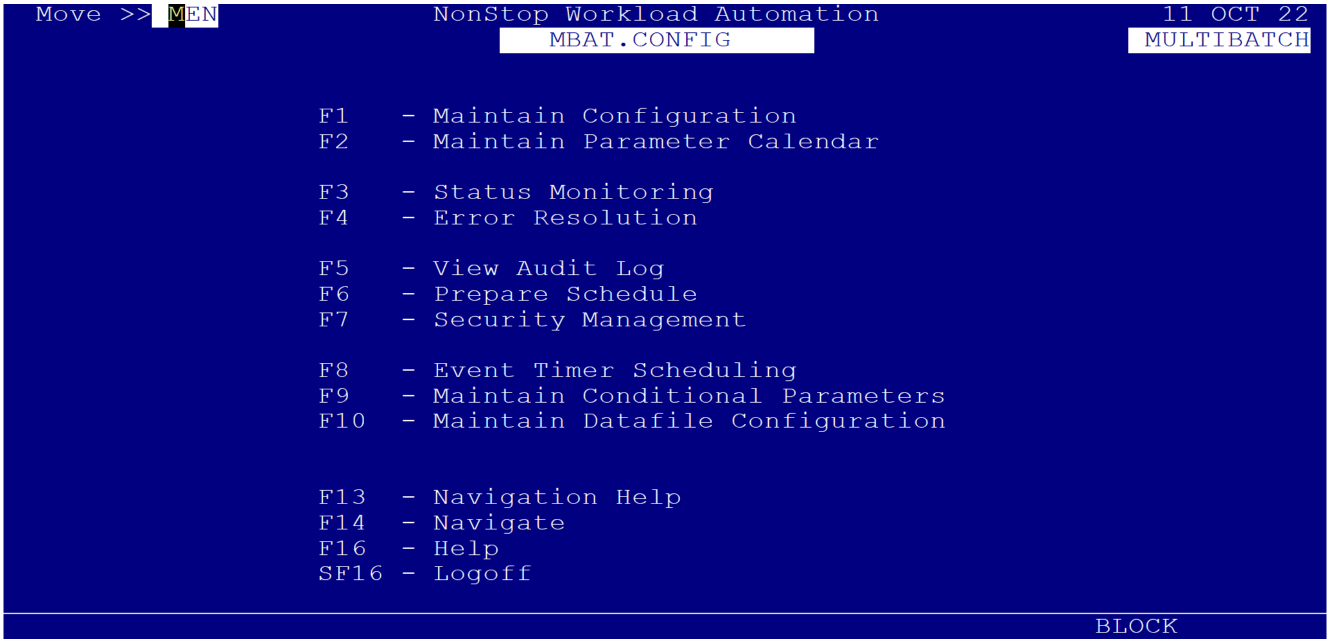
Identifier	Pathway Screen	Identifier	Pathway Screen
ALL	List Audit log	ELM	Event Timer Main List Schedule Menu
ALM	Audit Log Menu	EME	Event Timer Main Menu
CAL	List Assign Class	ERL	List Error Log
CAS	Maintain Assign Class	ERM	Error Log Menu
CBM	Maintain BMON Configuration	LOG	Logoff MultiBatch
CDE	Maintain Define Class	MBF	Facility Interface Menu (Only accessible from J,S,U Overview Screens)
CDF	Maintain Defines	MEN	Main Menu
CDL	List Define Classes	OEL	List Environment Class
CGM	Guardian Step Menu	OFL	Lift File Descriptor Class
CJB	Maintain Job Configuration	OPD	Operations Diary
CLA	List Alias / Wild Card	OWC	Maintain Owner File
CLD	List Defines	OWL	List All Owner Configuration Records
CLE	List Entity	PAC	Maintain Calendar Parameter
CLP	List Process	PAD	List All Parameter Calendar Records by Date
CML	Configure Monitor List	PAL	List All Parameter Calendar Records
CMM	Entity Configuration Menu	PAM	Calendar Maintenance Menu
CMS	Configure Monitor Step	PCL	List Prompt Class
COE	Maintain Environment Class	PGL	List Global Parameters
COF	Maintain File Descriptor Class	PGM	Create Global Parameters
COM	OSS Step Menu	PLL	List Local Parameters
COS	Maintain OSS Step Configuration	PLM	Create Local Parameters
CPA	Maintain Parameter Class	PME	Conditional Parameters Main Menu
CPC	Configure Prompt Class	PML	List Prompts
CPL	List Parameter Class	PRC	Maintain Prepare Configuration
CPM	Configure Prompt	PRL	List All Prepare Configuration Records
CSE	Maintain Segment Configuration	PRM	Prepare Configuration Menu
CST	Maintain Guardian Step Configuration	PSW	User ID / Password Validation
DCC	Maintain Data File Configuration	SCL	Maintain Security Class
DCL	List All Data File Configuration Records	SJB	Job Status Overview
DCM	Data File Configuration Menu	SOC	Current Status
DEF	Maintain System Defaults	SOM	Status Overview Menu
EAT	Create/Update Event Timer AT Schedule	SOV	BMON Status Overview
ECA	Maintain Event Timer Calendar File	SPR	Maintain Security Profile
ECR	Create/Update Event Timer CRONTAB Schedule	SSG	Segment Status Overview
EEV	Create/Update Event Timer EVERY Schedule	SUN	Unit Status Overview
EHM	Maintain Event-Timer Holiday File	VLL	View Loadlist File Records
ELI	Event Timer List Schedules		

Table 2.1 Valid Screen Identifiers

Chapter 3 Configure MultiBatch

3.1 Introduction

The previous chapter dealt with how the user logs onto the MultiBatch system and how the System "security hierarchy" is set up with level of access allowed to various users. This chapter deals with how a user configures their MultiBatch System once the user has logged on. The user has access to the required configuration screens via the Main Menu Screen (MEN) (Figure 3.1). This is accessed from the Logon Screen (See Figure 2.1) by pressing the SF15 function key or by using the MEN screen identifier and the Navigation facility (F14).



```
  Move >> MEN                NonStop Workload Automation                  11 OCT 22
                                 MBAT.CONFIG                              MULTIBATCH


                   F1    - Maintain Configuration
                   F2    - Maintain Parameter Calendar

                   F3    - Status Monitoring
                   F4    - Error Resolution

                   F5    - View Audit Log
                   F6    - Prepare Schedule
                   F7    - Security Management

                   F8    - Event Timer Scheduling
                   F9    - Maintain Conditional Parameters
                   F10   - Maintain Datafile Configuration


                   F13   - Navigation Help
                   F14   - Navigate
                   F16   - Help
                   SF16  - Logoff

                                                                       BLOCK
```

Figure 3.1 – Main Menu Screen (MEN)

The Main Menu provides the user with access to the following facilities:

F1 – Maintain Configuration. This accesses the MultiBatch Entity Configuration Menu (CMM). The options provided on this menu allow the user to Configure the MultiBatch System.

F2 – Maintain Parameter Calendar. This accesses the Calendar Maintenance Menu (PAM) which is used when inputting scheduling information to link dates to parameter names.

F3 – Status Monitoring. This accesses the Status Menu (SOM) which is used to monitor the MultiBatch System (See Chapter 5).

F4 – Error Resolution. This accesses the Error Log Menu (ERM) which is used to monitor any failures within the MultiBatch environment. (See Chapter 6).

F5 – View Audit Log. This accesses the Audit Log Menu (ALM) which is used to look at changes made to records in the database (See Chapter 7).

F6 – Prepare Schedule. This accesses the MultiBatch Prepare Configuration Menu (PRM) which is used to initiate the preparation of MultiBatch schedules (See Chapter 4).

F7 – Security Management. This accesses the Logon Screen (PSW) which is used to set up security access to the MultiBatch facilities (See Chapter 2).

F8 – Event Timer Scheduling. This allows the user to access the Event Timer Menu (EME). The user can use this facility to create a database of start times for nominated MultiBatch jobs (See Chapter 8).

F9 – Maintain Conditional Parameters. This accesses the Maintain Conditional Parameter screen (PME).

interpreter (e.g. <Disk> becomes $D01).

F10 – Maintain Datafile Configuration. This accesses the Maintain Datafile Configuration Menu (DCM). This is used to define the physical locations for all the data files in the MultiBatch system and the location of Object Code file names for processes which the system must start and interact with.

3.2 PWCONF FILE

The "PWCONF" Pathway configuration file, contains commands for the MultiBatch Pathway environment. It contains configuration commands for PATHWAY, TCP, PROGRAM and SERVER entities.

For each of these entities, process name and disk/subvolume references may have been changed by the user during the installation process to reflect their own system. Additionally, server PARAM values may have been changed.

The Pathway servers handle input from the MultiBatch screens. These servers are listed below, together with any associated parameters and screen mnemonics of the screens they serve.

EVENT-TIMER - see 'Chapter 8 Event Timer' for a full description on the functionality of Event Timer (EME) and the various screens and parameters this server class utilises.

SRV-ADMIN - handles input from the following screens: DCC, DCL, OWC, OWL, PSW, SCL, SPR. The following parameters are associated with this server:

> **TIMEOUT** - This is the time, in seconds, after which an inactive session will be logged out.
>
> **AUDIT-FLAG** - This determines the type of audit trail recorded and can be set to one of the following:
>
> | | |
> |---|---|
> | **N** | No audit details recorded. |
> | **S** | Summary details recorded. |
> | **Y** | Full audit details recorded. |
>
> **EXPAND-NAMES -** this parameter is set to **Y** or **N.** If it is set to Y, then any filename provided by the user is expanded to its full description, that is, Node, Disk, Subvolume and File using the current defaults
> (SET SERVER VOLUME). If Conditional Parameter (PME) processing is being performed, then this parameter **MUST** be set to **N**, otherwise, "illegal filename" errors will be encountered when inserting or amending Steps and Defines.
>
> **TITLE** - this is an optional parameter which consists of up to eight characters to identify the Pathway environment. If it is used, then the "title" appears in the top-right corner of the PSW screen. It is used to differentiate between environments when more than one is running at the same time. For example, MULTIBATCH, LIVE, TRAINING, etc.

SRV-AUDSRV - handles input from the ALL screen. There are no parameters associated with this server.

SRV-BMON-DETAIL - handles input from the MBF screen. There are no parameters associated with this server.

SRV-CONDITPARAM - handles the Shift/F11 "explain" processing from a number of screens**.**

SRV-CONFIG - handles input from the following screens: CAL, CAS, CBM, CLA, CLE, CLP, CJB, CML, CMS, COS, CPA, CPL, CSE, CST. The AUDIT-FLAG and EXPAND-NAMES parameters are associated with this server.

SRV-CONFIG-OSS - handles input from the following screens: COE, COF, COS, OEF, OEL. The AUDIT-FLAG and EXPAND-NAMES parameters are associated with this server.

SRV-CURR-STAT - handles input from the SOC screen. There are no parameters associated with this server.

SRV-DEFINE - handles input from screens CDE, CDF, CDL and CLD. The AUDIT-FLAG parameter is associated with this server.

SRV-ERRSRV - handles input from the ERL and OPD screens. The AUDIT-FLAG and EXPAND-NAMES are associated with this server; these are as previously described.

SRV-PARAM - handles input from the following screens: PAC, PAD, PAL, PRC, PRL. The AUDIT-FLAG and EXPAND-NAMES are associated with this server; these are as previously described.

SRV-PROMPT - handles input from the CPC, CPM, PCL, PML screens. There are no parameters associated with this server.

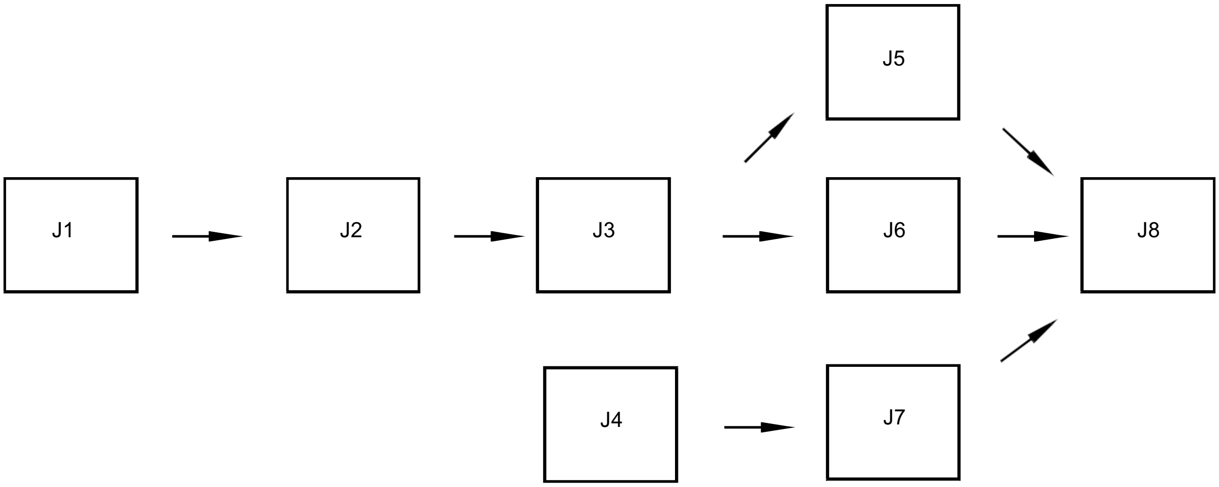
SRV-STATUS - handles input from the following screens: SJB, SOV, SSG and SUN. There are no parameters associated with this server.

SRV-WILDCARD - handles input from the CLA screen. There are no parameters associated with this server.

SYSTEM-HELP - provides help syntax to all Pathway screens via use of Function Key F16. There are no parameters associated with this server.

3.3 Configure BMON (CBM)

In order to demonstrate how easily the MultiBatch system can be configured (i.e. configure Jobs, set dependencies, etc.), the sample schedule shown in Figure 3.2 is used in this chapter.



J1, J2, J4, J5, J6 and J8 are to be run DAILY and J3 and J7 are to be run only at WEEKENDS.
J1 to J8 are to be configured into $BMN1 as jobs JOB1 to JOB8, as job names must consist of 4 characters.
For simplicity each JOB has only one Segment / Unit.

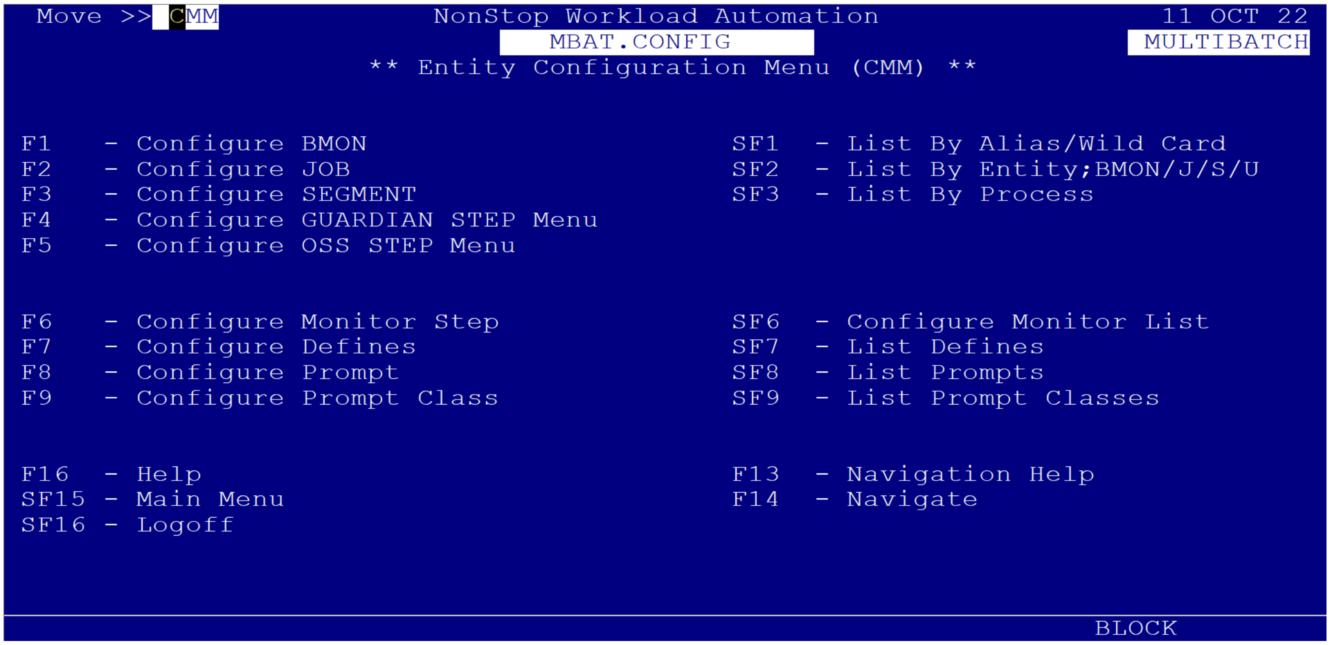
Figure 3.2 – Sample Job Schedule

Note: *You may wish to set the defaults using the* ***DEF*** *screen to enter the node name, BMON process name, etc.*

For this example, the user should logon as MBAT.CONFIG so that the user has security access to all of the MultiBatch facilities via the use of the ALLFACIL Security Class.

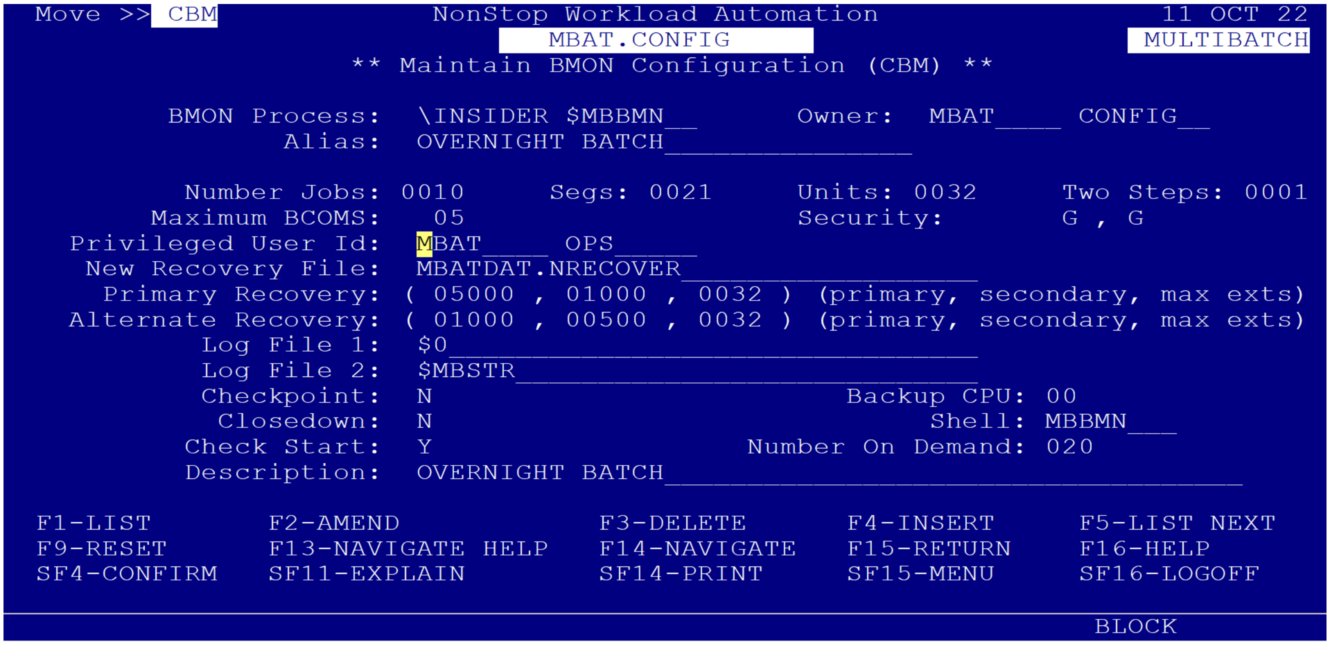
Once the Main Menu Screen is accessed, press the **F1** Maintain MultiBatch Configuration to display the Entity Configuration Menu (CMM) screen (See Figure 3.3).

This menu is used throughout this chapter to configure the entire MultiBatch System. Select the **F1** used to configure the current BMON process.



```
 Move >> CMM              NonStop Workload Automation                 11 OCT 22
                                 MBAT.CONFIG                         MULTIBATCH
                      ** Entity Configuration Menu (CMM) **


F1    - Configure BMON                          SF1   - List By Alias/Wild Card
F2    - Configure JOB                           SF2   - List By Entity;BMON/J/S/U
F3    - Configure SEGMENT                       SF3   - List By Process
F4    - Configure GUARDIAN STEP Menu
F5    - Configure OSS STEP Menu


F6    - Configure Monitor Step                  SF6   - Configure Monitor List
F7    - Configure Defines                       SF7   - List Defines
F8    - Configure Prompt                        SF8   - List Prompts
F9    - Configure Prompt Class                  SF9   - List Prompt Classes


F16   - Help                                    F13   - Navigation Help
SF15  - Main Menu                               F14   - Navigate
SF16  - Logoff




                                                                        BLOCK
```

Figure 3.3 – Entity Configuration Menu Screen (CMM)



```
 Move >> CBM              NonStop Workload Automation                 11 OCT 22
                                 MBAT.CONFIG                         MULTIBATCH
                     ** Maintain BMON Configuration (CBM) **

         BMON Process:  \INSIDER $MBBMN__      Owner:  MBAT____  CONFIG__
                Alias:  OVERNIGHT BATCH_______________

          Number Jobs: 0010     Segs: 0021     Units: 0032     Two Steps: 0001
        Maximum BCOMS:   05                    Security:        G , G
   Privileged User Id:  MBAT____  OPS______
    New Recovery File:  MBATDAT.NRECOVER__________________
     Primary Recovery: ( 05000 , 01000 , 0032 ) (primary, secondary, max exts)
   Alternate Recovery: ( 01000 , 00500 , 0032 ) (primary, secondary, max exts)
           Log File 1:  $0____________________________________
           Log File 2:  $MBSTR________________________________
           Checkpoint:  N                         Backup CPU: 00
            Closedown:  N                              Shell: MBBMN___
          Check Start:  Y                   Number On Demand: 020
          Description:  OVERNIGHT BATCH___________________________________

 F1-LIST       F2-AMEND            F3-DELETE     F4-INSERT     F5-LIST NEXT
 F9-RESET      F13-NAVIGATE HELP   F14-NAVIGATE  F15-RETURN    F16-HELP
 SF4-CONFIRM   SF11-EXPLAIN        SF14-PRINT    SF15-MENU     SF16-LOGOFF


                                                                        BLOCK
```

Figure 3.4 – Maintain BMON Configuration Screen (CBM)

Use this screen to input the following details:

The BMON Process and Owner. The BMON process name will be displayed if the defaults screen has been used.

The **Owner** needs to be set up by the user and it must have been registered in the “Maintain MultiBatch Owner” screen (OWC). See Chapter 2.

Alias. This allows the user to input a meaningful name which is linked to the Job, Segment or Unit. This is very useful for identifying Jobs etc., which are limited to a four-character name.

Maximum BCOMS. This is used to specify the maximum number of BCOMs that can access BMON at any one time. Defaults to 5.

user. These facilities are available for use only on that BMON where the Privileged User is defined.

New Recovery File. The file specified in this field is created when the START BMON command is executed. The file can be used in future runs of BMON to recover BMON to the state it was in when it stopped.

Note: *The maximum fileset value accepted by MultiBatch is 34 characters.*

Primary Recovery. BMON records the progress of its schedule to a TMF audited recovery and alternate recovery files. A user can accept the default Primary, Secondary and Max Extents, or specify their own values.

Alternate Recovery. This is the Alternate Key file for the Primary Recovery file. A user can accept the default Primary, Secondary and Max Extents, or specify their own values.

See Appendix H "Recovering a BMON" for further details.

Log File 1, Log File 2. These are log files which record BMON activity. A value for LOG1 is mandatory and both or either can be a terminal, printer, disk file, or process located anywhere in the NonStop Network.

It is highly recommended, however, that one of them is the MultiBatch Streamer Process (See Appendix G). This is required if Event Timer(EME) is to function and to ensure that the status database (SOM, SOV, SOC, etc.) is updated.

Note: *If the user is going to log to a separate disk file, then the file must already exist. To create a file of the correct type and size, enter the following at the TACL prompt:*

```
FUP CREATE <LOG-FILE>, TYPE E, REC 200, MAXEXTENTS <size>
```

Security. This referenced to NonStop Guardian file security and indicates which commands can be performed by various classes of user.

Any BCOM Owner can obtain status information from BMON, however, there are two other classes <x, y> which are entered in this field. These two classes are as follows:

x- this class relates to configuration changes to BMON and the commands are ADD, DELETE, ALTER STEP and FLAGS

y - this class relates to all other commands such as START JOB, HOLD SEGMENT, RESTART UNIT, ABORT UNIT and SHUTDOWN BMON (with or without !).

These two classes "**x**" and "**y**" can both have any of the following Guardian File security values:

"-"	SUPER.SUPER Access only.
"**O**"	Local, non-networked Owner-only Access.
"**U**"	Local or Networked User (Owner) only Access.
"**G**"	Local only, Group-wide Access.
"**C**"	Local or Networked, Group-wide Access
"**A**"	Any Local User.
"**N**"	Any Local or Network User.

For more details see the discussion regarding the BMON SET SECURITY command in section A.3 BMON Environment Commands

Checkpoint. This instructs the current BMON to start a backup process in the specified CPU and to

NonStop process pair. If the CPU specified is unavailable, then the highest available CPU is used.

Note: *If the user is going to utilise Checkpoint and a Backup CPU value, then the TACL BMON RUN command must include a <CPU Number> value. For example, CBM has Backup CPU set to 00 and the RUN BMON command has CPU 1 within its runtime params:*

RUN MBATOBJ.BMONL / ***CPU 1****, NAME $BMON, NOWAIT, TERM $ZHOME /*

Closedown. This field specifies whether or not BMON is to shut down when the number of jobs completed equals the number of jobs scheduled.

Checkstart. This field specifies whether or not checking is to be performed when a "Restart Unit" or "Start Job / Segment / Unit" is requested. If it is set to Y, then:

For a "restart" it checks if the unit has previously run.

For a "start", it checks for PREV and NEXT dependencies. If a Job/Segment has previous dependencies that have not completed or next dependencies that have already started, then the user is asked to confirm the requested actions before they are carried out.

Number On Demand. This value is used to allocate space in BMON for On Demand Jobs, see section see Chapter 9 On Demand Jobs. The value should be at least the expected number of concurrent On Demand Jobs for the BMON.

Shell. The Explain Shell value should be supplied if you wish to use the Explain function key SF11 to display the value of a Local or Global Conditional parameter (PME).

Description. This field is for information only and allows the user to give the BMON a descriptive reference, e.g. "End of Day Processing".

To configure the BMON Process for a MultiBatch Owner, e.g. APPL.OWNER1, enter the details displayed in Figure 3.4 and press the F4 function key. This defines the process $BMN1 belonging to the MultiBatch Owner APPL.OWNER1.

Once the **F4** function key is selected the system cross checks that the specified MultiBatch process belongs to the user entered in the Owner field. If this is not the case, then the following error message is displayed:

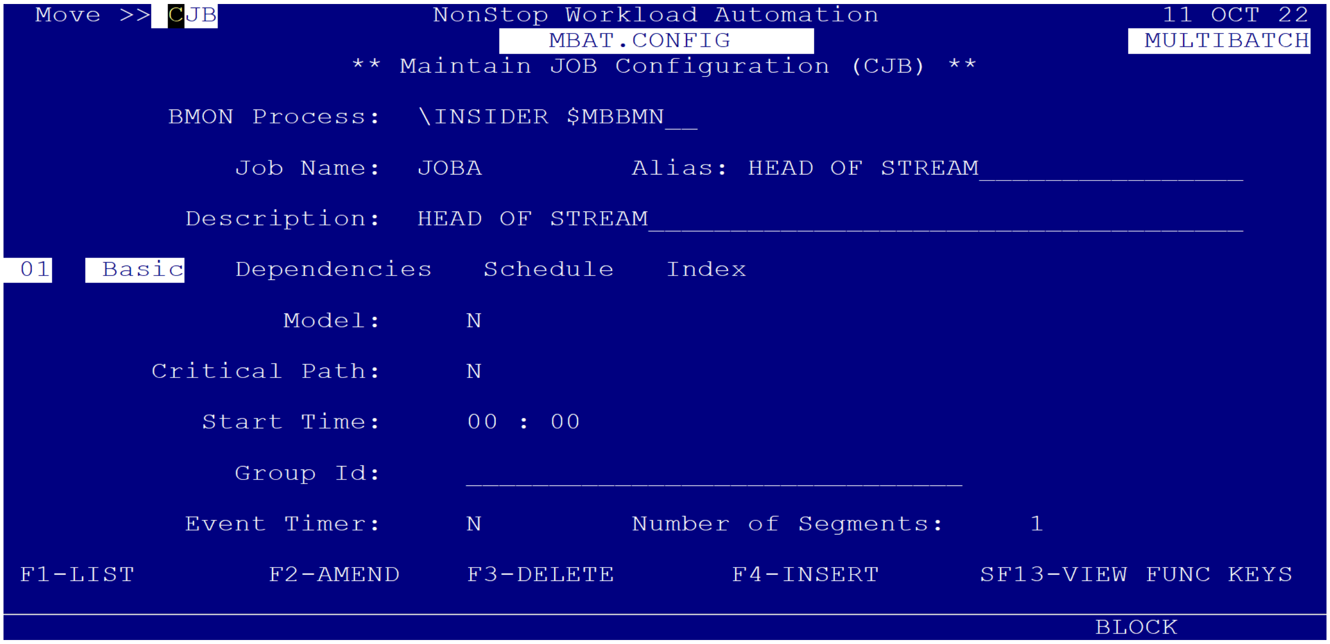
OWNER NOT REGISTERED ON OWNER FILE

Note: *If this occurs the user must be added to the OWC screen as described in Chapter 2.*

Note: *If you are still unsure of any of the fields on this screen, press the Help function key (F16) for a description of each field.*

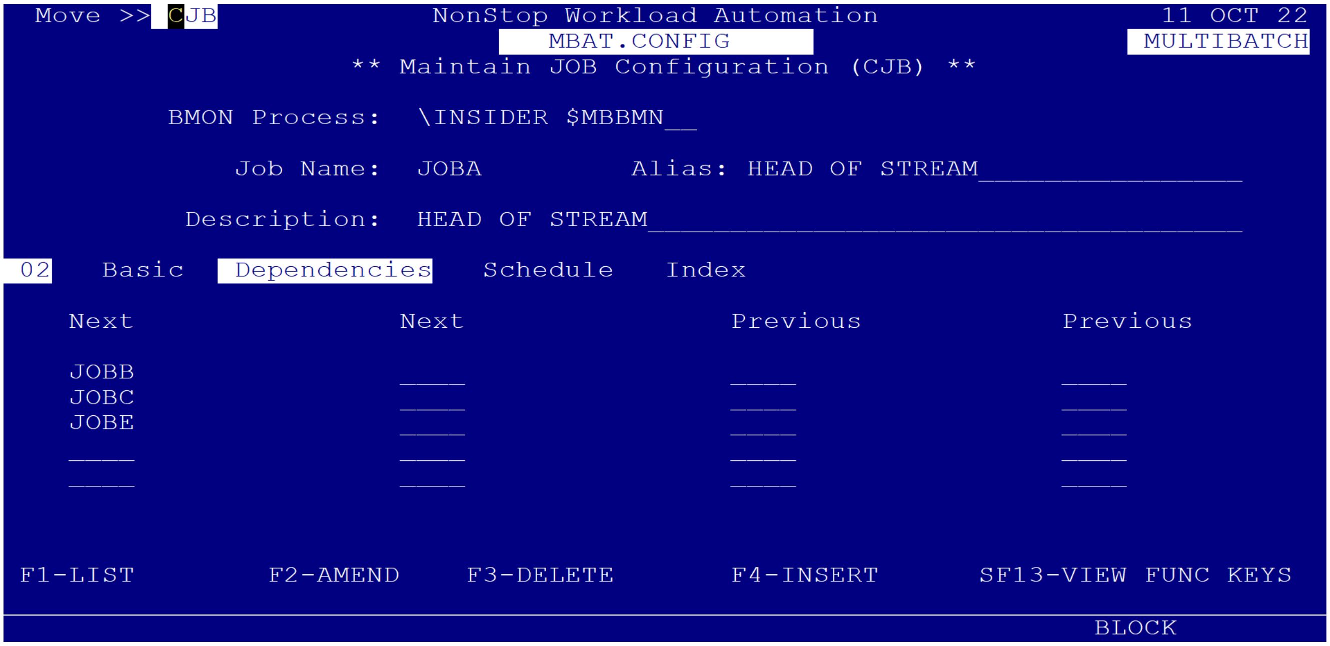
3.4 Configure JOB (CJB)

To configure JOBs within the MultiBatch environment, navigate to the Maintain Job Configuration Screen (CJB) (See Figure 3.5). This screen comprises four sub-screens or overlays.



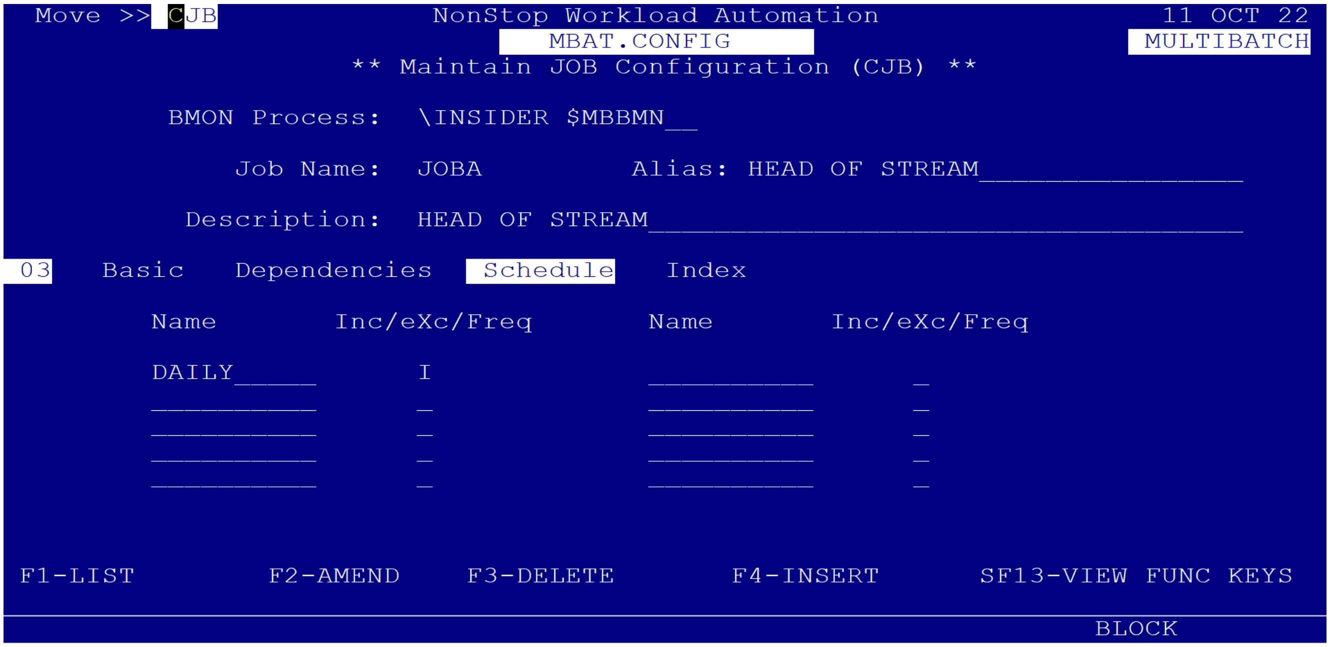
```
 Move >> CJB                    NonStop Workload Automation              11 OCT 22
                                   MBAT.CONFIG                          MULTIBATCH
                           ** Maintain JOB Configuration (CJB) **

          BMON Process:   \INSIDER $MBBMN__

              Job Name:   JOBA          Alias: HEAD OF STREAM________________

           Description:   HEAD OF STREAM____________________________________

 01    Basic     Dependencies    Schedule    Index

                 Model:      N

         Critical Path:      N

            Start Time:      00 : 00

              Group Id:      ______________________________

           Event Timer:      N          Number of Segments:     1

 F1-LIST        F2-AMEND     F3-DELETE        F4-INSERT       SF13-VIEW FUNC KEYS

                                                                          BLOCK
```

Figure 3.5 – Maintain Job Configuration Screen (overlay 1 of 4)



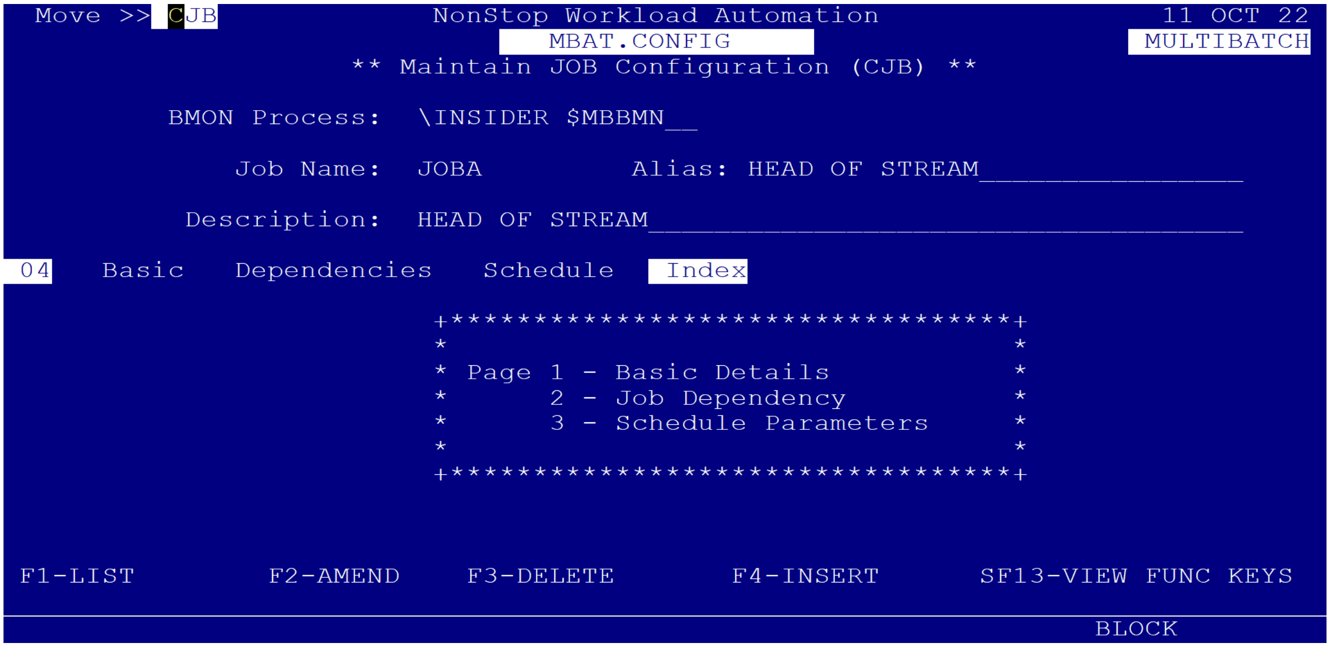
```
 Move >> CJB                    NonStop Workload Automation              11 OCT 22
                                   MBAT.CONFIG                          MULTIBATCH
                           ** Maintain JOB Configuration (CJB) **

          BMON Process:   \INSIDER $MBBMN__

              Job Name:   JOBA          Alias: HEAD OF STREAM________________

           Description:   HEAD OF STREAM____________________________________

 02    Basic    Dependencies    Schedule    Index

    Next                Next                Previous            Previous

    JOBB                ____                ____                ____
    JOBC                ____                ____                ____
    JOBE                ____                ____                ____
    ____                ____                ____                ____
    ____                ____                ____                ____


 F1-LIST        F2-AMEND     F3-DELETE        F4-INSERT       SF13-VIEW FUNC KEYS

                                                                          BLOCK
```

Figure 3.5 – Maintain Job Configuration Screen (overlay 2 of 4)



```
 Move >> CJB            NonStop Workload Automation                 11 OCT 22
                              MBAT.CONFIG                         MULTIBATCH
                   ** Maintain JOB Configuration (CJB) **

         BMON Process:  \INSIDER $MBBMN__

             Job Name:  JOBA          Alias: HEAD OF STREAM________________

          Description:  HEAD OF STREAM____________________________________

03    Basic   Dependencies   Schedule    Index

        Name        Inc/eXc/Freq        Name        Inc/eXc/Freq

        DAILY_____       I              __________       _
        __________       _              __________       _
        __________       _              __________       _
        __________       _              __________       _
        __________       _              __________       _


F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                 BLOCK
```

Figure 3.5 – Maintain Job Configuration Screen (overlay 3 of 4)



```
 Move >> CJB            NonStop Workload Automation                 11 OCT 22
                              MBAT.CONFIG                         MULTIBATCH
                   ** Maintain JOB Configuration (CJB) **

         BMON Process:  \INSIDER $MBBMN__

             Job Name:  JOBA          Alias: HEAD OF STREAM________________

          Description:  HEAD OF STREAM____________________________________

04    Basic   Dependencies   Schedule    Index

                        +***********************************+
                        *                                   *
                        * Page 1 - Basic Details            *
                        *      2 - Job Dependency           *
                        *      3 - Schedule Parameters      *
                        *                                   *
                        +***********************************+


F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                 BLOCK
```

Figure 3.5 – Maintain Job Configuration Screen (overlay 4 of 4)

Each overlay comprises a fixed “top half” display which contains the following information:

The BMON Process Name.

The Job Name.

The Alias (if used).

The Job Description (if used).

The Overlay currently displayed (number and name are highlighted).

The information supplied in the bottom half of each overlay is dependent on the overlay. The following overlays are provided:

Overlay 1. This is the Basic overlay and contains the following fields:

Jobs. Once set when a job is created, this field cannot be amended.

Critical Path. This field allows the user to specify whether or not the Job lies on a Critical Path. If this is the case, then this information is taken into account when preparing reports and presenting the status of the current BMON processes. In addition, extra diagnostic messages are generated on failure. There is no processing functionality - it is purely used for reporting.

Start Time. This allows the user to specify when the Job is to start. This is sometimes not specified (i.e. left at 00.00 for no start time; 24:00 is midnight) as only the start time for the first job is required with the others following on in sequence. It is sometimes used to specify that a Job is not to run until the time specified even if all of the previous Jobs are complete. Alternatively, the user can specify start times via the Event Timer facility (EME). See Chapter 8 Event Timer for further details.

Note: *The CJB Start Time value can also be amended via the RESCHDST Utility (see Appendix F). This is executed via a TACL session and saves the user having to logon to the MultiBatch Pathway in order to amend the CJB entry (or multiple entries). It can be useful for NonStop system clock changes.*

Group Id. This allows jobs to be grouped under a "group name". All jobs within a Group are re-runnable jobs, that is, they are reset to a configured (ready to run %000000) state when complete. Only the first Job can be started and this can only be done when all other Jobs in the group have been completed.

Note: **Group Id** is restrictive: All jobs must be in the same group; only job level dependencies; only one segment per job; only one unit per segment; cannot start a job in the group if it is already active; only a single job dependency in the flow, e.g. JOBA > JOBB > JOBC > JOBD.

Event Timer. This is a dynamic field and will be automatically updated when the job has been added to Event Timer (EME).

Number of Segments. This is a protected field which shows the user how many segments have been configured for the specified Job.

Overlay 2. This is the **Dependencies** overlay and displays the dependency information for the current Job. The Next columns are protected fields; information is displayed in these columns as the user adds dependencies for the Job in the Previous column.

Overlay 3. This is the **Schedule** overlay and displays information indicating when a Job is to be run (which date / day).

The user has access to two fields:

Name. This can be of three types:

Absolute. This type is a member of a closed set consisting of reserved words known to the MultiBatch system. These are: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY, WEEKDAY, WEEKEND, DAILY and NEVER.

Calendar. These can possess an arbitrary name which is related to a set of user-defined dates contained in the MultiBatch Calendar Parameter maintenance (PAM) database.

Frequency. This is a user derived name, e.g. ADHOC and if specified, then the Prepare screen (PRC) can reference the Frequency Name so that the job is selected for the schedule.

Inc/e**X**c/**F**req. This field is used in conjunction with the Name field to specify the frequency (e.g. MONDAY and **X** causes the Job to be run on every day **eXcept** Monday. Combinations can be used for the same job to provide a more "complicated" schedule (e.g. **I**nc MONDAY and **I**nc THURSDAY causes the Job to Run on both Mondays and Thursdays).

For a new Job, the required fields are automatically completed with the default values "Daily" and "I" which

Overlay 4. This is the **Index** overlay and provides the user with a list of the overlays available.

Note: *The different overlays can be accessed in one of two ways, which are:*

Use of the page up and page down keys.

Enter the number of the overlay in the "overlay number" field and press page down. With b) you could go directly from overlay 1 to 4 in one move.

With the above information about the overlays available, we will now configure JOB3 from Figure 3.2 as follows:

Overlay 1 – Basic.

Enter **N** for the Model field.

Enter **Y** in the Critical Path field.

Enter no start time (leave it at 00:00) as a start time will be specified for **JOB1**.

Press the **F4** function key to insert the new information and create a new record in the database.

Note: *The Number of Segments field is 0. This will be automatically incremented to 1 when the associated segment is added (and is incremented each time a segment is added).*

Overlay 2 – Dependencies.

Enter JOB2 in the Previous column. The Next column will be automatically filled in when Jobs 5 and 6 are added.

Press the **F2** function key to amend the record.

Overlay 3 – Schedule.

As JOB3 is to run only at weekends it can be specified in several ways, namely:

Entering WEEKEND and I (include).

Entering SATURDAY and I, plus SUNDAY and I.

Entering WEEKDAY and X (exclude).

Using the 'frequency type' by entering ADHOC01 (any name will do) and F.

Use the final option to demonstrate the Frequency type. This is explained in detail in the Scheduling section later in this chapter.

Press the **F2** function key to amend the record.

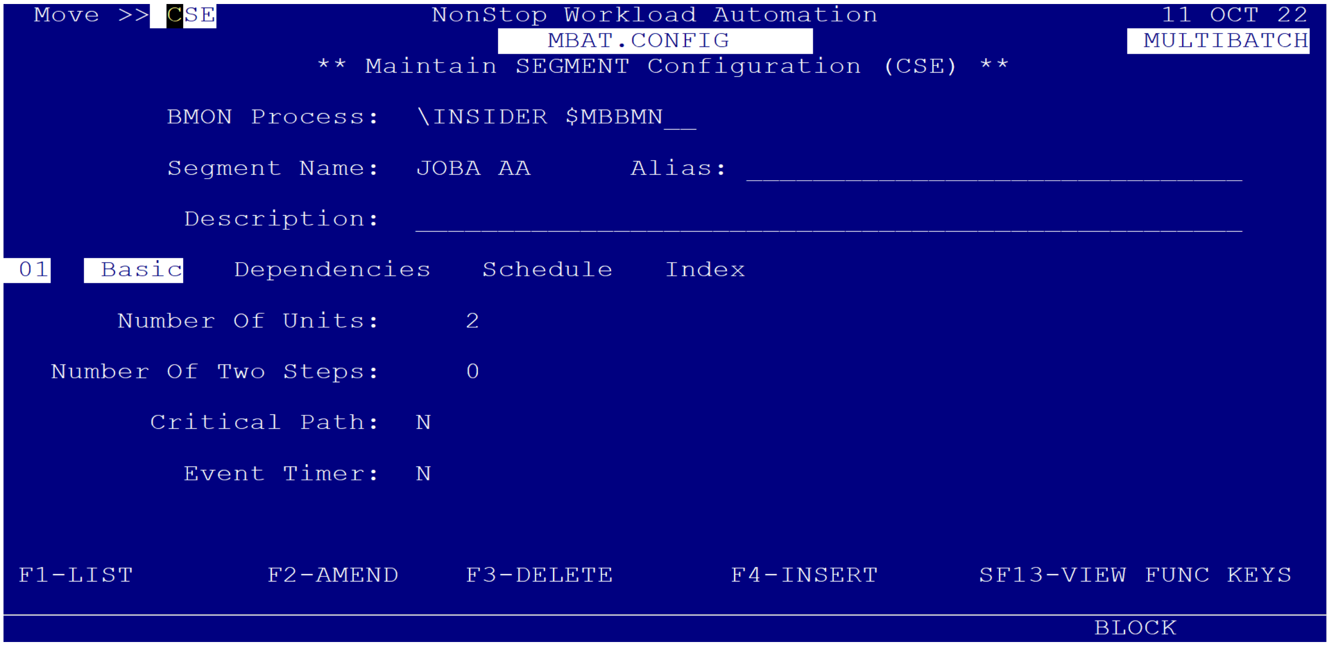
Once JOB3 has been successfully configured the process can be repeated for the other jobs until all have been successfully configured into the BMON process.

Note: *When configuring Jobs, it is usual to generate the overlays in sequence, that is, enter the Basic overlay for **all** the Jobs before moving on to the next overlay. Pressing the F4 function key on the first overlay creates the record; this is then amended on subsequent overlays using the F2 function key. All the overlay information for a given Job can be input together and then the F4 function key pressed to create the record; as all the information has been input in one go, the F2 Amend function key is not required.*

3.5 Configure SEGMENT (CSE)

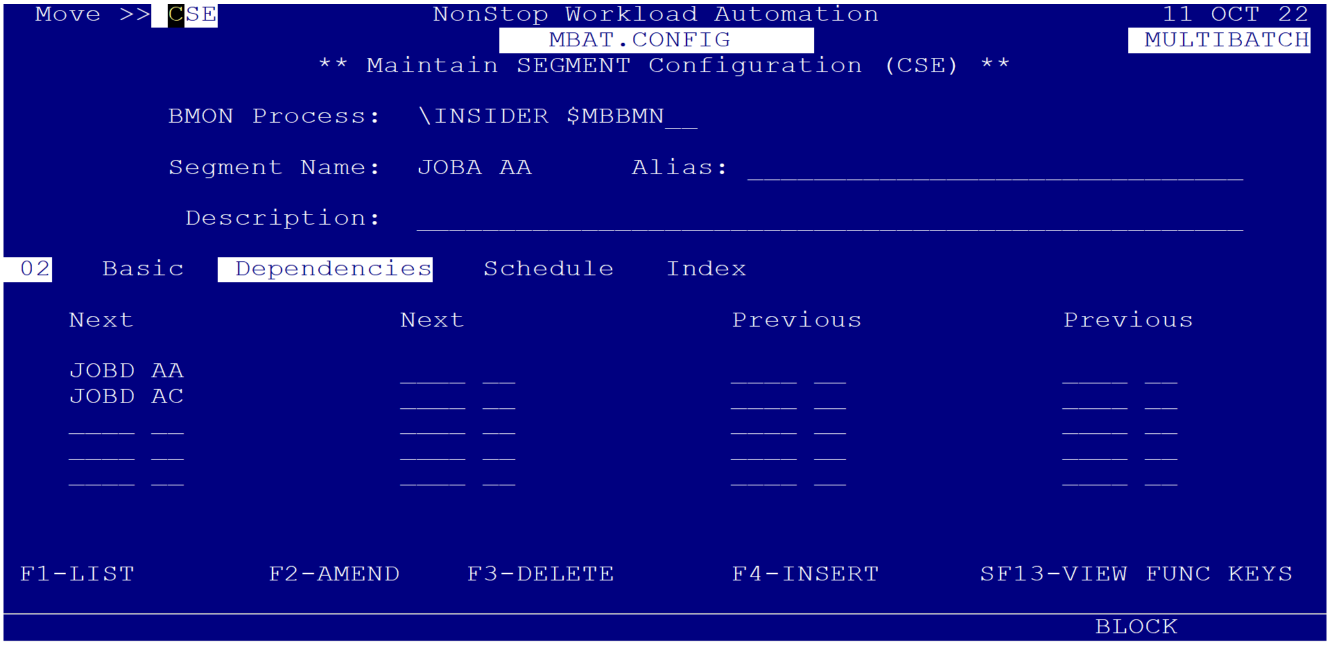
Navigate to the Entity Configuration Menu (CMM) screen and press the **F3** –Configure Segment function key.

This accesses the Maintain Segment Configuration Screen (CSE) (Figure 3.6). Like the Maintain Job Configuration Screen, this screen comprises four sub-screens or overlays.



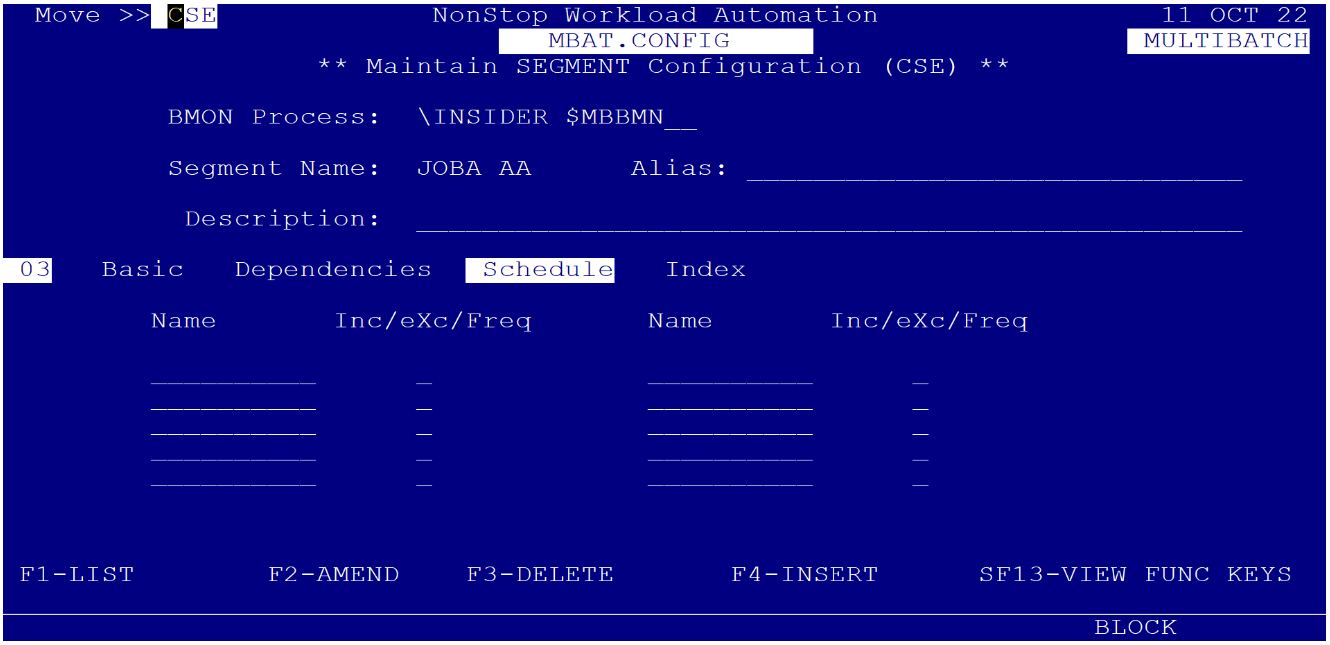
```
 Move >> CSE               NonStop Workload Automation              11 OCT 22
                                  MBAT.CONFIG                       MULTIBATCH
                   ** Maintain SEGMENT Configuration (CSE) **

         BMON Process:  \INSIDER $MBBMN__

         Segment Name:  JOBA AA      Alias: ______________________________

          Description:  __________________________________________________

01    Basic   Dependencies   Schedule   Index

      Number Of Units:    2

  Number Of Two Steps:    0

        Critical Path:  N

          Event Timer:  N


F1-LIST        F2-AMEND    F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS
                                                                  BLOCK
```

Figure 3.6 – Maintain Segment Configuration Screen (CSE) (overlay 1 or 4)



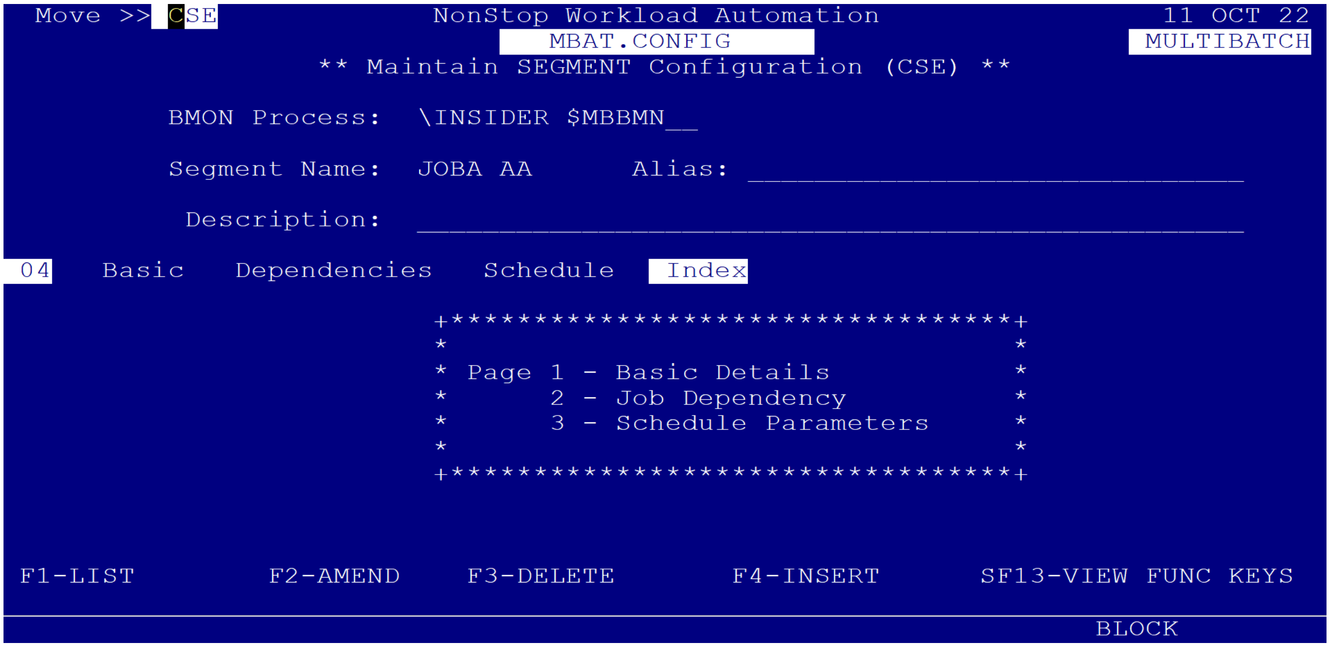
```
 Move >> CSE               NonStop Workload Automation              11 OCT 22
                                  MBAT.CONFIG                       MULTIBATCH
                   ** Maintain SEGMENT Configuration (CSE) **

         BMON Process:  \INSIDER $MBBMN__

         Segment Name:  JOBA AA      Alias: ______________________________

          Description:  __________________________________________________

02    Basic   Dependencies   Schedule   Index

    Next                Next                Previous            Previous

    JOBD AA             ____ __             ____ __             ____ __
    JOBD AC             ____ __             ____ __             ____ __
    ____ __             ____ __             ____ __             ____ __
    ____ __             ____ __             ____ __             ____ __
    ____ __             ____ __             ____ __             ____ __


F1-LIST        F2-AMEND    F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS
                                                                  BLOCK
```

Figure 3.6 – Maintain Segment Configuration Screen (CSE) (overlay 2 of 4)



```
  Move >> CSE                 NonStop Workload Automation              11 OCT 22
                                   MBAT.CONFIG                        MULTIBATCH
                       ** Maintain SEGMENT Configuration (CSE) **

          BMON Process:  \INSIDER $MBBMN__

          Segment Name:  JOBA AA      Alias: ______________________________

           Description:  __________________________________________________

 03    Basic   Dependencies   Schedule    Index

         Name        Inc/eXc/Freq        Name        Inc/eXc/Freq

         __________       _              __________       _
         __________       _              __________       _
         __________       _              __________       _
         __________       _              __________       _
         __________       _              __________       _


 F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                      BLOCK
```

Figure 3.6 – Maintain Segment Configuration Screen (CSE) (overlay 3 of 4)



```
  Move >> CSE                 NonStop Workload Automation              11 OCT 22
                                   MBAT.CONFIG                        MULTIBATCH
                       ** Maintain SEGMENT Configuration (CSE) **

          BMON Process:  \INSIDER $MBBMN__

          Segment Name:  JOBA AA      Alias: ______________________________

           Description:  __________________________________________________

 04    Basic   Dependencies   Schedule    Index

                                +***********************************+
                                *                                   *
                                * Page 1 - Basic Details            *
                                *      2 - Job Dependency           *
                                *      3 - Schedule Parameters      *
                                *                                   *
                                +***********************************+


 F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                      BLOCK
```

Figure 3.6 – Maintain Segment Configuration Screen (CSE) (overlay 4 of 4)

You can configure SEGMENTs in exactly the same way that JOBs were configured in the previous section. For this example, schedule, enter the JOB and SEGMENT id; see example in Figure 3.6, overlay 1 of 4. In a similar way, add a segment for the other jobs in the schedule.

Note: *It is not necessary to create Schedule parameters if the segment schedule is the same as the job. For example, if the jobs and segment both run daily, then the segment schedule can be left blank if the job runs daily. However, if the job runs daily but the segment runs on Mondays only, then "Monday/Include" is required. A segment never inherits job dependencies. Leaving it blank means that it is not connected to another segment but will run when the job runs.*

3.6 Configure STEP (CST)

Navigate to the Entity Configuration Menu (CMM) screen and press **F4** – Configure Guardian Step Menu or **F5** – Configure OSS Step Menu.

Each function key will take you to the CGM (Guardian Step Configuration Menu) or the COM (OSS Step Configuration Menu) screens respectively, from where after selecting the appropriate F1 function key, you will be taken to the CST (Maintain Step Configuration) or COS (Maintain OSS Step Configuration) screens.

Like the Maintain Job and Segment Configuration screens, these screens comprise a series of sub-screens or overlays.

Info: The Conditional Parameters (PME) facility allows a user to substitute an actual value such as $D01 with a parameter <Disk>. The "<" and ">" characters signify the delimiters of the parameter. The parameter will be expanded to its true value at the BUILD phase.

If you wish to know its value at the time of insert/amend, place the cursor anywhere within the field containing the parameter and press the **SF11** (EXPLAIN) function key. The value will be displayed on line 24 of your screen.

This facility is also available from the Maintain Define Class (CDE), Maintain Defines (CDF), Maintain Assign Class (CAS), Maintain Parameter Class (CPA), Maintain BMON (CBM), Maintain File Descriptors Class (COF) and Maintain Environment Class (COE) screens.

As with previous overlays the CST screen is divided into two distinct areas, with the upper half of the screen being common to each overlay; this provides details of the process name, STEP name and the Alias and Description (if applicable).

The following five overlays are common to both types of Step Screen:

Basic.
Schedule.
Conditional.
Defines.
Index.

The following overlays are only for Guardian Step Configuration:

Run Options.
Assign.
Params

The following overlays are common only for OSS Step configuration records:

An additional Basic Overlay (2).
Options Overlay.
EnvC (Environment Classes).
Env (Environment Params).
FDC (File Descriptor Classes).
FD (File Descriptor Classes).

Basic Overlay. The Basic overlay (See Figure 3.7) allows the user to specify the following fields:

The BMON Process, the Step Name, the Alias (if appropriate) and the Description (if appropriate). This can be done from any of the overlay screens and need only be input once. This will not be referred to when describing the other overlays.

Object File. For Guardian Steps only, this field is used to precisely identify the program to be executed.

Note: *The maximum fileset value accepted by MultiBatch is 34 characters.*

Default Vol/Sub-vol. This field specifies the System, Volume and Subvolume. This is used if in other screens/ overlays, the full information is not provided. This field is rarely used, because if it is changed by mistake, applications can pick up the wrong data with possible catastrophic results. This is not present on OSS Steps.

Process Name. For Guardian (and OSS) Steps this takes the form $nnnnn (where the first n is an alpha character and the following characters can be any alphanumeric characters) and is used to uniquely identify a running program.
However, if left blank the Operating System will be asked to allocate a name in the $Xnnn, $Ynnn, $Znnn range.

If the process is to execute on a remote system and it is necessary to be able to access the process, its name should consist of, at most, four characters and the "$"; this leaves a byte for the system to insert the node number.

Critical Path. This lets the user indicate whether or not the Step lies on a critical path. The permitted values of this field are Y and N.

Protocol. An interface between the BMON process and the current application which allows the process to talk to the BMON process (e.g. to report on progress). This interface involves adding standard routines to the Application being used and recompiling the software. For this reason, it is seldom used. The permitted values of this field are Y and N. Default is N. See Appendix J BMON Protocol for further details.

User ID. This field allows the user to specify whether the Unit should execute with the supplied Guardian User Id or with that of the BMON process Owner. The user must enter:

Y or N to indicate whether or not the feature is to be used.

A Group.User Id is in the form nnn,nnn where both three digit numbers can range from 000 to 255.

Password. A user must supply the password, when the users associated Security Class demands it and if a user is inserting, amending or deleting a step record. The associated security setting is CST02 and the password must be supplied if this value is "P".

Note: *The CST02 Facility must be set to Y or P on the SCL screen to access this feature.*

Maximum Time. This field is optional and allows the user to specify a maximum time for the job (start to end). If used, then a minimum of 2 minutes must be specified. Is it used in conjunction with the Warn / Abort parameter.

If **Warn/Abort** is set to W, then a warning message will be issued if the job has exceeded the Maximum Time value but the job will continue processing.

If **Warn/Abort** is set to A, and the job has not completed before its Maximum Time value, then the job will be aborted.

If the Warn/Abort value is set to A and it the job has not completed five minutes before the deadline, a message indicating that it will be aborted in five minutes is generated. The Job may finish before the deadline or the user can override the maximum time by entering the following BCOM command:

```
ALTER STEP <step-name> NOMAXTIME
```

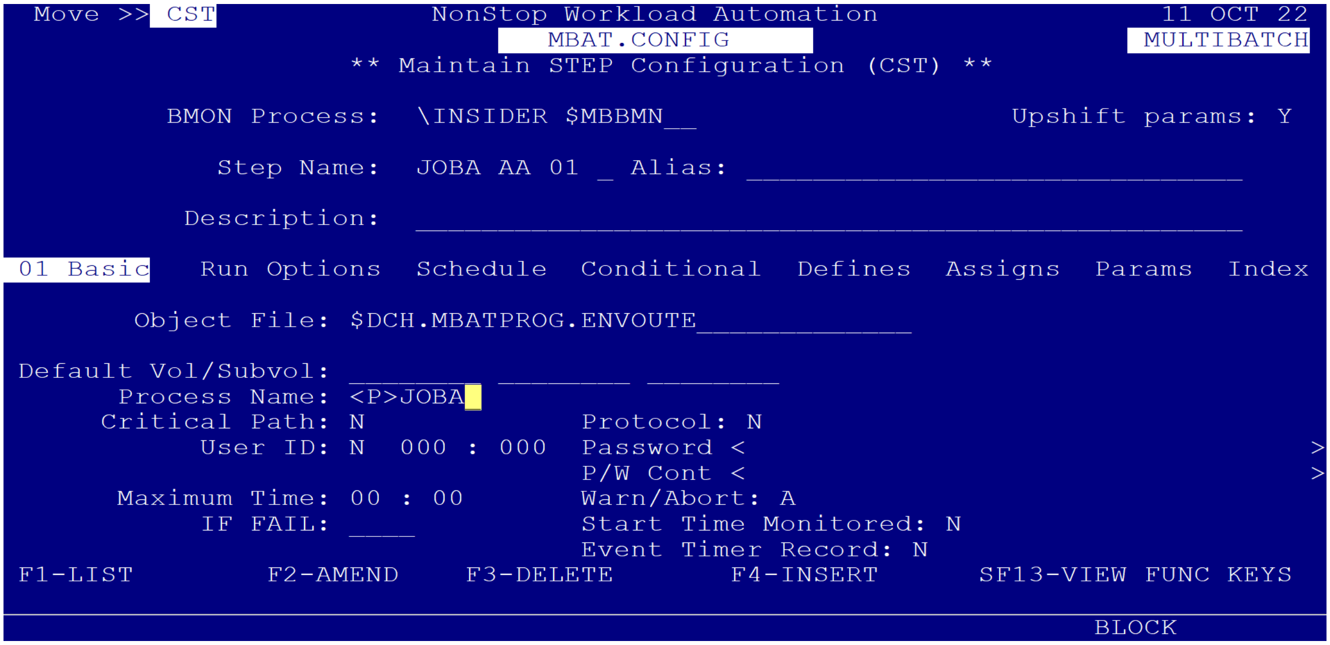
If the Job has not been stopped before the deadline, then it is aborted.

If Fail. If the unit fails for some reason, this field allows the user to specify another Job to be executed.

Start Time Monitored. Is read-only and if set to Y, then start time monitoring data has been created within the CMS (Configure Monitor Step) screen.

Event Timer Record. Is read-only and if set to Y, then the unit has been configured within the Event Timer (EME) menu.

Upshift Params. If set to Y, then all parameters in the 'Params' overlay and Run Params in the 'Run Options' overlay configured within this Step will be upshifted. If set to N, then the values entered by the user will be retained. The default is Y.

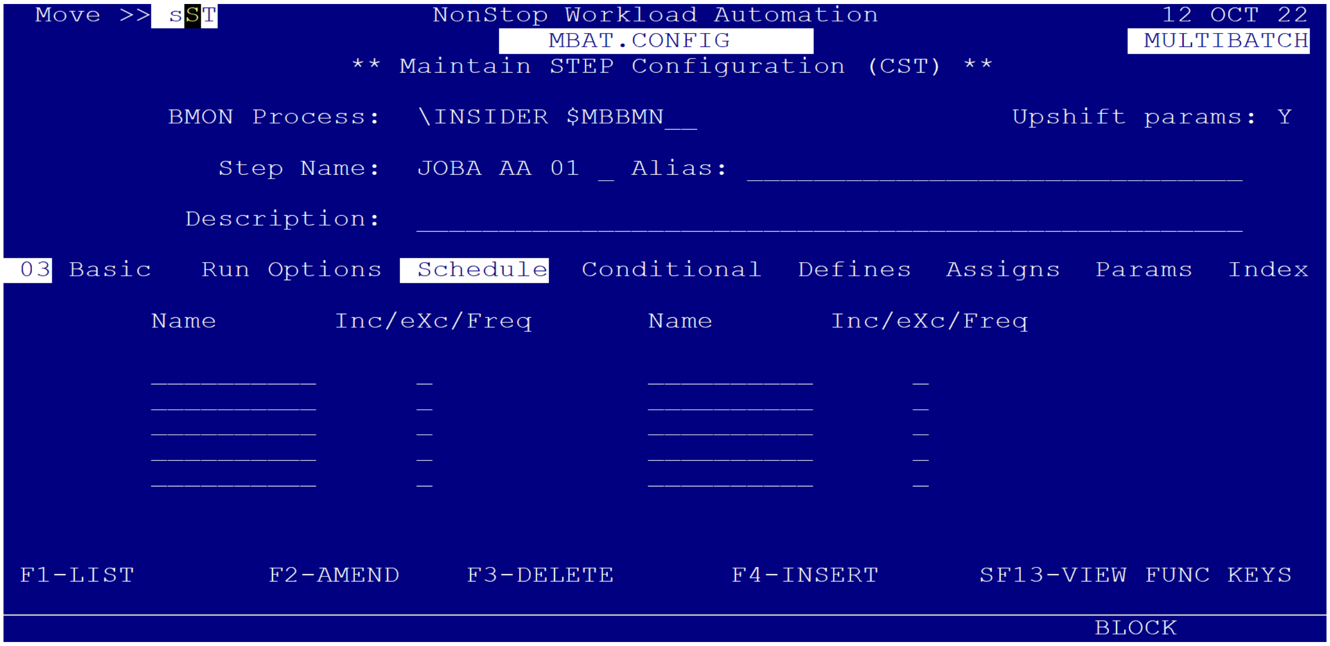


```
  Move >> CST            NonStop Workload Automation                 11 OCT 22
                               MBAT.CONFIG                          MULTIBATCH
                   ** Maintain STEP Configuration (CST) **

        BMON Process:  \INSIDER $MBBMN__                 Upshift params: Y

           Step Name:  JOBA AA 01 _ Alias: ______________________________

         Description:  __________________________________________________

 01 Basic   Run Options  Schedule  Conditional  Defines  Assigns  Params  Index

       Object File: $DCH.MBATPROG.ENVOUTE_____________

 Default Vol/Subvol: ________ ________ ________
       Process Name: <P>JOBA
      Critical Path: N             Protocol: N
            User ID: N  000 : 000  Password <                                >
                                   P/W Cont <                                >
       Maximum Time: 00 : 00       Warn/Abort: A
           IF FAIL: ____           Start Time Monitored: N
                                   Event Timer Record: N
 F1-LIST        F2-AMEND    F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS

                                                                      BLOCK
```

Figure 3.7 – Maintain Step Configuration (CST) - Basic

3.7 CST Schedule Overlay

The Schedule overlay (Figure 3.8) is as described for Jobs and Segments and allows the user to specify when the Unit is to be executed (e.g. TUESDAY I). This is the same for OSS Step Configuration.



```
  Move >> sST            NonStop Workload Automation                 12 OCT 22
                               MBAT.CONFIG                          MULTIBATCH
                   ** Maintain STEP Configuration (CST) **

        BMON Process:  \INSIDER $MBBMN__                 Upshift params: Y

           Step Name:  JOBA AA 01 _ Alias: ______________________________

         Description:  __________________________________________________

 03 Basic   Run Options  Schedule  Conditional  Defines  Assigns  Params  Index

          Name        Inc/eXc/Freq        Name        Inc/eXc/Freq

          __________       _              __________       _
          __________       _              __________       _
          __________       _              __________       _
          __________       _              __________       _
          __________       _              __________       _


 F1-LIST        F2-AMEND    F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS

                                                                      BLOCK
```

Figure 3.8 – Maintain STEP Configuration Screen (CST) - Schedule

3.8 CST Conditional Overlay

This overlay provides an additional mechanism schedule or skip a job depending on whether a Conditional Parameter is "TRUE" or "FALSE".

For example, the parameter database (PME) contains two parameters:

WEEKDAY = TRUE
WEEKEND = FALSE

Within the Maintain STEP Configuration (CST) screen - Conditional overlay - the following values will cause the unit to Run or be Skipped:

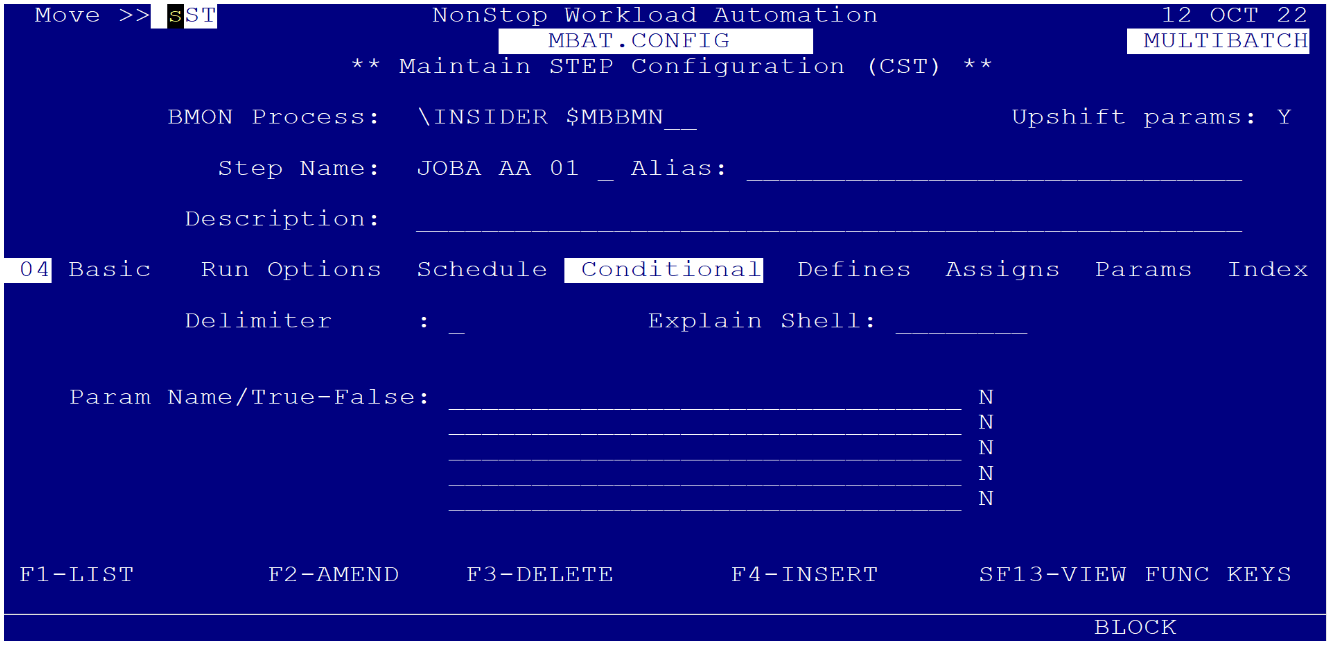
CST Conditional Parameter	**True/False**	**Flag 'Run / Skip'**
WEEKDAY	Y	Run
WEEKDAY	N	Skip
WEEKEND	Y	Skip
WEEKEND	N	Run

As many as five Param Name/True-False values can be supplied. If any one of these evaluates to skip, then the step will be skipped.

Note that when using this mechanism the step always appears in the schedule but can be set as skipped. This is different to the scheduling on the Schedule Overlay where if a step is not selected it will not appear in the schedule.

Note: See Appendix D.6 on the need for caution when using SKIP param within jobs which use the GROUPID function.

This overlay also provides the possibility of changing the conditional parameter delimiter for a step. It is unlikely that this will be needed; it is included here to cover the situation where the standard delimiters are needed within a configuration value. Be aware that this affects all conditional processing for the step, see 1.19 Conditional Parameters.



```
  Move >> SST                 NonStop Workload Automation                 12 OCT 22
                                   MBAT.CONFIG                         MULTIBATCH
                             ** Maintain STEP Configuration (CST) **

          BMON Process:  \INSIDER $MBBMN__                      Upshift params: Y

             Step Name:  JOBA AA 01 _ Alias: ______________________________

           Description:  __________________________________________________

 04 Basic    Run Options  Schedule  Conditional  Defines  Assigns  Params  Index

           Delimiter      : _            Explain Shell: ________


    Param Name/True-False: _______________________________ N
                           _______________________________ N
                           _______________________________ N
                           _______________________________ N
                           _______________________________ N


 F1-LIST        F2-AMEND     F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                    BLOCK
```

Figure 3.9 – Maintain STEP Configuration Screen (CST) - Conditional

3.9 CST Run Options, Defines, Assigns and Params

These four overlays on the Guardian Step Configuration Screen provide an interface to standard NonStop facilities as follows:

Run Options. This overlay (Figure 3.10) allows the user to set the various NonStop run options. A user can specify hard coded CPU values for a step, such as 01 and 02. BMON will attempt to start the step in CPU 1 and if this is not available, then CPU 2 is selected. Alternatively, a user can create local or global parameters to represent the CPU specification in the conditional parameter database through the PME facility. This is discussed in Appendix D. If these values exist, then value substitution will take place during the BUILD phase.

Note: *The CPU 1: and CPU 2: fields are utilised for conditional parameter values. However, these fields do not require the conditional parameter delimiters of "<" and ">". Therefore, specify the conditional parameter name without any delimiters, e.g. CPUX or CPUY.*

Defines. This overlay (Figure 3.11) allows the user to specify defines that will be passed to the step process, these are specified on the Configure Defines (CDF) and Configure Define Class (CDE) screen.

Assigns. This overlay (Figure 3.12) allows the user to specify a process's assigns either completely or by reference to an Assign Class, as specified on the Assign Class Definition (CAS) screen. A maximum of 60 Assigns can be catered for.

Note: *The maximum value for the LOGICAL field in MultiBatch is 31 characters.*

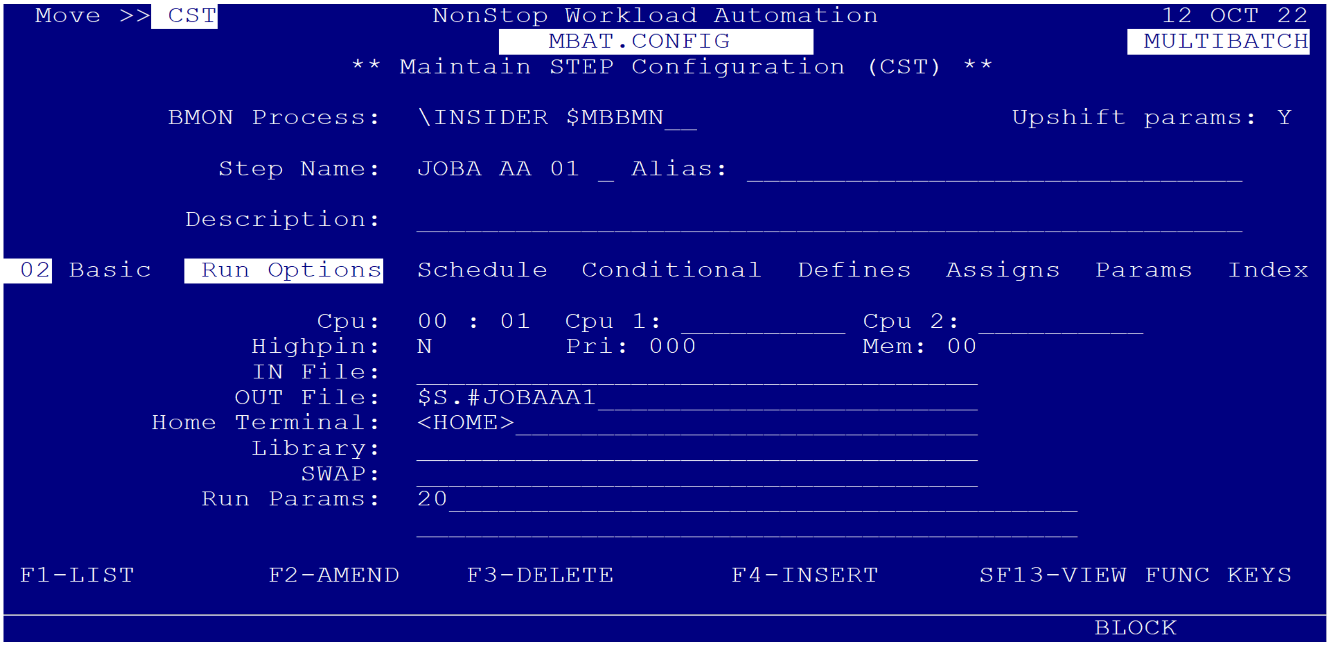
Params. This overlay (Figure 3.13) allows the user to specify a process's parameters either completely or by reference to a Param Class, as specified on the Param Class Definition (CPA) screen. The parameter value may be a string up to 80 characters long.

The length field is updated automatically if left set to 0. However, if you wish a parameter to have 3 trailing spaces, e.g. "TUESDAY   " set the length of the value, (in this example, seven characters and three spaces) to 10.

A maximum of 100 Params can be catered for.

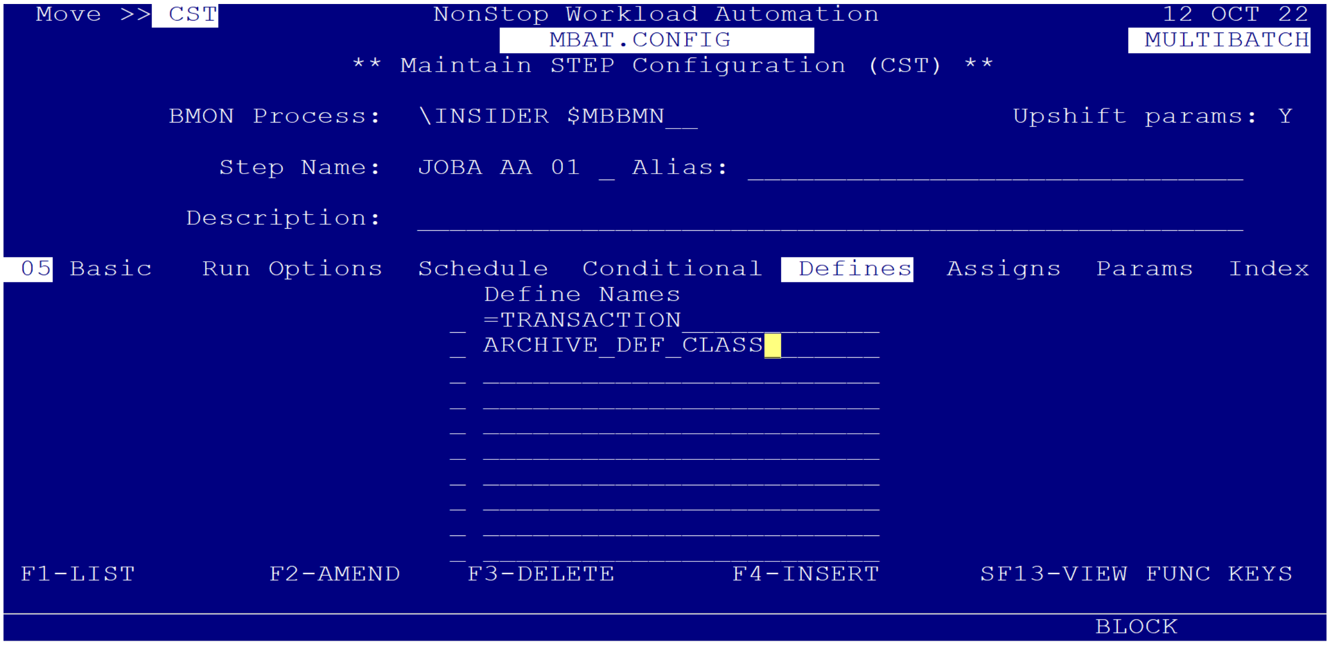
Note: *Combination of Name, Value & Index Len Values must not exceed 1024 characters otherwise a warning "Parameters Exceed 1024 Bytes in Length" will be issued during the BMON Build process.*

Configuring of Assign/Param/Define Classes and Defines is described later in this chapter.



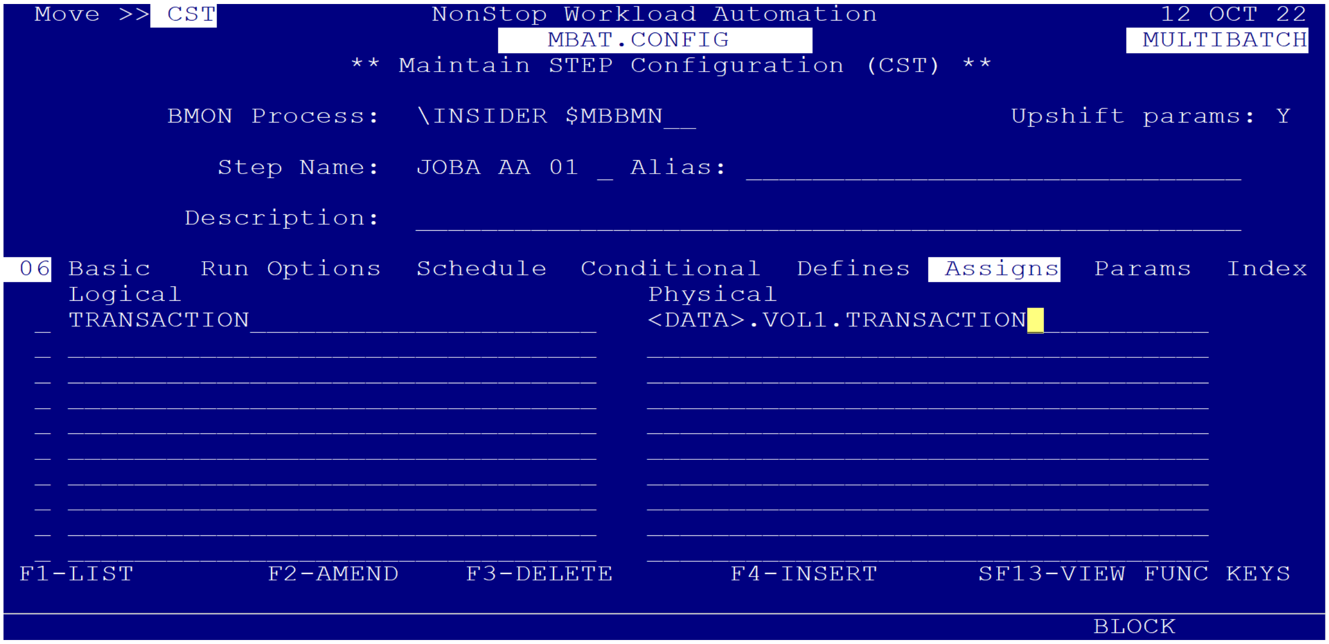
```
 Move >> CST             NonStop Workload Automation              12 OCT 22
                                 MBAT.CONFIG                      MULTIBATCH
                      ** Maintain STEP Configuration (CST) **

        BMON Process:  \INSIDER $MBBMN__                   Upshift params: Y

           Step Name:  JOBA AA 01 _ Alias: ______________________________

         Description:  __________________________________________________

02 Basic    Run Options  Schedule  Conditional  Defines  Assigns  Params  Index

                Cpu:  00 : 01  Cpu 1: __________  Cpu 2: __________
            Highpin:  N        Pri: 000           Mem: 00
            IN File:  __________________________________
           OUT File:  $S.#JOBAAA1________________________
      Home Terminal:  <HOME>____________________________
            Library:  __________________________________
               SWAP:  __________________________________
         Run Params:  20______________________________________
                      ________________________________________

 F1-LIST        F2-AMEND     F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS

                                                                 BLOCK
```

Figure 3.10 – Maintain Step Configuration Run (CST) – Run Options



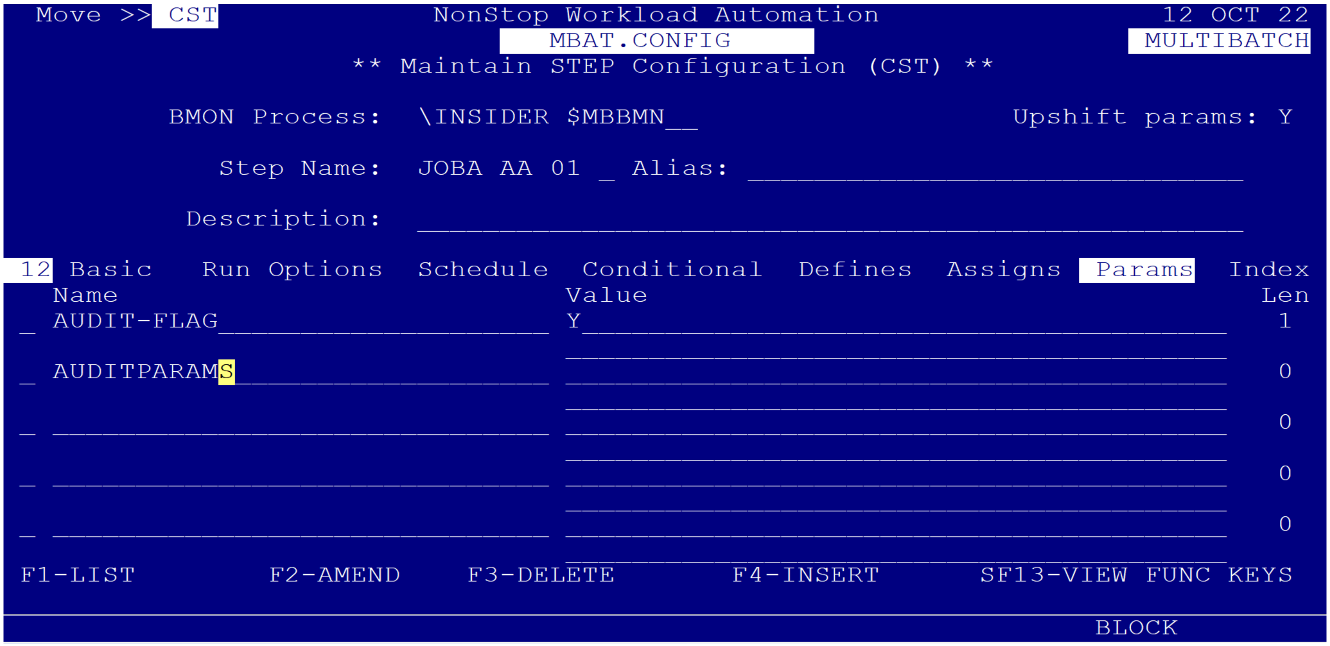
```
 Move >> CST                NonStop Workload Automation                12 OCT 22
                                 MBAT.CONFIG                          MULTIBATCH
                        ** Maintain STEP Configuration (CST) **

          BMON Process:  \INSIDER $MBBMN__                     Upshift params: Y

             Step Name:  JOBA AA 01 _ Alias: ______________________________

           Description:  __________________________________________________

 05 Basic   Run Options  Schedule  Conditional  Defines  Assigns  Params  Index
                             Define Names
                           _ =TRANSACTION____________
                           _ ARCHIVE_DEF_CLASS______
                           _ ________________________
                           _ ________________________
                           _ ________________________
                           _ ________________________
                           _ ________________________
                           _ ________________________
                           _ ________________________
                           _ ________________________
 F1-LIST         F2-AMEND     F3-DELETE        F4-INSERT       SF13-VIEW FUNC KEYS

                                                                      BLOCK
```

Figure 3.11 – Maintain Step Configuration (CST) – Defines



```
 Move >> CST                NonStop Workload Automation                12 OCT 22
                                 MBAT.CONFIG                          MULTIBATCH
                        ** Maintain STEP Configuration (CST) **

          BMON Process:  \INSIDER $MBBMN__                     Upshift params: Y

             Step Name:  JOBA AA 01 _ Alias: ______________________________

           Description:  __________________________________________________

 06 Basic   Run Options  Schedule  Conditional  Defines  Assigns  Params  Index
    Logical                            Physical
 _  TRANSACTION________________________  <DATA>.VOL1.TRANSACTION__________
 _  ___________________________________  __________________________________
 _  ___________________________________  __________________________________
 _  ___________________________________  __________________________________
 _  ___________________________________  __________________________________
 _  ___________________________________  __________________________________
 _  ___________________________________  __________________________________
 _  ___________________________________  __________________________________
 _  ___________________________________  __________________________________
 F1-LIST         F2-AMEND     F3-DELETE        F4-INSERT       SF13-VIEW FUNC KEYS

                                                                      BLOCK
```

Figure 3.12 – Maintain Step Configuration (CST) - Assigns



```
  Move >> CST               NonStop Workload Automation                12 OCT 22
                                  MBAT.CONFIG                         MULTIBATCH
                         ** Maintain STEP Configuration (CST) **

          BMON Process:  \INSIDER $MBBMN__                     Upshift params: Y

             Step Name:  JOBA AA 01 _ Alias: ______________________________

           Description:  __________________________________________________

12 Basic   Run Options  Schedule  Conditional  Defines  Assigns   Params   Index
   Name                          Value                                      Len
_  AUDIT-FLAG____________________ Y_______________________________________   1
                                 ________________________________________
_  AUDITPARAMS___________________ ________________________________________   0
                                 ________________________________________
_  ______________________________ ________________________________________   0
                                 ________________________________________
_  ______________________________ ________________________________________   0
                                 ________________________________________
_  ______________________________ ________________________________________   0
                                 ________________________________________
 F1-LIST        F2-AMEND     F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                     BLOCK
```

Figure 3.13 – Maintain Step Configuration (CST) - Params

3.10 OSS Options, Env, EnvC, FDC, FD (COS)

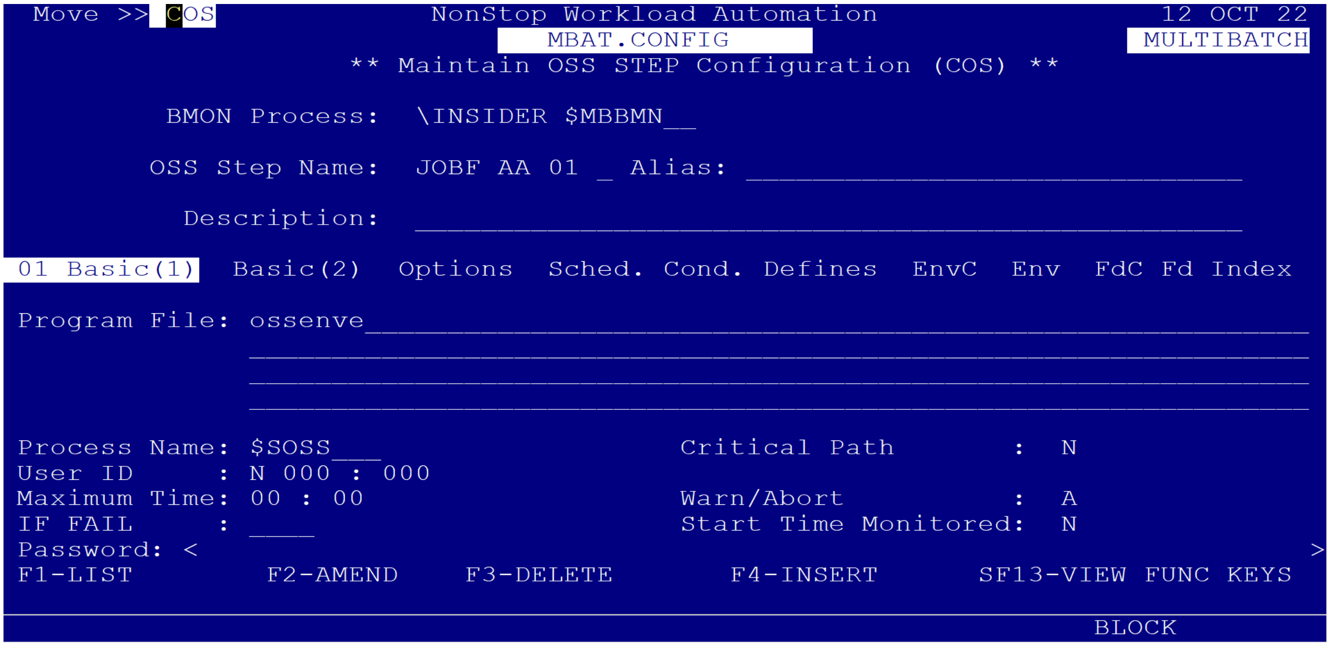
These overlays on the Maintain OSS Step Configuration Screen (COS) enables a user to configure OSS jobs to be run through MultiBatch.

Where specific OSS fields / overlays are documented, then these are described; those not described, utilise those overlays as described in the CST overlays.

Basic (1) (Figure 3.14)

Program File: The name of the application program (incl path name if required) to be executed. As with the OSS environment, this field is case sensitive. MultiBatch Conditional Parameters may be used here.

Process Name: This is the process name to be given to this OSS process. It must be preceded by a dollar sign ("$") and consists of a maximum of five alphanumeric characters. If the process is to execute on a remote system and it is necessary to be able to access the process, its name should consist of, at most, four characters and the "$"; this leaves a byte for the system to insert the node number.



```
 Move >> COS             NonStop Workload Automation               12 OCT 22
                              MBAT.CONFIG                        MULTIBATCH
                  ** Maintain OSS STEP Configuration (COS) **

         BMON Process:  \INSIDER $MBBMN__

        OSS Step Name:  JOBF AA 01 _ Alias: ______________________________

          Description:  __________________________________________________

01 Basic(1)  Basic(2)  Options  Sched. Cond. Defines  EnvC  Env  FdC Fd Index

Program File: ossenve_________________________________________________________
              ________________________________________________________________
              ________________________________________________________________
              ________________________________________________________________

Process Name: $SOSS___                      Critical Path       :  N
User ID     : N 000 : 000
Maximum Time: 00 : 00                       Warn/Abort          :  A
IF FAIL     : _____                         Start Time Monitored:  N
Password: <                                                                   >
F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                         BLOCK
```

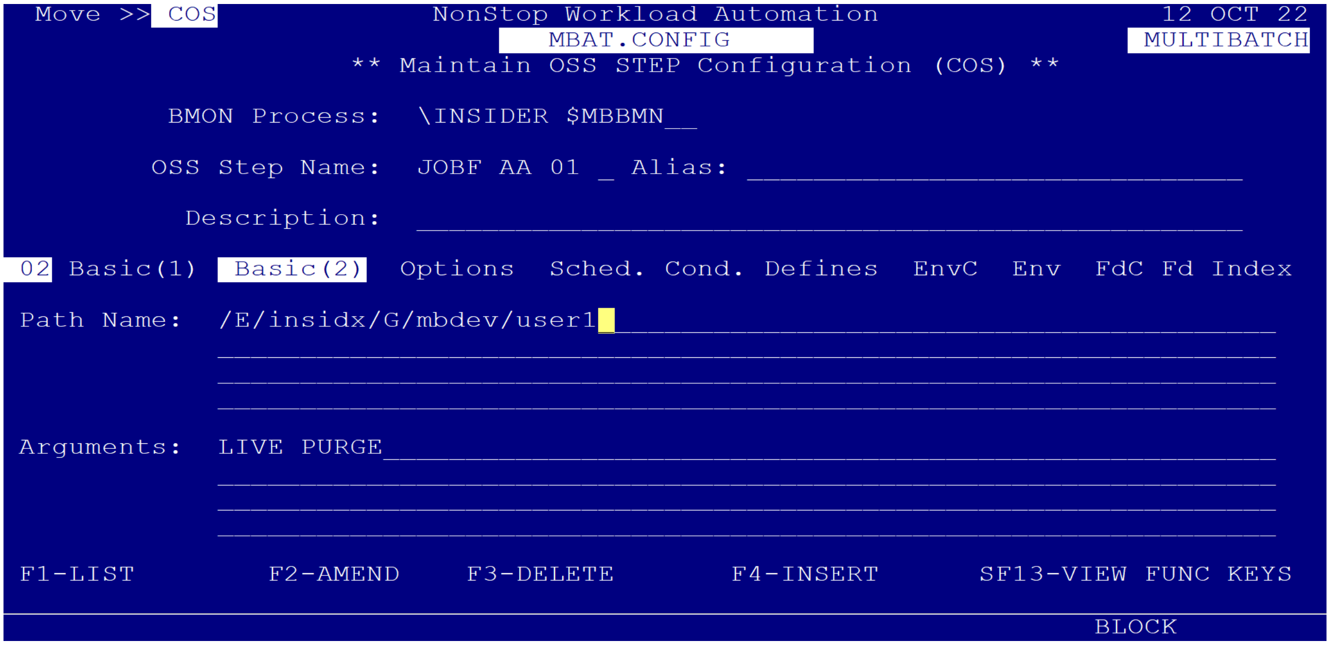
Figure 3.14 – Maintain OSS Step Configuration (COS) - Basic (1)

These overlays on the Maintain OSS Step Configuration Screen (COS) enables a user to configure OSS jobs to be run through MultiBatch.

Basic (2) (Figure 3.15)

Path Name: The path name can be used to fully qualify the OSS file name, or as described above, the Program File can specify the path name as part of the program file path. As with the OSS environment, this field is case sensitive. MultiBatch Conditional Parameters may be used here.

Arguments: This value is the equivalent of the Guardian startup string. It enables run time variable information to be passed to the newly created process. As with the OSS environment, this field is case sensitive. MultiBatch Conditional Parameters may be used here.



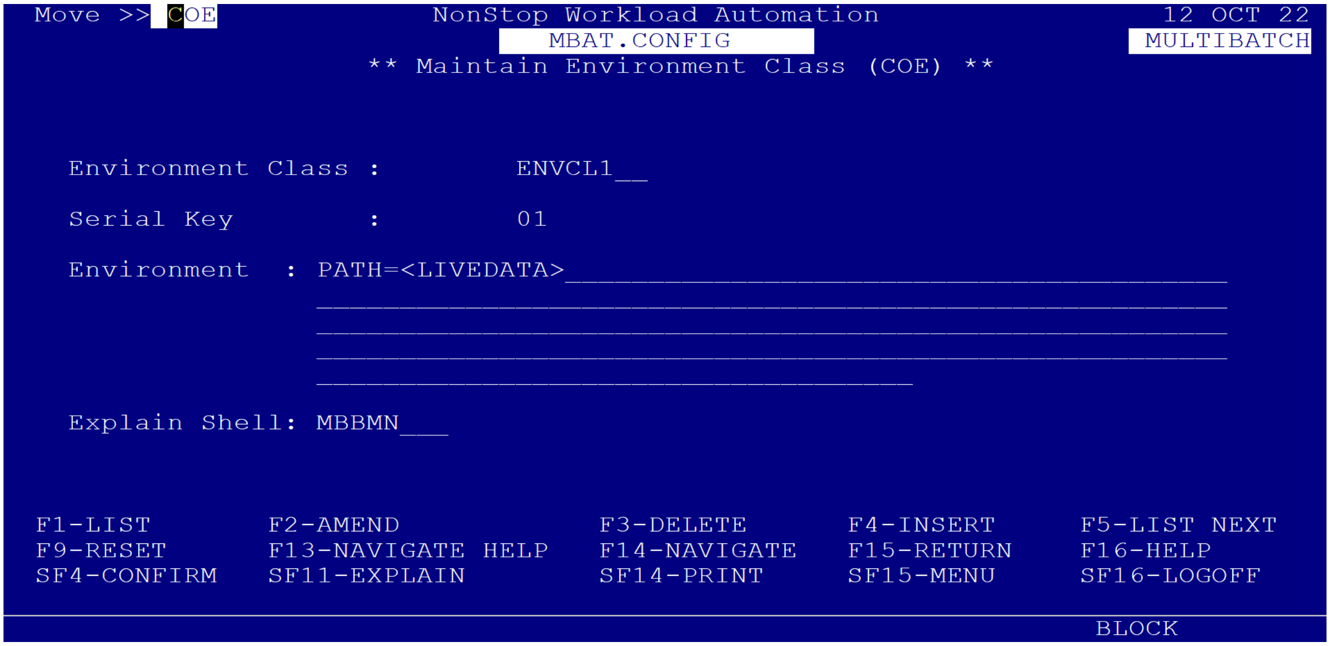
```
  Move >> COS            NonStop Workload Automation               12 OCT 22
                               MBAT.CONFIG                        MULTIBATCH
                   ** Maintain OSS STEP Configuration (COS) **

          BMON Process:  \INSIDER $MBBMN__

         OSS Step Name:  JOBF AA 01 _ Alias: ______________________________

           Description:  __________________________________________________

 02 Basic(1)  Basic(2)  Options  Sched. Cond. Defines  EnvC  Env  FdC Fd Index

 Path Name:  /E/insidx/G/mbdev/user1________________________________________
             _______________________________________________________________
             _______________________________________________________________
             _______________________________________________________________

 Arguments:  LIVE PURGE_____________________________________________________
             _______________________________________________________________
             _______________________________________________________________
             _______________________________________________________________

 F1-LIST          F2-AMEND     F3-DELETE        F4-INSERT       SF13-VIEW FUNC KEYS

                                                                  BLOCK
```

Figure 3.15 – Maintain OSS Step Configuration (COS) - Basic (2)

EnvC / Env: (Figures 3.16, 3.17 and 3.18)

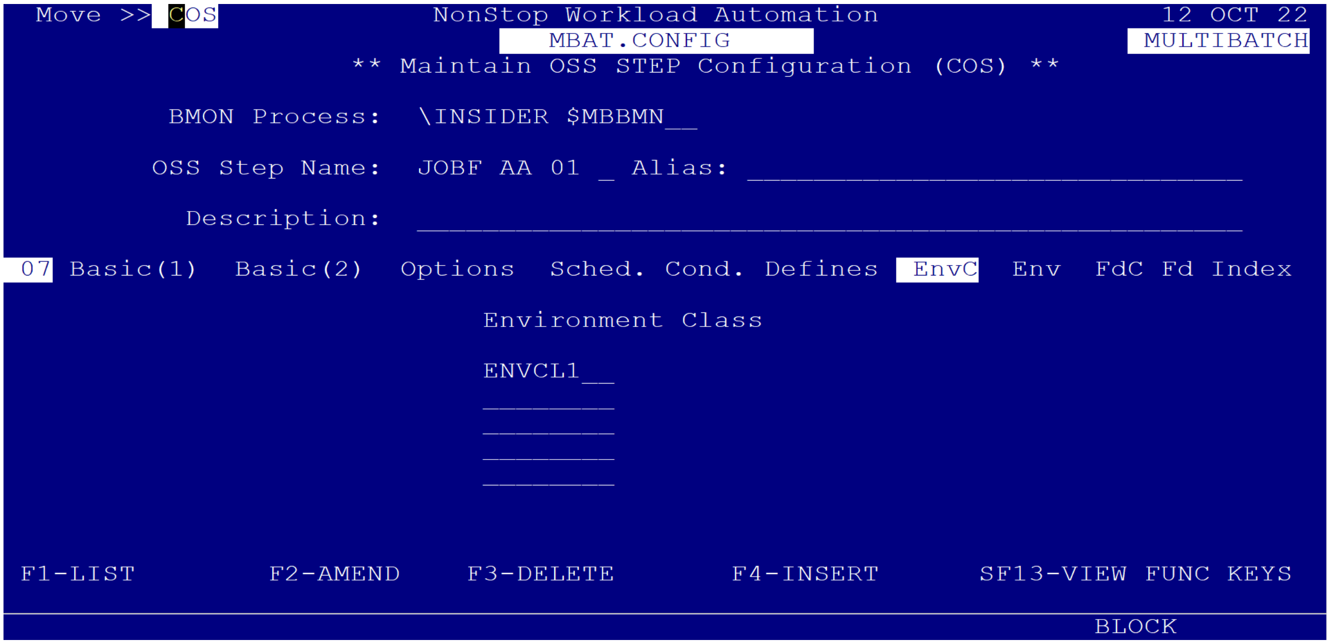
Via the COE (Maintain Environment Class), a user can configure an Environment Class. This Class name is then specified in the EnvC overlay and will be substituted with the underlying Environment Class values when the BMON is built. It can be beneficial to define an Environment Class for an environment name that may be referenced in multiple COS configurations, as this provides a central location for all changes.

If an Environment name is referenced by more than one COS entry, then all COS records need to be changed; by utilising an Environment Class, only a single change needs to be made in the COE screen.



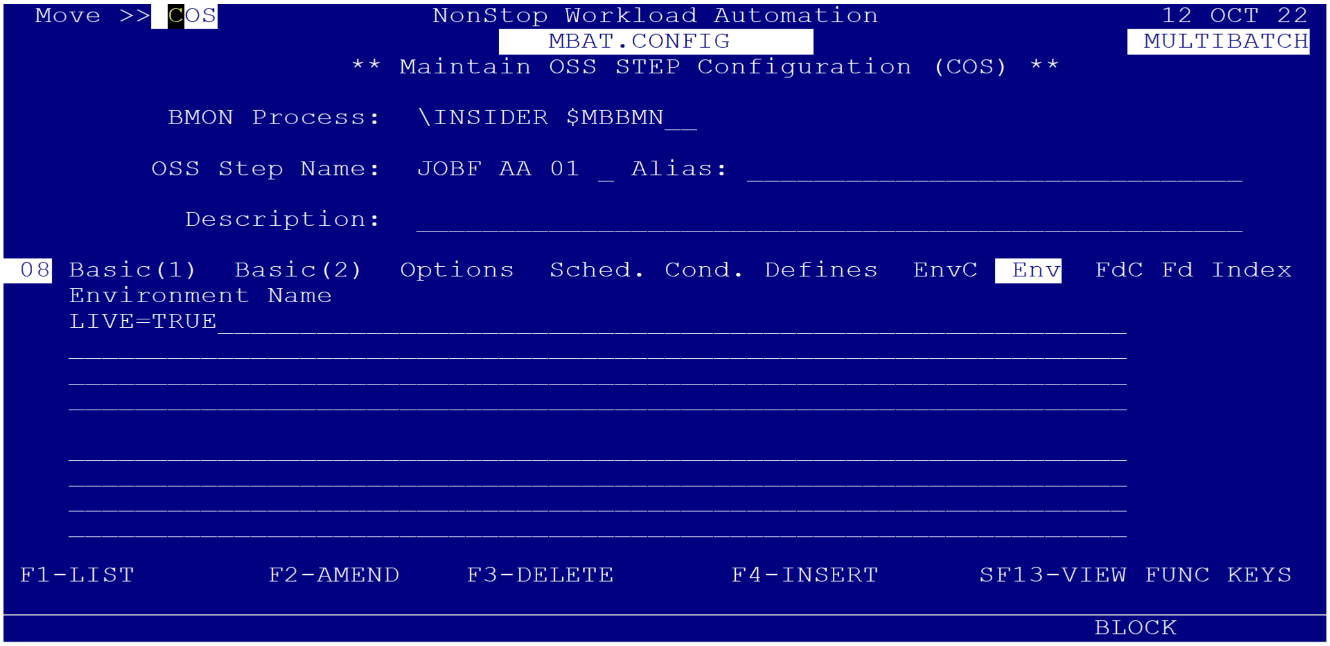
```
  Move >> COE            NonStop Workload Automation               12 OCT 22
                               MBAT.CONFIG                        MULTIBATCH
                    ** Maintain Environment Class (COE) **


    Environment Class :          ENVCL1__

    Serial Key        :          01

    Environment   : PATH=<LIVEDATA>________________________________________
                    _______________________________________________________
                    _______________________________________________________
                    _______________________________________________________
                    ____________________________________

    Explain Shell: MBBMN___


  F1-LIST       F2-AMEND            F3-DELETE     F4-INSERT     F5-LIST NEXT
  F9-RESET      F13-NAVIGATE HELP   F14-NAVIGATE  F15-RETURN    F16-HELP
  SF4-CONFIRM   SF11-EXPLAIN        SF14-PRINT    SF15-MENU     SF16-LOGOFF

                                                                  BLOCK
```

Figure 3.16 – Maintain Environment Class (COE)



```
  Move >> COS               NonStop Workload Automation                12 OCT 22
                               MBAT.CONFIG                            MULTIBATCH
                    ** Maintain OSS STEP Configuration (COS) **

         BMON Process:  \INSIDER $MBBMN__

        OSS Step Name:  JOBF AA 01 _ Alias: ______________________________

          Description:  __________________________________________________

07 Basic(1)  Basic(2)  Options  Sched. Cond. Defines  EnvC  Env  FdC Fd Index

                                    Environment Class

                                    ENVCL1__
                                    ________
                                    ________
                                    ________
                                    ________


 F1-LIST         F2-AMEND     F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                         BLOCK
```

Figure 3.17 – Maintain OSS Step Configuration (COS) – Environment Classes



```
  Move >> COS               NonStop Workload Automation                12 OCT 22
                               MBAT.CONFIG                            MULTIBATCH
                    ** Maintain OSS STEP Configuration (COS) **

         BMON Process:  \INSIDER $MBBMN__

        OSS Step Name:  JOBF AA 01 _ Alias: ______________________________

          Description:  __________________________________________________

08 Basic(1)  Basic(2)  Options  Sched. Cond. Defines  EnvC  Env  FdC Fd Index
   Environment Name
   LIVE=TRUE_______________________________________________________
   ________________________________________________________________
   ________________________________________________________________
   ________________________________________________________________

   ________________________________________________________________
   ________________________________________________________________
   ________________________________________________________________
   ________________________________________________________________

 F1-LIST         F2-AMEND     F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                         BLOCK
```

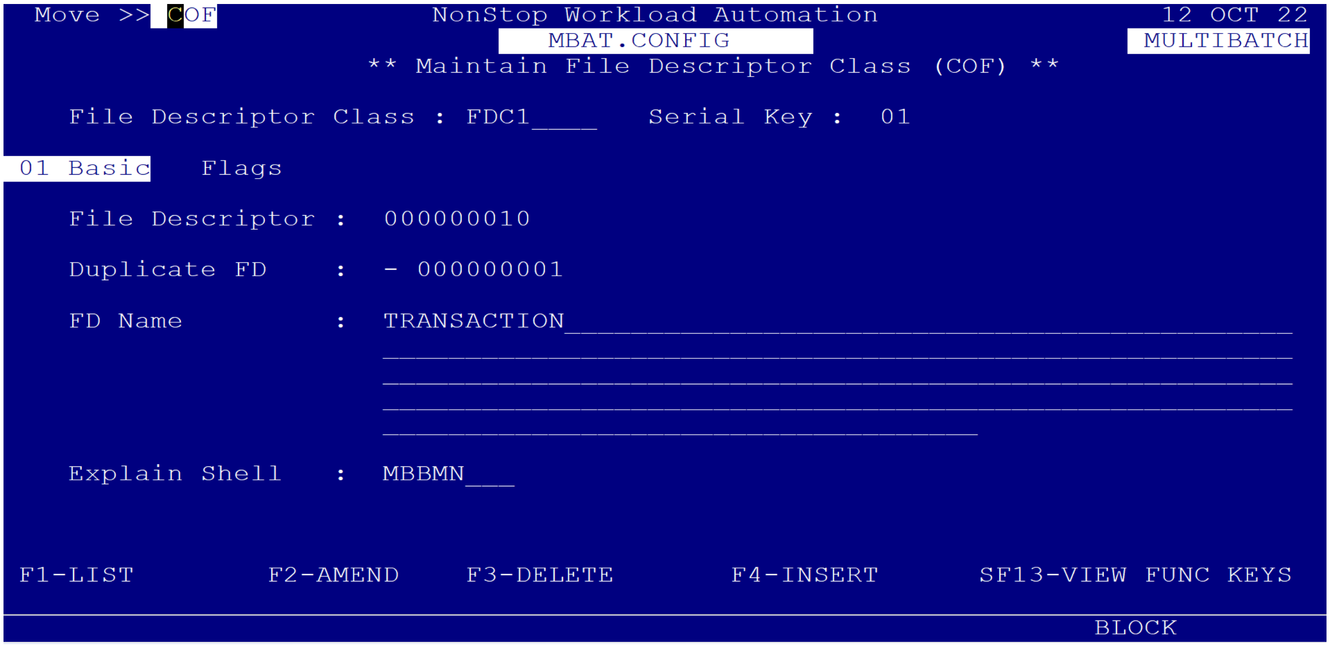
Figure 3.18 – Maintain OSS Step Configuration (COS) – Environment

COF (Figure 3.19)

Via the Maintain File Descriptor Class(COF), a user can configure an FD (File Descriptor) Class. This FD Class name, e.g. FDC1 in the image below, is then specified in the **COS FDC (Figure 3.20)** overlay and will be substituted with the underlying FD values when the BMON is built. It can be beneficial to define a File Descriptor Class for an FD that may be referenced in multiple COS configurations, as this provides a central location for all changes.

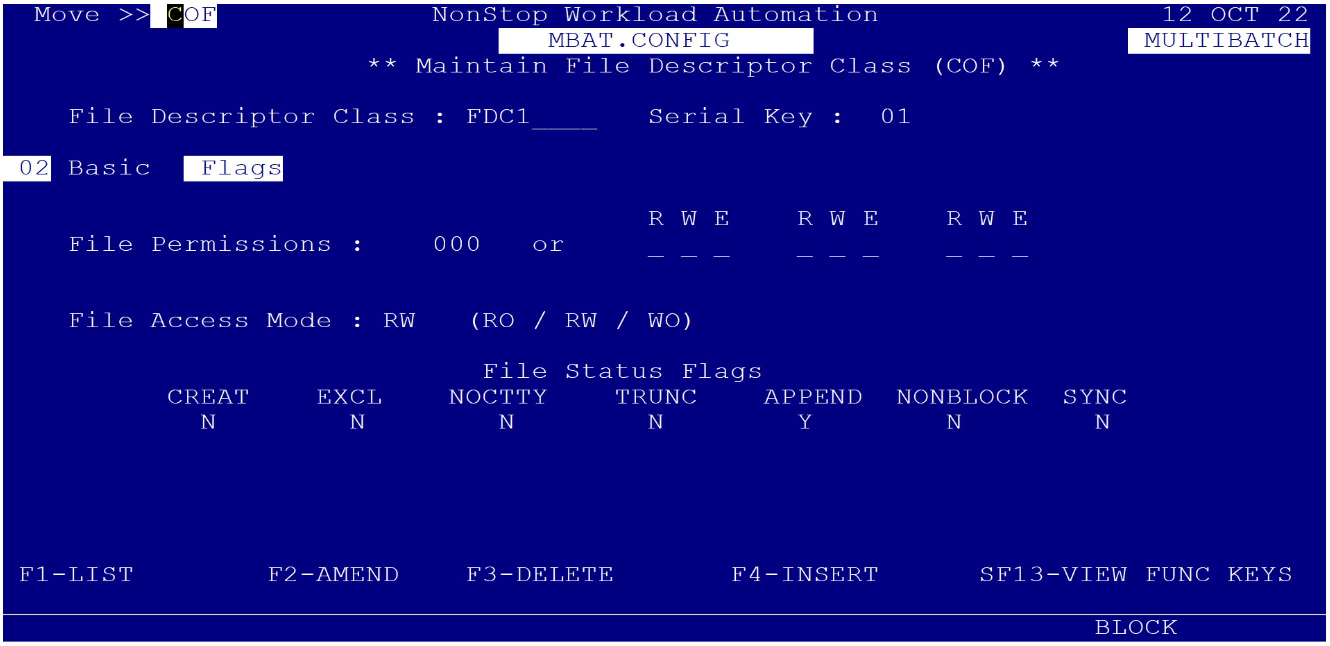
If an FD name is referenced by more than one COS entry, then all COS records need to be changed; by utilising a File Descriptor Class, only a single change needs to be made in the COF screen.

Placing the cursor on a COS FDC overlay class name and pressing F8 displays the associated File Descriptor Class (COF) screen.



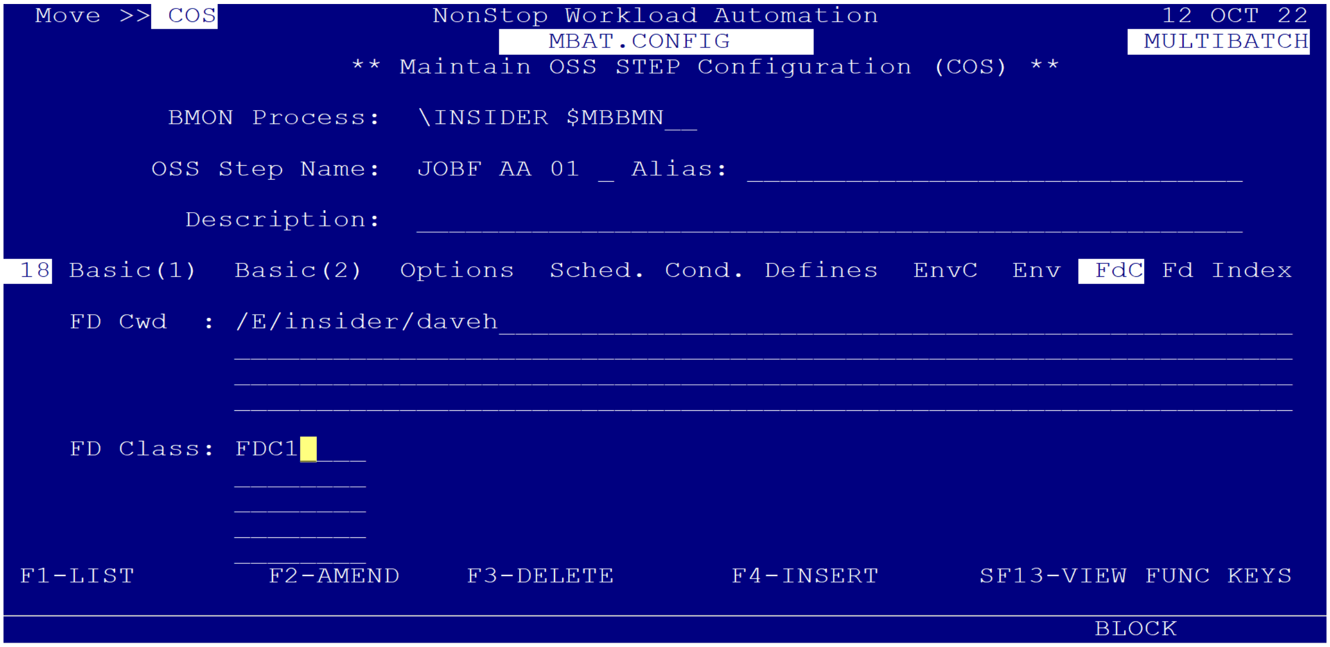
```
 Move >> COF            NonStop Workload Automation              12 OCT 22
                               MBAT.CONFIG                      MULTIBATCH
                  ** Maintain File Descriptor Class (COF) **

   File Descriptor Class : FDC1____    Serial Key :  01

01 Basic   Flags

   File Descriptor  :  000000010

   Duplicate FD     :  - 000000001

   FD Name          :  TRANSACTION___________________________________________
                       _______________________________________________________
                       _______________________________________________________
                       _______________________________________________________
                       ____________________________________

   Explain Shell    :  MBBMN___


F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                BLOCK
```

Figure 3.19 – Maintain File Descriptor (COF) - Basic



```
 Move >> COF            NonStop Workload Automation              12 OCT 22
                               MBAT.CONFIG                      MULTIBATCH
                  ** Maintain File Descriptor Class (COF) **

   File Descriptor Class : FDC1____    Serial Key :  01

02 Basic  Flags

                                          R W E    R W E    R W E
   File Permissions :   000   or          _ _ _    _ _ _    _ _ _


   File Access Mode : RW   (RO / RW / WO)

                             File Status Flags
        CREAT     EXCL    NOCTTY    TRUNC    APPEND  NONBLOCK  SYNC
          N         N       N         N        Y        N        N




F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                BLOCK
```

Figure 3.19 – Maintain File Descriptor (COF) - Flags



```
 Move >> COS                   NonStop Workload Automation                 12 OCT 22
                                       MBAT.CONFIG                         MULTIBATCH
                       ** Maintain OSS STEP Configuration (COS) **

          BMON Process:  \INSIDER $MBBMN__

         OSS Step Name:  JOBF AA 01 _ Alias: ______________________________

           Description:  ________________________________________________

18 Basic(1)  Basic(2)  Options  Sched. Cond. Defines  EnvC  Env  FdC Fd Index
    FD Cwd  : /E/insider/daveh________________________________________________
              ________________________________________________________________
              ________________________________________________________________
              ________________________________________________________________

    FD Class: FDC1____
              ________
              ________
              ________
              ________
F1-LIST        F2-AMEND     F3-DELETE        F4-INSERT       SF13-VIEW FUNC KEYS

                                                                         BLOCK
```

Figure 3.20 – Maintain OSS Step Configuration (COS) – File Descriptor Class

FD Overlay (Figure 3.21)

The File Descriptor is the equivalent of a GUARDIAN file assignment, linking together a logical file number with a physical file name.

FD Number: 0 is standard input, 1 is standard output, 2 is standard error, otherwise it is user-defined.

Duplicate FD: Determines whether this File descriptor is a duplicate of an earlier declared descriptor.

Valid Values are:

-1	This File descriptor is not a duplicate.
>0	This File descriptor is a duplicate of one specified earlier.

Access Mode (Supply one only):

RO	Open for reading only.
RW	Open for reading and writing.
WO	Open for writing only.

Flags:

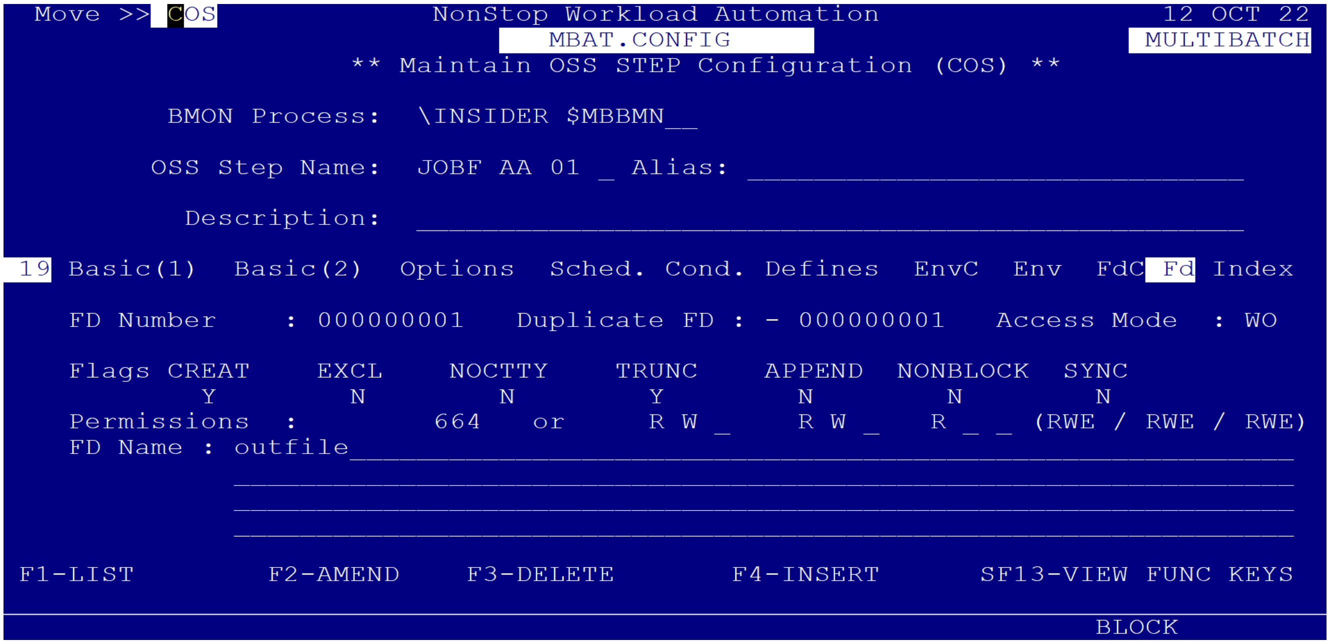
CREAT	Create file if it does not exist.
EXCL	Exclusive use.
NOCTTY	Do not assign controlling terminal.
TRUNC	Truncate.
APPEND	Set append mode.
NONBLOCK	Non-blocking mode.
SYNC	Write according to synchronised I/O file integrity completion.

Permissions: A User can either set the permissions by providing the Octal number or the alphabetic string. For example:

```
%0777 is RWE/RWE/RWE
%0104 is --E/---/R--
%0547 is R-E/R--/RWE
```

First RWE	= Owner permissions
Second RWE	= Group permissions
Third RWE	= Others permissions

FD Name: Is a string containing the OSS pathname of the file to be opened by the new process. The string is case sensitive. The string can contain MultiBatch conditional parameters.

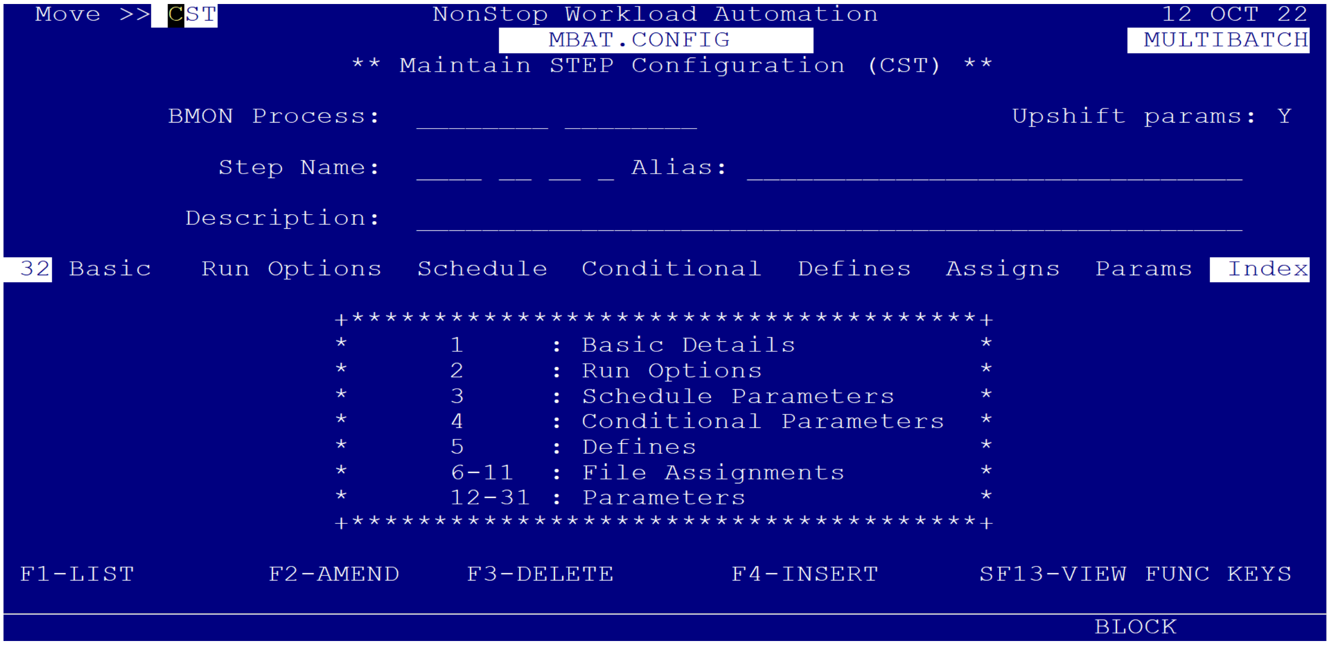


```
 Move >> COS              NonStop Workload Automation                  12 OCT 22
                              MBAT.CONFIG                            MULTIBATCH
                       ** Maintain OSS STEP Configuration (COS) **

        BMON Process:  \INSIDER $MBBMN__

       OSS Step Name:  JOBF AA 01 _ Alias: ______________________________

          Description: __________________________________________________

19 Basic(1)  Basic(2)  Options  Sched. Cond. Defines  EnvC  Env  FdC Fd  Index

   FD Number    : 000000001   Duplicate FD : - 000000001   Access Mode  : WO

   Flags CREAT    EXCL    NOCTTY    TRUNC    APPEND  NONBLOCK  SYNC
           Y        N        N        Y        N        N        N
   Permissions  :        664   or     R W _    R W _   R _ _  (RWE / RWE / RWE)
   FD Name : outfile________________________________________________________
             _______________________________________________________________
             _______________________________________________________________
             _______________________________________________________________

 F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                  BLOCK
```

Figure 3.21 – Maintain OSS Step Configuration (COS) – File Descriptor

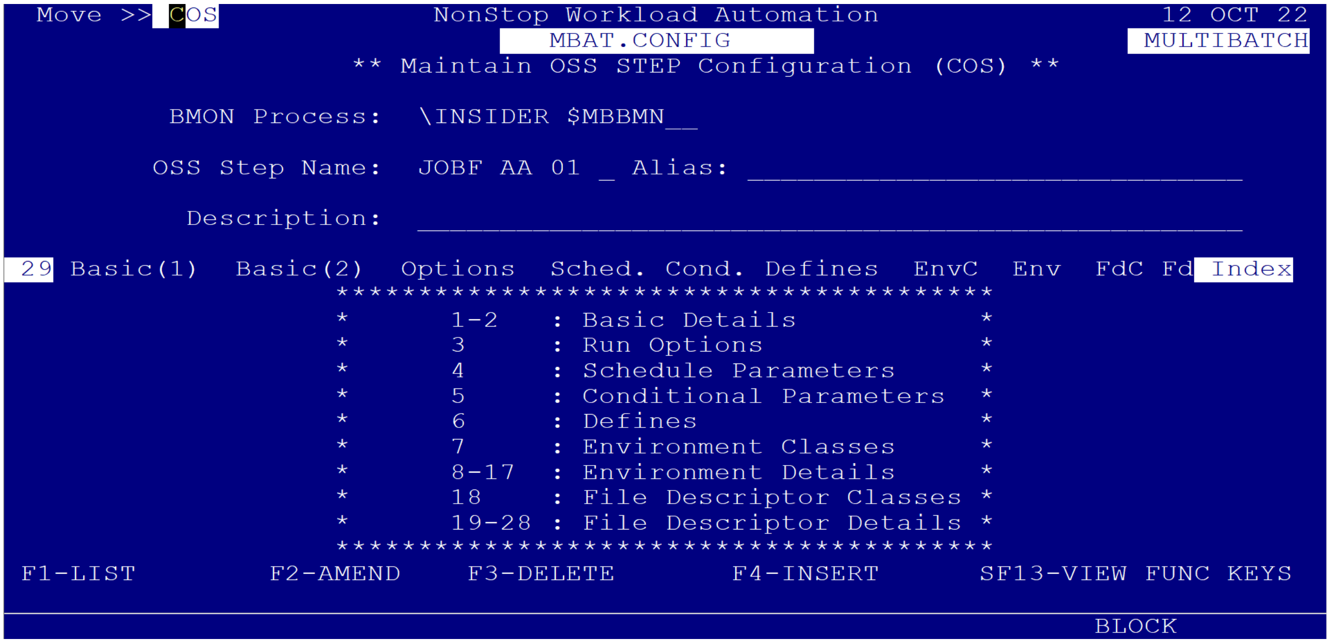
3.11 Index Overlays

As with the Job and Segment configuration screens, the Index Overlay (Figure 3.22 & 3.23) provides a list of the pages available.



```
 Move >> CST              NonStop Workload Automation                  12 OCT 22
                              MBAT.CONFIG                            MULTIBATCH
                       ** Maintain STEP Configuration (CST) **

        BMON Process:  ________ ________                     Upshift params: Y

           Step Name:  ____ __ __ _ Alias: ______________________________

         Description:  __________________________________________________

32 Basic   Run Options  Schedule  Conditional  Defines  Assigns  Params  Index

                    +*************************************+
                    *      1     : Basic Details          *
                    *      2     : Run Options            *
                    *      3     : Schedule Parameters    *
                    *      4     : Conditional Parameters *
                    *      5     : Defines                *
                    *      6-11  : File Assignments       *
                    *      12-31 : Parameters             *
                    +*************************************+

 F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                  BLOCK
```

Figure 3.22 – Maintain Step Configuration (CST) - Index



```
 Move >> COS                NonStop Workload Automation                 12 OCT 22
                                  MBAT.CONFIG                          MULTIBATCH
                       ** Maintain OSS STEP Configuration (COS) **

          BMON Process:  \INSIDER $MBBMN__

         OSS Step Name:  JOBF AA 01 _ Alias: ______________________________

           Description:  ________________________________________________

 29 Basic(1)  Basic(2)  Options  Sched. Cond. Defines  EnvC  Env  FdC Fd Index
                     ***************************************
                     *      1-2    : Basic Details            *
                     *      3      : Run Options              *
                     *      4      : Schedule Parameters      *
                     *      5      : Conditional Parameters   *
                     *      6      : Defines                  *
                     *      7      : Environment Classes      *
                     *      8-17   : Environment Details      *
                     *      18     : File Descriptor Classes *
                     *      19-28  : File Descriptor Details *
                     ***************************************
 F1-LIST        F2-AMEND     F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                     BLOCK
```

Figure 3.23 – Maintain OSS Step Configuration (COS) – Index

3.12 Configure Defines (CDF)

A DEFINE is a named set of attributes and values which is used to pass the definition of an entity to a process.

DEFINES for example, are processed by the tape process, spooler collector process or the file system process.

To configure DEFINES, navigate to the Entity Configuration Menu (CMM) screen and press the **F7** – CONFIGURE DEFINES function key. This accesses the Maintain Defines Screen (CDF) (Figure 3.24-31) which comprises eight sub-screens or overlays with each overlay relating to a specific DEFINE type (Tape, TapeCatalog, Spool, Map, Catalog, Defaults, Sort, or Subsort).

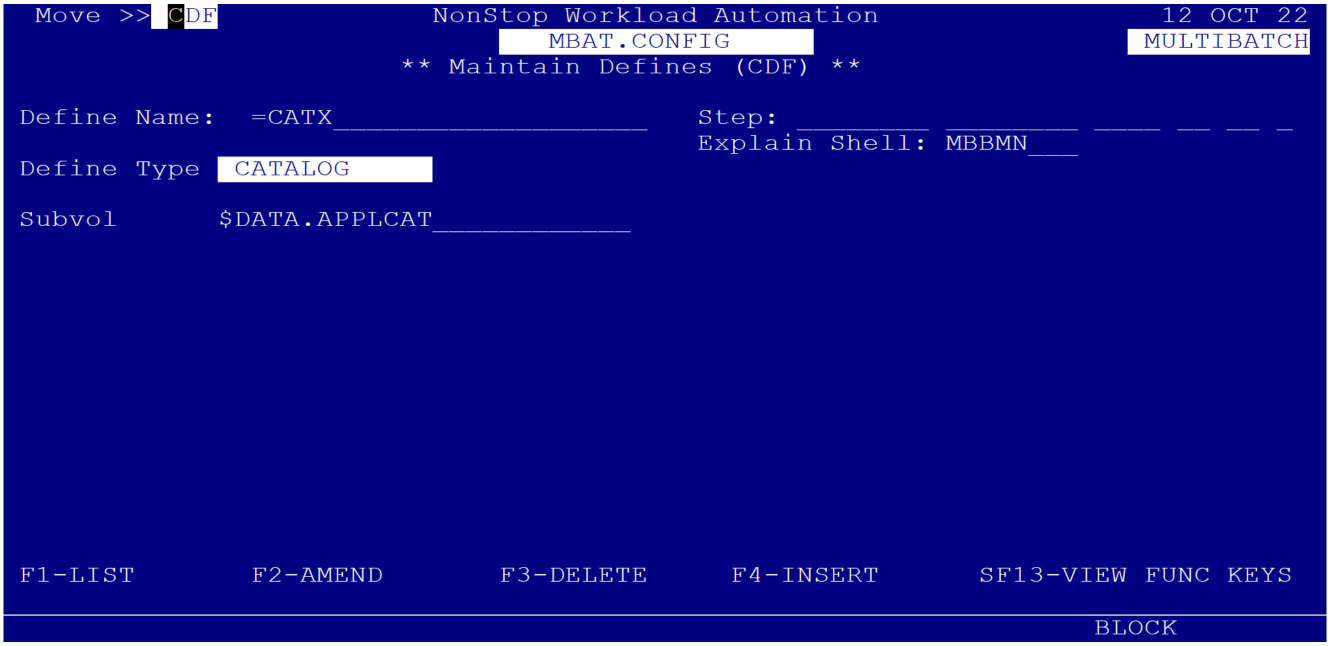
It is possible to create Global and Local Defines in the Define Database. A Global Define can be identified by the fact that the STEP value is set to spaces. When a STEP is retrieving Defines it attempts to read the Define database on a key of Define name and a Step name of spaces. This allows the user to generate Global Defines which can be accessed by many Steps and Local Defines which can only be accessed by a single Step. For example:

=MYSPOOL	JOBA.AA.01
=MYSPOOL	JOBA.AA.02
=MYSPOOL	(this Define is accessed by Steps other than the 2 above.)

The List Defines Screen (CLD) gives a list of all the DEFINES in the MultiBatch system together with the corresponding Entity types and Define types.

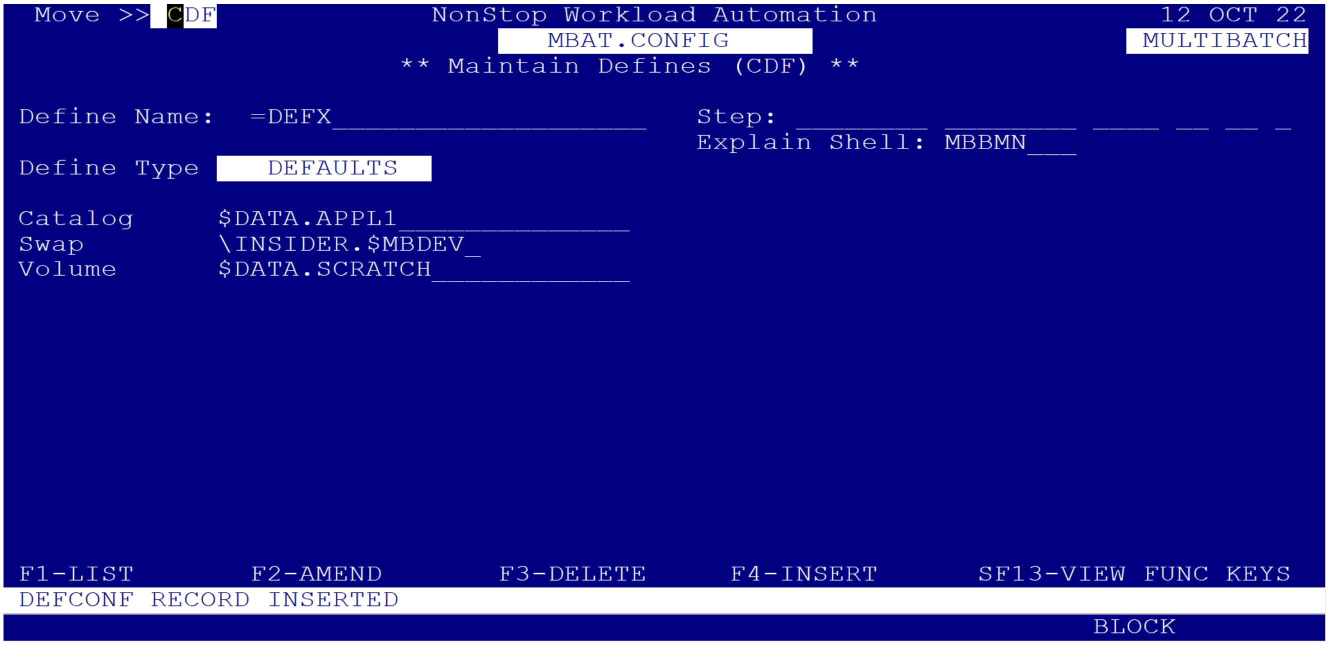
The **DEFINE** overlay on the Maintain STEP Configuration (CST) screen allows the user to configure DEFINES for STEPS. You can obtain detail on a DEFINE by placing a marker in the first column and press **F8** – Display Detail function key. This displays the appropriate Maintain Defines (CDF) overlay.

DEFINES can also be used on the Assign overlay or in the infile/outfile fields of the Run Options of the Maintain STEP Configuration (CST) / Maintain OSS STEP Configuration (COS) screen.



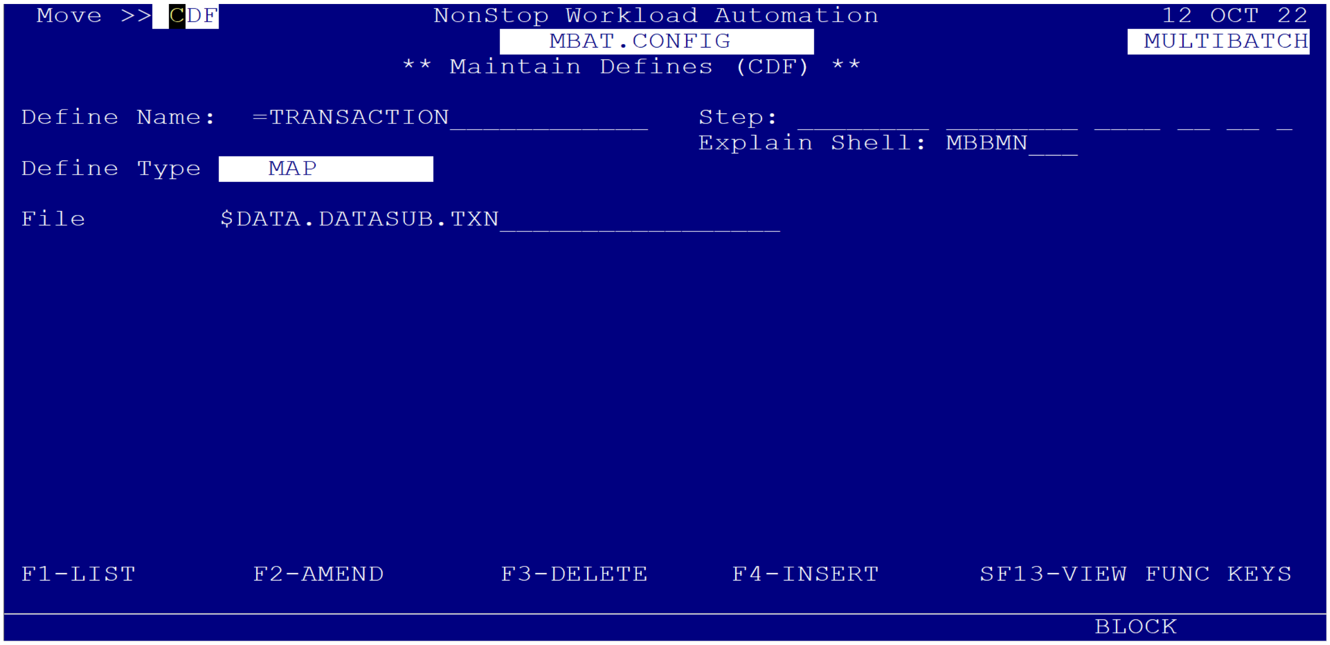
```
 Move >> CDF                   NonStop Workload Automation                12 OCT 22
                                    MBAT.CONFIG                         MULTIBATCH
                             ** Maintain Defines (CDF) **

Define Name:  =CATX____________________     Step: ________ ________ ____ __ __ _
                                            Explain Shell: MBBMN___
Define Type  CATALOG

Subvol       $DATA.APPLCAT____________









F1-LIST        F2-AMEND        F3-DELETE      F4-INSERT       SF13-VIEW FUNC KEYS

                                                                    BLOCK
```

Figure 3.24 – Maintain Defines Screen (CDF) (overlay 1 of 8)



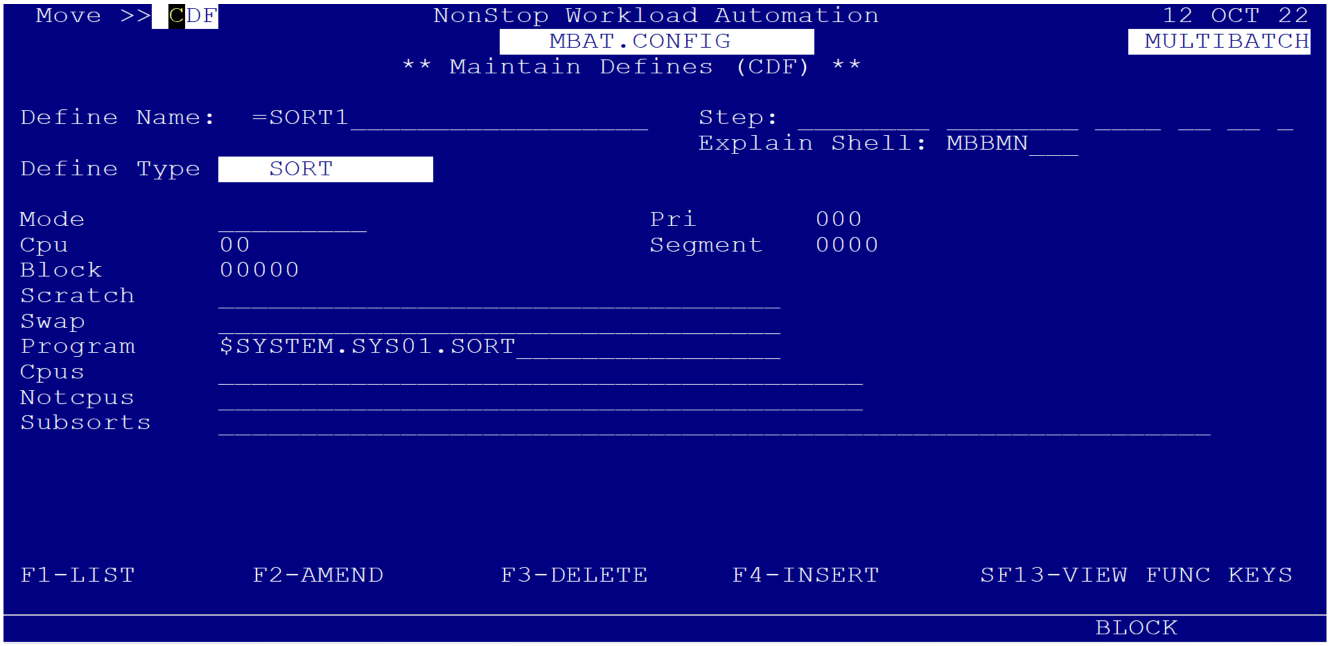
```
 Move >> CDF                 NonStop Workload Automation                  12 OCT 22
                                    MBAT.CONFIG                           MULTIBATCH
                             ** Maintain Defines (CDF) **

Define Name:  =DEFX___________________     Step: ________ ________ ____ __ __ _
                                           Explain Shell: MBBMN___
Define Type    DEFAULTS

Catalog       $DATA.APPL1_____________
Swap          \INSIDER.$MBDEV_
Volume        $DATA.SCRATCH____________











F1-LIST       F2-AMEND       F3-DELETE     F4-INSERT      SF13-VIEW FUNC KEYS
DEFCONF RECORD INSERTED
                                                                 BLOCK
```

Figure 3.25 – Maintain Defines Screen (CDF) (overlay 2 of 8)



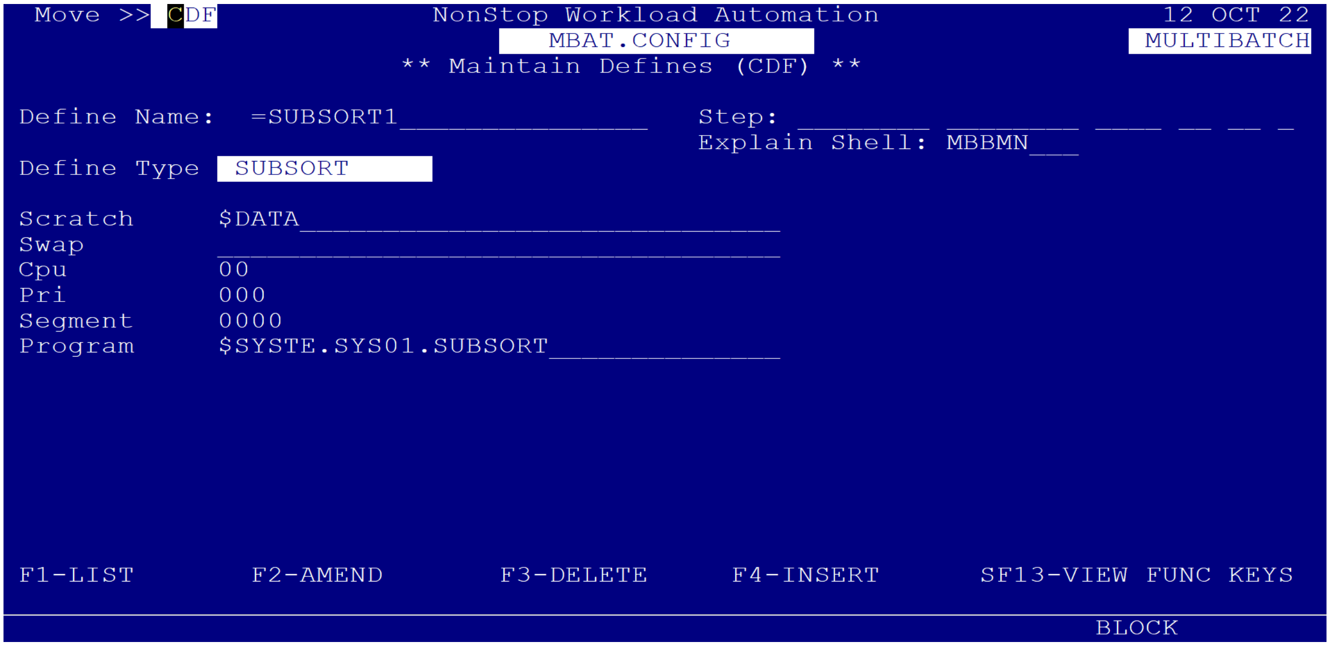
```
 Move >> CDF                 NonStop Workload Automation                  12 OCT 22
                                    MBAT.CONFIG                           MULTIBATCH
                             ** Maintain Defines (CDF) **

Define Name:  =TRANSACTION____________     Step: ________ ________ ____ __ __ _
                                           Explain Shell: MBBMN___
Define Type    MAP

File          $DATA.DATASUB.TXN_________________














F1-LIST       F2-AMEND       F3-DELETE     F4-INSERT      SF13-VIEW FUNC KEYS

                                                                 BLOCK
```

Figure 3.26 – Maintain Defines Screen (CDF) (overlay 3 of 8)



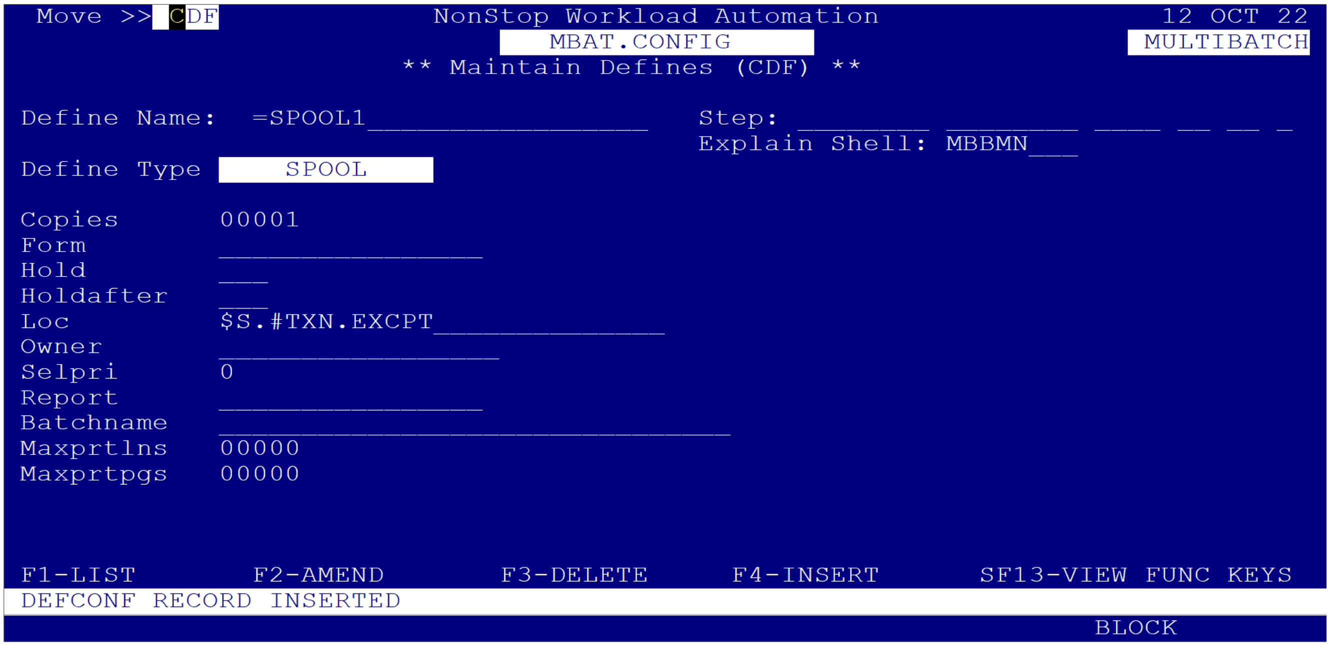
```
 Move >> CDF                  NonStop Workload Automation               12 OCT 22
                                    MBAT.CONFIG                        MULTIBATCH
                             ** Maintain Defines (CDF) **

Define Name:   =SORT1__________________    Step: ________ ________ ____ __ __ _
                                           Explain Shell: MBBMN___

Define Type       SORT

Mode           _________                Pri        000
Cpu            00                       Segment    0000
Block          00000
Scratch        __________________________________
Swap           __________________________________
Program        $SYSTEM.SYS01.SORT________________
Cpus           _______________________________________
Notcpus        _______________________________________
Subsorts       ____________________________________________________________




F1-LIST       F2-AMEND       F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS

                                                                  BLOCK
```

Figure 3.27 – Maintain Defines Screen (CDF) (overlay 4 of 8)



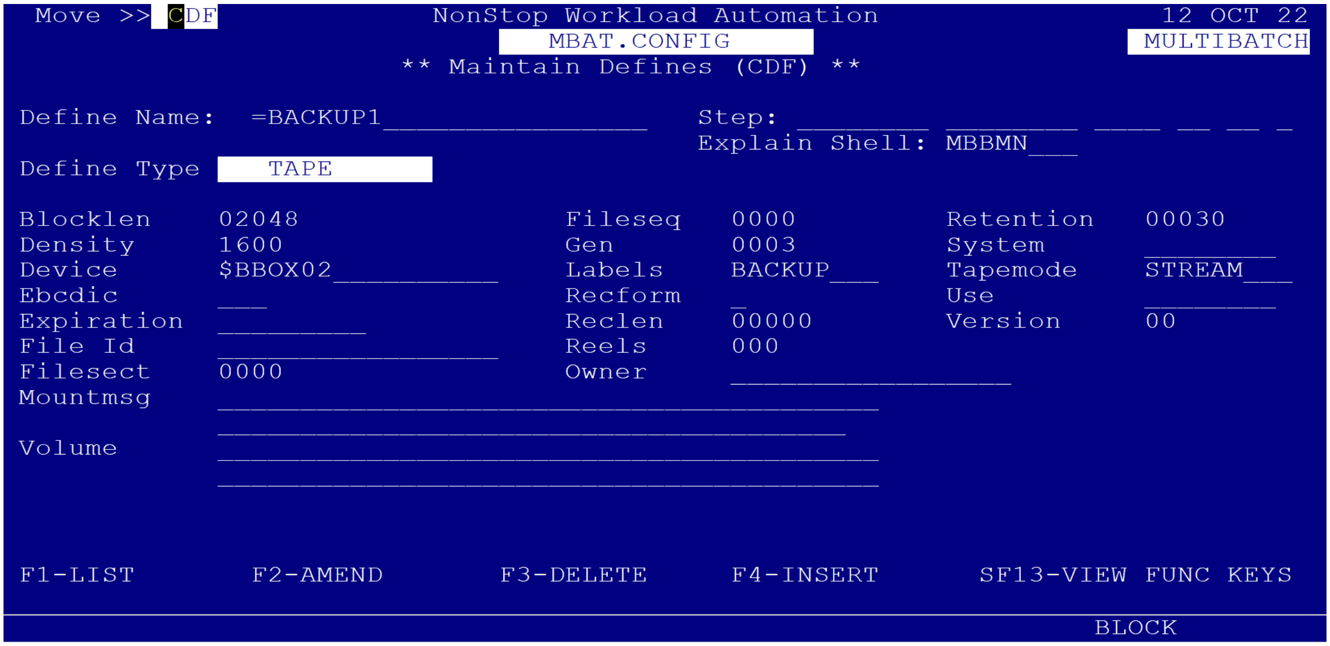
```
 Move >> CDF                  NonStop Workload Automation               12 OCT 22
                                    MBAT.CONFIG                        MULTIBATCH
                             ** Maintain Defines (CDF) **

Define Name:   =SUBSORT1_______________    Step: ________ ________ ____ __ __ _
                                           Explain Shell: MBBMN___

Define Type     SUBSORT

Scratch        $DATA_____________________________
Swap           __________________________________
Cpu            00
Pri            000
Segment        0000
Program        $SYSTE.SYS01.SUBSORT______________







F1-LIST       F2-AMEND       F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS

                                                                  BLOCK
```

Figure 3.28 – Maintain Defines Screen (CDF) (overlay 5 of 8)



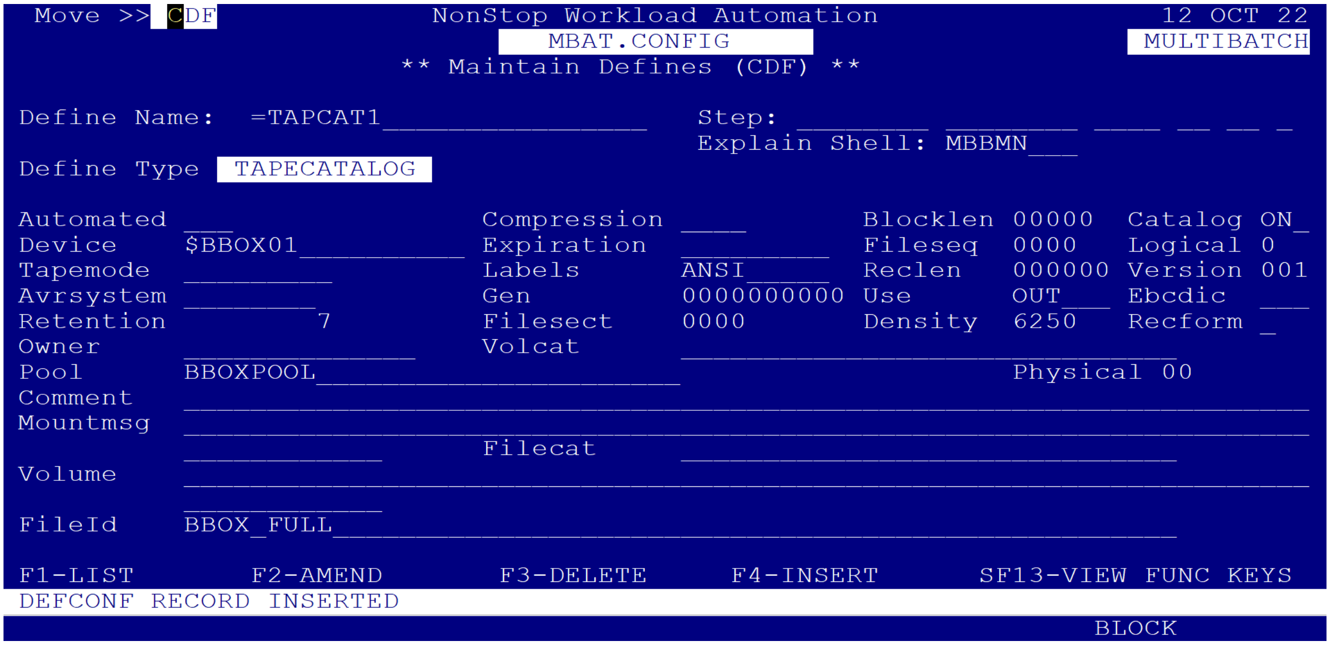
```
 Move >> CDF                 NonStop Workload Automation                12 OCT 22
                                    MBAT.CONFIG                        MULTIBATCH
                          ** Maintain Defines (CDF) **

Define Name:  =SPOOL1_________________          Step: ________ ________ ____ __ __ _
                                                Explain Shell: MBBMN___
Define Type     SPOOL

Copies       00001
Form         ________________
Hold         ___
Holdafter    ___
Loc          $S.#TXN.EXCPT______________
Owner        _________________
Selpri       0
Report       ________________
Batchname    _______________________________
Maxprtlns    00000
Maxprtpgs    00000



F1-LIST       F2-AMEND       F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS
DEFCONF RECORD INSERTED
                                                                      BLOCK
```

Figure 3.29 – Maintain Defines Screen (CDF) (overlay 6 of 8)



```
 Move >> CDF                 NonStop Workload Automation                12 OCT 22
                                    MBAT.CONFIG                        MULTIBATCH
                          ** Maintain Defines (CDF) **

Define Name:  =BACKUP1________________          Step: ________ ________ ____ __ __ _
                                                Explain Shell: MBBMN___
Define Type    TAPE

Blocklen     02048                   Fileseq    0000         Retention    00030
Density      1600                    Gen        0003         System       ________
Device       $BBOX02__________       Labels     BACKUP___    Tapemode     STREAM___
Ebcdic       ___                     Recform    _            Use          ________
Expiration   _________               Reclen     00000        Version      00
File Id      _________________       Reels      000
Filesect     0000                    Owner      _________________
Mountmsg     ________________________________________
             ______________________________________
Volume       ________________________________________
             ________________________________________


F1-LIST       F2-AMEND       F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS

                                                                      BLOCK
```

Figure 3.30 – Maintain Defines Screen (CDF) (sheet 7 of 8)



```
 Move >> CDF              NonStop Workload Automation                  12 OCT 22
                                MBAT.CONFIG                         MULTIBATCH
                       ** Maintain Defines (CDF) **

Define Name:  =TAPCAT1________________        Step: ________ ________ ____ __ __ _
                                              Explain Shell: MBBMN___
Define Type  TAPECATALOG

Automated ___                Compression ____       Blocklen 00000  Catalog ON_
Device    $BBOX01__________  Expiration  _________  Fileseq  0000   Logical 0
Tapemode  _________          Labels      ANSI_____  Reclen   000000 Version 001
Avrsystem ________           Gen         0000000000 Use      OUT___ Ebcdic  ___
Retention         7          Filesect    0000       Density  6250   Recform _
Owner     ______________     Volcat      ______________________________
Pool      BBOXPOOL______________________                     Physical 00
Comment   ____________________________________________________________________
Mountmsg  ____________________________________________________________________
          ____________       Filecat     ______________________________
Volume    ____________________________________________________________________
          ____________
FileId    BBOX_FULL___________________________________________________

F1-LIST       F2-AMEND       F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS
DEFCONF RECORD INSERTED
                                                                  BLOCK
```

Figure 3.31 – Maintain Defines Screen (CDF) (sheet 8 of 8)

3.13 Assign Class Definition (CAS)

The Assign Class feature allows the definition and maintenance of groups of ASSIGNs under a generic CLASS name which uniquely identifies that grouping within the MultiBatch environment.

It is beneficial to define an ASSIGN CLASS for a file referenced in many different parts of the system as this provides a central location for any changes.

If an ASSIGN for the same file exists for more than one CST record or BMON process, then all the CST records where the file is referenced need to be changed. You could, however, define an ASSIGN CLASS so that the change need only be made once in the Maintain Assign Class Screen (CAS) (Figure 3.32).

Overlays 6-11 of the Maintain Step Configuration screen, as previously described, allow the user to enter ASSIGNS. In order to refer to an Assign Class from these overlays the Physical column must be empty. Placing a marker beside an Assign Logical name and pressing the **F8**-Detail function key accesses the Maintain Assign Class Screen.

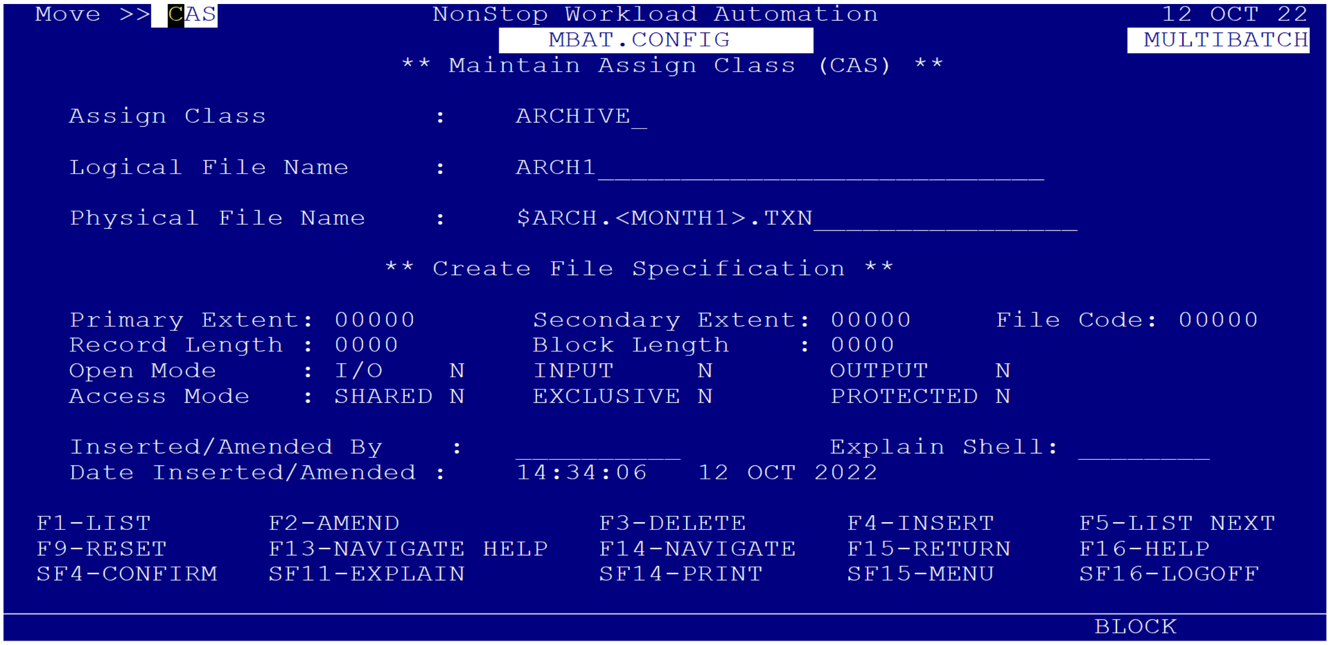
Note 1: *You cannot have an Assign Class called TACLCSTM. This allows MultiBatch to support the ASSIGN TACLCSTM, null feature.*

A standard feature of NonStop file assignment processing is the ability to submit information in the assign message that will allow an application to create a non-existent file. This extra information includes file code and extent sizes.

The feature is rarely used so to keep the CST assign overlays clear, if a user wishes to utilise the "create specification" feature then it has to be configured through the Assign Class screen.

Note 2: *The maximum value for the LOGICAL FILE NAME field in MultiBatch is 31 characters.*

Note 3: *The maximum fileset value accepted by MultiBatch is 34 characters.*

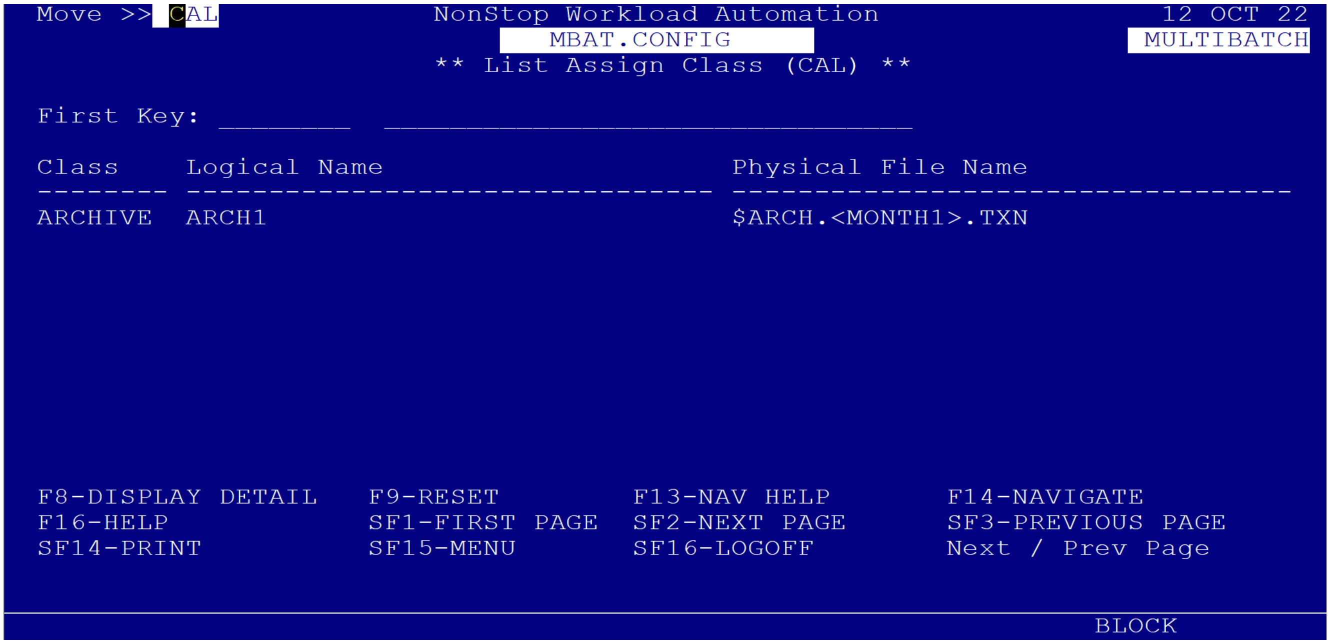


```
 Move >> CAS                 NonStop Workload Automation                 12 OCT 22
                                     MBAT.CONFIG                         MULTIBATCH
                           ** Maintain Assign Class (CAS) **

   Assign Class            :     ARCHIVE_

   Logical File Name       :     ARCH1___________________________

   Physical File Name      :     $ARCH.<MONTH1>.TXN________________

                          ** Create File Specification **

   Primary Extent: 00000        Secondary Extent: 00000     File Code: 00000
   Record Length : 0000         Block Length    : 0000
   Open Mode     : I/O     N    INPUT     N       OUTPUT    N
   Access Mode   : SHARED  N    EXCLUSIVE N       PROTECTED N

   Inserted/Amended By     :     __________       Explain Shell: ________
   Date Inserted/Amended :       14:34:06   12 OCT 2022

 F1-LIST       F2-AMEND             F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET      F13-NAVIGATE HELP    F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM   SF11-EXPLAIN         SF14-PRINT     SF15-MENU      SF16-LOGOFF

                                                                  BLOCK
```

Figure 3.32 - Maintain Assign Class Screen (CAS)

3.14 List Assign Class (CAL)

The List Assign Class Screen (CAL) (Figure 3.33) provides a list of all the ASSIGN Classes defined in the system together with the corresponding Logical/Physical filenames.



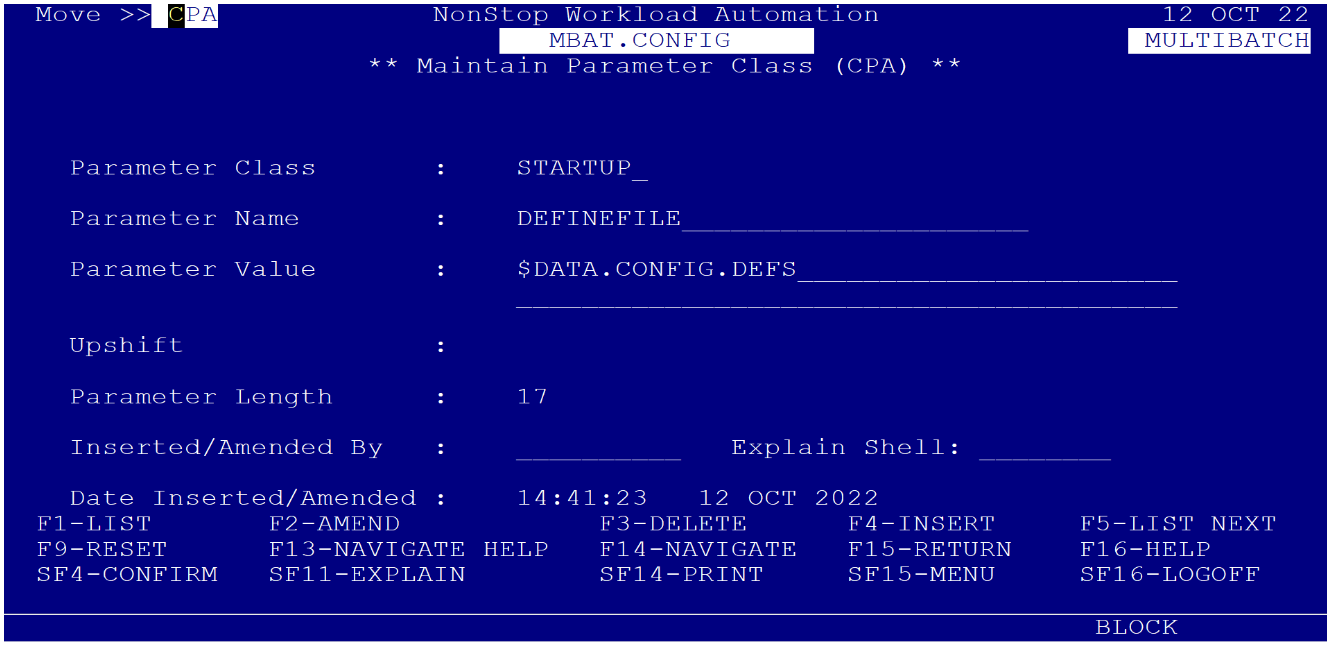
```
Move >> CAL                NonStop Workload Automation               12 OCT 22
                                   MBAT.CONFIG                     MULTIBATCH
                           ** List Assign Class (CAL) **

First Key: ________  ________________________________

Class     Logical Name                      Physical File Name
--------  --------------------------------  ----------------------------------
ARCHIVE   ARCH1                             $ARCH.<MONTH1>.TXN









F8-DISPLAY DETAIL   F9-RESET        F13-NAV HELP       F14-NAVIGATE
F16-HELP            SF1-FIRST PAGE  SF2-NEXT PAGE      SF3-PREVIOUS PAGE
SF14-PRINT          SF15-MENU       SF16-LOGOFF        Next / Prev Page


                                                                 BLOCK
```

Figure 3.33 – List Assign Class Screen (CAL)

3.15 Param Class Definition (CPA)

The PARAM Class Definition feature is available via the Maintain Parameter Class Screen (CPA) (Figure 3.34), which allows the definition and maintenance of groups of PARAM messages under generic CLASS names, uniquely identifying the grouping within the MultiBatch environment.

It is beneficial to define a PARAM CLASS for a parameter referenced in many different parts of the system as this provides a central location for any changes. If the same PARAM is referenced by more than one CST record or BMON process, then all the CST records where the PARAM is referenced, need to be changed. You could, however, define a PARAM Class so that the change would only need to be made on the Maintain Param Class (CPA) Screen (Figure 3.34).



```
 Move >> CPA                NonStop Workload Automation                  12 OCT 22
                              MBAT.CONFIG                               MULTIBATCH
                      ** Maintain Parameter Class (CPA) **


   Parameter Class          :    STARTUP_

   Parameter Name           :    DEFINEFILE_____________________

   Parameter Value          :    $DATA.CONFIG.DEFS_______________________
                                 ________________________________________

   Upshift                  :

   Parameter Length         :    17

   Inserted/Amended By      :    __________   Explain Shell: ________

   Date Inserted/Amended :       14:41:23   12 OCT 2022
 F1-LIST        F2-AMEND                F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET       F13-NAVIGATE HELP       F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM    SF11-EXPLAIN            SF14-PRINT     SF15-MENU      SF16-LOGOFF

                                                                   BLOCK
```

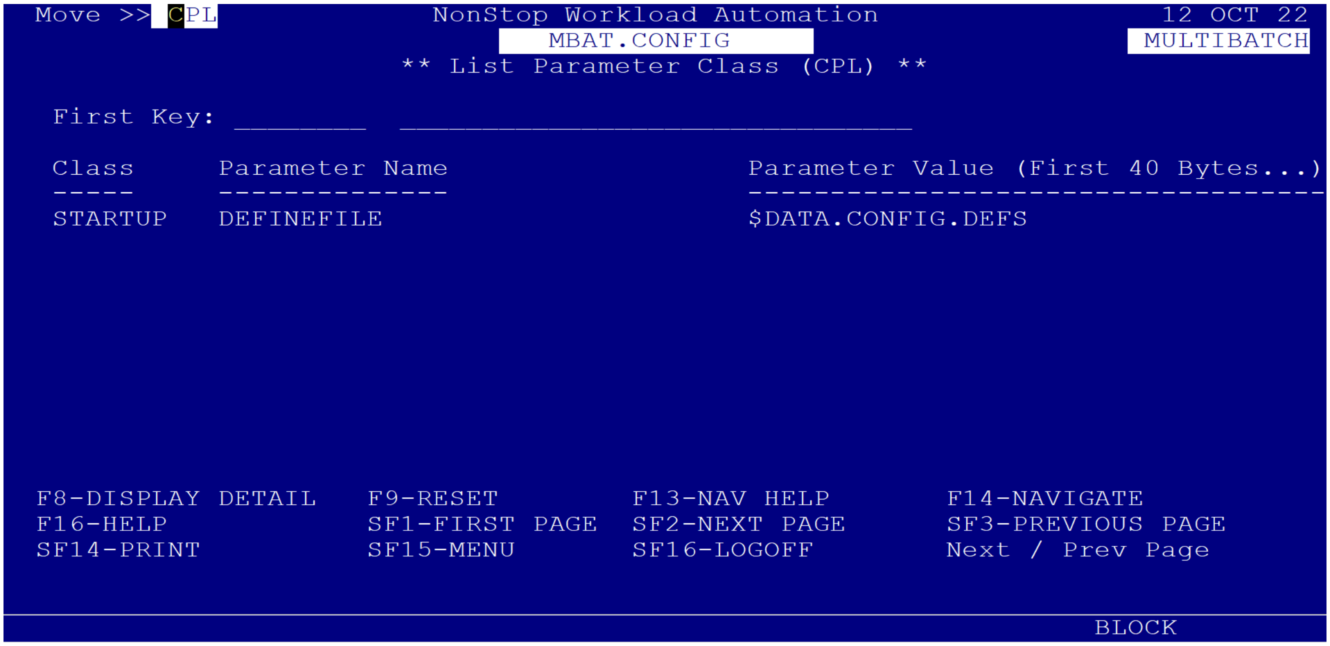
Figure 3.34 – Maintain Parameter Class Screen (CPA)

The **PARAM overlays** (Figure 3.14) of the Maintain Step Configuration Screen (as previously described) allow the user to enter PARAMs. In order to refer to a PARAM Class from these overlays the Value column must be empty. Placing a marker beside a Param Name and pressing the **F8** – Detail function key accesses the Maintain Parameter Class (CPA) screen.

Upshift: By default, the Parameter Value when saved is Upshifted. A user can enter N in the Upshift field, to ensure any lower-case values are retained.

Note 1: *The maximum Parameter Name Length in MultiBatch is 30 characters.*

The **List Param Class** (CPL) Screen (Figure 3.35) provides a list of all the PARAM classes defined in the system together with the corresponding Names/Values.



```
 Move >> CPL                 NonStop Workload Automation                  12 OCT 22
                                    MBAT.CONFIG                          MULTIBATCH
                            ** List Parameter Class (CPL) **

  First Key: ________  _______________________________

  Class     Parameter Name                  Parameter Value (First 40 Bytes...)
  -----     --------------                  -----------------------------------
  STARTUP   DEFINEFILE                      $DATA.CONFIG.DEFS









 F8-DISPLAY DETAIL   F9-RESET        F13-NAV HELP        F14-NAVIGATE
 F16-HELP            SF1-FIRST PAGE  SF2-NEXT PAGE       SF3-PREVIOUS PAGE
 SF14-PRINT          SF15-MENU       SF16-LOGOFF         Next / Prev Page


                                                                BLOCK
```

Figure 3.35 - List Parameter Class Screen (CPL)

3.16 File Descriptor Definition (COF)

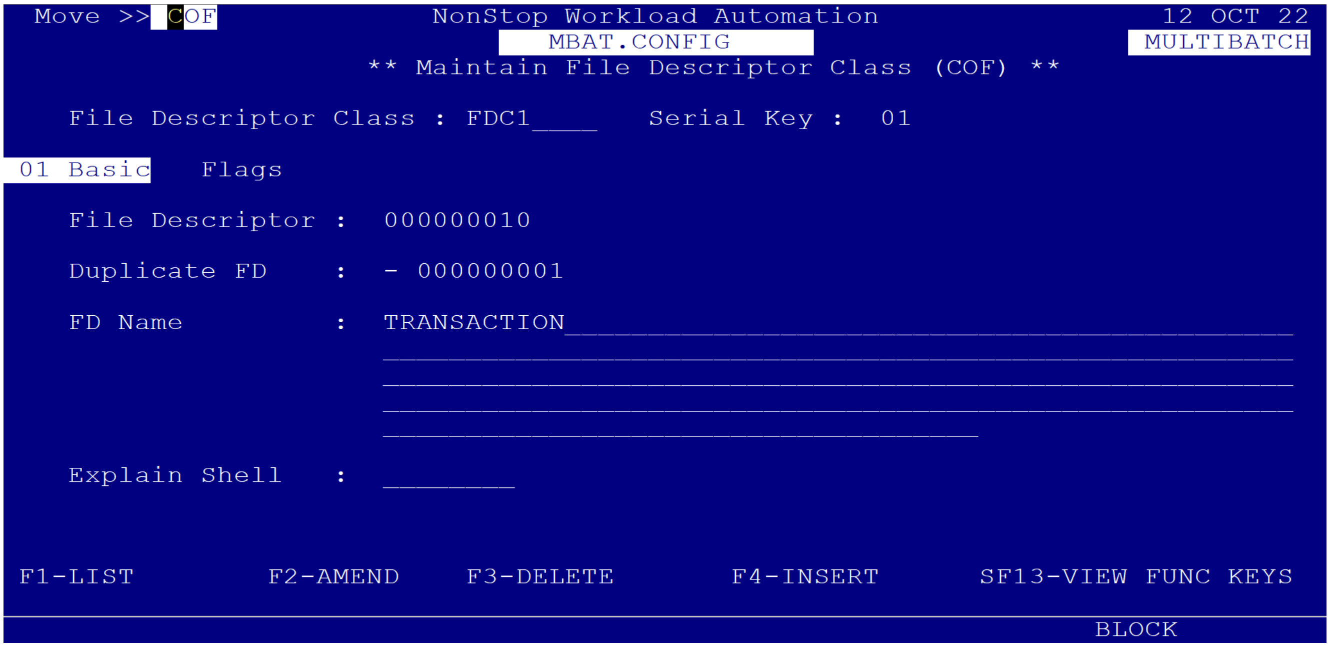
The File Descriptor Class feature allows the definition and maintenance of groups of File Descriptors under a generic CLASS name which uniquely identifies that grouping within the MultiBatch environment.

It is beneficial to define a FILE DESCRIPTOR CLASS for a file descriptor referenced in many different parts of the system as this provides a central location for any changes. If a FILE DESCRIPTOR for the same file exists for more than one BMON process, then all the COS records where the file descriptor is referenced need to be changed. It is possible, however, to define a FILE DESCRIPTOR CLASS so that the change need only be made on the Maintain File Descriptor Class screen (COF) (Figure 3.36). This is accessed via the F2 – Configure OSS FD Class function key on the OSS Step Configuration Menu (COM).

Overlays 19 – 28 of the Maintain OSS Step Configuration screen (COS), as previously described allow the user to enter FILE DESCRIPTORS. In order to refer to a File Descriptor Class from the Maintain OSS Step Configuration screen (COS) the File Descriptor Class name must be inserted into the first blank line of the FD Class field on the FDC overlay which is 20th overlay. Placing the cursor on the same line as a File Descriptor class and pressing the F8-Detail function key accesses the Maintain File Descriptor Class screen.

"**SERIAL**". This numeric field has a default value of "01", and can range from "01" to "99". The serial number is used to create and identify different file descriptor records for the same file descriptor class. This field is mandatory.

The Flags overlay can be seen in Figure 3.19 – Maintain File Descriptor (COF) – Flags above.

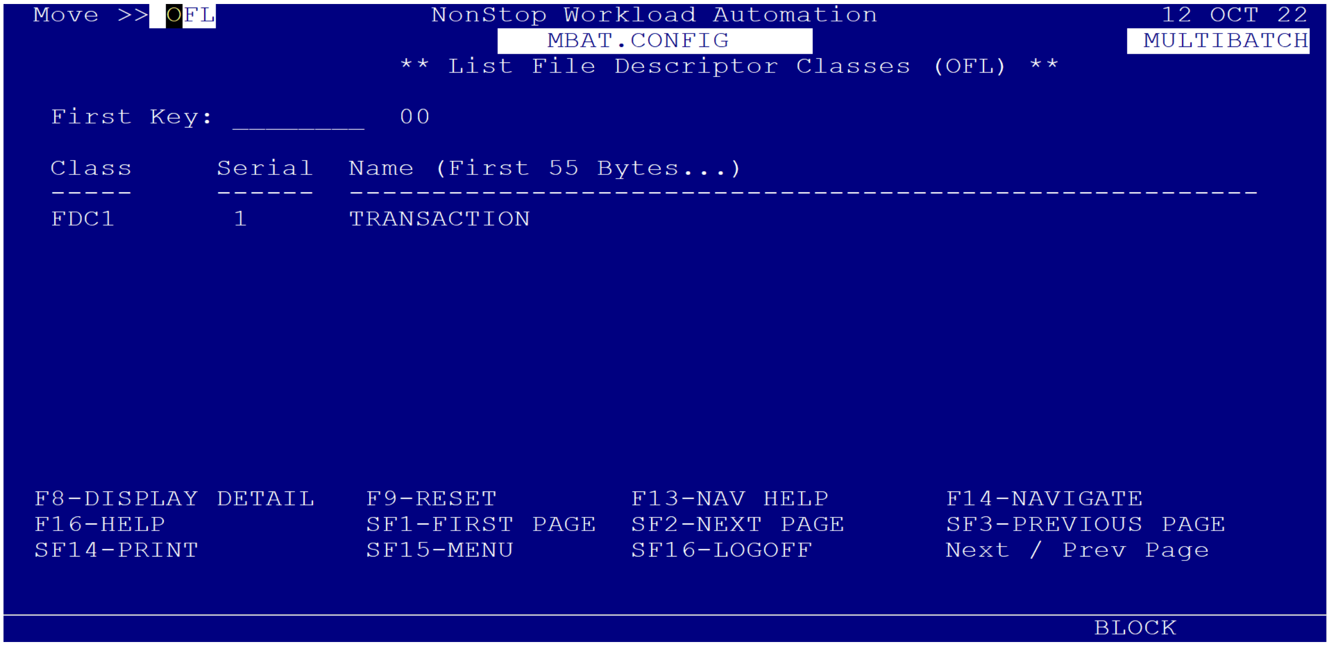


```
 Move >> COF            NonStop Workload Automation                12 OCT 22
                              MBAT.CONFIG                          MULTIBATCH
                    ** Maintain File Descriptor Class (COF) **

   File Descriptor Class : FDC1____    Serial Key :  01

 01 Basic    Flags

   File Descriptor :  000000010

   Duplicate FD    :  - 000000001

   FD Name         :  TRANSACTION_____________________________________________
                      ________________________________________________________
                      ________________________________________________________
                      ________________________________________________________
                      ____________________________________

   Explain Shell   :  ________


 F1-LIST        F2-AMEND     F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                   BLOCK
```

Figure 3.36 Maintain File Descriptor Class (COF)

3.17 List File Descriptor Class (OFL)

The List File Descriptor Class Screen (OFL) (Figure 3.37) is accessed from the Entity Configuration Menu Screen (CMM). This provides a list of all the File Descriptor classes defined in the system together with the first 55 bytes of the corresponding File Descriptor name



```
 Move >> OFL               NonStop Workload Automation              12 OCT 22
                                 MBAT.CONFIG                      MULTIBATCH
                        ** List File Descriptor Classes (OFL) **

  First Key: ________   00

  Class      Serial  Name (First 55 Bytes...)
  -----      ------  -------------------------------------------------------
  FDC1         1     TRANSACTION










 F8-DISPLAY DETAIL   F9-RESET        F13-NAV HELP        F14-NAVIGATE
 F16-HELP            SF1-FIRST PAGE  SF2-NEXT PAGE       SF3-PREVIOUS PAGE
 SF14-PRINT          SF15-MENU       SF16-LOGOFF         Next / Prev Page

                                                                 BLOCK
```

Figure 3.37 – List File Descriptor Class Screen (OFL)

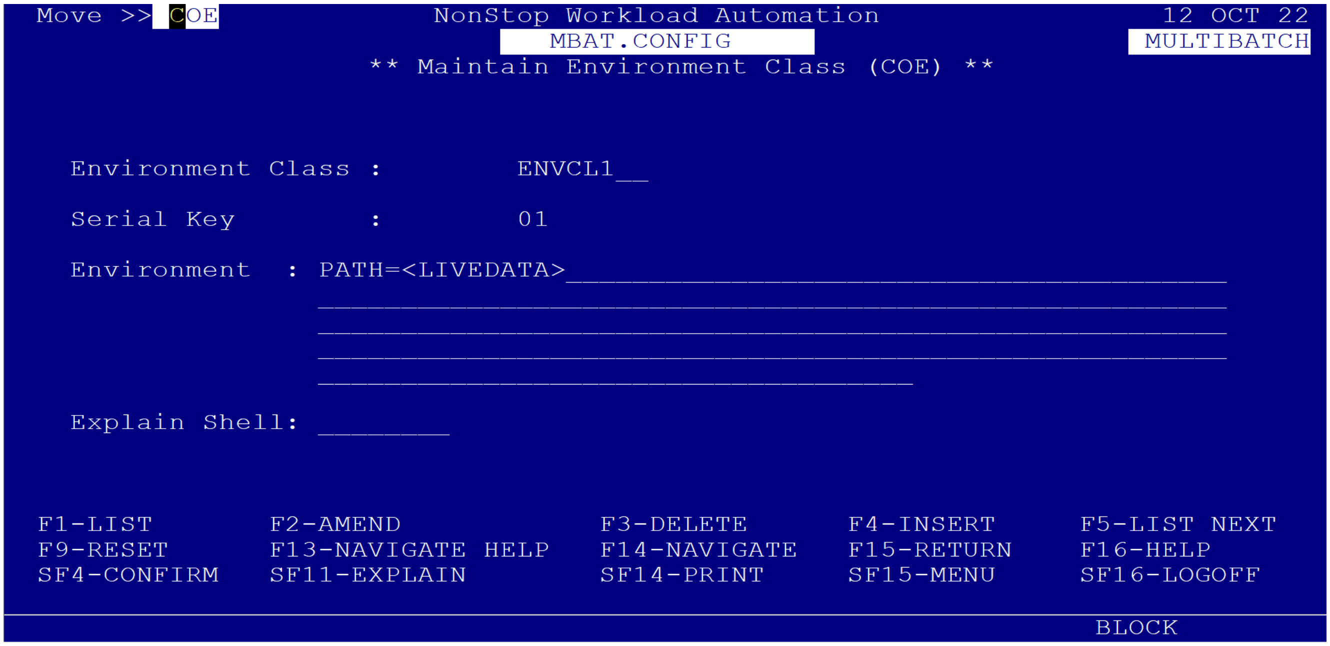
3.18 Environment Definition (COE)

The Environment Class feature allows the definition and maintenance of groups of Environments under a generic CLASS name which uniquely identifies that grouping within the MultiBatch environment.

It is beneficial to define an ENVIRONMENT CLASS for an environment parameter referenced in many different parts of the system as this provides a central location for any changes. If an ENVIRONMENT parameter for the same variable exists for more than one BMON process, then all the COS records where the environment is referenced need to be changed. It is possible, however, to define an ENVIRONMENT class so that the change need only be made on the Maintain Environment Class Screen (COE) (Figure 3.38). This is accessed via the F3 – Configure OSS Env Class function key on the Maintain OSS Step Configuration Menu (COM).

Overlays 8 – 17 of the Maintain OSS Step Configuration screen (COS), as previously described allow the user to enter ENVIRONMENT parameters. In order to refer to an Environment Class from the Maintain OSS Step Configuration screen (COS) the Environment Class name must be inserted into the first blank line of the Env Class field on the EnvC overlay. Placing the cursor on the same line as an Environment class and pressing the F8-Detail function key accesses the Maintain Environment Class screen.

SERIAL". This numeric field has a default value of "01", and can range from "01" to "99". The serial number is used to create and identify different file descriptor records for the same file descriptor class. This field is mandatory.

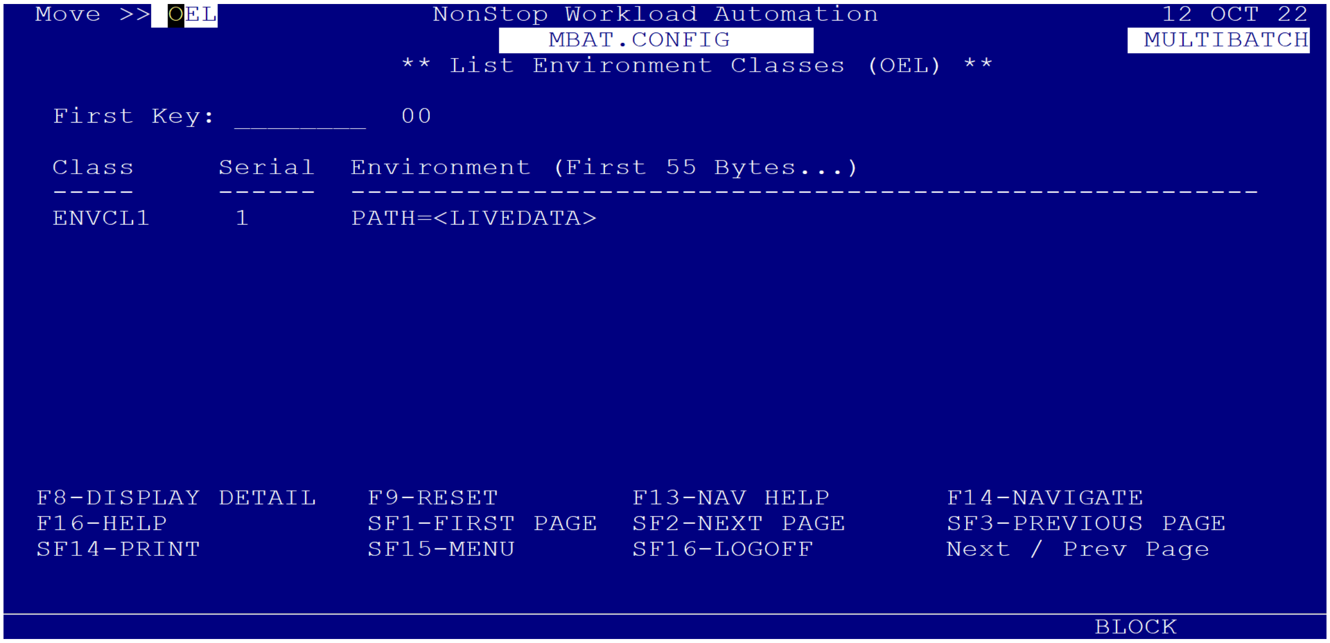


```
 Move >> COE               NonStop Workload Automation                 12 OCT 22
                               MBAT.CONFIG                           MULTIBATCH
                       ** Maintain Environment Class (COE) **


   Environment Class :          ENVCL1__

   Serial Key        :          01

   Environment  : PATH=<LIVEDATA>________________________________________
                  _______________________________________________________
                  _______________________________________________________
                  _______________________________________________________
                  ____________________________________

   Explain Shell: ________


 F1-LIST       F2-AMEND             F3-DELETE     F4-INSERT     F5-LIST NEXT
 F9-RESET      F13-NAVIGATE HELP    F14-NAVIGATE  F15-RETURN    F16-HELP
 SF4-CONFIRM   SF11-EXPLAIN         SF14-PRINT    SF15-MENU     SF16-LOGOFF

                                                                       BLOCK
```

Figure 3.38 Maintain Environment Class (COE)

3.19 List Environment Class (OEL)

The List Environment Class Screen (OEL) (Figure 3.39) is accessed by selecting SF3 - List OSS Env Classes, from the MultiBatch OSS Step Configuration Menu (COM). This provides a list of all the Environment classes defines in the system together with the first 55 characters of the Environment parameter.



```
 Move >> OEL               NonStop Workload Automation                 12 OCT 22
                               MBAT.CONFIG                           MULTIBATCH
                         ** List Environment Classes (OEL) **

  First Key: ________   00

  Class     Serial  Environment (First 55 Bytes...)
  -----     ------  -------------------------------------------------------
  ENVCL1     1      PATH=<LIVEDATA>









 F8-DISPLAY DETAIL   F9-RESET         F13-NAV HELP      F14-NAVIGATE
 F16-HELP            SF1-FIRST PAGE   SF2-NEXT PAGE     SF3-PREVIOUS PAGE
 SF14-PRINT          SF15-MENU        SF16-LOGOFF       Next / Prev Page


                                                                       BLOCK
```

Figure 3.39 – List Environment Class Screen (OEL)

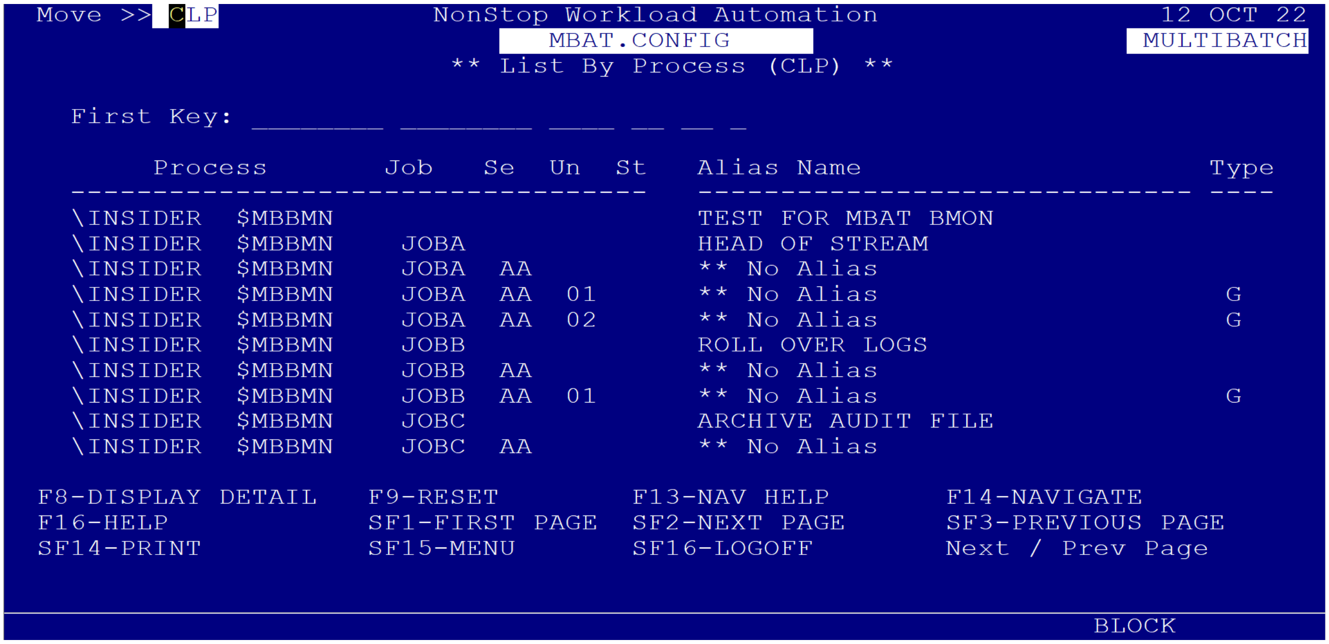
3.20 Configuration Overview

Once the BMON process is configured the user can get an overall view of the system in a variety of ways:

1. By accessing the List by Process Screen (CLP).
2. By accessing the List Entities Screen (CLE).
3. By accessing the List ALIASMAP / wild Card Screen (CLA).

3.21 List by Process (CLP)

The List by Process (CLP) Screen (Figure 3.40) is accessed via the **SF3** List by Process function key from the Entity Configuration (CMM) menu.



```
 Move >> CLP                   NonStop Workload Automation                  12 OCT 22
                                    MBAT.CONFIG                           MULTIBATCH
                                ** List By Process (CLP) **

   First Key: ________ ________ ____ __ __ _

        Process          Job    Se  Un  St   Alias Name                      Type
   -----------------------------------       ------------------------------- ----
   \INSIDER  $MBBMN                          TEST FOR MBAT BMON
   \INSIDER  $MBBMN    JOBA                  HEAD OF STREAM
   \INSIDER  $MBBMN    JOBA   AA             ** No Alias
   \INSIDER  $MBBMN    JOBA   AA  01         ** No Alias                       G
   \INSIDER  $MBBMN    JOBA   AA  02         ** No Alias                       G
   \INSIDER  $MBBMN    JOBB                  ROLL OVER LOGS
   \INSIDER  $MBBMN    JOBB   AA             ** No Alias
   \INSIDER  $MBBMN    JOBB   AA  01         ** No Alias                       G
   \INSIDER  $MBBMN    JOBC                  ARCHIVE AUDIT FILE
   \INSIDER  $MBBMN    JOBC   AA             ** No Alias

 F8-DISPLAY DETAIL   F9-RESET         F13-NAV HELP       F14-NAVIGATE
 F16-HELP            SF1-FIRST PAGE   SF2-NEXT PAGE      SF3-PREVIOUS PAGE
 SF14-PRINT          SF15-MENU        SF16-LOGOFF        Next / Prev Page


                                                                          BLOCK
```

Figure 3.40 – List by MultiBatch Process Screen (CLP)

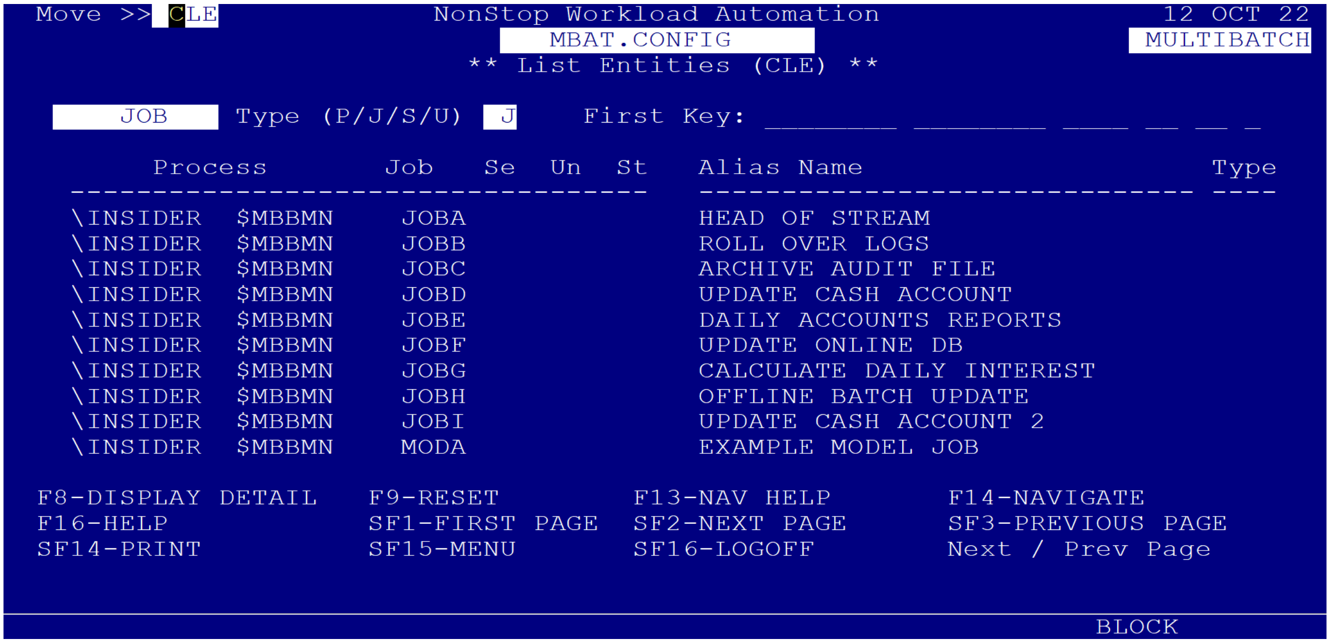
This is an informational screen which displays all the Jobs, Segments, Units and Steps in the database along with their Alias names.

The list can be started at a specific process Job, Segment or Unit by entering the relevant name in the First Key field and pressing the **SF1** function key.

To modify a record, enter any character in the first column next to the record to be changed and press the **F8** Display Detail function key. This takes the user to the relevant Configuration Screen (CBM, CJB, CSE, CST or COS) where changes can be made.

3.22 List Entities (CLE)

The List Entities (CLE) Screen (Figure 3.41) is accessed via the **SF2** List by Entity/BMON/J/S/U function key from the Entity Configuration (CMM) menu.



```
 Move >> CLE               NonStop Workload Automation                 12 OCT 22
                              MBAT.CONFIG                          MULTIBATCH
                          ** List Entities (CLE) **

   JOB      Type (P/J/S/U)  J    First Key: ________ ________ ____ __ __ _

        Process        Job    Se  Un  St   Alias Name                     Type
    -----------------------------------   ------------------------------ ----
    \INSIDER  $MBBMN    JOBA               HEAD OF STREAM
    \INSIDER  $MBBMN    JOBB               ROLL OVER LOGS
    \INSIDER  $MBBMN    JOBC               ARCHIVE AUDIT FILE
    \INSIDER  $MBBMN    JOBD               UPDATE CASH ACCOUNT
    \INSIDER  $MBBMN    JOBE               DAILY ACCOUNTS REPORTS
    \INSIDER  $MBBMN    JOBF               UPDATE ONLINE DB
    \INSIDER  $MBBMN    JOBG               CALCULATE DAILY INTEREST
    \INSIDER  $MBBMN    JOBH               OFFLINE BATCH UPDATE
    \INSIDER  $MBBMN    JOBI               UPDATE CASH ACCOUNT 2
    \INSIDER  $MBBMN    MODA               EXAMPLE MODEL JOB

 F8-DISPLAY DETAIL   F9-RESET        F13-NAV HELP      F14-NAVIGATE
 F16-HELP            SF1-FIRST PAGE  SF2-NEXT PAGE     SF3-PREVIOUS PAGE
 SF14-PRINT          SF15-MENU       SF16-LOGOFF       Next / Prev Page


                                                                  BLOCK
```

Figure 3.41 – List MultiBatch Entities Screen (CLE)

This is an informational screen which can be used to display selected records, by entering either ‘P’ (BMON Process), ‘J’ (Job), ‘S’ (Segment) or ‘U’ (Unit) in the TYPE field.

The list can be started at a specific BMON Process, Job, Segment, Step or Unit by entering the relevant name in the First Key field and pressing the **SF1** function key.

To modify a record, enter any character in the first column next to the record to be changed and press the **F8** Display Detail function key. This takes the user to the appropriate Configuration Screen as follows:

Type **P** - Maintain BMON Configuration Screen (CBM)

Type **J** - Maintain Job Configuration Screen (CJB).

Type **S** - Maintain Segment Configuration Screen (CSE).

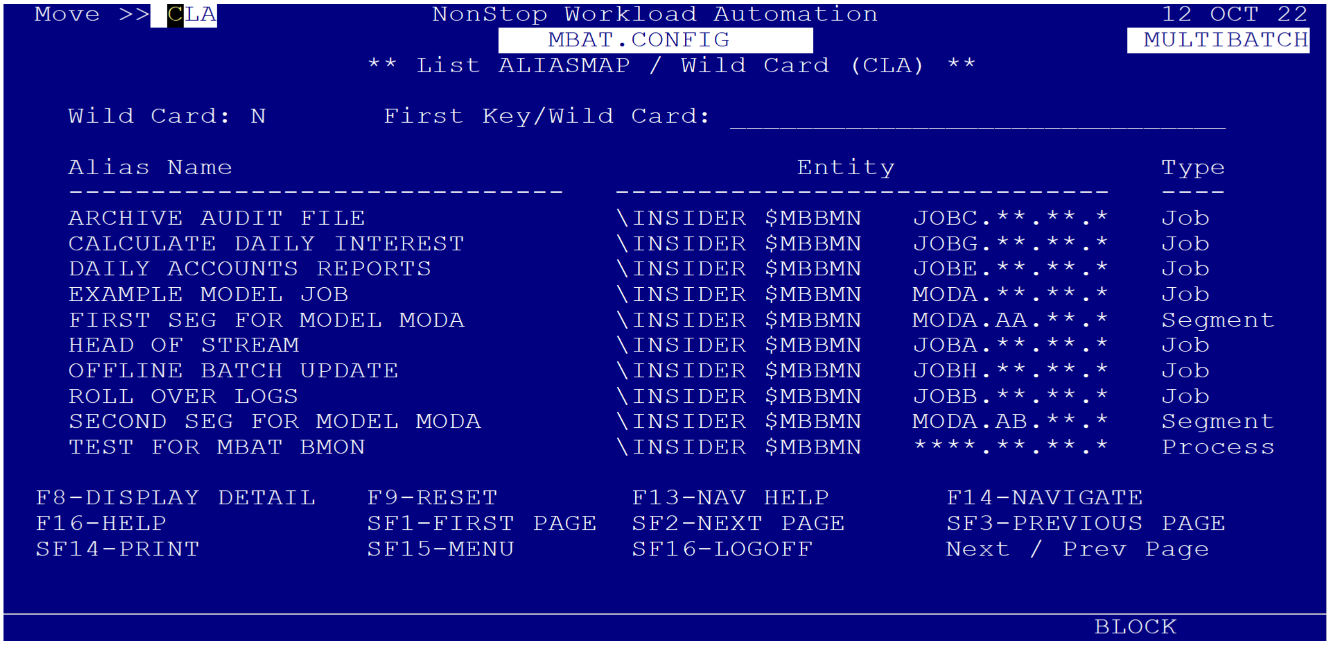
Type **U** - Maintain Step Configuration Screen (CST) or Maintain OSS Configuration Screen (COS).

3.23 List Alias (CLA)

A list of all the Alias Names on the MultiBatch system together with their corresponding MultiBatch entities and types, is given on the List All Alias Screen (CLA) (Figure 3.42). This is accessed via the S/F1 function key from the Entity Configuration Screen (CMM) or by using the Navigate facility.

Because the length of Job, Segment and Unit names is restricted (4+2+2 characters for Job, Segment and Unit respectively), this Alias facility is provided. This allows the user to create a link between a MultiBatch Job, Segment or Unit identity name and a longer, more meaningful name, e.g. JOBA could have an Alias of CLEAR SPOOLER. The user can create the alias at any of the configuration insert screens; CBM, CJB, CSE, CST and COS.

Note: *Step (G) is for a GUARDIAN process and Step (O) is for an OSS process.*



```
 Move >> CLA                 NonStop Workload Automation                12 OCT 22
                                  MBAT.CONFIG                          MULTIBATCH
                         ** List ALIASMAP / Wild Card (CLA) **

   Wild Card: N        First Key/Wild Card: ______________________________

   Alias Name                                   Entity                     Type
   ------------------------------   ------------------------------         ----
   ARCHIVE AUDIT FILE               \INSIDER $MBBMN   JOBC.**.**.*         Job
   CALCULATE DAILY INTEREST         \INSIDER $MBBMN   JOBG.**.**.*         Job
   DAILY ACCOUNTS REPORTS           \INSIDER $MBBMN   JOBE.**.**.*         Job
   EXAMPLE MODEL JOB                \INSIDER $MBBMN   MODA.**.**.*         Job
   FIRST SEG FOR MODEL MODA         \INSIDER $MBBMN   MODA.AA.**.*         Segment
   HEAD OF STREAM                   \INSIDER $MBBMN   JOBA.**.**.*         Job
   OFFLINE BATCH UPDATE             \INSIDER $MBBMN   JOBH.**.**.*         Job
   ROLL OVER LOGS                   \INSIDER $MBBMN   JOBB.**.**.*         Job
   SECOND SEG FOR MODEL MODA        \INSIDER $MBBMN   MODA.AB.**.*         Segment
   TEST FOR MBAT BMON               \INSIDER $MBBMN   ****.**.**.*         Process

 F8-DISPLAY DETAIL   F9-RESET          F13-NAV HELP      F14-NAVIGATE
 F16-HELP            SF1-FIRST PAGE    SF2-NEXT PAGE     SF3-PREVIOUS PAGE
 SF14-PRINT          SF15-MENU         SF16-LOGOFF       Next / Prev Page


                                                                    BLOCK
```

Figure 3.42 – List Alias Screen (CLA)

Note: *The Wildcard feature can be used to list selected Alias Names, e.g. if you only wanted to see the 'TABLE' alias names, enter Y in the Wild Card field and *TABLE* in First Key/Wild Card field.*

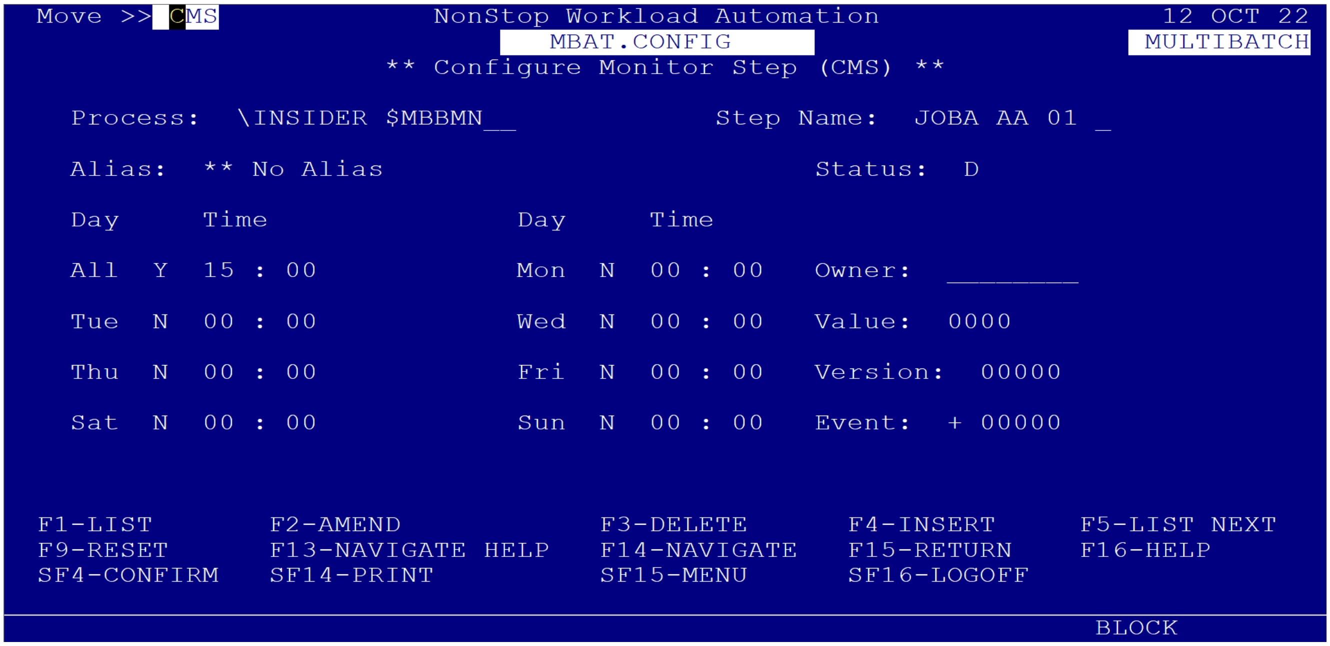
This provides a list of all Alias names containing the word 'TABLE'. You can also use "?" to match on single characters.

3.24 Configure Monitor Step (CMS)

The information entered on this screen is read and analyzed by the CLOCKMON process. See Chapter 5.14 'Start Time Monitoring (CLOCKMON)' on how to start the CLOCKMON process.

The CLOCKMON process operates in the background monitoring the status database file STEPSTAT.

At the appointed time, the appropriate STEPSTAT record is read and the status is checked. If the status is waiting, an EMS message is emitted reporting the non-start status. The CLOCKMON process must be running.



```
 Move >> CMS                 NonStop Workload Automation                12 OCT 22
                                   MBAT.CONFIG                        MULTIBATCH
                         ** Configure Monitor Step (CMS) **

   Process:  \INSIDER $MBBMN__           Step Name:  JOBA AA 01 _

   Alias:  ** No Alias                              Status:  D

   Day     Time                 Day     Time

   All  Y  15 : 00              Mon  N  00 : 00   Owner:  ________

   Tue  N  00 : 00              Wed  N  00 : 00   Value:  0000

   Thu  N  00 : 00              Fri  N  00 : 00   Version:  00000

   Sat  N  00 : 00              Sun  N  00 : 00   Event:  + 00000



 F1-LIST       F2-AMEND            F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET      F13-NAVIGATE HELP   F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM   SF14-PRINT          SF15-MENU      SF16-LOGOFF

                                                                BLOCK
```

Figure 3.43 Configure Monitor Step (CMS)

The following fields are available on this screen:

Process. This field represents the name of the BMON environment to be monitored. Valid input is NonStop node name and BMON process name.

Step Name. This field represents the name of the MultiBatch step whose start time is to be monitored. Valid input is, JOB, SEGMENT, UNIT and STEP (optional) identity.

Status. If the status is dormant then the monitor record is held on the STEPMON file but not processed by the CLOCKMON process. If the status is active, the monitor record is processed by CLOCKMON. Valid input is "A" (Active), or "D" (Dormant).

Day and Time. The Day flag is set to Y or N depending on whether monitoring of the start time of the step is required. The option "ALL" cannot be mixed with other days. If a Day flag is set to "Y", then a time must be supplied.

The time can be set between 00:01 (one minute past midnight) to 24:00 (midnight). 00:00 means that monitoring will not take place in which case the Day flag must be set to "N".

When a start time has been passed and the MultiBatch job has not been started, an EMS event will be emitted to signify this condition, based on the following fields:

Owner: This field is used to provide the Event Subsystem Owner.
Valid input is 1-8 alphabetic characters.

Value: This field is used to provide the Event Subsystem Value

Version: This field is used to provide the Event Subsystem Version.
Valid input is 0-32767.

Event: This field is used to provide the Event value.
Valid input is –32767 to +32767.

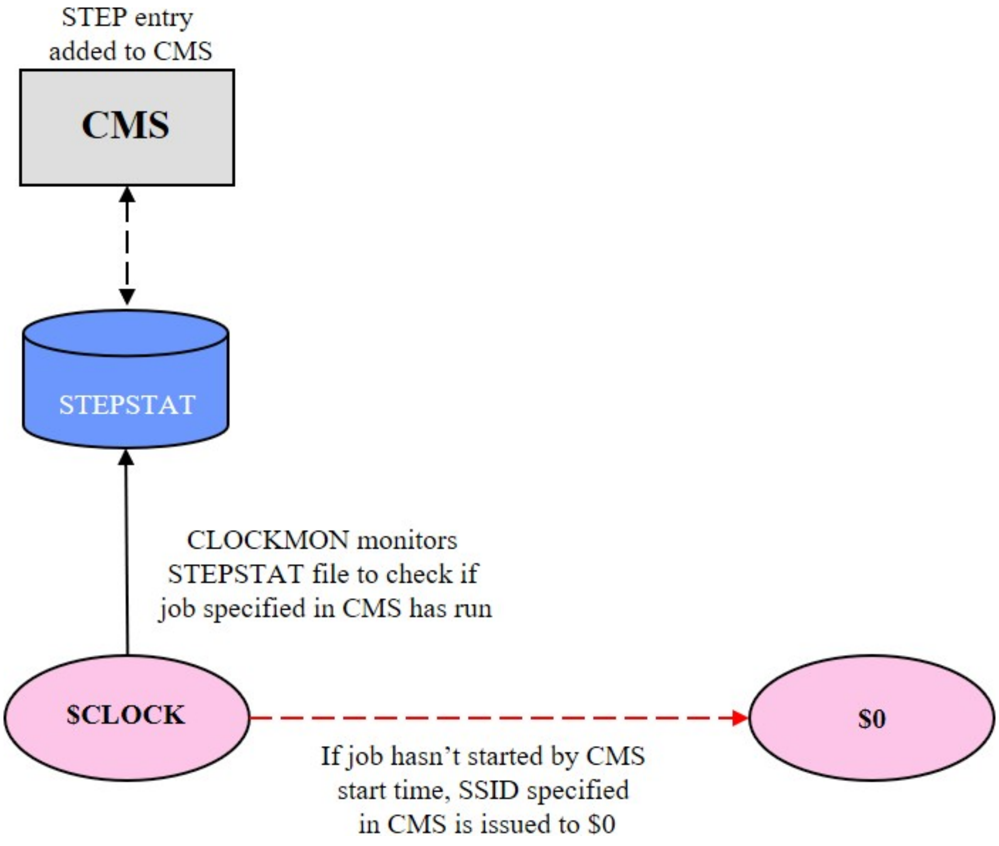

STEP entry
added to CMS
CMS
STEPSTAT
CLOCKMON monitors
STEPSTAT file to check if
job specified in CMS has run
$CLOCK
$0
If job hasn't started by CMS
start time, SSID specified
in CMS is issued to $0

Figure 3.44 CLOCKMON Architectural Diagram

3.25 Configure Monitor List (CML)

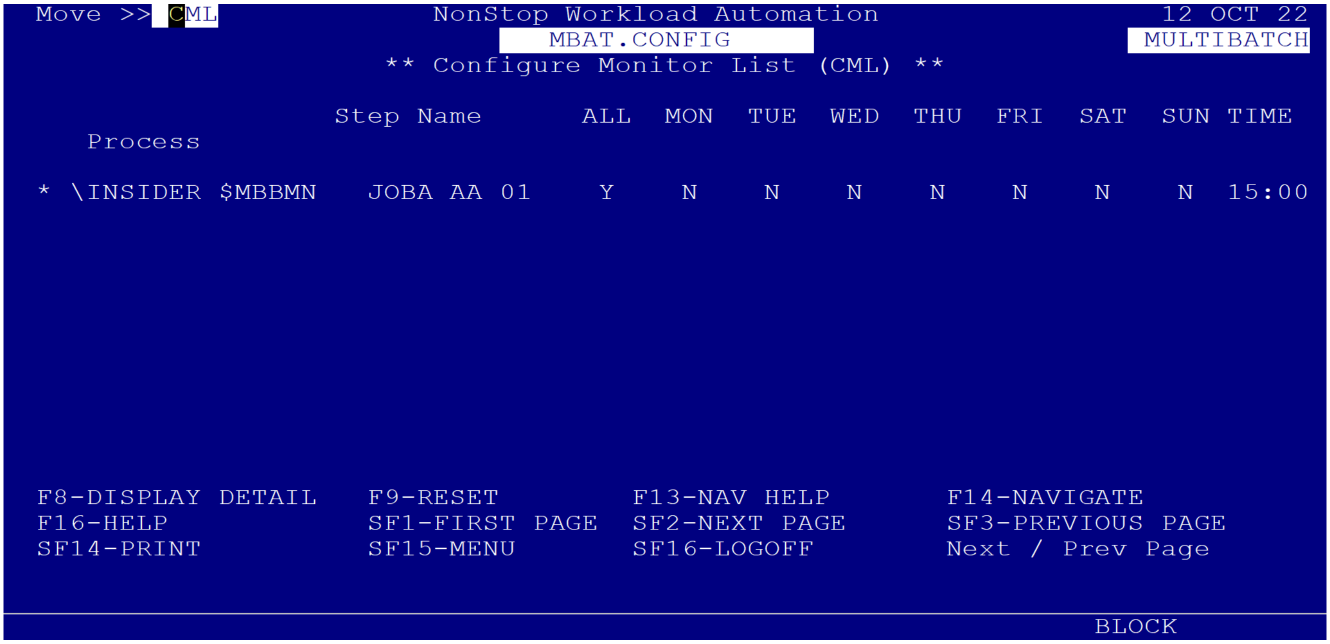
This is an informational screen displaying all the records in the STEPMON file.

The screen shows the names of the MultiBatch steps to be monitored together with the day selection criteria and the start time.

If more than one start time has been configured for a specific step, the TIME field will be set to "**Var.**" (short for Various).

To make modifications to a record, enter a character adjacent to the record to be changed and press the **F8**-DISPLAY DETAIL function key. This will navigate to the Maintain Monitor Step (CMS) screen so that amendments can be made to the STEPMON file.

Note: *Any Dormant CMS entries are identified by an asterisk to the left of the record.*



```
 Move >> CML             NonStop Workload Automation                12 OCT 22
                               MBAT.CONFIG                         MULTIBATCH
                    ** Configure Monitor List (CML) **

                          Step Name        ALL  MON  TUE  WED  THU  FRI  SAT  SUN TIME
    Process

 * \INSIDER $MBBMN    JOBA AA 01      Y    N    N    N    N    N    N    N  15:00











 F8-DISPLAY DETAIL    F9-RESET         F13-NAV HELP         F14-NAVIGATE
 F16-HELP             SF1-FIRST PAGE   SF2-NEXT PAGE        SF3-PREVIOUS PAGE
 SF14-PRINT           SF15-MENU        SF16-LOGOFF          Next / Prev Page

                                                                     BLOCK
```

Figure 3.45 Configure Monitor List (CML)

3.26 Configure Define Class (CDE)

Defines are described in section 3.13 Configure Defines (CDF).

Define Classes allow the grouping of defines by name so that Class Names can be associated with multiple STEPS.

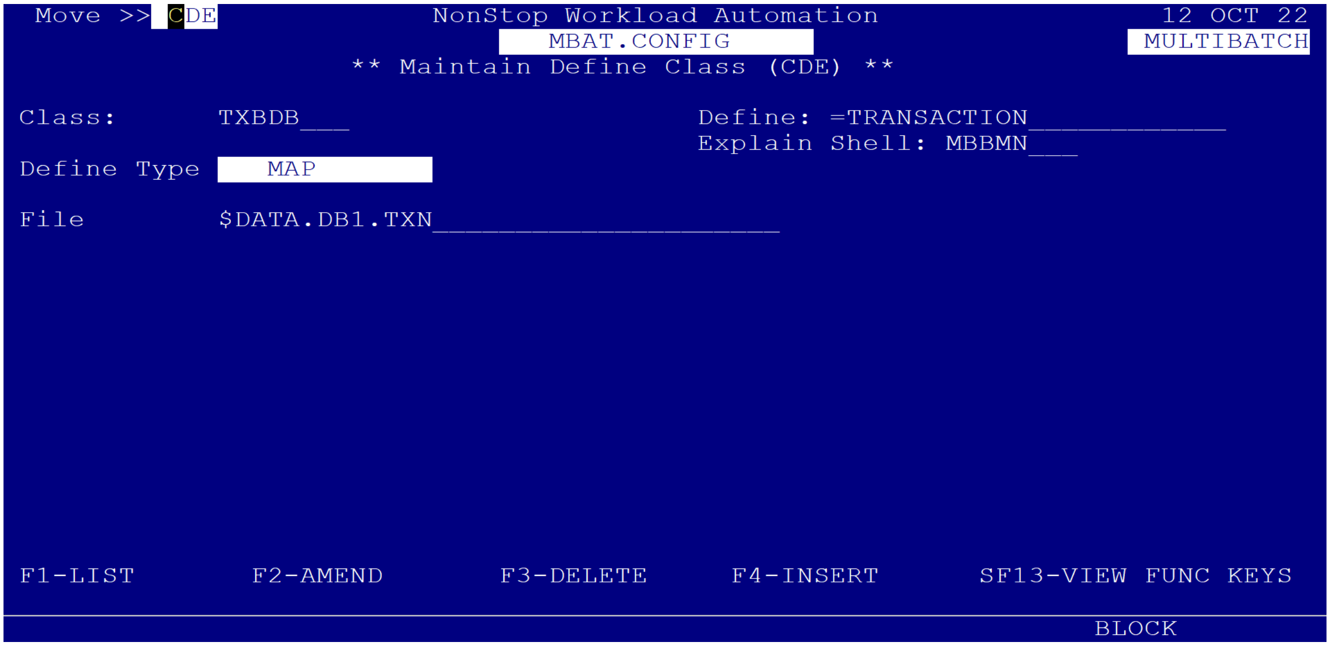
Defines Classes set up of the CDE screen are associated with steps on the Maintain STEP Configuration (CST) and the Maintain OSS STEP Configuration (COS) screens.

To CDE screen is available from the Guardian Step Configuration Menu (CGM) and the OSS Step Configuration Menu (COM) screens.

The Configure Define Class (CDE) screen (Figure 3.46) comprises of eight sub-screens or overlays with each overlay relating to a specific DEFINE type (Tape, TapeCatalog, Spool, Map, Catalog, Defaults, Sort, or Subsort).

Only the first overlay is show as the content of the overlays is common to that on Configure Defines (CDF) screen, see section 3.13.

The List Defines Classes (CDL) screen provides a list of all the Define Class entries in the MultiBatch system together with their corresponding Define types. It is possible to access the Configure Define Class (CDE) screen directly from the CDL screen.



```
 Move >> CDE             NonStop Workload Automation               12 OCT 22
                               MBAT.CONFIG                        MULTIBATCH
                    ** Maintain Define Class (CDE) **

Class:       TXBDB___                         Define: =TRANSACTION____________
                                              Explain Shell: MBBMN___
Define Type     MAP

File         $DATA.DB1.TXN_____________________









F1-LIST        F2-AMEND        F3-DELETE      F4-INSERT      SF13-VIEW FUNC KEYS

                                                                   BLOCK
```

Figure 3.46 – Maintain Define Class (CDE) (1st Overlay)

Chapter 4 Scheduling

4.1 Introduction

In the previous chapter we have generated Jobs, Segments and Units which we wish to execute. In this chapter we will see how to:

- **Set up** the scheduling information.
- **Prepare the Schedule**. The Prepare Phase takes "today's date" schedule, compares it with the Scheduling Database and produces a list of Jobs which match "today"; that is, those which have been selected to run.
- **Run the BMON program**. The BMON Build Phase takes the output from the Prepare Phase and uses this information when it is generating "todays schedule" that is to run.
- **Start the first job**. This initiates execution of "todays" schedule.

Each stage is described in detail and examples are provided where appropriate.

4.2 Setup Schedule Information

The scheduling of MultiBatch entities is controlled by using scheduling PARAMETERS. There are three scheduling Parameter types available:

1. Absolute.
2. Calendar.
3. Frequency.

4.3 Absolute

Absolute parameters are members of a closed set, comprising reserved words known to the MultiBatch system. These are:

- MONDAY
- TUESDAY
- WEDNESDAY
- THURSDAY
- FRIDAY
- SATURDAY
- SUNDAY
- WEEKDAY (Monday through Friday)
- WEEKEND (Saturday and Sunday)
- DAILY (Every day)
- NEVER (never selected).

As the frequency of these is known to the NonStop System, there is no necessity for corresponding Database entries to identify their occurrence.

MultiBatch can resolve these against any required date in the future when building a schedule. When configuring Jobs, Segments and Units (See Chapter 3) these are used within the Schedule Overlays along with **I** (Include) and **X** (eXclude).

For example, a Job may be configured with **DAILY I** and **THURSDAY X** which indicates that the job is not to be run on Thursdays.

4.4 Calendar

Calendar parameters can possess an arbitrary name, which is then related to a set of user-specified Dates contained within the MultiBatch database.

A description of how these dates are set up is given in the "Maintain Calendar" chapter.

4.5 Frequency

Frequency parameters resolve to one of two values, True or False, and are used whenever an ad-hoc type of schedule is required. The True/False value is not held within the database.

Note: *The MultiBatch system will always assume the Frequency type is False unless instructed otherwise when requested to Prepare a schedule.*

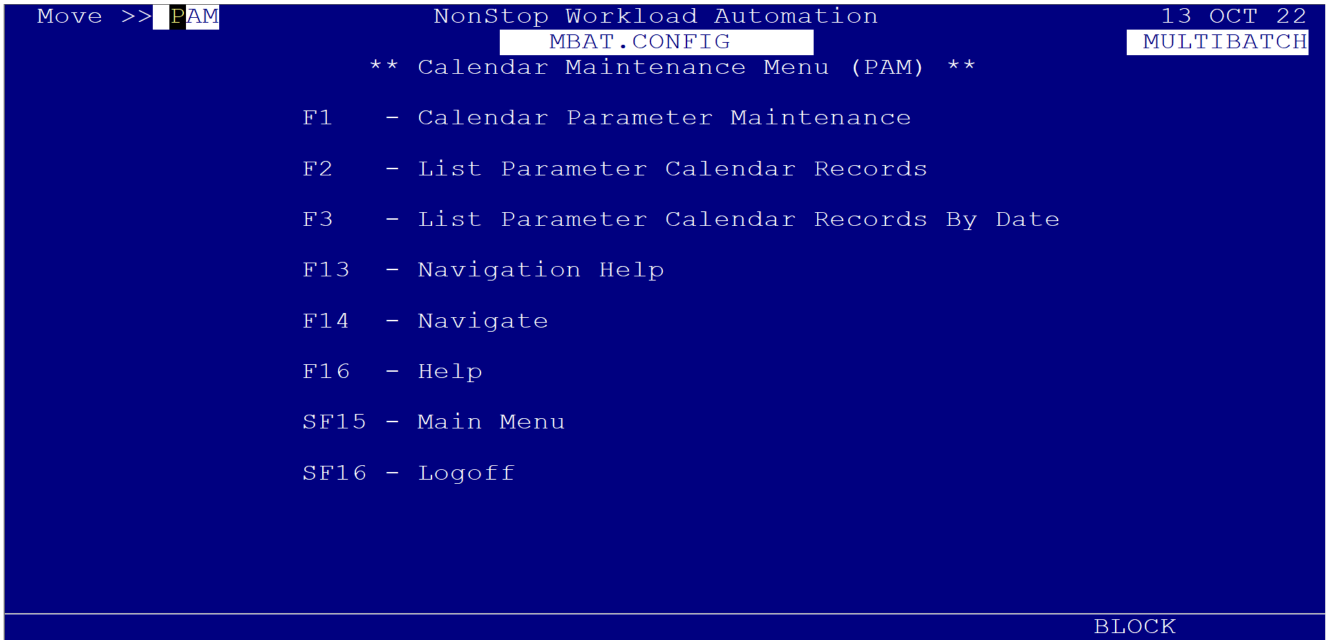
If the **True** value is not asserted by the user, any MultiBatch entity referencing a **Frequency** Parameter Type must also contain at least one **Calendar** or **Absolute** Parameter in order for its selection to succeed.

The presence of a **Frequency** type Parameter in a **Job**, **Segment** or **Unit** scheduling screen will be ignored during normal Schedule Preparation runs; it will not be considered when using the exclusion rules as for the other Parameter types.

4.6 Maintain Calendar (PAM)

If a Calendar parameter is to be used in the scheduling (e.g. Job configuration (CJB)) then it must be set up as a user specified date in the MultiBatch database before it is used, i.e. before the Configuration phase.

All calendar maintenance facilities are accessed via the MultiBatch Calendar Maintenance Menu (PAM) (Figure 4.1). This is accessed from the Main Menu (MEN) Screen by pressing the F2 function key or by using the Navigate facility.



```
  Move >> PAM                 NonStop Workload Automation                  13 OCT 22
                                   MBAT.CONFIG                          MULTIBATCH
                         ** Calendar Maintenance Menu (PAM) **

                     F1    - Calendar Parameter Maintenance

                     F2    - List Parameter Calendar Records

                     F3    - List Parameter Calendar Records By Date

                     F13   - Navigation Help

                     F14   - Navigate

                     F16   - Help

                     SF15  - Main Menu

                     SF16  - Logoff



                                                                          BLOCK
```

Figure 4.1 - MultiBatch Calendar Maintenance Menu (PAM)

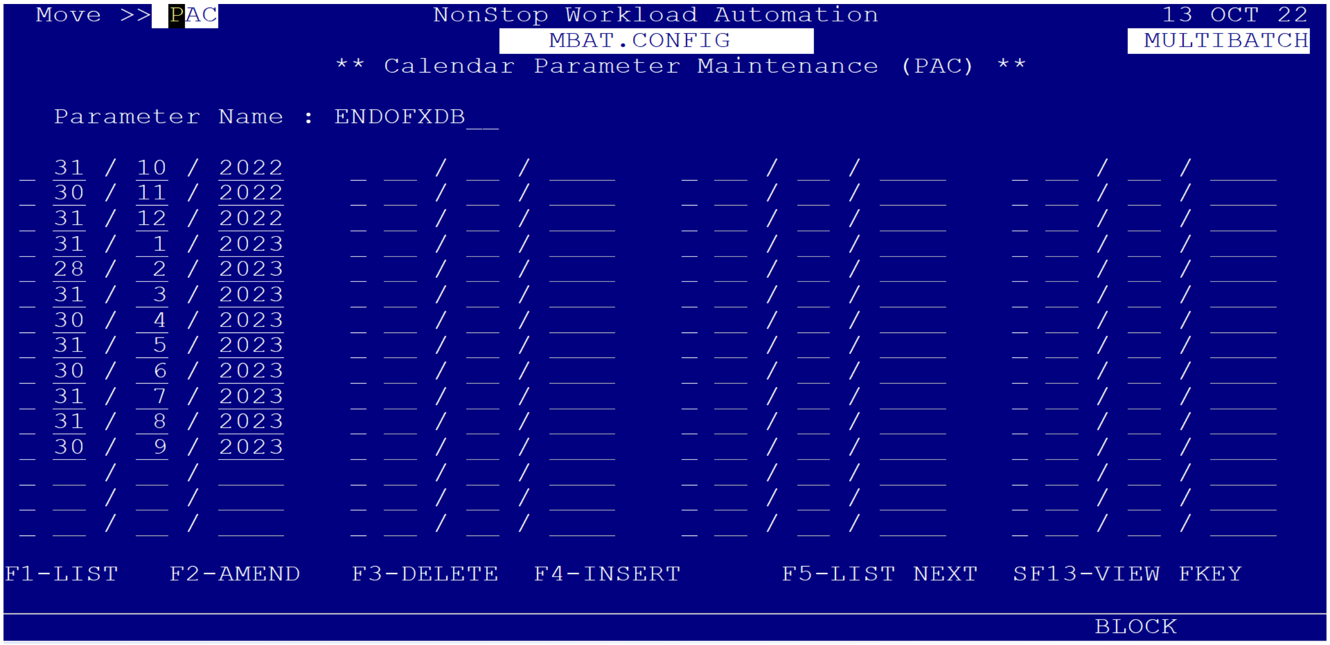
The Calendar Maintenance Menu provides the user with the following three options:

1. Maintain Parameter Calendar Records.
2. List Parameter Calendar Records.

3. List Parameter Calendar Records by Date.

4.7 Maintain Parameter Calendar Records (PAC)

Calendar maintenance is performed via the Maintain Parameter Calendar Screen (PAC) (Figure 4.2). This is accessed from the Calendar Maintenance Menu (PAM) and selecting F1, or by using the Navigate facility.



```
  Move >> PAC                  NonStop Workload Automation                 13 OCT 22
                                      MBAT.CONFIG                          MULTIBATCH
                         ** Calendar Parameter Maintenance (PAC) **

   Parameter Name : ENDOFXDB__

 _ 31 / 10 / 2022    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 30 / 11 / 2022    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 31 / 12 / 2022    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 31 /  1 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 28 /  2 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 31 /  3 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 30 /  4 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 31 /  5 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 30 /  6 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 31 /  7 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 31 /  8 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ 30 /  9 / 2023    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____
 _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____    _ __ / __ / ____

F1-LIST   F2-AMEND   F3-DELETE  F4-INSERT     F5-LIST NEXT  SF13-VIEW FKEY

                                                                   BLOCK
```

Figure 4.2 – Maintain Calendar Parameter Screen (PAC)

This screen allows the user to enter specified dates for a job to be run and to link these dates to an appropriate Calendar Parameter Name.

In the example displayed in Figure 4.2, the Calendar Parameter Name ENDOFXDB has been chosen and appropriate dates entered in the relevant columns. This feature is used to generate Parameter Names for dates which are used on a regular basis (e.g. transaction reports being run on every Thursday of every month) and then applying the Calendar Name to particular Jobs, Segments, or Units.

Entries can be made in blocks and in any order and all entries are validated before the database is updated. If any date input is invalid, then the invalid date is highlighted and the user informed of the error.

Note: *A marker must be placed in the first column of all dates to be added/deleted. The date format used is "<dd>/<mm>/<yyyy>", for example, 11/06/2022.*

The Calendar Parameters generated can now be used in scheduling when configuring the Jobs, etc. The parameters must be created before the schedule configuration phase.

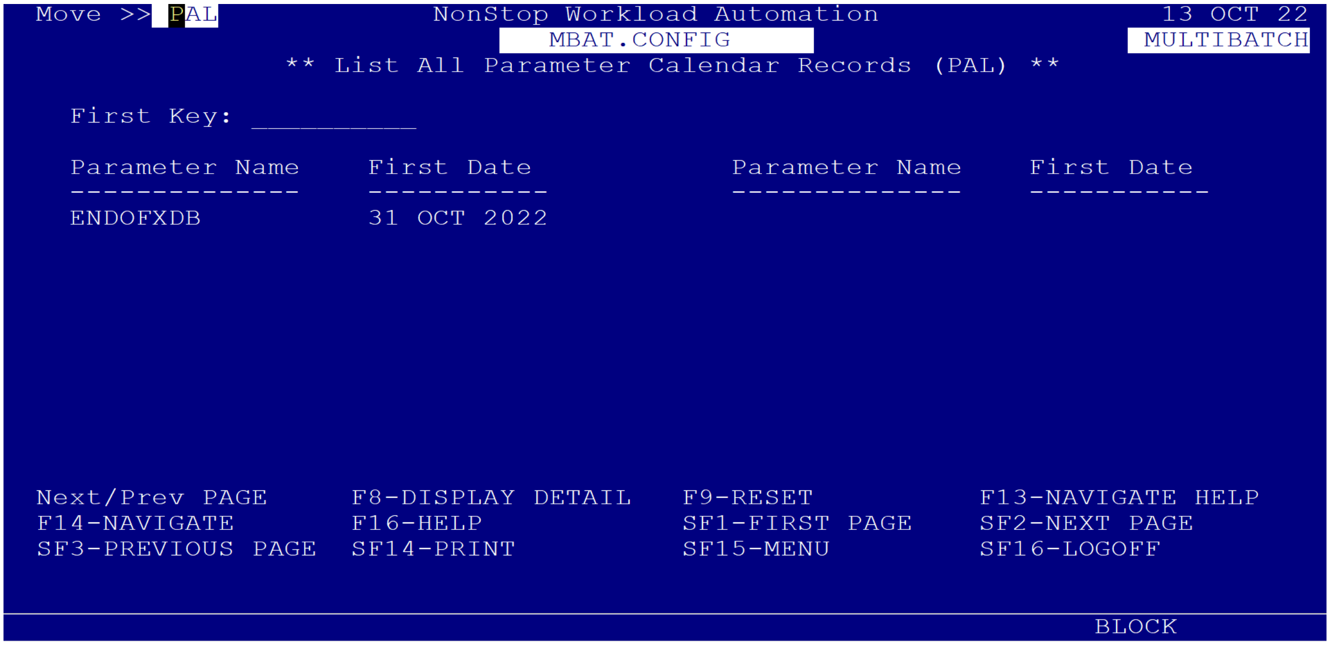
For delete and insert, place any character in the adjacent marker column and press **F3** or **F4** as appropriate. You can delete and insert at the same time, by using the amend function (**F2**) but the marker characters must be "I" for insert and "D" for delete.

4.8 List All Parameter Calendar Records (PAL)

Access the List All Parameter Calendar Records Screen (PAL) (Figure 4.3) via the function keys or Navigate facility to get an overall view of all the Parameter calendar Records defined in the MultiBatch system. This is an information screen which displays all the Parameter Names, together with their corresponding dates, in alphabetical order.

Note: *Only the first date for each Calendar Parameter is shown.*

The user may start the listing from any Parameter by entering the Parameter Name in the First Key field and pressing the SF1 function key. Only Parameters from the specified Parameter Name onwards are displayed. In addition, the user can amend a given Parameter by placing a marker beside the Parameter Name and pressing the F8-Detail function key. This takes the user to the Maintain Calendar Parameter Screen (PAC) where the entries for that Parameter Name can be updated.

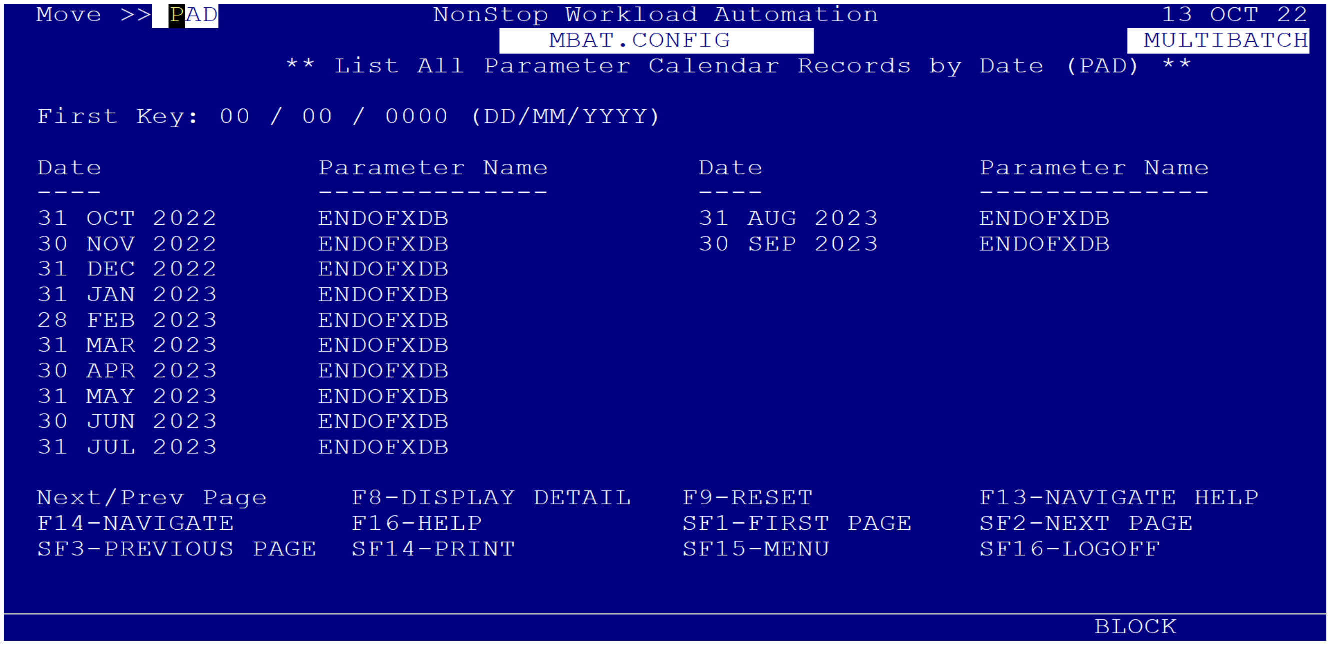


```
 Move >> PAL               NonStop Workload Automation                13 OCT 22
                                  MBAT.CONFIG                          MULTIBATCH
                 ** List All Parameter Calendar Records (PAL) **

   First Key: __________

   Parameter Name     First Date              Parameter Name     First Date
   --------------     -----------             --------------     -----------
   ENDOFXDB           31 OCT 2022










 Next/Prev PAGE      F8-DISPLAY DETAIL   F9-RESET           F13-NAVIGATE HELP
 F14-NAVIGATE        F16-HELP            SF1-FIRST PAGE     SF2-NEXT PAGE
 SF3-PREVIOUS PAGE   SF14-PRINT          SF15-MENU          SF16-LOGOFF


                                                                  BLOCK
```

Figure 4.3 – List All Parameter Calendar Records Screen (PAL)

The List All Parameter Calendar Records by Date Screen (PAD) (Figure 4.4) is accessed via the function keys or Navigate facility. As with the previous feature, this provides an overall view of all the Parameter Calendar Records defined in the MultiBatch system; the records are displayed in date order.

The user may start the listing from any specified date by entering the date in the First Key field and pressing the SF1 function key. Only Parameters from the specified date onwards are displayed. In addition, the user can amend a given Parameter by placing a marker beside the Parameter Name and pressing the F8-Detail function key. This takes the user to the Maintain Calendar Parameter Screen (PAC) where the entries for that Parameter Name can be updated.



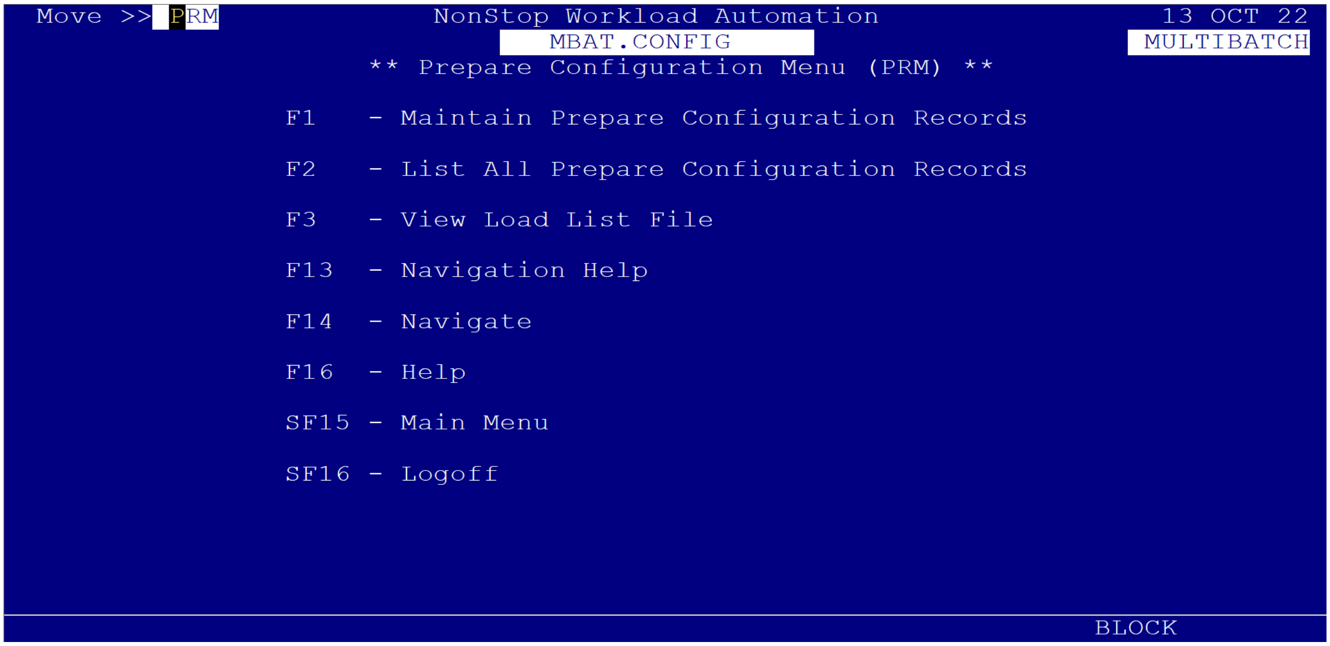
```
 Move >> PAD              NonStop Workload Automation                13 OCT 22
                                MBAT.CONFIG                         MULTIBATCH
               ** List All Parameter Calendar Records by Date (PAD) **

 First Key: 00 / 00 / 0000 (DD/MM/YYYY)

 Date                 Parameter Name          Date                 Parameter Name
 ----                 --------------          ----                 --------------
 31 OCT 2022          ENDOFXDB                31 AUG 2023          ENDOFXDB
 30 NOV 2022          ENDOFXDB                30 SEP 2023          ENDOFXDB
 31 DEC 2022          ENDOFXDB
 31 JAN 2023          ENDOFXDB
 28 FEB 2023          ENDOFXDB
 31 MAR 2023          ENDOFXDB
 30 APR 2023          ENDOFXDB
 31 MAY 2023          ENDOFXDB
 30 JUN 2023          ENDOFXDB
 31 JUL 2023          ENDOFXDB

 Next/Prev Page      F8-DISPLAY DETAIL   F9-RESET             F13-NAVIGATE HELP
 F14-NAVIGATE        F16-HELP            SF1-FIRST PAGE       SF2-NEXT PAGE
 SF3-PREVIOUS PAGE   SF14-PRINT          SF15-MENU            SF16-LOGOFF

                                                                     BLOCK
```

Figure 4.4 List All Parameter Calendar Records by Date Screen (PAD)

4.9 Prepare Phase (PRM)

Once the Parameter Calendar has been set up and the various Jobs, Segments etc., are configured and scheduled, the PREPARE phase can be performed. This is accessed via the Navigate facility or the F6 function key from the Main Menu (Figure 3.1). This displays the Prepare Configuration Menu Screen (PRM) (Figure 4.5). From this screen the user is presented with the following options:

- Maintain Prepare Configuration Records.
- List All Prepare Configuration Records.
- View Load List File.

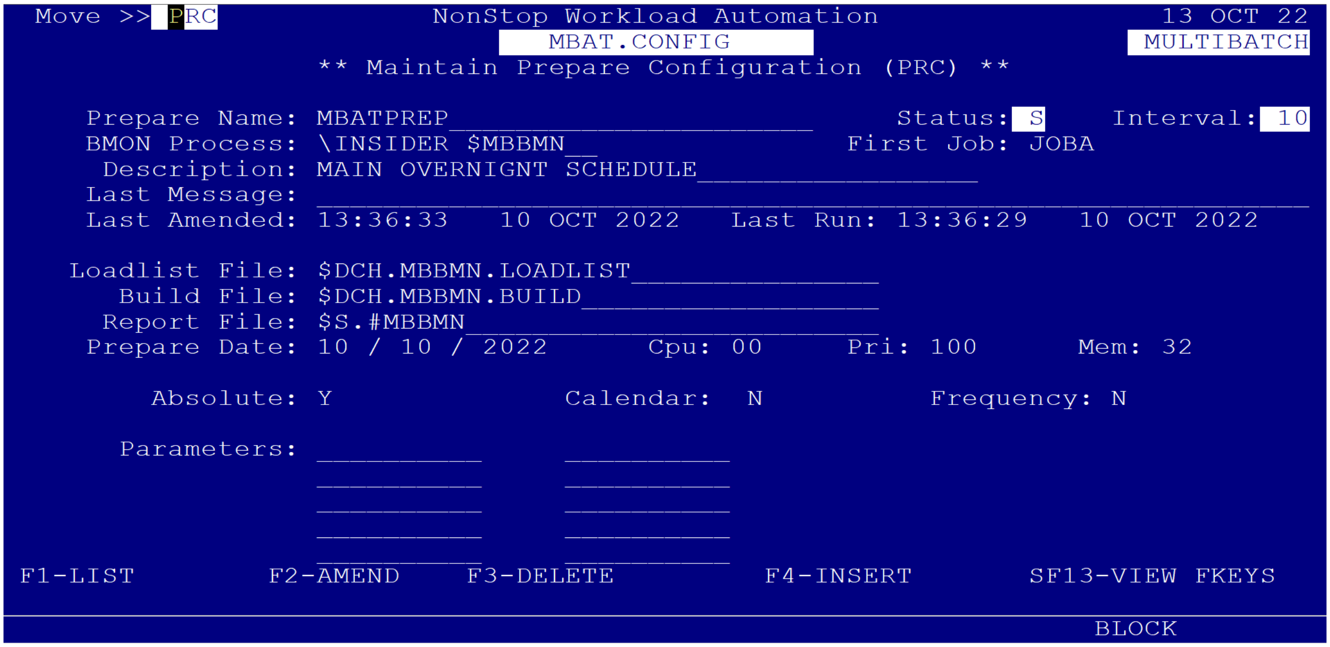


```
  Move >> PRM              NonStop Workload Automation                 13 OCT 22
                                 MBAT.CONFIG                         MULTIBATCH
                        ** Prepare Configuration Menu (PRM) **

                   F1   - Maintain Prepare Configuration Records

                   F2   - List All Prepare Configuration Records

                   F3   - View Load List File

                   F13  - Navigation Help

                   F14  - Navigate

                   F16  - Help

                   SF15 - Main Menu

                   SF16 - Logoff



                                                                          BLOCK
```

Figure 4.5 – Prepare Configuration Menu Screen (PRM)

4.10 Maintain Prepare Configuration Records (PRC)

The Maintain Prepare Configuration Screen (PRC) (Figure 4.6) is accessed from the Prepare Configuration Menu screen via the F1 function key or by using the Navigate facility.



```
  Move >> PRC              NonStop Workload Automation                 13 OCT 22
                                 MBAT.CONFIG                         MULTIBATCH
                     ** Maintain Prepare Configuration (PRC) **

     Prepare Name: MBATPREP______________________     Status: S    Interval: 10
     BMON Process: \INSIDER $MBBMN__               First Job: JOBA
      Description: MAIN OVERNIGNT SCHEDULE_________________
     Last Message: ____________________________________________________________
     Last Amended: 13:36:33   10 OCT 2022   Last Run: 13:36:29   10 OCT 2022

    Loadlist File: $DCH.MBBMN.LOADLIST_______________
       Build File: $DCH.MBBMN.BUILD__________________
      Report File: $S.#MBBMN_________________________
     Prepare Date: 10 / 10 / 2022      Cpu: 00      Pri: 100      Mem: 32

         Absolute: Y              Calendar:  N          Frequency: N

       Parameters: __________     __________
                   __________     __________
                   __________     __________
                   __________     __________
 F1-LIST        F2-AMEND    F3-DELETE        F4-INSERT       SF13-VIEW FKEYS

                                                                          BLOCK
```

Figure 4.6 – Maintain Prepare Configuration Screen (PRC)

Prepare Name. This is the name allocated to the Run Schedule by the user. If the Schedule already exists, then the user can input the Prepare Name and press the **F1**-List function key to display the details for the specified name. This field is mandatory.

BMON Process. This is the name of the MultiBatch BMON process that is to run the schedule (the same BMON Name as specified in the CBM screen, via the Configure phase). This field is mandatory.

Status. This field indicates the status of the PREPARE program. Possible values for this field are SCHEDULED (S), INITIATED (I), RUNNING (R) or FAILED (X).

The Prepare can only be executed from the SCHEDULED state. If for some reason the Prepare fails the user can correct the fault that caused the failure and change the "X" status back to "S", F2 to amend and then run the Prepare again.

Interval. This field allows you to enter an "interval" in seconds in which the screen will be automatically refreshed until it completes. The minimum and default value is 10 seconds. If you navigate to this screen and the PREPARE is already running, you can "switch on" the monitoring using the **F1** function key.

First Job. This is the name of the first Job to be executed when running the PREPARE. This Job must always be configured in order to provide a logical start point for the PREPARE process to build a schedule configuration. This field is mandatory.

Description. This is an optional field used to expand the Prepare Name to a more "meaningful" name, e.g. Cold load.

Last Message. This is a protected field which displays the last message generated by the Prepare program. If the execute fails, this message provides an indication as to where the fault occurred. An EMS event will also be issued.

Last Amended. This is also a protected field which shows when the Schedule was last amended and when it was last run (if ever).

Loadlist File. This is the file to which the schedule information is written when the PREPARE is executed. The Load List File can be viewed by pressing the **SF7** LOADLIST function key which displays the View Loadlist File Records Screen (VLL).

Build File. This file will contain the actual MultiBatch BCOM commands used to build the Prepare Configuration Schedule. If required, this filename and the Loadlist File name can be omitted, providing the report file is provided; typically, a user would only do this to generate a report to check out a schedule and not build it, e.g. "What is my schedule for the next Bank Holiday?" If omitted, then the first page of the Prepare Report will contain:

```
Loadlist File          : *Report Only
Build File             : *Report Only
```

Report File. This field specifies the name to be given to the report generated by the PREPARE program. If an existing filename is used, then it should be an entry-sequence file of 80 character record length. If the specified does not exist, then an entry-sequenced file will be created (with a record length of 80 bytes). This field is not mandatory and no report is generated if it is omitted.

Prepare Date. This field specifies the date for which the schedule is being prepared. The field is mandatory if the Calendar field is set to Y. This is very useful if you wish to check that on a specified date in the future, the correct jobs, segments and units will be selected for that date. Prepare can generate a report once it has completed, where you can view the results within the VLL screen (assuming a Loadlist file is being used), via key SF7-LOADLIST.

CPU. This field specifies the CPU in which the PREPARE process is to run.

Pri. This field is used to specify the Priority of the PREPARE (1 {low} to 199 {high}). This is a mandatory

Mem. This field is used to specify the amount of Memory (enter a working space value of 01 to 64) to be allocated to the Prepare program. This is a mandatory field.

Absolute. This field indicates (Y or N) whether or not the user wishes the Prepare program to determine if any Absolute parameters are true for the specified Prepare Date. If this field is set to Y, then MultiBatch entities which have Absolute parameters as part of their scheduling criteria are selected as eligible for processing subject to the Include/ Exclude setting of the individual entity configurations.

Calendar. This field (Y or N) indicates whether or not the user wishes the Prepare program to determine if any Calendar Parameter entries contain dates which match the Prepare Date. If the field is set to Y, then MultiBatch entities which match Calendar Parameter entries as part of their scheduling criteria are selected for processing.

Frequency. This field (Y, R or N) indicates whether or not the user wishes the Prepare program to assert the value of any Frequency parameter to be TRUE. If the field is set to Y or R, then MultiBatch entities which have those Frequency Parameter names as part of their scheduling criteria are selected. If the field is set to Y, then the Frequency parameters are deleted from the Prepare database once the Prepare has completed. If the field is set to R, then the Frequency parameters are retained for subsequent runs of the Prepare.

Parameters. If the Frequency field is set to Y or R then this field must contain the names of the relevant Frequency parameters.

The permissible combinations of the Absolute, Calendar and Frequency options are as follows:

Absolute = Y.

If Absolute is set to "Y", then a Prepare date must be supplied. In addition, "Absolute" parameters can be optionally set in the parameter fields. For example, if the Prepare date is 07/06/2021 and Absolute = "Y", then DAILY, WEEKDAY and MONDAY will be true. A user can optionally enter MONDAY in the parameter field to force this to be true.

Calendar = Y.

If Calendar is set to "Y", then a Prepare date must be supplied.

Frequency = Y or R.

If Frequency is set to "Y" or "R", then a Prepare date is not required, just the Frequency parameter values in the parameter field. If set to Y, then the field value will be removed when the Prepare has successfully completed. If set to R, then the field value is retained for subsequent runs of the Prepare.

A modified example of our schedule could be:

> Job1 will only run on the first Wednesday of every month (Calendar Parameter). Job2 will NOT run on a WEEKDAY (Absolute Parameter).
>
> Job3 and Job7 will only run on a WEEKEND (Absolute Parameter).
>
> Job4 will only run if the Frequency Parameter ADHOC01 is forced to TRUE. The remainder of the schedule will execute DAILY.

We are now ready to execute the **PREPARE**. This can be done in one of two ways:

Note that **PREPARE** can be run from the command line, the parameters as specified above are encoded in the start-up string as follows:

```
RUN PREPARE / IN <dataconf file> / <Prepare Name>, {TODAY or DATE <dd/mm/yyyy> or NEXT-
day or TOMORROW TODAY+nn,…{FREQ PARAM1 & PARAM2 &…..& PARAM10}
```

The first parameter is the Prepare Name as defines on the Maintain Prepare Configuration Screen (PRC) (Figure 4.6) screen.

The second parameter defines the schedule run date used along with the entity scheduling configuration data to define what is to be include in the schedule.

TODAY is today's date.

DATE <dd/mm/yyyy> provides a specific date

NEXT-day can be one of:

NEXT-MON NEXT-TUES NEXT-WED NEXT-THUR NEXT-FRI NEXT-SAT NEXT-SUN

TOMORROW TODAY+nn effectively defines when tomorrow is.

+nn is the number of days from today:

```
PREPARE / IN <dataconf file / <Prepare Name>, TOMORROW TODAY+2
```

FREQ are up to a maximum of 10 frequency parameters that are used during **PREPARE**. They are NOT held on the prepare database.

When running PREPARE from the Pathway interface the Maintain Prepare Configuration Screen (PRC) is used. This is explained in detail in section 4.11 'Prepare Run Screen'.

Note: *An OUT file is optional, as the results will be sent to the Prepare location as configured in the PRC Report File location.*

4.11 Prepare Run Screen

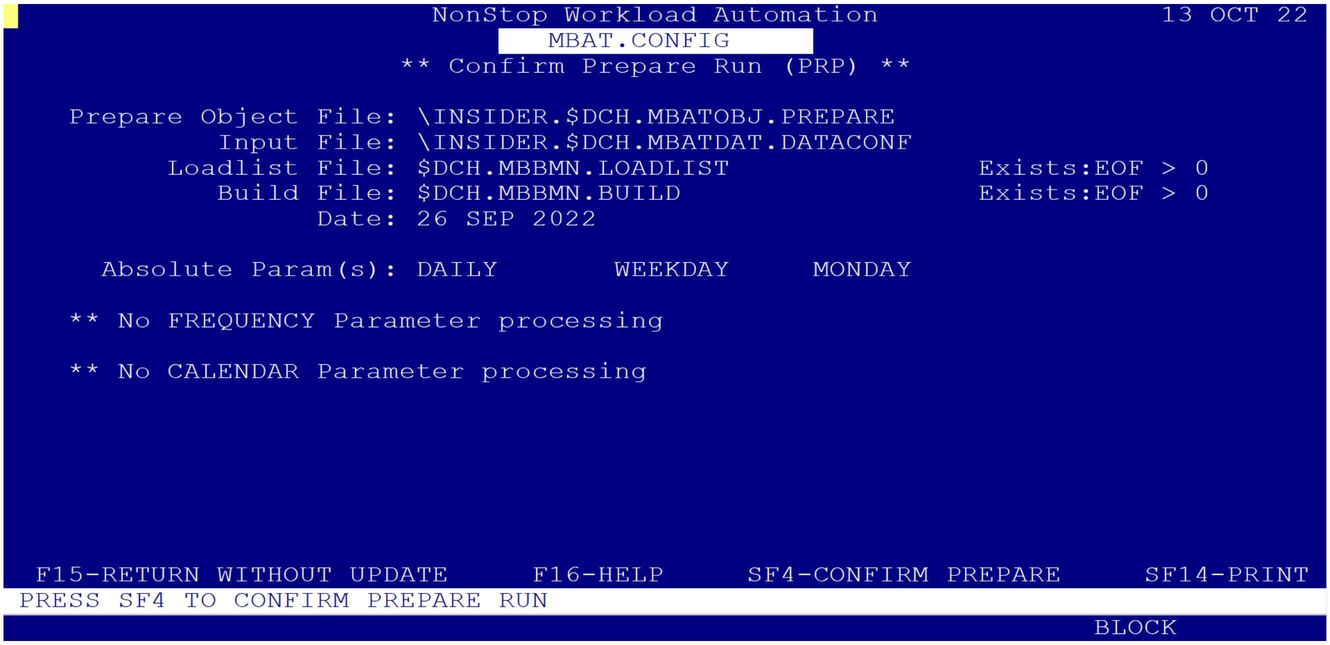
Navigate to the Maintain Prepare Configuration Screen (PRC) and press the SF6-RUN PREPARE function key. This displays the Confirm Prepare Run Screen (PRP) (Figure 4.7).

This is an informational screen which displays all the parameters which will be used when preparing the schedule. At this point the user can:

Press the **SF4**-CONFIRM PREPARE function key to execute the PREPARE if happy with the details displayed.

Press the **F15**-RETURN WITHOUT UPDATE function key to return to the Maintain Prepare Configuration (PRC) screen without making any changes. User can then make any necessary changes before executing the PREPARE.

When the **SF4** function key is pressed there is a short delay and the message "PREPARE REQUEST ACCEPTED" is displayed at the bottom of the screen. The user is then returned to the Maintain Prepare Configuration (PRC) Screen.



```
                         NonStop Workload Automation                     13 OCT 22
                              MBAT.CONFIG
                        ** Confirm Prepare Run (PRP) **

   Prepare Object File: \INSIDER.$DCH.MBATOBJ.PREPARE
            Input File: \INSIDER.$DCH.MBATDAT.DATACONF
         Loadlist File: $DCH.MBBMN.LOADLIST               Exists:EOF > 0
            Build File: $DCH.MBBMN.BUILD                  Exists:EOF > 0
                  Date: 26 SEP 2022

     Absolute Param(s): DAILY       WEEKDAY     MONDAY

   ** No FREQUENCY Parameter processing

   ** No CALENDAR Parameter processing


 F15-RETURN WITHOUT UPDATE     F16-HELP     SF4-CONFIRM PREPARE     SF14-PRINT
PRESS SF4 TO CONFIRM PREPARE RUN
                                                                     BLOCK
```

Figure 4.7 – Confirm Prepare Run Screen (PRP)

4.12 View Load List (VLL)

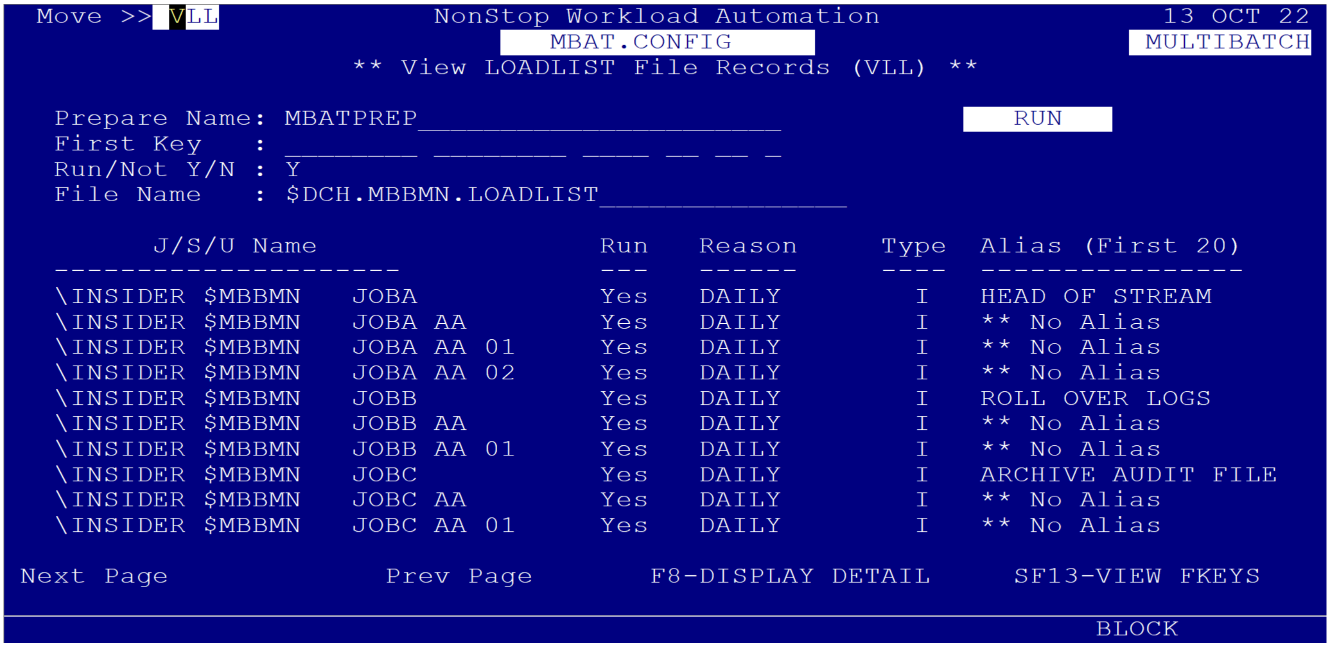
Press the **SF7**-LOADLIST function key to get an overall view of the current BMON environment.

This displays the View Loadlist File Screen (VLL); the following steps displays a list of Jobs which will run:

Enter the PREPARE Name (e.g. REPORT^SUITE) in the Prepare Name field, if not already made available via the SF7-LOADLIST function key.

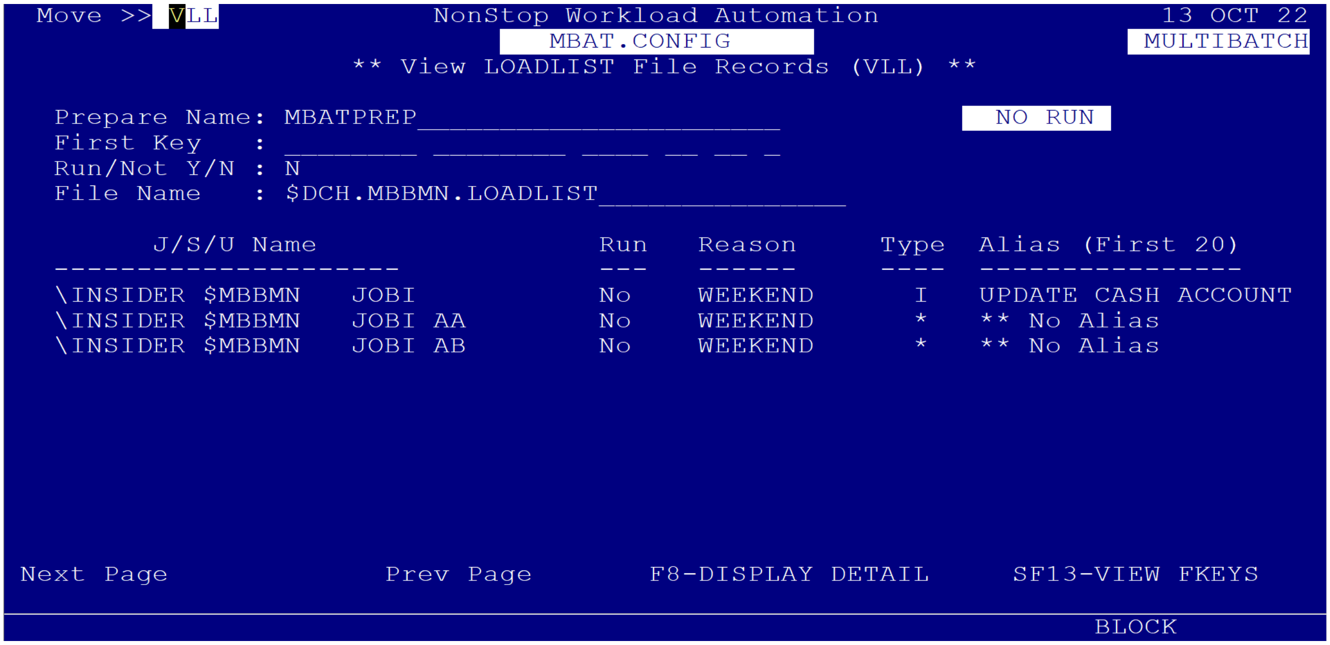
Enter Y in the RUN/NOT field. Press the SF1-FIRST PAGE function key to display all the Jobs that will run (Figure 4.8).

Repeat this process with N in the RUN/NOT field to display a list of Jobs which will not run (Figure 4.9).



```
  Move >> VLL             NonStop Workload Automation               13 OCT 22
                             MBAT.CONFIG                           MULTIBATCH
                   ** View LOADLIST File Records (VLL) **

   Prepare Name: MBATPREP______________________                RUN
   First Key   : ________ ________ _____ __ __ _
   Run/Not Y/N : Y
   File Name   : $DCH.MBBMN.LOADLIST_______________

         J/S/U Name                Run   Reason     Type  Alias (First 20)
   ---------------------           ---   ------     ----  ----------------
   \INSIDER $MBBMN   JOBA          Yes   DAILY        I   HEAD OF STREAM
   \INSIDER $MBBMN   JOBA AA       Yes   DAILY        I   ** No Alias
   \INSIDER $MBBMN   JOBA AA 01    Yes   DAILY        I   ** No Alias
   \INSIDER $MBBMN   JOBA AA 02    Yes   DAILY        I   ** No Alias
   \INSIDER $MBBMN   JOBB          Yes   DAILY        I   ROLL OVER LOGS
   \INSIDER $MBBMN   JOBB AA       Yes   DAILY        I   ** No Alias
   \INSIDER $MBBMN   JOBB AA 01    Yes   DAILY        I   ** No Alias
   \INSIDER $MBBMN   JOBC          Yes   DAILY        I   ARCHIVE AUDIT FILE
   \INSIDER $MBBMN   JOBC AA       Yes   DAILY        I   ** No Alias
   \INSIDER $MBBMN   JOBC AA 01    Yes   DAILY        I   ** No Alias

 Next Page             Prev Page        F8-DISPLAY DETAIL      SF13-VIEW FKEYS

                                                                   BLOCK
```

Figure 4.8 – View Load List File Records Screen (VLL) – Jobs to Run The



```
  Move >> VLL             NonStop Workload Automation               13 OCT 22
                             MBAT.CONFIG                           MULTIBATCH
                   ** View LOADLIST File Records (VLL) **

   Prepare Name: MBATPREP______________________               NO RUN
   First Key   : ________ ________ _____ __ __ _
   Run/Not Y/N : N
   File Name   : $DCH.MBBMN.LOADLIST_______________

         J/S/U Name                Run   Reason     Type  Alias (First 20)
   ---------------------           ---   ------     ----  ----------------
   \INSIDER $MBBMN   JOBI          No    WEEKEND      I   UPDATE CASH ACCOUNT
   \INSIDER $MBBMN   JOBI AA       No    WEEKEND      *   ** No Alias
   \INSIDER $MBBMN   JOBI AB       No    WEEKEND      *   ** No Alias









 Next Page             Prev Page        F8-DISPLAY DETAIL      SF13-VIEW FKEYS

                                                                   BLOCK
```

Figure 4.9 – View Load List File Records Screen (VLL) – Jobs Not to Run

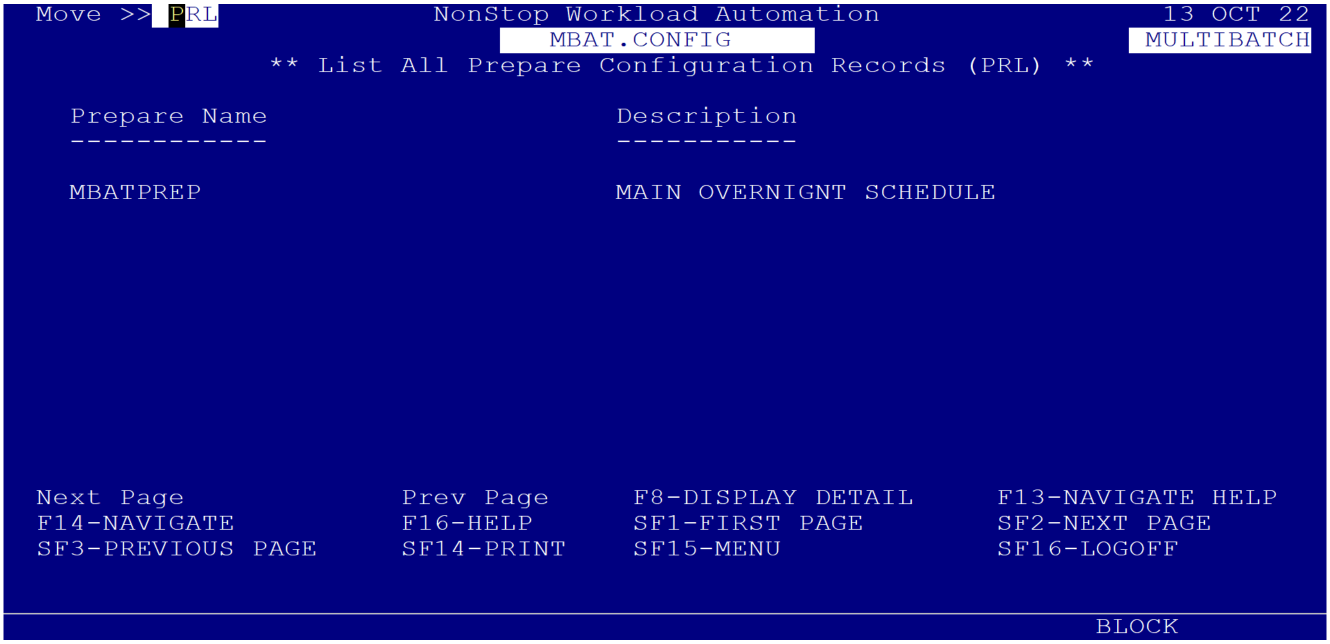
The View Load List Records Screen (VLL) can also be accessed from the Prepare Configuration Menu (PRM) via the F3 function key, or via the Navigate Facility.

This screen gives the user a list of all jobs which have/haven't run as previously described. The PREPARE phase is now complete and the schedule can be "Built".

4.13 List All Prepare Records (PRL)

The List All Prepare Configuration Records Screen (PRL) (Figure 4.10) is accessed from the Prepare Configuration Menu via the F2 function key or the Navigate Facility.

This screen displays a list of all Prepare Configuration Records. The user can use the function keys to page through the list and examine the listed records. The user can then place a marker by any record of interest and press the F8 function to display details of the selected record on the Maintain Prepare Configuration Screen (Figure 4.6).



```
 Move >> PRL              NonStop Workload Automation                 13 OCT 22
                                 MBAT.CONFIG                         MULTIBATCH
              ** List All Prepare Configuration Records (PRL) **

   Prepare Name                     Description
   ------------                     -----------

   MBATPREP                         MAIN OVERNIGNT SCHEDULE









 Next Page              Prev Page     F8-DISPLAY DETAIL     F13-NAVIGATE HELP
 F14-NAVIGATE           F16-HELP      SF1-FIRST PAGE        SF2-NEXT PAGE
 SF3-PREVIOUS PAGE      SF14-PRINT    SF15-MENU             SF16-LOGOFF


                                                                    BLOCK
```

Figure 4.10 – List All Prepare Configuration Records (PRL)

4.14 Build Phase

The Build Phase is achieved using the BCOM interface, which reads the file of MultiBatch configuration commands generated in the Prepare phase.

BCOM interprets the commands and passes them forward to a running MultiBatch process (BMON).

This file of configuration commands is known as the BUILD file and is specified on the Maintain Prepare Configuration Screen (Figure 4.6).

BCOM can access the BUILD file by entering the following commands at the TACL prompt (see Appendix H ‘Recovering a BMON’ for further details and example):

```
ASSIGN DATACONF, <Database Svol>.DATACONF

STBUILD/ IN <MultiBatch Database Svol>.DATACONF, OUT $S / <Prepare Name>

PURGE         <USER-SVOL>.ORECOVER
PURGE         <USER-SVOL>.ORECOVE0
FUP ALTER     <USER-SVOL>.NRECOVER, NO AUDIT
FUP ALTER     <USER-SVOL>.NRECOVE0, NO AUDIT
FUP RENAME    <USER-SVOL>.NRECOVER, <USER-SVOL>.ORECOVER
FUP RENAME    <USER-SVOL>.NRECOVE0, <USER-SVOL>.ORECOVE0
FUP ALTER     <USER-SVOL>.ORECOVER, ALTFILE (0,<USER-SVOL>.ORECOVE0)

BMON / NAME $<Process Name>, TERM <home term>, NOWAIT /


BCOM $<Process Name>; BUILD <Prepare Name>
```

Note 1: *The BUILD command locates the correct Build file by using the ASSIGN value to locate the and ultimately the PREPCONF file for the specified Prepare Name.*

processed. If this is not the case or the BMON process it has opened does not match the one configured in the Prepare Name then the BCOM program will ABEND.

The STBUILD utility creates the status database for "todays" run of BMON. The number of Jobs, Segments and Units configured and waiting is set to the totals issued by the PREPARE. The running, held and failed totals are reset to zero.

Note 2: *The ASSIGN specification only needs to be utilised once, during the BUILD process. Once the BMON has been built, the ASSIGN is stored within the running BMON process and a user does not need to specify the ASSIGN within their TACL macros or obey files.*

4.15 Run Schedule

Once the BUILD Phase is complete, the Schedule of "Jobs to Run" can be executed. There are four ways to "run" the schedule, namely:

1. From the TACL prompt through the MultiBatch BCOM interface.

2. From the MultiBatch Facility Interface Screen (MBF).

3. From Event Timer (EME).

4. From the Configure Job (CJB) Screen using the **Start Time** field.

Run from TACL

To run the schedule from the TACL prompt the user must first run BCOM by entering the following:

```
RUN BCOM $<Process Name>
```

The user can then:

Start a Job, Segment or Unit, by entering the following:

```
START JOB <Job Name>, e.g.: START JOB AAAA
START SEGMENT <Segment Name>, e.g.: START SEGMENT AAAA.AA
START UNIT <Unit Name>, e.g.: START UNIT AAAA.AA.01
```

Restart or Abort a Unit by entering:

```
RESTART UNIT <Unit Name>
ABORT UNIT <Unit Name>
```

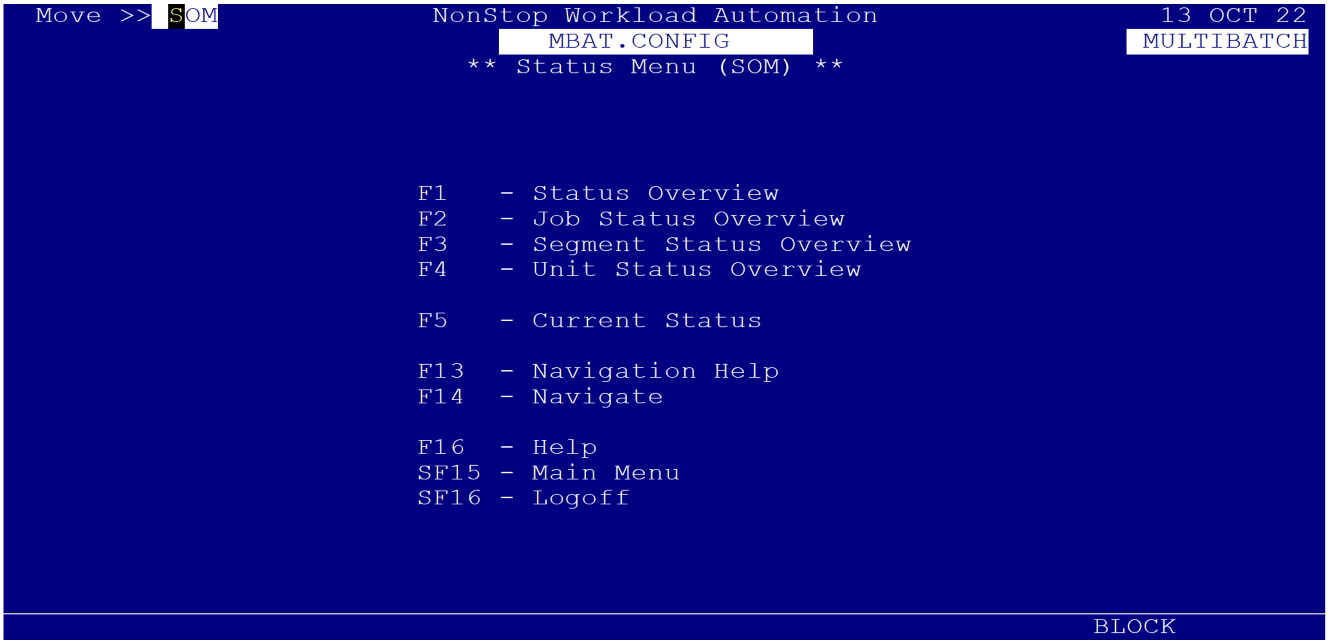
Hold or Release a Job, Segment, or Unit as follows:

```
HOLD JOB <Job Name>
HOLD SEGMENT <Segment Name>
HOLD UNIT <Unit Name>

RELEASE JOB <Job Name>
RELEASE SEGMENT <Segment Name>
RELEASE UNIT <Unit Name>
```

4.16 Facility Interface

The Facility Interface (MBF) allows the user to perform BCOM commands using a function key driven Pathway interface. The seven most common MBF operations can be performed by selecting the appropriate function key.

The MBF facility can only be accessed from the Job (SJB), Segment (SSG), or Unit (SUN) Status Monitoring screens. These can be accessed by navigating to the Status Menu (SOM) from where you can select a function key to navigate to the appropriate status display. (Figure 4.12)



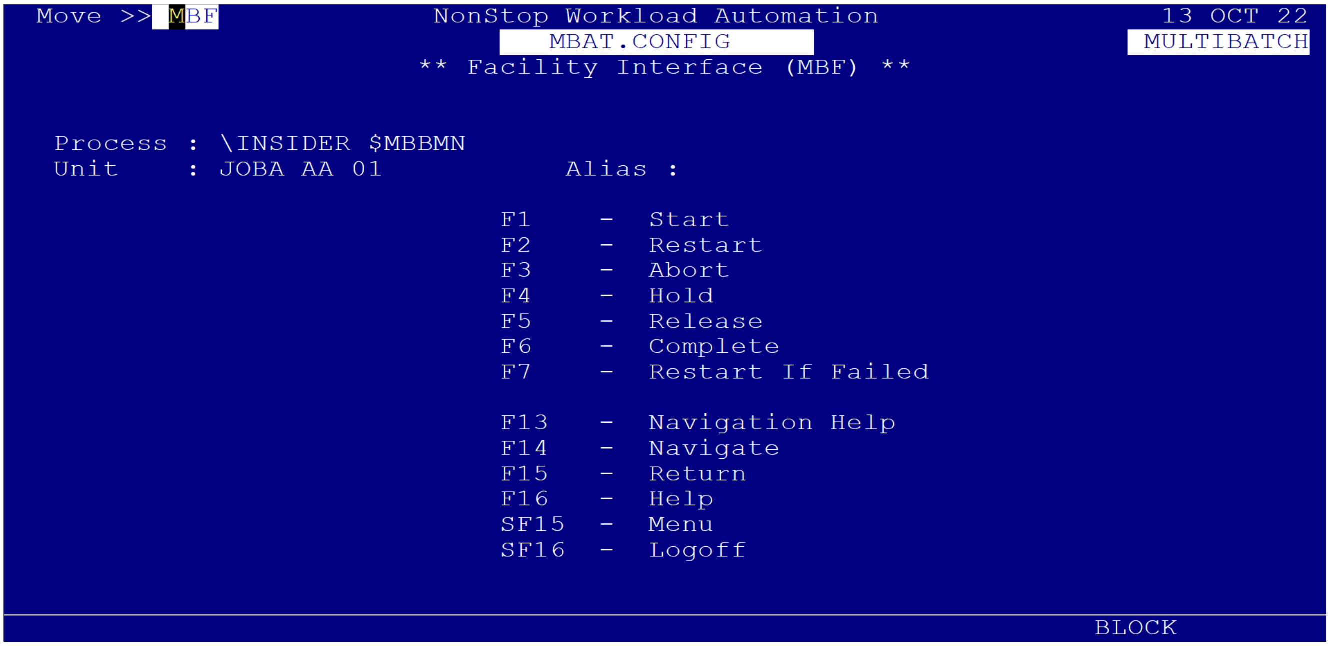
```
 Move >> SOM                NonStop Workload Automation              13 OCT 22
                                   MBAT.CONFIG                      MULTIBATCH
                             ** Status Menu (SOM) **



                           F1   - Status Overview
                           F2   - Job Status Overview
                           F3   - Segment Status Overview
                           F4   - Unit Status Overview

                           F5   - Current Status

                           F13  - Navigation Help
                           F14  - Navigate

                           F16  - Help
                           SF15 - Main Menu
                           SF16 - Logoff



                                                                    BLOCK
```

Figure 4.12 – Status Menu Screen (SOM)

The Facility Interface (MBF) can only be accessed by entering an **F** in the first column by, for example:

- A Job on the Status Job (SJB) screen and pressing the F8-DETAIL/FACILITIES function key
- A Segment on the Status Segment (SSG) screen and pressing the F8-DETAIL/FACILITIES function key
- A Step on the Status Unit (SUN) and pressing the F8-DETAIL/FACILITIES function key

An example of the MBF screen against a Step Name (JOBA.AA.01) is displayed in Figure 4.13.



```
 Move >> MBF                NonStop Workload Automation              13 OCT 22
                                   MBAT.CONFIG                      MULTIBATCH
                          ** Facility Interface (MBF) **


  Process : \INSIDER $MBBMN
  Unit    : JOBA AA 01             Alias :

                               F1     -   Start
                               F2     -   Restart
                               F3     -   Abort
                               F4     -   Hold
                               F5     -   Release
                               F6     -   Complete
                               F7     -   Restart If Failed

                               F13    -   Navigation Help
                               F14    -   Navigate
                               F15    -   Return
                               F16    -   Help
                               SF15   -   Menu
                               SF16   -   Logoff


                                                                    BLOCK
```

Figure 4.13 – Facility Interface Menu Screen (MBF)

This screen allows the user to START, HOLD or RELEASE a Job, etc by simply pressing the appropriate function key.

Note: *You can only ABORT and RESTART UNITS and so these function keys are only valid if the Facility Interface screen is accessed from the Status Unit (SUN) screen.*

Once a function key has been pressed the operation must be confirmed in case the function key was pressed accidentally. For example, if you wished to start a job, say JOB4, via SJB and then MBF, the following message appears at the bottom of the screen:

SELECT SF4 TO CONFIRM START

If the request is accepted, the appropriate status screen is displayed (after a five second delay) to reflect the new status.

If the request is rejected for any reason then an appropriate error message is displayed, for example:

REJECTED – INVALID COMMAND IN CURRENT STATE <nnn>

where <nnn> is the BMON error code (See Appendix 'E.2 BMON - BCOM Messages').

Chapter 5 Status Monitoring

5.1 Introduction

Status Monitoring from the MultiBatch Pathway screens is accomplished by ensuring that the WATCHER environment is active and able to monitor tokenised EMS events which are generated by the executing BMON processes and parsed by STREAMER.

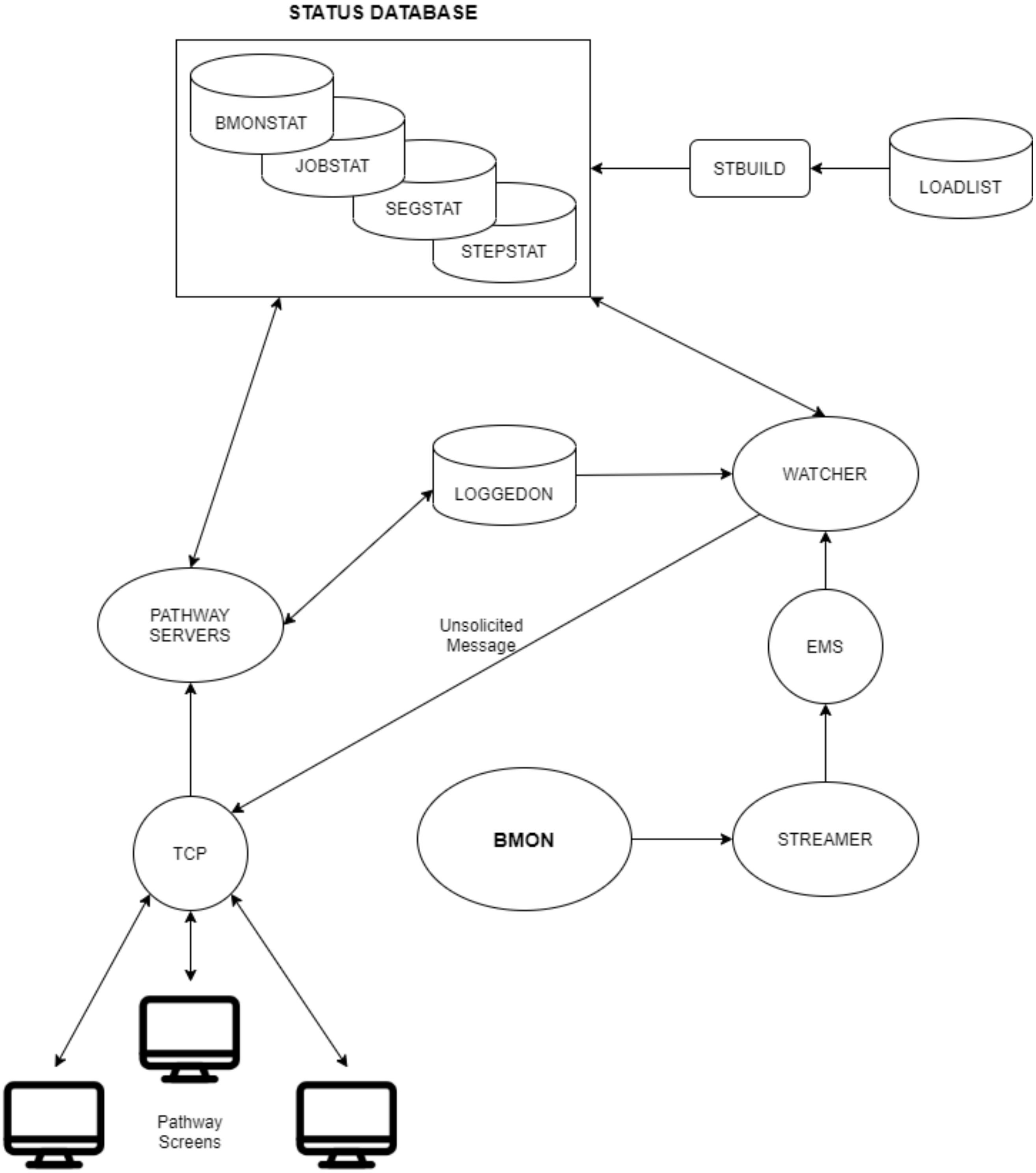


Figure 5.1 – Overview of Status Monitoring

5.2 Primary / Alt Collector Usage

MultiBatch can issue EMS events to the Primary Collector $0, or the user can specify an Alternate Collector in which to issue and monitor MultiBatch EMS events.

If a user wishes to issue MultiBatch events to an Alternate Collector, then the following steps need to be followed:

1. Configure an ALTCOLLn and ETACOLLn entry in the DCC facility.
2. Amend the various obey files supplied in the MultiBatch object subvolume, e.g. MBATOBJ.

In the obey files examples below, an Alternate Collector $ALT is being utilised. Corresponding entries in DCC need to exist, e.g.:

Logical Filename	Physical Filename
ALTCOLL1	<node>.$ALT
ETACOLL1	<node>.$ALT

Obey files to be amended:

RUNSTRM

```
RUN STREAMER/NAME $MSTR,IN MBATOBJ.BCNF0998,OUT $ALT,NOWAIT,CPU 0,TERM $ZHOME/BACKUP^CPU &
1,, DATACONF MBATDAT.DATACONF
```

RUNWAT

```
RUN WATCHER/NAME $WAT,NOWAIT,IN MBATDAT.DATACONF,TERM $ZHOME,OUT $ALT/CHANGES 1,REFRESH 30
```

CALTRIG - Note: Only amend the CALENDAR process to use OUT file.

```
RUN TRIGGER/NOWAIT,NAME  $TRG,IN MBATDAT.DATACONF,OUT $CAL/
RUN CALENDAR/NOWAIT,NAME $CAL,IN MBATDAT.DATACONF,OUT $ALT/
```

RUNCLOCK

```
RUN CLOCKMON/IN MBATDAT.DATACONF,NAME $CLOCK,OUT $ALT,NOWAIT/
```

Based on the above:

STREAMER process will issue BMON tokenised EMS events to $ALT.

WATCHER process will retrieve these and update the status database files and subsequently, the status monitor displays.

CALENDAR process will receive updates from the TRIGGER process as to when a job has been started in Event Timer. Calendar will subsequently issue Event Timer events to $ALT and when jobs have completed as reported to $ALT, update the Event Timer schedule database.

CLOCKMON process will issue its CMS configured EMS event to the $ALT Collector for monitoring by users.

5.3 Build Phase

The status monitoring as depicted in Figure 5.1 can be divided into four overlapping areas, namely:

1. The Build Phase.
2. The Pathway Environment.
3. The Watcher Environment.
4. The BMON Environment.

The Build Phase uses the Loadlist file from the Prepare to generate entries on the Status Database for the current BMON Process. Details are entered in each of the four database files with information on current Jobs, Segments, etc.

The Build Phase comprises three distinct elements:

1. **LOADLIST**. This file contains details from the Prepare phase (See Chapter 4), including the number of Jobs, Segments etc., for a particular BMON Process.

2. **STBUILD**. This program resets the values in the Status Database for a particular BMON process. Following the Prepare phase (See Chapter 4), STBUILD extracts the new values from the Prepare generated LOADLIST file and places them in the Status Database. A report for the current status of the BMON process is generated.

To execute the STBUILD program, enter the following at the TACL prompt:

```
STBUILD/IN <MultiBatch Database Svol>.DATACONF, OUT $S /<Prepare Name>, <CLEARDOWN>
```

IMPORTANT: STBUILD should NOT be executed while the particular BMON process is still running.

Note: *The **CLEARDOWN** parameter is optional and is just used to produce a report on the BMON Process and delete the entries in the Status Database. The BMON Process will then no longer appear in the Status Overview screens.*

3. **STATUS DATABASE**. This is common to three of the four areas; it contains four files (BMONSTAT, JOBSTAT, SEGSTAT and STEPSTAT) holding details of the MultiBatch schedule environments.

5.4 Pathway Environment

The Pathway environment comprises five distinct areas, namely:

1. **STATUS DATABASE.** This comprises the BMONSTAT, JOBSTAT, SEGSTAT and STEPSTAT files which contain the current status position in terms of the number of each type in configured, complete, running, waiting and held states. The CURRDEP file contains the dependency flow between jobs and segments and may differ from the original schedule because certain jobs and segments are not running.

2. **LOGGEDON FILE.** This database file holds details of which Pathway terminals are currently displaying Status Monitoring Screens.

3. **PATHWAY SERVERS.** Particular servers provide the interface between the Terminal Control Process (TCP) and the Status Database. In addition, it communicates with the **LOGGEDON** file to ensure that the data in this file is up to date.

4. **PATHWAY TCP.** The TCP provides the interface between the terminals and the Pathway Serves. In addition, when an "unsolicited" message is received from the Watcher Process it initiates an update of the status displays on the relevant Pathway terminals.

5. **PATHWAY TERMINAL.** This is the user interface to the system.

5.5 Watcher Environment

The Watcher environment comprises:

STATUS DATABASE

LOGGEDON FILE

EMS LOGS - this contains a log of EMS events generated by all of the NonStop subsystems and applications

PATHWAY TCP

WATCHER PROCESS – this Process filters information from the EMS LOG and receives information generated by BMON processes. This information is then used to update the Status Database on a time/number of events basis (i.e. update database after a specified number of events, or a specified time has elapsed). In addition, when the database has been updated, the LOGGEDON file is examined to determine which terminals are displaying status screens and an "unsolicited" message is sent to the PATHWAY TCP informing it of which terminals need to be updated.

Note: *Watcher can monitor up to 20,000 Jobs. If this number is exceeded, an EMS event is generated and monitoring continues for the first 20,000 alphabetically ordered steps.*

To run the Watcher Process, enter the following at the TACL prompt:

```
RUN WATCHER/IN <MultiBatch Database Svol>.DATACONF,NAME $<name>,NOWAIT/<run params>
```

where <run params> is a comma separated combination of the following:

- **CHANGES** *n* - Number of changes to take place between database updates (default = 10, min = 1, max = 1000)
- **REFRESH** *n* – Time in seconds between database updates from the Watcher Process. This is reset if CHANGES causes an update (default = 120 seconds).
- **LOGTIME** - A user can either specify a specific date & time (DD/MM/YYYY HH:MM:SS) or use the keyword - LAST-EVENT, i.e. LOGTIME LAST-EVENT.

  When the keyword LAST-EVENT is used, Watcher will read the BMONSTAT file to find the time of the last event processed and continue to process events from that point onwards. If a date and time is specified, then processing continues from this date and time. If this parameter is not specified, then the current date and time is used as the starting point.
- **TCP-TIME** *n* – Timeout value in seconds when writing an unsolicited message to a user Terminal (default = 60, min = 2, max = 300)
- **BACKUP-CPU** *n* – Backup CPU selection (default to highest available)

Examples:

```
RUN WATCHER / NAME $WAT, CPU 1, PRI 175, NOWAIT, IN MBATDAT.DATACONF, &
TERM $ZHOME / CHANGES 1, REFRESH 30, TCP-TIME 15, BACKUP-CPU 0
RUN WATCHER / NAME $WAT, CPU 0, PRI 175, NOWAIT, IN MBATDAT.DATACONF, &
TERM $ZHOME / CHANGES 1, REFRESH 30
RUN WATCHER / NAME $WAT, CPU 0, PRI 175, NOWAIT, IN MBATDAT.DATACONF, &
TERM $ZHOME / CHANGES 1, LOGTIME LAST-EVENT
```

5.6 BMON Environment

The BMON environment comprises the BMON Process which writes text messages to a Streamer process, which in turn, writes MultiBatch tokenised EMS events to the Primary $0 Collector, or to an Alternate Collector.

These events provide a history of successful and abnormal starts and terminations of MultiBatch jobs.

Note: *Any records in the LOGGEDON file which do not relate to active terminals result in the Pathway TCP generating a "Message Rejection Error" (HPE NonStop error message). This situation will only occur if terminals ABORT when they are displaying Status screens. To avoid this situation, the LOGGEDON file should always be cleared down when the system is shut down, or restarted.*

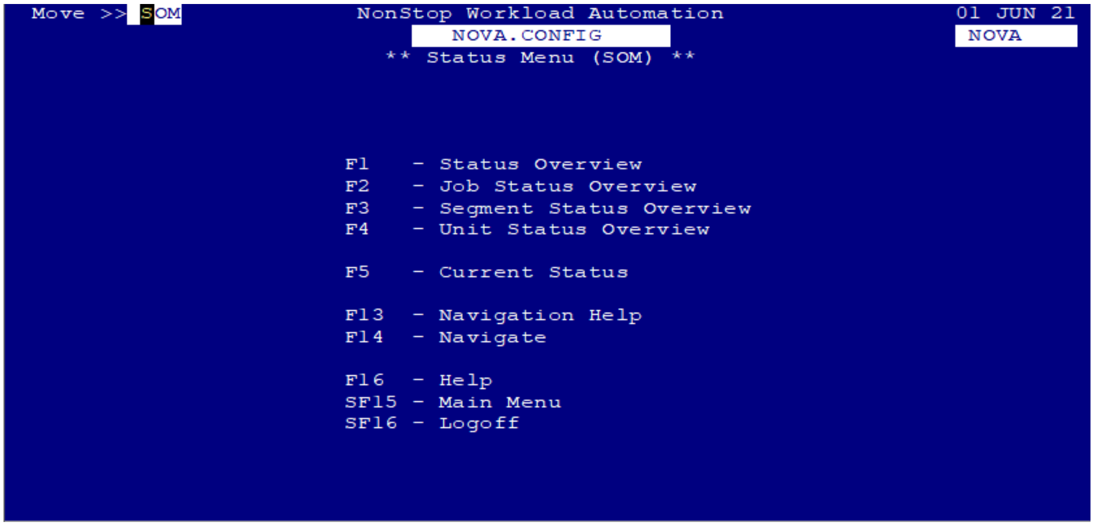
The LOGGEDON file is cleared automatically when the PSW Screen is displayed.

5.7 Status Monitoring Menu (SOM)

Having gained an overview of how the monitoring system works, we will now see how the Pathway system can be used to obtain a "real-time" view of the MultiBatch status.

In the previous section, it was explained how the Status Database is updated and how this is reflected in the monitoring displays which are updated as Jobs within the Currently configured BMON Process are executed. All the monitoring displays are accessed from the Status Menu Screen (SOM) (Figure 5.2).

The Status Menu is accessed by using the Navigate facility or by pressing the **F3** – STATUS MONITORING function key on the Main Menu Screen (MEN) (See Figure 3.1).



```
 Move >> SOM              NonStop Workload Automation                   01 JUN 21
                                NOVA.CONFIG                             NOVA
                           ** Status Menu (SOM) **



                         F1    - Status Overview
                         F2    - Job Status Overview
                         F3    - Segment Status Overview
                         F4    - Unit Status Overview

                         F5    - Current Status

                         F13   - Navigation Help
                         F14   - Navigate

                         F16   - Help
                         SF15  - Main Menu
                         SF16  - Logoff
```

Figure 5.2 – Status Menu Screen (SOM)

F1 – **Status Overview**. This accesses the Status Overview Screen (SOV) (Figure 5.3). This provides the user with an overview of the MultiBatch system at BMON process level.

Placing a marker beside a selected BMON process and pressing the **F8**-DETAIL function key displays the Unit Status display (SUN) screen, whereas, function key **SF8** will display the Status Job screen (SJB).

F2 – **Job Status Overview**. This accesses the Job Status Screen (SJB) (Figure 5.4). This screen provides the user with "run-time" information at the Job level for the current MultiBatch system. As with the previous option, selecting a Job and pressing the **F8** function key displays the associated Unit Status display, and the function key **SF8** display the Status Segment Screen (SSG).

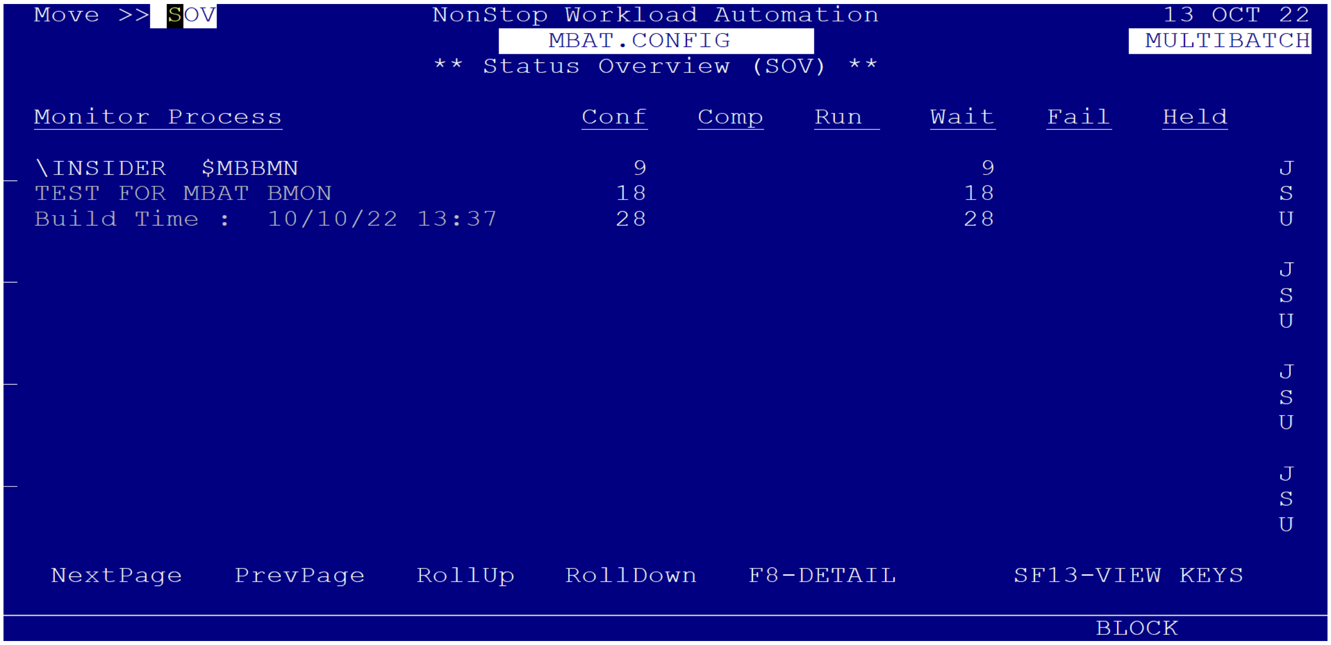
F3 – **Segment Status Overview**. This accesses the Segment Status Screen (SSG) (Figure 5.5). This screen provides the user with "run-time" information at the Segment level for the current MultiBatch system. As with the previous option, selecting a Job and pressing the **F8** function key (not SF8) displays the associated Unit Status display.

F4 – **Unit Status Overview**. This accesses the Unit Status Screen (SUN) (Figure 5.6). This screen provides the user with "run-time" information at the Unit level for the current MultiBatch system. An overlay showing information on the Status of the selected Job can be displayed by placing a marker by the selected Unit and pressing the **F8** function key.

F5 – **Current Status**. This accesses the Current Status Screen (SOC) (Figure 5.8) which displays Running, Failed Jobs and Dependency information at Job or Segment level.

5.8 Status Overview Screen (SOV)

Pressing the F1- Status Overview displays the Status Overview Screen (SOV) (Figure 5.3).



```
 Move >> SOV            NonStop Workload Automation                 13 OCT 22
                                 MBAT.CONFIG                       MULTIBATCH
                       ** Status Overview (SOV) **

 Monitor Process                     Conf   Comp   Run    Wait   Fail   Held

 \INSIDER  $MBBMN                       9                    9                J
 TEST FOR MBAT BMON                    18                   18                S
 Build Time :  10/10/22 13:37          28                   28                U

                                                                              J
                                                                              S
                                                                              U

                                                                              J
                                                                              S
                                                                              U

                                                                              J
                                                                              S
                                                                              U

  NextPage   PrevPage   RollUp   RollDown   F8-DETAIL     SF13-VIEW KEYS

                                                              BLOCK
```

Figure 5.3 – Status Overview Screen (SOV)

This screen provides a “snapshot” of the system at one particular moment in time which is constantly updated as the entities within the BMON Process execute and move through the various stages (Waiting, Running, Failed, etc.).

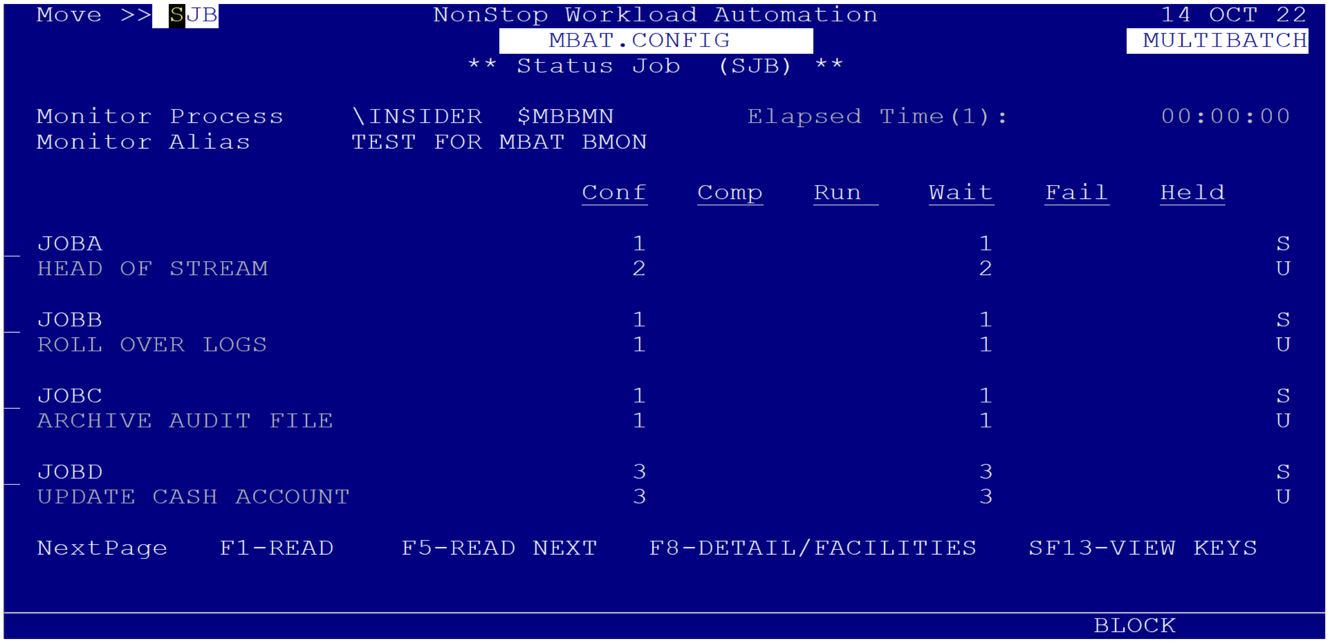
If any failures occur then these are displayed at the top of the Status Overview screen (that is, in the list displayed, failed Jobs are **always** at the top of the list). Details can be obtained on any failed Job or Unit as follows:

Press either F8-DETAIL or SF8-JOB DETAIL function keys to access Job Status Screen (Figure 5.4) for detailed Job information.

Note: *The Unit Status Screen and Job Status Screen can both be accessed from the Status Menu (Figure 5.2). These are described in detail in a later section.*

5.9 JOB Status Screen (SJB)

Pressing the F2-Job Status Overview from SOM accesses the Job Status Screen (SJB) (Figure 5.4). This screen is used if the user is primarily interested in obtaining Job information for a currently running BMON.



```
 Move >> SJB             NonStop Workload Automation              14 OCT 22
                             MBAT.CONFIG                          MULTIBATCH
                          ** Status Job  (SJB) **

 Monitor Process    \INSIDER  $MBBMN        Elapsed Time(1):          00:00:00
 Monitor Alias      TEST FOR MBAT BMON

                                   Conf   Comp   Run    Wait   Fail   Held

_JOBA                                 1                    1                 S
 HEAD OF STREAM                       2                    2                 U

_JOBB                                 1                    1                 S
 ROLL OVER LOGS                       1                    1                 U

_JOBC                                 1                    1                 S
 ARCHIVE AUDIT FILE                   1                    1                 U

_JOBD                                 3                    3                 S
 UPDATE CASH ACCOUNT                  3                    3                 U

 NextPage   F1-READ    F5-READ NEXT   F8-DETAIL/FACILITIES   SF13-VIEW KEYS


                                                                  BLOCK
```

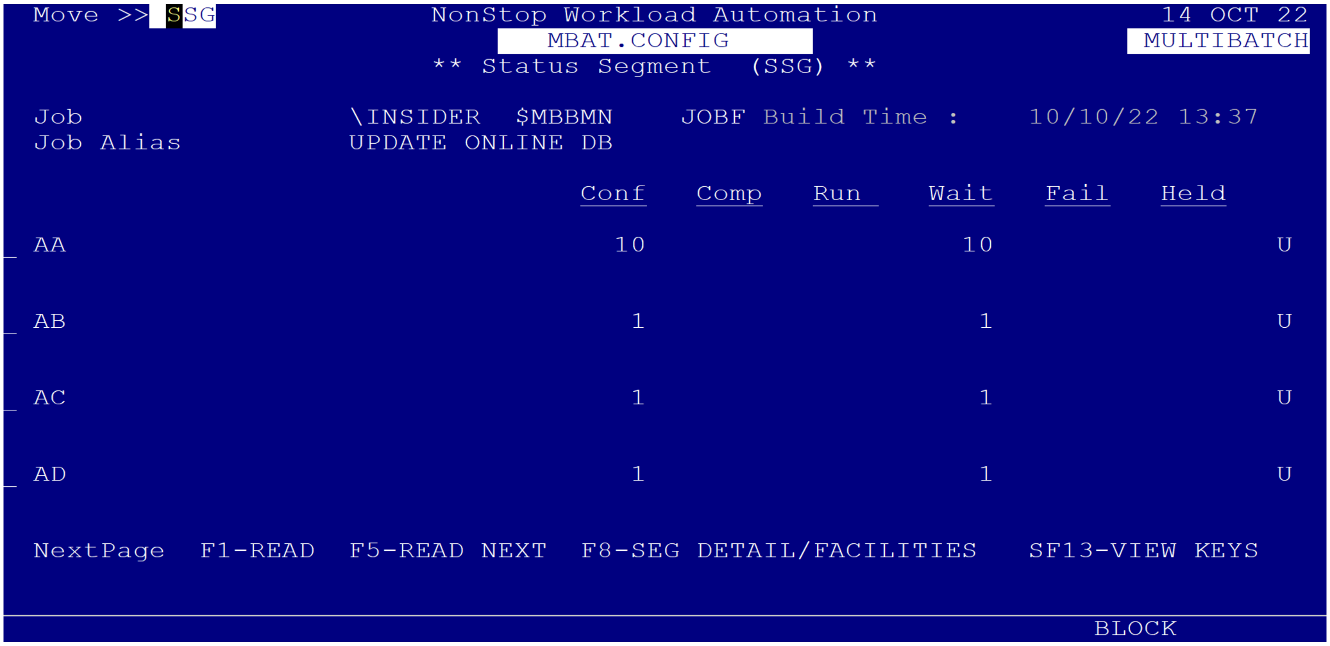
Figure 5.4 – Job Status Screen (SJB

To investigate why a Job has failed, the user can place a marker against a selected job and press either the SF8-SEGMENT DETAIL or F8-DETAIL/FACILITIES function key which accesses the Segment Status Screen (Figure 5.5).

Note the pressing F8-DETAIL/FACILITIES key with special marker value "F" will display the Facility Interface (MBF) screen (Figure 4.13) for the selected job.

5.10 SEGMENT Status Screen (SSG)

This screen is used if the user is primarily interested in obtaining Segment information for a currently running BMON.



```
 Move >> SSG             NonStop Workload Automation              14 OCT 22
                             MBAT.CONFIG                          MULTIBATCH
                        ** Status Segment  (SSG) **

 Job                \INSIDER  $MBBMN    JOBF Build Time :    10/10/22 13:37
 Job Alias          UPDATE ONLINE DB

                                   Conf   Comp   Run    Wait   Fail   Held

_AA                                  10                   10                 U

_AB                                   1                    1                 U

_AC                                   1                    1                 U

_AD                                   1                    1                 U

 NextPage  F1-READ  F5-READ NEXT  F8-SEG DETAIL/FACILITIES   SF13-VIEW KEYS


                                                                  BLOCK
```

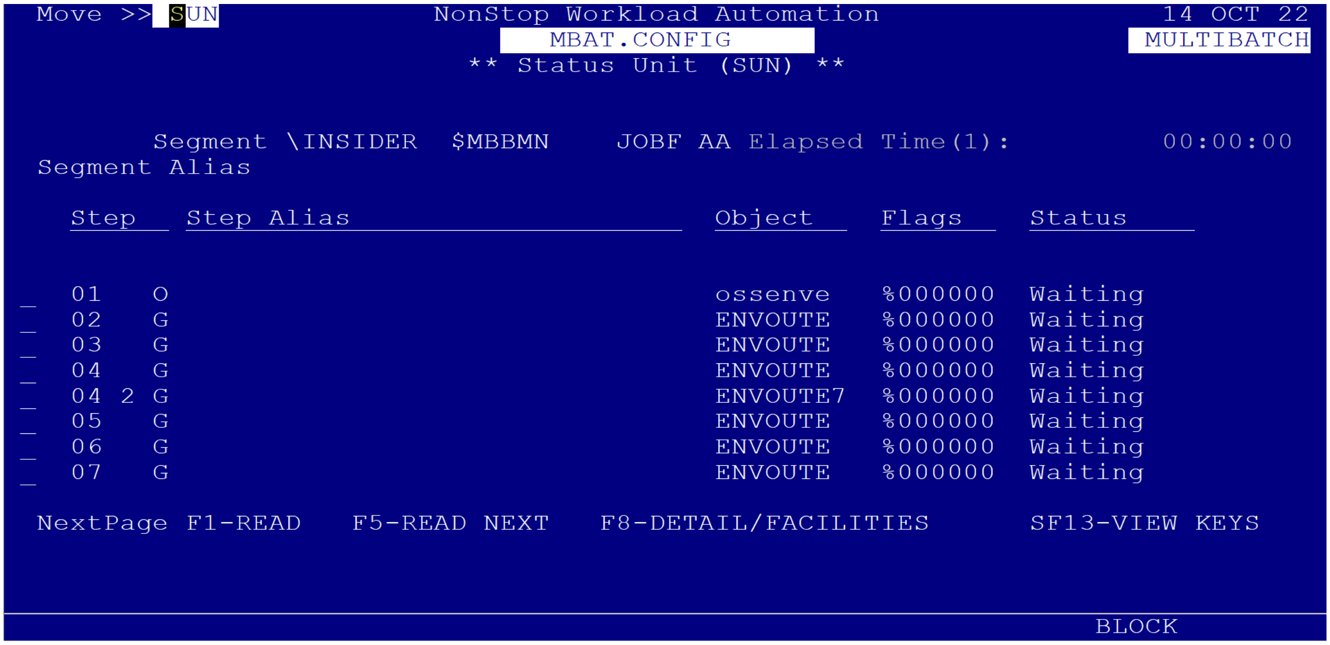
Figure 5.5 – Segment Status Screen (SSG)

To investigate why a Segment has failed, the user can place a marker against a selected segment and press the F8-SEG DETAIL/FACILITIES function key which accesses the Unit Status Screen (SUN) (Figure 5.6).

Note the pressing F8-SEG DETAIL/FACILITIES key with special marker value "F" will display the Facility Interface (MBF) screen (Figure 4.13) for the selected segment.

5.11 UNIT Status Screen (SUN)

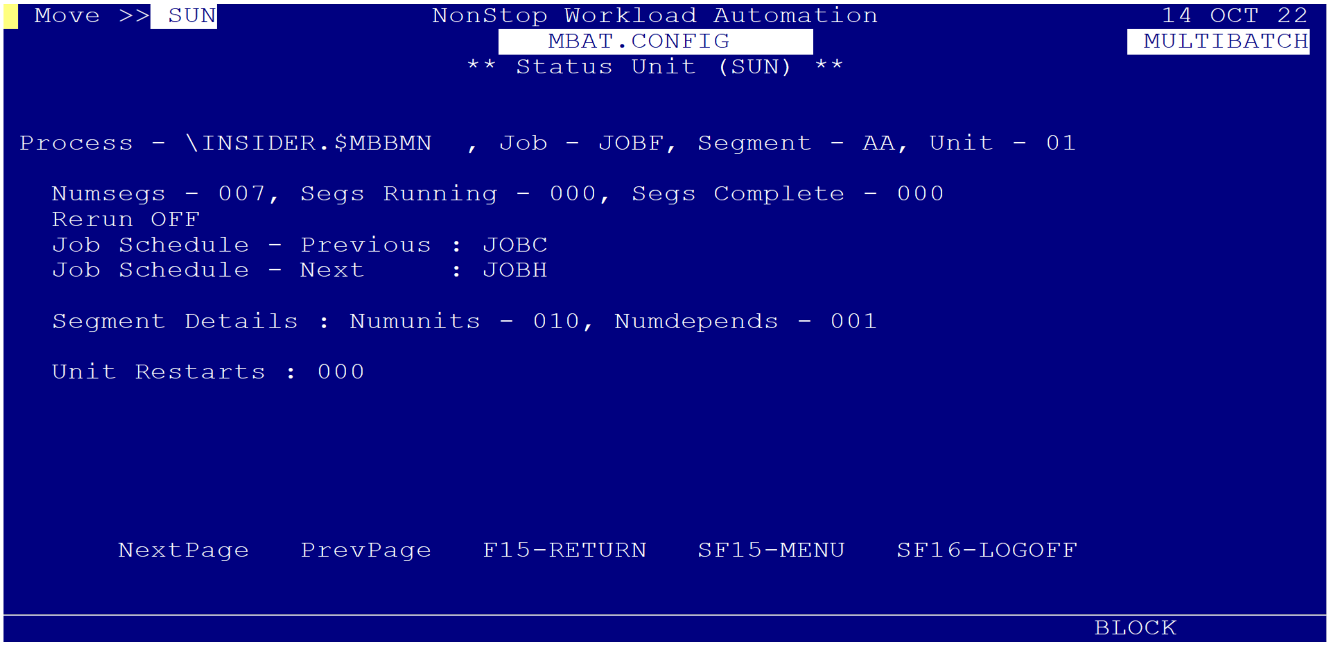
This screen is used if the user is primarily interested in obtaining Unit information for a currently running BMON.



```
 Move >> SUN                 NonStop Workload Automation                14 OCT 22
                                  MBAT.CONFIG                          MULTIBATCH
                              ** Status Unit (SUN) **


        Segment \INSIDER  $MBBMN   JOBF AA Elapsed Time(1):          00:00:00
 Segment Alias

   Step   Step Alias                               Object    Flags    Status


_  01   O                                          ossenve   %000000  Waiting
_  02   G                                          ENVOUTE   %000000  Waiting
_  03   G                                          ENVOUTE   %000000  Waiting
_  04   G                                          ENVOUTE   %000000  Waiting
_  04 2 G                                          ENVOUTE7  %000000  Waiting
_  05   G                                          ENVOUTE   %000000  Waiting
_  06   G                                          ENVOUTE   %000000  Waiting
_  07   G                                          ENVOUTE   %000000  Waiting

 NextPage F1-READ   F5-READ NEXT   F8-DETAIL/FACILITIES      SF13-VIEW KEYS



                                                                    BLOCK
```

Figure 5.6 – Unit Status Screen (SUN)

The user can place a marker by the Step number and press the **F8**-DETAIL function key. This provides the further details on the failed Unit. This information is displayed on pages as shown in Figure 5.7. The Next Page/Prev Page keys are used to move between the pages.



```
 Move >> SUN                 NonStop Workload Automation                14 OCT 22
                                  MBAT.CONFIG                          MULTIBATCH
                              ** Status Unit (SUN) **


Process - \INSIDER.$MBBMN  , Job - JOBF, Segment - AA, Unit - 01

  Numsegs - 007, Segs Running - 000, Segs Complete - 000
  Rerun OFF
  Job Schedule - Previous : JOBC
  Job Schedule - Next     : JOBH

  Segment Details : Numunits - 010, Numdepends - 001

  Unit Restarts : 000






      NextPage   PrevPage   F15-RETURN   SF15-MENU   SF16-LOGOFF



                                                                    BLOCK
```

Figure 5.7 – Unit / Step Status Screen

The number of pages which are available to the user depend on the complexity of the Job, number of Segments/ Units etc. The last overlay is indicated by a "Last Page" message displayed at the bottom of the

5.13 Current Status Screen (SOC)

The Current Status Screen (SOC) (Figure 5.8) is accessed by pressing the **F5** function key on the Status Menu Screen (SOM).

The Current Status Screen provides the user with a visual indication of running Jobs or Segments and dependency information. It will also report and highlight any failed jobs.

Figure 5.8 shows a screenshot of the system at a given moment in time. All running Jobs are shown highlighted on the screen and scheduling information is also displayed. Failures appear 'flashing' (or in italics depending on the type of equipment used) on the screen to the right of the separating bar to attract attention. When the schedule is running, the MultiBatch BMON process name and alias name are displayed in the appropriate fields, otherwise the alias field displays "ALL SCHEDULES INACTIVE".

The **SF11** function key allows the user to select Job or Segment mode. The default on entry is the Job mode which displays Running/Failed Jobs. Further detail on running Jobs can be achieved by entering the Job name in the Detail Job/Segment field and selecting the **F8**-Detail function key.

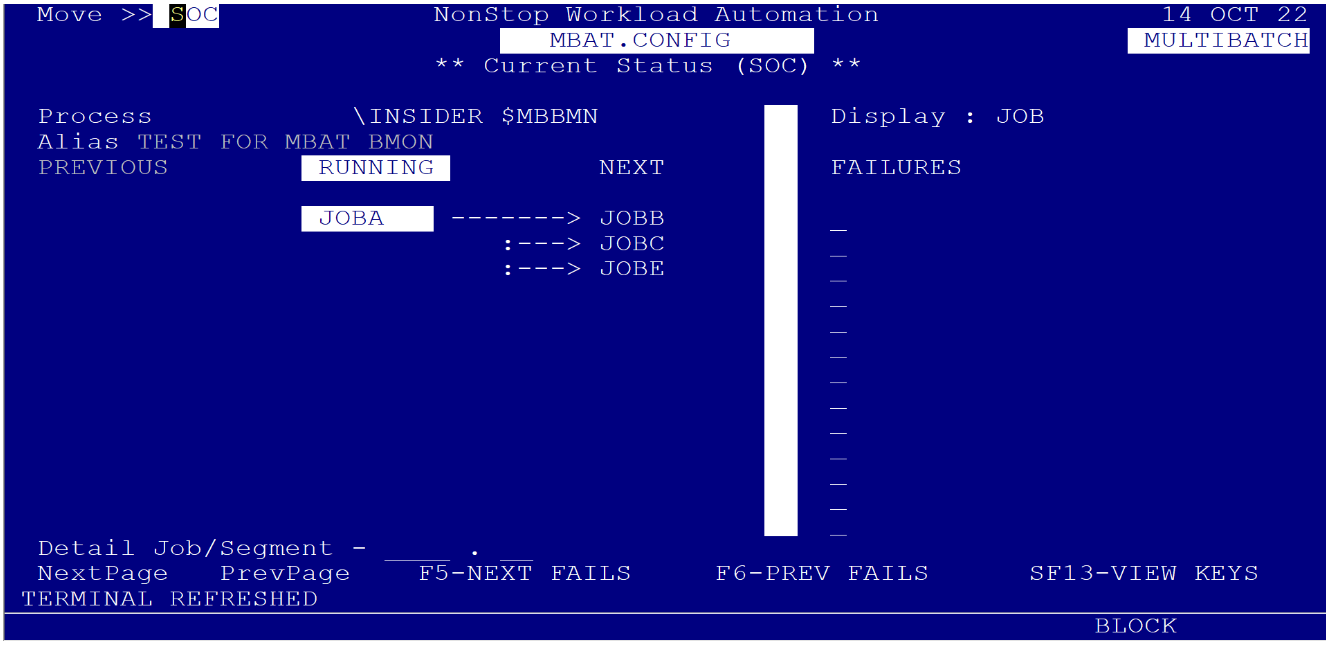
If Job Mode is selected, then the Segment Status (SSG) screen is displayed and for Segment mode the Unit Status (SUN) screen is displayed.

Further detail on a failed job can be obtained by placing a marker before the Job name in the Failures column and selecting the **SF8**-FAILURE-DEPENDENCIES function key. The appropriate dependency overlay is displayed giving you information on any dependent Jobs that will not execute.

The **SF12**-Freeze/Thaw function key can be used to "FREEZE" the display. In this state any updates are ignored and "FROZEN" is displayed, highlighted and blinking at the top of the screen.

The **SF12** function key is used again to "THAW" the screen. If the screen is inactive (no keys are pressed or no messages are received), then it is automatically thawed after 20 seconds.

When an update is received, data is re-displayed from the first running Job within the current BMON. If there are no running Jobs in the Current BMON then the first running Job in the System will be displayed. The first page of Failures is displayed for failed Jobs.



```
 Move >> SOC                NonStop Workload Automation                14 OCT 22
                                  MBAT.CONFIG                         MULTIBATCH
                           ** Current Status (SOC) **

 Process              \INSIDER $MBBMN                Display : JOB
 Alias TEST FOR MBAT BMON
 PREVIOUS             RUNNING            NEXT        FAILURES

                      JOBA     --------> JOBB        _
                                  :---> JOBC        _
                                  :---> JOBE        _
                                                     _
                                                     _
                                                     _
                                                     _
                                                     _
                                                     _
                                                     _
                                                     _
                                                     _
                                                     _
 Detail Job/Segment - _____ . ___
 NextPage   PrevPage    F5-NEXT FAILS     F6-PREV FAILS      SF13-VIEW KEYS
TERMINAL REFRESHED
                                                                BLOCK
```

Figure 5.8 – Current Status Screen (SOC)

5.14 Clearing a BMON from SOV

If a BMON is no longer required on the NonStop node, then the SOV screen may still display old statistics. In order to remove the BMON from the MultiBatch environment and the SOV display, execute the following steps:

Utilise STBUILD with a CLEARDOWN parameter, e.g.:

```
RUN MBATOBJ.STBUILD/ IN MBATDAT.DATACONF, OUT $S.#STB / <Prepare Name>, CLEARDOWN
```

Specify the <Prepare Name> of the BMON you wish to remove from the Status Database.

The CLEARDOWN parameter will clear all statistics from the status database and therefore, from the SOV display.

If this BMON should also no longer remain in the MultiBatch database, utilise the MIGRATOR facility to REMOVE the BMON to a transit file. See Appendix B Migrator.

5.15 Start Time Monitoring (CLOCKMON)

MultiBatch provides automatic scheduling facilities through:

- Event Timer (EME)
- SET JOB STARTTIME function (CJB)
- Job and Segment dependencies (CJB and CSE)

If a program administered through MultiBatch is not started in one of these three ways then that program requires a manual start through BCOM or the MBF PATHWAY screens by a user.

The CLOCKMON program provides a background monitoring process to prevent an oversight occurring during this manual start process.

To start up the CLOCKMON process:

```
RUN CLOCKMON/IN <Database Svol>.DATACONF, NAME <process-name>, NOWAIT/ <params>
```

<process name> must match that configured in the Data File Configuration Records list (DCL).

<params> are separated by a comma ",".

Values for <params> are:

BACKUP^CPU. Default is the highest available CPU. CLOCKMON automatically runs as a NonStop process pair and this parameter allows a user to specify their required backup CPU.

The event number and event SSID to be issued when a job has not been started at the appointed time are specified in the CMS Pathway screen.

```
OWNER
VALUE
VERSION
EVENT
```

You allocate your start times to given MultiBatch steps through the CMS Pathway screen. This function is described in Chapter 3.25: Configure Monitor Step (CMS).

If the allocated start time passes and a job has not started, then an EMS event is generated to $0 or an alternate collector.

Chapter 6 Error Resolution

6.1 Introduction (ERM)

In the previous chapter we have seen how the MultiBatch system is configured, the Schedules generated, and Jobs run. This chapter deals with the error facilities available to the user, to enable them to:

- Examine the Error Log which contains information on BMON Failure.

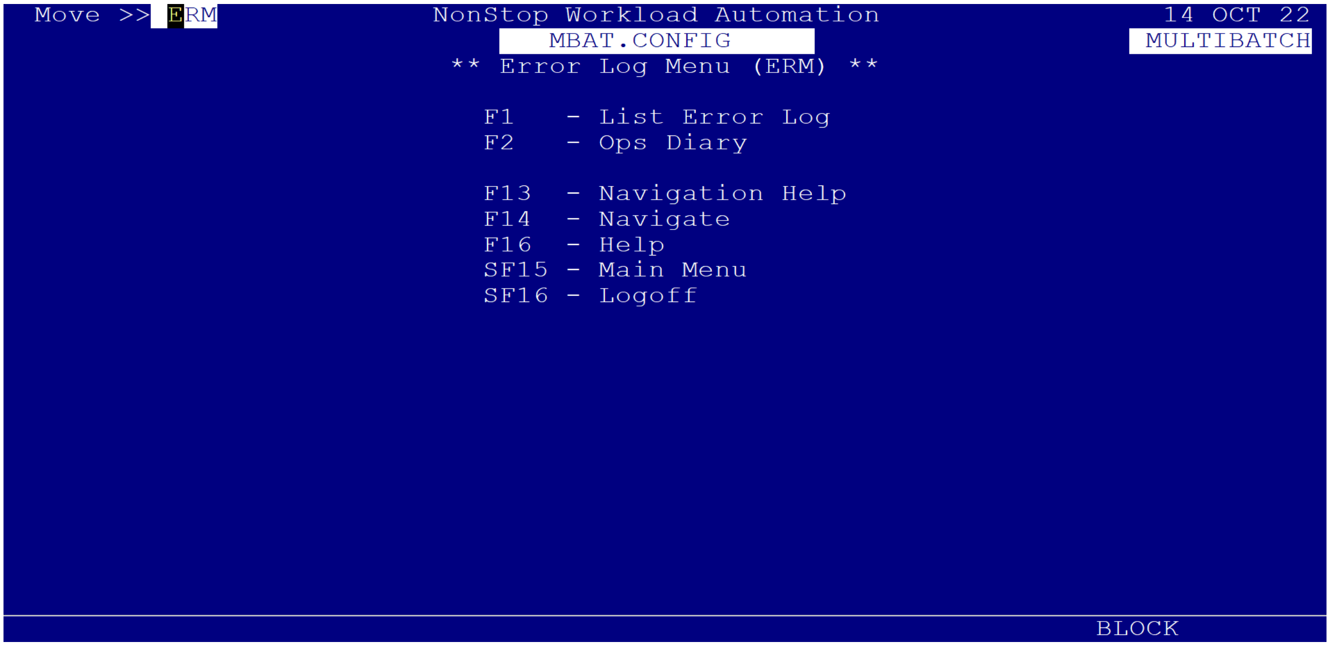
- Enter explanatory text pertaining to the failures using the Operations Diary.

The user can access these facilities in one of two ways, namely:

By pressing the **F4**-Error Resolution function key on the Main Menu (MEN) screen.

Entering the identifier ERM in the Move box and press the **F14**-Navigate function key on any screen.

In both cases the user is presented with the Error Log Menu (ERM) Screen (Figure 6.1).

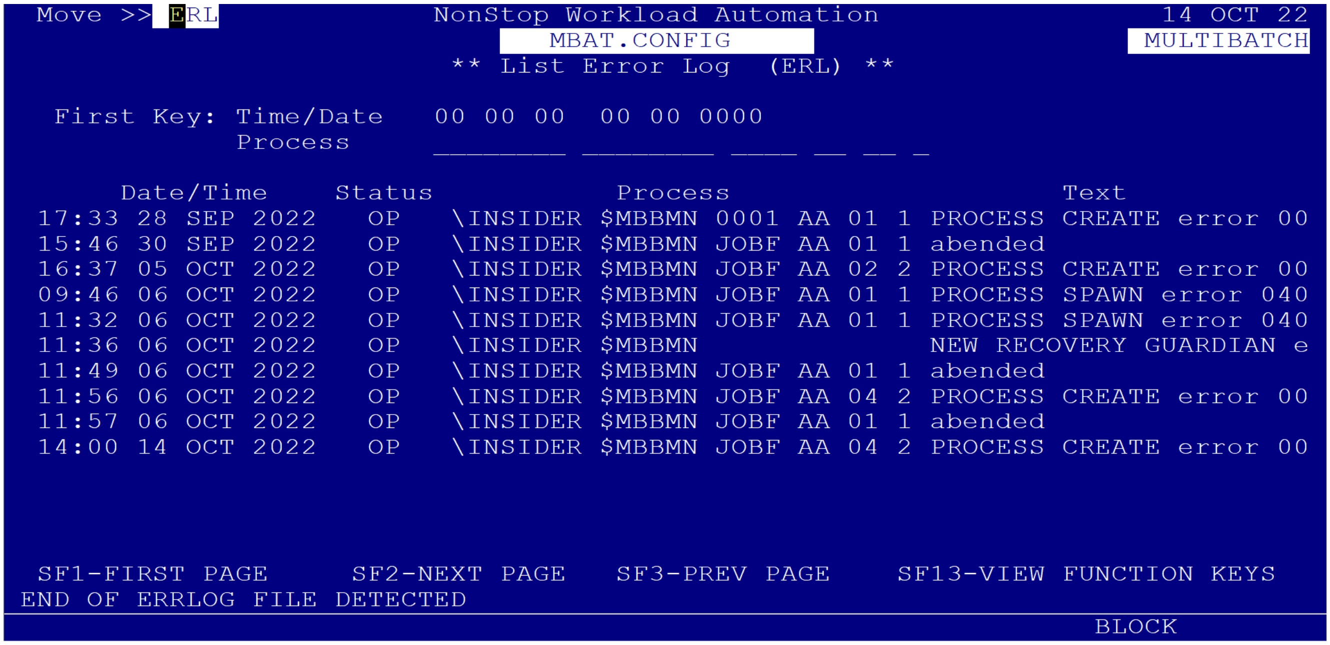


```
  Move >>  ERM                NonStop Workload Automation                 14 OCT 22
                                    MBAT.CONFIG                          MULTIBATCH
                              ** Error Log Menu (ERM) **

                                F1   - List Error Log
                                F2   - Ops Diary

                                F13  - Navigation Help
                                F14  - Navigate
                                F16  - Help
                                SF15 - Main Menu
                                SF16 - Logoff

                                                                    BLOCK
```

Figure 6.1 – Error Log Menu (ERM) Screen

6.2 Error Log (ERL)

Pressing the **F1**-List Error Log function key displays the List Error Log Screen (ERL) (Figure 6.2).



```
  Move >> ERL                NonStop Workload Automation                  14 OCT 22
                                     MBAT.CONFIG                        MULTIBATCH
                                ** List Error Log  (ERL) **

   First Key: Time/Date    00 00 00  00 00 0000
              Process      ________ ________ ____ __ __ _

       Date/Time     Status           Process                        Text
  17:33 28 SEP 2022   OP    \INSIDER $MBBMN 0001 AA 01 1 PROCESS CREATE error 00
  15:46 30 SEP 2022   OP    \INSIDER $MBBMN JOBF AA 01 1 abended
  16:37 05 OCT 2022   OP    \INSIDER $MBBMN JOBF AA 02 2 PROCESS CREATE error 00
  09:46 06 OCT 2022   OP    \INSIDER $MBBMN JOBF AA 01 1 PROCESS SPAWN error 040
  11:32 06 OCT 2022   OP    \INSIDER $MBBMN JOBF AA 01 1 PROCESS SPAWN error 040
  11:36 06 OCT 2022   OP    \INSIDER $MBBMN              NEW RECOVERY GUARDIAN e
  11:49 06 OCT 2022   OP    \INSIDER $MBBMN JOBF AA 01 1 abended
  11:56 06 OCT 2022   OP    \INSIDER $MBBMN JOBF AA 04 2 PROCESS CREATE error 00
  11:57 06 OCT 2022   OP    \INSIDER $MBBMN JOBF AA 01 1 abended
  14:00 14 OCT 2022   OP    \INSIDER $MBBMN JOBF AA 04 2 PROCESS CREATE error 00



  SF1-FIRST PAGE     SF2-NEXT PAGE   SF3-PREV PAGE    SF13-VIEW FUNCTION KEYS
 END OF ERRLOG FILE DETECTED
                                                                    BLOCK
```

Figure 6.2 – List Error Log Screen (ERL)

All errors which occur in the MultiBatch system are recorded by the Watcher Process in the Error Log. When the Error Log is accessed the user is presented with a screen (similar to Figure 6.2) which lists errors in time/date order. The following information is displayed for each error:

The date/time at which the error occurred.

The MultiBatch Unit which failed.

Explanatory text for all the system errors reported. The explanatory text is automatically generated by the MultiBatch system to provide a general description of the error code.

Note: *Only the first 23 characters of the error text are displayed; the complete message is located in the Error Txt field on the Operations Diary (OPD) Screen. The information displayed on the List Error Log Screen is protected and cannot be changed.*

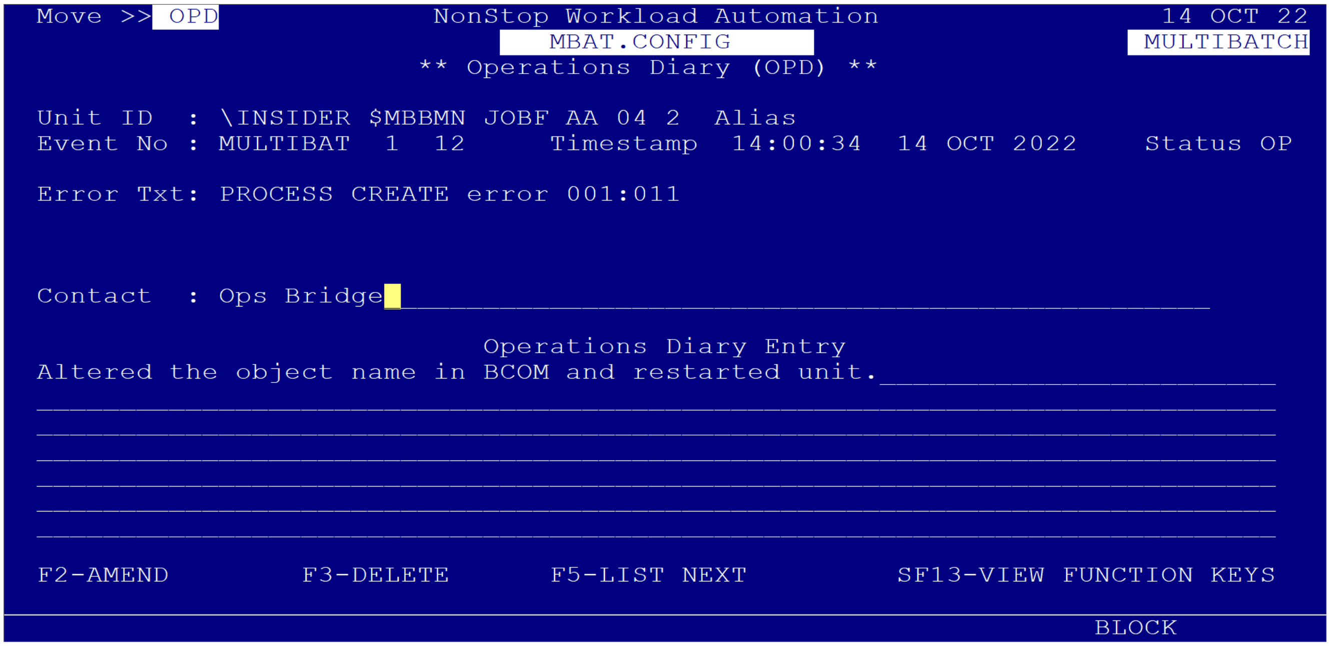
The only input which the user has to the List Error Log screen is the search criteria. Two fields are available which allow the user to specify the start time (Day/Date) and the process name.

Once the date/time and/or the process have been specified the user presses the **SF1**-First Page function key to display the first page of records which match the specified criteria.

6.3 Operations Diary (OPD)

The Operations Diary (OPD) Screen (Figure 6.3) can be accessed in one of three ways, namely:

- By using the Navigate Facility.
- By pressing the F2-Ops Diary function key on the Error Log Menu Screen (Figure 6.1).
- By placing a marker by the required record on the Error Log Screen and pressing the F8-Ops Diary function key.



```
 Move >> OPD              NonStop Workload Automation                    14 OCT 22
                               MBAT.CONFIG                              MULTIBATCH
                         ** Operations Diary (OPD) **

 Unit ID  : \INSIDER $MBBMN JOBF AA 04 2  Alias
 Event No : MULTIBAT  1  12     Timestamp  14:00:34  14 OCT 2022    Status OP

 Error Txt: PROCESS CREATE error 001:011



 Contact  : Ops Bridge_________________________________________________

                           Operations Diary Entry
 Altered the object name in BCOM and restarted unit.________________________
 ___________________________________________________________________________
 ___________________________________________________________________________
 ___________________________________________________________________________
 ___________________________________________________________________________
 ___________________________________________________________________________
 ___________________________________________________________________________

 F2-AMEND        F3-DELETE      F5-LIST NEXT        SF13-VIEW FUNCTION KEYS

                                                                   BLOCK
```

Figure 6.3 – Operations Diary (OPD) Screen

The user has access to only three fields in this screen, namely:

1. **STATUS**. This is a two-character identifier which may be entered to represent the state of an event (e.g. OP = open, CL = closed).

2. **CONTACT**. This field can be used by the user to identify the person who may be contacted for further information (e.g. on how the problem was resolved).

3. **OPERATIONS DIARY ENTRY**. This field allows the user to enter up to two pages of information relating to the reported error (the Next/Prev Page key is used to move between the pages).

The fields UNIT ID and TIMESTAMP are protected. The user presses the **F5**-LIST NEXT function key to read records from the Operations Diary.

The Amend and Delete function keys are assigned to two separate Security Class settings, OPD 01 (Amend) and OPD 02 (Delete). This will allow maintenance of this screen to be split between a User and Supervisor for example.

Chapter 7 View Audit Logs

7.1 Introduction (ALM)

The View Audit Log Facility allows the user to access the Audit Logs which keep a record of changes made to any of the screens in the MultiBatch environment. The audit information is stored in the following two database files:

- Audit Log File (AUDLOG): holds a summary of the audit information.
- Audit Detail File (AUDDET): holds detailed information on changes made using any MultiBatch database facility.

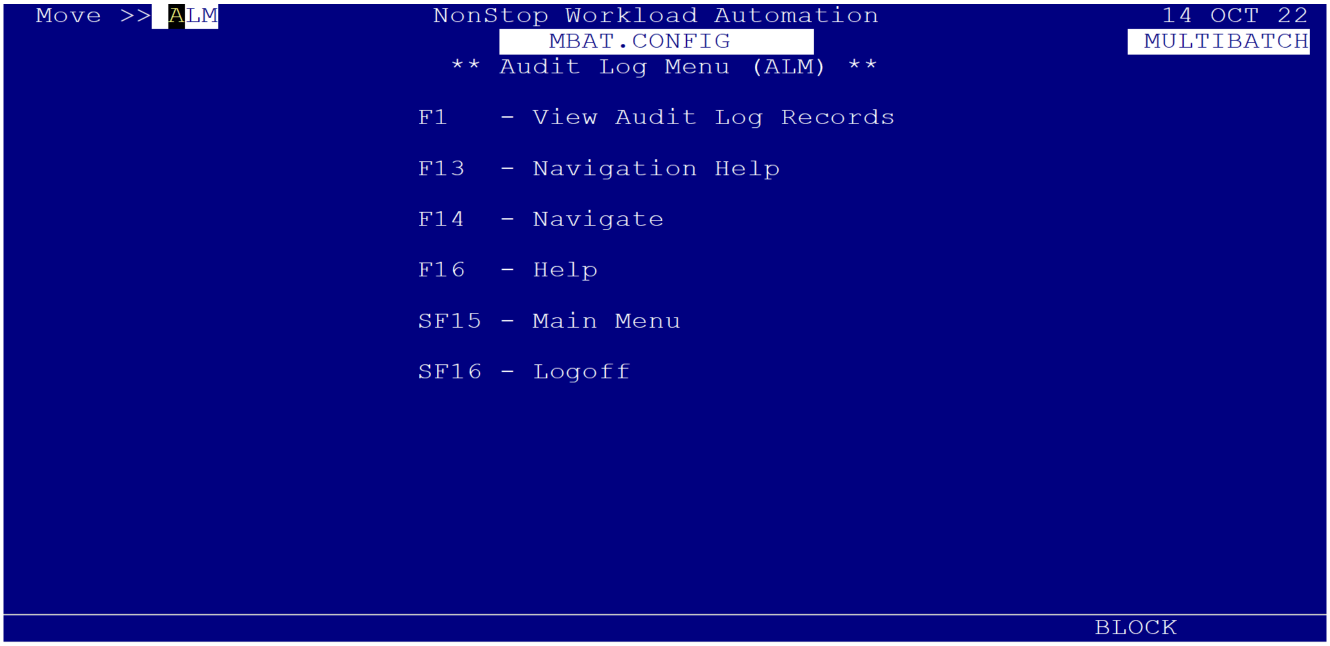
The information stored in Audit Log is displayed on the screen and is described in the following sections.

The information in the Audit Detail File is used, together with the information in the Audit Log File, by the AUDREPT facility to generate an Audit Print Report; see Appendix F.3.

The facility can be accessed in one of two ways, namely:

- By using the Navigate facility and the ALM screen identifier.
- By pressing the F5-VIEW AUDIT LOG function key on the Main Menu Screen (MEN) (Figure 3.1).

The Audit Log Menu Screen (ALM) (Figure 7.1) is then displayed.



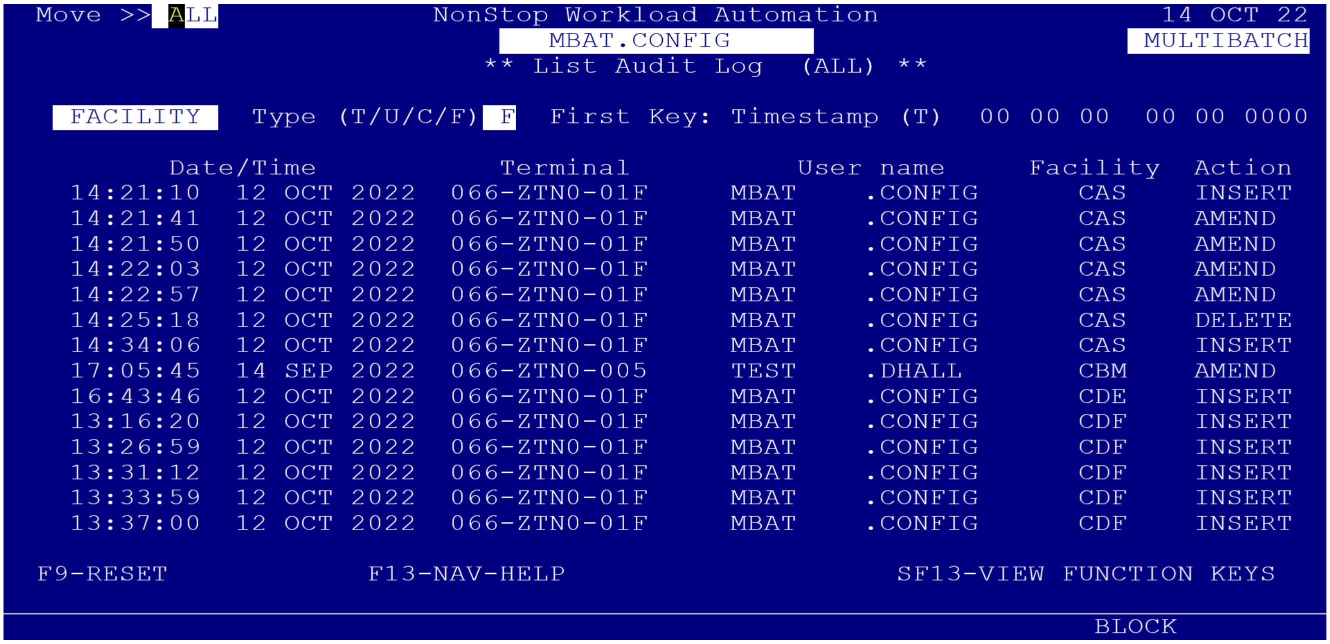
```
 Move >> ALM                NonStop Workload Automation                  14 OCT 22
                                  MBAT.CONFIG                           MULTIBATCH
                              ** Audit Log Menu (ALM) **

                         F1   - View Audit Log Records

                         F13  - Navigation Help

                         F14  - Navigate

                         F16  - Help

                         SF15 - Main Menu

                         SF16 - Logoff

                                                                     BLOCK
```

Figure 7.1 – Audit Log Menu (ALM) Screen

7.2 Audit Logs (ALL)

To use the Audit Log facility, press the F1-VIEW AUDIT LOGS function key at the Audit Log Menu.

This displays the List Audit Log Screen (ALL) (Figure 7.2).



```
 Move >> ALL              NonStop Workload Automation                    14 OCT 22
                                MBAT.CONFIG                              MULTIBATCH
                             ** List Audit Log  (ALL) **

   FACILITY   Type (T/U/C/F) F  First Key: Timestamp (T)  00 00 00  00 00 0000

          Date/Time            Terminal          User name       Facility  Action
    14:21:10  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CAS     INSERT
    14:21:41  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CAS     AMEND
    14:21:50  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CAS     AMEND
    14:22:03  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CAS     AMEND
    14:22:57  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CAS     AMEND
    14:25:18  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CAS     DELETE
    14:34:06  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CAS     INSERT
    17:05:45  14 SEP 2022  066-ZTN0-005    TEST    .DHALL       CBM     AMEND
    16:43:46  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CDE     INSERT
    13:16:20  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CDF     INSERT
    13:26:59  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CDF     INSERT
    13:31:12  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CDF     INSERT
    13:33:59  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CDF     INSERT
    13:37:00  12 OCT 2022  066-ZTN0-01F    MBAT    .CONFIG      CDF     INSERT

 F9-RESET             F13-NAV-HELP              SF13-VIEW FUNCTION KEYS

                                                                BLOCK
```

Figure 7.2 – List Audit Log Screen (ALL)

This screen displays all recorded changes to the database, for each change and identifies the following:

- The date and time at which the change occurred.
- The Pathway terminal at which the change occurred.
- The user name for the user logged on to the Pathway terminal.
- The facility being used when the change occurred (e.g. CJB to configure Job).
- The action being taken when the change occurred (e.g. INSERT).

When first displayed the user must select the order in which changes are to be displayed by specifying the TYPE field. The user then presses the **SF1**-FIRST PAGE function key to display the first page of changes in the specified order.

7.3 Type Field

The TYPE field can take one of the following values:

- T – Date/Time order (Timestamp). This lists the changes in the order which they occurred with the earliest change listed first.
- C – Terminal Name. This lists the changes by Pathway Terminal name (in alphabetical order). If there is more than one change for any given terminal they are then listed in date/time order.
- U – User Name. This lists the changes in user Id order. As with the Terminal name they are listed in alphabetical order and, if more than one occurs for a given user, they are then listed in Date/Time order.
- F – Facility (screen name). This lists the changes in facility (screen identifier) order. As with the Terminal Name and user Id they are listed alphabetically and then by Time/Date.

The user also has the option of listing the Audit Log from a specific position by entering a value in the FIRST KEY field.

The FIRST KEY field has a number of associated overlays and the user must ensure to select the correct overlay to go with the TYPE field selected using the **SF9**-VIEW LIST KEYS function key.

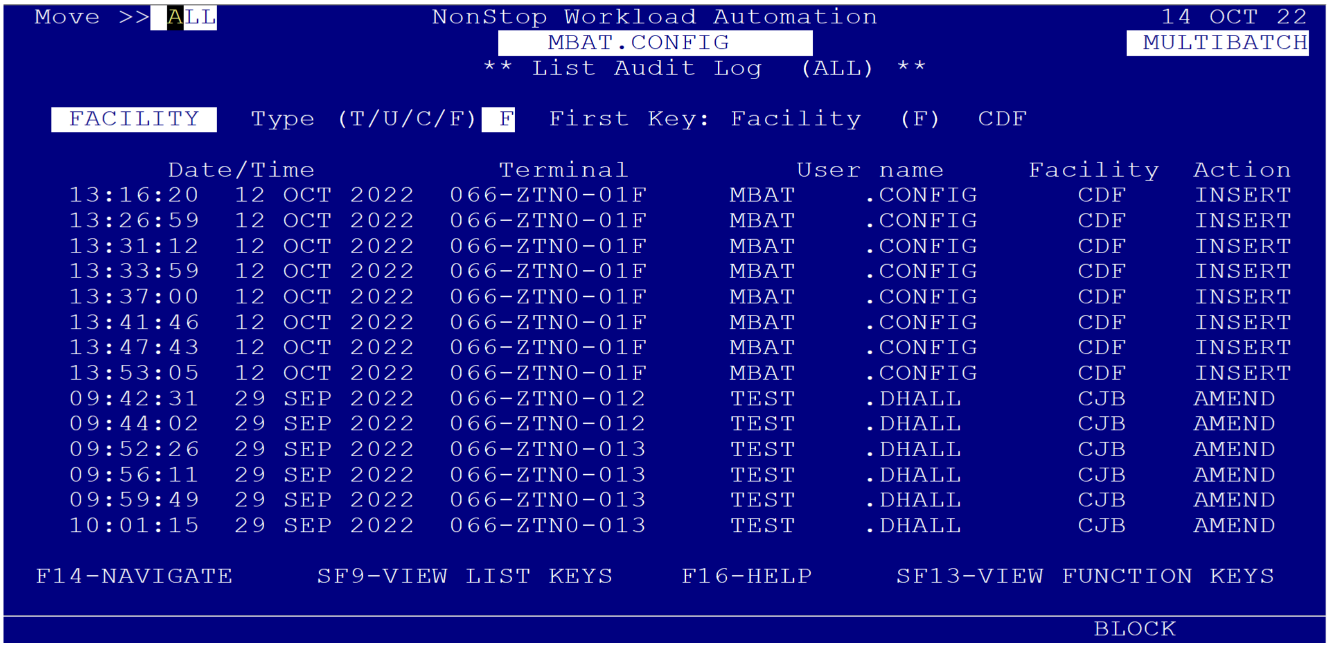
The following examples demonstrate some of these TYPE fields.

7.4 Example 1

To view modifications from Maintain Defines (CDF) onwards, the user:

- Enters “F” in the TYPE field.
- Presses the SF9-VIEW LIST KEYS function key. This is done repeatedly until FACILITY (F) appears.
- Enters “CDF” in the FACILITY field.
- Presses the SF1-FIRST PAGE function key to display the specified information (Figure 7.3).

Note: *If there are no records for the selected Facility, then the next Facility is displayed in alphabetical order.*



```
  Move >> ALL                NonStop Workload Automation                 14 OCT 22
                                   MBAT.CONFIG                       MULTIBATCH
                              ** List Audit Log  (ALL) **

   FACILITY    Type (T/U/C/F) F  First Key: Facility  (F)  CDF

          Date/Time            Terminal           User name      Facility  Action
    13:16:20  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    13:26:59  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    13:31:12  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    13:33:59  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    13:37:00  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    13:41:46  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    13:47:43  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    13:53:05  12 OCT 2022  066-ZTN0-01F      MBAT     .CONFIG      CDF     INSERT
    09:42:31  29 SEP 2022  066-ZTN0-012      TEST     .DHALL       CJB     AMEND
    09:44:02  29 SEP 2022  066-ZTN0-012      TEST     .DHALL       CJB     AMEND
    09:52:26  29 SEP 2022  066-ZTN0-013      TEST     .DHALL       CJB     AMEND
    09:56:11  29 SEP 2022  066-ZTN0-013      TEST     .DHALL       CJB     AMEND
    09:59:49  29 SEP 2022  066-ZTN0-013      TEST     .DHALL       CJB     AMEND
    10:01:15  29 SEP 2022  066-ZTN0-013      TEST     .DHALL       CJB     AMEND

  F14-NAVIGATE     SF9-VIEW LIST KEYS    F16-HELP     SF13-VIEW FUNCTION KEYS

                                                                  BLOCK
```

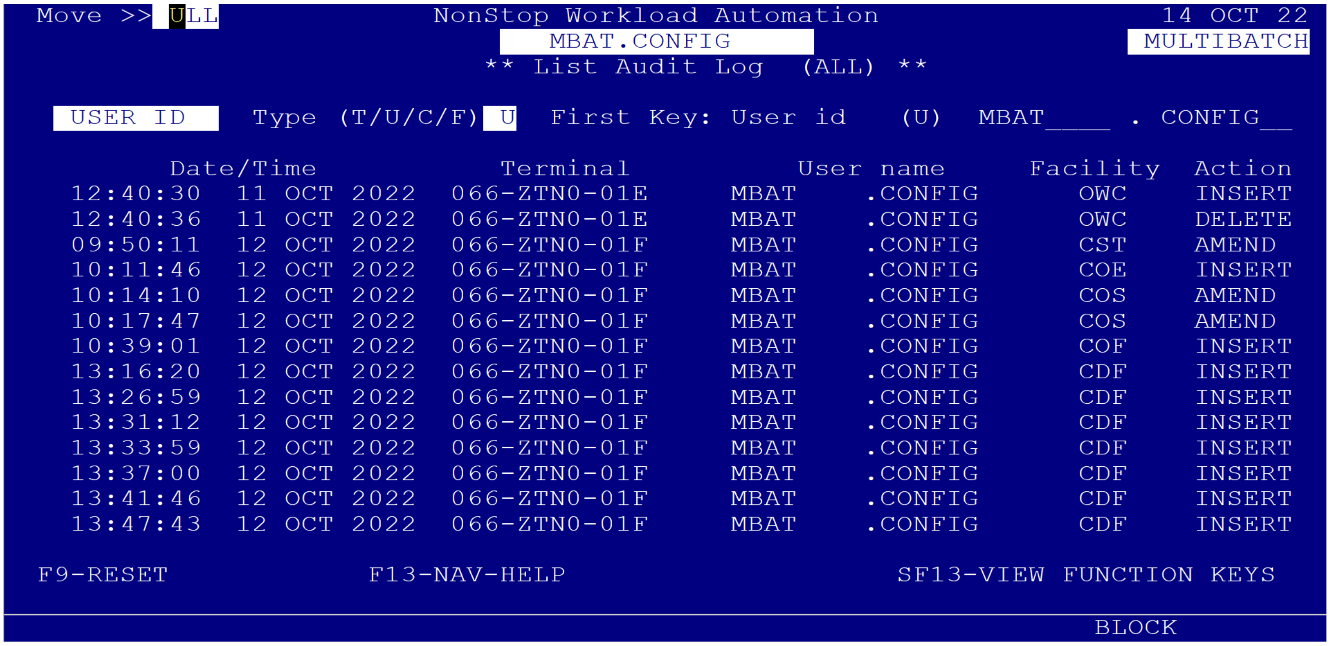
Figure 7.3 – List Audit Log Example 1 (ALL)

7.5 Example 2

To only view modifications made MBAT.CONFIG, the user:

- Enters "U" in the TYPE field.
- Presses the SF9-VIEW LIST KEYS function key. This is done repeatedly unit USER ID (U) appears.
- Enters "MBAT.CONFIG" in the USER ID field.
- Presses the SF1-FIRST PAGE function key to display the specified information (Figure 7.4).

Note: *Ensure all other previous fields are cleared, e.g. clear the PAC (F) value, or navigate back to ALL to obtain a blank screen.*



```
  Move >> ULL             NonStop Workload Automation              14 OCT 22
                                 MBAT.CONFIG                     MULTIBATCH
                           ** List Audit Log  (ALL) **

   USER ID    Type (T/U/C/F) U  First Key: User id   (U)  MBAT____ . CONFIG__

           Date/Time           Terminal          User name     Facility  Action
     12:40:30  11 OCT 2022  066-ZTN0-01E     MBAT     .CONFIG       OWC     INSERT
     12:40:36  11 OCT 2022  066-ZTN0-01E     MBAT     .CONFIG       OWC     DELETE
     09:50:11  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CST     AMEND
     10:11:46  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       COE     INSERT
     10:14:10  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       COS     AMEND
     10:17:47  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       COS     AMEND
     10:39:01  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       COF     INSERT
     13:16:20  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CDF     INSERT
     13:26:59  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CDF     INSERT
     13:31:12  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CDF     INSERT
     13:33:59  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CDF     INSERT
     13:37:00  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CDF     INSERT
     13:41:46  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CDF     INSERT
     13:47:43  12 OCT 2022  066-ZTN0-01F     MBAT     .CONFIG       CDF     INSERT

  F9-RESET             F13-NAV-HELP                SF13-VIEW FUNCTION KEYS

                                                              BLOCK
```

Figure 7.4 – List Audit Log (ALL) Example 2

Chapter 8 Event Timer

8.1 Introduction

Event Timer is designed to supplement the dependency based scheduling mechanisms provided by MultiBatch; this combination of facilities allows the user to create and build complex schedules for execution of their current and future workloads.

Various facilities are provided to allow definition of the Time and Date upon which a specific MultiBatch entity (at either JOB, SEGMENT, or UNIT level) will first execute, the periodicity at which subsequent execution of that same entity will occur, and also to allow definition of the Date and Time of the Final execution of the specified entity, up to six dates if necessary in the AT facility where a Job, or Segment, or Unit can run, or not run.

Further scheduling options are available via Event Timer CRONTAB, where a wide range and combinations of minutes, hours, dates, days, months, weeks, years can be created.

Facilities are also provided to support the creation, maintenance and use of one or more "Calendar" Files, which can each contain a series of dates upon which specific MultiBatch Jobs are to be executed.

This is supplemented by facilities to allow the creation and maintenance of a "Holiday File"; this can contain an entry for each Date which the Company has decided is not a "Working Day" for their organization.

Scheduling of Event Timer entries can be configured within screens AT (EAT), EVERY (EEV), CRONTAB (ECR).

Event Timer utilises two processes, started via obey file CALTRIG, which is located in the MultiBatch Object sub volume.

e.g. MBATOBJ.

The obey file contains two commands to start the Trigger and Calendar processes. Either the Trigger and Calendar process name, or both names, can be changed if necessary to avoid conflict with an existing process name; these names will have been specified during the MultiBatch installation process.

Note: If at a later stage, the Calendar process is changed, then the reference to the process name must be changed in the Pathway definition of the Event Timer server. This parameter format is:

```
SET SERVER PARAM CALENDAR-PROCESS "$<name>"
```

where $<name> is your chosen new name for the Calendar process.

This change is required as the Event Timer Pathway server notifies the Calendar Process that an amendment has been made to the Event Timer Schedule database, via an inter-process message; it requires the name of the Calendar Process in order to do this.

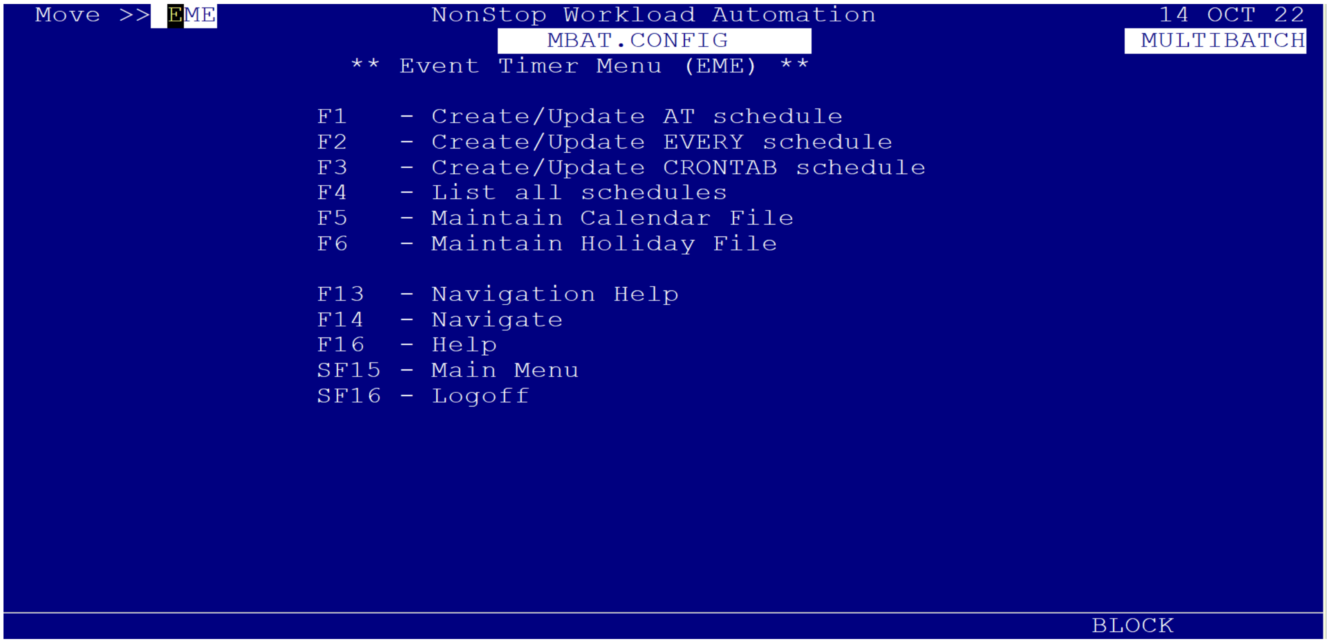
Once a change has been made to the Calendar Process name, both the Event Timer Pathway server and the Calendar process **must** be restarted to ensure the re-synchronization of both. Failure to do so will prevent reliable operation of Event Timer.

The Trigger Process can have any name as it does not need to know or be known by other components in the Event Timer function. However, the Trigger Process does need to know the process name of the Calendar process; the Calendar Process is supplied as the OUT file when initiating the Trigger process.

Example contents of CALTRIG:

```
RUN MBATOBJ.TRIGGER / NOWAIT, NAME $ETRG, IN MBATDAT.DATACONF, OUT $CAL/

RUN MBATOBJ.CALENDAR / NOWAIT, NAME $CAL,  IN MBATDAT.DATACONF/

COMMENT     The TRIGGER Process needs to access the TRIGFILT File.

COMMENT     To ensure correct operation, TRIGFILT and TRIGGER

COMMENT     MUST reside in the same subvolume.

COMMENT     If this is not the case the TRIGGER process will ABEND.
```

8.2 Main Menu (EME)



```
 Move >> EME                NonStop Workload Automation                 14 OCT 22
                                  MBAT.CONFIG                           MULTIBATCH
                         ** Event Timer Menu (EME) **

                       F1    - Create/Update AT schedule
                       F2    - Create/Update EVERY schedule
                       F3    - Create/Update CRONTAB schedule
                       F4    - List all schedules
                       F5    - Maintain Calendar File
                       F6    - Maintain Holiday File

                       F13   - Navigation Help
                       F14   - Navigate
                       F16   - Help
                       SF15  - Main Menu
                       SF16  - Logoff







                                                                    BLOCK
```

Figure 8.1 – Event Timer Main Menu

The Event Timer Main Menu (Figure 8.1) is accessed from the MultiBatch Main Menu screen - MEN (Figure 3.1) by pressing the **F8 Event Time Scheduling** function key.

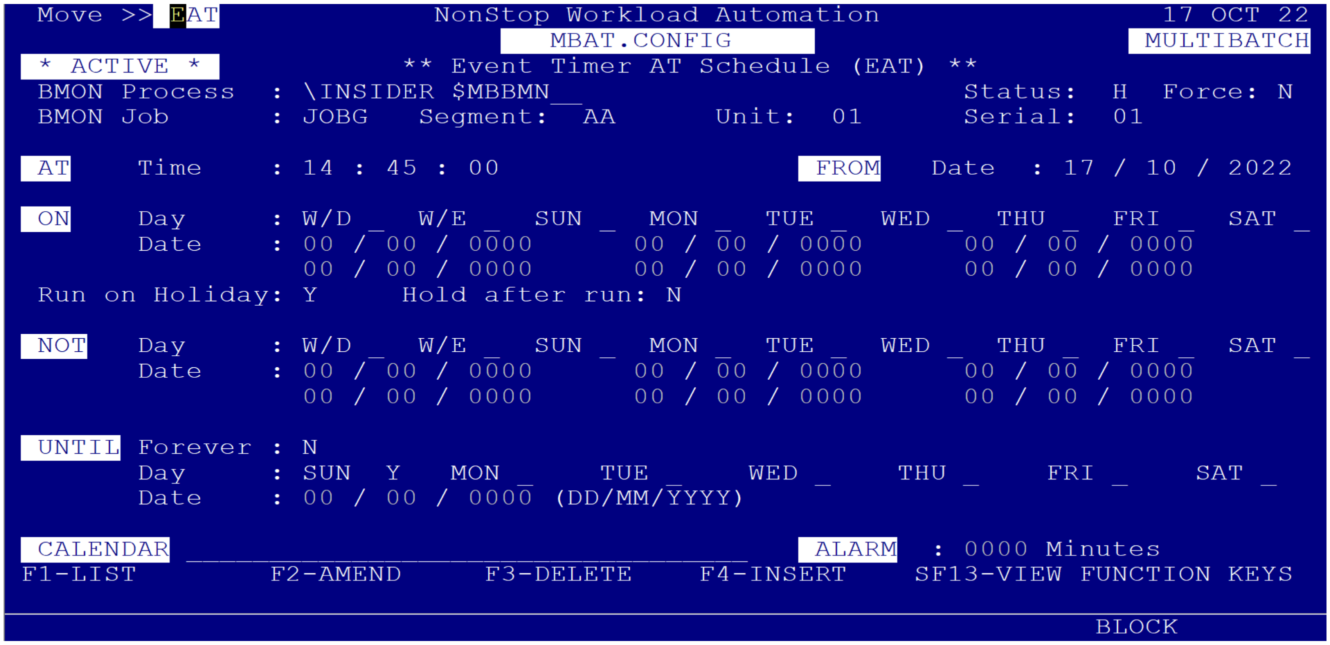
The Event Timer Main Menu gives the user access to the Event Timer Pathway screens which are described in the following sections.

8.3 Create/Update AT Schedule (EAT)

Selecting function key **F1** at the EME Menu displays the “Event Timer AT Schedule” Screen (Figure 8.2).

This provides facilities for AT Schedule Definition and Maintenance.

Note: *Any error or informative messages are displayed at the bottom of the display under the list of available function keys*



```
  Move >> EAT                 NonStop Workload Automation                     17 OCT 22
                                     MBAT.CONFIG                             MULTIBATCH
  * ACTIVE *              ** Event Timer AT Schedule (EAT) **
  BMON Process     : \INSIDER $MBBMN__                            Status:  H   Force: N
  BMON Job         : JOBG    Segment:  AA       Unit:  01         Serial:  01

  AT      Time     : 14 : 45 : 00                      FROM    Date   : 17 / 10 / 2022

  ON      Day      : W/D _   W/E _   SUN _   MON _   TUE _   WED _   THU _   FRI _   SAT _
          Date     : 00 / 00 / 0000          00 / 00 / 0000          00 / 00 / 0000
                     00 / 00 / 0000          00 / 00 / 0000          00 / 00 / 0000
  Run on Holiday: Y       Hold after run: N

  NOT     Day      : W/D _   W/E _   SUN _   MON _   TUE _   WED _   THU _   FRI _   SAT _
          Date     : 00 / 00 / 0000          00 / 00 / 0000          00 / 00 / 0000
                     00 / 00 / 0000          00 / 00 / 0000          00 / 00 / 0000

  UNTIL   Forever  : N
          Day      : SUN  Y   MON _     TUE _     WED _     THU _     FRI _     SAT _
          Date     : 00 / 00 / 0000 (DD/MM/YYYY)

  CALENDAR __________________________________            ALARM  : 0000 Minutes
 F1-LIST        F2-AMEND      F3-DELETE     F4-INSERT     SF13-VIEW FUNCTION KEYS

                                                                         BLOCK
```

Figure 8.2 – Create/Update MultiBatch ‘AT’ Schedule (EAT) Screen

The system is designed such that all entries made on this screen are immediately vetted for consistency and conformance to a set of rules governing the permissible mixing of criteria. Any attempt to create a schedule record which violates these rules is rejected with an error message.

Internally, these rules are implemented such that a Schedule Record can only be created if:

- It would allow the JOB, SEGMENT or UNIT it references to be executed at least once, at some future time or date.
- It does not conflict with another Schedule Record with the same JOB, SEGMENT, UNIT and SERIAL.

The Screen Fields and their meanings are as follows:

“**BMON PROCESS**”. The name of the Node and BMON Process which controls the JOB, SEGMENT or UNIT to be scheduled for execution. This is a mandatory entry.

“**STATUS**”. This Field is read by the PATHWAY Schedule Build Server, and updated by the Event Timer Calendar and Trigger Processes during various stages in the initiation and execution phases. Possible values are:

S Scheduled	(this Screen/Trigger Process)
I Initiated	(Calendar Process)
R Running	(Trigger Process)
X Failed	(Trigger Process)
H Held	(this Screen/Trigger Process)

The Default value is “**S**”.

“**FORCE**”. Placing “**Y**” in this Field allows the user to force the IMMEDIATE execution of a JOB/SEGMENT/UNIT.

Note: *Use of the “FORCE” facility will NOT affect the scheduling of the NEXT execution of that entity. You cannot ‘INSERT’ a record with the ‘FORCE’ Field set to ‘Y’; this can only be done if you are amending an*

The Default Value is "**N**".

"**BMON JOB**". The four-character name of the MultiBatch Job for which you are creating a schedule record.

This is a mandatory entry, and MUST be exactly four characters in length to satisfy the MultiBatch Job Name syntax requirements.

"**SEGMENT**". The two-character name of the MultiBatch Segment to be invoked.

This is an optional entry, but must be specified if the UNIT Field is to contain an entry; if omitted the entity will be invoked at Job Level.

"**UNIT**". The two-digit number of the MultiBatch Unit to be invoked.

This is an optional entry: if omitted the entity will be invoked at SEGMENT or JOB Level depending upon the contents of the preceding Field.

"**SERIAL**". This numeric Field has a default value of "01", and can range from "0" to "99". The serial number is used to create different schedule records for the same MultiBatch Job, Segment or Unit, e.g. JOBA @ 18:00 on Monday, Tuesday and Wednesday is serial number 01 and JOBA @ 19:00 on Thursday and Friday is serial number 02. This field is mandatory.

"**AT TIME**". This is the (24-hour clock) time of day that the specified entity is to be executed for each schedules occasion. This is a mandatory Field.

Note: *If the users wishes to run an AT entry at multiple times of the day, then the AT record must be amended to these future times, following the completion of the first AT entry. It is recommended to use the Every (EEV) screen to run multiple occurrences of the JSU in the same day.*

"**FROM DATE**". During an initial setup of a Schedule Record, in advance of live implementation, a FUTURE "FROM DATE" could be set on this Screen as the Record is inserted.

This will identify the Date FROM which that Schedule Record will be deemed Operational.

This allows the "FROM DATE" to be used as the basis for calculation of the FIRST execution Timestamp for a Schedule Record.

This is a mandatory Field; Default value is the current Date.

"**ON DAY**". Placing "Y" in the W/D (Weekday) Field will cause execution to be scheduled on ALL Weekdays (Monday to Friday) up to, and including, the "UNTIL" value.

Placing "Y" in the W/E (Weekend) Field will cause execution on EACH Saturday AND Sunday up to and including the "UNTIL" value.

A "Y" may be placed in up to six of the individual Days (SUN, MON, TUE, …SAT) Fields. You cannot fill in BOTH W/D and W/ E Fields, all of the Day Fields or any combination of Day and Weekend. (If you wish to schedule a Job to be run EVERY day, you just need to fill in the "AT TIME" and a suitable "UNTIL" value).

"**ON DATE**". This field allows entry of up to six different Dates on which to schedule execution.

These can be in any order, but must ALL be in the "future". (The current date can be input, but the "AT TIME" must not be less than, or equal to, the current time). You cannot supply values for BOTH "ON DAY" and "ON DATE".

"**RUN ON HOLIDAY**". A 'Y' in this Field will cause execution to be rescheduled without reference to the Holiday File. Any Schedule Record which has this Field set will be assessed as eligible for execution on the days for which an entry exists in the Holiday File. The default value is 'Y'. This can be changed if necessary, by altering the Pathway Event Timer server class Parameter 'RUN-ON-HOLIDAY' to 'N'.

"**HOLD AFTER RUN**". A 'Y' in this field will cause the BMON Job to be placed in the HOLD state in the Event Timer AT screen, once it has finished executing. The default value is 'N'.

"**NOT DAY**". Placing 'Y' in the W/D (Weekday) Field will prevent execution on ALL Weekdays up to and including the "UNTIL" value if this falls on a Weekday.

Placing "Y" in the W/E (Weekend) Field will prevent execution on any Saturday or Sunday up to and including "UNTIL" value if this falls on a Weekend.

A "Y" may be placed in up to six of the individual Days (SUN, MON, TUE, …SAT) Fields. You cannot fill in BOTH the W/D and W/E Fields, OR the W/E and all of the DAY Fields, OR any combination which would prohibit the Job from running at all.

Note: *Any combination of "ON DAY" and "NOT DAY" is disallowed.*

"**NOT DATE**". Up to six different Dates may be specified upon which execution is NOT to take place.

The "NOT DATES" can be in any order, but must ALL be in the "future". (The current date can be input, but the "AT TIME" specified must not be less than or equal to the current time).

Note: *you cannot combine the use of BOTH **"NOT DAY"** and **"NOT DATE"**. Combinations of "**ON DATE**" and "**NOT DATE**" are disallowed.*

"**UNTIL FOREVER**". Placing "Y" in this Field signals that execution is to be scheduled (at the intervals specified in the previous Fields), until further notice. You cannot use an "ON DATE" with "UNTIL FOREVER". (Or an "UNTIL" of any variety).

"**UNTIL DAY**". You can specify only one DAY Field, and execution will be scheduled up to AND including day. The system will determine whether to accept an "UNTIL DAY" entry.

From the standpoint of the current Day and Time when the schedule record is created, after taking the "FROM DATE" into consideration:

Specifying an "UNTIL DAY" of the Current day is not allowed if "AT TIME" is less than the current time. You

cannot specify "UNTIL FOREVER" and an "UNTIL DAY".

"**UNTIL DATE**". This Field allows entry of a single Date, which must be in the future. Execution will be up to AND including that date.

You cannot use BOTH an "ON DATE" and an "UNTIL DATE". You cannot specify either "UNTIL FOREVER" or "UNTIL DAY" with "UNTIL DATE".

If execution is scheduled for Weekends ONLY, "UNTIL DATE" must fall on a Weekend. Similarly, for execution scheduled ONLY for Weekdays, the "UNTIL DATE" must fall on an otherwise valid processing day.

"**CALENDAR**". Use this Field to specify the name of a Calendar File which has been previously build using the "MAINTAIN CALENDAR FILE" Facilities. The Calendar filename must contain at least the vol.subvolume.filename, with a node name prefix if required.

The nominated Calendar File will be opened, and the first suitable date used to calculate the Timestamp for next execution of this MultiBatch entity. Once this has completed the system will consult the same Calendar File for the next execution Timestamp.

Event Timer can support any number of Calendar Files; it may be useful to arrange that the names of these accord to a standard which reflects the purpose of each. For example, a Calendar File which contains the dates of the last Thursday of each month for 2021 could be named **$DATA.CAL2021.LASTTHUR**

Note: *An example audited Calendar file is supplied within the MultiBatch database, e.g. MBATDAT.CALFIL. If you wish to create your own Calendar file, then the file attributes are:*

REC 6
BLOCK 4096
EXT (4 , 32)
KEYLEN 6

A Schedule History of the selected record can be viewed by pressing **SF5** which will display the 'Schedule History' screen. (Figure 8.7). The fields for this screen are described below:

"**INSERTED/LAST AMENDED**". This contains the Date and Time (the Timestamp) of the creation of this Record or the last user amended to its values.

"**PREVIOUS RUN REQUESTED**". The Timestamp at which the last request specified this MultiBatch JOB, SEGMENT or UNIT should be executed.

This includes any adhoc request via the "FORCE" function of the CREATE/UPDATE Screen.

"**PREVIOUS RUN FINISHED**". The Timestamp at which this MultiBatch JOB, SEGMENT or UNIT last completed execution. If the entity is still running, this Field will contain either:

- The timestamp of completion of the last run.
- The caption ** Never Run ** if the current execution is the first or only run of this entity.

In both cases the legend will be displayed in Inverse Video to the right of this field: "**<<<< INCOMPLETE**"

"**PROJECTED NEXT RUN**". The Timestamp at which the Calendar Process will next initiate execution of this entity.

"**PROJECTED LAST RUN**". The Timestamp at which the Calendar Process will initiate the final execution of this entity; if a Calendar File has been specified, the LAST RUN Date will be taken from that.

"**CALENDAR FILE**". If this Schedule Record was set up to use a Calendar File, the name is displayed here.

"**CURRENT STATUS**". This Field can display one of several values:

- **SCHEDULED**. The entity is not currently processing; the Calendar process has calculated its next execution Timestamp.
- **INITIATED**. The Calendar Process has created an EMS Event to signal execution of this entity.
- **RUNNING**. The Trigger Process has accessed MultiBatch via the BCOM command interface, and successfully requested execution of this entity.
- **FAILED**. The Trigger process has either:

  1. Not been able to persuade MultiBatch to execute this entity; the reason will be sent to $0, or an Alternate Collector as an EMS Event Message.

  2. Detected that the MultiBatch entity started but stopped executing because it suffered an abnormal termination. The MultiBatch Log files will contain the reason for the failure.

- **HELD**. The Schedule Server has been requested (by the user) to suspend the next Schedules execution of this entity. (A value of "H" has been placed into the "STATUS" Field of the CREATE/UPDATE Screen for this Schedule Record).
- **EXPIRED**. The next execution of this Schedule Record would be beyond its "PROJECTED LAST RUN" Timestamp, and will no longer be automatically selected for execution by the Calendar Process.

Whilst the Schedule History Screen is displayed, this system will update the Screen every 30 seconds to reflect any change to the Schedule Records status.

If you specify a CALENDAR FILE as part of the definition of a Schedule Record, you cannot specify any of: "ON DAY", "ON DATE", "NOT DAY", "NOT DATE", "UNTIL FOREVER", "UNTIL DAY" or "UNTIL DATE".

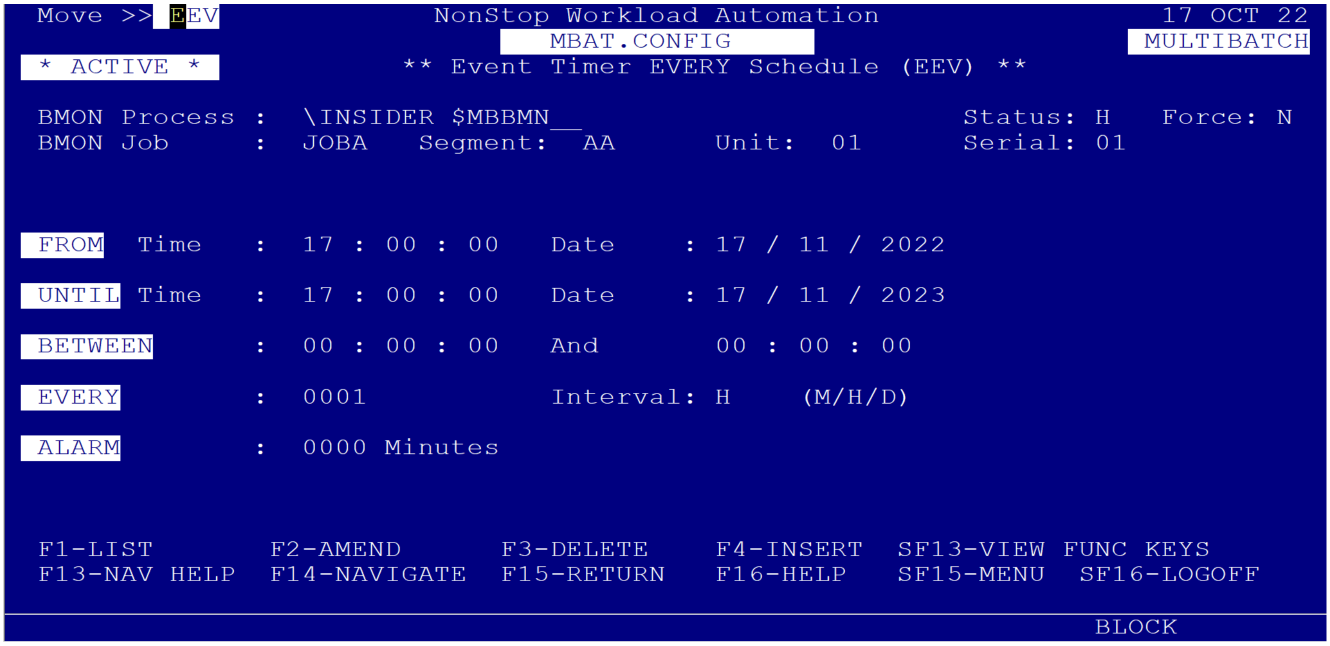
"ALARM". This Field will accept a value of between 1 and 9999 minutes.

After the Calendar process initiates an Event to signal execution of the JOB/SEGMENT/UNIT, it will completed within the ALARM period.

The "ACTION NEEDED" Event will be displayed at the VIEWPOINT Console in Reverse Video; it will be cancelled by the Trigger process issued an "ACTION COMPLETION" Event once it detects that the entity has completed processing.

8.4 Create/Update EVERY Schedule (EEV)

Selecting the **F2** function key from the Main Menu takes the user to the Create/Update MultiBatch Every Schedule Screen (Figure 8.3). This provides facilities for every schedule definition of maintenance.



```
 Move  >> EEV             NonStop Workload Automation                 17 OCT 22
                               MBAT.CONFIG                           MULTIBATCH
 * ACTIVE *             ** Event Timer EVERY Schedule (EEV) **

 BMON Process  :   \INSIDER $MBBMN__                      Status: H    Force: N
 BMON Job      :   JOBA    Segment:__AA       Unit:  01   Serial: 01


 FROM   Time   :   17 : 00 : 00    Date     : 17 / 11 / 2022

 UNTIL  Time   :   17 : 00 : 00    Date     : 17 / 11 / 2023

 BETWEEN       :   00 : 00 : 00    And        00 : 00 : 00

 EVERY         :   0001            Interval: H     (M/H/D)

 ALARM         :   0000 Minutes



 F1-LIST       F2-AMEND      F3-DELETE     F4-INSERT    SF13-VIEW FUNC KEYS
 F13-NAV HELP  F14-NAVIGATE  F15-RETURN    F16-HELP     SF15-MENU  SF16-LOGOFF

                                                                 BLOCK
```

Figure 8.3 – Create/Update MultiBatch Every Schedule (EEV) Screen

If you choose to run a job through the EVERY schedule, then it is the responsibility of the user to reload the BMON environment between runs. Alternatively, a re-runnable JOB configuration can be used, using GROUPID.

The first two lines of fields beginning with BMON Process are common with the Create/Update MultiBatch 'AT' Schedule Screen (EAT) screen. See that screen above for details on these fields.

The other fields provided are as follows:

"**FROM Time/Date**". This specifies the (24-hour clock) time and date at which the first execution of the specified unit will take place.

"**UNTIL Time/Date**". This specifies the (24-hour clock) time and date at which execution of the unit will expire.

"**BETWEEN**","**AND**". These fields allow you to specify the timeframe during which the JOB will run each day within the period specified in the "FROM/UNTIL" fields.

"**EVERY**", "**Interval**". These fields express the time interval between successive runs of the unit. The interval may be one of:

M - MINUTES

H - HOURS

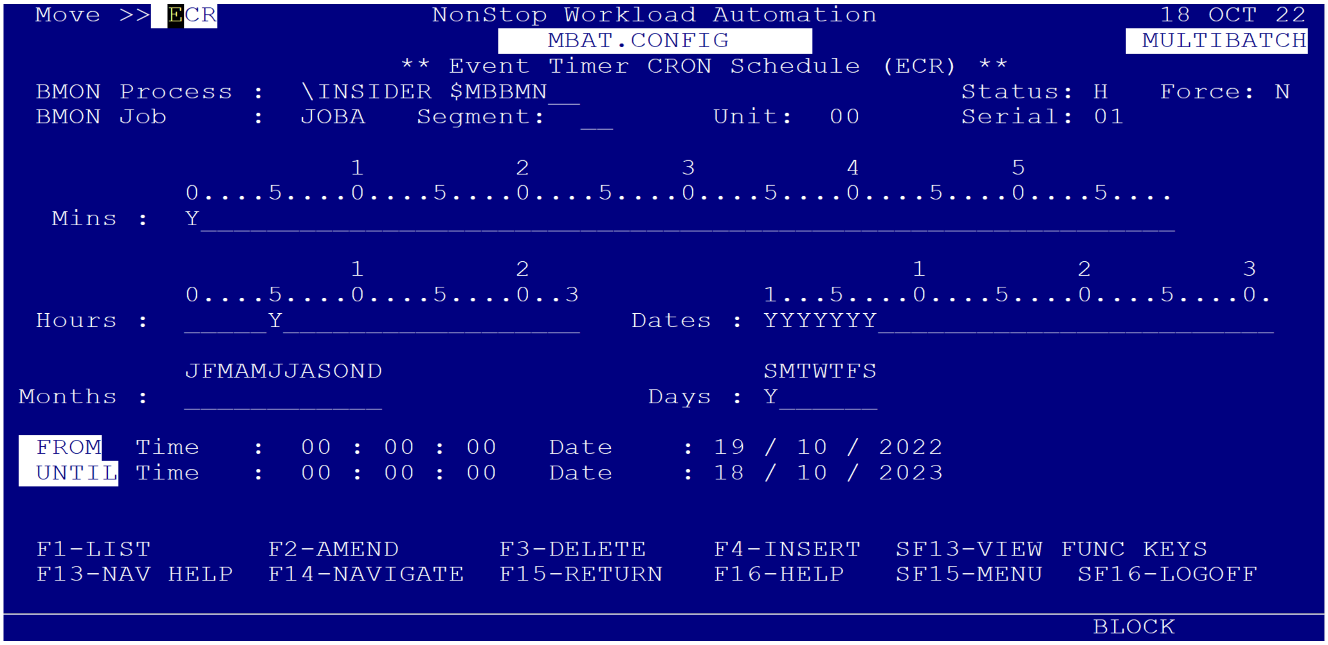
D - DAYS

"**ALARM**". This field is the same as the alarm field on the AT Schedule screen.

As with the previous menu, the Schedule History Screen (Figure 8.7) can be accessed by using the **SF5** function key.

8.5 Create/Update CRONTAB Schedule (ECR)

Selecting the **F3** function key from the Main Menu takes the user to the Create/Update MultiBatch CRONTAB Schedule Screen (Figure 8.4). This provides facilities for CRONTAB schedule definition of maintenance.



```
 Move >> ECR                   NonStop Workload Automation                 18 OCT 22
                                    MBAT.CONFIG                            MULTIBATCH
                            ** Event Timer CRON Schedule (ECR) **
 BMON Process :  \INSIDER $MBBMN__                          Status: H    Force: N
 BMON Job     :  JOBA    Segment: __      Unit:  00         Serial: 01

                         1         2         3         4         5
                0....5....0....5....0....5....0....5....0....5....0....5....
  Mins :        Y___________________________________________________________

                         1         2                                1         2         3
                0....5....0....5....0..3                   1...5....0....5....0....5....0.
 Hours :        _____Y__________________        Dates :    YYYYYYY________________________

                JFMAMJJASOND                               SMTWTFS
Months :        ____________                     Days :    Y______

FROM  Time    :  00 : 00 : 00    Date     : 19 / 10 / 2022
UNTIL Time    :  00 : 00 : 00    Date     : 18 / 10 / 2023


 F1-LIST        F2-AMEND       F3-DELETE     F4-INSERT   SF13-VIEW FUNC KEYS
 F13-NAV HELP   F14-NAVIGATE   F15-RETURN    F16-HELP    SF15-MENU  SF16-LOGOFF

                                                                          BLOCK
```

Figure 8.4 – Create/Update MultiBatch CRONTAB Schedule (ECR) Screen

If you choose to run a job through the CRONTAB schedule, then it is the responsibility of the user to reload the BMON environment between runs. Alternatively, a re-runnable JOB configuration can be used.

The first two lines of fields beginning with BMON Process are common with the Create/Update MultiBatch 'AT' Schedule Screen (EAT) screen. See that screen above for details on these fields.

The **Mins/Hours/Dates/Months/Days** fields specify when you want the unit to run.

Examples:

- If you specify a "Y" at the 10 min marker, then the JOB/SEG/UNIT will run every 10 mins.
- If you specify a "Y" at the 10 min marker and at the M (Monday) Days field, then the JOB/SEG/UNIT will run every 10 mins, every Monday.
- For a JOB/SEG/UNIT to run twice a day at 05:00 and 17:00 hours, every Sunday & Monday, put a "Y" against the 5th & 17th hours, a "Y" against the 0 Minute field and a "Y" on the first two Days fields.
- For a JOB/SEG/UNIT to run on the 1st Sunday of the month at 5am, put Y markers against the, zero Minute field, 5th Hour, 1st Day field and the first 7 Dates fields.

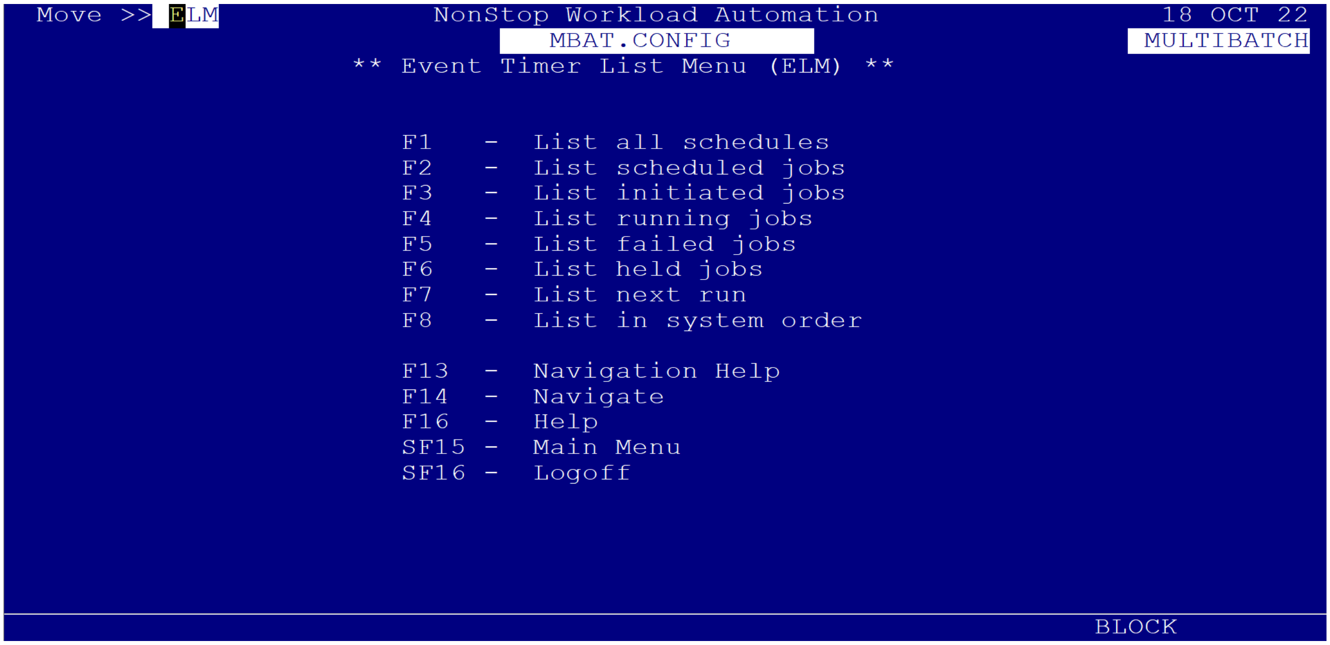
"**FROM Time/Date**". This specifies the (24-hour clock) time and date after which the first execution of the specified unit will take place.

"**UNTIL Time/Date**". This specifies the (24-hour clock) time and date at which execution of the unit will expire.

As described throughout the Event Timer chapter, the Schedule History Screen (Figure 8.7) can be accessed by using the **SF5** function key.

8.6 List Schedule

The List Schedules Screen (Figure 8.5) is accessed via the **F4** function key from the Event Timer Menu (EME) (Figure 8.1).

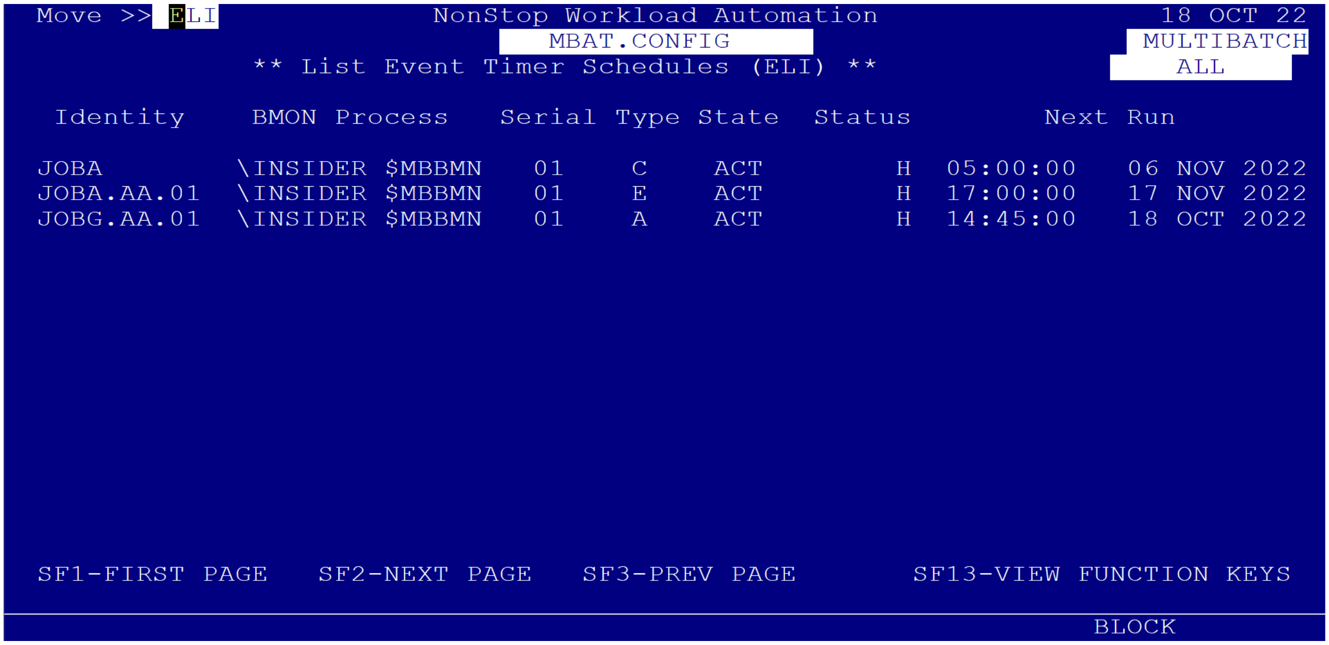


```
 Move >> ELM              NonStop Workload Automation                 18 OCT 22
                                MBAT.CONFIG                         MULTIBATCH
                    ** Event Timer List Menu (ELM) **


                          F1   -  List all schedules
                          F2   -  List scheduled jobs
                          F3   -  List initiated jobs
                          F4   -  List running jobs
                          F5   -  List failed jobs
                          F6   -  List held jobs
                          F7   -  List next run
                          F8   -  List in system order

                          F13  -  Navigation Help
                          F14  -  Navigate
                          F16  -  Help
                          SF15 -  Main Menu
                          SF16 -  Logoff




                                                                  BLOCK
```

Figure 8.5 – List Schedules Screen

This screen allows the user to select a set of list types for Schedule Records in the Database. All the "LIST" screen types have a similar layout with the legend "ALL SCHEDULED", "ALL RUNNING" etc. displayed in the top.

Figure 8.6 shows an example of the List All MultiBatch Schedules Screen.



```
 Move >> ELI              NonStop Workload Automation                 18 OCT 22
                                MBAT.CONFIG                         MULTIBATCH
               ** List Event Timer Schedules (ELI) **                  ALL

  Identity     BMON Process     Serial Type State  Status        Next Run

 JOBA         \INSIDER $MBBMN    01     C    ACT         H  05:00:00  06 NOV 2022
 JOBA.AA.01   \INSIDER $MBBMN    01     E    ACT         H  17:00:00  17 NOV 2022
 JOBG.AA.01   \INSIDER $MBBMN    01     A    ACT         H  14:45:00  18 OCT 2022











 SF1-FIRST PAGE   SF2-NEXT PAGE   SF3-PREV PAGE      SF13-VIEW FUNCTION KEYS

                                                                  BLOCK
```

Figure 8.6 – List All MultiBatch Schedules (ELI) Screen

All LIST Screens are very similar, being designed to display the selected subset of the total Schedule Database. The Screen is divided into a number of columns as follows:

"**IDENTITY**". This field is an aggregate of the three fields "BMON JOB", "SEGMENT" and "UNIT" from the "CREATE/ AMEND SCHEDULE" Screen.

"IDENTITY".

"**SERIAL**". This has the same meaning as the "SERIAL" field from the "CREATE/AMEND" Screen.

"**TYPE**". This field details the type of schedule record in display, and may contain one of two values:

- "**A**" indicating that the schedule record is an AT schedule;
- "**E**" indicating that the schedule record is an EVERY schedule.
- "**C**" indicating that the schedule record is a CRONTAB schedule

"**STATE**". This field reflects this Records Database state, and will contain:

- "**ACT**". This means that this Record is ACTIVE, and still has future execution Timestamps available.
- **"EXP**". This only appears on the "LIST ALL" Screen and means that this record has EXPIRED and has no further execution Timestamps available within its configured lifespan.

"**STATUS**". This field has the same possible range of values as the "STATUS" field of the "CREATE/AMEND" Screen.

- S Scheduled
- I Initiated
- R Running
- X Failed
- H Held

These values will be displayed at their own offset within the confines of this Field, so the user can easily distinguish between the various states.

Any entity which has Failed, or which is RUNNING but has an Alarm outstanding will be displayed in Inverse Video.

This Field will be blank if "STATE" is "EXPIRED".

"**NEXT RUN**". The Timestamp at which the Calendar Process will initiate the next automatic execution of this entity. This field will be blank if "STATE" is "EXPIRED".

Function keys are provided to allow paging through the list of Schedule Records.

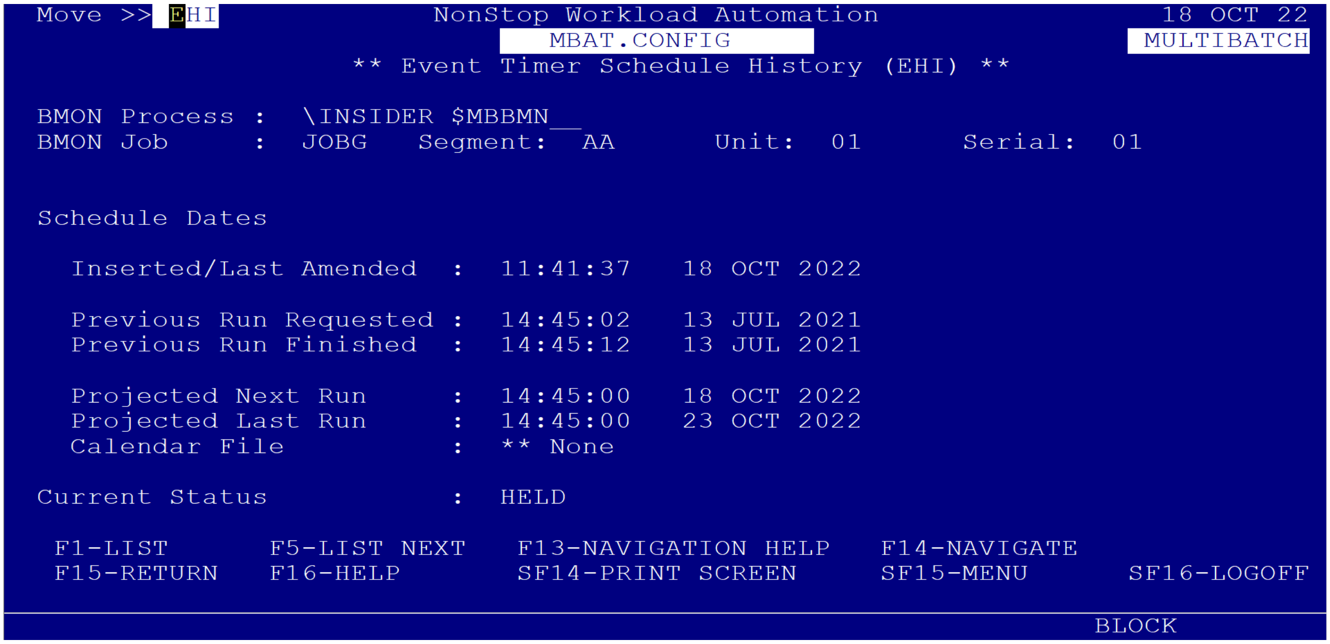
The user is able to navigate to the "CREATE/AMEND" or "HISTORY" Screens for any of the displayed items from any LIST Screen, by placing any character as a marker to the left of a Screen line and pressing the appropriate Function Key **F8**.

The "**LIST INITIATED JOBS**" Screen is not expected to contain entries, as the "INITIATED" State should normally occupy the short period of time between the Calendar Process firing an EMS Event and the Trigger Process updating the Database to reflect either a "RUNNING" or "FAILED" state.

It is however possible to obtain entries in the "INITIATED" Screen list if the Trigger Process has a number of such Events to Process, and is also waiting for MultiBatch to service an execution request.

8.7 Schedule History

Pressing the **SF5**-HISTORY function key from any of the three Schedule set up screens displays the Schedule History Screen (Figure 8.7).



```
 Move >> EHI               NonStop Workload Automation                    18 OCT 22
                                  MBAT.CONFIG                            MULTIBATCH
                        ** Event Timer Schedule History (EHI) **

 BMON Process  :   \INSIDER $MBBMN__
 BMON Job      :   JOBG   Segment:  AA      Unit:  01      Serial:  01


 Schedule Dates

   Inserted/Last Amended  :  11:41:37   18 OCT 2022

   Previous Run Requested :  14:45:02   13 JUL 2021
   Previous Run Finished  :  14:45:12   13 JUL 2021

   Projected Next Run     :  14:45:00   18 OCT 2022
   Projected Last Run     :  14:45:00   23 OCT 2022
   Calendar File          :  ** None

 Current Status           :  HELD

  F1-LIST      F5-LIST NEXT   F13-NAVIGATION HELP   F14-NAVIGATE
  F15-RETURN   F16-HELP       SF14-PRINT SCREEN     SF15-MENU          SF16-LOGOFF

                                                                     BLOCK
```

Figure 8.7 – Schedule History (EHI) Screen

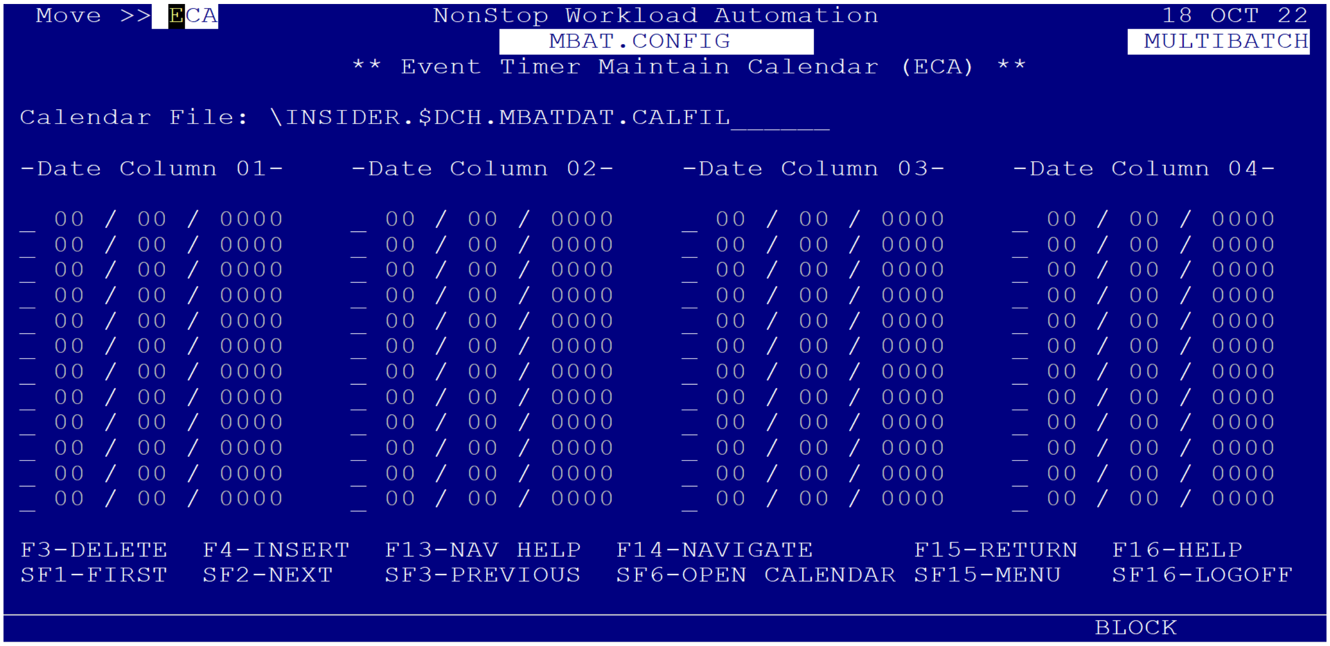
This Screen displays information from the Schedule Database Record for the selected Job.

8.8 Create Calendar File (ECA)

A Calendar File is a structured file which contains a list of dates determined (by the user) as belonging to a logical "set".

This facility allows the user to build and maintain any number of Calendar Files. It is accessed from the Main EME Menu (Figure 8.1) by pressing the **F5** function key when the user is presented with the Calendar File Maintenance Screen. Initially, this screen displays no data and the user **MUST** know the name of and enter the name of the Calendar File and press the **SF6** function key to open the file ready for processing. If the file does not exist or it is an invalid File type an error message is displayed.

Once the Calendar File is successfully opened the first page of entries is displayed on the screen (Figure 8.8). All unused positions are filled with "00/00/0000" and are shown "dimmed".



```
 Move >> ECA               NonStop Workload Automation                    18 OCT 22
                                  MBAT.CONFIG                            MULTIBATCH
                        ** Event Timer Maintain Calendar (ECA) **

Calendar File: \INSIDER.$DCH.MBATDAT.CALFIL______

-Date Column 01-    -Date Column 02-    -Date Column 03-    -Date Column 04-

_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000
_ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000    _ 00 / 00 / 0000

F3-DELETE  F4-INSERT  F13-NAV HELP  F14-NAVIGATE        F15-RETURN  F16-HELP
SF1-FIRST  SF2-NEXT   SF3-PREVIOUS  SF6-OPEN CALENDAR   SF15-MENU   SF16-LOGOFF

                                                                     BLOCK
```

Entries can be made in blocks and in any order; the contents of the Screen will be validated before the entries are allowed to update the Calendar file; once they are successfully written to the Calendar File, the next retrieval will display them in Date order. Existing entries can be left on-screen; if a new Date is to be added to a Calendar File, it should be entered at any free position on the Screen, and any character placed as a marker in the space provided to the left of each Date Field.

Use of the **F4** function key will then cause the system to validate and insert all marked entries into the nominated Calendar File. If existing dates are to be deleted, the procedure is to first display the Screen Page containing that Date or Dates, place a marker against it and use the **F3** Function Key to request Deletion.

If there are several dates to be deleted, spanning a number of Screen pages, then the operation should be performed against each Screen page separately.

The operation of the Calendar File facility is integrated with the main "AT SCHEDULE" Screen for creation of Schedule Records, and the Calendar File has no function other than simply to contain a sequence of Dates which are relevant to your Organization.

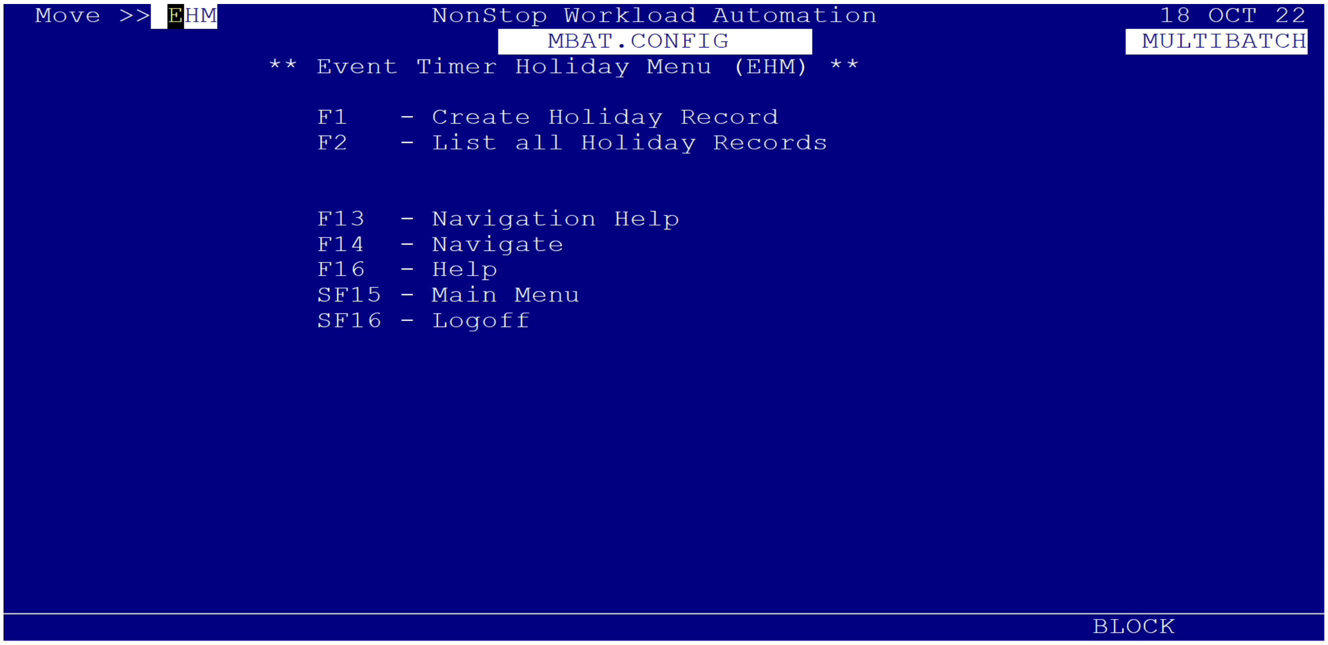
Note: *An example audited Calendar file is supplied within the MultiBatch database, e.g. MBATDAT.CALFIL.*

If you wish to create your own Calendar file, then the file attributes are:

AUDITED TYPE K
REC 6
BLOCK 4096
EXT (4 , 32)
KEYLEN 6

8.9 Holiday File (EHM)

A Holiday File is a structured file which contains a list of dates for which it has been decided that no Jobs are to be scheduled (e.g. Public Holidays when the Organization may be closed). This facility is accessed from the Main EME Menu (Figure 8.1) via the **F6** function key. When accessed, the user is presented with the Maintain Holiday File Menu (Figure 8.9).



```
 Move >> EHM                  NonStop Workload Automation                 18 OCT 22
                                    MBAT.CONFIG                          MULTIBATCH
                   ** Event Timer Holiday Menu (EHM) **

                      F1    - Create Holiday Record
                      F2    - List all Holiday Records


                      F13   - Navigation Help
                      F14   - Navigate
                      F16   - Help
                      SF15  - Main Menu
                      SF16  - Logoff









                                                                        BLOCK
```

Figure 8.9 – Maintain Holiday Menu (EHM) Screen

This presents the user with the following two options:

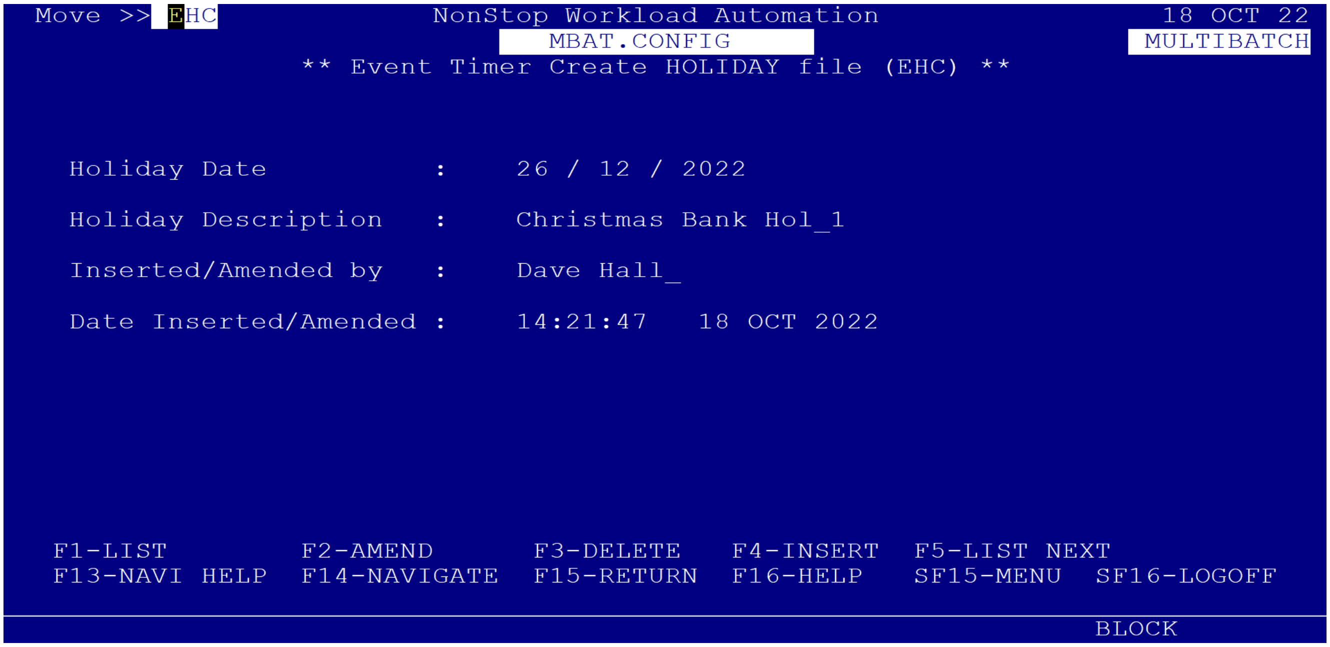
- Create Holiday Record.
- List All Holiday Records.

8.10 Create Holiday File (EHC)

Pressing F1 from the Holiday File (EHM) Screen presents the user with the Create Holiday Records (EHC) Screen (Figure 8.10). The user can enter the following information:

- The Holiday Date when no schedules are to be run.
- A description of the Holiday (for example, Christmas Day).
- An indication as to who inserted/amended the record.

The date of the insertion/amendment is automatically inserted by the Event Timer system.



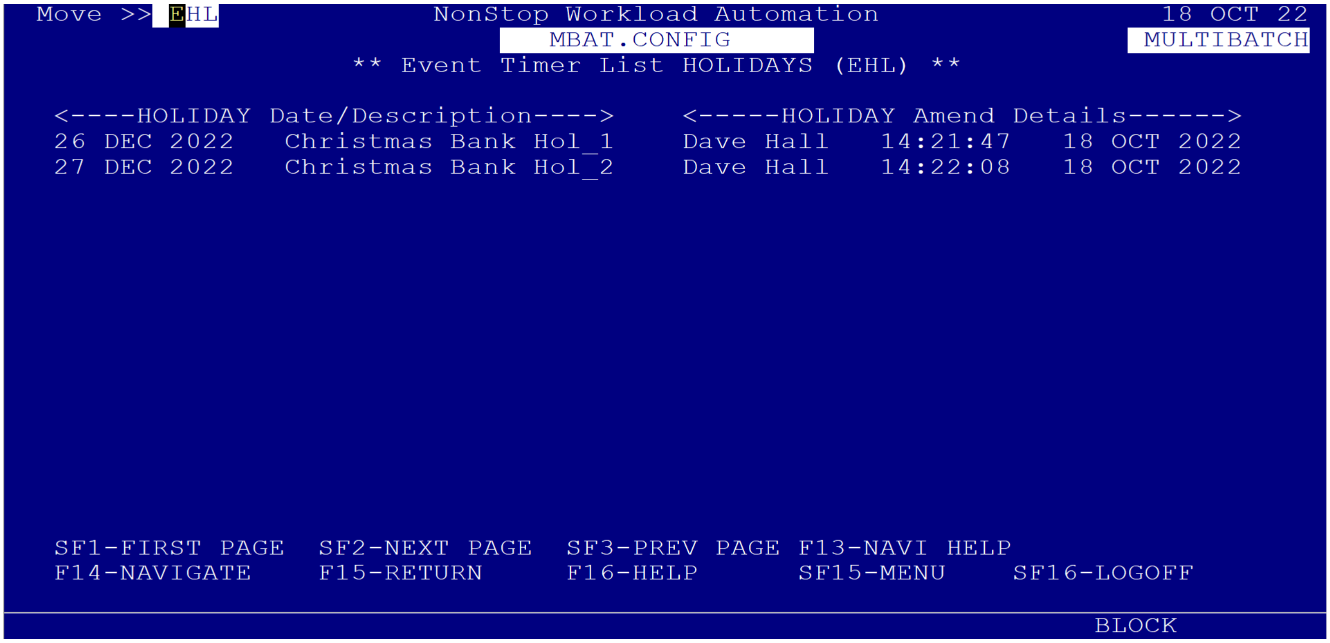
```
 Move >> EHC               NonStop Workload Automation                 18 OCT 22
                                  MBAT.CONFIG                      MULTIBATCH
                    ** Event Timer Create HOLIDAY file (EHC) **


   Holiday Date             :    26 / 12 / 2022

   Holiday Description      :    Christmas Bank Hol_1

   Inserted/Amended by      :    Dave Hall_

   Date Inserted/Amended :       14:21:47   18 OCT 2022






  F1-LIST        F2-AMEND      F3-DELETE    F4-INSERT   F5-LIST NEXT
  F13-NAVI HELP  F14-NAVIGATE  F15-RETURN   F16-HELP    SF15-MENU   SF16-LOGOFF

                                                                  BLOCK
```

Figure 8.10 – Create Holiday Record Screen

8.11 List All Holiday Records

The List All Holiday Records Screen (Figure 8.11) gives the user a list of all Holiday Records currently in the database. The screen is accessed from the Holiday menu screen via the F2 function key.

The user can scroll backwards and forwards through the records; it is not possible to access the Create Holiday Record Screen directly from the List screen.



```
 Move >> EHL               NonStop Workload Automation                 18 OCT 22
                                  MBAT.CONFIG                      MULTIBATCH
                       ** Event Timer List HOLIDAYS (EHL) **

  <----HOLIDAY Date/Description---->    <-----HOLIDAY Amend Details------>
  26 DEC 2022   Christmas Bank Hol_1    Dave Hall   14:21:47   18 OCT 2022
  27 DEC 2022   Christmas Bank Hol_2    Dave Hall   14:22:08   18 OCT 2022











  SF1-FIRST PAGE  SF2-NEXT PAGE  SF3-PREV PAGE F13-NAVI HELP
  F14-NAVIGATE    F15-RETURN     F16-HELP      SF15-MENU     SF16-LOGOFF

                                                                  BLOCK
```

8.12 Response to Failure

This section details the response to Event Timer failures and how the facility can be stopped for a specified period.

CPU Failure Involving Calendar or Trigger Processes

Both of these processes run as NonStop process pairs. Should the CPU containing a Primary or Backup TRIGGER or CALENDAR process fail, the following sequence of actions will take place:

If Primary CPU Fails:

- The backup process will immediately assume responsibility for processing and therefore become the Primary process.

- The Primary process will now start a new Backup process in the highest available CPU on the System, and create an EMS Event noting the action taken (See Appendix E).

If Backup CPU Fails:

- The Primary process will be informed of the failure of the backup process CPU, and will immediately start a new backup process in the highest available CPU on the System.
- An EMS Event will be created noting the action taken (See Appendix E).

If a CPU failure results in there only being a single CPU available, both CALENDAR and TRIGGER will continue to execute as single processes and an EMS Event will be issued.

When another CPU becomes available then a Backup process will be automatically created and another EMS Event will be generated to chronicle that action.

8.13 System Failure

A stoppage in Event Timer can either be due to a system or process failure, or it can be at the user's instigation.

Unplanned Outage

Upon reloading the NonStop system after an unplanned outage, such as a system crash, the sequence of restarting the Event Timer system components is as follows:

1. Start the MultiBatch PATHWAY System **ONLY**.

2. Run PSCHED (see Appendix F: Utilities) against the Database DATACONF file and check spooler results, e.g.:

```
RUN MBATOBJ.PSCHED / IN MBATDAT.DATACONF / ALL
```

3. Recover the BMON against any MultiBatch BMON Process active at the time of the crash. (See Appendix H 'Recovering a BMON' for further information).

4. **DO NOT** START the TRIGGER and CALENDAR processes just yet. There are some vital clerical checks which need to be performed:

Determine whether any Event Timer controlled MultiBatch entities were actually executing at the time.

This can be gleaned either from examination of the MultiBatch log files, through use of EMSDIST, VIEWPOINT, or by studying the output messages from a BMON Recovery session which highlights failure conditions.

Additionally, run the UTRECOV utility on the Recovered BMON which can report on any jobs which have Failed, Completed, or Not Started. (See Appendix H 'Recovering a BMON' and Appendix F.12: 'UTRECOV' utility for further information).

Any that were still running should have either an “R” or “I” status in the Schedule Database. Make a note of the identifier of each.

5. Run the PSCHED program against the DATACONF file; cross reference and verify these potential “R” or “I” status entries.

6. Examine the PSCHED output for “X” entries; determine from the MultiBatch sources whether the “X” entries equate to any “failed to start” or “failed to complete execution” MultiBatch entity.

The decision to rerun/restart/re-attempt any MultiBatch entity in the “R” or “X” state is entirely dependent upon your Organizations requirements for that particular Application program.

Once the Pathway System is restored, “X” and “R” entries can be reset to “S”, at which point their next execution timestamp will be calculated.

7. Any Schedule Record which PSCHED states as being in a “I” state, but is not shown as “incomplete” or “still running” by the MultiBatch sources above, has not resulted in the MultiBatch entity starting processing.

Once the Pathway system has been started, the “I” Database records can be set to “S”, upon which the NEXT execution timestamp will be calculated.

8. If any of the former “I”, “X” or “R” entries are required to be processed before that time/date, then the “FORCE” facility can be used to perform this in a controlled manner for each. Use of FORCE will not prejudice execution at the next scheduled timestamp.

9. For those entries which previously had “X” or “R” status, and which the output from RECOVER BMON disclosed as being incomplete, the FLAGS setting within MultiBatch will prevent FORCE from causing them to execute.

This can be circumvented by either reloading the MultiBatch monitor process concerned, by direct use of RESTART UNIT, or ALTER STEP commands within MultiBatch.

When you have checked and managed all exceptions in the above way, you can now start the CALENDAR and TRIGGER processes.

You should be aware that any entries in the Schedule Database which are set to “**S**” Status, and are past their Scheduled Execution Timestamp, will be immediately selected for execution once the CALENDAR and TRIGGER processes are started.

If you are using the Duplicate Database Facility, refer to the next section for details on how to run the Integrity Checking Programs in the event of a system crash.

Planned Outage

Should the Event Timer System require stopping for any period, this should be performed as follows:

1. Stop the CALENDAR and TRIGGER processes.

2. Shutdown the Pathway System.

3. Shutdown any MultiBatch environment controlled by Event Timer.

4. Run the PSCHED program against the Database DATACONF file.

It may be that the system will be unavailable for only a short time, and that there are no Schedule Records eligible for execution within that timeframe; this will be disclosed by the PSCHED print.

If this is the case, then to start the System again you only need to:

1. Reload any MultiBatch environment to be controlled by Event Timer.

2. Start the CALENDAR and TRIGGER processes.

If the outage is going to be for a period which will span the execution timestamps of several entities, the following sequences should be followed when shutting down the System:

1. Stop the TRIGGER and CALENDAR processes.

2. Run the PSCHED program against the Database DATACONF file.

3. Optionally, use the Pathway System to alter to “**H**” status any records which span the downtime, and which you want to control manually following reload.

4. Shutdown the Pathway System.

5. Optionally, use RESCHED (not PSCHED) to either: Place ALL Schedule Records in a HOLD state.

Use the FROM facility to globally recalculate forward the execution timestamp of all Schedule Records.
Use the HOLDON facility to globally alter all Schedule Records to “**H**” status. Run the PSCHED program again, retaining all generations of the output.

Once the System is available again you should:

1. Start the Pathway System.

2. Start any MultiBatch environments controlled by Event Timer.

3. Run RESCHED as required.

4. Make any manual amendments necessary to control execution of Schedule Records.

5. Start the CALENDAR and TRIGGER processes.

Options for controlling the RESCHED and PSCHED programs have already been discussed in the previous sections.

8.14 Duplicate Database Facility

This facility allows the user to automatically maintain a duplicate copy of the Event Timer audited database files, for purposes of contingency.

The second database files would normally be created on a separate node, but they could exist on the same node as the primary Event Timer database. For the latter, the database files would have to be on another subvolume.

The use of the facility is optional and the placement of the duplicate database is user configurable.

The user must create the duplicate database files as the Event Timer package will not do this automatically. The files must be placed on a disk volume that is enabled for TMF and the files must be **AUDITED**.

The duplicate database files must all reside on the same subvolume.

The requirements are:

1. Copy of the Scheduler files:

SCHEDULE

SCHEDUL0

2. Copy of the Holiday file:

HOLIDAY

3. Copy of Event Timer Calendar files, e.g.:

The PATHWAY server class 'EVENT-TIMER' requires an additional parameter, DUPLICATE-DB.

Its value must be set to the name of the node, volume and subvolume that contains the duplicate database.

```
SET SERVER PARAM DUPLICATE-DB \BACKUP.$DATA.ETMDUP
```

If the DATACONF locations for SCHEDULE and HOLIDAY are \LIVE.$DATA.MBATDAT.SCHEDULE and \LIVE.$DATA.MBATDAT.HOLIDAY, then the duplicate database files will be SCHEDULE and HOLIDAY on a suitable backup node, volume, subvolume, e.g. \BACKUP.$DATA.ETMDUP.

The Calendar and Trigger processes only update the SCHEDULE file, and not the HOLIDAY files.

A run time parameter indicates the placement of the duplicate SCHEDULE file is required as follows:

```
RUN TRIGGER/IN <Database Svol>.DATACONF, NAME $<Trigger Process Name>,NOWAIT, &
OUT $<Calendar Process Name>/<PARAMS>

RUN CALENDAR/IN <Database Svol>.DATACONF, NAME $<Calendar Process Name>, &
NOWAIT/<PARAMS>
```

where <params> are:

BACKUP^CPU <cpu-spec>

DUPLICATE^DB <node-vol-svol-schedule filename>.

The position of the parameters is fixed. If you are using the default value of BACKUP^CPU and omit this parameter, a comma "," is required before the DUPLICATE^DB parameter.

For example:

```
RUN TRIGGER/IN <Database Svol>.DATACONF, NAME $<Trigger Process Name>,NOWAIT, &
OUT $<Calendar Process Name>/,DUPLICATE^DB \BACKUP.$DATA.ETMDUP.SCHEDULE

RUN CALENDAR/IN <Database Svol>.DATACONF, NAME $<Calendar Process Name>, &
NOWAIT/, DUPLICATE^DB \BACKUP.$DATA.ETMDUP.SCHEDULE
```

8.15 Maintaining Database Integrity

Any PATHWAY transaction that updates both databases will only be successful when ALL updates are completed. For example: if the duplicate database GUARDIAN security is altered to restrict access to the PATHWAY, any subsequent PATHWAY transactions will:

- Update the Primary database.
- Fail on the update of the Duplicate database.
- The transaction will be aborted and the primary updates will be backed out. An error message will appear on line 24 of the current PATHWAY screen.

In these circumstances, updates to the primary database by the Event Timer Pathway system will only be permitted if the original problem is fixed, or the Event Timer server class parameter DUPLICATE-DB is removed.

Any failure to update the duplicate database by the Calendar and Trigger processes will NOT lead to the delay of potentially crucial processing. The amendments to the primary database will ALWAYS be applied, the failure of the duplicate database amendment will be reported, and processing will continue normally. Subsequent amendments to the duplicate database will be attempted but because of previous failures, the results may be unpredictable.

Database integrity can be restored without closing down the Event Timer application. A number of utilities are provided that helps to compare the contents of the primary and duplicate databases and bring the two sets of files into step.

database files. Therefore, if duplicate^db functionality is required, add a new DCC record called DUP^DB with the location of the duplicate Event Timer database files.

Note *The "Integrity" utilities may fail if the SCHEDULE, SCHEDUL0, HOLIDAY and all the Calendar files are not audited.*

1. **SCHEDULE** file.

This has the following syntax:

RUN INTEGSHD / IN <Database Svol>.DATACONF /

This program also checks any calendar files referenced in the primary schedule file. For example, if a primary schedule file record references a calendar file….

\LIVE.$D01.MBATDAT.LASTFRI

and the duplicate schedule file is on subvolume:

\BACKUP.$D03.MBATDUP

then the INTEGSHD opens the file \BACKUP.$D03.MBATDUP.LASTFRI and checks that it is in step with..

\LIVE. $D01.MBATDAT.LASTFRI.

2. **HOLIDAY file.**

This has the following syntax:

RUN INTEGHOL / IN <Database Svol>.DATACONF /

8.16 Restarting Applications on a Contingency Node

To restart the Event Timer application on another node, complete the following steps:

1. Close down the primary PATHWAY system.

2. Stop the primary Calendar process.

3. Stop the primary Trigger process.

If the primary database is out of step with the duplicate database, run the INTEGSHD and INTEGHOL processes as applicable.

The duplicate schedule file may contain references to Calendar files resident on the primary system. If this is not appropriate they can be changed clerically through the PATHWAY system.

Alternatively run the INTEGCAL program from the contingency node to perform this task:

RUN INTEGCAL / IN <Database Svol>.DATACONF /

If the duplicate schedule file lives on:

\LIVE.$D03.ETMDUP.SCHEDULE

and the Schedule records point at Calendar files on:

\BACKUP.$D01.ETMDUP.CALFIL

then the Schedule file will be updated to reference \LIVE.$D03.ETMDUP.CALFIL content.

Start the application on the Contingency node using the obey files PWLOAD (PATHWAY) and CALTRIG (Calendar and Trigger processes).

Chapter 9 On Demand Jobs

9.1 Introduction

In addition to providing functionality for schedules that are run at regular times and intervals MultiBatch also provides support for on demand processing. This functionality allows a user to run, for example, a set of reports, on an ad hoc basis.

Model Jobs are set up in base data, these jobs are then be used as the basis for creating On Demand Jobs in BCOM. On Demand Job are specifically associated with the BCOM user that created them. On Demand Jobs can contain multiple dependent segments and are in effect mini schedules of processing that can be repeated with altered parameters whenever needed.

After an On Demand Job is created from a Model Job it is then possible to modify the On Demand Job. A user may wish to change, for example, the file from which the On Demand Job will generate a compiled object.

On Demand Jobs require the use of the BMON recovery file, see Maintain BMON Configuration (CBM) in Chapter 3.

An On Demand Job always runs under the user id of the BCOM from where it is submitted. For this reason a licensed version of BMON, usually BMONL, must be run to execute the On Demand job functionality.

9.2 Creating Model Jobs

To use On Demand Jobs it is first necessary to set up Model Jobs, see Chapter 3 Configure MultiBatch. On the Configure JOB (CJB) you can set the Model Job flag to signify a Model Job. Model Jobs are restricted as follows when setting up base data:

1. They cannot have previous dependencies.
2. Other jobs cannot have Model Jobs as previous dependencies.
3. Segment dependencies cannot refer to jobs outside of the Model Job.
4. They cannot have a start time.
5. They cannot have a group id.
6. They cannot be included in Event Timer processing.
7. They cannot be included in Clockmon processing.
8. Jobs must be scheduled “DAILY” and “I” (Include).
9. Segments and Units cannot be scheduled.
10. Steps cannot have IF FAILED jobs.
11. Steps cannot define a User Id.

The standard screens in Chapter 3 Configure MultiBatch are used to create the model segments, units and steps.

Model Jobs are the basis for On Demand Jobs. When an On Demand Job is created the specified Model Job is copied and a new job is created specifically for use by the BCOM user that requested it.

Model Jobs cannot be created using BCOM.

Model Jobs cannot be altered or started from BCOM.

Model Jobs do not appear in the Status Monitoring database.

9.2 Using On Demand Jobs

On Demand Jobs are identified by four numeric characters, all other jobs within MultiBatch have ids that begin with an alphabetic character. In this way there are 9999 identifiers within a BMON for On Demand Jobs.

When the BCOM ADD FROM MODEL <job> command is used a job is created for the user with an id of the next numeric identifier that is available. The <job> specified in this command must be a Model Job. The On Demand Job created has all the characteristics of Model Job except for the job name. Segment dependencies are modified to reference the new On Demand Job name,

You should be aware that On Demand job identifiers are reused on cyclical basis, identifiers are only reused where the processing for previous job that used the identifier is complete.

ALTER command. See Appendix A BCOM Syntax for the Step Commands:

1. ALTER STEP <xxxx.yy.zz[.n]> <commands…>
2. (n) ALTER OSS STEP <xxxx.yy.zz[.n]> <commands…>

Only a BCOM running under the user id that executed the ADD FROM MODEL <job> command can issue ALTER commands for the job created.

An On Demand job is submitted for execution using the SUBMIT JOB command.

If the SUBMIT JOB command is not executed and the user exits BCOM or issues another ADD FROM MODEL <job> then the job is removed and is no longer available.

The INFO JOB <job> and STATUS JOB <job> commands can be used to retrieve details of On Demand Jobs.

The STATUS JOB *,ONDEMAND lists all the current running On Demand Jobs.

9.3 Setting the number of On Demand Jobs

The Number On Demand field on the Maintain BMON Configuration (CBM) screen defines the expected number of concurrent On Demand Jobs for a BMON. By setting this number BMON is able to allocate resources to manage the On Demand Jobs.

It is typically expected that a specific BMON or number of BMONs are set up specifically to handle On Demand Jobs. In this scenario the Number On Demand field can be set to its maximum value of 999.

The resources available to a running BMON are displayed using the BCOM INFO BMON command. By monitoring the Maximum and Actual figures it is possible to gain insight into the number of on demand jobs that need to be allocated.

9.3 Security

Those users who are to use On Demand jobs should have access to both security classes as defined on the Maintain BMON Configuration (CBM) screen. Also note, to ABORT, ALTER or RESTART a step that is part of an On Demand Job access to the second security class is required.

Although On Demand jobs can be run within any BMON, consideration should be given to dedicating a specific BMON to On Demand Jobs. In this way it may be left to run continuously.

Appendix A BCOM Syntax

A.1 Introduction

The MultiBatch Pathway system is designed to make the scheduling and running of Jobs as user friendly as possible. This involves cutting down the amount of detail the user has to input from the command prompt.

Appendix A provides the user with a complete list of the BCOM syntax, required if the MultiBatch Pathway environment did not exist.

All the facilities provided by MultiBatch can be replicated using the BCOM syntax. In addition, it provides some facilities which are not available via the Pathway screens.

The Appendix comprises a list of the BCOM commands followed by a description of the commands.

A.2 Overview

BCOM / BMON COMMANDS

```
CHECKPOINT BMON [, BACKUP^CPU n]
```

Note: CHECKPNT can be used as an abbreviated form

```
INFO BCOM
INFO BMON
LOG1 <file-name>
LOG2 <file-name>
OBEY <file-name>
OPEN BMON <process-name>
RESET ALL
BMON DELIMITERS
SET BCOM
DELIMITERS <character>
GLOBALPARAMS <file-name>
INTERACTIVE <ON | OFF>
LISTING {ON} {OFF}
LOCALPARAMS <file-name>
PAGING <ON | OFF>
SHELLNAME <value>
UPSHIFT {ON} {OFF}
SET BMON
CLOSEDOWN {ON} {OFF}
CHECKSTART {ON} {OFF}
MAXASSIGNS <n>
MAXBCOMS <n>
MAXDEFINES <n>
MAXDEPENDS <n>
MAXENVS <n>
MAXFDS <n>
MAXJOBS <n>
MAXPARAMS <n>
MAXSEGS <n>
MAXUNITS <n>
NEWRECOVER <file-name>
OWNER <m,n>
PRIV <m,n>
SECURITY <m,n>
SHOW ALL
BMON SHUTDOWN BMON [!]
START BMON
STATUS BMON
SWITCH BMON
SYNC IN <replicated-recovery-file-name> [!], TRANSLATE <rules-file-name>
SYNC OUT <newrecover-file-name>
```

JOB COMMANDS

```
ADD FROM MODEL <xxxx>
ADD JOB <xxxx>
DELETE JOB <xxxx>
HOLD JOB <xxxx>
INFO JOB    <*>
            Prefix<*>
            <xxxx> [, DETAIL]
RELEASE JOB <xxxx>
RESET JOB <xxxx>
SET JOB     CRITICAL {ON} {OFF}
            GROUPID <group-identifier>
            MAXSEGS <n>
            PREV    <xxxx>
            RERUN {ON} {OFF}
            STARTTIME <hh:mm>
SHOW JOB
START JOB <xxxx>
STATUS JOB <*> [, <Qualifier>]
             Prefix<*>
             <xxxx> [, Detail]

where <Qualifier> is one of:
                         Aborted       Complete     Failed
                         Ondemand      Protocol     Running
                         Stopped       Twostep      Waiting

SUBMIT JOB
```

SEGMENT COMMANDS

```
ADD SEGMENT <xxxx.yy>
DELETE SEGMENT <xxxx.yy>
HOLD SEGMENT <xxxx.yy>
INFO SEGMENT <xxxx.yy> [, DETAIL]
RELEASE SEGMENT <xxxx.yy>
RESET SEGMENT
SET SEGMENT   CRITICAL {ON} {OFF}
              MAXUNITS <n>
              MAXDEPENDS <n>
              PREV <xxxx.yy>
SHOW SEGMENT
START SEGMENT <xxxx.yy>
STATUS SEGMENT <xxxx.nn> [, DETAIL]
```

UNIT COMMANDS

```
ABORT UNIT         <xxxx.yy.zz>
DELETE UNIT        <xxxx.yy.zz>
HOLD UNIT          <xxxx.yy.zz>
INFO UNIT          <xxxx.yy.zz> [, DETAIL]
INFO UNIT          <xxxx.yy.zz> , ASSIGNS
INFO UNIT          <xxxx.yy.zz> , DEFINES
RELEASE UNIT       <xxxx.yy.zz>
RESTART UNIT       <xxxx.yy.zz>
START UNIT         <xxxx.yy.zz>
STATUS UNIT        <xxxx.yy.zz> [, DETAIL]
```

STEP COMMANDS

```
ADD STEP <xxxx.yy.zz[.n]>
ADD STEP DEFINE <=defname>
```

Defines are added to set context and can be seen using SHOW STEP.

```
ADD OSS-STEP <xxxx.yy.zz[.n]>
ALTER STEP <xxxx.yy.zz[.n]> <commands…>
```

Where commands are:

```
            ADD DEFINE <=defname>
            ASSIGN <log>, <create-open-specs>
                <log> can identify an existing STEP assign to be amended.
                For <create-open-specs>, see SET STEP command below.
            CPU m, n
            DEFAULT <file-name>
            DEFINE <=defname> <attrib> <value>
            DELETE ASSIGN <log>
            DELETE DEFINE <=defname>
            DELETE PARAM <name>
            FLAGS <%nnnnnn>
            HIGHPIN {ON} {OFF}
            INFILE <file-name>
            LIBRARY <file-name>
            MEM n
            NAME <$name>
            NOMAXTIME
            OBJECT <file-name>
            OUTFILE <file-name>
            PARAM <name>,"value"
                <name> can identify an existing PARAM to be amended
            PRI n
            PROTOCOL {ON} {OFF}
            RUNOFF
            RUNON
            RUNPARAMS <"...">
            SWAP <file-name>
            TERM <file-name>
            TEXT <"...">
```

```
(n) ALTER OSS STEP <xxxx.yy.zz[.n]> <commands…>
```

(n) is used for OSS-ENV, OSS-FD, DELETE OSS-ENV and DELETE OSS-FD. It is used to identify the entry as per the number shown using INFO UNIT <unit name>, DETAIL

Where commands are:

```
            OSS-PROGRAM <program-name>
            OSS-PATHNAME <pathname>
            OSS-ARGUMENTS <arguments> [, <arguments>]
            OSS-FDCWD <current^working^directory>
            OSS-ENV <env details>
            OSS-FD <fd>, <dupfd>, <name>, <access flag>, <mode>
            DELETE OSS-ENV
            DELETE OSS-FD
```

```
RESET ASSIGN
RESET GUARDIAN-STEP
RESET OSS-ENV
RESET OSS-FD
RESET OSS-STEP
RESET PARAM
RESET STEP
SET STEP <commands…>
```

Where commands are:

ASSIGN <log>, <phys>, <create-open-specs>
create-open-specs are:

CODE <file-code>
REC <record-size>
BLOCK <block-size>

One of:
SHARED
PROTECTED
EXCLUSIVE

One of:
I-O
INPUT
OUTPUT

One of:
EXT n
EXT (n,n)

See the TACL manual for a detailed description.

To amend an assign in SET context, use the same <log> value with amended <phys> and <create-open-specs>.

To remove an assign from SET context use:

ASSIGN <log>

That is use <log> without further parameters. <log> is a logical name previously set up using SET STEP ASSIGN.

CPU <x,y>
CRITICAL {ON} {OFF}
DEFAULT <vol.subvol>
DEFINE <attribute>, <value>

Attribute initially CLASS with value as one of:
TAPECATALOG
TAPE
DEFAULTS
CATALOG
SORT
SUBSORT
MAP
SPOOL

After setting CLASS attribute values can be set as follows. See the TACL reference manual for definition of possible values.

To see the set values, use SHOW DEFINE.

To add the define to SET context using the DEFINE context use ADD STEP DEFINE <define name>.

To remove a define from SET context use DELETE STEP DEFINE <define name>.

For TAPECATALOG attribute can be:
AUTOMATED
AVRSYSTEM
BLOCKLEN

COMMENT
COMPRESSION
DENSITY
DEVICE
EBCDIC
EXPIRATION
FILECAT
FILEID
FILESECT
FILESEQ
GEN
LABELS
LOGICAL
MOUNTMSG
OWNER
PHYSICAL
POOL
RECFORM
RECLEN
RETENTION
TAPEMODE
USE
VERSION
VOLCAT
VOLUME

For TAPE attribute can be:
BLOCKLEN
DENSITY
DEVICE
EBCDIC
EXPIRATION
FILEID
FILESECT
FILESEQ
GEN
LABELS
MOUNTMSG
OWNER
RECFORM
RECLEN
REELS
RETENTION
SYSTEM
TAPEMODE
USE
VERSION
VOLUME

For DEFAULTS attribute can be:
CATALOG
SWAP
VOLUME

For CATALOG attribute can be:
SUBVOL

For SORT attribute can be:
BLOCK
CPU
CPUS
MODE

PRI
PROGRAM
SCRATCH
SEGMENT
SUBSORTS
SWAP

For SUBSORT attribute can be:
CPU
PRI
PROGRAM
SCRATCH
SEGMENT
SWAP

For MAP attribute can be:
FILE

For SPOOL attribute can be:
COPIES
FORM
HOLD
HOLDAFTER
LOC
OWNER
SELPRI
REPORT
BATCHNAME
MAXPRINTLINES
MAXPRINTPAGES

HIGHPIN {ON} {OFF}
IFFAIL <XXXX>
INFILE <file-name>
LIBRARY <file-name>
LIKE <xxxx.yy.zz[.n]>
MAXTIME <hh:mm>
MEM <n>
NAME <process-name>
OBJECT <file-name>
OUTFILE <file-name>
PARAM <name>, "param value"

To remove a parameter from SET context, use PARAM <name> without a value.

To change a param value restate PARAM <name>, "param value" with the new param value.

PRI <n>
PROTOCOL {ON} {OFF}
RUN [NOT] <param name>
RUNPARAMS "<params>"
SWAP <file-name>
TERM <term-name>
TEXT "<text>"
TYPE {OSS} {GUARDIAN}

SET OSS-STEP <commands…>
Where commands are:
OSS-PROGRAM <program-name>

If program-name does not contain / then OSS-PATHNAME will be used to qualify the value.

OSS-PATHNAME <pathname>
OSS-ARGUMENTS

<arguments> is a space separated list.

OSS-ENV <environment-parameter>
<environment-parameters> is expected to be a name value pair in the form name=value.

Values added to the OSS-STEP context are given a number, seen when using SHOW OSS-STEP.

To amend an environment-parameter use (<number>) SET OSS-ENV "<environment-parameters>".

To remove an environment-parameter from OSS-SET context, use just (<number>) SET OSS-STEP OSS-ENV.

OSS-FDCWD <current-working-directory>
OSS-FD <fd>, <dupfd>, <name>, <access flag>, <mode>

Values added to the OSS-FD context are given a number, seen when using SHOW OSS-STEP.

To amend an FD use (<number>) SET OSS-STEP OSS-FD <fd>, <dupfd>, <name>, <access flag>, <mode>

To remove an FD from OSS SET context, use just (<number>) SET OSS-STEP OSS-FD.

OSS-FLAG <flag> ON/OFF

flag is one of CREAT, EXCL, NOCTTY, TRUNC, APPEND, NOBLOCK or SYNC

These flags are used by subsequent SET OSS-FD, ALTER OSS-STEP <xxxx.yy.zz[.n]> OSS-FD or ALTER OSS-STEP <xxxx.yy.zz[.n]> ADD OSS-FD

SHOW STEP

BMON RECOVERY COMMANDS

SYNC IN <replicated-recovery-file-name> [!], TRANSLATE <rules-file-name>
SYNC OUT [<newrecover-file-name>]

MISCELLANEOUS COMMANDS

COMMENT <text>
EXIT
FC [line number | search string]
FLAGS [%nnnnnn]
HISTORY [number of lines]
HELP <command>
VOLUME <vol.subvol>

Parameter substitution may be used at BMON configuration time to select parameter values according to a pre-configured parameter database.

Prior to using parameter substitution, GLOBALPARAMS, LOCALPARAMS and SHELLNAME must be declared, along with DELIMITERS if the default value delimiters are not to be used.

Parameters for substitution should be enclosed within delimiters to allow substitution to take place. Values of expanded parameters will be output as they are expanded.

A.3 BMON Environment Commands

The BCOM Environment Commands are used in setting or displaying values controlling the structure and operating mode of the BMON environment.

CHECKPOINT BMON [, BACKUP^CPU n]

This command instructs the currently accessed BMON process to start a Backup process in the nominated CPU, and to run as a NonStop Process Pair. If a backup CPU is not specified, or the one specified is not available, then the highest available CPU will be used.

During the BMON BUILD phase, there is a significant Checkpointing overhead whilst large internal tables are constructed. To avoid this, the recommended method is to include the CHECKPOINT BMON command as the last statement in the BCOM input file containing the configuration information. The command may also be issued interactively at any time during the life of any BMON process.

INFO BCOM

This command displays the current settings:

```
DELIMITERS          < >
GLOBALPARAMS        <file-name>
LOCALPARAMS         <file-name>
SHELLNAME           <shellname>
LISTING             <on | off>
PAGING              <on | off>
INTERACTIVE         <on | off>
UPSHIFT             <on | off>

ON DEMAND JOB       <N/A | nnnn>
```

INFO BMON

INFO BMON returns the “OWNER” and “SECURITY” settings of BMON as described in “SET BMON” and also the current “LOG1” and “LOG2” files if they have been specified.

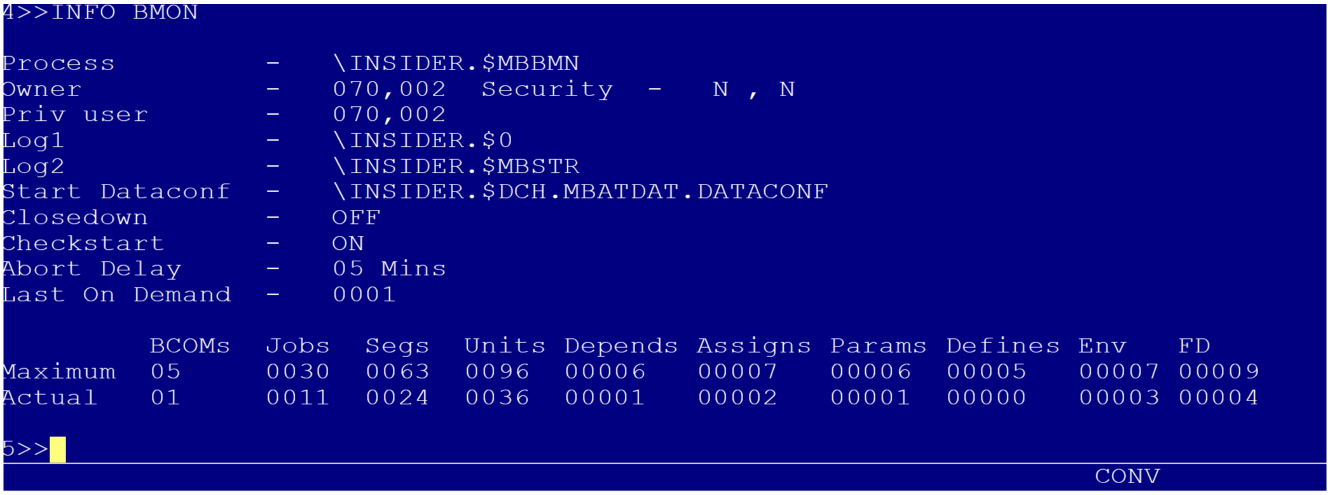
During the initial build of the BMON, a DATACONF Assign is specified. This Assign value is stored within the running BMON and therefore, users do not need to specify an Assign for the DATACONF location.

This DATACONF can be viewed via the INFO BMON command.

The maximum and actual number allocated for the following are also returned in tabular form:

1. **BCOMs** communicating with this BMON
2. **JOBS** allocated within BMON
3. **SEGMENTs** allocated within BMON
4. **UNITS** allocated within BMON
5. **DEPENDS** allocated within BMON
6. **ASSIGN** messages within BMON
7. **PARAM** messages within BMON
8. **DEFINES** allocated within BMON
9. **ENVs** allocated within BMON
10. **FDs** allocated within BMON

Figure A.1 is an example of typical information returned from an INFO BMON command.



```
4>>INFO BMON

Process         -   \INSIDER.$MBBMN
Owner           -   070,002  Security  -   N , N
Priv user       -   070,002
Log1            -   \INSIDER.$0
Log2            -   \INSIDER.$MBSTR
Start Dataconf  -   \INSIDER.$DCH.MBATDAT.DATACONF
Closedown       -   OFF
Checkstart      -   ON
Abort Delay     -   05 Mins
Last On Demand  -   0001

         BCOMs  Jobs  Segs  Units Depends Assigns Params Defines Env    FD
Maximum  05     0030  0063  0096  00006   00007   00006  00005   00007 00009
Actual   01     0011  0024  0036  00001   00002   00001  00000   00003 00004

5>>
                                                                           CONV
```

Figure A.1 – INFO BMON Output

LOG1 <file-identity>

This command specifies the identity of the LOG1 file. Can be a Terminal, Printer, Disk File or Process located within your NonStop Network. It is recommended that LOG1 specifies the STREAMER PROCESS.

The current LOGn File is closed (if one is open) and the new LOGn file opened. The nominated file must exist; if not, the previous LOGn file will continue to be used.

OPEN BMON <$process-name>

This BCOM command opens the BMON specified by "process-name". The purpose of this command is to switch from the current BMON to access another.

RESET BMON

This BCOM command resets the BMON attributes to their default values and has no parameters. For default settings see "SET BMON" command.

RESET DELIMITERS

This BCOM command resets the parameter delimiters to their default values of < and >.

SET BCOM <BCOM-parameter>

This command establishes values for the BCOM parameter settings. BCOM parameter is one of the following:

DELIMITERS <char> Use this command to alter the parameter Delimiters used by BCOM. Default values are < and >. Any string contained between Delimiters will be considered a parameter and resolution will be attempted from the local file and the global file as applicable.

GLOBALPARAMS <file-name> This command defines the location of the GLOBAL parameter file to be accessed for parameter resolution.

INTERACTIVE {ON} {OFF} This command causes a log message (of the form "<Job/Segment/Unit>
<added / deleted> interactively") to appear when adding / deleting a Job, Segment or Unit. The default value is ON if the input file for the BCOM session is a terminal. It is primarily targeted at accurately maintaining the status database.

LISTING <on | off> Use this command to suppress echoing of input files to output devices.

file to BCOM. This file will be accessed for parameter resolution.

PAGING <on | off> This command may be used to display a page full of information at a time. IF set to ON, then the prompt "Continue?" is displayed at the base of the screen. Enter "Y" to display the remaining information.

SHELLNAME <shell name> The defined shell name will be used during access of the local parameter file for parameter resolution.

UPSHIFT {ON} {OFF} By default, MultiBatch upshifts parameters. For cases where lower case parameters are required, set UPSHIFT to OFF.

SET BMON <BMON-parameter>

This command establishes values for the total BMON environment when BMON is initially started. BMON-parameter is one of the following:

CLOSEDOWN {ON} {OFF}
CHECKSTART {ON} {OFF}
MAXASSIGNS <n>
MAXBCOMS <n>
MAXDEFINES <n>
MAXDEPENDS <n>
MAXENVS <n>
MAXFDS <n>
MAXJOBS <n>
MAXPARAMS <n>
MAXSEGS <n>
MAXUNITS <n>
NEWRECOVER <file-name>
OWNER <m,n>
PRIV <m,n>
SECURITY <m,n>

The default values of the SET BMON attributes are shown in Figure A.2.

Attribute	**Default**	**Minimum**	**Maximum**
CLOSEDOWN	OFF	N/A	N/A
CHECKSTART	OFF	N/A	N/A
MAXASSIGNS	30000	0	30000
MAXBCOMS	20	1	20
MAXDEFINES	30000	0	30000
MAXDEPENDS	2500	0	2500
MAXENVS	30000	0	30000
MAXFDS	30000	0	30000
MAXJOBS	2500	1	2500
MAXPARAMS	30000	0	30000
MAXSEGS	5000	1	5000
MAXUNITS	10000	1	10000
NEWRECOVER	None	N/A	N/A
OWNER	000,000	N/A	N/A
PRIV	None	N/A	N/A
SECURITY	N,N	N/A	N/A

Figure A.2 - SET BMON Default Values and Limits

Parameters and their meaning are as follows:

CLOSEDOWN <on | off> This specifies whether the BMON process should automatically perform a SHUTDOWN BMON command upon successful completion of all jobs configured within the BMON process. The default value is CLOSEDOWN OFF, that is the BMON process will not SHUTDOWN automatically.

CHECKSTART <on | off> This prompts the BCOM user for confirmation if:

1. A restart Unit is requested on a Unit that has previously run.

2. A START Job/Segment/Unit is requested and PREV dependencies are incomplete.
3. A START Job/Segment/Unit is requested and NEXT dependencies have already been started.

MAXASSIGNS <n>. This command indicates the maximum number of ASSIGN messages that BMON is to accommodate. For more information on BMON's handling of ASSIGN messages see "SET STEP ASSIGN" command.

MAXBCOMS <n>. This command specifies the maximum number of BCOMs that can open BMON at any one time, including that via the Status Monitoring MultiBatch Facility (MBF) screen.

MAXDEFINES<n>. This command indicates the maximum number of DEFINES that BMON is to accommodate. For more information on BMON's handling of DEFINES see "SET STEP DEFINE" and "ADD STEP DEFINE" commands.

MAXDEPENDS <n>. This is the maximum number of dependent processes that may exist within BMON. "n" must be less than or equal to the value of MAXUNITS. This parameter could be described as the maximum number of STEP 2s in the configuration.

MAXENVS <n> This command indicates the maximum number of OSS ENVIRONMENT messages that BMON is to accommodate. For more information on BMON's handling of PARAM messages see "SET OSS-STEP OSS-ENV" command.

MAXFDS <n> This command indicates the maximum number of OSS FD messages that BMON is to accommodate. For more information on BMON's handling of PARAM messages see "SET OSS-STEP OSS-FD" command.

MAXJOBS <n>. This is the maximum number of jobs that can exist within this BMON; BMON creates an internal "JOB TABLE" of appropriate size depending upon this value.

MAXPARAMS <n>. This command indicates the maximum number of PARAM messages that BMON is to accommodate. For more information on BMON's handling of PARAM messages see "SET STEP PARAM" command.

MAXSEGS <n>. This is the maximum number of SEGMENTs that can exist within all JOBs within BMON. "n" must be greater than or equal to the value of MAXJOBS.

MAXUNITS <n>. This is the maximum number of units that may exist within all segments within BMON. "n" must be greater than or equal to the value of MAXSEGS.

NEWRECOVER <file-name>. The filename will be created when "START BMON" is executed.

This file can be used in future runs of BMON to recover BMON to the state it was in when stopped, when it will be utilised by the BCOM SYNC command in order to recover a BMON. See Appendix H 'Recovering a BMON' for further details.

OWNER <m , n>. This command allocates the Group.User identity of BMON, where m = Group identity and n = user identity.

This is used when vetting whether the owner of a BCOM process is allowed to perform certain commands, and is dependent upon the setting of "SET BMON SECURITY".

PRIV <m, n>. This command specifies the identity of a privileged user. The privileged user is permitted access to commands normally restricted only to SUPER.SUPER.

SECURITY <x, y>. This setting works in conjunction with the SET BMON OWNER command, and is the classification of access to the BMON environment from any BCOM process.

This is similar to NonStop File security and indicates which commands are permitted to be performed by classes of users. Any BCOM "owner" can get status information from BMON, via the STATUS and INFO commands, however there are two further classes of Command which are directly controlled by the

The first class, which relates to “x” above, are those that involve Configuration changes to BMON; these are “ADD”s, and “DELETE”s.

Every other command, such as “START JOB”, “HOLD SEGMENT”, “ABORT UNIT” relates to the second security class - ‘y’.

“**x**” and “**y**” can both have any of the following values:

“**-**” SUPER.SUPER Access only.

“**O**” Local, non-networked Owner-only Access

“**U**” Local or Networked User (Owner) only Access

“**G**” Local only, Group-wide Access

“**C**” Local or Networked, Group-wide Access

“**A**” Any Local user

“**N**” Any Local or Network user.

All of which relate to standard NonStop Guardian “RWEP” values for File Security.

The first BCOM to access a BMON in order to issue “START BMON”, must have the same Creator Accessor ID (CAID) as the BMON and is capable of performing any command.

The BCOM commands and the associated security attributes are detailed below.

Command	Attribute A (class x)	Attribute B (class y)	No security control
ABORT		X	
ADD	X		
ADD FROM MODEL	X	X	
ALL			X
ALTER		X	
ALTER OSS-STEP		X	
COMMENT			X
CHECKPNT			X
DELETE	X		
EXIT			X
FC			X
FLAGS			X
HOLD		X	
INFO			X
LOG1	X		
LOG2	X		
OBEY			X
OPEN			X
RELEASE		X	
RESET			X
RESTART		X	
SET			X
SET OSS-STEP			X
SHOW			X
SHUTDOWN		X	
SHUTDOWN!		X	
START			
STATUS			X
SUBMIT JOB		X	
SYNC			X
VOLUME			X

Note: To use the following commands, the user must be logged on as SUPER.SUPER or as the BMON Privileged User:

SHUTDOWN !

ALTER with FLAGS

ALTER with RUNON

ALTER with RUNOFF.

SHOW BMON

This command lists the current settings of the “SET BMON” attributes.

SHUTDOWN BMON or SHUTDOWN BMON [!]

This command causes BMON to stop running.

This command requires the second class of security for execution (see “SET BMON SECURITY”).
To use SHUTDOWN BMON !, then a BMON Privileged User must be used, or super.super.

START BMON

The command “START BMON” can only be issued once to a BMON Process and informs BMON of the size of its internal Tables in memory.

The "SET BMON" command values are passed to BMON at this point and BMON is then ready for any valid input commands.

STATUS BMON

This command returns the time and date that BMON started running, the number of JOBs that are currently active (i.e. have at least one Unit still running), and the number of JOBs that have all their units completed successfully (see Section 2 for an explanation of a successfully completed unit).

A list of all BCOMs actively connected to this BMON will also be returned, and the type of Security they possess (see "SET BMON SECURITY" for more details).

SWITCH BMON

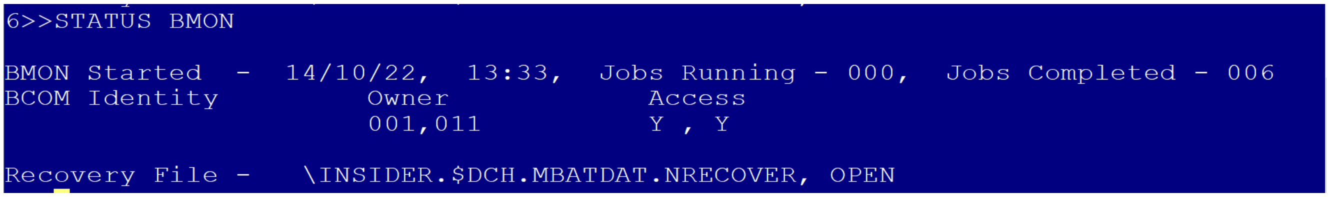
When BMON is running as a process pair this commands swaps the processes so that the back-up process becomes the primary process and vice-versa.

See the CHECKPOINT BMON command for running BMON as a process pair.

SYNC

SYNC IN <replicated-recovery-file-name> [!], TRANSLATE <rules-file-name>

This command starts BMON using the <replicated-recovery-file> to reconstruct BMON's tables to the state they were in when the BMON stopped writing to the new recovery file. The <rules-file-name> is used to change the node/volume/ subvolume/directory which BMON was using previously to what BMON will use now.



```
6>>STATUS BMON

BMON Started  -  14/10/22,  13:33,  Jobs Running - 000,  Jobs Completed - 006
BCOM Identity          Owner              Access
                       001,011             Y , Y

Recovery File -   \INSIDER.$DCH.MBATDAT.NRECOVER, OPEN
```

SYNC OUT [<newrecover-file-name>]

This command will cause BMON to write all its internal tables to the new recovery file. If the newrecover-file is not specified, then a SET BMON NEWRECOVER command should have been issued previously.

Figure A.3 - STATUS BMON Command

A.4 JOB Commands

The Job Oriented Commands are involved in creating or displaying entities relating to BMON at Job level.

ADD FROM MODEL <xxxx>

This command takes a copy of a Model Job and create an On Demand Job. Job name references in segment dependencies are changed to match the new On Demand Job.

The referenced job <xxxx> must be a Model Job.

The command will be rejected if BMON cannot allocate any of the required resources. See the INFO BMON command.

ADD JOB <xxxx>

This command inserts the Job "xxxx" into the internal JOB Table and allocates this Job's Segment table to a suitable size to accommodate the "SET JOB MAXSEGS" values.

The command will be rejected if BMON cannot allocate any of the required resources. See the INFO BMON

The Job identity must consist of four alphanumeric characters, the first of which must be alphabetic.

DELETE JOB <xxxx>

This command removes all trace of Job <xxxx>. The command will fail if any UNIT is running when the command is executed.

Internal totals will be adjusted if they are affected (i.e. number of jobs within BMON).

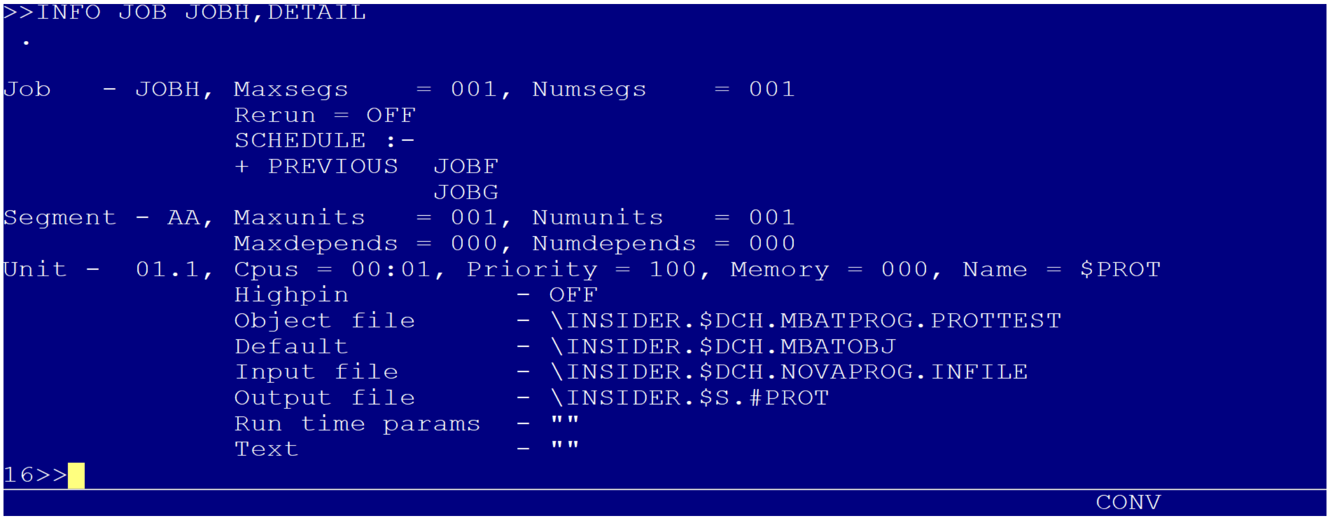
HOLD JOB <xxxx>

When this command is issued no UNITs within JOB <xxxx> may be Started or Restarted unless the JOB has a subsequent "RELEASE" command issued for it. (See "RELEASE JOB" command).

This command has no effect upon any UNITs already running.

INFO JOB <xxxx> [, DETAIL]

This command returns the characteristics set with "SET JOB", "SET SEGMENT" and "SET STEP" that were subsequently added to this JOB.



```
>>INFO JOB JOBH,DETAIL
 .

Job    - JOBH, Maxsegs    = 001, Numsegs    = 001
              Rerun = OFF
              SCHEDULE :-
              + PREVIOUS  JOBF
                          JOBG
Segment - AA, Maxunits   = 001, Numunits   = 001
              Maxdepends = 000, Numdepends = 000
Unit -   01.1, Cpus = 00:01, Priority = 100, Memory = 000, Name = $PROT
              Highpin              - OFF
              Object file          - \INSIDER.$DCH.MBATPROG.PROTTEST
              Default              - \INSIDER.$DCH.MBATOBJ
              Input file           - \INSIDER.$DCH.NOVAPROG.INFILE
              Output file          - \INSIDER.$S.#PROT
              Run time params      - ""
              Text                 - ""
16>>
                                                                  CONV
```

If "*" is used for the Job Identity, information relating to all jobs is returned.

"DETAIL" is an invalid parameter with "INFO JOB *".

An example of "INFO JOB" is shown in Figure A.4.

Figure A.4 - Example of an INFO JOB, DETAIL Command

INFO JOB <wildcard>

A user can submit partial job names with an asterisk as a wildcard character. For example, INFO JOB A* will return basic information for all jobs starting with **A**.

Note: *See Appendix F.4: BCFORMAT which describes how a user can return selected STEP configuration details, without recourse to the use of, DETAIL.*

RELEASE JOB <xxxx>

When issued, the "HOLD" is taken off Job<xxxx>. If JOB <xxxx> is already processing, or was not in "HOLD" this command has no effect.

RESET JOB

SET JOB <job-parameter>

This command establishes JOB parameters for use when the "ADD JOB" command is executed.

The <job-parameter> can be one of the following:

1. **CRITICAL {ON} {OFF}**. This command allows clerical setting of the critical path. A failed JOB or a HOLD JOB command issues extra information in the log message if the entity is on the critical path.
2. **GROUPID <group-identifier>**. This command specifies a group identifier up to a maximum of 30 characters. The Groupid option is only valid for re-runnable jobs.
3. **MAXSEGS <n>**. The maximum number of segments that can exist within a subsequently added Job.
4. **PREV <xxxx>**. The identity of a JOB that must be complete, i.e. all UNITS in all SEGMENTS must have been processed to a normal end, before BMON will automatically start a subsequently added JOB. A Job may be dependent on completion of up to ten other JOBs before it automatically starts. Any given Job may have up to ten other jobs dependent on it. Any Job referenced by this command must have previously been added to the BMON configuration.
5. **RERUN {ON} {OFF}**. Specifies whether a job is to be considered as a re-runnable Job. A re-runnable job must have a Groupid specified. All Units defined as part of a runnable job will be reset into a configured state upon successful completion of the Job. Re-runnable jobs can only have job level dependencies that are within the same Groupid.
6. **STARTTIME hh:mm**. If no other dependencies exist, the job will automatically be invoked at STARTTIME. If STARTTIME is used in conjunction with Job dependencies the Job will only start when all previous Jobs have completed successfully and the STARTTIME has been reached or passed. The STARTTIME is considered to be a futuristic time from BMON start time; that is a Job configured with a start time of 14.20 in a BMON schedule which started at 14.30 would not run until 14.20 on the following day. If STARTTIME is passed and the Job has not started, then a warning message is written to the MultiBatch Log.

	DEFAULT	MINIMUM	MAXIMUM
CRITICAL	Off	N/A	N/A
GROUPID	None	N/A	N/A
MAXSEGS	1	1	200
PREV	None	None	10
RERUN	Off	N/A	N/A
STARTTIME	None	N/A	N/A

Figure A.5 - SET JOB Default Values and Limits

SHOW JOB

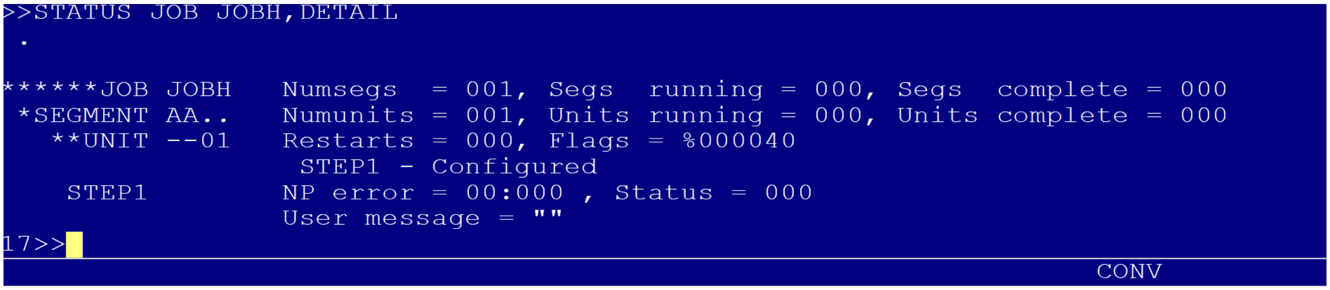
This command outputs the current setting of the "SET JOB" attributes.

START JOB <xxxx>

This command will cause all UNITs with JOB identity <xxxx> to be Started by BMON if they are in a suitable state (not running or previously started).

STATUS JOB <xxxx> [, DETAIL | <Qualifier>]

This command returns the current state of a given Job and its components (SEGMENTS, UNITS, STEPS). See Figure A.6.



```
>>STATUS JOB JOBH,DETAIL
 .

*****JOB JOBH   Numsegs  = 001, Segs  running = 000, Segs  complete = 000
 *SEGMENT AA..  Numunits = 001, Units running = 000, Units complete = 000
   **UNIT --01  Restarts = 000, Flags = %000040
                 STEP1 - Configured
    STEP1       NP error = 00:000 , Status = 000
                User message = ""
17>>
                                                                  CONV
```

Figure A.6 - Example of STATUS JOB, DETAIL Command

If "*" is used for the Job identity the status of all jobs is returned.

"DETAIL" is an invalid parameter with" STATUS JOB *".

If the Qualifier is used the resulting output will consist only of Jobs which meet the qualifier criteria.

Qualifier can be one of:
1. Aborted
2. Complete
3. Failed
4. Ondemand
5. Protocol
6. Running
7. Stopped
8. Twostep
9. Waiting.

STATUS JOB <wildcard>

A user can submit partial job names with an asterisk as a wildcard character. For example, STATUS JOB B* will return basic status results for all job names starting with **B**. The ,DETAIL option is unavailable with the wildcard.

SUBMIT JOB

This command starts each segment of the current On Demand Job where the segment does not have any previous dependencies.

A.5 SEGMENT Commands

Segment Commands are involved in creating or displaying entities relating to BMON at SEGMENT level.

ADD SEGMENT <xxxx.yy>

SEGMENT identity <yy> must consist of two alpha, numeric, or alphanumeric characters.

This command inserts SEGMENT <yy> into the SEGMENT table of JOB <xxxx>. SEGMENT <xxxx.yy> will then have a "UNIT Table" constructed for it of a size indicated by the parameters set by the "SET SEGMENT" command.

This command will be rejected in the event the Segment table for Job <xxxx> is full or if the Global maximum number of UNITs within BMON is exceeded (see "SET BMON MAXUNITS" command).

DELETE SEGMENT <xxxx.yy>

This command removes all trace of SEGMENT <xxxx.yy>. The command will fail if any Unit belonging to this Segment is running when the command is executed.

Internal totals will be adjusted if they are affected (i.e.: number of segments within Job <xxxx>).

HOLD SEGMENT <xxxx.yy>

This command causes Segment <xxxx.yy> to reject any Start or Restart commands that pertain to any of its constituent UNITs.

The opposite command is "RELEASE SEGMENT". SEGMENTs having all UNITs currently processing are not affected.

INFO SEGMENT <xxxx.yy> [, DETAIL]

This command returns the characteristics of a given SEGMENT, set with "SET SEGMENT" and "SET STEP", that were subsequently added to this Segment.

It returns exactly the same information as "INFO JOB" with only SEGMENT and UNIT information pertaining to the requested SEGMENT.

RELEASE SEGMENT <xxxx.yy>

This command removes the HOLD status from Segment <xxxx.yy>.

RESET SEGMENT

This returns the "SET SEGMENT" attributes to their defaults.

SET SEGMENT <segment-parameter>

This command establishes Segment parameters for use when the "ADD SEGMENT" command is executed. The default values of these attributes are shown in Figure A.7.

The Segment Parameter can be one of the following:

1. **CRITICAL {ON} {OFF}**. This command allows clerical setting of the critical path. A failed SEGMENT or a HOLD Segment command issues extra information within the EMS event text if the entity is on the critical path.
2. **MAXDEPENDS <n>**. This sets the maximum number of two-step processes that may exist within a subsequently added SEGMENT.
3. **MAXUNITS <n>**. This sets the maximum number of UNITs that may exist within a subsequently added SEGMENT.
4. **PREV <xxxx.yy>**. The identity of a Segment that must complete before BMON will automatically start a subsequently added Segment. A given Segment may have up to ten other segments dependent on it. Any Segment referenced by this command must have already been configured in BMON.

	Default	Minimum	Maximum
CRITICAL	Off	N/A	N/A
MAXDEPENDS	0	0	200
MAXUNITS	1	1	200
PREV	None	None	10

Figure A.7 - SET SEGMENT Default Values and Limits

SHOW SEGMENT

This command returns the current settings of the "SET SEGMENT" attributes.

START SEGMENT <xxxx.yy>

This command will cause all UNITs belonging to Segment identity <xxxx.yy> to be run, if they are in a suitable state (not running or previously started).

Note: *A segment of a re-runnable Job cannot be started.*

STATUS SEGMENT <xxxx.yy> [, DETAIL]

This command returns the current state of a given Segment. It returns exactly the same information as "STATUS JOB" with only Segment and Unit information pertaining to the Segment requested.

A.6 UNIT Commands

Unit Commands are involved in creating or displaying entities relating to BMON at UNIT level.

ABORT UNIT <xxxx.yy.zz>

Any STEP that is running within this UNIT will have an attempt made to stop it by BMON. The step(s) concerned will have their "Abort Unit requested via BCOM" flags altered.

The flag will be set whether or not the attempt is successful.

DELETE UNIT <xxxx.yy.zz>

This command deletes this Unit from BMON. This command will fail if the Unit is running.

HOLD UNIT <xxxx.yy.zz>

This command prevents this Unit from being Started or Restarted. The opposite command is "RELEASE UNIT". If the Unit is already running it is allowed to continue.

INFO UNIT <xxxx.yy.zz> [, DETAIL]

This command returns the characteristics of a given Unit set with "SET STEP" that has subsequently been added to this <JOB.SEGMENT>.

It returns the same information as "INFO JOB" with only the Unit information pertaining to the Unit requested.

When detail is requested all the STEP chacteristics are displayed. If there are more than 10 ASSIGNs or DEFINEs then the number configured is returned rather than the details.

INFO UNIT <xxxx.yy.zz> , ASSIGNS

This command returns the characteristics of a given Unit as set with "SET STEP ASSIGN".

It can be used to return ASSIGN details where the INFO UNIT <xxxx.yy.zz>, DETAIL command has returned the number of configured ASSIGNs.

INFO UNIT <xxxx.yy.zz> , DEFINES

This command returns the characteristics of a given Unit as set with "SET/ADD STEP DEFINE".

It can be used to return DEFINE details where the INFO UNIT <xxxx.yy.zz>, DETAIL command has returned the number of configured DEFINEs.

RELEASE UNIT <xxxx.yy.zz>

This command takes the HOLD status off the specified UNIT.

RESTART UNIT <xxxx.yy.zz>

The specified Unit will be restarted if a previous Start has been issued ("Process Creation Attempted" flag set for STEP 1 of the unit).

A "Protocol" Unit cannot be restarted if previously completed successfully. A non-Protocol Unit may be restarted as many times as required regardless of what any previous runs achieved.

START UNIT <xxxx.yy.zz>

This command causes a new Process to be created for Step 1 of the given Unit. If the Process is successfully

This command will be rejected if there has previously been an attempt to run the Unit (“process creation attempted” flag set for STEP 1 of the unit).

Note: *A unit within a re-runnable job cannot be started.*

If a previous attempt has failed, then “RESTART UNIT” must be used.

STATUS UNIT <xxxx.yy.zz> [, DETAIL]

This command returns the current state of a given Unit. It returns the same information as “STATUS JOB” without any Unit information other than that relating to the Unit requested. For more information, see “STATUS JOB” command.

A.7 STEP Commands

Step commands are involved in creating or displaying entities relating to BMON at Step level.

ADD STEP <xxxx.yy.zz[.n]>

This command is used to add a Step incorporating the current "SET STEP" attributes.

"n" represents either a 1 or 2 indicating the first or second Step for interdependent processes. If absent, STEP 1 is assumed. If Step 1 is being added a new entry is inserted in the Segment's Unit table.

In both cases, a Step table is created for the attributes and status of the Step.

The command will not be executed if this segments Unit table is full (see "SET SEGMENT MAXUNITS") or the number of interdependent processes would be exceeded (see "SET SEGMENT MAXDEPENDS").

ADD STEP DEFINE <=defname>

This command is used to add a define from the current working set into the define table for the current step. All defines in the define table will be associated with the step when an ADD STEP command is issued.

A maximum of 300 defines may be added per step, each define being validated when added. The define table is only reset when a RESET STEP command is issued.

ALTER STEP <xxxx.yy.zz <.n>> commands

These commands alter previously set attributes that require altering after a Step has been added. The parameters and syntax are the same as for "SET STEP".

The following attributes only may be altered:

```
ASSIGN <log>, <physical>
CPU m , n
DEFAULT <file-name>
DEFINE <=defname> <attrib> <value>
DELETE ASSIGN <log>
DELETE DEFINE <=defname>
DELETE PARAM <name>
FLAGS %nnnnnn
HIGHPIN {ON} {OFF}
INFILE <file-name>
LIBRARY <file-name>
MEM n
NAME <$name>
NOMAXTIME
OBJECT <file-name>
PARAM <name>,"value"
PRI n
PROTOCOL {ON} {OFF}
RUNOFF
RUNON
RUNPARAMS <"...">
SWAP <file-name>
TERM <file-name>
TEXT <"...">
```

Note: *The ALTER STEP command which allows FLAGS amendment and RUNOFF / RUNON amendment is reserved for a user logged on as the local SUPER.SUPER or the BMON Privileged User Id.*

The command will not be accepted if it produces an illogical status (i.e., running when BMON has no Process Id in that STEPs table entry, or complete but still running, etc.).

The RUNOFF / RUNON option allows a step to be modified such that the step will / will not run when requested. If the unit is skipped, any dependencies will continue as if the unit had completed successfully.

The NOMAXTIME option clears the MAXTIME for the specified step. It may be used when the step is running or when dormant. It is envisaged that this command will only be required in critical Applications or System failure situations when Database Consistency may be in jeopardy, and should ONLY be used after consultation with the relevant Support function within your Organization.

RESET STEP

This command reverts the "SET STEP" attributes to their default values.

RESET ASSIGN

This command clears the STEP ASSIGN table.

RESET PARAM

This command clears the STEP PARAM table.

SET STEP <step-parameter>

This command establishes STEP attributes for use when the "ADD STEP" command is executed.
The default values of these attributes are shown in Figure A.8.

The Step Parameter can be one of the following:

1. **ASSIGN <logical-unit etc..>.** There may be up to 300 separate "SET STEP ASSIGN" statements configured for use by each Step. This command is used to carry out Logical to Physical file assignment at run time. The syntax of the parameters is the same as for the TACL.

   Note: *The maximum value for the LOGICAL field is 31 characters.*

2. **ASSIGN TACLCSTM.** Allows you to enter a NULL assignment for TACLCSTM

3. **CRITICAL {ON} { OFF}.** This command allows clerical setting of the critical path. A failed STEP or a HOLD Step command issues an extra log message if the entity is on the critical path.

4. **CPU <m , n>.** The CPU that the Step is to run in is defined together with an alternative for when a CPU is unavailable at run time, "m" is the first choice "n" the alternative.

5. **DEFAULT <file-name>**. This Process's default volume.subvolume to be sent in the Startup message.

6. **DEFINE <attribute> <attribute-value>**. This command is used to set define attributes in the working set. Before any other attribute can be set, the DEFINE CLASS must be set, which resets the working set to default values.
   Validation will be performed to prevent illegal attribute values being set. The define will be added to the define table only when an ADD STEP DEFINE command is issued.

7. **HIGHPIN {ON} {OFF}**. This command can be used to create the step as a process executing higher than PIN 255 in a given CPU. **Note**: Your process needs to be coded and compiled to run at High Pin otherwise it will revert to a Low Pin process.

8. **IFFAIL <xxxx>**. Setting the IFFAIL Job will cause the specified Job to be invoked should this step abend or finish abnormally. **Note**: The IFFAIL Job must exist before it can be specified for this step.

9. **INFILE <file-name>**. The file name to be sent to this Process in the Startup message, as for "TACL".

10. **LIKE xxxx.yy.zz <.n>**. This command sets the values of the "SET STEP" attributes to be like those of the previously added Step specified.

11. **MAXTIME <hh:mm> [,WARN].** An appropriate warning message is generated <n> minutes before the process is aborted, where <n> can be set by a user and defaults to 5. To set <n> see the RUN

12. **MEM <n>**. The memory to be allocated as for the TACL parameter.

13. **NAME <$x>**. The name to be given to this Process; if absent this will result in a Process name being chosen by the Operating System. All BMON initiated processes are named processes. For Two-Step Units, nomination and use of a Process Name for STEP 1 is mandatory to enable STEP 2 to write to STEP 1 in the event of dependent processes being required.

14. **OBJECT <file-name>**. The Object Filename for this Step. This is a required parameter.

15. **OUTFILE <file-name>**. The Filename to be sent to this Process in the Startup message, as for TACL.

16. **PARAM <name>,”value”**. This has the same function as when issued from the TACL prompt, and will be passed to the STEP at run-time as an inter-process message in PARAM format.

    Note*:*

    The “value” field must be enclosed in quotes.
    The <name> can be a maximum of thirty characters, and the “value” can be a maximum of eighty characters in length, excluding the quotes.
    There can be up to 100 SET STEP PARAM statements for each STEP.

17. **PRI <n>**. The priority this Process will run at.

18. **PROTOCOL {ON} {OFF}**. This parameter indicates whether a Step obeys BMONs Application Interface Protocol.

19. **RUN [NOT] <param-name>**. This command sets the RUNOFF / RUNON marker at configuration time, depending upon a predefined Boolean parameter. The parameter will be expanded to either true or false indicating, along with optional use of NOT, whether the particular step is to be invoked during the next execution of the schedule.
    Any step which will not run through use of this command will be treated as a successful completion by any dependencies.

20. **RUNPARAMS <”x.   ”>**. Up to 80 characters to be sent as the runtime Parameter in the Startup message. The value must be enclosed in quotes or obliques. The first quote or oblique will be taken to be the delimiter for the RUNPARAMS, e.g.:

    a. SET STEP RUNPARAMS “SECURE ABC, “NONO””
    b. SET STEP RUNPARAMS /SECURE ABC, “NONO”/

    This attribute can be used where “FUP COPY A,B” type processing is required, when it would save having to set up an input file for the FUP.

21. **SWAP <file-name>**. The Guardian Swap file to be used by this Step, as for TACL.

22. **TERM <file-name>**. The Home Terminal of this Step, if required to be different from the BMON Process Hometerm.

23. **TEXT <”.x..”>**. This parameter is simply provided to allow some annotation within BMON as to the purpose of this Step. Up to 40 characters of text may be supplied between either quotes or obliques. The contents of this field are displayed when “INFO, DETAIL” commands are actioned.

	Default
ASSIGN	No assign messages passed
CPU	0,1
DEFAULT	As BMON’s default volume, subvolume
DEFINE	No defines passed
HIGHPIN	Off
INFILE	BMON’s home terminal
LIBRARY	No runtime library allocated

MEM	Memory as allocated at compile time
NAME	Unnamed Process
OBJECT	Required Parameter
OUTFILE	BMON's home terminal
PARAM	No PARAM messages passed
PRI	100
PROTOCOL	On for Step 1, always off for Step 2
RUNPARAMS	No runtime parameters
SWAP	Temporary swap file
TERM	BMON's home terminal
TEXT	No text annotation

Figure A.8 - SET STEP Default Values

SHOW DEFINE

This command shows the current define working set.

SHOW STEP

This command displays the current settings relating to "SET STEP".

A.8 OSS Step Commands

ADD OSS-STEP <xxxx.yy.zz[.n]>

This command is used to add an OSS Step incorporating the current SET OSS-STEP attributes, and many of the SET STEP attributes.

"n" represents either a 1 or 2 indicating the first or second Step for interdependent processes. If absent, STEP 1 is assumed. If Step 1 is being added a new entry is inserted in the Segment's Unit table.

In both cases a Step table is created for the attributes and status of the Step.

The command will not be executed it this segments Unit table is full (see "SET SEGMENT MAXUNITS") or the number of interdependent process would be exceeded (see "SET SEGMENT MAXDEPENDS").

ALTER OSS-STEP <xxxx.yy.zz<.n>> commands

These commands change previously set attributes that require altering after an OSS step has been added. The following parameters and syntax are the same as for "SET OSS-STEP".

```
OSS-PROGRAM   <program-name>
OSS-PATHNAME  <pathname>
OSS-ARGUMENTS <arguments> [, <arguments>]
OSS-FDCWD     <current^working^directory>
ADD OSS-ENV   <env details>
ADD OSS-FD    <fd>, <dupfd>, <name>, <access flag>, <mode>
```

The following parameters and syntax are different to the "SET OSS-STEP" syntax:

```
(<env^number>)ALTER OSS-STEP <xxxx.yy.zz<.n>> OSS-ENV <env details>
(<fd^number>) ALTER OSS-STEP <xxxx.yy.zz<.n>> OSS-FD  <fd>,<dupfd>,<name>,<access
  flag>,<mode>
(<env^number>)ALTER OSS-STEP <xxxx.yy.zz<.n>> DELETE OSS-ENV
(<fd^number>) ALTER OSS-STEP <xxxx.yy.zz<.n>> DELETE OSS-FD
```

The env^number and fd^number are values which uniquely identify the particular record within the step. These values can be found by typing INFO JOB <Job Name>, Detail. The number of each parameter will be displayed in parentheses at the beginning of each parameter.

when Database Consistency may be in jeopardy, and should ONLY be used after consultation with the relevant Support function within your Organization.

For example, an OSS-STEP has these OSS parameters configured:

```
Unit - 02.1, Cpus = 00:01, Priority = 100, Memory = 000, Name = $JOBE
           Highpin       - OFF
           Pathname      - /E/appl/G/dev/user/
           Program file - ossenve OSS Arguments           - “”
           Text          - “”
           FD CWD        - /E/appl/tmp/user
           File Descriptors :-
  (01)    * FD Number: 0003, Dup FD: -0001, Name: outfile, Flag: %H00009, Mode:
  %0777
          *Environment :-
  (01)    * path=$path
```

To alter the Environment ‘path’, use the following syntax:

(01) alter oss-step <job id>.<seg id>.<unit id> oss-env path=$path2

Whereby, the above **(01)** entry is specified first.

RESET OSS-ENV

This command clears the OSS-STEP OSS-ENV table.

RESET OSS-FD

This command clears the OSS-STEP OSS-FD table.

RESET OSS-STEP

This command reverts the “SET OSS-STEP” attributes to their default values.

SET OSS-FLAG <flag> ON/OFF

Flag is one of CREAT, EXCL, NOCTTY, TRUNC, APPEND, NOBLOCK or SYNC.

These flags are used by subsequent SET OSS-FD, ALTER OSS-STEP <xxxx.yy.zz[.n]> OSS-FD and ALTER OSS-STEP <xxxx.yy.zz[.n]> ADD OSS-FD.

These relate to standard POSIX FD flags as follows:

Creation Flags (Supply zero or more)

O_CREAT(0x8)	Create file if it does not exist.
O_EXCL(0x20)	Exclusive use flag.
O_NOCTTY(0x8000)	Do not assign controlling terminal.
O_TRUNC(0x10)	Truncate flag.

Status Flags (Supply zero or more)

O_APPEND(0x4)	Set append mode.
O_NONBLOCK (0x4000)	Non-blocking mode.
O_SYNC(0x10000)	Write according to synchronised I/O file integrity completion.

Note that this value is combined as a Hex value with the SET OSS-FLAG <flag> values when a steps details are displayed, see below:

```
Unit - 02.1, Cpus = 00:01, Priority = 100, Memory = 000, Name = $JOBE Highpin
                        - OFF
           Pathname     - /E/appl/G/dev/user/
           Program file - ossenve OSS Arguments          - ""
           Text         - ""
           FD CWD       - /E/appl/tmp/user File Descriptors :-
  (01)    * FD Number: 0003, Dup FD: -0001, Name: outfile, Flag: %H00009, Mode:
  %0777
          *Environment :-
  (02)      * path=$path
```

SET OSS-STEP <commands>

This command establishes STEP attributes for use when the "ADD OSS-STEP" command is executed. These parameters do not have any default attributes.

SET OSS-STEP OSS-PROGRAM <program-name>

The OSS program name for this step. This is a required parameter.

SET OSS-STEP OSS-PATHNAME <pathname>

A list of path prefixes separated by a colon to further identify the OSS-PROGRAM parameter.

SET OSS-STEP OSS-ARGUMENTS <arguments> [, <arguments>]

A list of arguments to be passed to the main function of the OSS program.

SET OSS-STEP OSS-ENV <environment-parameters>

A parameter that helps to describe the environment of the new process.
There can be up to 20 SET OSS-STEP OSS-ENV parameters for each STEP.

SET OSS-STEP OSS-FDCWD <current-working-directory>

SET OSS-STEP OSS-FD <fd>, <dupfd>, <name>, <access flag>, <mode>

fd: The file descriptor to be opened:

0 = Standard Input
1 = Standard Output
2 = Standard Error
Other = User Defined

dupfd: Indicates whether the file specified in fd^number can be opened as a Duplicate

> 0 The file descriptor is a duplicate of a file descriptor previously specified in Fd^number

-1 The file descriptor is not a duplicate. Open the file according to the following values

name: The pathname of the file to be opened by the new process. A relative pathname is resolved with respect to the Current Working Directory.

access flag: This field is ignored when the file is opened as a duplicate.
One of the following flags may be supplied:

"Read Only" or 0
"Write Only" or 1
"Read Write" or 2

mode: Is the read, write and execute permissions of the file to be created when the file is to be opened with the Open Create flag set. This is expressed as an octal number, e.g. %0755

There can be up to 10 SET OSS-STEP OSS-FD parameters for each STEP.

A.9 Miscellaneous Commands

The miscellaneous commands do not relate specifically to Job, Segment, Unit, Step or the overall BMON environment

COMMENT <text>

Any comment line will be ignored by BCOM, merely serving to document a command file.

EXIT (or <E>)

When this command is executed, BCOM interface stops running.

FC

This is the standard NonStop implementation of the "FC" command.

FLAGS <%nnnnnn>

The flags command provides an expansion of the Status Flags relating to a given Step as obtained by the STATUS Commands for the JOB, SEGMENT or UNIT entities.

There are in all sixteen Flags which can be set or unset, however in the interests of readability they are grouped together into six numbers which have an Octal number base.

The first and last of these numbers can only have a value of either 0 or 1; the rest can theoretically range from 0 through 7, however only a small number of combinations represent a valid state.

When a STEP is first configured, and before any attempt is made to initiate the Process it represents, the FLAGs can have only one of four possible values:

%000040 - This Step observes BMON Protocol.

%000000 - No special attributes, (Non-Protocol STEP).

%000044 - This Step observes BMON Protocol and will be placed in INSPECT when Started.

%000004 - No special attributes, (Non-Protocol STEP), but will be placed in INSPECT when started.

Once any STEP is selected for initiation, either by Operator choice or the Dependency Scheduling mechanism, the FLAGs setting is used by BMON to track and record the progress of each vital stage of Process Creation, Parameter passing and STEP completion status.

Until familiarity is gained with the small number of FLAGs settings which will occur in a stable Applications and Operational environment, the FLAGS Command can be used to translate any numeric setting into the textual version as follows:

If no parameter is supplied a meaning of all possible Flags settings will be displayed (Figure A.10); the meaning of those bits requested will be displayed giving a history of the Step progress.

The parameter must be preceded by a "%" and then consist of six Octal digits (as returned by "STATUS UNIT"). This information indicates the history of a Process.

For Protocol processes, further information is available to BMON from the Status Field of Protocol messages.

This is an indication of the last Status received by BMON from that Process; any accompanying text sent with it (e.g. "JOB COMPLETED OK") is available from the DETAIL display of a STATUS Command for that JOB, SEGMENT or UNIT.

BIT	**MEANING**
0	Running.
1	**Newprocess called**
2	**Newprocess message received**
3	**Startup message written**
4	Aborted by operator
5	Open message received
6	Close message received
7	CPU failure
8	Abend message received
9	**Stop message received**
10	BMON Protocol in use
11	Initialise message received
12	Param file requested
13	INSPECT ON (TACL RUND)
14	Not used
15	**STEP complete**

For example, in BCOM, entering FLAGS %070101 provides the following summary:

```
1>> FLAGS %070101

(7)     Newprocess called

        Newprocess message received

        Startup message written

(1)     Stop message received

(1)     STEP complete
```

Figure A.9 - Status Flags Significance

HELP <command>[,detail]

This command is implemented as for NonStop utilities. Some commands contain two levels of definition, for example "HELP SET" would return a list which can be further qualified for detail of a command, such as "HELP SET BMON". Additional information can be obtained by using Detail Qualifier.

HISTORY <number of lines>

This command lists the previous commands which have been entered.

VOLUME<vol.subvol>

As within NonStop Command Interpreters, this command sets BCOMs default volume. If no parameter is supplied, the default when BCOM was started will be reinstated as the current default will be used in expanding all filenames (e.g.: "LOG1 subvol.log1").

Appendix B Migrator

B.1 Introduction

This Appendix describes the migration facility provided as part of the MultiBatch system. This facility allows the relocation of data between MultiBatch database environments, without the need for update of data via the MultiBatch Pathway. The migration facility will be referred to as Migrator throughout this Appendix.

The Migrator is executable in one of three modes, namely:

- EXTRACT
- INSERT
- REMOVE

Although each mode performs a distinct function they will usually be used together (either two or all three) to provide a full complement of relocation facilities for MultiBatch environments.

Each Migrator mode performs its operation using a MultiBatch database and a transition file.

In the EXTRACT and REMOVE modes, data is sourced from a MultiBatch database with selected data inserted into the transition file.

In INSERT mode, the data is copied from the transition file to the target MultiBatch database, with any value substitution taking place at this time. The transition file may subsequently be deleted, or retained to provide audit trail and backout facilities.

On insertion, the transit file must have been created at the same level as the extraction/removal. For example, insertion at job level can only be performed using a transit file produced by a job level extraction or removal.

Note: *The Transition file must be an Entry Sequence file with a REC size of 4062.*

B.2 Migrator Processes

The Migrator process uses only one TMF transaction. The number of locks allowed in this TMF transaction should be specified using the TMF parameter:

PARAM MAX-LOCKS NNNN

where NNNN indicates the maximum number of record locks allowed in a single TMF transaction. NNNN defaults to 0 indicating that full file locking should be used. This means that any problems in the migration will result in a full backout and the database will be restored to its previous state. If the full file locking option is selected, you will have to FREEZE and STOP any PATHWAY servers which have your target database open.

In addition to the TMF Record Locking parameter the following parameters (Default N) can be input from the TACL prompt before the Migrator is used:

PARAM UPDATE-CAL-PARAM	**Y/N**	**(PARAMCAL file)**
PARAM UPDATE-ASSIGN-CLASS	**Y/N**	**(ASSCLASS file)**
PARAM UPDATE-DEFINE-CLASS	**Y/N**	**(DEFCLASS file)**
PARAM UPDATE-PARAM-CLASS	**Y/N**	**(PARCLASS file)**
PARAM UPDATE-DEFINE	**Y/N**	**(DEFCONF file)**
PARAM UPDATE-PROMPT-CLASS	**Y/N**	**(PRMCLASS file)**
PARAM UPDATE-ENV-CLASS	**Y/N**	**(STEPOSEC file)**
PARAM UPDATE-FD-CLASS	**Y/N**	**(STEPOSFC file)**
PARAM UPDATE-EVENT-TIMER	**Y/N/S**	**(SCHEDULE file)**
PARAM PREVIEW-CHANGES	**Y/N**	
PARAM PREVIEW-CHANGES-REPORT	**Y/N**	

For example: **PARAM UPDATE-DEFINE Y**

These allow the user to indicate if, on insertion, certain database files are to be updated if a duplicate key is encountered - Y indicates update is to be performed.

The UPDATE-EVENT-TIMER param has a value of Y/N/S:

EXTRACT process

Y - During Extract, the EventTimer/SCHEDULE record(s) are extracted
N - During Extract, no EventTimer/SCHEDULE records are extracted
S - During Extract, the EventTimer/SCHEDULE record(s) are extracted

INSERT process

Y - During Insert, the EventTimer/SCHEDULE status is changed to H (Held)

N - During Insert, no EventTimer/SCHEDULE records are inserted

S - During Insert, the EventTimer/SCHEDULE status remains unchanged

These parameters are necessary as the specified files (paramcal, assclass, defclass, parclass, defconf, steposfc and steposec) do not have records deleted by the Migrator during the Remove mode. Housekeeping procedures may need to be developed to clear any associated entries.

The following parameters will report the effect of using the alias substitution facility.

PARAM PREVIEW-CHANGES	**Y/N**	(default N)
PARAM PREVIEW-CHANGES-REPORT	**<spooler location>**	(default $S.#ALIAS)

The report will contain details of old and new aliases and converted aliases that will be duplicate or longer than 30 characters.

This feature is only valid when running the Migrator in INSERT mode. No database updates will be performed. The Migrator operates at the BMON, Job, Segment and Unit levels.

To run the Migrator, the following is input at the TACL prompt:

```
RUN MIGRATOR / IN <Database Svol>.DATACONF, OUT <vol>.<subvol>.TRANSIT /{MODE}{LEVEL} &
[,NODE <replacement node>] [,PROCESS <replacement process>]
```

where MODE is **EXTRACT**, **INSERT** or **REMOVE**

LEVEL is:

[<old-node>.]<process name> for BMON level.

[<old-node>.]<process name>.<jobname> for JOB level.

[<old-node>.]<process name>.<jobname>.<segname> for SEGMENT level

[<old-node>.]<process name>.<jobname>.<segname>.<unit name> for UNIT level.

The replacement Node and/or Process are only applicable to the INSERT option. If either are specified, then the MIGDEFS file is not accessed.

EXTRACT

The EXTRACT mode allows data to be extracted from a source MultiBatch BMON environment\Job\Segment\Unit into a transition file. All data relating to the specified BMON environment\Job\Segment\Unit will be selected from the MultiBatch database files, and entered into a single transition file. This mode does not perform any data modification for node names, process names, disk volumes, etc. The files that will be accessed by the EXTRACT mode are given later in this Appendix.

INSERT

Using Migrator's INSERT mode allows data to be moved from a transition file to the target MultiBatch database. All records contained within the transition file which relate to the named BMON process will be inserted into the target database. This mode also incorporates data modification to allow installation values to be included for node names, process names, disk volumes, BMON names and aliases. Values which are to be replaced may be specified as part of the run command (for single node and/or single process name updates) or by using a definition file (for multiple replacement values), as follows:

- The definition file must be called MIGDEFS, residing in the same subvolume as the specified DATACONF file.
- The MIGDEFS file must be an edit type file.
- The file can contain only comment lines indicated by an "*" in column 1 and/or node\volume\subvolume\ BMON replacement entities, again starting in column 1.
- The file is terminated by a blank line.

An example of a definitions file (MIGDEFS) is provided on the supplied database subvolume, e.g. MBATDAT.

The INSERT mode will terminate in one of two ways, when the PREVIEW-CHANGES parameter is set to N (i.e. Update):

- Normal termination occurs when the end of the specified transition file is detected.
- Should a file error be encountered, an abnormal termination is reported, and the insert phase ends. Data already inserted into the target area will not be removed, if this is required it can be achieved by use of the REMOVE function on the target database.

REMOVE

The REMOVE mode performs a similar operation to the EXTRACT mode, with the addition of data deletion from the specified database. Hence all data that is selected from the database, and placed into the transition file, is also deleted from the MultiBatch database. This feature will allow removal of an existing environment to a fallback file, prior to release of an enhanced environment

The operations which can be carried out using the Migrator modes, together with examples, are detailed in the following sections.

Examples:

```
RUN MIGRATOR/in DATACONF, out TEMP.TRANSIT/INSERT \DEV.$BMON , NODE \LIVE

RUN MIGRATOR/in DATACONF, out TEMP.TRANSIT/EXTRACT \LIVE.$BMON

RUN MIGRATOR/in DATACONF, out TEMP.TRANSIT/REMOVE \TEST.$BMON

RUN MIGRATOR/in DATACONF, out TEMP.TRANSIT/INSERT $BMON.JOBC.AA.01
```

B.3 Operations Provided

The Migrator modes can be used to:

- Copy data from one environment to another.
- Replace an existing BMON environment\Job\Segment\Unit

- Insert a BMON environment\Job\Segment\Unit. This assumes that a transition file already exists.
- Remove a specified environment from a node.

In the examples given it is assumed that data is to be moved from a test environment node (node \TEST) to a live environment (node \LIVE) for a BMON process called $BMON.

All MultiBatch files to be accessed are identified from the DATACONF file which is always specified as the IN file when running the Migrator facility. The transition file is always specified as the OUT file, regardless of the mode
being performed.

EXTRACT

To copy data from one environment is achieved by using the following Migrator modes:

- EXTRACT \TEST.$BMON to transition file ONE. This copies the \TEST.$BMON environment into a transition file. The \TEST environment database remains unchanged.
- INSERT \TEST.$BMON using transition file ONE replacing \TEST with \LIVE. The data from transition file ONE is inserted into the \LIVE environment (identified by the IN dataconf file). Prior to the writing any record, \TEST is replaced with \LIVE.

REPLACE

To replace an existing Live BMON environment by an environment from a test database is achieved using the following Migrator modes:

- REMOVE \LIVE.$BMON to transition file ONE. This clears the live database of all \LIVE.$BMON data, and provides a transition file which may be used for subsequent fallback.
- EXTRACT \TEST.$BMON to transition file TWO. This copies the test \TEST.$BMON environment into a second transition file. The \TEST environment remains unchanged.
- INSERT \TEST.$BMON using transition file TWO, replacing \TEST with \LIVE. The data from transition file TWO is inserted into the \LIVE environment (identified by the IN dataconf file). Prior to the writing of any record, \TEST will be replaced with \LIVE.

INSERT

In order to insert a BMON environment a transition file must already exist from which the insertion can be performed.

Typically, stand-alone insertion would be used when a transition file had been created by an earlier REMOVE or EXTRACT. The insertion is achieved as follows:

- INSERT \TEST.$BMON using transition file ONE, replacing \TEST with \LIVE. The data from transition file ONE will be inserted into the \LIVE environment. Prior to the writing of any record, \TEST is replaced with \LIVE.

REMOVE

Stand alone removal of data allows specified environments to be deleted from a node for housekeeping purposes. A removal of data is achieved as follows.

- REMOVE \LIVE.$BMON to transition file one. This clears the live database of all \LIVE.$BMON data, and provides a transition file which may be used for subsequent fallback.

When the Migrator terminates successfully the following processing summary message is displayed on the Migrator home terminal.

```
Migrator - <Operation> \LIVE.$BMON: Number of Jobs processed = nnn Migrator -
<Operation> \LIVE.$BMON: Number of Segs processed = nnn Migrator - <Operation>
\LIVE.$BMON: Number of Steps processed = nnn Migrator - <Operation> \LIVE.$BMON:
Complete.
```

<Operation> can be Extract, Remove or Insert

B.4 Migrator Files

The following files are accessed by MultiBatch and updated as required during the INSERT mode:

- ALIASMAP
- ASSCLASS
- BMONCONF
- DEFCLASS
- DEFCONF
- JOBCONF
- JOBDEP
- PARAMCAL
- PARCLASS
- PREPCONF
- SEGCONF
- SEGDEP
- STEPASS
- STEPCONF
- STEPDEF
- STEPOSS
- STEPOSF
- STEPOSE
- STEPOSFC
- STEPOSFE
- STEPPAR
- STEPPROM
- STPROMPT

Not all the data fields in these files are updated during INSERT. The following fields are not updated (please note, this is not a definitive list):

File Name	**Field Name**
BMONCONF	BMON Description
JOBCONF	Job Description
PREPCONF	Prepare Name
SEGCONF	Segment Description
STEPCONF	Group Name (re-runnable feature) / Step Description
(Text) STEPPAR	Parameter Values.

Appendix C Datafile Configuration

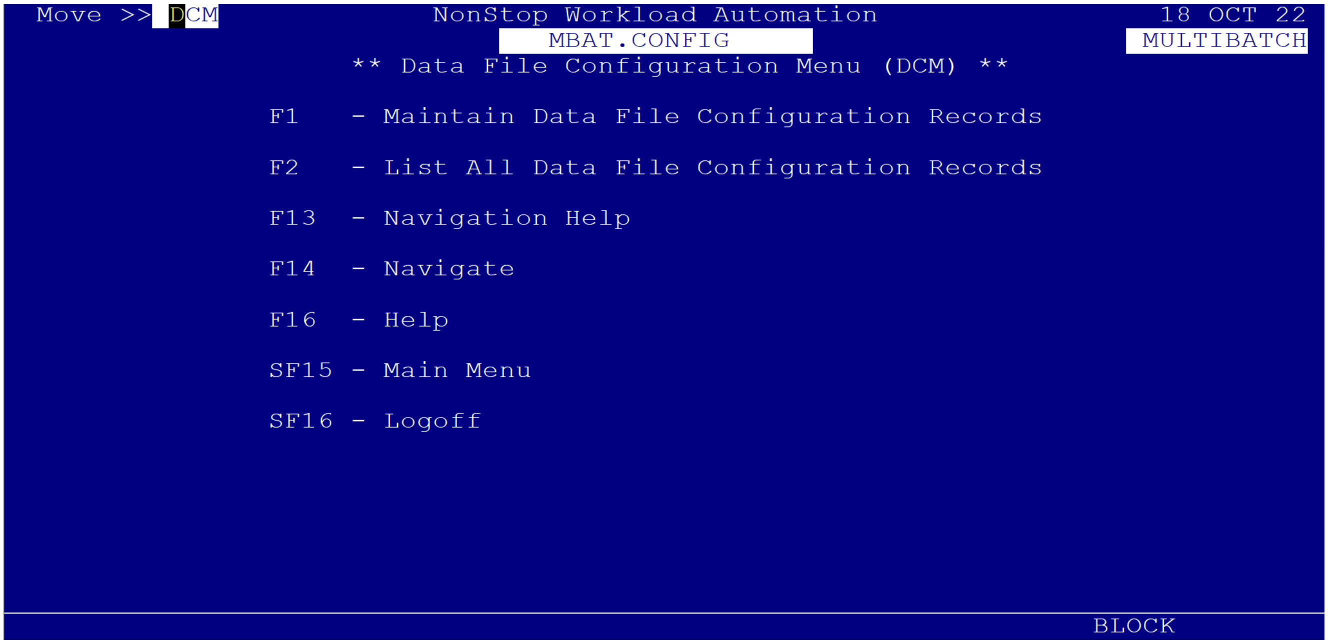
C.1 Introduction (DCM)

The Data File Configuration facility (DCC) is used to modify the DATACONF file which holds "pointers" to the files in the MultiBatch database. This facility in not normally used by the user as once the MultiBatch software is loaded, DATACONF should then be correct for the system it is running on.

Optionally, a user may wish to utilise an Alternate Collector for receiving MultiBatch EMS events. If so, see **Chapter 5.2 Primary / Alternate Collector Usage** which describes how to set up additional DCC entries for this purpose.

The facility is accessed via the Navigate facility by pressing the **F10** function key on the Main Menu Screen (Figure 3.1).

When first accessed the user is presented with the Data File Configuration Menu (DCM) Screen (Figure C.1).



```
  Move >> DCM            NonStop Workload Automation            18 OCT 22
                               MBAT.CONFIG                   MULTIBATCH
                    ** Data File Configuration Menu (DCM) **

                F1   - Maintain Data File Configuration Records

                F2   - List All Data File Configuration Records

                F13  - Navigation Help

                F14  - Navigate

                F16  - Help

                SF15 - Main Menu

                SF16 - Logoff





                                                                BLOCK
```

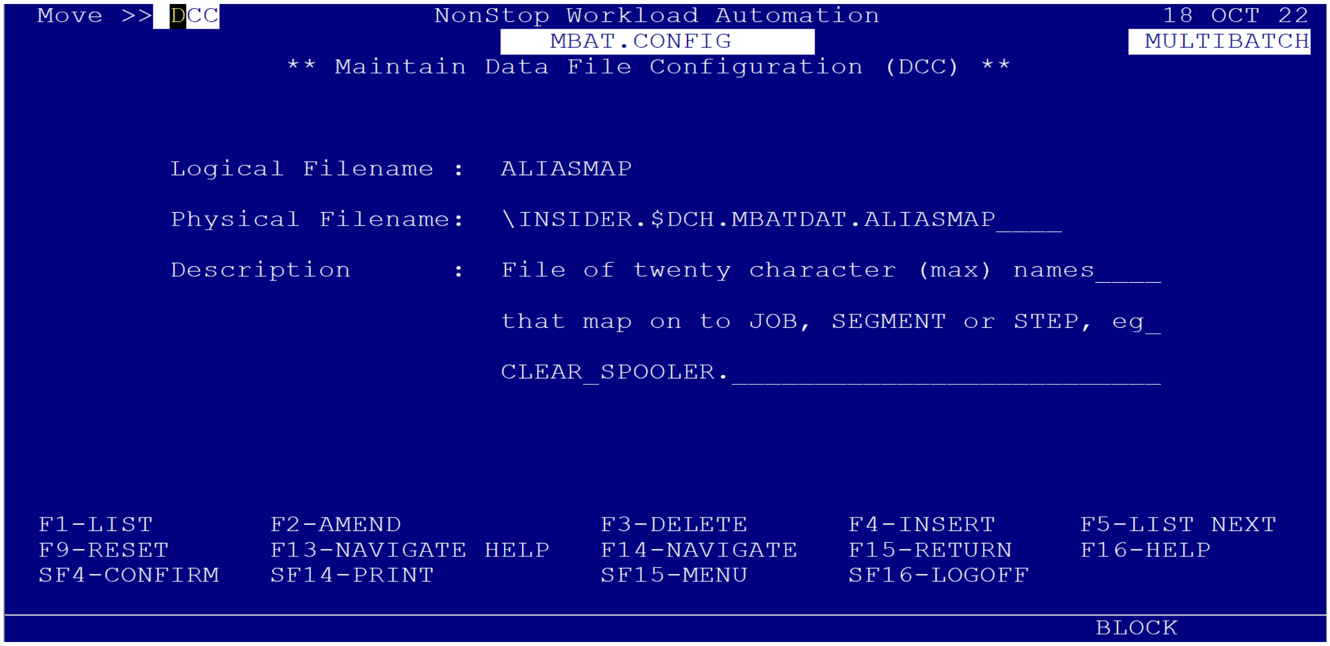
Figure C.1 - Data File Configuration Menu (DCM)

Apart from the standard function keys (HELP, NAVIGATE etc.) the user is presented with two options:

- Maintain Data File Configuration Records.
- List All Data File Configuration Records.

C.2 Maintain Datafile Configuration Records (DCC)

Pressing the **F1** function key displays the Maintain Data Configuration Screen (DCC) (Figure C.2).



```
 Move >> DCC               NonStop Workload Automation                   18 OCT 22
                                  MBAT.CONFIG                          MULTIBATCH
                  ** Maintain Data File Configuration (DCC) **


         Logical Filename :   ALIASMAP

         Physical Filename:   \INSIDER.$DCH.MBATDAT.ALIASMAP____

         Description      :   File of twenty character (max) names____

                              that map on to JOB, SEGMENT or STEP, eg_

                              CLEAR_SPOOLER.__________________________



 F1-LIST        F2-AMEND             F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET       F13-NAVIGATE HELP    F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM    SF14-PRINT           SF15-MENU      SF16-LOGOFF

                                                                 BLOCK
```

Figure C.2 - Maintain Data File Configuration Records (DCC)

This screen allows the user to Add/Delete/Amend Data File Configuration Records. The user is presented with the following options:

F1-LIST. When the Logical Filename is input this function key is used to display the associated details for the specified name.

F2-AMEND, **F3**-DELETE and **F4**-INSERT. These function keys allow the user to amend/delete records in the database and to add new records.

F5-LIST NEXT. This key steps the user through the database records one at a time.

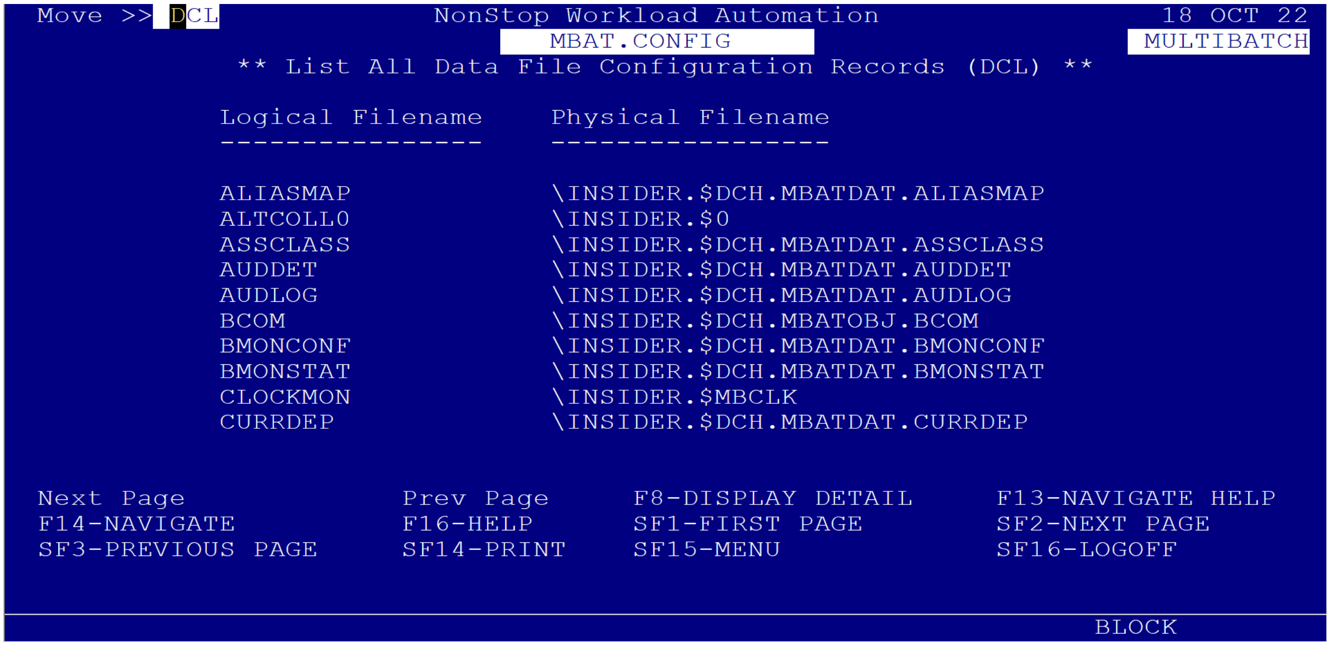
F9-RESET. This key resets all fields to "blank" to allow the user to insert a new record.

SF4-CONFIRM. This key is a fail-safe device associated with the **F3** function key. When the user tries to delete a record from the database, the user is prompted to confirm his decision to delete before any action is taken.

C.3 List All Data Configuration Records (DCL)

Pressing the **F2** function key at the Data File Configuration Menu screen (See Figure C.1) displays the List All Data File Configuration Records Screen (DCL) (Figure C.3).

This screen displays a list (first page) of all Data File Configuration Records in the database. The function key and the Page Up/ Down keys can be used to step through the list. Placing a marker by a record and pressing the **F8**-DETAIL function key displays the associated Maintain Data File Configuration Record Screen (See Figure C.2).



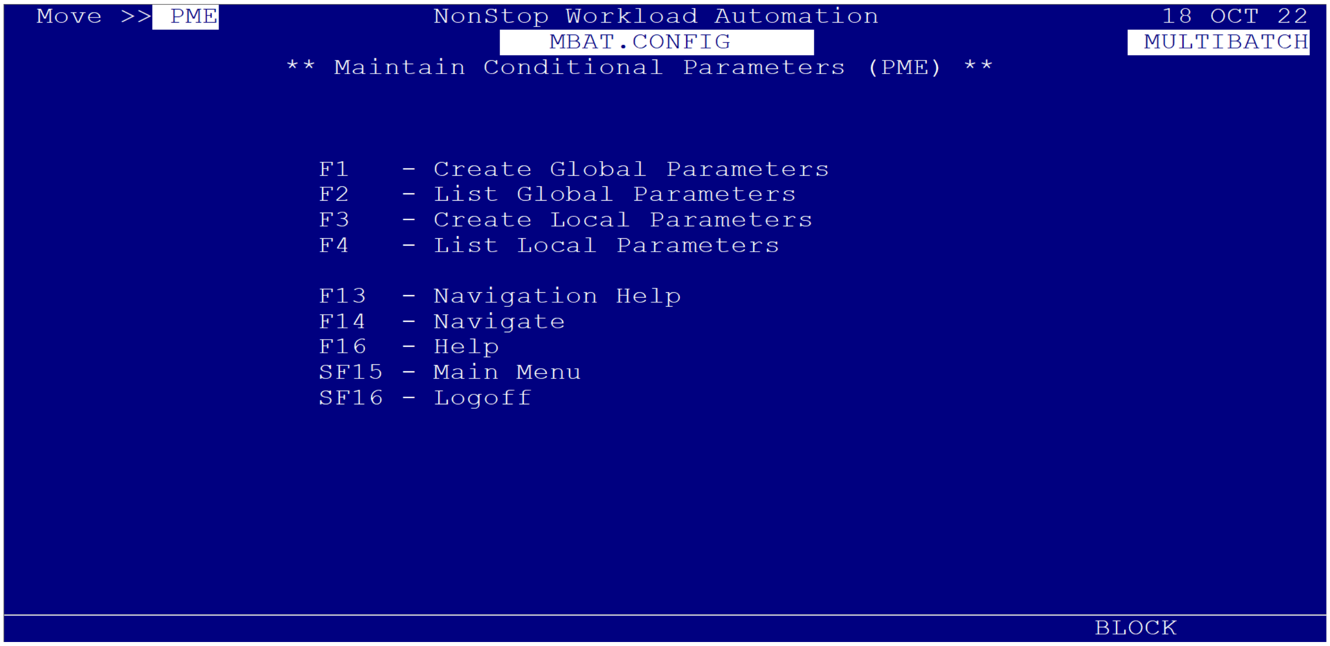
```
 Move >>  DCL             NonStop Workload Automation                  18 OCT 22
                              MBAT.CONFIG                              MULTIBATCH
                ** List All Data File Configuration Records (DCL) **

             Logical Filename     Physical Filename
             ----------------     -----------------

             ALIASMAP             \INSIDER.$DCH.MBATDAT.ALIASMAP
             ALTCOLL0             \INSIDER.$0
             ASSCLASS             \INSIDER.$DCH.MBATDAT.ASSCLASS
             AUDDET               \INSIDER.$DCH.MBATDAT.AUDDET
             AUDLOG               \INSIDER.$DCH.MBATDAT.AUDLOG
             BCOM                 \INSIDER.$DCH.MBATOBJ.BCOM
             BMONCONF             \INSIDER.$DCH.MBATDAT.BMONCONF
             BMONSTAT             \INSIDER.$DCH.MBATDAT.BMONSTAT
             CLOCKMON             \INSIDER.$MBCLK
             CURRDEP              \INSIDER.$DCH.MBATDAT.CURRDEP


 Next Page              Prev Page     F8-DISPLAY DETAIL     F13-NAVIGATE HELP
 F14-NAVIGATE           F16-HELP      SF1-FIRST PAGE        SF2-NEXT PAGE
 SF3-PREVIOUS PAGE      SF14-PRINT    SF15-MENU             SF16-LOGOFF


                                                                  BLOCK
```

Figure C.3 - List All Data File Configuration Records (DCL)

Appendix D Conditional Processing

D.1 Introduction (PME)

Chapter 4 showed how conditional parameters can be used in building schedules. This facility can be accessed via the **F9** function key from the Main Menu (MEN) or by navigating to **PME**.



```
  Move >> PME                NonStop Workload Automation                   18 OCT 22
                                    MBAT.CONFIG                           MULTIBATCH
                     ** Maintain Conditional Parameters (PME) **



                       F1    - Create Global Parameters
                       F2    - List Global Parameters
                       F3    - Create Local Parameters
                       F4    - List Local Parameters

                       F13   - Navigation Help
                       F14   - Navigate
                       F16   - Help
                       SF15  - Main Menu
                       SF16  - Logoff







                                                                    BLOCK
```

Figure D.1 - Maintain MultiBatch Parameter Screen (PME)

Four screens are provided which allow the user to:

- Create and maintain Global parameters.
- List Global parameters.
- Create and maintain Local parameters.
- List Local parameters.

Information can be obtained from the HELP database by pressing the F16 function key on any screen.

D.2 List Global Parameters

The List All Global Parameter Records Screen (Figure D.2) is accessed by pressing the **F2** function key on the PME Menu screen (Figure D.1).

Figure D.2 - List All Global Parameter Records Screen

When accessed the first page of Global parameters defined in the database is displayed. The following function keys are available:

SF1-FIRST, **SF2**-NEXT and **SF3**-PREVIOUS. These keys allow the user to step through the database records by displaying the first page of records and the next / previous page respectively.

F8-DETAIL. The user can place a marker (any character) by an entry in the list (column 3) and press the F8 function key to obtain more detail on the entry. This facility displays the Maintain Global Parameter File for the specified parameter (Figure D.5).

F13-NAV HELP. This display the list of available facilities.

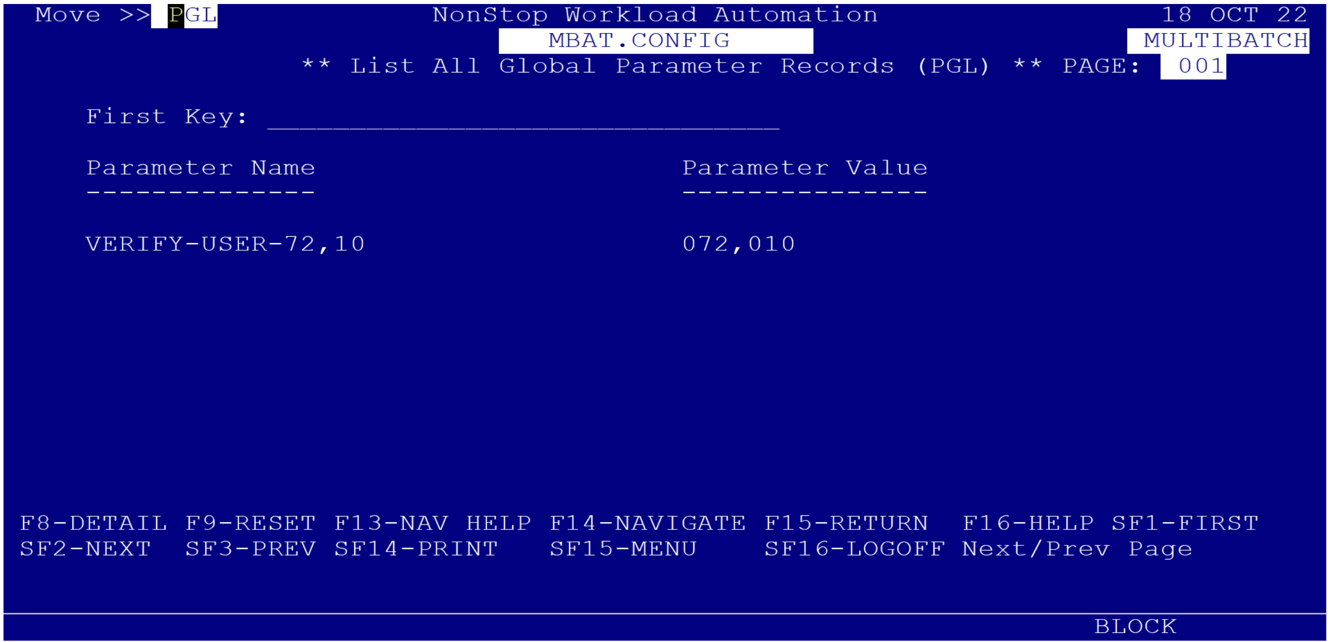
F14-NAVIGATE. Enter the three letter facility in the Move field and press F14 to navigate to this facility.

F16-HELP. This displays details of the various fields available on the screen.

SF15-MENU. This returns the user to the Menu Screen (Figure D.1).

SF16-LOGOFF. This logs the user out of the MultiBatch Pathway.

A maximum 100 pages of Global Parameter Records is allowed.



```
 Move >> PGL             NonStop Workload Automation                18 OCT 22
                                 MBAT.CONFIG                         MULTIBATCH
              ** List All Global Parameter Records (PGL) ** PAGE:  001

    First Key: _______________________________

    Parameter Name                      Parameter Value
    --------------                      ---------------

    VERIFY-USER-72,10                   072,010








 F8-DETAIL F9-RESET F13-NAV HELP F14-NAVIGATE F15-RETURN  F16-HELP SF1-FIRST
 SF2-NEXT  SF3-PREV SF14-PRINT   SF15-MENU    SF16-LOGOFF Next/Prev Page

                                                                 BLOCK
```

D.3 Maintain Global Parameters

The Maintain Global Parameter Screen (Figure D.3) is accessed by pressing the **F1** function key or by selecting DETAIL from the List All Global Parameter Record Screen (Figure D.2). If the screen is accessed from the Menu screen, then all the displayed fields are blank. This screen allows the user to generate new Global Parameters in the database and amend/delete current records. The user has several options available:

The user can enter data in the appropriate fields (if the record was blank) and press the **F4** function key to Insert a new record into the database.

Can amend/delete the existing record (if one is displayed) using the **F2/F3** function keys.

Note: *When the **F3**-DELETE function key is used the user is prompted to confirm that this action is really required, by pressing the **SF4**-CONFIRM function key.*

The operator can press the **F5** function key which displays the first record in the database. This record can then be amended/deleted using the **F2/F3** function keys.

The operator can press the **F9** function key to reset all fields to BLANK. In addition, the following function keys are available:

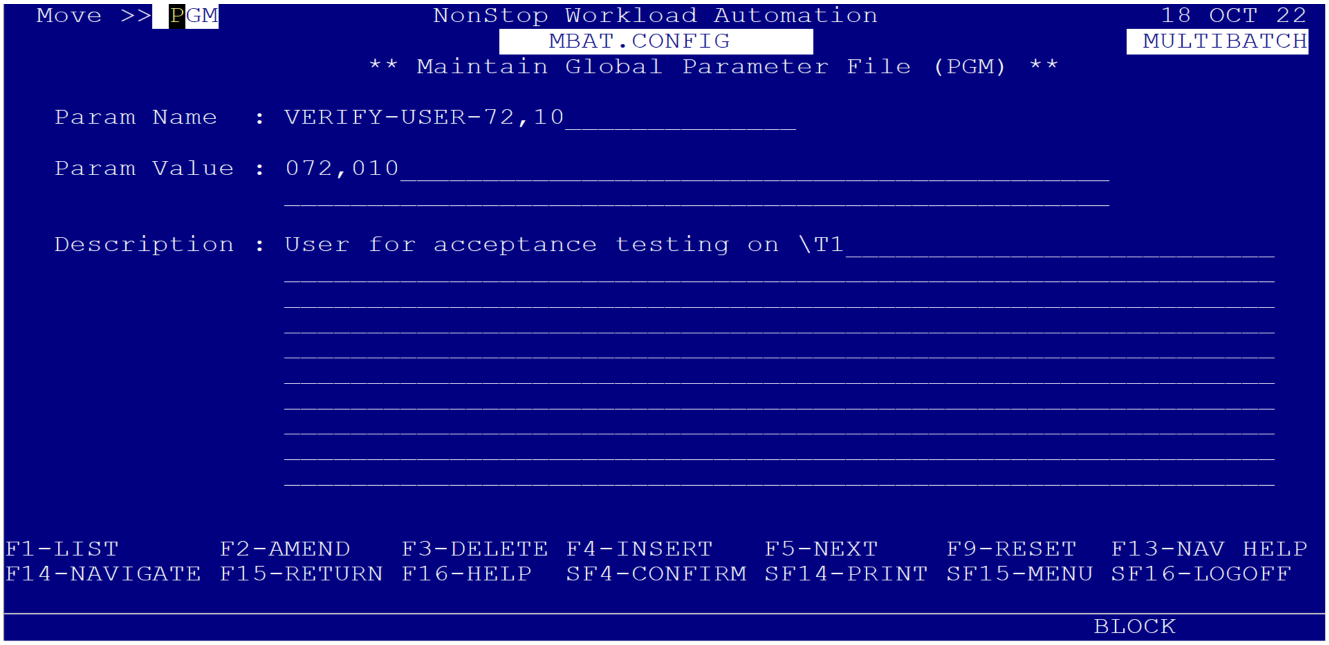
F13-NAV HELP. This display the list of available facilities.

F14-NAVIGATE. Enter the three letter facility in the Move field and press F14 to navigate to this facility.

F16-HELP. This displays details of the various fields available on the screen.

SF15-MENU. This returns the user to the Menu Screen (Figure D.1).

SF16-LOGOFF. This logs the user out of the MultiBatch Pathway.

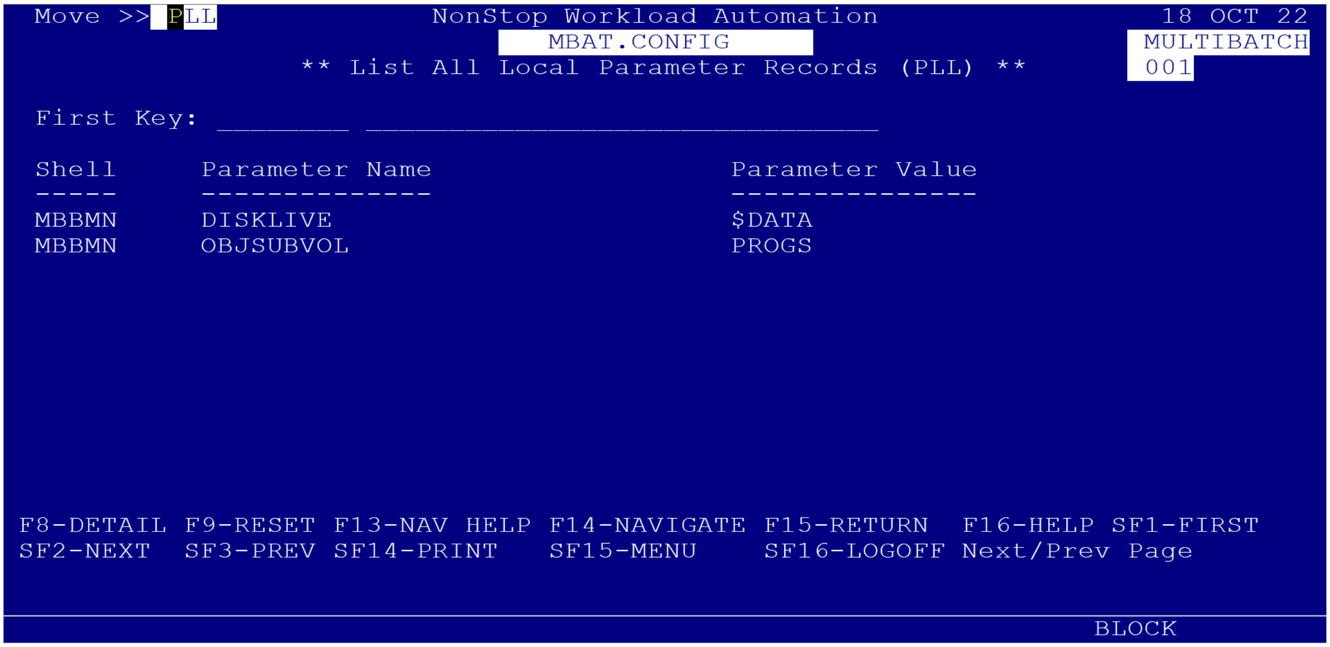


```
 Move >> PGM              NonStop Workload Automation              18 OCT 22
                               MBAT.CONFIG                        MULTIBATCH
                    ** Maintain Global Parameter File (PGM) **

  Param Name  : VERIFY-USER-72,10______________

  Param Value : 072,010____________________________________________
                ___________________________________________________

  Description : User for acceptance testing on \T1__________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________

F1-LIST      F2-AMEND   F3-DELETE F4-INSERT  F5-NEXT    F9-RESET   F13-NAV HELP
F14-NAVIGATE F15-RETURN F16-HELP  SF4-CONFIRM SF14-PRINT SF15-MENU SF16-LOGOFF

                                                                 BLOCK
```

Figure D.3 - Maintain Global Parameter File Screen

D.4 List Local Parameters

The List All Local Parameter Records Screen (Figure D.4) is accessed by pressing the **F4** function key on the Menu screen (Figure D.1).



```
 Move >> PLL               NonStop Workload Automation               18 OCT 22
                              MBAT.CONFIG                           MULTIBATCH
                ** List All Local Parameter Records (PLL) **         001

 First Key: ________ _______________________________

 Shell     Parameter Name                  Parameter Value
 -----     --------------                  ---------------
 MBBMN     DISKLIVE                        $DATA
 MBBMN     OBJSUBVOL                       PROGS

F8-DETAIL F9-RESET F13-NAV HELP F14-NAVIGATE F15-RETURN  F16-HELP SF1-FIRST
SF2-NEXT  SF3-PREV SF14-PRINT   SF15-MENU    SF16-LOGOFF Next/Prev Page

                                                                  BLOCK
```

Figure D.4 - List All Local Parameter Records Screen

When accessed the first page of Local parameters defined in the database is displayed. The following function keys are available:

SF1-FIRST, **SF2**-NEXT and **SF3**-PREVIOUS. These keys allow the user to step through the database records by displaying the first page of records and the next/previous page respectively.

F8-DETAIL. The user can place a marker (any character) by an entry in the list (column 3) and press the **F8** function key to obtain more detail on the entry. This facility displays the Maintain Global Parameter File for the specified parameter (Figure D.5).

F13-NAV HELP. This display the list of available facilities.

F14-NAVIGATE. Enter the three letter facility in the Move field and press F14 to navigate to this facility.

F16-HELP. This displays details of the various fields available on the screen.

SF15-MENU. This returns the user to the Menu Screen (Figure D.1).

SF16-LOGOFF. This logs the user out of the MultiBatch Pathway.

A maximum of 100 pages of pages of Local Parameter Records is allowed.

D.5 Maintain Local Parameters

The Maintain Local Parameter Screen (Figure D.5) is accessed by pressing the **F3** function key or by selecting DETAIL from the List All Local Parameter Record Screen (Figure D.4). If the screen is accessed from the Menu screen, then all the displayed fields are blank. This screen allows the user to generate new Local Parameters in the database and amend/delete current records. The user has several options available:

The user can enter data in the appropriate fields (if the record is blank) and press the **F4** function key to Insert a new record into the database.

The user can amend/delete the existing record (if one is displayed) using the **F2/F3** function keys.

Note: *When the **F3**-DELETE function key is used the user is prompted to confirm that this action is really required, by pressing the **SF4**-CONFIRM function key.*

The user can press the F5 function key which displays the first record in the database. This record can then be amended/deleted using the F2/F3 function keys.

The user can press the F9 function key to reset all fields to BLANK. In addition, the following function keys are available:

F13-NAV HELP. This display the list of available facilities.

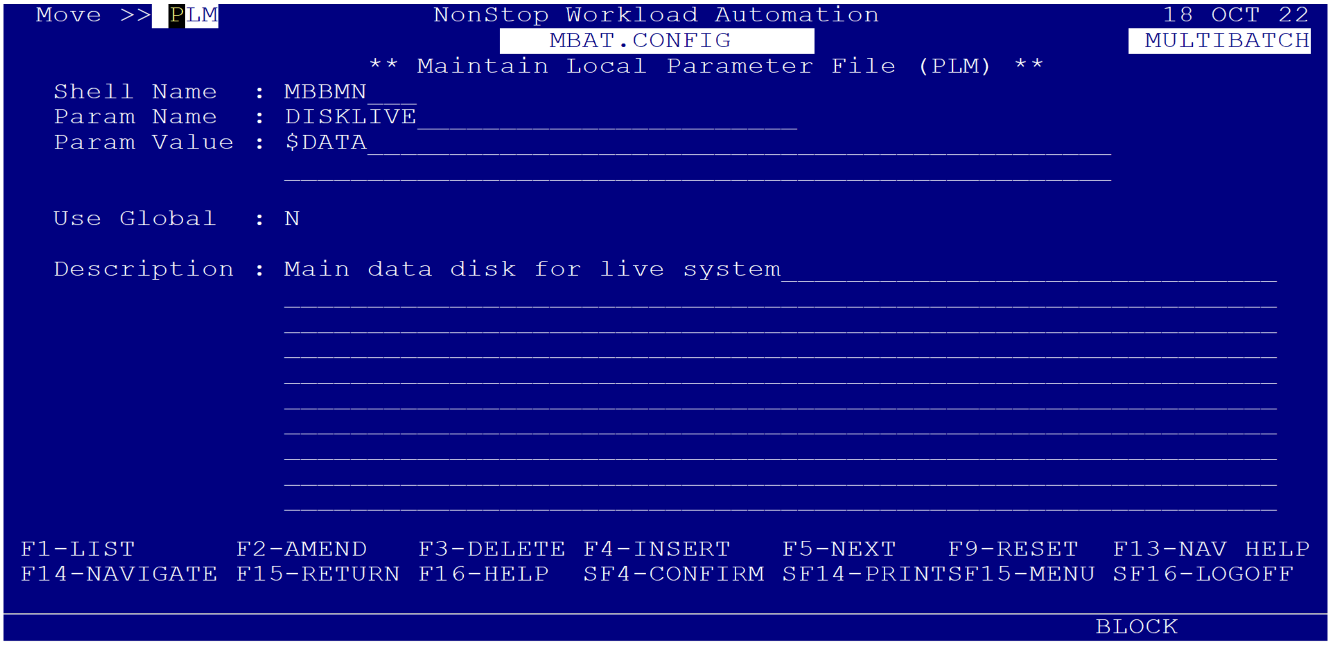
F14-NAVIGATE. Enter the three letter facility in the Move field and press F14 to navigate to this facility.

F16-HELP. This displays details of the various fields available on the screen.

SF15-MENU. This returns the user to the Menu Screen (Figure D.1).

SF16-LOGOFF. This logs the user out of the MultiBatch Pathway.

Note: *The Use Global field must be specified as Y or N. If this field is set to Y, then the parameter must exist on the Global Param file.*



```
 Move >> PLM              NonStop Workload Automation            18 OCT 22
                              MBAT.CONFIG                       MULTIBATCH
                    ** Maintain Local Parameter File (PLM) **
  Shell Name  : MBBMN___
  Param Name  : DISKLIVE_______________________
  Param Value : $DATA_______________________________________________
                ____________________________________________________

  Use Global  : N

  Description : Main data disk for live system______________________________
                ______________________________________________________________
                ______________________________________________________________
                ______________________________________________________________
                ______________________________________________________________
                ______________________________________________________________
                ______________________________________________________________
                ______________________________________________________________
                ______________________________________________________________
                ______________________________________________________________

 F1-LIST      F2-AMEND   F3-DELETE F4-INSERT  F5-NEXT   F9-RESET  F13-NAV HELP
 F14-NAVIGATE F15-RETURN F16-HELP  SF4-CONFIRM SF14-PRINTSF15-MENU SF16-LOGOFF

                                                                 BLOCK
```

Figure D.5 - Maintain Local Parameter File Screen

D.6 Use of SKIP Param with GROUPID

When a job or multiple jobs are configured in the CJB screen with a GROUPID name, then these are automatically given the 'Rerunnable' attribute.

This means that when all the JOBS in the same GROUPID are complete, the FLAGS of ALL underlying STEPS will be automatically reset to %000000 (Waiting), enabling a user to run a sequence of JOBs many times without having to reload the BMON environment.

This Appendix describes the impact of using SKIP (or RUNOFF) when jobs are configured with a GROUPID.

Consider for example three JOBs all with the same GROUPID and a single job dependency between each:

Job SKP1 starts and completes.

This is followed by the start and completion of successive dependencies, with all steps resetting themselves so that they are ready to run again:

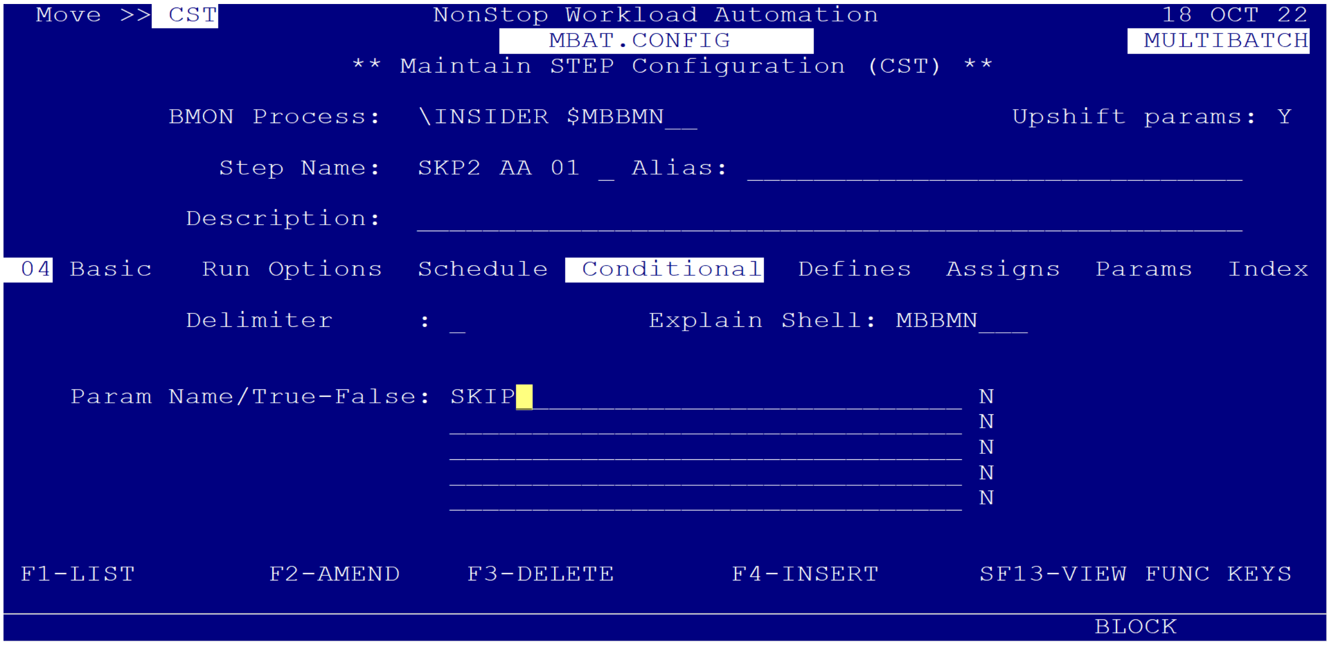
```
18-06-13 09:43:13 \LIVE.$MSTR MULTIBAT.1.0    000082 SKP1.**.**.*, job has been
reset to a configured state,\LIVE.$BMON 13/06/18, 09:43

18-06-13 09:44:13 \LIVE.$MSTR MULTIBAT.1.0    000082 SKP2.**.**.*, job has been
reset to a configured state, \LIVE.$BMON 13/06/18, 09:44

18-06-13 09:45:13 \LIVE.$MSTR MULTIBAT.1.0    000082 SKP3.**.**.*, job has been
reset to a configured state, \LIVE.$BMON 13/06/18, 09:45
```

Note: *The BMON keeps an internal record of the status of jobs that have the same GROUPID. If a GROUPID job fails, then the BMON records the GROUP of jobs as Incomplete. Only on successful completion of ALL steps, will BMON record the GROUP as being complete.*

Now, consider the impact of skipping a job, e.g. the middle job – SKP2 – is skipped via use of a BCOM $<BMON> ALTER STEP SKP2.AA.01 RUNOFF command, or via a Local Conditional Parameter (PME) entry containing, for example:

```
      Shell Name  : BMON
      Param Name  : SKIP
      Param Value : TRUE
```

The SKIP Param Name – SKIP - is configured in the CST Conditional Overlay as:



```
  Move >> CST               NonStop Workload Automation              18 OCT 22
                                   MBAT.CONFIG                      MULTIBATCH
                         ** Maintain STEP Configuration (CST) **

          BMON Process:  \INSIDER $MBBMN__                    Upshift params: Y

             Step Name:  SKP2 AA 01 _ Alias: ______________________________

           Description:  ________________________________________________

 04 Basic   Run Options  Schedule  Conditional  Defines  Assigns  Params  Index

           Delimiter      : _           Explain Shell: MBBMN___


    Param Name/True-False: SKIP_____________________________ N
                           _________________________________ N
                           _________________________________ N
                           _________________________________ N
                           _________________________________ N


 F1-LIST        F2-AMEND     F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                  BLOCK
```

Figure D.6 - CST SKIP Param

Once job SKP1 starts and completes, the next dependent job SKP2 is skipped and is marked in the BMON as follows:

```
******JOB SKP2   Numsegs  = 001, Segs  running = 000, Segs  complete = 001
 *SEGMENT AA..   Numunits = 001, Units running = 000, Units complete = 001
   **UNIT --01   Restarts = 000, Flags = %000001                >> SKIP <<
                  STEP1 - Step Complete
```

Therefore, the step has not run; not marked as complete; not reset back to %000000 within the BMON GROUPID table.

Result: job SKP1 completes and resets to configured after it runs but the next dependent job, after SKP2, SKP3 will not start and remains at configured.

An STATUS JOB SKP* provides:

```
>>STATUS JOB SKP*
******JOB SKP1   Numsegs  = 001, Segs  running = 000, Segs  complete = 000
 *SEGMENT AA..   Numunits = 001, Units running = 000, Units complete = 000
   **UNIT --01   Restarts = 000, Flags = %000000
                  STEP1 - Configured

******JOB SKP2   Numsegs  = 001, Segs  running = 000, Segs  complete = 001
 *SEGMENT AA..   Numunits = 001, Units running = 000, Units complete = 001
   **UNIT --01   Restarts = 000, Flags = %000001                >> SKIP <<
                  STEP1 - Step Complete

******JOB SKP3   Numsegs  = 001, Segs  running = 000, Segs  complete = 000
 *SEGMENT AA..   Numunits = 001, Units running = 000, Units complete = 000
   **UNIT --01   Restarts = 000, Flags = %000000
                  STEP1 - Configured
```

Appendix E Messages, Responses and Event Messages

E.1 Introduction

This appendix contains a list of all error and informational messages generated by the MultiBatch system.

The messages are split into two broad groups, namely:

- Error messages.
- Information messages.

Error Messages generated by Event Timer, Streamer, Watcher, Migrator, Clockmon and MbatCom Processes are also provided.

The MultiBatch Pathway server messages will be displayed on line 24 of the user's Pathway session.

E.2 MultiBatch SSIDs

Generally, MultiBatch EMS SSIDs are in the form of either of the following:

a) MULTIBAT.n.0

b) INSIDER.n.0

Specifically, by program:

a) STREAMER on behalf of BMON - MULTIBAT.1.0

b) TRIGGER - INSIDER.6.0

c) CALENDAR - INSIDER.5.0

d) Event Timer Integrity Programs - INSIDER.7.0

e) STREAMER - INSIDER.1.0

f) WATCHER - INSIDER.15.0

g) CLOCKMON - INSIDER.5.0

h) BCOM/MBATCOM - MULTIBAT.2.0

i) SRVADMN - INSIDER.13.0 (failed licence check)

j) GETPRMPT - INSIDER.28.0 and MULTIBAT.1.0

k) MIGRATOR - INSIDER.25.0

l) PREPARE – INSIDER.16.0

m) STATSRV – INSIDER.15.0 and INSIDER.16.0.

Where programs encounter issues with system calls such as those used to set up EMS events TANDEM.12.0 is used.

E.3 BMON - BCOM Messages

These messages are produced by BMON and returned to a user BCOM session and for each, the following information is supplied:

- The error number and message displayed on the screen to the user.
- A brief description of the error and likely causes.

Event Number	Details	Cause / Action
401	Shutdown failed - job(s) still running	This response will be received if "SHUTDOWN BMON " has been issued but there are processes still running. "SHUTDOWN BMON ! " can be used in these circumstances but you need to be the Privileged User ID (see INFO BMON command via BCOM, or view the CBM screen) or user SUPER.SUPER ID in order to execute this command.
402	Initialised protocol unit(s) incomplete	This response will be received if "SHUTDOWN BMON" has been issued but there are processes still running. "SHUTDOWN BMON !" can be used in these circumstances but you need to be the Privileged User ID (see INFO BMON command via BCOM, or view the CBM screen) or user SUPER.SUPER ID in order to execute this command.
403	Fail on log file open - Guardian error <nn>	Attempt to open the BMON log file failed with the indicated Guardian error.
404	BMON already started	Received in response to "START BMON" or "RECOVER BMON" when BMON has already been started.
406	You do not have the authority to execute this command	This error message is concerned with the security status of BCOM. The security settings can be observed via "STATUS BMON". For details of the security system refer to Appendix A regarding "SET BMON SECURITY".
407	Job level dependencies exist for this segments job identity	An attempt to add a segment with a link to another segment has failed. This second segment already has job dependencies. Job and segment level dependencies cannot reside under the same job.
408	Segment level dependencies exist for this job.	An attempt to add a job with a link to another job has failed. This second job already has segment dependencies. Job and segment level dependencies cannot reside under the same job.
409	Issued by multiple BMON Interfaces: BCOM, LIBUTIL, STATSRV, TRIGGER	Status 409 is commonly generated where customer programs interfacing with BMON using LIBUTIL need to be modified to include the latest delivered LIBUTIL procedures. Ensure the same release of BCOM, LIBUTIL, STATSRV or TRIGGER are utilised when accessing BMON.
411	Maximum number of jobs permitted have already been allocated	An "ADD JOB" command has been rejected due to the maximum number of jobs allowed in this BMON configuration, as having already been reached. The "INFO BMON" command will provide details.
412	Maximum number of segments within BMON would be exceeded	An "ADD JOB" command has been rejected due to the maximum number of segments allowed in this BMON configuration, as having already been reached. The "INFO BMON" command will give details. The number of segments allocated within BMON is updated when a Job is added despite the segments being undefined at that time.

413	Job identity specified already exists	An “ADD JOB” command has been rejected due to the specified Job already being known to BMON. “INFO JOB xxxx” will return information regarding the Job.
414	Specified job identity does not exist.	An “ADD SEGMENT JJJJ.SS” command was executed but the job the segment belongs to does not exist.
415	Segment identity already exists for this Job identity	An “ADD SEGMENT” command has been rejected due to the specified Segment already being known to BMON. “INFO JOB xxxx” or “INFO SEGMENT xxxx.yy” will return information regarding the Segment concerned.
416	Maximum number of segments within this Job identity have already been allocated.	An “ADD SEGMENT” command has been rejected due to the maximum number of segments allowed, in this Job, as having already been reached. The “INFO JOB” command will give details.
417	Maximum number of units within BMON would be exceeded	An “ADD SEGMENT” command has been rejected due to the maximum number of units allowed in this BMON configuration, as having already been reached. The “INFO BMON” command will give details. The number of units allocated within BMON is updated when a Segment is added despite the units being undefined at that time.
418	Maximum number of interdependent processes within BMON would be exceeded.	An “ADD SEGMENT” command has been rejected due to the maximum number of interdependent processes allowed, in this BMON configuration, as having already been reached. The “INFO BMON” command will give details. The number of interdependent processes allocated within BMON is updated when a Segment is added despite the interdependent processes being undefined at that time.
419	This Job.Segment does not exist.	An attempt has been made to reference a Job.Segment that is unknown to BMON. The “STATUS JOB xxxx” command may be helpful.
420	Unit identity for this Job.Segment identity already exists	An “ADD STEP xxxx.yy.zz.1” command has been rejected due to the specified Unit already being known to BMON. “INFO JOB xxxx”, “INFO SEGMENT xxxx.yy” or “INFO UNIT xxxx.yy.zz” will return information regarding the Unit concerned.
421	Maximum number of units within this Job.Segment identity have already been allocated	An “ADD STEP xxxx.yy.zz.1” command has been rejected due to the maximum number of units allowed, in this Job.Segment, having already been reached. The “INFO SEGMENT” command will give details.
422	step1 must exist for step2 to be added	The “ADD STEP xxxx.yy.zz.2” command is not accepted until step1 exists.
423	Max num of interdependent processes within this Job. Segment identity have already been allocated	An “ADD STEP xxxx.yy.zz.2” command has been rejected due to the maximum number of interdependent processes allowed, in this Job.Segment, having already been reached. The “INFO SEGMENT” command will give details.
424	step2 for this Job.segment.unit already exists	An “ADD STEP xxxx.yy.zz.2” command has been rejected due to the specified Unit already possessing a step2. “INFO JOB xxxx”, “INFO SEGMENT xxxx.yy” or “INFO UNIT xxxx. yy.zz” will return information regarding the Unit concerned.
425	protocol “ON” not allowed on step2	An attempt to specify a second step with PROTOCOL ON failed after the first unit was added with PROTOCOL OFF.

426	step2 cannot be added after "START UNIT" has been submitted	Once an attempt to start a given Unit has been made that Unit cannot have a step2 added to it. If this causes problems, it is advised that another Unit be added, possibly after deletion of the Unit in error (ref "DELETE UNIT").
427	Cannot specify both segments to be dependent upon each other	Attempt to link a segments to itself.
428	Specified next segment does not exist	Attempt to link a segment to one which does not exist.
429	MAXASSIGNS within BMON would be exceeded	An "ADD STEP" command has been rejected due to the maximum number of assigns allowed in this BMON configuration, as having already been reached. The "INFO BMON" command will give details.
429	MAXDEFINES within BMON would be exceeded	An "ADD STEP" command has been rejected due to the maximum number of defines allowed in this BMON configuration, as having already been reached. The "INFO BMON" command will give details.
429	MAXPARAMS within BMON would be exceeded	An "ADD STEP" command has been rejected due to the maximum number of PARAMs allowed in this BMON configuration, as having already been reached. The "INFO BMON" command will give details
429	MAXENVIRONMENTS within BMON would be exceeded	An "ADD STEP" command has been rejected due to the maximum number of OSS environments allowed in this BMON configuration, as having already been reached. The "INFO BMON" command will give details.
429	MAXFILEDESCRIPTORS within BMON would be exceeded	An "ADD STEP" command has been rejected due to the maximum number of OSS file descriptors allowed in this BMON configuration, as having already been reached. The "INFO BMON" command will give details.
430	this Job.segment.unit does not exist	An attempt has been made to reference a Job.segment.unit that is unknown to BMON. The "INFO JOB xxxx" or "INFO SEGMENT" commands may be helpful.
431	invalid command in current state	This error is returned by various commands where the command is not compatible with the status of the entity. An example where this may happen is an attempt to "START UNIT" when the Unit concerned is already running. A "STATUS" on the entity concerned should be sufficient to ascertain the problem.
432	Newprocess fail on step1	This message will be returned if the "PROCESS_CREATE_" call fails straight away in response to a "START UNIT" command. Information regarding the error can be found with a "STATUS UNIT xxxx.yy.zz , DETAIL" command. It should be noted that the lack of this error message does not guarantee that the Process concerned has been started successfully.
433	Next segment already has maximum previous dependencies	Attempt to link a segment to another which already has the maximum number of dependencies

434	no units have been started	This error is returned in response to a "START JOB" or "START SEGMENT" command when no units have started running. The reason for the failure to start any units is best found with a "STATUS JOB" or "STATUS SEGMENT" command.
435	Job.Segment started with errors	This error is returned in response to a "START SEGMENT" command when not all the units in this Segment have started running. The reason for the failure to start the units concerned is best found with a "STATUS SEGMENT" command.
436	Job started with errors	This error is returned in response to a "START JOB" command when not all the units in this Job have started running. The reason for the failure to start the units concerned is best found with a "STATUS JOB" command.
437	Job held	An attempt has been made to "START" a Job that is in a hold state. For more information refer to "HOLD JOB".
438	Segment held	An attempt has been made to "START" a Segment that is in a held state. For more information refer to "HOLD SEGMENT"
439	Unit held	An attempt has been made to "START" or "RESTART" a Unit that is in a held state. For more information refer to "HOLD UNIT"
440	previous dependent Job for scheduler does not exist	For the Job being added a scheduler reference has been made to a non-existent Job. All "PREV" jobs referenced must have already been added.
441	previous Job already has max scheduled dependencies	For the Job being added a scheduler reference has been made to a Job that already has its maximum allowable dependent jobs configured.
442	previous dependent Job for scheduler does not exist	For the Job being added a scheduler reference has been made to a non- existent Job. All "PREV" jobs referenced must have already been added.
443	previous Segment already has max scheduled dependencies	For the Segment being added a scheduler reference has been made to a Segment that already has its maximum allowable dependent segments configured.
444	maximum number of segments within a job would be exceeded	Addition of the requested segment would result in the number of configured segments exceeding the defined maximum segments for the job.
445	maximum number of units within a segment would be exceeded	Addition of the requested unit would result in the maximum configured units for this segment being exceeded.
446	GroupId must be specified for rerunnable Jobs	An attempt to add a job failed because the RERUN attribute was set to ON and a GROUPID was not specified. Rerunnable jobs must have a GROUPID specified.
447	rerunnable Jobs can have a maximum of one dependency	Addition of a rerunnable Job failed because more than one previous Job was defined. Rerunnable Jobs are only permitted with single dependency chaining.
448	previous job must be part of the same groupid	This error is generated if an attempt to add a rerunnable Job with a previous dependency outside the specified groupid is made. Rerunnable Jobs can only have dependencies that have the same groupid.

449	previous rerunnable Job already has a dependency	An attempt was made to add a rerunnable Job with a PREV Job specified. The specified previous Job already has a next Job specified; rerunnable Jobs can only have a single dependency chaining.
450	Jobs with a groupid must be rerunnable	An attempt to add a Job with a groupid failed because the rerun attribute was not set. Jobs with a groupid must be rerunnable.
451	only first Job of rerunnable group may be started	The START JOB command is not permitted for a rerunnable Job unless the Job is the initial Job within a groupid.
452	IFFAIL Job cannot be a rerunnable Job with previous dependencies	This error indicates that an add step was attempted with an IFFAIL Job. The named IFFAIL Job was a rerunnable Job which was not the initial Job in the specified group.
453	steps within rerunnable Jobs cannot use the IFFAIL option	An add step was attempted, within a rerunnable Job, which specified an IFFAIL Job. The IFFAIL attribute is not valid for steps within rerunnable Jobs.
454	an initial Job already exists for the specified group	Addition of a rerunnable Job failed because the Job had no previous dependency and an initial Job already exists for the specified group.
455	maximum number of groupids permitted have already been added	An attempt to add a job with a previously undefined groupid failed because the maximum number of groups already exist.
458	segment dependencies not permitted in rerunnable Jobs	An attempt to add a segment failed because the segment had dependencies, and the segment is part of a rerunnable Job. Dependencies are only permitted at Job level for rerunnable Jobs.
456	this unit is part of a rerunnable Job. Start only permitted at Job level	The start unit command cannot be used for units which are part of a rerunnable Job.
457	this segment is part of a rerunnable Job. Start only permitted at Job level.	The start segment command is not permitted for segments which form part of a rerunnable job.
459	rerunnable jobs cannot use the starttime facility	An add Job command failed because the Job was specified as rerunnable with a starttime. Rerunnable Jobs cannot utilise the starttime facility.
460	specified Job is part of currently active group	An attempt to start a Job which is part of an active group was attempted. The start is rejected.
461	Specified previous job does not exist	Attempt to link to a job which does not exist.
462	Previous job already has maximum dependencies	Attempt to link a job which already has the maximum number of dependencies.
463	The specified dependency already exists	Attempt to link a job. This link already exists.
464	Cannot specify both jobs to be dependent upon each other	Attempt to link a job to itself.
465	Specified next job does not exist	Attempt to link a job to one which does not exist.
466	Next job already has maximum previous dependencies	Attempt to link a job to another which already has the maximum number of dependencies

467	Internal error - Link Proc A	Error occurred during processing of a LINK command. Report this to HPE GNSC.
468	Specified previous segment does not exist	Attempt to link a segment which does not exist.
469	Previous segment already has maximum next dependencies	Attempt to link a segment which already has maximum number of dependencies.
470	The specified dependency already exists	Attempt to link a segment. This link already exists.
471	step2 does not exist	Attempt to alter a step 2 but this does not exist.
472	Step flags cannot be altered to a "currently running" state	An attempt to "ALTER FLAGS" to a state where this Step is running, from a non-running state is not valid.
473	Step flags cannot be set to this status, "running and complete"	An attempt to "ALTER FLAGS" to a state where this Step is both running and complete is not valid.
474	Internal error - Link Proc B	Error occurred during processing of a LINK command. Report this to HPE GNSC.
475	Job level time dependency exist for this segments job identity	An attempt to add a segment has failed because the segment's job has a time dependency.
476	Step2 cannot be added if Step1 has MAXTIME specified	An attempt to add a two-step unit has failed because the first step is configured under MAXTIME. MAXTIME can only be specified for single step units.
477	Specified IFFAIL job does not exist	The Job specified in the IFFAIL clause must already exist within the BMON.
478	Specified ASSIGN does not exist for this step	An attempt to alter an ASSIGN has been made for an ASSIGN which does not exist.
479	No PARAMS have been sent for this step	An alter param command has been issued for a step which has no params set.
480	Specified PARAM does not exist for this step	An attempt to alter a PARAM has been made for a PARAM which does not exist.
481	Define error on step 1	An attempt to start a job has failed because an error related to the defines has occurred.
482	Alteration would cause param block to exceed 1024 bytes	Alteration of the specified parameter would cause the param block to exceed 1024 bytes.
483	Too many OSS arguments on step 1	The job has too many OSS arguments configured.
484	Step type not OSS	An attempt to use an ALTER OSS-STEP command on a step which is a Guardian type STEP not OSS.
486	File error with BMONDEFS file	A file error has occurred during a SYNC IN command
487	Error in "TRANSLATE" file - Too many NODES	During a SYNC IN command the TRANSLATE/BMONDEFS file contains too many NODE translate lines. Max value = 10.
487	Error in "TRANSLATE" file - Illegal old node	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a NODE line which contains an illegal old NODE.

487	Error in “TRANSLATE” file - Illegal new node	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a NODE line which contains an illegal new NODE.
487	Error in “TRANSLATE” file - Unknown new node	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a NODE line in which a NODE is not known.
488	Error in “TRANSLATE” File - Too many VOLUMES	During a SYNC IN command the TRANSLATE/BMONDEFS file contains too many VOLUME translate lines. Max value = 10.
488	Error in “TRANSLATE” File - Illegal old VOLUME	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a VOLUME line which contains an illegal old VOLUME.
488	Error in “TRANSLATE” File - Illegal new VOLUME	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a VOLUME line which contains an illegal new VOLUME.
489	Error in “TRANSLATE” File - Too many SUBVOLS	During a SYNC IN command the TRANSLATE/BMONDEFS file contains too many SUBVOLS translate lines. Max value = 10.
489	Error in “TRANSLATE” file - Illegal old SUBVOL	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a SUBVOL line which contains an illegal old SUBVOL.
489	Error in “TRANSLATE” file - Illegal new SUBVOL	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a SUBVOL line which contains an illegal new SUBVOL.
490	Error in translating filename - node	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a NODE line which contains an illegal node.
490	Error in translating filename - volume	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a VOLUME line which contains an illegal volume.
490	Error in translating filename - subvol	During a SYNC IN command the TRANSLATE/BMONDEFS file contains a SUBVOL line which contains an illegal subvol.
490	Error in translating filename - filename length	During a SYNC IN command an error has occurred converting a filename - the converted filename is too long.
490	Error in translating filename - old fname	During a SYNC IN command an error has occurred converting an old filename.
490	Error in translating filename - default	During a SYNC IN command an error has occurred converting the step default.
491	Error translating params	During a SYNC IN command an error has occurred converting the step parameters.
492	Recovery file BMON process name does not match my process name	The SYNC IN command has failed because the BMON process name does not match the process name found in the old RECOVERY file.
493	Error accessing RECOVERY file	During SYNC IN a file error has occurred on the RECOVERY file.
493	File error with OLD RECOVERY file	During SYNC IN a file error has occurred on the old RECOVERY file.
494	Error with POOL deallocation	The pool allocated during a SYNC IN cannot be deallocated. Report this to HPE GNSC.

495	Internal error - (add step)	Error occurred during processing of a SYNC IN command. Report this to HPE GNSC.
496	Internal error - (add seg)	Error occurred during processing of a SYNC IN command. Report this to HPE GNSC.
497	Internal error - (add job)	Error occurred during processing of a SYNC IN command. Report this to HPE GNSC.
498	BMON busy - please try again	Error occurred during a SYNC OUT command. Try the command again.
501	Unable to open specified BMON	An attempt to “OPEN BMON” has failed. Check BMON is running.
502	No current BMON open	An attempt to access BMON fails due to BMON not having been opened yet. Refer to Appendix A “RUNNING BCOM “ for details.
503	These details cannot be returned at this level	An attempt has been made to obtain extra detail in a command that cannot support it. This can occur, for example, in “STATUS JOB *, DETAIL” which is invalid whereas “STATUS JOB xxxx, DETAIL” is valid.
504	Param block exceeds 1024 bytes	An attempt to “SET STEP PARAM” would result in a PARAM array exceeding 1024 bytes in length. The maximum length of the interprocess message bearing the PARAM messages is 1024 bytes, which is a NonStop restriction, consisting of, as follows: ((length of <name> + 1) + (length of <value> + 1)) * number of PARAMs
505	MAXJOBS cannot exceed MAX- SEGS	Due to the structure of BMON, this would be an illogical setup. For this reason when these ”SET BMON” attributes are set in this manner BCOM will reject any “START BMON” command.
506	MAXSEGS cannot exceed MAXUNITS	Due to the structure of BMON, this would be an illogical setup. For this reason when these ”SET BMON” attributes are set in this manner BCOM will reject any “START BMON” command.
507	MAXDEPENDS cannot exceed MAXUNITS	Due to the structure of BMON, this would be an illogical setup. For this reason when these “SET BMON” attributes are set in this manner BCOM will reject any “START BMON” command.
508	Old and New Recovery Files must be specified	For “RECOVER BMON” both an old and a new recovery file are required.
509	Object File-name must be specified	For “ADD STEP” commands the object file name to be run is the required parameter.
509	OSS Program name must be specified	For “ADD STEP” commands where the Step Type is OSS the program to be run is a mandatory parameter.
509	OSS Pathname must be specified	For “ADD STEP” commands where the Step Type is OSS the pathname of the program to be run is a mandatory parameter.
510	Object and Swap files must be on the same System	For “ADD STEP” commands the object file and it’s swap file must reside on the same system. This is a NonStop requirement.

511	Object and Library files must be on the same System	For “ADD STEP” commands the object file and it’s library file must reside on the same system. This is a NonStop requirement.
512	BMON not started	A command other than (LOG1, LOG2, START BMON, RECOVER BMON,OPEN BMON) has been received when BMON has not been ”START”ed or “RECOVER”ed .
513	There is no BMON currently open	A command that requires a call to BMON has been issued when no BMON is currently open.
514	Unable to access BMON	A command that requires a call to BMON has been issued which fails due to a file error on the BMON Process. Find out what has happened to the previously active BMON.
515	Incompatible releases of BMON and BCOM	The versions of BMON and BCOM do not match; Contact HPE GNSC.
516	Specified Unit cannot be found	A “SET STEP LIKE xxxx.yy.zz.n” has been issued which fails due to the Unit concerned not existing.
517	Specified Unit has no STEP2	A “SET STEP LIKE xxxx.yy.zz.2” has been issued which fails due to the Unit having no step2.
518	Unable to open initial BMON	An attempt to “OPEN BMON” on the initial BMON has failed. Check the specified BMON process is running.
519	Failure on EDITREADINIT	An attempt to open and access a supposedly edit type input file has failed. Check the input file is a valid edit file that contains data.
520	Unable to OPEN help-file	The file “BMONHELP” which should be residing on the same volume as the BCOM object file has returned a file error on an open attempt. Seek assistance from support personnel to investigate the problem.
521	No HELP currently available	The command asked for is not one of those currently available in BCOM
522	Unable to find Help record	The HELPFILE has returned a file-error on Key position.
523	Unable to read Help record	The HELPFILE has returned a file-error on Read.
524	Max previous jobs already set	The maximum number of “SET JOB PREV” statements have already been issued.
525	Max previous segments already set	The maximum number of “SET SEGMENT PREV” statements have already been issued.
526	Maximum number of ASSIGNs reached	An attempt to add an assign has failed due to the maximum of ASSIGN statements for a step has already being reached.
526	Maximum number of DEFINESs reached	An attempt to add an define has failed due to the maximum of DEFINE statements for a step has already being reached.
527	Maximum number of PARAMs reached	An attempt to “SET STEP PARAM” has failed due to the maximum of 100 PARAM statements already being reached. This is a BMON restriction which limits the number of PARAM statements to 100 per Step.

528	Set Type must be Guardian or OSS	An "Add Step" command has been rejected due to the Step Type not being Guardian or OSS.
528	Maximum number of FD Entries reached	An "Add Step" command has been rejected due to the maximum number of FD entries allowed, in this BMON configuration, as having already been reached. The "INFO BMON" command will give details.
529	Maximum number of Environment Parameters reached	An "Add Step" command has been rejected due to the maximum number of Environment Parameters allowed, in this BMON configuration, as having already been reached. The "INFO BMON" command will be give details.
530	LOCALPARAMS File Error : nnn	Attempt to access the previously defined LOCAL PARAMS file failed with the indicated Guardian error.
531	GLOBALPARAMS File Error : nnn	This message indicates that an attempt to access the previously defined GLOBAL PARAMS file failed with the specified Guardian error.
533	GLOBALPARAMS - Empty File	The specified GLOBAL PARAMS file contains no parameter data.
534	SET BCOM GLOBALPARAMS not declared	Before attempting to use MultiBatch parameter facilities, the GLOBALPARAMS, LOCALPARAMS and SHELLNAME identifiers must be supplied. The above errors indicate that the specified identifier had not been supplied, hence parameter resolution was not possible.
535	SET BCOM LOCALPARAMS not declared	Before attempting to use MultiBatch parameter facilities, the GLOBALPARAMS, LOCALPARAMS and SHELLNAME identifiers must be supplied. The above errors indicate that the specified identifier had not been supplied, hence parameter resolution was not possible.
536	SET BCOM SHELLNAME not declared	Before attempting to use MultiBatch parameter facilities, the GLOBALPARAMS, LOCALPARAMS and SHELLNAME identifiers must be supplied. The above errors indicate that the specified identifier had not been supplied, hence parameter resolution was not possible.
537	Variable not found in LOCAL PARAMS	A parameter substitution failed because the variable parameter for substitution was not defined in the LOCAL PARAMS file.
538	Variable not found in GLOBAL PARAMS	A parameter requiring substitution was declared in LOCALPARAMS as a 'use globals' variable, but was undefined in the specified GLOBALPARAMS.
539	Qualified status not permitted at this level.	A status qualifier may only be used with a STATUS JOB * command.
540	Unable to open help detail file.	The help detail file (BMONHLPD) could not be opened hence detailed help is not available.
541	No help detail currently available.	There is currently no detailed help available for the specified subject.
542	Unable to find help detail record	Detailed help for the specified subject could not be located within the detail help file. Check validity and contents of this file.
543	Unable to read help detail record	The detailed help could not be read from the help file. Check that the file has not been corrupted.

544	SET LIKE facility not available with SYNTAX option	The SET LIKE facility cannot be used in conjunction with the SYNTAX option for BCOM.
545	OBEY FILE not specified	A file name must be specified with the OBEY command.
546	invalid file name specified	An invalid Guardian file name has been specified with the OBEY command. Re-issue with a valid file name.
547	OBEY FILE error nnnn, <file name>	Error nnnn occurred whilst trying to access the specified OBEY file.
548	SPECIFIED FILE is not an edit file	The OBEY command must be used with an edit type file.
549	Error on EDITREAD of obey file	Attempt to read OBEY file failed. Check for corruption of the edit file and then retry.
550	FC not permitted in obey file	The FC command is not a permitted option within an OBEY file.
551	OBEY FILE is empty	The specified file is an empty file. Please retry with a valid OBEY file.
552	Nested OBEY commands detected	The OBEY command cannot be invoked from within an OBEY file.
553	MAXTIME must be at least 2 minutes	The MAXTIME for a step must be between 2 minutes and 23 hours and 59 minutes. The value specified did not fall within these bounds.
554	MAXTIME cannot be greater than 24:00	The MAXTIME for a step must be between 2 minutes and 23 hours and 59 minutes. The value specified did not fall within these bounds.
555	MAXTIME not permitted in 2-step unit	You are not allowed to specify a MAXTIME for a 2-step unit
556	Attribute OWNER requires CLASS SPOOL or CLASS TAPE	Attempt to specify an owner for a define where the define class does not require this attribute.
557	Attribute SWAP requires CLASS DEFAULTS or CLASS SORT / SUBSORT	Attempt to specify a swap volume for a define where the define class does not require this attribute.
558	Attribute VOLUME requires CLASS DEFAULTS or CLASS TAPE	Attempt to specify a volume for a define where the define class does not require this attribute.
559	CLASS TAPE must be used before use of tape attributes	Attempt to specify tape attributes for a define where the define class is not TAPE.
560	Invalid attribute value	Attempt to specify a define attribute. The attribute is not valid.
561	CLASS SPOOL must be used before use of Spool attributes	Specify the define CLASS as SPOOL before setting the SPOOL attribute.
562	CLASS MAP must be used before use of FILE attribute	Specify the define CLASS as MAP before setting the FILE attribute.
563	CLASS DEFAULTS must be used before use of Default attributes	Specify the define CLASS as DEFAULTS before setting the DEFAULT attribute.

564	CLASS SORT/SUBSORT must be used before use of SORT attributes	Specify the define CLASS as SORT or SUBSORT before set ting the SORT attribute.
565	Attribute SUBVOL requires use of CLASS CATALOG	Specify the define CLASS as CATALOG before setting the SUBVOL attribute.
566	Maximum defines already set	Attempt to add a define when the maximum defines have already been set
567	Invalid define name	Attempt to add a define but define name specified is not valid
568	Define attributes not set	Before adding a define set the define CLASS to be one of : CATALOG, DEFAULTS, MAP, SORT, SPOOL, SUBSORT, TAPECATALOG, TAPE.
570	DEFINESETATTR error : nnnn, attribute : aaaaaaaaaaa	An attempt to add a define has failed because the CALL to DEFINESETATTR has failed with the error given in the message. Check SYSTEM PROCEDURE CALLS manual for details.
571	DEFINEVALIDATEWORK error : nnnn	An attempt to add a define has failed because the CALL to DEFINEVALIDATEWORK has failed with the error given. Check SYSTEM PROCEDURE CALLS manual for details.
576	No defines have been set for this step	An attempt has been made to ALTER STEP DEFINE but this step does not have any defines configured.
577	DEFINE attribute/value not valid	An attempt has been made to ALTER STEP DEFINE but the attribute or value is invalid.
578	Define name not found for this step	An attempt has been made to ALTER STEP DEFINE but this step does not have a define with this name configured.
579	Define <define-name> already exists for step	Either an attempt has been made to ALTER STEP ADD DEFINE but this step already has this define name configured or an attempt has been made to ADD STEP DEFINE with a duplicate name.
580	PREPARE NAME not specified	BCOM BUILD requested but no PREPARE name was specified
581	UNABLE TO OPEN DATACONF nnn <dataconf-file>	Error on accessing MultiBatch DATACONF file; nnn is the Guardian file system error. This file should be assigned before a BMON is built and BCOM is started. An example of this is... ASSIGN DATACONF, $<disk>.<subvol>.<dataconf>
583	UNABLE TO OPEN PREPCONF nnn <prepconf-file>	Error on accessing MultiBatch PREPARE configuration database; nnn is the Guardian file system error. This file should be configured in the MultiBatch PATHWAY on screen DCC.
584	UNABLE TO ACCESS PREPCONF nnn <prepconf-file>	Error on accessing MultiBatch PREPARE configuration database; nnn is the Guardian file system error. This file should be assigned in the MultiBatch PATHWAY on screen DCC.
585	UNABLE TO OPEN BUILD nnn <build -file>	Error on accessing MultiBatch BUILD file for the specified PREPARE name (nnn is the Guardian file system error). The location of this file can be displayed on screen PRC in the MultiBatch PATHWAY.

586	UNABLE TO ACCESS BUILD nnn <build-file>	Error on accessing MultiBatch BUILD file for the specified PREPARE name (nnn is the Guardian file system error). The location of this file can be displayed on screen PRC in the MultiBatch PATHWAY.
587	PREPARE NAME does not exist	PREPARE name specified does not exist in the MultiBatch PREPCONF database. This database can be listed on screen PRL in the MultiBatch PATHWAY system.
588	NO BMON process open	Before the BUILD phase can be executed, the user needs to open a MultiBatch process. An example follows... RUN BMON $BMON RUN BCOM $BMON ; BUILD <prepare name>
589	BUILD file not for this BMON process	BCOM has a BMON process open but it is not for the PREPARE name specified. For example, if the PREPARE name ARCHIVE^DAILY was for MultiBatch BMON $ARCH, this error message would be displayed if the following command was used: RUN BCOM $BMON ; BUILD ARCHIVE^DAILY
590	Extended Segment cannot be allocated	An error indicating an internal BMON error, contact HPE GNSC.
591	Extended Segment cannot be used	An error indicating an internal BMON error, contact HPE GNSC.
592	History Pool Buffer cannot be allocated	Contact HPE GNSC.
593	History Pool Buffer PUTSPACE error	Contact HPE GNSC.
594	History Pool Buffer GETSPACE error	Contact HPE GNSC.
595	Set OSS-PROGRAM before OSS-PATHNAME	The program name must be entered before the pathname as validation of the path requires a fully qualified file.
596	HOMETERM format error	The home terminal provide for an OSS step could not be formatted into an OSS path
597	OSS LIBRARY file format error	The Guardian library provided for an OSS Step could not be converted into an OSS Path
598	OSS SWAP file format error	The Guardian swap file provided for an OSS Step could not be converted into an OSS Path
599	MEMORY POOL error	Contact HPE GNSC.
600	Checkpoint error	It was not possible to run BMON as a process pair
601	CPU does not exist or is not available	It was not possible to run BMON as a process pair
601	Can't run primary and backup in the same CPU	It was not possible to run BMON as a process pair

601	BMON is not running with a back-up	It was not possible to run BMON as a process pair
602	No job segments available to be submitted	When submitting an On Demand Job no segments could be started. The job have already been submitted
603	You must run with a recovery file to facilitate On Demand Jobs	To use On Demand Jobs you must run BMON with a recovery file. On Demand Jobs are stored in the recovery file.
604	Licensed BMON required to facilitate On Demand Jobs	On Demand Jobs are run under the ID of the BCOM user that creates them. A licensed version of BMON is required to start processes under user id other than the user id of the BMON process.
605	Specified job identity is not a model job	When using the BCOM command ADD FROM MODEL <job> the job identified is not a Model Job
606	No job number available	All the range of On Demand Job numbers (1-9999) are in use. Retry the command at a later time.
607	You cannot start elements of a model job	The BCOM START JOB/SEGMENT/UNIT commands are not allowed for On Demand Jobs. On Demand Jobs are started using SUBMIT JOB.
608	You do not have authority to restart the step	Only the owner of an On Demand Job can restart a step.
609	You cannot alter steps of a model job	Model Jobs are set up in base data and cannot be altered.
610	You do not have authority to alter the step	Only the owner of an On Demand Job can alter a step.
611	You cannot hold or release an on demand job	It is not possible to hold or release an On Demand Job
612	You do not have authority to abort the step	Only the owner of an On Demand Job can abort a step.
613	You cannot held/release parts of a model job	Model Jobs are used as the bais for On Demand Jobs, in this way they are not available to be run
801	Invalid message received by BMON, contact support	This is a critical error indicating an internal error within BMON. Contact HPE GNSC.
802	Insufficient data pool, contact support	Internal error occurred within BMON, contact HPE GNSC.
802	Insufficient data pool, contact support, *WARNING step configuration is not complete	Internal error occurred within BMON, contact HPE GNSC.

E.4 BMON Build Messages

The following messages are for information only and are generated during the BMON BUILD procedure, e.g. RUN BCOM $BMON; BUILD <Prepare Name>.

They are not errors and will have no adverse effect on configuring and running MultiBatch.

Message	Description
>>>>>>>>>> LOCALPARAM expanded <param name> : <param value>	The named parameter was expanded using the LOCALPARAMS Database to the value shown.
>>>>>>>>>> GLOBALPARAM expanded <param name> : <param value>	The named parameter was expanded using the GLOBALPARAMS Database to the value shown.
>>>>>>>>>> Run param expanded to TRUE	The parameter named in a SET STEP RUN command expanded to TRUE or the parameter named in a SET STEP RUN NOT command expanded to FALSE.
>>>>>>>>>> Run param expanded to FALSE - Unit will not run	The parameter specified in a SET STEP RUN command expanded to FALSE or the parameter specified in a SET STEP RUN NOT command expanded to TRUE. This unit will be skipped at execution time. Dependencies will treat the unit as completed successfully, and the Log messages will reflect that the Unit has been skipped in this manner.
>>>>>>>>>> STEP will run as User Id nnn, nnn	This message is only relevant to MultiBatch. The program will run as User Id nnn, nnn.

E.5 Checkstart Warning Messages

The following messages are generated when the BMON when the Check Start field in the CBM screen is set to Y.

A user can check if Checkstart is set by entering an INFO BMON command in a BCOM session.

Message	Description
1) WARNING >>> UNIT xxx.xx.xx has already completed >>> Please confirm you wish to restart Unit xxxx.xx.xx Y/N	This message will be displayed when a BCOM user attempts to RESTART a Unit which has already been run.
WARNING >>> PREV JOB xxxx has not completed >>> WARNING >>> NEXT JOB xxxx has already been started A START of JOB xxxx will cause ALL SEGMENTS AND UNITS in this job to start! Please confirm you wish to start Job xxxx Y/N	These messages will be displayed when a BCOM user attempts to START a Job, whose previous dependencies are incomplete and whose next dependencies have already been started.
WARNING >>> PREV SEGMENT xxxx. xx has not completed >>> WARNING >>> NEXT SEGMENT has already been started A START of SEGMENT will cause ALL UNITS in this Segment to start! Please confirm you wish to start Segment xxxx.xx Y/N Please confirm you wish to start Unit xxxx.xx.xx Y/N	These messages will be displayed when a BCOM user attempts to START a Segment, whose previous dependencies are incomplete and whose next dependencies have already been started. A START UNIT command will check for Segment level dependencies.

E.6 STEP Messages

These events are generated when the BMON process writes to its LOG1 or LOG2 file which is typically the MultiBatch STREAMER process. See Chapter 1.9 'Log Files' for further details on use of LOG files.

The STREAMER process then converts the original BMON source events into these tokenised events and are issued to $0 or an Alternate Collector process. See Appendix G 'MultiBatch Streamer Process' for further details. The EMS SSID and Event Subject is set to:

MULTIBAT.1.0 (SSID)

STEP ID (Subject)

The meanings of messages relating to STEP "xxxx.yy.zz.n" are documented here, where (C) indicates a Critical event.

Event Number	Message	Cause
3	xxxx.yy.zz.*, started by operator	This Unit has started running. The start was initiated via operator input (not the scheduler).
4	xxxx.yy.zz.*, started Step 2	STEP 2 of this Unit has been started following a successful start or restart of STEP1.
5	xxxx.yy.zz.*, restarted by operator	This Unit has been restarted following operator input "RESTART UNIT"
6	xxxx.yy.zz.*, unit complete n:n	This indicates that the UNIT has completed, or has completed for two step UNITs. The "n:n" field indicates the order in which each step completed. As an example, a 1:2 would indicate that step 1 had finished followed by step 2 and vice versa.
7	xxxx.yy. **.*, Segment complete	All units within this Segment have successfully completed.
8	xxxx.**.**.*, JOB COMPLETE !!!	All segments within this Job successfully completed.
9	xxxx.yy.zz.*, started by scheduler	This Unit has started running. The start was initiated automatically by the scheduler (not operator input).
10 (C)	xxxx.**.**.*, scheduled Job HELD	During a scheduler initiated "START" at Job level, this Job was found to be in a hold state.
11 (C)	xxxx.yy.zz.n, abended	This Step has abended.
12 (C)	xxxx.yy.zz.n, PROCESS CREATE error nnn:nnn	This indicates that the Guardian Step failed on creation. The error is indicated within the nn:nnn (see - Guardian System Procedure Calls Ref Manual Volume II).
13	xxxx.yy.zz.*, unit initialised	This protocol Step has indicated to BMON that it has initialised its control file.
14	xxxx.yy.zz.n, user message - "..."	This Protocol Step has sent this message to be logged.
15	xxxx.yy.zz.*, unit complete - "…"	This protocol Step has completed and has optionally sent a message to be logged.
17 (C)	xxxx.yy.zz.n, stopped, incomplete	This Protocol Step has stopped running without sending a "user protocol 199" message.

18 (C)	xxxx.yy.zz.n, aborted by operator	BCOM command "ABORT UNIT" has been acted upon.
19	xxxx.yy.zz.n, this Step's flags altered by operator	A BCOM running under SUPER.SUPER, or the BMON's Privileged User, has amended this steps flags.
20 (C)	xxxx.yy.**.*, Segment HELD	During an operator initiated "START" at Job level, this Segment was found to be held. Other segments will not have been affected.
21 (C)	xxxx.yy.zz.*, Unit HELD	During an operator initiated "START" at Job or Segment level, this Unit was found to be held. Other units will not have been affected.
22 (C)	xxxx.yy.**.*, scheduled Segment HELD	During a scheduler initiated "START" at Job or Segment level, this Segment was found to be in a hold state. If Job level then other segments within this Job will be unaffected.
23 (C)	xxxx.yy.**.*, Scheduled Unit HELD	During a scheduler initiated "START" at Job or Segment level, this Unit was found to be in a hold state. Other units within this job/segment will have been unaffected.
24	xxxx.yy.zz.*, restart, initiated by BMON	This Step has been restarted after a CPU failure.
26	xxxx.yy.zz.n Open process .... file error - nnn	Event 26 is generated when nnn is generally Guardian error 14, or 201. This means that the step can be safely autorestarted by BMON. Event 87 is generated for other Guardian errors.
28 (C)	xxxx.yy.zz.*, <<<< UNIT INCOMPLETE, INVESTIGATE!	This message can be emitted during a SYNC (Recover BMON command) operation; As BMON rebuilds its internal tables from the current recover File, it checks each STEP for a FLAGS status which indicates the STEP was still running, and outputs this message when any are found.
30 (C)	xxxx.yy.zz.n, stopped	This Step has stopped running, probably without processing. The Process has failed to accept Startup, PARAM or ASSIGN messages in the standard manner.
41 (C)	xxxx.yy.zz.n, stopped, incomplete. CC = nnnn	This NON-Protocol Step has stopped running, and has returned a Completion Code indicating an unsuccessful run. See Appendix A9 Miscellaneous Commands for the setting of step status flags.
44 (C)	xxxx.yy.zz.n, UNIT SKIPPED!!!, schedule continues	The specified unit was not executed because the SKIP UNIT indicator was set, either as a result of a ALTER STEP RUNOFF command, or a SET STEP RUN <param> command expanding to false. The unit should be treated as having completed successfully.
45 (C)	xxxx.yy.**.*, scheduled Segment HELD (job level)	During a Scheduler initiated "START" at Segment level, this Segment was found to be in a hold state at Job level.
46	FILE ERROR ON fffff GUARDIAN err nnn	This message is output in the event of a file error on the one of the following files: OLD RECOVERY NEW RECOVERY $RECEIVE In certain circumstances such as "START BMON" or "SYNC IN", these may result in BMON abending. Note that you may see file errors reported on the OLD RECOVERY file only during a SYNC

47	LOG1 fffff FILE ERROR nnn - CLOSED	Guardian error “nnn” occurred on logging file “fffff”. The file will be closed.
48	LOG2 fffff FILE ERROR nnn - CLOSED	Guardian error “nnn” occurred on logging file “fffff”. The file will be closed.
49 (C)	xxxx.yy.zz.n, CPU failure	While this Step was running the CPU it was running in failed. If the Step is a “protocol” Step and the Step has not completed (user message code 199) BMON will attempt to “RESTART” the Unit.
50	xxxx.yy.zz.n, Write [ASSIGN] message, error - nnn	Whilst trying to write the given message type, Guardian error ”nnn” occurred.
51	xxxx.yy.zz.n, Write [PARAM] message, error - nnn	Whilst trying to write the given message type, Guardian error “nnn” occurred.
52	xxxx.yy.zz.n, Write [Startup] message, error - nnn	Whilst trying to write the given message type, Guardian error “nnn” occurred.
54	xxxx.**.**.*, scheduled job awaiting STARTTIME	All previous jobs for the specified job have completed successfully, however, the schedule cannot continue until the STARTTIME is reached. At STARTTIME the schedule will re-activate itself.
55 (C)	xxxx.yy.zz.*, ABORT will be forced at MAXTIME in <nn> minutes.	This unit will reach its maximum execution time in <nn> minutes at which point the unit will be aborted. To prevent this an ALTER STEP, MAXTIME command should be issued.
56 (C)	xxxx.yy.zz.*, Aborted at MAXTIME	This unit was aborted by BMON when its execution time reached the specified
57 (C)	BMON closedown initiated	BMON is shutting down following successful completion of all jobs. Info: The BMON in the CBM screen will have its ‘Closedown’ field set to Y.
58 (C)	xxxx.yy.zz.*, ADD DEFINE error nnnn.	An attempt to add the specified define failed with error nnnn. Refer to the NonStop System Procedure Call Manual for an explanation of the error.
59 (C)	xxxx.yy.zz.*, SET DEFINE error nnnn.	An attempt to set an attribute for the specified define failed with error nnnn. Refer to the NonStop System Procedure Call Manual for an explanation of the error.
60	xxxx.**.**.*, Failure condition initiated	A failure job has been started after a unit with a clause abended or finished abnormally.
61 (C)	xxxx.**.**.*, job held	This log message is issued when the HOLD JOB command is issued from BCOM.
62 (C)	xxxx.xx.**.*, segment held	This log message is issued when the HOLD SEGMENT command is issued from BCOM.
63 (C)	xxxx.xx.xx.x, unit held.	This log message is issued when the HOLD UNIT command is issued from BCOM.
64	xxxx.**.**.*, job released	This log message is issued when the RELEASE JOB command is issued from BCOM.
65	xxxx.xx.**.*, segment released	This log message is issued when the RELEASE SEGMENT command is issued from BCOM.

66	xxxx.xx.xx.x, unit released	This log message is issued when the RELEASE UNIT command is issued from BCOM.
67	xxxx.**.**.*, started in cpu 01 (nnn,nnn)	This log message would be issued immediately after a Newprocess error 10, CPU not found, to signify that the UNIT has been started in another CPU.
68 (C)	xxxx.xx.xx.x, this Step's run state altered to OFF|ON	A Step can be configured but skipped at Run time by use of the SET STEP RUN <param-name> command. You can alter the Run state of the Unit using the ALTER STEP xxxx.xx.xx RUNOFF | RUNON. This Log message would be issued following this command.
69 (C)	xxxx.**.**.*, job on Critical Path	This LOG message is issued if a UNIT finishes abnormally or a Job/Segment or Unit is held and is classed as being on the Critical path. This attribute can be set using the SET JOB/SEGMENT/UNIT CRITICAL ON command.
70 (C)	xxxx.**.**.*, segment on Critical Path	This LOG message is issued if a UNIT finishes abnormally or a Job/Segment or Unit is held and is classed as being on the Critical path. This attribute can be set using the SET JOB/SEGMENT/UNIT CRITICAL ON command. This LOG message is issued if a UNIT finishes abnormally or a Job/Segment or Unit is held and is classed as being on the Critical path. This attribute can be set using the SET JOB/SEGMENT/ UNIT CRITICAL ON command.
71 (C)	xxxx.**.**.*, unit on Critical Path	This LOG message is issued if a UNIT finishes abnormally or a Job/Segment or Unit is held and is classed as being on the Critical path. This attribute can be set using the SET JOB/SEGMENT/UNIT CRITICAL ON command.
72 (C)	xxxx.**.**.*, Short stop message. Stopped, incomplete	In some circumstances the Guardian Operating System sends a Parent process a short "STOP" message when a Child process has completed. The missing part of the Stop message contains the completion code and so MultiBatch is unable to check whether the UNIT finished successfully. A clerical check is required to ascertain whether the UNIT completed normally.
73	xxxx.**.**.*, JOB added interactively	This Log message is generated when a Job is added from a BCOM session, an OBEY file, or an IN file after the initial configuration is complete. The SET BCOM INTERACTIVE ON command is used to signify an interactive ADD. These messages are used by the MultiBatch WATCHER process to accurately maintain the STATUS database.
74	xxxx.**.**.*, SEGMENT added interactively	This Log message is generated when a Segment is added from a BCOM session, an OBEY file or an IN file after the initial configuration is complete. The SET BCOM INTERACTIVE ON command is used to signify an interactive ADD. These messages are used by the MultiBatch WATCHER process to accurately maintain the STATUS database.
75	xxxx.**.**.*, UNIT added interactively	This Log message is generated when a Unit is added from a BCOM session, an OBEY file or an IN file after the initial configuration is complete. The SET BCOM INTERACTIVE ON command is used to signify an interactive ADD. These messages are used by the MultiBatch WATCHER process to accurately maintain the STATUS database.

76	xxxx.**.**.*, JOB deleted interactively	This Log message is generated when a Job is deleted from a BCOM session, an OBEY file or an IN file after the initial configuration is complete.
77	xxxx.**.**.*, SEGMENT deleted interactively	This Log message is generated when a Segment is deleted from a BCOM session, an OBEY file or an IN file after the initial configuration is complete.
78	xxxx.**.**.*, UNIT deleted interactively	This Log message is generated when a Unit is deleted from a BCOM session, an OBEY file or an IN file after the initial configuration is complete.
79 (C)	xxxx.**.**.*, attempt to start JOB out of sequence	If SET BMON CHECKSTART is set to ON, then any attempt to Start a JOB out of sequence results in a BCOM user being prompted for confirmation. If the user replies in the affirmative, the appropriate Log message is issued.
80 (C)	xxxx.**.**.*, attempt to start SEGMENT out of sequence	If SET BMON CHECKSTART is set to ON, then any attempt to Start a SEGMENT out of sequence results in a BCOM user being prompted for confirmation. If the user replies in the affirmative, the appropriate Log message is issued.
81 (C)	xxxx.**.**.*, attempt to start UNIT out of sequence	If SET BMON CHECKSTART is set to ON, then any attempt to Start a UNIT out of sequence results in a BCOM user being prompted for confirmation. If the user replies in the affirmative, the appropriate Log message is issued.
82	xxxx.**.**.*, JOB has been reset to a configured state	Upon successful completion of the specified job, the flags of all steps under the job have been reset to a configured state.
83 (C)	xxxx.**.**.*, Repetitive start failures ... step not automatically restarted	If in attempting to start a step, BMON encounters an error 14, the unit will automatically be restarted. If 5 x successive attempts to start all fail with error 14, then the automatic restart functionality will be suspended and the above message will be generated.
84 (C)	xxxx.**.**.*, IFFAIL job is part of active group - not started	Failure of a unit caused an IFFAIL job to be invoked. The specified IFFAIL job was a rerunnable job which is part of an active rerunnable group.
85 (C)	xxxx.**.**.*, IFFAIL job - no units have been started	An attempt to invoke a failure job failed; no units within the job are started.
86 (C)	xxxx.**.**.*, IFFAIL job started with errors	An attempt to invoke a failure condition job resulted in one or more units generating an error at start time
87	xxxx.yy.zz.n, Open Process. file error - nnn	Event 87 is generated for errors other than when nnn = 14, or 201.
88	xxxx AAAA.01.01.1, <audit-log-change>	This event is generated, when a BCOM user makes a configuration change using the ALTER STEP command. Examples of <audit-log-change> include “this Step’s ASSIGN | PARAM deleted” and “this Step’s CPU altered from 00:01 to 01:00”.
89	xxxx WARNING! Starttime reached before end of previous job	The specified starttime for a job has been reached before the previous job has completed.

90 (C)	xxxx WARNING! Process has reached MAXTIME	This is only a warning message. The process is allowed to continue.
92	xxxx Internal error - “find^pid^proc <Letter>	Contact HPE GNSC with details of the error message
93 (C)	xxxx.**.**.*, failure job held	At the point BMON was to start a step the associated JOB/SEGMENT/UNIT was found to be on HOLD.
94	xxxx.**.**.*, DEFINE “=<define name>” has been added to this step	A BCOM ALTER STEP <name>, add DEFINE command has been successfully executed
95 (C)	xxxx.**.**.*, PROCESS SPAWN error nnnn	The PROCESS_SPAWN_ procedure call used to create OSS step processes has reported an error.
96	JOBA.AA.01.1, User_GetInfo_ error – nnn	This event will only be seen if the user has configured SET STEP USERID processing. The most common value of ‘nnn’ is 011. This means that the user id configured in the STEPCONF file has been deleted since the step was configured.
98	BMON PROCESS CONTINUING WITHOUT RECOVERY - USE SYNC OUT COMMAND	An error has occurred writing to the recovery file. The processing is continuing without using the recovery file. To restart the writing to a recovery file use the SYNC OUT command.
99	SYNC IN - error : “TRANSLATE” file - <message>	During a SYNC IN command the TRANSLATE/BMONDEFS file: 1) contains too many entries for Node/Volume/Subvolume/Directory. Max value = 10. 2) has a badly formatted Node/Volume/Subvolume/Directory
101	SYNC IN error : Recovery File BMON processname does not match my processname	The SYNC IN command has failed because the BMON process name does not match the process name found in the old RECOVERY file.
102	User Authenticate error - nnn/mmm	The USER_AUTHENTICATE_ procedure call returned an error. This procedure is used when switching users for a step that specifies the user id under which it is to run.
104	BMON SYNC IN failure	Fatal error during BMON SYNC IN. BMON abended.
105	xxxx.**.**.*, Memory POOL error	During the construction of a request to spawn an OSS process a memory pool error was encountered. Contact HPE GNSC.
106	xxxx.**.**.*, OSS LIBRARY file format error	The Guardian library provided for an OSS Step could not be converted into an OSS Path

107	xxxx.**.**.*, OSS SWAP file format error	The Guardian swap file provided for an OSS Step could not be converted into an OSS Path
108	xxxx.**.**.*, HOMETERM format error	The home terminal provide for an OSS step could not be formatted into an OSS path

E.7 Non-Critical Messages

The meanings of messages relating to the environment of BMON that do not necessarily result in BMON abending, are documented here.

Message	Cause
NEW RECOVERY FILE = “system.$vol.subvol.file”	The recovery file identity is as indicated.
BMON STARTED!!!	BMON has successfully negotiated a “START BMON” command.
OLD RECOVERY FILE = “node.$Vol.subvol.file”	This message is output during a SYNC IN operation.
BMON RESTARTED!!!	BMON has successfully negotiated a “RECOVER BMON” command.
SHUTDOWN BMON REQUESTED	A shutdown of BMON is being performed.
Insufficient pool, “…”	A message of this type will be output in the event of a datapool (memory) shortage. Processing will continue, but systems support should be advised if this type of error ever occurs.
FILE ERROR ON fffff GUARDIAN err nnn	This message is output in the event of a file error on the one of the following files: OLD RECOVERY NEW RECOVERY $RECEIVE In certain circumstances such as “START BMON” or “SYNC IN”, these may result in BMON abending. Note that you may see file errors reported on the OLD RECOVERY file only during a SYNC IN operation
BMON running with no recovery file!!!	BMON has been started with no NEWRECOVER file specified. No NEWRECOVER file specified during a “RECOVER BMON” An attempt to recover BMON was performed without specifying the new recovery file.
RECOVERY FILE IS INCONSISTENT, BMON PROCESS CONTINUING WITHOUT RECOVERY	After a file error on the “NEWRECOVER” file this message indicates that BMON will continue processing without a recovery file.
LOG1 fffff FILE ERROR nnn - CLOSED	Guardian error “nnn” occurred on logging file “fffff”. The file will be closed.
LOG2 fffff FILE ERROR nnn - CLOSED	Guardian error “nnn” occurred on logging file “fffff”. The file will be closed.

E.8 Nonstop Operation Messages

These messages are all associated with the non stop functions of BMON.

Message	Cause / Action
BMON Backup process started in Cpu nn	This message is output after a CHECKPOINT BMON command, when BMON has started its Backup process and is running non stop. It will also be output if there is a takeover following a CPU failure, or if the Primary or Backup process have been stopped for any reason.
BMON Checkallocateseg Error: nnnn	This message would be output if the call to the Guardian System Procedure CHECKALLOCATESEGMENT fails for any reason during the initialisation of BMONs Backup process. The error numbers are listed in the NonStop System Procedure Calls manual.
BMON Checkopen Error: <filename> nnnn	This message would be output if the System Procedure Call CHECKOPEN fails for any reason during Checkpointing to BMONs Backup process. The <filename> indicates which of the Files experienced the problem, and nnnn is the NonStop File System Error number.
BMON Checkpoint Error nnnn:nnnn	This message would be output if the System Procedure Call CHECKOPEN fails for any reason during Checkpointing to BMONs Backup process. The <filename> indicates which of the Files experienced the problem, and nnnn is the NonStop File System Error number.
BMON Checkpoint integrity check failed - closing down,	These messages would be output if BMON discovers an internal inconsistency after the Backup process assumes responsibility for processing after a CPU failure. This should only occur after a rapid succession of multiple CPU failures, or if the Primary process is stopped twice in succession before a consistent checkpoint had been reached. In these circumstances a RECOVER BMON could be successful, as the internal Checkpoint integrity check is performed before the RECOVERY file is updated or accessed.
BMON MAY be recoverable using RECOVER BMON; consult Manual	These messages would be output if BMON discovers an internal inconsistency after the Backup process assumes responsibility for processing after a CPU failure. This should only occur after a rapid succession of multiple CPU failures, or if the Primary process is stopped twice in succession before a consistent checkpoint had been reached. In these circumstances a RECOVER BMON could be successful, as the internal Checkpoint integrity check is performed before the RECOVERY file is updated or accessed.
BMON process continues as a single process....	These messages would be output if the System Procedure Call CHECKPOINT fails for any reason during checkpointing to BMONs Backup process. BMON will normally continue as a single process, and you may restart a backup process, possibly in a different CPU depending upon the error, by use of CHECKPOINT BMON command. The error numbers are listed in the System Procedure Calls manual.

E.9 Critical Messages

These messages will result in BMON abending. BMON must be a named Process. If an attempt is made to run BMON unnamed, then BMON will abend.

Event Number	Message	Cause / Action
33	Old recovery file not created by BMON	An attempt has been made to "SYNC" (Recover BMON process), with the old Recovery file specified being invalid.
33	Old recovery file created by BMON with a Process name of - "$name "	To SYNC (Recover BMON process), the old file must have been created by a BMON with the same Process name as the current BMON process.

Internal error - "…"

If this type of error ever occurs HPE GNSC should be contacted.

You should make copies of the following files to aid diagnosing the problem:

- BMON configuration file, or a PAK file of the MultiBatch database *CONF files.
- BMON RECOVERY file.
- BMON LOG file (if also logging status to a disk file).
- NonStop EMS Events / Messages.

ALLOCATESEGMENT error

If this type of error ever occurs, Systems Support should be advised.

DEFINEPOOL error

If this type of error ever occurs, Systems Support should be advised.

Can't allocate table from pool

If this type of error ever occurs, Systems Support should be advised.

PUTPOOL error - "…"

If this type of error ever occurs, Systems Support should be advised.

E.10 Event Timer Messages

This section details messages generated by the Event Timer facility, incorporating event messages from TRIGGER, CALENDAR processes and the Event Timer Database Integrity program.

All communication from TRIGGER, CALENDAR processes and the Integrity Program are via the Event Management Subsystem.

EMS Events are generated to reflect the actions performed and responses obtained by these processes when accessing a MultiBatch environment to request execution of a scheduled entity, or when dealing with any other Files or Processes within the Event Timer regime.

All Events have a unique signature, composed of the SubSystem ID (SSID), Event Number and Event Subject. SSID and Event Subject values are set as follows:

TRIGGER

INSIDER.6.0	(SSID)
DS-EVENTTRIGGER	(Subject)

CALENDAR

INSIDER.5.0	(SSID)
DS-EVENTCALENDAR	(Subject)

INTEGRITY

INSIDER.7.0	(SSID)
DS-EventTimer-Integrity	(Subject)

All these MultiBatch Events are delivered with the EMS Emphasis Token set to TRUE.

All Events carry an informative message in the form of an Event Text Token. The Event Numbers and Event Text Token contents for each EMS Event from each process are shown in the following tables.

Where an error is indicated:

The "nnnn" field will carry the NonStop File Error encountered. The <filename> field will carry the name of any File involved.

The <proc> <job> fields will carry the name of the BMON Process and JOB identity concerned.

The <nperror> field will contain an Error Code returned by the call to NEWPROCESS when a non stop Primary process is attempting to create a Backup Process.

The <bstatus> field contains any error obtained when accessing a BMON process. These error numbers are documented in '**Appendix E-2: BMON TO BCOM Messages**'.

TRIGGER Messages

This section details messages generated by the TRIGGER process:

INSIDER.6.0 event^number.

Event Number	Details	Action
500	MultiBatch - ET-Trigger : Trigger process must be named	Rerun the trigger as a named process.
501	MultiBatch - ET-Trigger : Error on <file> CHECKOPEN: NNNN	<file> is the file where CHECKOPEN failed. Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
502	MultiBatch - ET-Trigger : Error on CHECKPOINT NNNN : NNNN	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
503	MultiBatch - ET-Trigger : Backup process started in cpu NN	Information only.
951	MultiBatch - ET-Trigger : Every Schedule <proc> <job> next scheduled run < current time. Re-calculated next run.	Information only.
987	One of: MultiBatch - ET-Trigger : Illegal ETACOLL DATACONF record. <xxxxxxxxxxxx> error: <nnn> MultiBatch - ET-Trigger : Too many ETACOLL DATACONF records - Will use the first 5 only. MultiBatch - ET-Trigger : Illegal ETACOLL DATACONF record - EMS alternate collector must be specified	The alternate collectors that Trigger connects to are listed in the DATACONF as ETACOLLn records. The file names are one of: 1) illegal 2) too numerous, the maximum is 5 3) legal but are not EMS collectors Correct the DATACONF ETACOLLn record and restart the TRIGGER process. <xxxxxxxxxxxx> is the procedure that returned an error. <nnn> is the error number
988	MultiBatch - ET-Trigger : Run integrity check program against duplicate SCHEDULE database	Issued following an error updating the duplicate database.
989	MultiBatch - ET-Trigger : Error on access to duplicate database : nnnn	TRIGGER process is unable to access the duplicate database. Run Integrity check against the duplicate SCHEDULE database.
990	MultiBatch - ET-Trigger : Invalid file specification for duplicate database <filename>	TRIGGER process is unable to access the duplicate database.
991	MultiBatch - ET-Trigger : Scheduled job <proc> <job> held because of HOLIDAY file entry. Next run date recalculated.	Information only.
992	MultiBatch - ET-Trigger : Error on <action> BMON : nnnn - <process>	TRIGGER process is unable to communicate with the BMON process. Check that the BMON process is running.
993	DSETIMER ALARM CANCELLED <proc> <job>	Information only.
994	MultiBatch - ET-Trigger : Error on call to EMS procedure: <s> nnnn. Refer to DSM programming manual for details.	Error on communicating with EMS distributor Contact HPE GNSC. If the error is negative <s> is "_"

995	MultiBatch - ET-Trigger : Error on call to EMS procedure: <s> nnnn. Refer to DSM programming manual for details.	Error on setting up EMS distributor call. Contact HPE GNSC. If the error is negative <s> is "-"
996	MultiBatch - ET-Trigger : Schedule Record Expired - Node: <nnnnnnnn> BMON: <bbbbbbbb> Job/ Seg/Unit: <job.segment.unit> Serial = nn	When a job completes, the next run time is calculated. If the next date is beyond the final run date, then the record is set to expired and this event is generated.
997	MultiBatch - ET-Trigger : Error detected on creation of process EMSDIST – nnnn : nnnn	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
998	MultiBatch - ET-Trigger : TMF error detected : nnnn	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
999	MultiBatch - ET-Trigger : Error nnnn on access to <filename>	nnnn is a standard file error number.
1000	MultiBatch - ET-Trigger : V<number> STARTED : Copyright <year>	Information only.
1100	MultiBatch - ET-Trigger : Process Abended - too many takeovers	This is an internal checkpointing issue. The process has abended to prevent continuous primary/backup takeovers. Restart the Calendar process.
1101	MultiBatch - ET-Trigger : Switchover successful	Information only.
1102	MultiBatch - ET-Trigger : Primary process abended.	Backup is now primary process.
INSIDER. 5.0 various	MultiBatch - ET-Trigger : Error returned from BMON : <ssss> - <process>	An error has occurred when accessing a BMON process. Refer to 'BMON - BCOM Messages' appendix. <ssss> is the status returned from BMON and <process> is the BMON process.

CALENDAR Messages

This section details messages generated by the CALENDAR process:

INSIDER.5.0 event^number.

Event Number	Details	Cause / Action
EMS Reply Code	MultiBatch - ET-Calendar : Error on call to EMS procedure : <sub status>	Check EMS programming manual for <sub status>
100 / 101	MultiBatch - ET-Calendar : Error on access to COLLECTOR $0 <s> nnnn	File open or write error on $0. <s> is either ":" or "-" if value reported is negative. nnnn is the EMS sub status.
500	MultiBatch - ET-Calendar : Calendar process must be named	Rerun calendar as a named process.
501	MultiBatch - ET-Calendar : Error on <file> CHECKOPEN : NNNN	<file> is the file where CHECKOPEN failed. Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
502	MultiBatch - ET-Calendar : Error on CHECKPOINT NNNN : NNNN	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
503	MultiBatch - ET-Calendar : Backup process started in cpu NN	Information only.
504	MultiBatch - ET-Calendar : WARNING - OUT file not an EMS collector - therefore using $0	The OUT file specified in the RUN command is not an EMS collector so using the default which is $0.
900	DSETIMER REQUEST START JOB|SEG|UNIT <name>	Information only.
901	DSETIMER FORCE START JOB|SEG|UNIT <name>	Information only.
993	MultiBatch - ET-Calendar : Run integrity check program against duplicate SCHEDULE database.	Issued following an error updating the duplicate database.
994	MultiBatch - ET-Calendar : Error on access to duplicate database : nnnn	CALENDAR process is unable to access the duplicate database. Run Integrity check against the duplicate SCHEDULE database.
995	MultiBatch - ET-Calendar : Invalid file specification for duplicate database <file name>	CALENDAR process is unable to access the duplicate database.
996	MultiBatch - ET-Calendar : Invalid SCHEDULE record detected	Report the incident to HPE GNSC. A copy of the SCHEDULE file will be required.
997	MultiBatch - ET-Calendar : TMF error detected : nnnn	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
998	DSETIMER ALARM: <name> > x MINUTES	Alarm issued by Event-Timer if the job has been executing for longer than X minutes.
999	MultiBatch - ET-Calendar : Error on access to filename : nnnn	nnnn is a standard file error number.

1000	MultiBatch - ET-Calendar : V<number> STARTED : Copyright <year>	Information only.
1100	MultiBatch - ET-Calendar : Process Abended - too many takeovers	This is an internal checkpointing issue. The process has abended to prevent continuous primary/backup takeovers. Restart the Calendar process.
1101	MultiBatch - ET-Calendar : Switchover successful	Information only.
1102	MultiBatch - ET-Calendar : Primary process abended	Backup is now Primary process

INTEGRITY Messages

This section details messages generated by the INTEGRITY program:

INSIDER.7.0 event^number.

Event Number	Details	Cause / Action
1	!!DS-EventTimer: Database Integrity Program - ERROR ON BEGIN TRANSACTION nnnn	The INTEGRITY program is unable to start a TMF TRANSACTION so aborts.
2	Message !!!DS-EventTimer: Database Integrity Program – ERROR ON FILE <file> STATUS xx:nnnn	The INTEGRITY program cannot access the given file so aborts.
3	!!! DS-EventTimer: Database Integrity Program - NO CHANGES MADE TO DUPLICATE <file>	The INTEGRITY program reports this if no changes have been made.
5	Message !!!DS-EventTimer:Database Integrity Program - UNABLE TO ASSIGN CALENDAR FILE: <cal-file-name>, ERROR nnnn	The INTEGRITY program cannot access the CALENDAR file
6	Message !!!DS-EventTimer: Database Integrity Program - ERROR ON ACCESS TO CALENDAR FILE: <cal-file-name>, ERROR nnnn	The INTEGRITY program cannot access the CALENDAR file
7	Message !!!DS-EventTimer: Database Integrity Program – nnnn CALENDAR FILES REGENERATED	The INTEGRITY program reports that it has updated nnnn calendar files

E.11 Streamer Process Messages

This section details messages generated by the STREAMER (Streamer) facility.

For the STREAMER 'Streamer' process, the EMS SSID and Event Subject is

set to: INSIDER.1.0 (SSID)

MULTIBATCH-STREAMER (Subject)

Event Number	Details	Cause / Action
100	!!! MULTIBATCH STREAMER : Error on call to EMS procedure: - nnnn	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
101	!!! MULTIBATCH STREAMER : Error on call to EMS procedure: - nnnn	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
500	!!! MULTIBATCH STREAMER : Streamer process must be named	Restart the streamer as a named process.
501	"!!! MULTIBATCH STREAMER Error on FFFFFF CHECKOPEN : NNNN where, FFFFFF = "$RECVE" or "$0" or "NONCNF" NNNN = Error code	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
502	!!! MULTIBATCH STREAMER Error on CHECKPOINT AAAA : BBBB where, AAAA = checkpoint status bits 0:7 BBBB = checkpoint status bits 8:15	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
503	!!! MULTIBATCH STREAMER - Backup process started in cpu NN	Information only.
506	!!! MULTIBATCH STREAMER - Inspect prompt received from PPPPP. Inspect stopped, save file FFFFFFFFFFFFFFFFFFFFFFFFFF created. Where, PPPPP Sending process in format $<process> or PID eg 01,210 FFF..FFF Name of the SAVEABEND file.	Sending process will have abended. It may need restarting. Streamer continues processing.
507	!!! MULTIBATCH STREAMER - Debug prompt received from PPPP. Debug stopped. Program filename: FFFFFFFFFFFFFFFFFFFFFFFF Where, PPPPP Sending process in format $<process> or PID eg 01,210 FFF..FFF Name of Object code stopped.	Sending process will have stopped and may need restarting. Streamer continues processing.
977	!!! MULTIBATCH STREAMER - Error on ALLOCATESEGMENT xxxx : xxxx	If this happens then the streamer process abends.
978	!!! MULTIBATCH STREAMER running with ALIASMAP file (xxxxx)	Information only.
979	!!! MULTIBATCH STREAMER running without ALIASMAP file	Information only.
980	!!! MULTIBATCH STREAMER Error on UPDATE of NON^CONFIG file EEE FFFFFFFFFFFFFFFF Where, EEE GUARDIAN file system error FFF..FFF File name	Non^Config records file is closed. Streamer continues processing.

981	!!! MULTIBATCH STREAMER - Streamer process continuing with existing configuration	An attempted refresh failed because access to the updated streamer configuration file failed. Fix the underlying File system error and reissue “MBATCOM $<STREAMER>; REFRESH” command.
982	!!! MULTIBATCH STREAMER Following message will be suppressed until HH:MM:SS	Information only. Event threshold has been reached.
983	!!! MULTIBATCH STREAMER Following message suppressed NNNN times in SSSS seconds	Information only. Threshold period has elapsed. Event messages will be sent to EMS until Threshold is reached again.
984	!!! MULTIBATCH STREAMER Failure to access Output file N EEE FFFFFFFFFFFFFFFFFFFFFFFFF Where, N 1 or 2 EEE GUARDIAN file system error FFF..FFF File name	The text portion of the EMS Event could not be written to a designated device. EMS event is still generated and streamer continues processing. Fix underlying File system error, or remove devices from streamer records in PATHWAY and refresh streamer.
985	!!! MULTIBATCH STREAMER Illegal Output File FFFFFFFFFFFFFFFFFFFFFFFFFFFFF Where, FFFF... FFFFF File name	Designated output device has illegal file name. Alter name in streamer record via PATHWAY system and reload the streamer process.
986	!!! MULTIBATCH STREAMER running with ‘Non Configured Records’ file	Information only. Any records not found in translation tables will be reported in EMS and written away to a disk file.
987	!!! MULTIBATCH STREAMER Illegal ‘Non Configured Records’ file:Type E:Len 255”;	An attempt to open an existing Non config records file failed because it was not of the right attributes. This is to prevent overwriting an existing data file. Create a valid file or specify a non-existent file and it will be created for you.
988	!!! MULTIBATCH STREAMER Error on CREATE of NON^CONFIG file EEE FFFFFFFFFFFFFFF Where, EEE GUARDIAN file system Error FFFF...FFFFF Filename	Creation of a New Non^config file failed (e.g. disk unavailable). If this happens at streamer startup then the streamer abends.
989	!!! EVENT STREAMER Error on OPEN of NON^CONFIG file EEE FFFFFFFFFFFFFFFFFFF Where, EEE GUARDIAN file system Error FFFF... FFFFF File Name	Creation of New Non config file failed (e.g. disk down). If this happens at streamer startup then the streamer abends.
990	!!! MULTIBATCH STREAMER Illegal ‘Non Configured Records’ file FFFFFFFFFFFFFFFF Where, FFFF...FFFFF File Name	Open of New Non config file failed because the filename was illegal. If this happens streamer startup then the streamer abends.
991	!!! MULTIBATCH STREAMER running without ‘Non Configured Records’ file	Information only. Any records not found in translation tables will be reported in EMS but will not be written away to a disk file. If you wish to use the NON^CONFIG option, then stop the streamer and restart with the NON^CONFIG parameter.
992	!!! MULTIBATCH STREAMER - Connected to Collector $PPPPP Where, $PPPPP Collector Name	Information only. You can route your events to an alternate collector using the OUT option of the streamer (eg. RUN STREAMER / OUT $ACOL,........etc /)
993	!!! MULTIBATCH STREAMER - Error: INFILE is invalid/ old format config file	The streamer configuration file specified in the IN run option is of an invalid format. It must be a key sequenced file and, from MultiBatch Release 6.0, 998 bytes long. The streamer process will abend. Restart with the correct file.

994	!!! MULTIBATCH STREAMER - Error: Incorrect mixture of FIXED and IGNORE/EVENT config. records detected";	The streamer configuration contains a mixture of Event ("E") and Fixed ("F") configuration records. If the Fixed option is selected it must be the only record on the file.
995	!!! MULTIBATCH STREAMER - Error: INFILE is not a disk file"	The streamer configuration file, specified in the IN option is not a disk file. The streamer will abend. Restart with correct streamer config file.
996	!!! MULTIBATCH STREAMER - Error on access to <filename> : NNNN Where, NNNN GUARDIAN file system error.	File error has occurred on specified file. The streamer will abend. Restart once error has been resolved
997	!!! MULTIBATCH STREAMER - Error: STREAMER configuration file is empty	The Streamer configuration file is empty. Streamer will abend. Restart with correct streamer config file.
998	!!! MULTIBATCH STREAMER - Warning: Maximum number of streamer entries read from INFILE - process continues....	Streamer has read 1000 entries from the streamer config file into internal tables. This is the maximum size. However, the file contains more records. Stop and reconfigure the streamer if this is inappropriate.
999	!!! MULTIBATCH STREAMER - No streamer record found: XXXXXXXXX...XXXXXXXXXX Where, XXXXX..XXXXXX Up to 193 Bytes of the text message	The streamer was unable to turn this into an event, because no translation rules exist for it in the streamer config file. Streamer continues processing. If the NON^CONFIG option is being used, this text will be written away to a disk file.
1000	!!! MULTIBATCH STREAMER-V<number> : Copyright <date>	Information only.
1100	!!! MULTIBATCH-STREAMER Process.- Process Abended - too many takeovers.	This is an internal checkpointing issue. The process has abended to prevent continuous primary/backup takeovers. Restart the Streamer process.
1101	!!! MULTIBATCH-STREAMER Process. - Switchover successful.	Information only.

E.12 Watcher Messages

All Watcher error messages are written to $0 and comprise the subsystem affected, the event number and the error message. The messages appear as follows:

For the WATCHER process, the EMS SSID and Event Subject are

set to: INSIDER.15.0 (SSID)

MULTIBATCH (Subject)

The Watcher error messages are detailed below.

Event Number	Details	Cause / Action
1	MULTIBATCH - Watcher : Unable to allocate EXTENDED memory. Error #nnn	The watcher process is unable to allocate memory so is not able to continue. Contact HPE GNSC.
5	MULTIBATCH - Watcher : The number of BMONs configured exceeds the maximum value. Events for some BMONs may be discarded.	The number of BMONS which WATCHER is monitoring exceeds its maximum value. So events pertaining to these additional BMONs will be ignored.
6	MULTIBATCH - Watcher : The number of JOBS configured exceeds the maximum value. Events for some JOBS may be discarded.	The number of JOBs which WATCHER is monitoring exceeds its maximum value. So events pertaining to these additional JOBs will be ignored.
7	MULTIBATCH - Watcher : The number of Segments configured exceeds the maximum value. Events for some SEGMENTS may be discarded.	The number of SEGMENTs which WATCHER is monitoring exceeds its maximum value. So events pertaining to these additional SEGMENTs will be ignored.
8	MULTIBATCH - Watcher : The number of STEPs configured exceeds the maximum value. Events for some STEPs may be discarded.	The number of STEPs which WATCHER is monitoring exceeds its maximum value. So events pertaining to these additional STEPs will be ignored.
9	MULTIBATCH - Watcher : Error during access of LOGGEDON : nnnn. Status terminals not updated.	Watcher cannot access the LOGGED file. Therefore PATHWAY STATUS screens will not be updated.
10	MULTIBATCH - Watcher : Error nnnn attempting access to TCP.	Watcher cannot access the PATHWAY TCP process. Therefore PATHWAY STATUS screens will not be updated.
11	MULTIBATCH - Watcher : Error nnnn attempting to write to TCP terminal.	Watcher cannot write to the PATHWAY TCP terminal. Therefore PATHWAY STATUS screens will not be updated.
12	MULTIBATCH - Watcher : Watcher failed in its attempt to send an unsolicited message to a MultiBatch PATHWAY screen showing the SOV or equivalent screen.	Watcher cannot access the PATHWAY TCP process. Therefore PATHWAY STATUS screens will not be updated.
16	MULTIBATCH - Watcher : aaaaaaaa record not defined in configuration file : <filename>	aaaaaaaa record does not exist in the DATACONF file where aaaaaaaa is either JOBSTAT/SEGSTAT/STEPSTAT. Add these records to DATACONF and restart WATCHER.

17	MULTIBATCH - Watcher : Inconsistent counts: xxxx- STAT/zzz/nnnn nnnnnnnn.pppppppp.jjjj.ss.uu	WATCHER has detected that there are inconsistent counts in either JOBSTAT/SEGSTAT/STEPSTAT. Contact HPE GNSC.
19	MULTIBATCH - Watcher : Backup process has started successfully	Watcher backup process started. Information message
19	MULTIBATCH - Watcher : Unable to start backup process. Error #	The primary WATCHER process has not been able to start its backup process. It will continue processing as a single process. NCOM can be used to try starting a BACKUP process when the error has been resolved. See Appendix F.8.
19	MULTIBATCH - Watcher : Unable to check EXTENDED memory. Error #nnn	The primary WATCHER process has not been able to checkpoint data to it backup process.
20	MULTIBATCH - Watcher : Invalid LOGTIME parameter. LOGTIME must be DD/MM/YYYY HH:MM:SS	The LOGTIME parameter specified in the RUN command is not valid. Restart the Watcher process with corrected startup parameters.
22	MULTIBATCH - Watcher : An alter step message could not be reflected in the status database. Event text token is unavailable.	An ALTER STEP command has been issued. WATCHER cannot process this event
25	MULTIBATCH - Watcher : Unable to communicate with workstation server.	Watcher failed in its attempt to send an unsolicited messages to a MultiBatch PATHWAY screen showing SOV or an equivalent screen.
30	MULTIBATCH - Watcher : Could not connect to collector aaaaaaaaaaaaaaaaaaaaaaa (nnnn)	The WATCHER process cannot connect to an alternate collector. Note: The alternate collectors that Watcher connects to are listed in DATACONF as a sequence of ALTCOLLn records.
32	MULTIBATCH - Watcher : Illegal ALTCOLL DATACONF record. FILE_GETINFOBYNAME error: nnnn	The WATCHER process has detected an error in one of the ALTCOLLn records in the DATACONF file. The file name is illegal
34	MULTIBATCH - Watcher : Too many ALTCOLL DATACONF records - Will use the first 5 only.	The WATCHER process has detected that there are too many ALTCOLLn records in the DATACONF file.
36	MULTIBATCH - Watcher : Illegal ALTCOLL DATACONF record - EMS alternate collector must be specified.	The WATCHER process has detected an error in one of the ALTCOLLn records in the DATACONF file. The file name specified is not an EMS collector.
46	MULTIBATCH - Watcher : A JOB added interactively cannot be monitored as the maximum number of JOBs is already configured.	The internal WATCHER table sizes have been exceeded and in some circumstances jobs, segments and units added interactively cannot be monitored. EMS events for these elements will be ignored.
47	MULTIBATCH - Watcher : A SEGMENT added interactively cannot be monitored as the maximum number of SEGMENTs is already configured	The internal WATCHER table sizes have been exceeded and in some circumstances jobs, segments and units added interactively cannot be monitored. EMS events for these elements will be ignored.

48	MULTIBATCH - Watcher : A STEP added interactively cannot be monitored as the maximum number of STEPs is already configured.	The internal WATCHER table sizes have been exceeded and in some circumstances jobs, segments and units added interactively cannot be monitored. EMS events for these elements will be ignored.
500	MULTIBATCH - Watcher : Process must be named.	Specify a NonStop Process Name when starting Watcher.
501	MULTIBATCH - Watcher : Error on CHECKPOINT xxxx. xxxx	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
502	MULTIBATCH - Watcher : Error on CHECKOPEN <filename>:nnnn	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
994 / 995	MULTIBATCH - Watcher : Error on call to EMS procedure: nnnn	Refer to DSM Programming Manual for details. Note: Error 994 refers to an SPI error Error 995 refers to an EMS error.
997	MULTIBATCH - Watcher : Process creation failure: nnnn.nnnn	The WATCHER is unable to create an EMS distributor process. Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
998	MULTIBATCH - Watcher : TMF error : nnnn	The WATCHER process encountered a TMF error when trying to write to the ERRLOG file. Codes are documented in System Procedure Calls manual.
999	MULTIBATCH - Watcher : File error : nnnn	Codes are documented in System Procedure Calls manual.
1000	MULTIBATCH - Watcher : V<number> : Started	Information only.
1100	MULTIBATCH - Watcher : Process Abended - too many takeovers.	This is an internal checkpointing issue. The process has abended to prevent continuous primary/backup takeovers. Restart the Watcher process.
1101	MULTIBATCH - Watcher : Switchover successful.	Information only.
1102	MULTIBATCH - Watcher : Primary Process abended.	Primary process stopped, Backup process becomes Primary and new Backup process created.
1998	MULTIBATCH - Watcher : Error starttime is zero: \nnnnnnn.$ppppp : (eeee,a,a)	The recalculation of the statistics because of the completion of a job, will lead to corrupt figures. The trace information will highlight the section within Watcher where this corruption occurs. Send this message along with copies of the BMONSTAT, JOBSTAT, SEGSTAT and STEPSTAT files to HPE GNSC.

1999	MULTIBATCH - Watcher : Error decrementing stat counts : \nnnnnnn.$ppppp : (eeee,a,a)	The recalculation of the statistics because of the completion of a job, will lead to corrupt figures. The trace information will highlight the section within Watcher where this corruption occurs. Send this message along with copies of the BMONSTAT, JOBSTAT, SEGSTAT and STEPSTAT files to HPE GNSC. The SOV Statistics can be reset by rebuilding the BMON, e.g. via an SBMON obey file, which includes running an STBUILD. STBUILD will clear down the starts ready for a BMON rebuild. See Chapter H 'Recovering a BMON' for example contents of an SBMON file.

E.13 Migrator Messages

The Migrator Error Messages contain two types of messages, namely, information messages and error messages.

Information Messages

When running MIGRATOR the following information messages are generated, example:

***** MIGRATOR – Preview Complete *****

The "preview" details will be in the nominated spooler location.

***** MIGRATOR – Preview Complete: No alias changes/duplicates detected *****

No "preview" report will be generated in these circumstances.

When the Migrator terminates successfully the following processing summary message is displayed on the Migrator home terminal.

```
Migrator - <Operation> \LIVE.$BMON: Number of Jobs processed = nnn
Migrator - <Operation> \LIVE.$BMON: Number of Segs processed = nnn
Migrator - <Operation> \LIVE.$BMON: Number of Steps processed = nnn
Migrator - <Operation> \LIVE.$BMON: Complete.
```

<Operation> can be Extract, Insert or Remove:

*** MIGRATOR - Extraction complete***

*** MIGRATOR - Insertion complete***

*** MIGRATOR - Removal complete***

ERROR MESSAGES

The error messages generated by the Migrator facility are given below.

1) STARTUP PROCESSING:

Invalid param string

Invalid process name

selected Invalid process name

Invalid job name

Invalid segment name Invalid unit name

Replacement values only permitted for INSERT Invalid NODE specified

Invalid PROCESS specified TRANSIT : Transit file empty

ERRORTransit incompatible with MIGRATOR version

ERRORInsert on different level to transit file

2) MIGDEFS PROCESSING:

Unable to EDITREADINIT MIGDEFS

Unable to EDITREAD MIGDEFS

Too many NODE defines

Too many VOLUME

defines Too many

SUBVOL defines Too

many BMON defines Too

many ALIAS defines

Unable to READ TRANSIT file

MIGDEFS <new/old> <VOLUME/BMON> must begin with '$' < input buffer>

MIGDEFS <new/old> NODE must begin with '\' <input buffer>

MIGDEFS <new/old> <ALIAS/SUBVOL> must begin with a non space character: <input

buffer> MIGDEFS new and old ALIAS values must not be spaces

Preview requested but no ALIAS values specified in MIGDEFS

3) MAIN PROCESSING:

SEGDEP insertion failed: Seg XXXX.XX with link (P/N) to nonexistent segment YYYY.YY New alias will be > 30 characters <old-alias-name>

Unable to KEYPOSITION on dataconf file (ALT) Unable to READ dataconf file

Unable to KEYPOSITION on dataconf file (AS) Unable to open <filename>

Unable to POSITION on <filename>

Unable to READUPDATE <filename>

Unable to WRITEUPDATE <filename>

TRANSIT : Error on <filename> write Data inconsistency, = <define name>

TRANSIT : Error reading transit file <Filename> : Error on WRITE: nnnn, where nnnn is the associated Guardian error code

Attempting to insert segment <segment id> with link to non-existent segment <segment id> Invalid TRANSIT record detected

Unable to WRITEUPDATE <filename>

Unable to OPEN $RECEIVE

INFILE must be DATACONF file

Unable to ALLOCATESEGMENT

```
TRANSIT : Error on Operation depth write Unable to open TRANSIT file
JOBDEP insertion failed: JOB XXXX with link (P/N) to nonexistent job YYYY
```

E.14 Clockmon Error Messages

All Clockmon error messages are written to $0 and comprise the subsystem affected, the event number and the error message.

For the CLOCKMON process, the SSID and Event Subject is:

set to: INSIDER.5.0	(SSID)
CLOCKMON	(Subject)

The CLOCKMON error messages are detailed below.

Event Number	Details	Cause / Action
event number as specified in STEPMON record	MULTIBATCH - Clockmon : STEPMON record not on STEPSTAT file : <step name>	CLOCKMON process issues this event when the STEPSTAT record corresponding to the STEPMON record does not exist. Need to run STBUILD program to build the *STAT files.
event number as specified in STEPMON record	MULTIBATCH - Clockmon : warning: start time reached for : <step name>	CLOCKMON process issues this event when it detects that the time specified in the STEPMON record has passed.
1-9	MULTIBATCH - Clockmon : Error on call to EMS procedure : <sub status>	Check EMS programming manual for <error code>
100/101	MULTIBATCH - Clockmon : Error on access to COLLECTOR (<coll name>)	File open or write error on <coll name>
500	MULTIBATCH - Clockmon : process must be named	Supply a Guardian Process name when starting CLOCKMON.
501	MULTIBATCH - Clockmon : Error on CHECKOPEN for <filename>	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
502	MULTIBATCH - Clockmon : Error on CHECKPOINT : nnnn nnnn	Codes are documented in System Procedure Calls manual. Contact HPE GNSC.
503	MULTIBATCH - Clockmon : Backup process started in cpu : nn	CLOCKMON Backup process has started.
900	MULTIBATCH - Clockmon : Tracking buffer has maximum entries(nnnn)	Warning message, CLOCKMON can only monitor nnnn steps for any one time
999	MULTIBATCH - Clockmon : File error nnnn on <filename>	nnnn is a Guardian error number
1000	MULTIBATCH - Clockmon : V<number> STARTED : Copyright <year> HPE NonStop	Information only.

1100	MULTIBATCH - Clockmon : Process Abended - too many takeovers	This is an internal checkpointing issue. The process has abended to prevent continuous primary/backup takeovers. Restart the Clockmon process.
1101	MULTIBATCH - Clockmon : Switchover successful	Information only.
1102	MULTIBATCH - Clockmon : Primary process abended.	Backup is now primary process.

E.15 MBATCOM Messages

MBATCOM allows access to the MultiBatch NonStop processes. Commands can be entered interactively or using an obey file. If commands are entered interactively, then error messages are returned directly. If an obey file is used and an error occurs, then the error message is written to the outfile and an EMS event is generated containing the error text.

For the MBATCOM process, the SSID and Event Subject is:
MULTIBAT.2.0 (SSID)
MULTIBAT (Event Subject)

The MBATCOM error messages are detailed below.

Event Number	Details
501	Unable to open specified PROCESS
502	No current PROCESS open
513	There is no PROCESS currently open
514	Unable to access Process
515	Incompatible releases of MBATCOM & PROCESS
518	Unable to open initial PROCESS
519	Failure on EDITREADINIT
546	Invalid filename specified
546	Invalid filename specified
547	OBEY FILE error nnnn
548	Specified file is not an edit file
549	Error on EDITREAD of obey file
550	FC not permitted in obey file
551	OBEY FILE is empty
552	Nested OBEY commands detected
593	History Pool Buffer PUTSPACE error
594	History Pool Buffer GETSPACE error

Appendix F MultiBatch Utilities

F.1 ALRECON

An Alias name is recorded in a lookup table - ALIASMAP file - and is also registered as an alternate key in the BMONCONF, JOBCONF, SEGCONF and STEPCONF files.

If this pairing gets out of step, then subsequent updates via the PATHWAY configuration facility can be rejected with an "ALIASMAP record not found" diagnostic. The ALRECON utility allows a user to fix this database inconsistency.

The command syntax is:

```
      RUN ALRECON / IN <MultiBatch-database>.DATACONF / <execution-mode>
```

Where:

`<execution-mode>` is either UPDATE or REPORT.

If no parameter is specified, it will run in REPORT mode.
Any non UPDATE value is treated as REPORT mode.

REPORT mode displays the discrepancies on your home terminal, but does not update the database.

Note: *It is recommended that you execute the utility in REPORT mode first before updating the database.*

UPDATE mode displays the discrepancies on your home terminal, and also updates the database. An example report follows:

```
ALRECON RUNNING IN UPDATE MODE

ALIAS HD-HOUSEKEEPING                 CREATED FOR <NODE>$HDBAT
ALIAS J-DAILY-MEASURE                 CREATED FOR <NODE>$HDBAT MEAS
ALIAS FULL-WEEKLY-CHECK-S02           CREATED FOR <NODE>$HDBAT FCH2AA
ALIAS DAILY-CAPITAL-TRACKS-TEST-U     CREATED FOR <NODE>$HDBAT CAPTAA01
NUMBER OF CHANGES MADE: 0004

ALRECON COMPLETE
```

If during an attempt to create the alias record, the alias name has already been allocated to a different job/segment/unit, a message will be displayed, e.g.:

```
!! <NODE> $HDBAT SAFEAA ALIAS CHECK-SAFEGUARD-SUPER-SUPER-S EXISTS FOR <NODE> $HDBAT FCH1AA
```

F.2 AUDLOGHK

A user can delete or archive information from the audit detail (AUDDET) and the audit summary (AUDLOG) files. These two files hold the details of PATHWAY based maintenance actions: insert, delete and amendment of records.

To list out the syntax, enter TACL command: RUN AUDLOGHK

```
  MultiBatch AUDLOG house keeping routine

  AUDLOGHK/ IN <dataconf-file>, OUT <list-file>/ Options

  IN      - Location of the DATACONF file
  OUT     - Listing file
  Options - A,B,C
            A: PREVIEW/DELETE[!]/ARCHIVE[!]
            B: DDMMMYYYY HHMMSS : Date spec
               or DDD : Number of days to retain
            C: Archive spec AUDLOG & AUDDET archive files

e.g. RUN AUDLOGHK/IN DATACONF,OUT $S.#LIST/PREVIEW,01MAR2019 100000
```

Note: *Even when running in PREVIEW mode, the date and time must be specified, e.g.:*

```
RUN AUDLOGHK/IN MBATDAT.DATACONF/PREVIEW,14JAN2019 100000

AUDLOG Date Range : 07 JAN 2019 11:40:20 to 14 JAN 2019 10:00:00
408 AUDLOG records are within the date range.

AUDDET Date Range : 07 JAN 2019 11:40:21 to 14 JAN 2019 10:00:00
1224 AUDDET records are within the date range.
```

To implement the archiving facility, you will need to create a copy of the AUDLOG and AUDDET archive files and they must be empty. If the archived files contain records during the ARCHIVE function, then a message will be issued, e.g.:

"AUDLOG Archive file must be empty to prevent problems with duplicate records" or
"AUDDET Archive file must be empty to prevent problems with duplicate records"

In some cases, the AUDLOG and AUDDET files may have been opened by a server class as a user is logged onto the MultiBatch Pathway. Therefore, it is recommended to create a copy of the AUDLOG and AUDDET files.

The files will have the same attributes as the original files but the alternate keys are not required, e.g.:

```
FUP
-SET LIKE AUDLOG
-RESET ALTFILE
-RESET ALTKEY
-CREATE LOG1203
```

The files can also be renamed to something else and used as in the example below:

```
RUN AUDLOGHK/IN MBATDAT.DATACONF, OUT $S.#LIST/ARCHIVE,&
01MAR2019 100000, MBATDAT.LOG1203 & MBATDAT.DET1203
```

Note: *Keep to the sequence of Options when archiving the audit files, i.e.:*

```
ARCHIVE, DATE <DDMMYYY HHMMSS>, <Copy AUDLOG file> & <Copy AUDDET file>
```

F.3 AUDREPT

The Audit Detail File information can be viewed at any time by using the provided print program, AUDREPT. To execute, enter the following at the TACL prompt:

```
RUN AUDREPT / IN <Database Svol>.DATACONF, OUT <spooler location> / {param}
```

where {param) must be one of the following:

ALL	- for the whole Audit Detail File
FROM <dd/mm/yyyy, hh:mm>	- only from the specified date
TO <dd/mm/yyyy, hh:mm>	- only to the specified date

Or

FROM <dd/mm/yyyy, hh:mm> **TO** <dd/mm/yyyy, hh:mm>

Note: *The Auditing facility may be turned off by setting a parameter in the **PWCONF** file.*

*This Pathway parameter is called "**SET SERVER PARAM AUDIT-FLAG**".*

It can be set to Y, N or S as required. If the parameter is set to 'S' , then only summary information will be collected and held in the Audit Log File.

F.4 BCFORMAT

When executing an INFO JOB, SEGMENT, or UNIT command via BCOM, basic details are returned, unless a user specifies, DETAIL parameter after the INFO command, e.g. INFO JOB JOBA, DETAIL. In which case, all configured STEP attributes of a job will be supplied.

A user however, can alternatively specify which STEP attributes are returned without the use of a, DETAIL parameter via means of an edit file called BCFORMAT.

This file is located in the object subvolume, e.g. MBATOBJ and the format is:

- Columns 1 to 10 - **Field Name**
- Column 11 - **Display Flag**, being either Y or N Example content:

```
* BCFORMAT file
* used to customize what is displayed by BCOM
* for a INFO JOB command
*
args      N
assigns   Y
critical  N
cpu       Y
default   Y
defines   Y
fail      N
fdinfo    N
highpin   N
infile    Y
library   Y
max       Y
mem       Y
name      Y
object    Y
outfile   Y
param     Y
rtparams  Y
pathname  Y
pri       Y
program   Y
skip      N
swap      N
term      Y
text      Y
```

The rules for how BCOM locates the BCFORMAT edit file are either:

1. Through a =BCFORMAT Define, e.g.

```
ADD DEFINE =BCFORMAT, FILE <VOL.SUBVOL>.BCFORMAT
```

2. Or, the BCFORMAT file is located in the MultiBatch object subvolume, e.g. MBATOBJ.

If BCFORMAT cannot be located or opened, then BCOM will still execute and an INFO JOB results will be displayed as follows:

```
>>INFO JOB JOBE

Job    - JOBE, Maxsegs    = 001, Numsegs    = 001
              Rerun = ON , Groupid = BMON
Segment - AA, Maxunits   = 002, Numunits   = 002
              Maxdepends = 000, Numdepends = 000
Unit - 01.1,  Cpus = 00:01, Priority = 100, Memory = 000,
              Highpin        - OFF
              Object file    - \LIVE.$SYSTEM.SYSTEM.TACL
```

An example of using a BCFORMAT file is shown on the next page.

BCFORMAT has Highpin set to N and all other lines set to Y and the results of an INFO JOB will contain:

```
1>>INFO JOB JOBE

Job    - JOBE, Maxsegs    = 001, Numsegs    = 001
              Rerun = ON , Groupid = BMON
Segment - AA, Maxunits   = 002, Numunits   = 002
              Maxdepends = 000, Numdepends = 000
Unit - 01.1, Cpus = 00:01, Priority = 100, Memory = 000,
              Object file     - <NODE>.$SYSTEM.SYSTEM.TACL
              Default         - $MBAT.BMON
              Input file      - \LIVE.$MULTIBATCH.BMON.PROMPT
              Output file     - \LIVE.$S.#JOBE
              Home terminal   - \LIVE.$ZHOME
              Run time params - “”
              Text            - “”
Unit - 02.1, Cpus = 00:01, Priority = 100, Memory = 000,
              Pathname        - /bin/
              Program file    - sleep
              Text            -“”
```

BCFORMAT has all lines including Highpin set to Y and the results of an INFO JOB will contain:

```
>>INFO JOB JOBE

Job    - JOBE, Maxsegs    = 001, Numsegs    = 001
              Rerun = ON , Groupid = BMON
Segment - AA, Maxunits   = 002, Numunits   = 002
              Maxdepends = 000, Numdepends = 000
Unit - 01.1,  Cpus = 00:01, Priority = 100, Memory = 000,
              Highpin         - OFF
              Object file     - \LIVE.$SYSTEM.SYSTEM.TACL
              Default         - $MBAT.BMON
              Input file      - \LIVE.$MBAT.BMON.PROMPT
              Output file     - \LIVE.$S.#JOBE
              Home terminal   - \LIVE.$ZHOME
              Run time params - “”
              Text            - “”
Unit - 02.1, Cpus = 00:01, Priority = 100, Memory = 000,
              Highpin         - OFF
              Pathname        - /bin/
              Program file    - sleep
              Text            - “”
```

F.5 ERRLOGHK

A user can delete or archive information from the Error Log (ERRLOG) file.

The file holds details of MultiBatch run time errors issued by BMON and which are captured by the Watcher process. Previously you had to stop the MultiBatch software, then COPY and PURGEDATA the file.
To list out the syntax key RUN ERRLOGHK:

```
MultiBatch ERRLOG house keeping routine
---------------------------------------
  ERRLOGHK/ IN <dataconf-file>, OUT <list-file>/ Options
  IN      - Location of the DATACONF file
  OUT     - Listing file
  Options - A,B,C
            A: PREVIEW/DELETE[!]/ARCHIVE[!]
            B: DDMMMYYYY HHMMSS : Date spec
               or DDD : Number of days to retain
            C: Archive spec ERRLOG archive file

e.g. RUN ERRLOGHK/IN DATACONF,OUT $S.#LIST/PREVIEW,02MAR2010 100000
```

Note: *Even when running in PREVIEW mode, the date and time must be specified, e.g.:*

```
RUN ERRLOGHK/IN MBATDAT.DATACONF/PREVIEW, 14JAN2019 10000

ERRLOG Date Range : 07 JAN 2019 12:56:46 to 14 JAN 2019 10:00:00
61 ERRLOG records are within the date range.
```

To implement the archiving facility, you will need to create a copy of the ERRLOG archive file and it must be empty.

The file will have the same attributes, (including the **AUDIT** setting) as the original file but the alternate keys are notrequired, e.g.:

```
FUP
-SET LIKE ERRLOG
-RESET ALTFILE
-RESET ALTKEY
-CREATE ERR1203
```

The file can also be named to something else, as in the example below:

```
RUN ERRLOGHK / IN MBATDAT.DATACONF, OUT $S.#LIST /ARCHIVE,01MAR2019 1603000,&
MBATDAT.ERR1203
```

If the archived files contain records during the ARCHIVE function, then a message will be issued, e.g.:

```
"ERRLOG Archive file must be empty to prevent problems with duplicate records"
```

F.6 ETPRINT

The ETPRINT utility is used to report on the contents of the Event Timer (EME) SCHEDULE file, where all 'AT', 'EVERY', 'CRONTAB' entries are stored.

The ETPRINT utility requires an Assign to be specified in TACL before it is run.

The Assign value must refer to the SCHEDULE file which is located in the MultiBatch Database subvolume, e.g.:

```
ASSIGN SCHEDULE, <DISK>.MBATDAT.SCHEDULE
```

After the ASSIGN has been specified, the utility can then be run:

```
RUN ETPRINT
```

On completion, the TACL prompt will be returned and results can be viewed through Peruse.

Note: *The spooler location is fixed at $S.#SCHED.*

The spooler job will contain a report on the configuration details for all AT, CRONTAB, EVERY entry. Examples of reports are provided on the following pages.

'AT' SCHEDULE

This AT entry contains a schedule for JOBA (no Segment, or Unit).

- It is Active.
- It is Scheduled.
- Runs at 15:35 hours.
- Runs seven days of the week.
- No specific days, dates, weekend, or weekdays specified.
- Runs Forever (no expiry date).
- Will also run on dates in any Event Timer configured Holidays.

```
JOB...............  JOBA
SEGMENT...........
UNIT..............  00
PROCESS...........  \LIVE$BMON
SERIAL............  01
TYPE..............  A                                        A: AT
ACTIVE/EXPIRED....  A                                        A: ACTIVE
VERSION. .........  07
CALENDAR FILE.....
INSERTED..........  2018 03 12 15 44 26
LAST..............  9999 99 99 99 99 99                      RUN FOREVER
REQUESTED.........  2018 07 01 15 35 03
FINISHED..........  2018 07 01 15 35 03
NEXT..............  2018 07 02 15 35 00
ALARM TRIGGERED...  0000 00 00 00 00 00
PARAMS............  \LIVE$BMON; START JOB JOBA
STATUS............  S                                        S: SCHEDULED
FORCE. ...........  N
AT TIME...........  15 35 00                                 START TIME
ON DAY............
ON DATE 1.........  00 00 00
ON DATE 2.........  00 00 00
ON DATE 3.........  00 00 00
ON DATE 4.........  00 00 00
ON DATE 5.........  00 00 00
ON DATE 6.........  00 00 00
NOT DAY...........
NOT DATE 1........  00 00 00
NOT DATE 2........  00 00 00
NOT DATE 3........  00 00 00
NOT DATE 4........  00 00 00
NOT DATE 6........  00 00 00
UNTIL DAY.........
UNTIL DATE........  0000 00 00
UNTIL FOREVER.....  Y                                       Y: RUN FOREVER
ALARM.............  00000
ALARM FLAG........  N
RUN ON HOLIDAY..... Y
HOLD AFTER RUN..... N
```

CRONTAB Schedule

This CRONTAB entry contains a schedule for EOD (no Segment, or Unit).

- It is Active.
- Job has been Held.
- Runs at 17:30 hours.
- Runs up to 31 days every month.
- Runs 12 months of the year.
- Only runs on weekdays.
- Duration is: From 17:28:00 on 02/07/2018, Until 18;00:00 on 02/05/2019.

Note: *The FROM Date/Time does not indicate its next run; the actual start date/time is indicated by the Y flags. So, its first run time after insertion will be 17:30 and not 17:28.*

The UNTIL Date/Time does indicate its last run time. So, its last run time in this example will be 18:00, unless the last UNTIL time matches the configured 17:30 time.

```
JOB...............  EOD
SEGMENT...........
UNIT..............  00
PROCESS...........  \LIVE$BMON
SERIAL............  01
TYPE..............  C                                       C: CRONTAB
ACTIVE/EXPIRED....  A                                       A: ACTIVE
VERSION...........  07
CALENDAR FILE.....
INSERTED..........  2018 07 02 17 25 12
LAST..............  2019 05 02 18 00 00
REQUESTED.........  2018 07 02 17 30 02
FINISHED..........  2018 04 25 15 30 28
NEXT..............  2018 07 02 17 30 00
ALARM TRIGGERED...  0000 00 00 00 00 00
PARAMS............  \LIVE$BMON;START JOB EOD
STATUS............  H                                       H: HELD
FORCE.............  N
FROM..............  2018 07 02 17 28 00
UNTIL.............  2019 05 02 18 00 00

                              1         2         3         4         5
                    0....5....0....5....0....5....0....5....0....5....0....5...
MINS.............. ______________________________Y_____________________________

                              1         2
                    0....5....0....5....0...4
HOURS............. _________________Y______

                             1         2         3
                    1...5....0....5....0....5....0.
DATES.............. YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY

                    JFMAMJJASOND
MONTHS............. YYYYYYYYYYYY

                    SMTWTFS
DAYS.............. _YYYYY_

ALARM.............. 00000
```

EVERY Schedule

This EVERY entry contains a schedule for AAAA (no Segment, or Unit).

- It is Active.
- It is currently Running.
- Job runs Every 12 Minutes
- Duration is: From: 10:45 on 03/07/2018, until 21:00 on 21/02/2019
- Only runs between the hours of 10:45 and 21:00 hours.

```
JOB...............  AAAA
SEGMENT...........
UNIT..............  00
PROCESS...........  \LIVE$BMON
SERIAL............  01
TYPE..............  E                     E: EVERY
ACTIVE/EXPIRED....  A                     A: ACTIVE
VERSION...........  07
CALENDAR FILE.....
INSERTED..........  2018 07 03 10 39 58
LAST..............  2019 02 21 21 00 00
REQUESTED.........  2018 07 03 11 09 06
FINISHED..........  2018 07 03 11 07 03
NEXT..............  2018 07 03 11 09 00
ALARM TRIGGERED...  0000 00 00 00 00 00
PARAMS............  \LIVE$BMON;START JOB AAAA
STATUS............  R                     R: RUNNING
FORCE.............  N
FROM..............  2018 07 03 10 45 00
UNTIL.............  2019 02 21 21 00 00
BETWEEN...........  0010 45 00 21 00 00
INTERVAL..........  0012
INTERVAL TYPE.....  M                     M: MINUTES
ALARM.............  00000
ALARM FLAG........  N
```

F.7 GETPRMPT

Within a TACL obey file or macro, it is possible to declare a PARAM command whereby this command would result in an Operator Prompt being created and serviced. A facility to cater for PARAM handling is available within MultiBatch.

A TACL IN file as used in a MultiBatch step, could take the form as shown below:

```
COMMENT Delete your previous OUTVAR file
COMMENT It will be recreated by GETPRMPT
PURGE $LIVE.MBATDAT.OUTVAR
COMMENT Start the GETPROMPT program
RUN $LIVE.MBATOBJ.GETPRMPT/IN $LIVE.MBATDAT.DATACONF, OUT $LIVE.MBATDAT.OUTVAR/&
[#PARAM MBAT-STEPNAME], 30
FUP COPY $LIVE.MBATDAT.OUTVAR
OBEY $LIVE.MBATDAT.OUTVAR
#OUTPUT MultiBatch Job     = [#PARAM MBAT-STEPNAME]
#OUTPUT Date               = [todays^date]
#OUTPUT Start^key          = [start^key]
#OUTPUT End^key            = [end^key]
```

Note: *The value "30" above represents an Alarm value. This value is optional and signifies the length of time in seconds that the program waits for the prompts to be serviced before issuing an alert in EMS.*

Firstly, the BMON process on starting must include the parameter FORWARD^STEPNAME Y to enable this processing, as shown in the example below:

```
RUN BMON / NAME <BMON Name>, CPU 1, NOWAIT/ FORWARD^STEPNAME Y
```

This will result in the name of the MultiBatch step being forwarded to the step as the parameter MBAT-STEPNAME. The format of the step name will be <NODE>.$BMON.JJJJJ.SS.UU.X, where X is 1, 2 or *.

Secondly, the TACL IN file must include the following two lines, a call to a program called GETPRMPT, using the syntax as shown below:

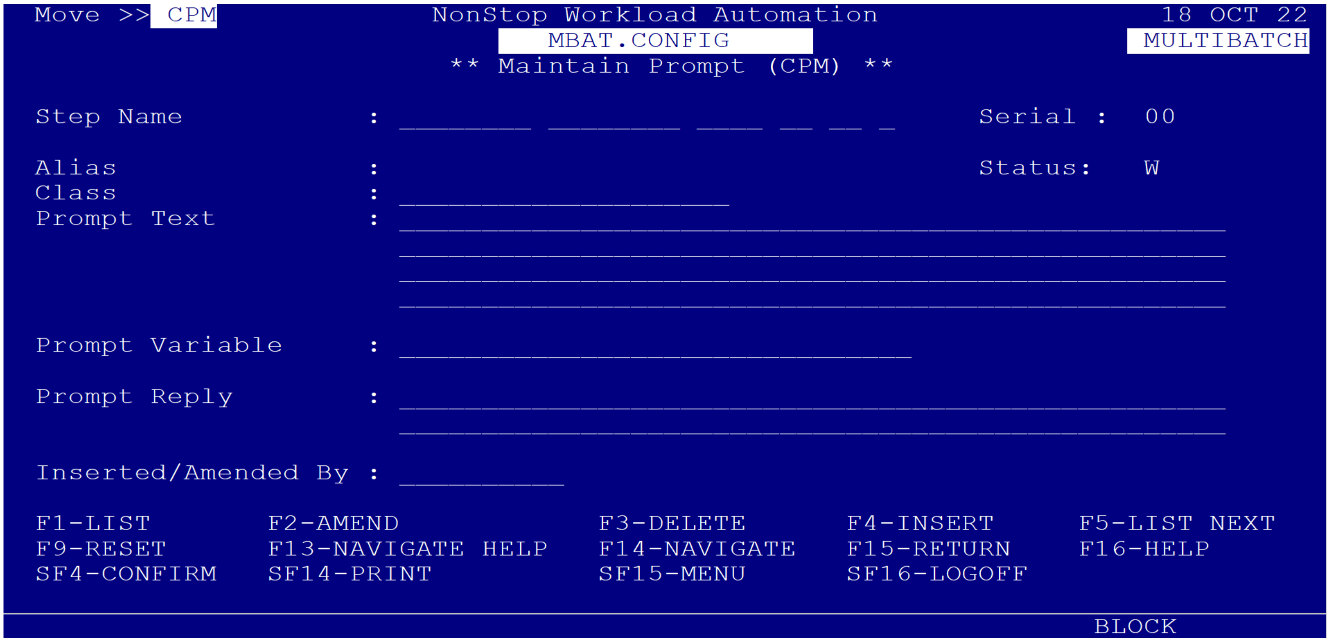
```
RUN MBATOBJ.GETPRMPT / IN MBATDAT.DATACONF , OUT MBATDAT.OUTVAR / &
[#PARAM MBAT-STEPNAME] , 30
```

Pathway screens exist which allow for the configuration of prompts and prompt classes.

A screen exists to allow for the monitoring of any prompts that are found to be in an "Outstanding (O)" state and allows the user to respond to and acknowledge the prompts to allow the Job to continue processing.

Configure PROMPT Screen

To configure a PROMPT, navigate to the MultiBatch Maintain Prompt screen by placing "**CPM**" in the Move box and press F14. Alternatively, you can select **F8** from the Entity Configuration Menu (CMM).



```
 Move >> CPM                 NonStop Workload Automation                  18 OCT 22
                                  MBAT.CONFIG                             MULTIBATCH
                           ** Maintain Prompt (CPM) **

 Step Name                : ________ ________ ____ __ __ _     Serial :  00

 Alias                    :                                   Status:   W
 Class                    : ____________________
 Prompt Text              : __________________________________________________
                            __________________________________________________
                            __________________________________________________
                            __________________________________________________

 Prompt Variable          : _______________________________

 Prompt Reply             : __________________________________________________
                            __________________________________________________

 Inserted/Amended By :      __________

 F1-LIST       F2-AMEND            F3-DELETE     F4-INSERT     F5-LIST NEXT
 F9-RESET      F13-NAVIGATE HELP   F14-NAVIGATE  F15-RETURN    F16-HELP
 SF4-CONFIRM   SF14-PRINT          SF15-MENU     SF16-LOGOFF

                                                                        BLOCK
```

Figure F.1 - Maintain Prompt (CPM) - 1

Complete the following fields:

Step Name consisting of: Node, BMON Process, Job, Segment, Unit.

Serial: Enter 01 if this is the first entry; increment the value for subsequent entries.

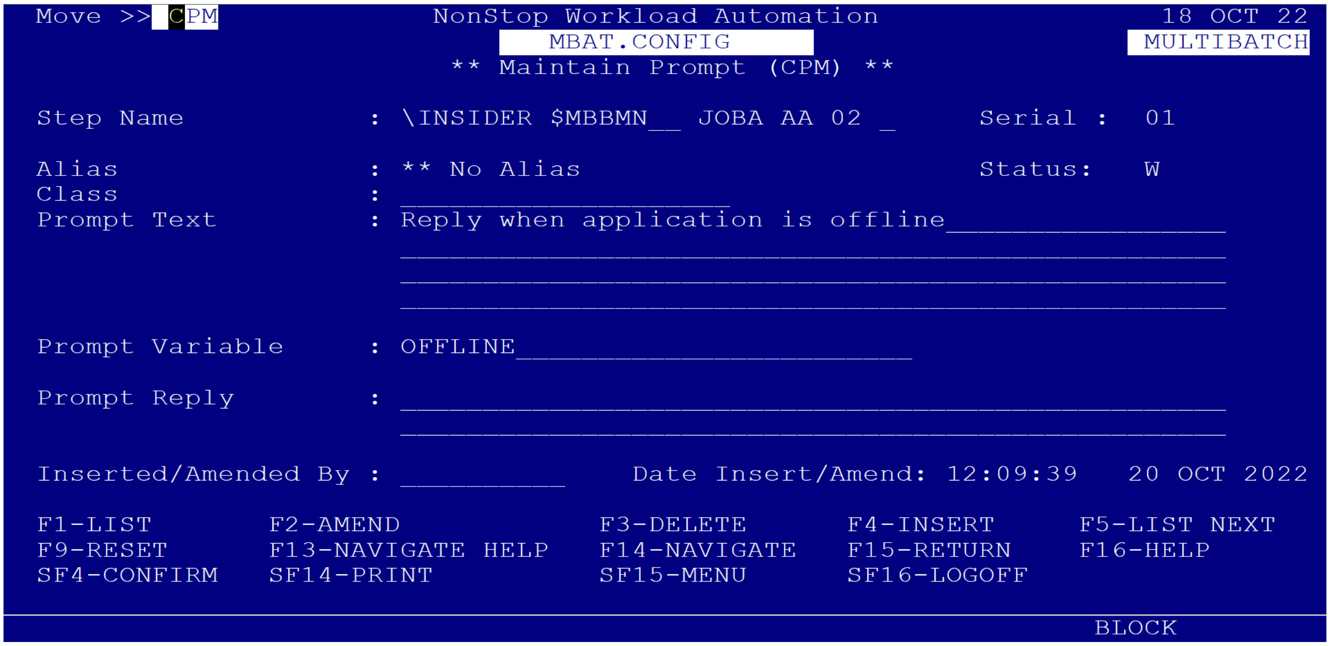
Alias Name: If the Unit/Step being inserted has an Alias name, then details will be supplied here on completion of the insert (F4).

Prompt Text: This is issued as part of an EMS Event, informing the user that a prompt requires attention. Prompt Variable: Enter the same value as stored within the INFILE obey file / macro.

Prompt Reply: Leave empty during the insert. This field is populated in response to the EMS prompt message.

Inserted / Amended by: Enter initials for the person inserting the prompt. Execute **F4** - Insert to complete entry.

Example of a configured prompt:



```
 Move >> CPM               NonStop Workload Automation                18 OCT 22
                               MBAT.CONFIG                           MULTIBATCH
                         ** Maintain Prompt (CPM) **

 Step Name            : \INSIDER $MBBMN__ JOBA AA 02 _     Serial :  01

 Alias                : ** No Alias                        Status:   W
 Class                : ____________________
 Prompt Text          : Reply when application is offline_________________
                        __________________________________________________
                        __________________________________________________
                        __________________________________________________

 Prompt Variable      : OFFLINE________________________

 Prompt Reply         : __________________________________________________
                        __________________________________________________

 Inserted/Amended By : __________    Date Insert/Amend: 12:09:39   20 OCT 2022

 F1-LIST        F2-AMEND            F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET       F13-NAVIGATE HELP   F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM    SF14-PRINT          SF15-MENU      SF16-LOGOFF

                                                                   BLOCK
```

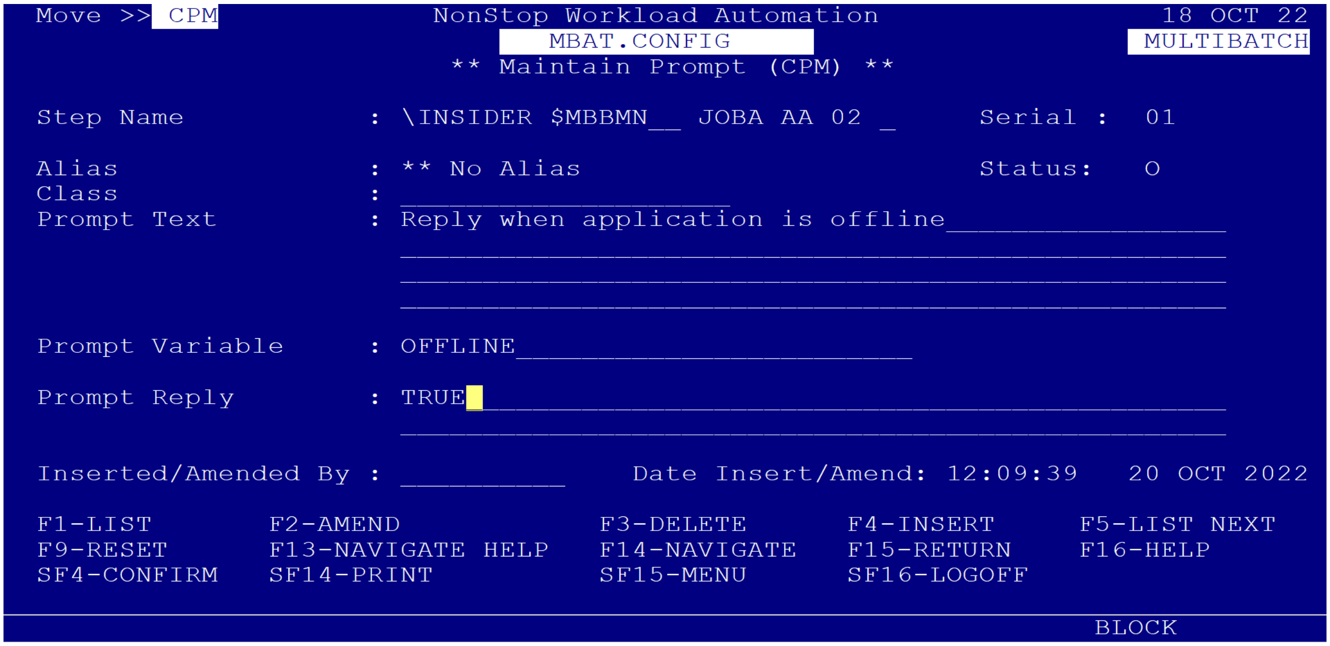
Figure F.2 - Maintain Prompt (CPM) - 2

Note: *Once inserted, the Status field changes to 'W' (Waiting).*

When the step's TACL IN file is run, the GETPRMPT program will update all the Prompt records by changing the step status from 'W' (Waiting) to 'O' (Outstanding).

An EMS event will also be issued, e.g.:

```
2022-10-19 16:38:14 \INSIDER.$:0:117:1938947672 MULTIBAT.1.0 200 Prompt
Outstanding for Step: \INSIDER.$MBBMN.JOBA.AA.02.*
```

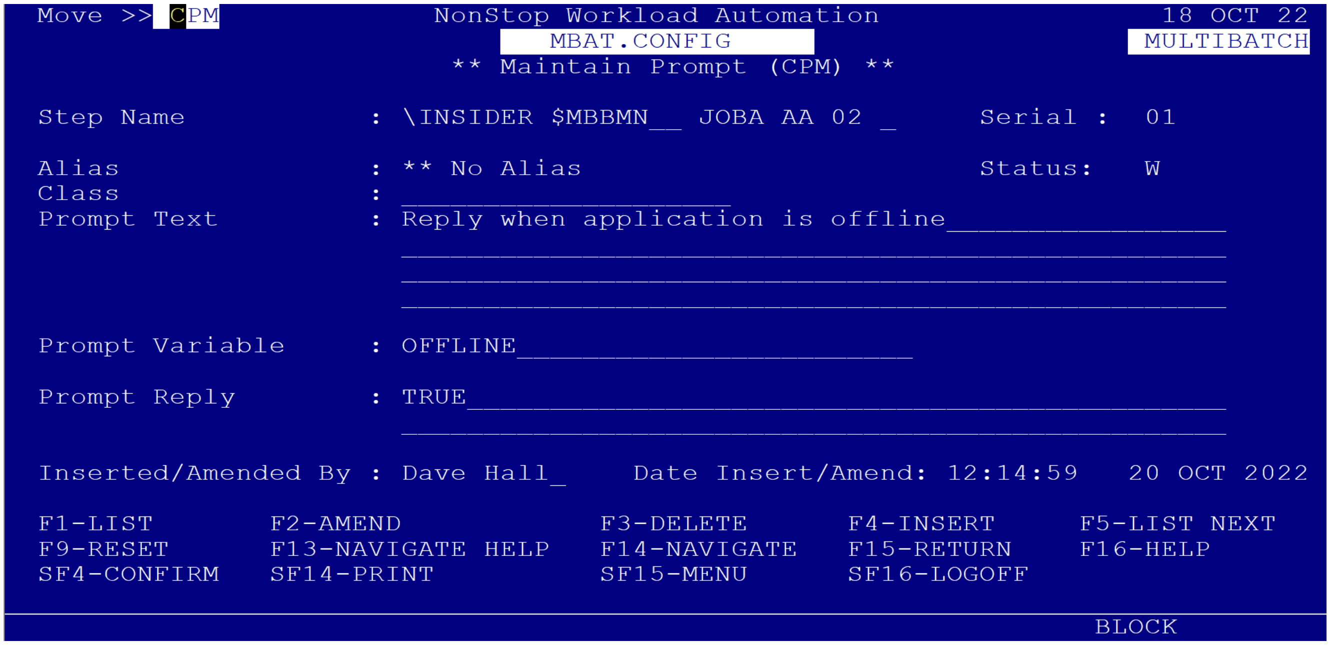


```
 Move >> CPM               NonStop Workload Automation                18 OCT 22
                               MBAT.CONFIG                           MULTIBATCH
                         ** Maintain Prompt (CPM) **

 Step Name            : \INSIDER $MBBMN__ JOBA AA 02 _     Serial :  01

 Alias                : ** No Alias                        Status:   O
 Class                : ____________________
 Prompt Text          : Reply when application is offline_________________
                        __________________________________________________
                        __________________________________________________
                        __________________________________________________

 Prompt Variable      : OFFLINE________________________

 Prompt Reply         : TRUE______________________________________________
                        __________________________________________________

 Inserted/Amended By : __________    Date Insert/Amend: 12:09:39   20 OCT 2022

 F1-LIST        F2-AMEND            F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET       F13-NAVIGATE HELP   F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM    SF14-PRINT          SF15-MENU      SF16-LOGOFF

                                                                   BLOCK
```

Figure F.3 - Maintain Prompt (CPM) - 3

The GETPRMPT program will wait for the user to answer all the prompts, using the MultiBatch Maintain Prompt screen (CPM).

Complete and review the associated step's prompt parameters before finally accepting the values by amending the prompt Status value from 'O' to a 'C' (Complete), then selecting the F2 key to amend, e.g.:



```
 Move >> CPM                  NonStop Workload Automation                18 OCT 22
                                  MBAT.CONFIG                          MULTIBATCH
                              ** Maintain Prompt (CPM) **

 Step Name             : \INSIDER $MBBMN__ JOBA AA 02 _      Serial :  01

 Alias                 : ** No Alias                         Status:   W
 Class                 : ____________________
 Prompt Text           : Reply when application is offline_________________
                         __________________________________________________
                         __________________________________________________
                         __________________________________________________

 Prompt Variable       : OFFLINE________________________

 Prompt Reply          : TRUE______________________________________________
                         __________________________________________________

 Inserted/Amended By : Dave Hall_    Date Insert/Amend: 12:14:59   20 OCT 2022

 F1-LIST       F2-AMEND          F3-DELETE      F4-INSERT      F5-LIST NEXT
 F9-RESET      F13-NAVIGATE HELP F14-NAVIGATE   F15-RETURN     F16-HELP
 SF4-CONFIRM   SF14-PRINT        SF15-MENU      SF16-LOGOFF

                                                                     BLOCK
```

Figure F.4 - Maintain Prompt (CPM) - 4

A status of C (Complete) will lead to the edit file specified in the GETPRMPT OUT option being created with a set of #PUSH and #SET values which can then be obeyed by the TACL IN file, e.g.:

```
#PUSH   OFFLINE
#SET    OFFLINE TRUE
OUTVAR  OFFLINE
```

If a prompt has not been responded to, or there are further prompts in an 'O' (Outstanding) state, then EMS events will be issued every 30 seconds, e.g.:

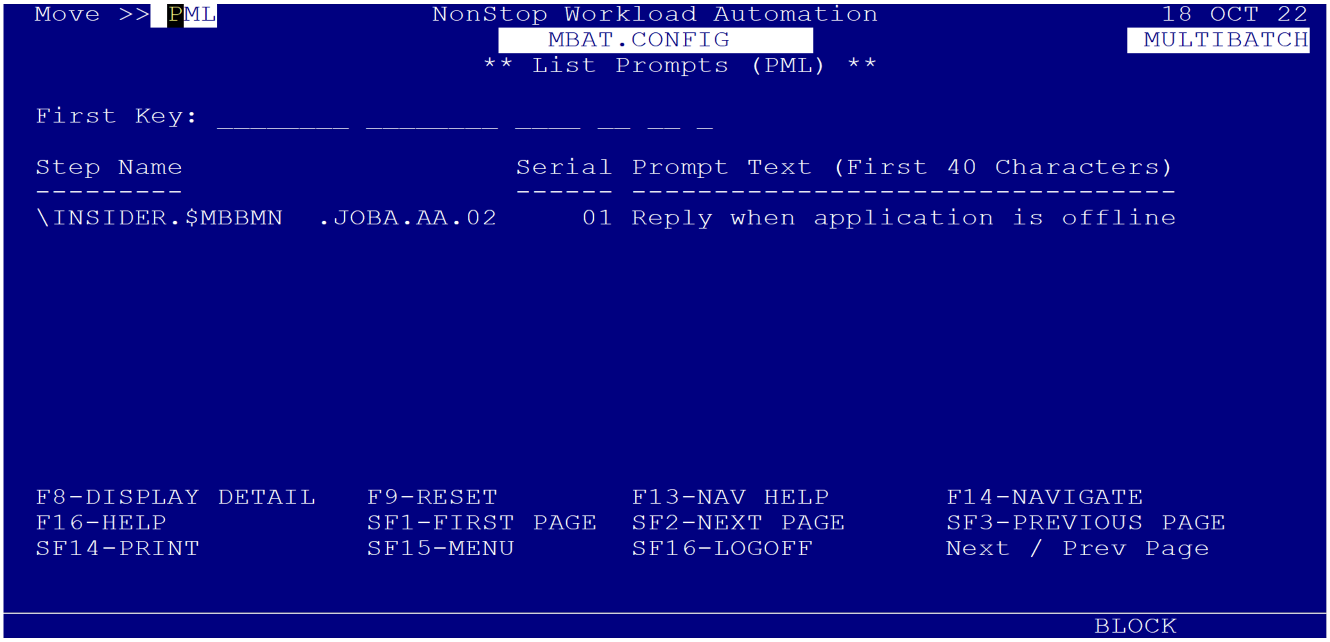
```
2022-10-20 12:42:07 \INSIDER.$:0:105:173804364 MULTIBAT.1.0 203 Alarm: Prompt
outstanding for 00030 seconds on Step: \INSIDER.$MBBMN.JOBA.AA.02
```

Alternatively, prompts can be rejected by placing an 'X' in the Status field and selecting the F2 key to amend. A status of X (Rejected) will lead to the job being abended. It can be resubmitted at a later point.

The MultiBatch Migrator utility exports and imports prompts to a remote database.

LIST PROMPT Screen

To configure a PROMPT, navigate to the MultiBatch: List Prompt screen by placing "**PML**" in the Move box and press F14. Alternatively, you can select **F8** from the Entity Configuration Menu (CMM).

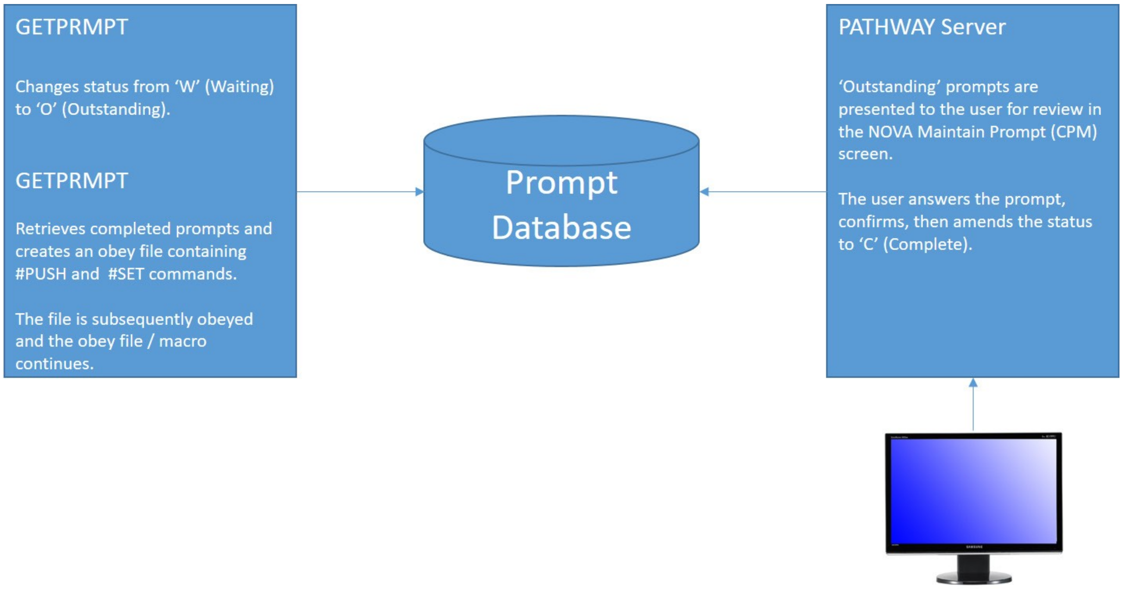


```
 Move >> PML             NonStop Workload Automation              18 OCT 22
                               MBAT.CONFIG                        MULTIBATCH
                         ** List Prompts (PML) **

 First Key: ________ ________ ____ __ __ _

 Step Name                     Serial Prompt Text (First 40 Characters)
 ---------                     ------ ----------------------------------------
 \INSIDER.$MBBMN  .JOBA.AA.02      01 Reply when application is offline

 F8-DISPLAY DETAIL   F9-RESET        F13-NAV HELP       F14-NAVIGATE
 F16-HELP            SF1-FIRST PAGE  SF2-NEXT PAGE      SF3-PREVIOUS PAGE
 SF14-PRINT          SF15-MENU       SF16-LOGOFF        Next / Prev Page

                                                                  BLOCK
```

Figure F.5 – List Prompts (PML)

To view the configuration of a prompt from this list, place a marker next to the prompt and select F8 - Detail.

GETPRMPT Architecture



GETPRMPT

Changes status from 'W' (Waiting) to 'O' (Outstanding).

GETPRMPT

Retrieves completed prompts and creates an obey file containing #PUSH and #SET commands.

The file is subsequently obeyed and the obey file / macro continues.

PATHWAY Server

'Outstanding' prompts are presented to the user for review in the NOVA Maintain Prompt (CPM) screen.

The user answers the prompt, confirms, then amends the status to 'C' (Complete).

PROMPT Classes

To configure a PROMPT, navigate to the MultiBatch: List Prompt screen by placing "**PCL**" in the Move box and press F14. Alternatively, you can select **SF9** from the Entity Configuration Menu(CMM).

To view the configuration of a prompt from this list, place a marker next to the prompt and select F8 - Detail.

F.8 MBATCOM

MBATCOM is a command line interface to allow a user to perform certain commands against the following MultiBatch NonStop processes:

- CALENDAR
- CLOCKMON
- STREAMER
- TRIGGER
- WATCHER

The commands available are as follows.

PROCESS RELATED COMMANDS

- BACKUPCPU <cpu>
- INFO mbatcom | INFO
- OPEN <process-name>
- SWITCH | <cpu>
- WARMBOOT

MISCELLANEOUS COMMANDS

- ALL
- COMMENT
- EXIT
- FC [line number | search string]
- HISTORY [number of lines]
- OBEY <file-name>

MBATCOM Process Commands

The MBATCOM Process Commands are used in communicating to the MultiBatch NonStop process.

BACKUPCPU <CPU>

This command instructs the NonStop process to use the specified CPU for its Backup Process.

INFO MBATCOM

This command displays the current setting.

```
LISTING   <on | off>
PAGING    <on | off>
PROCESS   <processname>
```

INFO

INFO returns various information from the NonStop Process such as Primary & Backup PIDs, Files which are open, Collectors which are being used.

The following is an example of typical information returned from a Streamer Process.

```
PROCESS NAME - $MSTR
OBJECT FILE - \LIVE.$MBAT.MBATOBJ.STREAMER
NODE NAME - \LIVE
PRIMARY CPU - 00
PRIMARY PIN - 0802
BACKUP CPU - 01
BACKUP PIN - 0614
CREATION TIMESTAMP - 2021:05:18 08:43:36:86:21
DATACONF FILE - \LIVE.$MBAT.MBATDAT.DATACONF
ALIASMAP FILE - \LIVE.$MBAT.MBATDAT.ALIASMAP
BCNF0998 FILE - \LIVE.$MBAT.MBATOBJ.BCNF0998
Non config FILE - none
Number of Alias Entries - 00000000
```

OPEN <$process-name>

This command opens the specified “process-name”. The purpose of this command is to switch from one MultiBatch NonStop process to another.

SWITCH

This command instructs the NonStop process to switch the primary and backup processes. If the Backup CPU is already the Backup for the process, then it will select the next highest available CPU.

WARMBOOT

This command instructs the NonStop process to restart any EMS distributor processes & to reread database files.

NCOM Misc Commands

COMMENT <text>

Any comment line will be ignored by NCOM, merely serving to document a command file.

EXIT (or <E>)

When this command is executed, NCOM interface stops running.

FC

This is the standard NonStop implementation of the “FC” command.

HELP <command>

For example:

HELP BACKUPCPU would reveal BACKUPCPU <cpu>

HISTORY <no lines>

F.9 UTPARHK

The UTPARHK Utility allows obsolete LOCAL/GLOBAL Conditional parameters to be reported / deleted from the MultiBatch configuration database.

This program has been built to read through the GLOBAL and LOCAL conditional parameter files and identify those conditional parameters which are not used within these files:

- BMONCONF (BMON configuration including log files and recover filenames),
- STEPCONF (STEP configuration including in, out, hometerm, library, swap & object filenames, defaults, process name, CPU specification)
- STEPASS (STEP ASSIGN configuration)
- STEPPAR (STEP PARAMETER configuration)
- STEPOSS (OSS STEP configuration)
- STEPOSE (OSS STEP Environment)
- STEPOSF (OSS STEP FDs)
- ASSCLASS (Assign Classes)
- PARCLASS (Parameter Classes)
- DEFCONF (Define configuration)
- DEFCLASS (Define Class configuration)
- STEPOSEC (OSS Environment Class)
- STEPOSFC (OSS FD Classes)

This program reports on and optionally deletes the obsolete conditional parameters.

The program syntax is:

```
MultiBatch LOCAL/GLOBAL Params Housekeeping
-------------------------------------------
UTPARHK/ IN <dataconf-file>, OUT <list-file>/<U or P>
IN      - Location of the DATACONF file
OUT     - Listing file
Options - <U or P>:
          UPDATE  = report and delete unused parameters
          PREVIEW = report but don't delete

e.g.  RUN UTPARHK/IN DATACONF,OUT $S.#LIST/PREVIEW

This program processes assign, define, param, env or fd classes for each
BMON shell found. This may produce errors that, on examination, can be
ignored.
You are advised to initially use the PREVIEW option to validate the
parameters that will be removed before using the UPDATE option.
```

F.10 UTPCALHK

A user can delete or archive information from the Calendar file (PARAMCAL)

The file holds details of MultiBatch calendar entries; e.g. BANKHOL, LASTSUN and can be used as part of the Include / Exclude scheduling functionality.

To list out the syntax, enter RUN UTPCALHK.

```
MultiBatch PARAMCAL Housekeeping
--------------------------------

  UTPCALHK/ IN <dataconf-file>, OUT <list-file>/ Options

  IN      - Location of the DATACONF file
  OUT     - Listing file

  Options - A,B,UPDATE/PREVIEW, dateA,dateB
            A: Y/N - Delete Old Dates, default N
            B: Y/N - Calendar usage report, default N

            UPDATE/PREVIEW: Update = delete
                            Preview = report and don't delete
                            Default = PREVIEW

            dateA: dd/mm/yyyy - Delete records before this date
                            Default = before today
            dateB: dd/mm/yyyy - Usage reported after this date
                            Default = after today
```

The second option will check that a JOB, SEGMENT or UNIT record that is using calendars, has at least one future date configured beyond ‘dateB’. This is an attempt to prevent jobs not being scheduled because they have run out of Calendar entries.

Example:

```
$MBAT.MBATOBJ 1> RUN UTPCALHK /IN MBATDAT.DATACONF / Y, Y, PREVIEW

PARAMCAL Obsolete Date report:   Calendars before 04/05/2021
                                 No Obsolete Dates found.


CALENDAR Usage report:   Dates after 04/05/2021
-------------------------------------------------------------------
<NODE> $BMON   JOBB      J-BMON-JOBB

                         CALENDAR No Future Dates for this calendar
```

F.11 UTRECOV

The UTRECOV utility is used to report on a BMON status using its Recovery file. Jobs / segments / units are listed in a variety of reports.

For example, if COMPLETED is specified, then all the jobs / segments / units which are completed are listed in the report.

If FAILED is specified, then all the jobs / segments / units which have failed are listed in the report.

If NOT STARTED is specified, then all the jobs / segments / steps which have not yet started are listed.

The DETAIL report lists all jobs / segments and steps.

If "LAST COMPLETED, 10" is specified then those jobs / segs / units which completed within the last 10 minutes will be listed with the most recent listed first.

UTRECOV can be useful following a site-swap for example, where a BMON has been recovered using a replicated Recovery file and you wish to obtain a listing and a summary of the various states the schedule has reached, e.g. list all failed jobs.

UTRECOV syntax:

```
MultiBatch BMON RECOVER Reporting Tool
--------------------------------------

UTRECOV/ IN <recover-file>, OUT <list-file>/ Options

      IN      - Location of the RECOVER file
      OUT     - Listing file
      Options - COMPLETED
                FAILED DETAIL
                NOT STARTED
                LAST COMPLETED[, nnn] where nnn is number of minutes
                ALL[, nnn] where nnn is number of minutes

 e.g. RUN UTRECOV/IN RECOVER,OUT $S.#LIST/ALL, 10
```

F.12 UTUSERHK

The UTUSERHK Utility allows obsolete Guardian User Ids to be reported / deleted from the MultiBatch security database.

This program has been built to review the MultiBatch security database and optionally report on, or delete obsolete GUARDIAN user ids from the MultiBatch database.

The program will review the following four files:

- BMONCONF - The name and configuration details of a BMON record. This includes the owner and privileged user id of the BMON, held in integer format.
- STEPCONF - The user id that a step should execute as. This is held in integer format and is only active if the CST or COS ‘use user id’ field is true.
- SYSOWNER - A MultiBatch security system file that holds users can have access to PATHWAY facilities but only for certain BMONs.
- SECPROFL - A MultiBatch security profile provides a correlation between a security class and a user. This needs to be in place for the user to logon.

The program syntax is:

```
RUN UTUSERHK/IN <DATACONF-LOCATION>,OUT <LISTING-FILE>/A,B,C,D, UPDATE | PREVIEW
```

The default <LISTING-FILE> file is the home terminal.

```
      A:       Y/N – Review BMONCONF, default N
      B:       Y/N – Review STEPCONF, default N
      C:       Y/N – Review SYSOWNER, default N
      D:       Y/N – Review SECPROFL, default N

      UPDATE | PREVIEW:     Update = delete, PREVIEW report and don’t delete.
      Default=PREVIEW
```

The basic principle is that the utility will never delete any records from BMONCONF or STEPCONF as this would prevent a BMON from completing a PREPARE/BUILD.

Manual changes are required in PATHWAY screens CBM and CST to correct issues highlighted in the report.

Obsolete users discovered on the SYSOWNER and SECPROFL files will result in the associated record being deleted in UPDATE mode.

Syntax errors will be reported to the home terminal.

After running the utility remember to also manually amend entries reported via the CBM, CST and COS screens if required.

F.13 MIGRCONP

On occasions, a user may install and create a new MultiBatch environment consisting of database, object subvolumes, along with the associated Pathway and MultiBatch processes.

The user can then configure BMONs from the beginning but if this is too onerous, for example, a user wants to create a BMON with several hundred jobs, segments and units, then there is the option to use the MIGRATOR facility.

However, MIGRATOR does not migrate any associated Conditional Parameters as stored in the Local and Global Enscribe database files. Therefore, to help a user when Migrating BMONs from a current MultiBatch environment into this new database the user can utilise the MIGRCONP utility.

To use MIGRCONP, a Help syntax is provided:

```
MultiBatch LOCAL/GLOBAL Migrator
--------------------------------

RUN MIGRCONP /OUT <list-file>/

e.g.

RUN MIGRCONP /OUT $S.#LIST/

Before running this migrator program:

  1  Ensure new empty LOCAL and GLOBAL files have been created

  2  Create the following 5 file assigns:

'NEW-LOCAL'  - This should point to the new empty LOCAL file

'OLD-LOCAL'  - This should point to the LOCAL file to be migrated

'NEW-GLOBAL'- This should point to the new empty GLOBAL file

'OLD-GLOBAL'- This should point to the GLOBAL file to be migrated

'BMONDEFS'   - This should point to the migration definitions edit file.
```

As stated, ensure these five assigns are specified before running MIGRCONP, then simply RUN MIGRCONP.

This utility will extract the Conditional Parameters from the old Local and Global files, into the new Local and Global files.

The BMONDEFS (located in the MultiBatch Object subvolume, e.g. MBATOBJ) file needs to be amended to reference the old node and the new node values, e.g.:

```
      NODE      \LIVE     \BACKUP
      VOLUME    $SYSTEM   $SYSTEM
      SUBVOL    MBATOBJ   MBATOBJ2
      SUBVOL    SYS01     SYS02
      DIR       bin       bin2
```

The old values all start in column 9. The new values all start in column 17.

F.14 PROGRAMMATIC INTERFACES

The MultiBatch product is shipped with two libraries (ETIMLIB and LIBUTIL), enabling user applications to communicate with Event Timer and BMON respectively.

The libraries can be utilised by user application code written in the language of your choice: COBOL, TAL or C/C++. Examples are provided on how to integrate the libraries with COBOL.

Code 100 and natively compiled 500 and 800 library files are available, e.g.:

```
             CODE                EOF   LAST MODIFIED  OWNER  RWEP     PExt    SExt
ETIMLIB        100              90978 23JUL2021 10:41   1,11  NUNU      14      14
ETIMLIB5       500              75352 23JUL2021 11:51   1,11  NUNU      28      28
ETIMLIB8       800             113712 23JUL2021 10:41   1,11  NUNU      42      42

LIBUTIL        100              27490 23JUL2021 10:42   1,11  NUNU      14      14
LIBUTIL5       500              41896 23JUL2021 11:52   1,11  NUNU      28      28
LIBUTIL8       800              68176 23JUL2021 10:42   1,11  NUNU      42      42
```

Descriptions on how to integrate both libraries with your application are provided in their respective sections.

Note: *The LIBUTIL definition is different between code 100 and native builds, i.e., code 500 and 800; code 500 and code 800 use the same definition.*

Note: *BMON, the NonStop process which initiates, monitors and logs the progress of your job workload has a number of interfaces. As with most HPE monitor processes this interface uses the message based $RECEIVE mechanism to pass Inter Process Messages (IPMs) from a command interface. The command interfaces to BMON are from:*

1) *BCOM – the Command Line Interface*

2) *TRIGGER – when processing jobs, segments and units set up in the Event Timer scheduler*

3) *LIBUTIL – when using the programmatic interface to HOLD, RELEASE, START, RESTART, ABORT, RESTART, IF FAILED Jobs, Segments and Units*

4) *STATSRV – provides BMON status details for the MultiBatch Screen COBOL Interfaces.*

When these command interfaces communicate with BMON an initial message is sent to ensure that further messages will be compatible with the BMON version.

BMON returns status 409 when the version of messages used by a command interface do not match the BMON version.

The BMON interface is only updated for a major release, in this situation all of BCOM, TRIGGER, LIBUTIL and STATSRV are also part of the release. In this way the 409 status shows there is a configuration issue.

Status 409 is commonly generated where customer programs interfacing with BMON using LIBUTIL need to be modified to include the latest delivered LIBUTIL procedures.

ETIMLIB

The ETIMLIB programmatic interface module is designed to be bound into user application code and allows the user to manipulate the status and next run time and date of an Event Timer **'AT'** Schedule record.

Important: *ETIMLIB is only applicable for '**AT**' entries. CRONTAB and EVERY schedule manipulation by ETIMLIB is currently unavailable.*

The original **AT** schedule record must be created through the Event Timer Pathway system. The ETIMLIB programmatic interface will then allow:

1) Checking for the existence of a schedule record.

2) Retrieval of the status and next run date of a schedule record.

3) Forcing the immediate run of a job.

4) Alteration of the next run of a job.

5) Override for "held" and "expired" jobs.

6) Automatic "hold after" capability on the completion of a successful run of a job.

LIBRARY CALLS

For all ETIMLIB objects, ensure that the ETIMLIB values are referenced in the COBOL application program SPECIAL NAMES section, e.g.:

```
SPECIAL-NAMES
   FILE "=ETIMLIB" is ETIMLIB
```

Note: *The =ETIMLIB define must reference the correct library object code, e.g.*

```
ADD DEFINE =ETIMLIB, FILE MBATOBJ.ETIMLIB.
```

The interface will take the form of a TAL library, named ETIMLIB, which can be bound into a program and can be executed thus, from COBOL:

```
ENTER TAL "update^DS^EVENTTIMER^schedule" OF ETIMLIB
USING UPDATE-SCHEDULE-RECORD.
```

ETIMLIB - COBOL Layout

The layout below, can be found within the PINTCOPY source file, located in the MultiBatch Object subvolume, e.g. MBATOBJ.

```
?SECTION UPDATE-SCHEDULE-RECORD

   03 USR-SWITCH-REPLY           PIC S9(4) COMP.   (1)
   03 USR-SCHEDULE-KEY.                            (2)
      05 USR-BMON-ENTITIES.
         07 USR-BMON-JOB         PIC X(4).
         07 USR-BMON-SEGMENT     PIC XX.
         07 USR-BMON-UNIT        PIC 99.
      05 USR-BMON-NODE           PIC X(8).
      05 USR-BMON-PROCESS        PIC X(8).
      05 USR-SERIAL              PIC 99.
   03 USR-SCHEDULE-FILENAME      PIC X(34).        (3)
   03 USR-CALENDAR-PROCESS       PIC X(16).        (4)
   03 USR-REPLY-TEXT             PIC X(80).
   03 USR-DATE.
      05 USR-YEAR                PIC S9(4) COMP.
      05 USR-MONTH               PIC S9(4) COMP.
      05 USR-DAY                 PIC S9(4) COMP.
```

```
03 USR-TIME.
   05 USR-HOUR              PIC S9(4) COMP.
   05 USR-MINUTE            PIC S9(4) COMP.
   05 USR-SECOND            PIC S9(4) COMP.
03 USR-AMEND-STATUS         PIC X.
03 USR-RUN-WHEN-HELD        PIC X.           (5)
03 USR-RUN-WHEN-EXPIRED     PIC X.           (6)
03 USR-HOLD-AFTER-RUN       PIC X.           (7)
```

NOTE	FIELD	DESCRIPTION
(1)	USR-SWITCH-REPLY: **On-Call**	0: Read record and return next date in USR-TIME-DATE and status in USR-AMEND-STATUS. 1: Update schedule with the values in USR-DATE and USR-TIME. 2: Force the running of the job. 3: Amend the status of the record with the value in USR-AMEND-STATUS.
	USR-SWITCH-REPLY: **On-Reply**	0: Success. 100: Invalid switch supplied. 101: File error: nnnn : <file name> 102: Calendar process not specified 103: Amend status value must be S or H 104: Run when held value must be Y or N 105: Run when expired value must be Y or N 106: Hold after run value must be Y or N 107: Invalid day/month/year 108: Invalid hour/minute/second 109: Time and date provided is <current time and date> 110: Schedule record does not exist 111: TMF error: nnnn 112: Schedule record has expired 113: Schedule record is held 114: Schedule status is not S or H 115: File error: nnnn : <calendar name> 116: Schedule not **AT** type
(2)	USR-SCHEDULE-KEY	Key of SCHEDULE record. Unused fields set to spaces and zeros as appropriate.
(3)	USR-SCHEDULE-FILENAME	Name of SCHEDULE file, which will be expanded using system, volume and subvolume names of the object code that this routine is bound into. **NOTE**: It is advisable to provide the fully qualified file name.
(4)	USR-CALENDAR-PROCESS	Name of Calendar Process. This process needs to be notified after the schedule changes so that it can calculate whether a job now needs to run.
(5)	USR-RUN-WHEN-HELD	If the job is currently held, should it run? Y or N.
(6)	USR-RUN-WHEN-EXPIRED	If the SCHEDULE record has expired, should it run? Y or N.
(7)	USR-HOLD-AFTER-RUN	Once the job has run, and the next run date is calculated, should the job be held automatically? Y or N.

Example COBOL source code is provided below.

Compiler directives are used to differentiate between builds for different platforms.

The example COBOL source code below is contained within the file called PINTEG, located in the MultiBatch Object subvolume, e.g. MBATOBJ.

```
?SYMBOLS, INSPECT

 IDENTIFICATION DIVISION.
 PROGRAM-ID.     UPDATE-AT-SCHEDULE.
 AUTHOR.         MULTIBATCH DEVELOPMENT.
 INSTALLATION.   MULTIBATCH DEV ENV.
 DATE-WRITTEN.   MAY-2021.

 ENVIRONMENT DIVISION.

  CONFIGURATION SECTION.
     SOURCE-COMPUTER.           TANDEM/16.
     OBJECT-COMPUTER.           TANDEM/16.

  SPECIAL-NAMES.
?IF 1
* Standard TNS (code 100) build
     FILE "=ETIMLIB" IS ETIMLIB.
?ENDIF 1

?IF 5
* Standard TNS/X (code 500) build
     FILE "=ETIMLIB5" IS ETIMLIB.
?ENDIF 5

?IF 7
* Standard TNS/R (code 700) build
     FILE "=ETIMLIB7" IS ETIMLIB.
?ENDIF 7

?IF 8
* Standard TNS/E (code 800) build
     FILE "=ETIMLIB8" IS ETIMLIB.
?ENDIF 8

 DATA DIVISION.

 WORKING-STORAGE SECTION.

 01 UPDATE-SCHEDULE-RECORD.
    COPY UPDATE-SCHEDULE-RECORD OF "PINTCOPY".

 PROCEDURE DIVISION.

****************************
 000-CONTROL SECTION.
 000-001.
****************************

     MOVE 2                            TO USR-SWITCH-REPLY
     MOVE "JJJJSSUU\NODE   $BMON   01" TO USR-SCHEDULE-KEY.
     MOVE "$VOLUME.SUBVOL.SCHEDULE   " TO USR-SCHEDULE-FILENAME.
     MOVE "                          " TO USR-CALENDAR-PROCESS.
     MOVE SPACES                       TO USR-REPLY-TEXT.
     MOVE 2021                         TO USR-YEAR.
     MOVE 07                           TO USR-MONTH.
     MOVE 07                           TO USR-DAY.
     MOVE 10                           TO USR-HOUR.
     MOVE 10                           TO USR-MINUTE.
     MOVE 10                           TO USR-SECOND.
     MOVE "N"                          TO USR-RUN-WHEN-HELD
                                          USR-RUN-WHEN-EXPIRED
                                          USR-HOLD-AFTER-RUN.

   ENTER TAL "update^DS^EVENTTIMER^schedule" OF ETIMLIB
         USING UPDATE-SCHEDULE-RECORD.


 000-EXIT.
     STOP RUN.
```

ETIMLIB - Code 100 PINTEG Compile

When attempting to compile the supplied example source code - PINTEG - ensure the following is applied:

- DEFINE = ETIMLIB has been configured to reference the correct ETIMLIB library
- Compilation is performed in the same subvolume as the PINTCOPY file, i.e. the MultiBatch Object subvolume
- The COBOL compilation command specifies the use of toggle 1, e.g.:

```
COBOL85/IN PINTEG, OUT $S.#PIN/PINTEGO ; SETTOG 1
```

ETIMLIB5 - Code 500 PINTEG Compile

When attempting to compile the supplied example source code - PINTEG - ensure the following is applied:

- DEFINE = ETIMLIB5 has been configured to reference the correct ETIMLIB5 library
- Compilation is performed in the same subvolume as the PINTCOPY file, i.e. the MultiBatch Object subvolume
- The XCOBOL compilation command specifies the use of toggle 5, e.g.

```
XCOBOL/IN PINTEG, OUT $S.#PIN/PINTEGO ; RUNNABLE, SETTOG 5
```

ETIMLIB8 - Code 800 PINTEG Compile

When attempting to compile the supplied example source code - PINTEG - ensure the following is applied:

- DEFINE = ETIMLIB8 has been configured to reference the correct ETIMLIB8 library
- Compilation is performed in the same subvolume as the PINTCOPY file, i.e. the MultiBatch Object subvolume
- The ECOBOL compilation command specifies the use of toggle 8, e.g.

```
ECOBOL/IN PINTEG, OUT $S.#PIN/PINTEGO ; RUNNABLE, SETTOG 8
```

ETIMLIB5 - Code 500 Integration

The code 500 ETIMLIB Library file is shipped as ETIMLIB5.

Make the code or define changes as required to pick up this ETIMLIB5 object, or alternatively rename the

To integrate the code 500 library file, load the defines that will allow any MAP references to be satisfied. Compile the application code; for example, COBOL application USERAPP:

```
XCOBOL/IN USERAPP, OUT $S.#UA/USERAPPO
```

Link the compiled objects using XLD to create the final object:

```
XLD/IN UALINK, OUT $S.#UAL/
```

Where an example XLD script file, UALINK, would be:

```
--------------------------------------------
---        HP Native COBOL Link file      ---
---              mmm yyyy                 ---
--------------------------------------------
-----------------------------------------------------------
---      List of object files to link      ---
-----------------------------------------------------------
=USERLIB1
-----------------------------------------------------------
--- User libraries to link into object ---
-----------------------------------------------------------
=VPROCLIBRARY5
-----------------------------------------------------------
--- System libraries to link into object ---
-----------------------------------------------------------
-lZCOBDLL
-lZCREDLL
-----------------------------------------------------------
---            Linker Options             ---
-----------------------------------------------------------
-ALLOW_DUPLICATE_PROCS
-b dynamic
-verbose
-----------------------------------------------------------
---           Final object file          ---
-----------------------------------------------------------
-o =BINARYFILE
```

Which uses these defines:

=USERLIB1 is the XCOBOL compiled object code: USERAPPO.

=VPROCLIBRARY5 is any other (optional) library such as VPROC library.

=BINARYFILE is the output from the XLD session, e.g. USERAPPL: this is the code that you execute.

ETIMLIB8 - Code 800 Integration

The code 800 ETIMLIB Library file is shipped as ETIMLIB8.

Make the code or define changes as required to pick up this ETIMLIB8 object, or alternatively rename the file to ETIMLIB; if older ETIMLIB objects exist, then also rename as required.

To integrate the code 800 library file, load the defines that will allow any MAP references to be satisfied. Compile the application code; for example COBOL application USERAPP:

```
ECOBOL/IN USERAPP, OUT $S.#UA/USERAPPO
```

Link the compiled objects using ELD to create the final object:

```
ELD/IN UALINK, OUT $S.#UAL/
```

Where an example ELD script file, UALINK, would be:

```
--------------------------------------------
---       HP Native COBOL Link file      ---
---             mmm yyyy                 ---
--------------------------------------------
--------------------------------------------
---     List of object files to link     ---
--------------------------------------------
=USERLIB1
--------------------------------------------
--- User libraries to link into object ---
--------------------------------------------
=VPROCLIBRARY8
--------------------------------------------
--- System libraries to link into object ---
--------------------------------------------
-lZCOBDLL
-lZCREDLL
--------------------------------------------
---            Linker Options            ---
--------------------------------------------
-ALLOW_DUPLICATE_PROCS
-b dynamic
-verbose
--------------------------------------------
---           Final object file         ---
--------------------------------------------
-o =BINARYFILE
```

Which uses these defines:

=USERLIB1 is the ECOBOL compiled object code: USERAPPO

=VPROCLIBRARY8 is any other (optional) library such as VPROC library.

=BINARYFILE is the output from the ELD session, e.g. USERAPPL: this is the code that you execute.

LIBUTIL

The LIBUTIL programmatic interface module is designed to be bound into user application code and allows the application to communicate directly with the MultiBatch scheduler, BMON.

It supports the following commands through the use of high level function calls:

1) HOLD (Job, Segment or Unit)
2) RELEASE (Job, Segment or Unit)
3) START (Job, Segment or Unit)
4) RESTART (Unit only)
5) ABORT (Unit only)
6) RESTART, IF FAILED (Unit only)

Note: *These LIBUTIL commands are not subject to security checking as this functionality should be embedded in the application making the LIBUTIL call.*

For all LIBUTIL objects, ensure that the LIBUTIL values are referenced in the COBOL application program SPECIAL NAMES section, e.g.:

```
SPECIAL-NAMES
   FILE “=LIBUTIL” is LIBUTIL
```

Note: *The =LIBUTIL define must reference the correct library object code, e.g.*

```
ADD DEFINE =LIBUTIL, FILE MBATOBJ.LIBUTIL
```

The interface will take the form of a TAL library, named LIBUTIL, which can be bound into a program and can be executed from COBOL; example calls:

Validating a BMON Entry:

```
ENTER TAL “mb^val^entity” OF LIBUTIL
    USING ws-bmon    OF linkage-data-1,
          ws-entity  OF linkage-data-1,
          ws-dataconf,
          ws-alias
          ws-reply   OF linkage-data-1
   GIVING reply-code OF linkage-data-1
```

Requesting BMON to Execute a Command:

```
ENTER TAL “mb^do^command” OF LIBUTIL
    USING ws-bmon    OF linkage-data-2
          ws-entity  OF linkage-data-2
          ws-command OF linkage-data-2
          ws-reply   OF linkage-data-2
   GIVING reply-code OF linkage-data-2
```

COBOL Layout

The layout below can be found within the HARNCOPY source file, located in the Object subvolume.

Note: *There are minor differences in the layouts used for TNS (code 100) and native (code 500 and 800) builds.*

```
?SECTION LINKAGE1-TNS

    03 ws-bmon                             PIC X(17).          (1)
    03 ws-entity                           PIC X(10).          (2)
    03 ws-alias                            PIC X(30).          (3)
    03 ws-reply.                                                 (5)
      05 reply-code                         PIC S9(4) COMP.
      05 reply-bmon                         PIC X(17).
      05 reply-entity                       PIC X(10).
      05 reply-alias-name                   PIC X(30).
      05 reply-first-unit                   PIC X(10).
      05 reply-fname                        PIC X(35).
      05 reply-fio-err                      PIC S9(4) COMP.
      05 reply-text                         PIC X(80).

?SECTION LINKAGE2-TNS

    03 ws-bmon                              PIC X(17).           (1)
    03 ws-entity                            PIC X(10).           (2)
    03 ws-command                           PIC 9(4) COMP.       (3)
    03 ws-reply.                                                 (4)
      05 reply-code                         PIC S9(4) COMP.
      05 reply-bmon                         PIC X(17).
      05 reply-entity                       PIC X(10).
      05 reply-fname                        PIC X(35).
      05 reply-fio-err                      PIC S9(4) COMP.
      05 reply-text                         PIC X(80).

?SECTION LINKAGE1-NATIVE

    03 ws-bmon                              PIC X(18).           (1)
    03 ws-entity                            PIC X(10).           (2)
    03 ws-alias                             PIC X(30).           (3)
    03 ws-dataconf                          PIC X(26).           (4)
    03 ws-reply.                                                 (5)
      05 reply-code                         PIC S9(4) COMP.
      05 reply-bmon                         PIC X(18).
      05 reply-entity                       PIC X(10).
      05 reply-alias-name                   PIC X(30).
      05 reply-first-unit                   PIC X(10).
      05 reply-fname                        PIC X(36).
      05 reply-fio-err                      PIC S9(4) COMP.
      05 reply-text                         PIC X(80).

?SECTION LINKAGE2-NATIVE

    03 ws-bmon                              PIC X(18).           (1)
    03 ws-entity                            PIC X(10).           (2)
    03 ws-command                           PIC 9(4) COMP.       (3)
    03 ws-reply.                                                 (4)
      05 reply-code                         PIC S9(4) COMP.
      05 reply-bmon                         PIC X(18).
      05 reply-entity                       PIC X(10).
      05 reply-fname                        PIC X(36).
      05 reply-fio-err                      PIC S9(4) COMP.
      05 reply-text                         PIC X(80).
```

The layouts of LINKAGE1 and LINKAGE2, plus notes (n) on the selected fields are provided on the next two pages.

LINKAGE1 as used with:

ENTER TAL “mb^val^entity”

NOTE	FIELD	DESCRIPTION
(1)	ws-bmon	Optional fully qualified BMON process name, e.g. <NODE>.$BMON
(2)	ws-entity	Optional MultiBatch Entity (Job, Segment, Unit), e.g. JOBA.AA.01
(3)	ws-alias	Optional MultiBatch Job Alias
(4)	ws-dataconf	Optional MultiBatch DATACONF file, e.g. $VOL.MBATDAT.DATACONF

One of the following combinations is required:

- BMON Process Name and Entity (Job, Segment, or Unit)
- BMON Process Name, Job Alias and DATACONF file
- Job Alias and DATACONF file: the ALIASMAP file is used to obtain the BMON and Entity name details

All alphabetic fields should be space padded and numeric fields should be null (0) padded.

NOTE	FIELD	DESCRIPTION
(5)	ws-reply	**Reply-FIO-Err 0 : Validation Successful**
		Reply-Code 0 : Success, BMON OK
		Reply-Code 1 : Success, JOB OK
		Reply-Code 2 : Success, SEGMENT OK
		Reply-Code 3 : Success, UNIT OK
(5)	ws-reply	**Reply-FIO-Err -1 : Validation Error**
		Attempting to use BMON:
		Reply-Code 9001 : BMON Name Invalid
		Reply-Code 9003 : BMON Cannot be Opened
		Reply-Code 9006 : BMON Cannot be Written to
		Attempting to use ALIASMAP:
		Reply-Code 9000 : Alias supplied Without DATACONF
		Reply-Code 9001 : DATACONF invalid / ALIASMAP Name invalid
		Reply-Code 9002 : DATACONF / ALIASMAP cannot be opened
		Reply-Code 9003 : Cannot position on DATACONF / ALIASMAP record
		Reply-Code 9004 : DATACONF / ALIASMAP cannot be read
		Reply-Code 9005 : Alias is for a BMON

The Reply-Text will have more detail about the cause of the error.

LINKAGE2 as used with:

ENTER TAL "mb^do^command"

NOTE	FIELD	DESCRIPTION
(1)	ws-bmon	Optional fully qualified BMON process name, e.g. <NODE>.$BMON
(2)	ws-entity	Optional MultiBatch Entity (Job, Segment, Unit), e.g. JOBA.AA.01
(3)	ws-command	Command to perform, where:
		1: HOLD (Job, Segment or Unit)
		2: RELEASE (Job, Segment or Unit)
		3: START (Job, Segment or Unit)
		4: RESTART (Unit Only)
		5: ABORT (Unit Only)
		6: RESTART, IF FAILED (Unit Only)

NOTE	FIELD	DESCRIPTION
(4)	ws-reply	**Reply-FIO-Err 0 : Command Successful**
		Reply-Code 1 : JOB Commands OK
		Reply-Code 2 : SEGMENT Commands OK
		Reply-Code 3 : UNIT Commands OK
(4)	ws-reply	**Reply-FIO-Err -1 : Command Error**
		Attempting to use BMON:
		Reply-Code 9000 : No BMON or no ENTITY Supplied
		Reply-Code 9001 : Invalid BMON
		Reply-Code 9002 : BMON Cannot be Opened
		Reply-Code 9006 : BMON cannot be written to
		Reply-Code 9007 : Restart or Abort attempt on a Job or Segment
		Reply-Code 9008 : Invalid Command

The Reply-Text will have more detail about the cause of the error.

Example COBOL Source Code

Compiler directives are used to differentiate between builds for different platforms.

Note: *The different builds will use a unique define for a specific version of LIBUTIL. For example, when compiled with toggle 5, a build of the code 500 object will be performed that utilises the define =LIBUTIL5, which should have been previously configured to point to the LIBUTIL5 library file.*

The example COBOL source code below and on the following pages is contained within the file called MHARNESS, located in the MultiBatch Object subvolume, e.g. MBATOBJ.

```
?SYMBOLS, INSPECT
 IDENTIFICATION DIVISION.
 PROGRAM-ID. MHARNESS.

 ENVIRONMENT DIVISION.

 CONFIGURATION SECTION.

 SPECIAL-NAMES.
?IF 1
* Standard TNS (code 100) build
    FILE "=LIBUTIL" IS LIBUTIL.
?ENDIF 1

?IF 5
* Standard TNS/X (code 500) build
    FILE "=LIBUTIL5" IS LIBUTIL.
?ENDIF 5

?IF 7
* Standard TNS/R (code 700) build
    FILE "=LIBUTIL7" IS LIBUTIL.
?ENDIF 7

?IF 8
* Standard TNS/E (code 800) build
    FILE "=LIBUTIL8" IS LIBUTIL.
?ENDIF 8

 DATA DIVISION.

 WORKING-STORAGE SECTION.

?IF 1
* Standard TNS (code 100) build
  01 linkage-data-1.
     COPY LINKAGE1-TNS OF "HARNCOPY".

  01 linkage-data-2.
     COPY LINKAGE2-TNS OF "HARNCOPY".
?ENDIF 1

?IFNOT 1
* TNS/R (code 700) or TNS/E (code 800) build or TNS/X (code 500)
  01 linkage-data-1.
     COPY LINKAGE1-NATIVE OF "=HARNCOPY".

  01 linkage-data-2.
     COPY LINKAGE2-NATIVE OF "=HARNCOPY".
?ENDIF 1

  01 text-1   PIC X(37) VALUE "ENTER AS MUCH INFORMATION AS YOU KNOW".

  01 text-2   PIC X(47) VALUE "Please enter BMON process e.g. \MYNODE.$AA01".

  01 text-3   PIC X(70) VALUE "Please enter required operation from the following list".

  01 text-4   PIC X(37) VALUE "1)  HOLD    (Job, Segment or Unit)".

  01 text-5   PIC X(37) VALUE "2)  RELEASE (Job, Segment or Unit)".

  01 text-6   PIC X(37) VALUE "3)  START   (Job, Segment or Unit)".

  01 text-7   PIC X(37) VALUE "4)  RESTART (Unit only)".

  01 text-8   PIC X(37) VALUE "5)  ABORT   (Unit only)".

  01 text-9   PIC X(37) VALUE "6)  RESTART, IF FAILED (Unit only)".

  01 text-10  PIC X(75)
     VALUE "Please enter complete entity name (job, segment and unit) e.g. AAAA.BB.01".

  01 text-11  PIC X(61)
 01 text-12  PIC X(44) VALUE "Please enter alias name e.g. Processing Jobs".

 01 message-line  PIC X(80).

PROCEDURE DIVISION.

A-CONTROL-SECTION.

   PERFORM CA-INITIALISE.

   PERFORM CC-CALL-TAL-PROGS.

A-CONTROL-SECTION-EXIT.
   STOP RUN.

CA-INITIALISE SECTION.
   DISPLAY text-1.
   DISPLAY text-2.
   ACCEPT  ws-bmon OF linkage-data-1.
   DISPLAY text-3.
   DISPLAY text-4.
   DISPLAY text-5.
   DISPLAY text-6.
   DISPLAY text-7.
   DISPLAY text-8.
   DISPLAY text-9.
   ACCEPT  ws-command.
   DISPLAY text-10.
   ACCEPT  ws-entity OF linkage-data-1.
   DISPLAY text-11.
   ACCEPT  ws-dataconf.
   DISPLAY text-12.
   ACCEPT  ws-alias.

 CA-INITIALISE-END.
   EXIT.

CC-CALL-TAL-PROGS SECTION.

   ENTER TAL "nv^val^entity" OF LIBUTIL
       USING  ws-bmon    OF linkage-data-1,
              ws-entity  OF linkage-data-1,
              ws-dataconf,
              ws-alias
              ws-reply   OF linkage-data-1
       GIVING reply-code OF linkage-data-1

   IF reply-code OF linkage-data-1 = -1
       MOVE reply-text OF ws-reply OF linkage-data-1 TO message-line
   ELSE
       MOVE reply-text OF ws-reply OF linkage-data-1 TO message-line
   END-IF

   DISPLAY message-line

   IF reply-code OF linkage-data-1  =  00
       MOVE reply-bmon           OF linkage-data-1 TO ws-bmon OF linkage-data-2

       MOVE reply-entity         OF linkage-data-1 TO ws-entity OF linkage-data-2
       ENTER TAL "nv^do^command" OF LIBUTIL
           USING  ws-bmon        OF linkage-data-2
                  ws-entity      OF linkage-data-2
                  ws-command     OF linkage-data-2
                  ws-reply       OF linkage-data-2
           GIVING reply-code     OF linkage-data-2

       MOVE reply-text OF ws-reply OF linkage-data-2 TO message-line

       DISPLAY message-line
   END-IF.

 CC-CALL-TAL-PROGS-EXIT.
```

LIBUTIL - Code 100 MHARNESS Compile

When attempting to compile the supplied example source code - MHARNESS - ensure the following is applied:

- DEFINE =LIBUTIL has been configured to reference the correct LIBUTIL library
- ADD DEFINE =HARNCOPY , FILE <MultiBatch Object sub volume>.HARNCOPY
- Compilation is performed in the same subvolume as the HARNCOPY file, i.e. the MultiBatch Object subvolume
- The COBOL compilation command specifies the use of toggle 1, e.g.

```
COBOL85/IN MHARNESS, OUT $S.#HARN/HARNESO; SETTOG 1
```

LIBUTIL5 - Code 500 MHARNESS Compile

When attempting to compile the supplied example source code - MHARNESS - ensure the following is applied:

- DEFINE =LIBUTIL5 has been configured to reference the correct LIBUTIL5 library
- ADD DEFINE =HARNCOPY, FILE <MultiBatch Object subvolume>.HARNCOPY
- Compilation is performed in the same subvolume as the HARNCOPY file, i.e. the MultiBatch Object subvolume
- The XCOBOL compilation command specifies the use of toggle 5, e.g.

```
XCOBOL/IN MHARNESS, OUT $S.#HARN/HARNESO ; RUNNABLE , SETTOG 5
```

LIBUTIL8 - Code 800 MHARNESS Compile

When attempting to compile the supplied example source code - MHARNESS - ensure the following is applied:

- DEFINE =LIBUTIL8 has been configured to reference the correct LIBUTIL8 library
- ADD DEFINE =HARNCOPY , FILE <MultiBatch Object subvolume>.HARNCOPY
- Compilation is performed in the same subvolume as the HARNCOPY file, i.e. the MultiBatch Object subvolume
- The ECOBOL compilation command specifies the use of toggle 8, e.g.

```
ECOBOL/IN MHARNESS, OUT $S.#HARN/HARNESO ; RUNNABLE , SETTOG 8
```

LIBUTIL5 - Code 500 Integration

The code 500 LIBUTIL Library file is shipped as LIBUTIL5.

Make the code or define changes as required to pick up this LIBUTIL5 object, or alternatively rename the file to LIBUTIL; if older LIBUTIL objects exist, then also rename as required.

To integrate the code 500 library file, load the defines that will allow any MAP references to be satisfied. Compile the application code; for example, COBOL application USERAPP:

```
XCOBOL/IN USERAPP, OUT $S.#UA/USERAPPO
```

Link the compiled objects using XLD to create the final object:

```
XLD/IN UALINK, OUT $S.#UAL/
```

Where an example XLD script file, UALINK, would be:

```
--------------------------------------------
---       HP Native COBOL Link file      ---
---              mmm yyyy               ---
-------------------------------------------
-------------------------------------------
---     List of object files to link     ---
-------------------------------------------
=USERLIB1
-------------------------------------------
--- User libraries to link into object ---
-------------------------------------------
=VPROCLIBRARY5
-------------------------------------------
--- System libraries to link into object ---
-------------------------------------------
-lZCOBDLL
-lZCREDLL
-------------------------------------------
---            Linker Options            ---
-------------------------------------------
-ALLOW_DUPLICATE_PROCS
-b dynamic
-verbose
-------------------------------------------
---           Final object file         ---
-------------------------------------------
-o =BINARYFILE
```

Which uses these defines:

=USERLIB1 is the XCOBOL compiled object code: USERAPPO.

=VPROCLIBRARY5 is any other (optional) library such as VPROC library.

=BINARYFILE is the output from the XLD session, e.g. USERAPPL: this is the code that you execute.

LIBUTIL8 - Code 800 Integration

The code 800 LIBUTIL Library file is shipped as LIBUTIL8.

Make the code or define changes as required to pick up this LIBUTIL8 object, or alternatively rename the file to LIBUTIL; if older LIBUTIL objects exist, then also rename as required.

To integrate the code 800 library file, load the defines that will allow any MAP references to be satisfied.

Compile the application code; for example, COBOL application USERAPP:

```
ECOBOL/IN USERAPP, OUT $S.#UA/USERAPPO
```

Link the compiled objects using XLD to create the final object:

```
ELD/IN UALINK, OUT $S.#UAL/
```

Where an example ELD script file, UALINK, would be:

```
-------------------------------------------
---          HP Native COBOL Link file          ---
---                 mmm yyyy                 ---
-------------------------------------------
-------------------------------------------
---     List of object files to link     ---
-------------------------------------------
```

```
--- User libraries to link into object ---
-------------------------------------------------------------
=VPROCLIBRARY8
-------------------------------------------------------------
--- System libraries to link into object ---
-------------------------------------------------------------
-lZCOBDLL
-lZCREDLL
-------------------------------------------------------------
---                    Linker Options                    ---
-------------------------------------------------------------
-ALLOW_DUPLICATE_PROCS
-b dynamic
-verbose
-------------------------------------------------------------
---                  Final object file                   ---
-------------------------------------------------------------
-o =BINARYFILE
```

Which uses these defines:

=USERLIB1 is the ECOBOL compiled object code: USERAPPO.

=VPROCLIBRARY8 is any other (optional) library such as VPROC library.

=BINARYFILE is the output from the ELD session, e.g. USERAPPL: this is the code that you execute.

F.15 PSCHED

A utility program is provided for use with the Event Timer System, namely:

PSCHED

This is intended to provide a screen listing or hard-copy record of the Schedule Database. The syntax for execution of PSCHED is (default spooler location is #SCHED):

```
      PSCHED / IN DATACONF / <option>
```

The options provided are as follows:

```
!!! PSCHED / IN dataconf filename, OUT spooler.loc/ OPTION

!!! DEFAULT OUT is $S.#SCHED

!!! OPTIONS - HELP
!!!         - ALL
!!!         - FROM DD/MM/YYYY,[HH:MM]
!!!         - TO DD/MM/YYYY,[HH:MM]
!!!         - FROM DD/MM/YYYY,[HH:MM] TO DD/MM/YYYY,[HH:MM]
!!!         - TODAY
!!!         - TOMORROW or TM
```

HELP. Prints the syntax for running PSCHED.

ALL. Returns information about all Schedule Records in the Database.

FROM DD/MM/YYYY, [hh:mm]. Prints a listing of all entities scheduled for execution from the date specified.

TO DD/MM/YYYY, [hh:mm]. Prints a listing of all entities scheduled for execution up to the date specified.

This may be used in conjunction with FROM TO list the scheduled entities between two dates, for example:

> **FROM DD/MM/YYYY, [hh:mm] TO DD/MM/YYYY, [hh:mm]**. Prints a listing of all entities between the specified dates and times.
>
> **TODAY**. Prints a listing of all entities scheduled for execution during the current day.
>
> **TOMORROW or TM**. Prints a listing of all entities scheduled for execution during the day following the current day.

F.16 RESCHED

RESCHED performs a resynchronization of the system following a possible failure of the NonStop system where DS-Event Timer is running.

The RESCHED program is provided to cater for circumstances when the Event Timer system is to be restarted following a system crash, cold-load or extended outage for any other reason.

When the background CALENDAR process starts, it immediately reads the Schedule Database looking for records which are "ready to run".

In this context, this would include any records which have passed their "Projected Next Run" without the Calendar Process generating an EMS Event as a signal to the Trigger Process that the entity should be started (because the clock has moved on while the System was down or unavailable, and the Calendar Process was therefore unable to perform its duty.)

Upon finding each Schedule Record which qualifies, the Calendar Process would immediately generate an EMS Event, irrespective of the current Time and Date, and the Trigger Process would dutifully attempt to start the requested MultiBatch entity.

This may be a perfectly reasonable thing to do following a System Crash as you may wish to have Event Timer "catch up" in this way.

However, after an unplanned System outage there would obviously be a number of other factors to be taken into account before beginning processing of the workload, such as first reloading the appropriate MultiBatch monitor processes (or running Recover BMON for each of them).

The user has the option to control the behavior of the Event Timer facility by using the RESCHED program provided for this purpose.

The syntax for execution of RESCHED is:

```
RESCHED / IN <DATACONF filename>, OUT <spooler-location> / <options>
```

A summary of the changes made to any Schedule Records will be reported to the terminal from where RESCHED is executed, irrespective of the OUT device specified.

There are two types of Parameters used as <options> for RESCHED, namely, "FROM" parameters and "HOLD" parameters. These parameters can be used either singly or in combination. The effect of their use is as follows:

FROM NOW. All Schedule Records with STATUS of "S" will be rescheduled. The "NOW" value will be interpreted to mean from "midnight just passed".

FROM DD/MM/YYYY, hh:mm:ss. All Schedule Records with a STATUS of "S" will be rescheduled; the date and time value will be used to calculate the date of the next execution of each of these records.

HOLDON[!]. All Schedule Records with a STATUS of "S" will be given a new STATUS of "H", which will 'HOLD' the entire Schedule. Where the [!] option is used all records are changed to status of "H" regardless of the original status.

HOLDOFF[!]. All Schedule Records with a STATUS of "H" will be given a new STATUS of "S". Where the [!] option is used all records are changed to status of "S" regardless of the original status.

Note: Where **HOLDON[!]** and **HOLDOFF[!].** are used as the only option, they must be preceded by a comma to indicate that this is the case.

FROM DD/MM/YYYY hh:mm:ss, HOLDON[!]. This is an invalid combination of options, as the HOLDON would first change all "S" STATUS Records to "H" STATUS, then attempt to recalculate all "S" STATUS Schedule Records. As there would not then be any in an "S" STATE, this would be a meaningless combination.

Records to "S" STATUS, then recalculate all "S" STATUS Schedule Records to next execute from the supplied "FROM" date. Where the [!] option is used the all records are changed to status of "H" regardless of the original status. Where the [!] option is used all records are changed to status of "S" regardless of the original status.

Use of the RESCHED facility makes most sense in a situation when you wish to manipulate all Schedule Records by: Stopping the CALENDAR and TRIGGER Processes to force the Event Timer system to a complete halt.

Using the HOLDON facility to globally switch all Records to "H" STATUS.
Using the FROM DD/MM/YYYY facility to force recalculation from a future date (such as tomorrow).

Using the HOLDOFF facility to switch all Schedule records back into "S" STATUS ready for a resumption of processing from the new date.

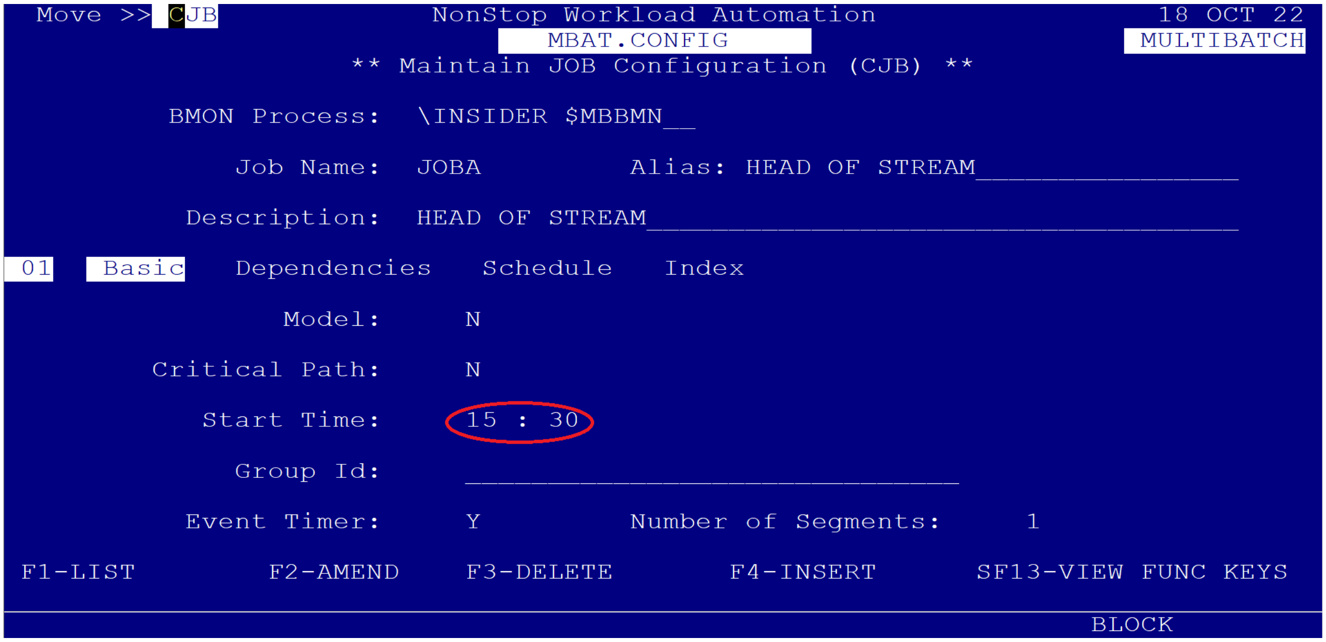
Restarting the CALENDAR and TRIGGER processes.

This sequence allows you to easily visualise what is happening at each stage and to perform clerical checks on the state of all records, either by use of the Pathway Screens or the PSCHED utility, at the end of each of these stages.

F.17 RESCHDST

The RESCHDST utility can be used to update the MultiBatch JOBCONF database so that the MultiBatch JOB 'start time' value is set to a new value or adjusted from its current value.

This value can be set manually in the MultiBatch PATHWAY Configure Job (CJB) screen.



```
  Move >> CJB                NonStop Workload Automation               18 OCT 22
                                    MBAT.CONFIG                       MULTIBATCH
                         ** Maintain JOB Configuration (CJB) **

          BMON Process:  \INSIDER $MBBMN__

              Job Name:  JOBA         Alias: HEAD OF STREAM________________

           Description:  HEAD OF STREAM____________________________________

 01   Basic    Dependencies   Schedule   Index

                 Model:     N

         Critical Path:     N

            Start Time:     15 : 30

              Group Id:     ______________________________

           Event Timer:     Y         Number of Segments:     1

 F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                     BLOCK
```

Figure F.6 – Maintain Job Configuration (CJB)

RESCHDST Syntax

An example OBEY file RUNRST and associated sample SWITCHST file are supplied in the MultiBatch object subvolume; example contents of RUNRST:

RUNRST

```
                                                              Notes
  PARAM PREVIEW-SWITCHST N                                    (1)
  ASSIGN SWITCHST, <DISK>.MBATOBJ.SWITCHST                    (2)
  RUN RESCHDST/ IN <DISK>.MBATDAT.DATACONF, OUT $S/           (3)
```

Note (1):

To allow users to verify that the content of the SWITCHST file is correct, a run time parameter (PREVIEW-SWITCHST) can be set. A report will be produced, but no file updates will be undertaken.

If the parameter value is "Y" or "y", then a preview will be executed. Any other value will result in updates being applied.

The parameter is not mandatory. The default will be Update and not Preview.

The Preview will check that the SWITCHST record exists in the JOBCONF file and provide a summary of the action to be taken against it.

Note (2):

The SWITCHST file will be used to drive the 'start time' updates made to specific records in the JOBCONF file.

This section deals with the format of this SWITCHST file.

Each SWITCHST definition must occupy a separate line in the edit file.

JOB NAME

ACTION

JOB NAME is in the format NODE.BMON.JOB. The values will be validated as below; errors will be reported to the home terminal and the program will then abend.

Node: Start with a "|”, minimum length 2, maximum 8.

BMON: Start with a “$”, minimum length 2, maximum 6.

JOB: Length 4.

ACTION can be either A or T (Adjust or Time)

A[+|-] HH:MM or

T HH:MM [!]

The ‘A’ option will add or subtract the HH:MM time value from the current job time.

A- 01:00 will subtract one hour from the current job time.

A+ 02:30 will add two hours thirty minutes to the current job time.

Format notes:

There must be no space character between ‘A’ and the +|- character. There must be a space character between +|- and the hour.

The hour value must be 2 digits.

Midnight is stored in the JOBCONF file as 24:00, if you attempt to alter an existing value of 00:00 (no start time) a warning is issued.

The ‘T’ option will set the current job time to HH:MM. T 21:30 will set the start time to 21:30.

Format notes:

There must be a space character between ‘T’ and the hour. The hour value must be 2 digits.

Midnight is stored in the JOBCONF file as 24:00, so T 24:00 will set the start time to midnight, 00:01 is one minute past midnight and 00:00 is no start time.

If the current start time is switched off (00:00) and you want to set it, then the “!” option is required.

T 21:30 !

If you want to switch off the current start time then the “!” option is required.

T 00:00 !

The values in the SWITCHST file will be shifted to upper case before validation and subsequent use in key positioning.

A ‘?’ character in column 1 will treat the rest of the line as comment. Blank lines will be ignored.

Note (3)

The location of the JOBCONF file is determined by reading that record from the DATACONF file.

A report file will be produced detailing the changes made. The changes are also listed out to the home terminal.

Errors are reported to the home terminal and not to EMS. The program will abend if errors are detected.

RESCHDST will create and end or abort TMF transactions. The TMF subsystem must be active when RESCHDST is executing.

Example SWITCHST records

\A.$BMON.JOBA, T 09:00!	Set the start time for JOBA to 09:00. If the start time is not set, allow the update.

\A.$BMON.JOBA, T 24:00	Set the start time for JOBA to midnight
\A.$BMON.JOBA, A+ 01:00	Add 1 hour to the start time.
\A.$BMON.JOBA, A- 01:00	Subtract 1 hour from the start time.
?Overnight archiving suite	This is a comment line.

Error Messages

All error messages will be written to the RESCHDST home terminal. There is no EMS output.

SWITCHST File Errors

The SWITCHST file format must be:

<NODE>.<BMON>.<JOB>, <ACTION>

<BMON>, <JOB>, <SERIAL> and <ACTION> are mandatory.

<NODE> must begin with a "\" character and be 2 to 8 characters in length.

<BMON> must begin with a "$" character and be 2 to 6 characters in length.

<JOB> must be 4 characters in length.

<ACTION> must be A[+|-] HH:MM or T HH:MM

If the syntax of a SWITCHST record is incorrect, then the RESCHDST program will abend and one of the following errors will be displayed. JOBCONF changes made prior to the abend will not be backed out.

Each message will be prefixed with:

!!! Error in SWITCHST file (\<NODE>.<BMON Process>.<JOB ID>)

The possible errors are:

- Invalid node name
- BMON process name not found
- JOB name must be specified
- Invalid BMON process name
- No comma found after JOB
- Invalid job name
- Invalid option specified (Note A or T)
- Invalid TIME specified
- Invalid TIME (HH:MM)

The SWITCHER file must be an edit file, e.g. SWITCHST.

- !!! Illegal syntax : SWITCHER file must be an EDIT file

If you specify a job, segment, or unit name in the SWITCHST file and it does not exist in the JOBCONF file, then RESCHDST will continue to process, but the user will observe the following warning:

<NODE>.$BMON.JOBA : Not found in JOBCONF file

File Errors

!!! Error on access to FFFFFFFF : EEEE

FFFFFFF is the file name and can be; DATACONF, HOMETERM, JOBCONF, OUTFILE or SWITCHST.

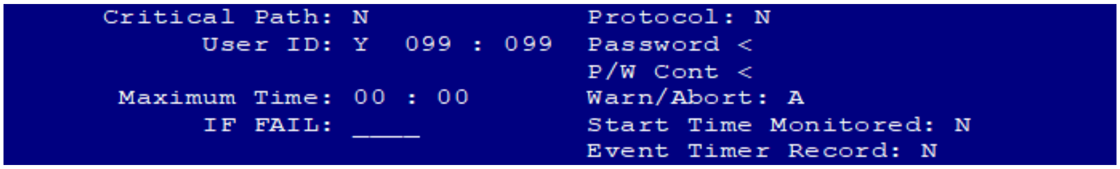
EEEE is the Guardian file error.

!!! TMF error detected : EEEE

Appendix G Verify User Id

The Maintain STEP Configuration (CST) and Maintain OSS STEP Configuration (COS) screens allow a user to configure a Guardian group, user id that a MultiBatch step will execute as.

For example:



```
      Critical Path: N              Protocol: N
            User ID: Y   099 : 099  Password <
                                    P/W Cont <
       Maximum Time: 00 : 00        Warn/Abort: A
            IF FAIL: ____           Start Time Monitored: N
                                    Event Timer Record: N
```

Figure F.7 – Maintain Step Configuration (CST) – User ID

The standard options are:

1. Use the User ID feature = Y.

   Supply a Guardian user in the format group number, user number, e.g. 099,099. The user id must exist on the NonStop system where the configuration is taking place. The Step will then run under this specified group, user ID number.

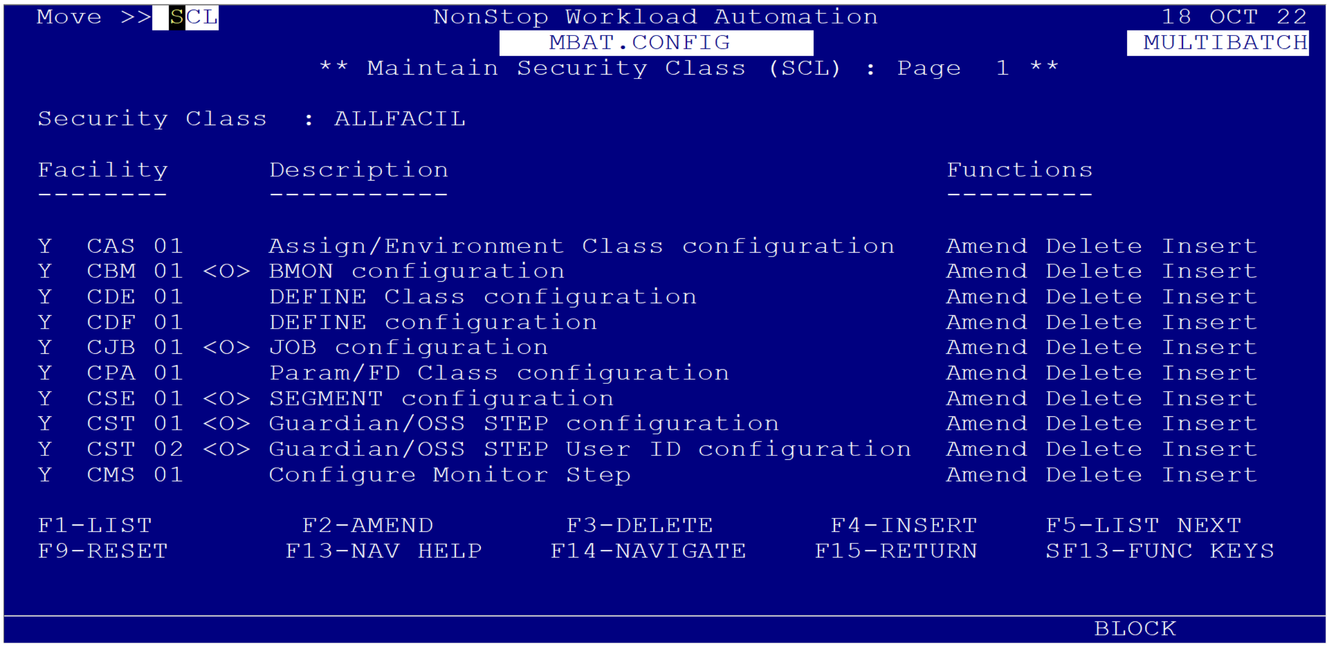
2. Use the User ID feature = N.

   Step when run, will assume the user id of the parent BMON process.

In addition to utilising Y or N, there is a third option V, where a user can allocate a group, user id which does not exist on the MultiBatch node where the CST configuration is being performed.

This may be useful on say a 'control' node where MultiBatch schedules are configured before eventual migration to Live, Dev and Test nodes for example and where this group, user id does exist.

To utilise the Verify User function, the CST 02 Facility in Security Class, or Classes, must be set to Y or P, e.g.:



```
 Move >> SCL              NonStop Workload Automation                 18 OCT 22
                               MBAT.CONFIG                           MULTIBATCH
                    ** Maintain Security Class (SCL)  : Page   1 **

 Security Class   : ALLFACIL

 Facility       Description                                 Functions
 --------       -----------                                 ---------

 Y  CAS 01      Assign/Environment Class configuration      Amend Delete Insert
 Y  CBM 01 <O> BMON configuration                          Amend Delete Insert
 Y  CDE 01      DEFINE Class configuration                  Amend Delete Insert
 Y  CDF 01      DEFINE configuration                        Amend Delete Insert
 Y  CJB 01 <O> JOB configuration                           Amend Delete Insert
 Y  CPA 01      Param/FD Class configuration                Amend Delete Insert
 Y  CSE 01 <O> SEGMENT configuration                       Amend Delete Insert
 Y  CST 01 <O> Guardian/OSS STEP configuration             Amend Delete Insert
 Y  CST 02 <O> Guardian/OSS STEP User ID configuration     Amend Delete Insert
 Y  CMS 01      Configure Monitor Step                      Amend Delete Insert

 F1-LIST          F2-AMEND       F3-DELETE       F4-INSERT     F5-LIST NEXT
 F9-RESET        F13-NAV HELP   F14-NAVIGATE    F15-RETURN    SF13-FUNC KEYS


                                                             BLOCK
```

Figure F.8 – Maintain Security Class Configuration (SCL) – CST 02

Creating a List of Supported Users

If the CST 02 Security Class Facility is set to Y, then the group, user ID on CST/COS will be validated and must exist.

If the CST 02 Security Class Facility is set to P, then then the group, user ID on CST/COS will be validated, must exist and the password field must be completed. See Chapter 2.5 'Building Security Classes (SCL)' for further details on the use of Y and P values.

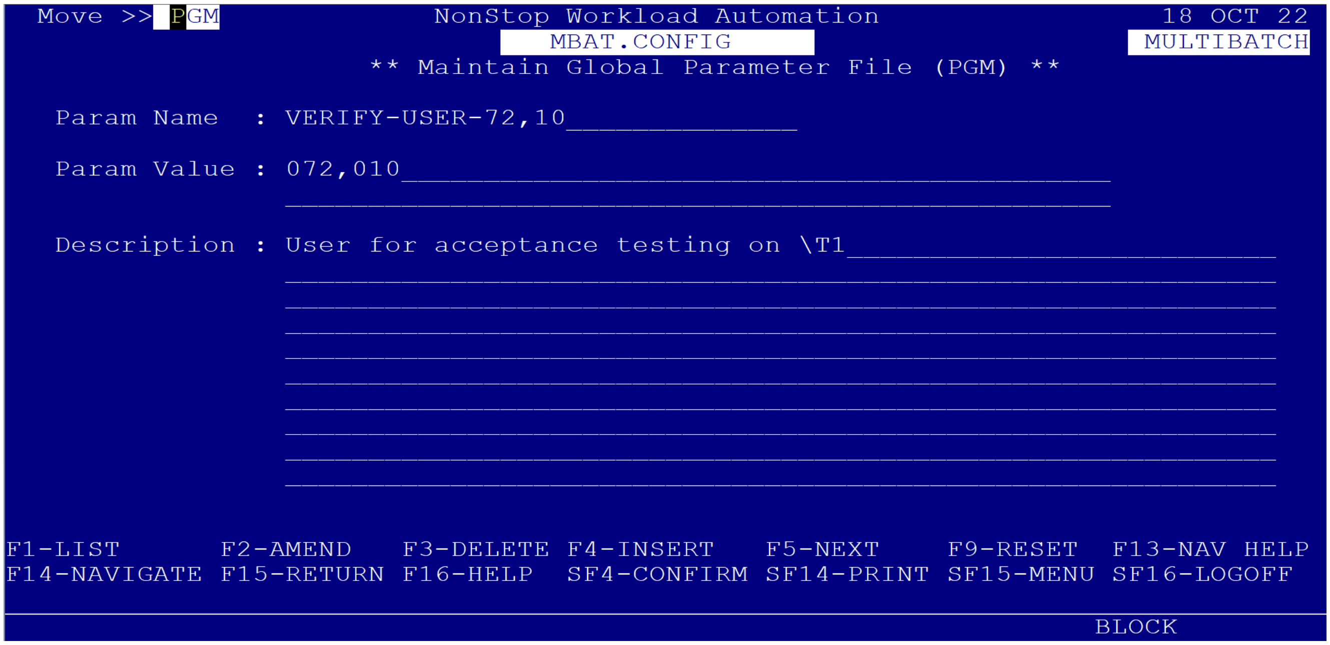
However, within CST/COS, if the 'User ID:' field is set to V (Verify User), then this group, user ID needs to have been pre-configured within the Maintain Conditional Parameter Files Menu (PME) as a Global parameter.

To create a Global parameter, the Security Class must have the 'Conditional Parameters Screens' facility set to Y:

Creating a Global Parameter

Navigate to the PME facility and select F1 'Create Global Parameters'.

Configure a Param Name, Param Value and if required, a suitable Description, e.g.:



```
  Move >> PGM                 NonStop Workload Automation                18 OCT 22
                                   MBAT.CONFIG                          MULTIBATCH
                      ** Maintain Global Parameter File (PGM) **

   Param Name  : VERIFY-USER-72,10______________

   Param Value : 072,010___________________________________________
                 __________________________________________________

   Description : User for acceptance testing on \T1______________________________
                 ____________________________________________________________
                 ____________________________________________________________
                 ____________________________________________________________
                 ____________________________________________________________
                 ____________________________________________________________
                 ____________________________________________________________
                 ____________________________________________________________
                 ____________________________________________________________
                 ____________________________________________________________

F1-LIST      F2-AMEND   F3-DELETE F4-INSERT   F5-NEXT    F9-RESET  F13-NAV HELP
F14-NAVIGATE F15-RETURN F16-HELP  SF4-CONFIRM SF14-PRINT SF15-MENU SF16-LOGOFF

                                                                BLOCK
```

Figure F.9 – Maintain Global Parameter File (PGM) – Verify User

Note: *Do not use the PME Local Parameter database, as this will not be accessed by the 'Verify User' function.*

Param Name

The value must have a prefix of "VERIFY-USER-" followed by a Serial Number, e.g.:

- VERIFY-USER-1
- VERIFY-USER-72,10

The numbers do not have to be sequential, i.e. they do not need start at 1 and rise in sequence. Alternatively, the VERIFY-USER- can include a group, user id which may assist with identification.

Param Value

The value must be a Guardian user id in the form of group, user ID number. It does not matter if the group, user ID exists or not, as it will not be validated.

The two group, user Param Values must each be three characters in length.

The comma must be in position four.

There should be no space separators.

Param Value examples:

Valid:	255,255
Invalid:	1,11
Invalid:	255 , 255
Invalid:	SUPER.SUPER

As this is a general parameter maintenance screen, there will no validation of the input.

Description

Free format text.

Once the entry is complete, press F4 to insert the data.

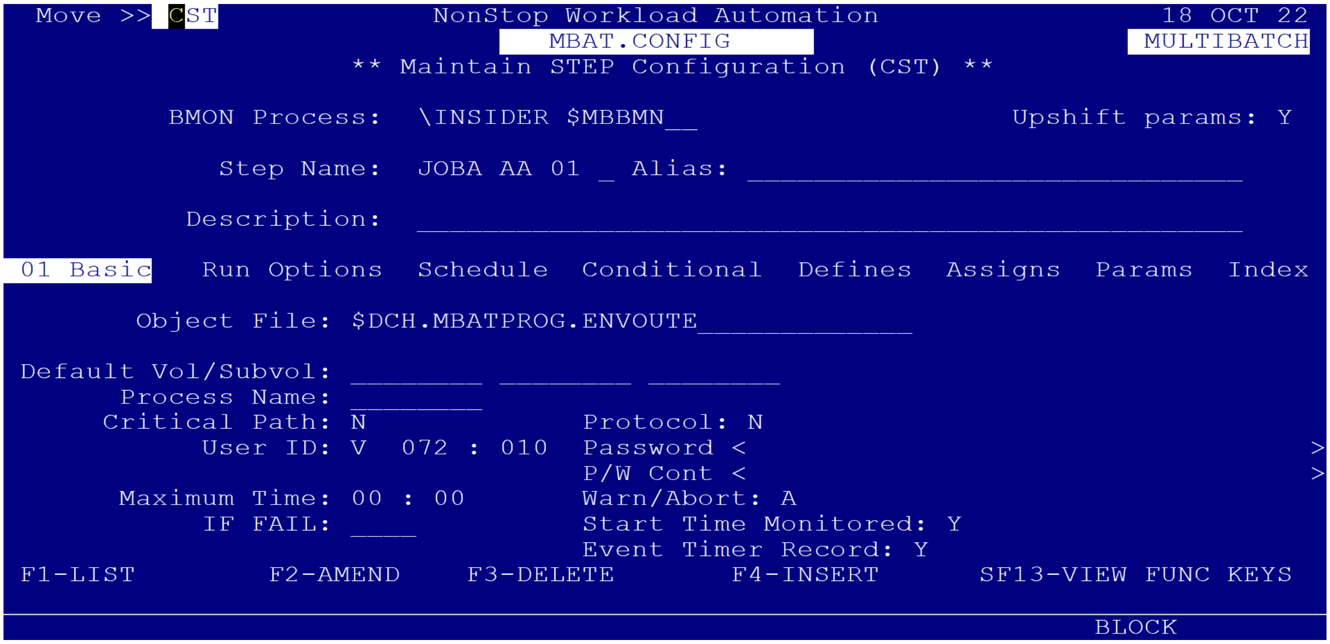
A list of all Global Parameters can be viewed by selecting F2 "List Global Parameters" from the "Maintain Conditional Parameter Files Menu (PME)" screen.

The standard tab, mark and F8 (Detail) functions are supported from the List screen.

To complete the user of a Global Parameter within a CST configuration, see instructions on next page.

CST/COS Configuration

Example CST configuration showing the use of the V (Verify User) value and associated group, user ID.



```
 Move >> CST              NonStop Workload Automation                 18 OCT 22
                              MBAT.CONFIG                            MULTIBATCH
                      ** Maintain STEP Configuration (CST) **

        BMON Process:  \INSIDER $MBBMN__                   Upshift params: Y

           Step Name:  JOBA AA 01 _ Alias: ______________________________

         Description:  __________________________________________________

 01 Basic   Run Options   Schedule   Conditional   Defines   Assigns   Params   Index

       Object File: $DCH.MBATPROG.ENVOUTE_____________

 Default Vol/Subvol: ________ ________ ________
       Process Name: ________
      Critical Path: N             Protocol: N
            User ID: V  072 : 010  Password <                              >
                                   P/W Cont <                              >
       Maximum Time: 00 : 00       Warn/Abort: A
            IF FAIL: ____          Start Time Monitored: Y
                                   Event Timer Record: Y
 F1-LIST        F2-AMEND    F3-DELETE       F4-INSERT      SF13-VIEW FUNC KEYS

                                                                 BLOCK
```

Figure F.10 – Maintain Step Configuration (CST) – Verify User

The existing N and Y settings and any associated validation remain unchanged.

If the 'User ID:' field is set to V, then the CST insertion or amendment will look for a "VERIFY-USER-" prefixed value in the Global Parameter file. If this does not exist, then the following message will be displayed:

> "USERID IS NOT VALID (NOT FOUND IN GLOBAL FILE)"

If the value does exist, then the record will be inserted/updated as appropriate.

Appendix H MultiBatch Streamer Process

H.1 MultiBatch Streamer

The Streamer process is supplied as part of the MultiBatch software and is started by a program called STREAMER. The process uses supplied translation rules to translate log messages written to it, into tokenised EMS events.

Any log messages that the Streamer process does not recognize will result in an EMS event being emitted which will give details of the text. This situation should not occur as the streamer configuration is updated and re-released at each software upgrade.

The MultiBatch Stream process runs as a non stop process pair.

The process can act as a home terminal for your application, processing COBOL library errors and handling INSPECT and DEBUG prompts.

If a DEBUG prompt is detected, the trapped program is stopped. If an INSPECT prompt is detected, then a SAVEABEND file is created and the trapped program is stopped. In both instances, an EMS event is emitted giving details of the trap.

To Start the MultiBatch Streamer process, enter the following at the TACL prompt:

```
RUN <Object Svol>.STREAMER / NAME $<your-choice>, IN BCNF0998, OUT &
<Collector>, NOWAIT, CPU <N>, TERM <TERM> / BACKUP^CPU <N>, NON^CONFIG &
<filename>, DATACONF <Database Subvol.DATACONF>
```

Note: *The Streamer Process is required for the successful operation of the Watcher Process and Event Timer.*

The BCNF0998 Streamer configuration file is supplied as part of your MultiBatch release.

The OUT <collector> run option may be specified if you wish to send events to an Alternate collector, rather than using $0, which is used by default, e.g. RUN STREAMER / OUT $ACOL,   etc.

The BACKUP^CPU value is the target cpu for the backup process of the MultiBatch Streamer program. The process will ALWAYS run as a non stop process pair. The default value is "highest available cpu" which is also adopted if the BACKUP^CPU value is invalid.

The optional NON^CONFIG file is the destination for any records for which the streamer has no translation rules. It must be Entry Sequenced and 255 bytes long and if this is not the case the Streamer abends with an appropriate EMS diagnostic produced. If the file does not exist, then one is created. The default is no mismatch records are dumped.

The DATACONF file parameter allows the STREAMER process to locate the ALIASMAP file and place an applicable alias in a log message e.g. "JOB.AA.01.* (WEEKLY-UPDATE), unit complete".

Note: *If the user alters an existing alias or creates a new alias, it may not be implemented by the Streamer process immediately. The streamer will check every 100 messages for updates to the ALIASMAP file and re-read the new ALIASMAP table if appropriate. This is to make the streamer processing as efficient as possible. If a new alias needs to be implemented immediately, then warmboot Streamer via the NCOM utility.*

Note: *Any BMON process that has a "**stopped**" streamer open would close the streamer log file. It would need to be reopened using a BCOM LOG1 or LOG2 command.*

The parameters are in fixed positions:

a) If the BACKUP^CPU parameter is defaulted then the parameter string must be ", NON^CONFIG <file>".

b) If the BACKUP^CPU and NON^CONFIG parameters are defaulted then the parameter string must be ", DATACONF <file>".

Appendix I Recovering a BMON

The MultiBatch BMON process writes its configuration status of its schedule to a TMF Audited Recovery file. See Figure 3.4 – Maintain BMON Configuration Screen (CBM)

This Recovery file is TMF protected and therefore, can be replicated to a backup/standby NonStop system using suitable data replication products.

In the event of say a system crash or a BMON failure, the BMON configuration including its schedule status up to the point of failure can be recovered, either on the current production node, or following a site-swap, on the 'new' production node.

I.1 Recovering a BMON

The MultiBatch BMON process writes status of its schedule to a TMF Audited Recovery file. See Figure 3.4 – Maintain BMON Configuration Screen (CBM)

This Recovery file is TMF protected and therefore, if required, can be replicated to a backup/standby NonStop system using suitable data replication products.

In the event of say a system crash or a BMON failure, the BMON configuration including its schedule status up to the point of failure can be recovered, either on the current production node, or following a site-swap, on the 'new' production node.

In order to recover from a BMON failure, perhaps due to a CPU crash, or where a site-swap has been invoked, there is a configuration pre-requisite.

To ensure the BMON can be Recovered, the supplied BMONDEFS file needs to be configured appropriately.

This file is resident on the MultiBatch Object subvolume, e.g. MBATOBJ and contains old and new node names, volumes and subvolumes, including old and new OSS directory values.

During the Recovery of a BMON, the old values within the configuration will be automatically changed to the new values.

For example, if a Production node is called \LIVE and a site-swap to \BACKUP was invoked, then the BMONDEFS file on \BACKUP will contain the current Production node value of \LIVE and the standby/backup node value of \BACKUP, along with appropriate volume and subvolume names:

BMONDEFS - Used in the SYNC IN process and MIGRCONP to migrate conditional params. Amend contents to translate nodes, volumes, subvolumes and OSS directories.

```
NODE    \LIVE   \BACKUP
VOLUME  $LIVE   $APPS
VOLUME  $DATA01 $DATA02
SUBVOL  APPDATA1APPDATA2
SUBVOL  CONFIG1 CONFIG2
DIR     olddir  newdir
```

The old values start in column **9** and the new values start in column **17**.

An example obey file called SBMON is supplied on the MultiBatch Object subvolume, e.g. MBATOBJ and a description of the contents is provided in Appendix K SBMON Obey File.

I.2 Example BMON Recovery

An example of recovering a <NODE1> BMON process on the <NODE> node is provided.

The user had decided to ensure that the TMF protected <NODE1> BMON Recovery file is being replicated to the <NODE2> backup node.

<NODE1> NODE

- <NODE1> BMON is processing its schedule.
- <NODE1> BMON Recovery file is being replicated to the <NODE2> node via suitable means, e.g. RDF.
- During the running of the schedule, the <NODE1> node crashes.
- A site-swap is invoked.

<NODE2> NODE

- The BMONDEFS file contains old and new values of <NODE1> and <NODE2> respectively, along with any volume and subvolume changes.
- A BMON Recovery is performed.
- BMON is recovered and old values for <NODE1>, including any volume and subvolumes are automatically amended to those for <NODE2>.
- The recovered BMON includes completed, waiting, held, failed jobs (jobs which had failed prior to the site-swap, or were running when the system crashed and therefore will be in an incomplete state).
- Systems, Applications and Users can now swiftly determine where the schedule and any application changes were up to.

Example TACL results for the above scenario are provided below.

<NODE1> BMON process, $BMON is running and the schedule is in progress.

The BMON <disk>.<subvol>.recovery file is being replicated to the <NODE2> node.

The <NODE1> node crashes and a site-swap is invoked.

On the <NODE2> node, the BMONDEFS file contains the following old and new values:

```
NODE    <NODE1><NODE2>
VOLUME  <DISK1><DISK2>
SUBVOL  APPLONE APPLTWO
```

On the <NODE2> node, the BMON is recovered as follows:

```
ASSIGN DATACONF, <NODE2>.<DISK>.MBATDAT.DATACONF *
RUN <DISK>.MBATOBJ.BMONL / NAME $BMON, NOWAIT /
RUN <DISK>.MBATOBJ.BCOM $BMON
```

*The ASSIGN value references the MultiBatch Database where a copy of the <NODE>.BMON process is configured (perhaps created using the MIGRATOR utility), along with its environmentalised params and values, e.g. <NODE>, <DISK>, subvolumes, conditional params, assigns, defines, etc.

```
BCOM - D07 - (19FEB21) MultiBatch
1>> LOG1 $MSTR (1)
2>> SET BMON NEWRECOVER <DISK>.<SUBVOL>.NEWREC (2)
3>> SYNC IN <disk>.<subvol>.RECOVER , TRANSLATE <DISK>.MBATOBJ.BMONDEFS (3)
```

(1) LOG1 is the MultiBatch Streamer process which converts original text events into tokenised MultiBatch events.

(2) Specify a new BMON Recovery file (**Note**: This file must **not** exist).

(3) Using the BMON SYNC command, reference the original (and replicated) BMON Recovery file which will be used to create the new BMON Recovery file and in the process, amend the old to new values based on the contents stored in the BMONDEFS file.

EMS events will be issued reporting on the status of the schedule, e.g.:

```
18-04-05 14:42:00 <NODE2>.$ MULTIBAT.1.0 000027 OLD RECOVERY FILE =
"<disk>.<subvol>.MBATOBJ.RECOVER",
<NODE>.$BMON 05/04/18, 14:42

18-04-05 14:42:00 <NODE2>.$MSTR MULTIBAT.1.0 000008 ENDJ.**.**.*,{EOD-END-OF-DAY} JOB COMPLETE
!!!, <NODE2>.$BMON 05/04/18, 14:42

18-04-05 14:42:00 <NODE2>.$MSTR *MULTIBAT.1.0 000028 DBUP.AA.01.*,{DBUP-AA-01-STOP-ODBC-DAY}
<<<< UNIT INCOMPLETE, INVESTIGATE !, <NODE2>.$BMON 05/04/18, 14:42

18-04-05 14:42:00 <NODE2>.$MSTR *MULTIBAT.1.0 000028 DBYA.AA.01.*,{DBYA-AA-01-END-OF-Y-DBC}
<<<< UNIT INCOMPLETE, INVESTIGATE !, <NODE2>.$BMON 05/04/18, 14:42

18-04-05 14:42:01 <NODE2>.$MSTR *MULTIBAT.1.0 000029 BMON restarted: Copyright 1990-2010,
<NODE2>.$BMON 05/04/18, 14:42

18-04-05 14:42:01 <NODE2>.$MSTR MULTIBAT.1.0 000001 NEW RECOVERY FILE =
"<NODE2>.<DISK>.MBATOBJ. NEWREC", <NODE2>.$BMON 05/04/18, 14:42
```

Within BCOM, the INFO JOB <name> , DETAIL command will reveal the new configuration, displaying the new node, volume and subvolume changes, e.g.:

```
RUN <DISK>.MBATOBJ.BCOM $BMON ; INFO JOB ENDJ, DETAIL

BCOM - D07 - (19FEB21) MultiBatch
INFO JOB ENDJ, DETAIL
Job - ENDJ, Maxsegs = 001, Numsegs   = 001 Rerun
              = OFF
              SCHEDULE :-
              +  NEXT     SDBJ
                          SDEJ
                          SNEJ
                          SWAJ
                          SXAJ
                          SYAJ

Segment - AA, Maxunits   = 001, Numunits   = 001
              Maxdepends = 000, Numdepends = 000

Unit - 01.1, Cpus = 01:00, Priority = 130, Memory = 032, Name = $ENDJ
              Highpin          - ON
              Object file     - <NODE2>.$SYSTEM.SYSTEM.TACL       (255,222)
              Default         - <NODE2>.<DISK2>.APPLTWO
              Input file      - <NODE2>.<DISK2>.APPLTWO.ENDJOBJ
              Output file     - <NODE2>.$S.#ENDJ
```

```
Assign :-
* AUDIT, <NODE2>.<DISK2>.APPLTWO.AUDIT
* EXCEPT, <NODE2>.<DISK2>.APPLTWO.EXCEPT
* RUNTIME, <NODE2>.<DISK2>.APPLTWO.RUNTIME
* TXN-FILE, <NODE2>.<DISK2>.APPLDAT.TXNFILE
Defines :-
* =MISC_STAT_FILE
 CLASS              MAP
 FILE               <NODE2>.<DISK2>.APPLTWO.STATS
* =MISC-REPT-FILE
 CLASS              MAP
 FILE               <NODE2>.<DISK2>.APPLTWO.REPORT
Param :-
* REPORTS, “TRUE”
* SYSTEM-TYPE, “PRODUCTION”
* OPERATOR-DEBUG-LEVEL, “LOW”
```

BCOM Status commands can be performed to check on the status of jobs, segments and units, e.g. STATUS JOB *, STATUS JOB *, FAILED.

A MULTIBATCH utility called UTRECOV is available where a user can produce a report listing for various states of the schedule. See “Appendix F.12 - UTRECOV” for further details.

Appendix J Troubleshooting

Issue	Comments
Error 409 Issued	BMON is the NonStop process which initiates, monitors and logs the progress of your job workload and has a number of interfaces. As with most NonStop monitor processes this interface uses the message based $RECEIVE mechanism to pass Inter Process Messages (IPMs) from a command interface. The command interfaces to BMON are from: 1) BCOM – the Command Line Interface 2) TRIGGER – when processing jobs, segments and units set up in the Event-Timer scheduler 3) LIBUTIL – when using the programmatic interface to HOLD, RELEASE, START, RESTART, ABORT Jobs, Segments and Units, or RESTART, IF FAILED units. 4) STATSRV – provides BMON status details for the MultiBatch Screen COBOL Interfaces. When these command interfaces communicate with BMON, an initial message is sent to ensure that further messages will be compatible with the BMON version. BMON returns status 409 when the version of messages used by a command interface do not match the BMON version. When BCOM receives this 409 reply from BMON, it will issue the following message via the BCOM command line interface: *** ERROR 515 *** Incompatible releases of BCOM & BMON The BMON interface is only updated for a major release; in this situation, all of BCOM, TRIGGER, LIBUTIL and STATSRV form part of the release. In this way the 409 status shows that there is a configuration issue. Status 409 is commonly generated where customer programs interfacing with BMON using LIBUTIL need to be modified to include the latest delivered LIBUTIL procedures.
CST Screen – Two choices of CPU	When a job runs, then it checks if CPU n is available in Choice 1. If it is not, then a PROCESS_CREATE error 10 is issued and the job is started in Choice 2 CPU; additional log messages are issued informing the user of this activity. If Choice 2 is also unavailable, then the job does not run.

Error 14 occurs during a steps process creation	Any nowaited process creation on NonStop follows the same conventions. Step 1 - creates an empty process Step 2 - opens this process and sends it a startup message and if appropriate, assign and parameter details. If the gap between these two Steps is too long, the new process may time out and ABEND. How long this gap is, is up to the application developer. In the case of the COBOL developer it is out of his control and is 60 seconds. This delay may be caused by poor system performance, or by poor design of the schedule. MultiBatch is coded to trap error 14 and 201 in this circumstance and to recreate the job. It will have 5 attempts before issuing a "repetitive failure" log message.
Why is a "2050" Add Define error issued?	This means that the Define already exists in the BMON Process File Segment (PFS). When BMON starts a job, it loads all the Defines for the job into its PFS and starts the job. The Defines are then propagated to the job, then deleted from the PFS. If a Define already exists in the PFS of the TACL that started the BMON, then it will end up in the PFS of the BMON. If a job then has the same Define name added at the start up time of the job, then this clash results in the 2050 error. Solution: Include SET DEFMODE OFF command in the obey file that starts the BMON process.
Open Process ... file error - 020 occurred when running a unit/step on a remote node. **Is accompanied with MULTIBAT.1.0 87 error, e.g.:** **yyyy-mm-dd hh:mm:ss** **<node> MULTIBAT.1.0 087** **jjjj.ss.uu.s, Open process** **.... file error - 020, \<local node>.<BMON Process Name>**	The BMON process on the <local node> is running a job.segment.unit on a remote node but it fails with MULTIBATCH error 87, along with a Guardian error 20. It is recommended to refer to NonStop Expand Network Mgmt Guide for further details but in this example, the process for the remote unit, has tried to communicate with a five character BMON process name, running on the parent node, e.g. $BMONA. Limit the parent BMON process name to four characters, excluding the $ sign, e.g. $BMON. Alternatively, the process name for the configured step (see CST screen), may also have been configured with five characters. Utilise four characters for processes which run remotely and for the parent BMON process if running remote jobs.
***** ERROR 406 *** You do not have the authority to execute this command**	See 'Configure BMON' Chapter which describes the BMON Security fields. The user ID that is trying to execute the command is determined by these BMON security values. These values help to determine who can execute particular commands, e.g. Start J,S,U, Alter step Flags, Shutdown BMON, Shutdown BMON !

A job has failed to start at expected Start Time	Could be a number of reasons, e.g. job is already completed, has failed, is held. Check status job via BCOM command STATUS JOB <id> , DETAIL and also INFO JOB , DETAIL. Status will show current status job. Info will show if any dependencies exist and if they have completed. Check if associated Event-Timer entries are configured, scheduled, failed. If the job in question is critical to the job flow, configure a CLOCKMON entry to alert the user if the job has failed to start at an expected start time. See Chapter 3 'Configure MultiBatch' and sub-chapter 'Configure Monitor Step' for further details.
Restart of a J,S,U in SOV fails with: **"REJECTED - YOU DO NOT HAVE AUTHORITY TO EXECUTE THIS COMMAND (406)"**	This is the same symptom as described for the *** ERROR 406 *** description above. Check BMON Security settings compared to the Pathway logged on user.
Error 10 Alias Name is issued during Migrator Insert	Migrator is used to Extract, Remove and Insert BMON configurations. During the Insert process, if an Alias Name within a BMON, Job, Segment or Unit already exists, then Guardian error 10 is issued. The database MIGDEFS file can be used to replace Alias Names which exists in the Transit file, to new values. See Appendix B Migrator for further details.
A J,S,U does not exist in the newly built BMON	Check the job, segment, unit 'schedule' configuration in the database, as a J, S, U could have had particular scheduling parameters configured, e.g. Saturday I. If the Unit / Step has been configured to only run on a particular day, or calendar date, then if the Prepare has been run which does not match these configured days, dates, the J,S,U will not be selected. Check the last entry of Prepare VLL for Not Run entries or check the Prepare spooler output.
An Every schedule exists in Event Timer but the job does not run a second time.	The job needs to utilise a GROUPID in order to reset itself back to %000000 in readiness for its next Every run.

What are Rerunnable Jobs? (Use of GROUPID function)	The original BMON schedule was built to run overnight from A to Z, to be closed down, then reloaded for a similar exercise the following night. The FLAGS prevent the schedule from being rerun, without being reloaded. This was deliberate, because executing the same job twice on the same day had potentially disastrous consequences. Now that the product is used for more than just executing a large schedule once a day, its one weakness was processing regular job submissions or parallel instances of the same job. As a concession to a particular customer, the "Rerunnable / GroupId" function was produced. The customer's requirement was to execute a sequence of four programs on an ad hoc basis, the culmination being a printed report. In these circumstances, it was unacceptable to expect them to alter flags or reload the environment after each request. They can now put the jobs that form this mini-schedule into the same GROUP id, using the Configure Job (CJB) screen. The flags will be reset to %000000, when a job completes successfully. **Note:** It is restrictive: ● All jobs must be in the same group id, otherwise it breaks. ● Dependencies can only be at job level and with only one dependency. ● Each job can only contain one segment. ● Each segment can only contain one unit. ● You can only start the first job. ● You cannot start the group if it is already active. • Ensure any dependencies are in the same GROUPID. If the first job is in a GROUP but the next job isn't, then the next job will not start.
Following installation, the Pathway screens hang	Check the Database files are audited - probably a deadlock between the Pathway server and TMF AUDIT not being switched on for the database. See <database>.AUDITON file content to determine what database files should be audited.
Recovered BMON - no EMS events issued	Check if LOG1 or LOG2 has an STREAMER (Streamer) process configured. See Chapter 1, sub-chapter 1.10 Log Files for further details.
Unable to shutdown BMON via BCOM	Check if jobs are still running. Check BMON Security flags *x , y* via BCOM command - INFO BMON. The second class y enables SHUTDOWN BMON to take place but if jobs are running, then this is rejected. See Chapter 3, sub-chapter 3.3 Configure BMON for further details.

My BMON is executing and jobs are running, but why are the Status screens not refreshing?	There could be a number of reasons: The MultiBatch EMS events are extracted by the Watcher process and written to a Status database - is the Watcher process running? Ensure that the Watcher is pointing at the correct database by checking the DATACONF file for the location of the *STAT files. Perform a FUP LISTOPENS on these files because the Watcher holds them open all the time. Check that the BMON has a STREAMER logger process selected as one of its two log files. BCOM $<bmon-name>; INFO BMON. When you start the Watcher you specify how often that you want the database updated in the REFRESH n seconds, and NUM^CHANGES values. Check the obey file, e.g. RUNWAT to ensure that these parameters are set to reasonable values.
My Status database figures are nonsense! Why?	At the beginning of each schedule run and as part of the rebuild, you need to run the STBUILD program which puts the statistics from the last run out to the spooler and resets the numbers for today. Waiting and Configured totals will then be equal and the running, complete and failed values will be 0. If this is not done or the program failed, then the Watcher does not trap this condition. The first unit started will result in –1 from the waiting column and +1 in the running column and the figures will therefore be incorrect.
Following a clean installation, a user cannot logon, as the MultiBatch database does not have any user ID included.	Only a Security Class - ALLFACIL - exists but no Security Profiles. A user needs to be associated with this Security Class. See Chapter 2 Security Management, sub-chapter 2.3 Add First User.
When adding a Unit / Step in the Configure Step screen, I can associate a Guardian user Id with this step without being asked for a password but I need to ensure that a password is utilised.	Check the Security Class associated with the MultiBatch logon user id (see Security Profile) and that the CST02 (P) facility is set to P.
What is a 'Unit Skipped Message'?	MultiBatch provides a comprehensive scheduling facility. In the Job sequence A->B->C, if B does not run at the weekend, it will not be loaded on Saturday and Sunday and A and C will be linked together. Before the existence of MultiBatch Pathway and database, the only way to achieve this was to load all 3 jobs, but manually mark "B" as not running. In this instance it would be "skipped" and the log message would be created. The 'Skip' facility is part of the Conditional Parameters function. Based on Jobs A->B->C, where B does not run at a weekend, then: On the "Conditional" overlay of the Configure Step screen, you would create an entry "WEEKEND" and "Y" for step "B". In the LOCAL parameter database (in screen PME), you would set the WEEKEND parameter to "TRUE" or "FALSE" depending on the day. At BUILD time, BCOM would read this database and assess whether WEEKEND was true or not and therefore turn on/off the run flag for the step. If the flag was off, the job would be loaded but skipped when its execution time arrived.

Why are there two versions of BMON - BMONL & BMONU?	There is a facility available only to MultiBatch users - run jobs as specific users. This is a powerful function that eases Security issues. To do this, BMON uses a procedure call and consequently as it contains privileged code, BMON needs to be licensed. Some installations are wary about installing third party licensed code, so BMONU (unlicensed) without the procedure call is provided and BMONL (licensed) which does have procedure call. It is the Users decision which one they choose.
Why am I periodically getting "Error 14" from the Watcher process when it connects to a PATHWAY TCP?	When a User navigates to a Status screen, an entry is made on the LOGGEDON database file. As the Watcher reaches "REFRESH" or "NUM^CHANGES" and dumps any changes to the Status database, it consults the LOGGEDON file and sends an unsolicited message to the PATHWAY TCP supporting the terminal, so that the latest numbers can be retrieved and displayed. If the PATHWAY terminal is aborted before it has had a chance to remove itself from the LOGGEDON file, then the Watcher will attempt to send a message to a non-existent terminal. If the TCP is still running, then this is ok as an error is returned for the non-existent terminal and Watcher deletes it from LOGGEDON. If the TCP has gone then the Error 14 appears. If the PATHWAY has been closed down and brought back, but because the TCP has no hard coded process name it reappears as an $X, $Y or $Z name, then the same problem will persist. It is because of this problem that we insist on the MultiBatch PATHWAY TCP process being named.
What is the CLEARDOWN option used for in STBUILD?	STBUILD is used to reset the status figures from their representation of the close of the previous run of the schedule to the way that they will look at the beginning of the next schedule. The old figures are written away to a report file, deleted from the BMONSTAT, JOBSTAT, SEGSTAT and STEPSTAT files and replaced with new figures from the Loadlist file. If you just want to archive and delete the old figures because the schedule is not to be reloaded for another week, then you can run STBUILD in CLEARDOWN mode.
Why was my job not started by Event Timer?	Sometimes the Calendar and Trigger processes are not running. Calendar watches the clock and at the appropriate time raises an EMS event - this is intercepted by Trigger which then communicates with the BMON scheduler. Facilities exist within Event Timer to schedule for certain days. It may be that the job is not meant to run today. Check the "List Next Run" Event-Timer PATHWAY screen or use the PSCHED print utility to find out when it should run. When you make a change to the Schedule database, the Calendar process needs to know that the database has been updated to see if it needs to start a job earlier than expected. The EVENT-TIMER PATHWAY knows the location of the Calendar process because of an EVENT-TIMER Server parameter. If this is incorrect, then Calendar may well miss its exact cue. It looks by default once every 30 minutes just in case.

Can a user utilise an Alias user id to logon?	Yes, but the underlying Group.User ID needs to be configured in the Security Class facility. For example, super_ops is the alias logon but the underlying Guardian Group.User Id of super.operator needs to be configured in MultiBatch.
Can an application communicate with MultiBatch via library functions?	The ability to communicate with MultiBatch and Event-Timer is available via the supplied LIBUTIL and ETIMLIB libraries. See Appendix F.15: Programmatic Interfaces.
AUDREPT results contain information from other MultiBatch environments.	Before running the AUDREPT utility, execute a CLEAR ASSIGN DATACONF to clear any assignments for any other MultiBatch environments.

Appendix K BMON Protocol

BMON supports a set of messages that allows the process associated with a step to communicate with it to store the unit's status. BMON maintains the status along with an optional message that can viewed as part of the information returned by BCOM when providing a unit's status. This mechanism is known as BMON Protocol.

The BMON Protocol is only supported for Guardian steps; it is not supported for OSS steps. The CST (Maintain STEP Configuration) screen has a field to define that BMON Protocol is to be used for the step.

The functionality is not supported for the 2nd step of two step units, only step 1 can control the status of the unit.

BMON Protocol provides a tight integration between the program and BMON where:

a. specific bits in the step status flags are utilised, these provide extra status information for the unit

b. the process itself can define and report when it is complete

c. the process can take advantage of BMON's fault tolerance to log progress by reporting its status, on restart it can
retrieve the last logged status

d. when requesting status details using BCOM the extra details are provided.

The communication is via standard inter-process communication and requires the step process to open BMON. BMON will be the process's creator and the process id to be opened can be obtained using the MOM procedure.

Highlighted below are the step status flags that relate specifically to BMON Protocol. Note that the opening and closing of BMON for the purposes of the Protocol sets bits 5 and 6 respectively in the status flags.

0 Running

1 Newprocess called

2 Newprocess message received

3 Startup message written

4 Aborted by operator

5 Open message received – BMON is currently opened by the process

6 Close message received – BMON was open but is now closed by the process

7 CPU failure

8 Abend message received

9 Stop message received

10 BMON Protocol in use - SET STEP PROTOCOL ON command used as part of step set up

11 Initialise message received – process has informed BMON that it has started

12 Param file requested – process has requested its param file

13 INSPECT ON

14 Not used

15 STEP complete

See A.9 Miscellaneous Commands for a detailed description of the status flags.

Examples of the coding required to implement the BMON Protocol can be provided on request.

The COBOL message structure used to communicate with BMON is as follows:

```
01  BMON-PARAM.
  03 B-STATUS                     PIC 999 COMP.
    88  STARTUP                   VALUE 100.
    88  INITIALISED               VALUE 101.
    88  UPDATE-STATUS             VALUE 102.
    88  COMPLETED                 VALUE 199.
```

```
   88  STOP                     VALUE 299.

 03 B-MESSAGE.
   05 B-PROCESS-STATUS          PIC 999 COMP.
   05 B-MESSTEXT                PIC X(78).

 03 B-REPLY REDEFINES B-MESSAGE.
   05 B-PARAM-FILE              PIC X(34).
   05 B-LAST-PROCESS-STATUS     PIC 999 COMP.
   05 FILLER                    PIC X(44).
```

This single message is used for both requests to BMON and replies from BMON.

The first four B-STATUS values are provided by a step process for requests the next two are used by BMON for replies.

B-MESSAGE values are provided by any message from the step process other than STARTUP. Values will update the internal values held by BMON and reported by BCOM. If B-PROCESS-STATUS is zero, then the value held for the step by BMON will not be updated. Likewise, if B-MESSTEXT is spaces, then the value held for the step by BMON will not be updated.

Before any processing takes place for a unit B-PROCESS-STATUS is zero and B-MESSTEXT is spaces.

Generally, replies are two bytes to report RESUME or STOP, exceptionally the reply is 38 bytes for STARTUP where B-REPLY fields are populated.

Initially or for a restart the user process should send STARTUP. The process's in-file is returned in B-PARAM-FILE. B-LAST-PROCESS-STATUS will be returned with the last B-STATUS held by BMON for the step, this is to facilitate a potential restart. The B-STATUS reply will be RESUME and Flag bit 12 will be set.

The in-file is supplied as a mechanism for the process to store context information about its progress, if required.

If the step process has an initialisation phase, then INITIALISED can be used to log completion of the phase. BMON will log a "unit initialised" message. The B-STATUS reply will be RESUME and Flag bit 11 will be set.

If B-MESSAGE is to be updated, then the step process can send UPDATE-STATUS. B-MESSTEXT will be written to the BMON log if it is not spaces. The B-STATUS reply will be RESUME.

When a step process is complete it can send COMPLETED and BMON will update flag bit 15. For non BMON Protocol processes BMON uses the system stop message from the process for the same purpose.   If a message is provided in B-MESSTEXT this will be appended to a unit complete message logged by BMON. The B-STATUS reply will be STOP.

Appendix L SBMON Obey File

An SBMON obey file is supplied in the MultiBatch Object subvolume as an example TACL obey file for starting BMON.

```
STOP <BmonName>                                                         (1)
ASSIGN DATACONF,<DAT-SVOL>.DATACONF                                     (2)
VOLUME <DATA-SVOL>
                                                                        (3)
[#IF [#FILEINFO /EXISTENCE/ ORECOVER]
|THEN|
    PURGE ORECOVER
    PURGE ORECOVE0
]

[#IF [#FILEINFO /EXISTENCE/ NRECOVER]
|THEN|
    FUP ALTER NRECOVER,  NO AUDIT
    FUP ALTER NRECOVE0,  NO AUDIT
    FUP RENAME NRECOVER, ORECOVER
    FUP RENAME NRECOVE0, ORECOVE0
    FUP ALTER ORECOVER,  ALTFILE (0,ORECOVE0)
]

VOLUME <OBJ-SVOL>

RUN STBUILD/IN <DAT-SVOL>.DATACONF, OUT $S.#ST/<Prepare-Name>           (4)
RUN BMONU / NAME <BmonName>, CPU 1, NOWAIT/                             (5)
DELAY 2 SECS
RUN BCOM <BmonName>; BUILD <PrepareName>                                (6)
```

1) Stops the BMON process (this is an optional command). A user can use the BCOM SHUTDOWN command instead.
2) References the DATACONF file where the PREPCONF file containing the <Prepare-Name> is stored.
3) Volume to the location of the BMON Recovery file and purge and rename the current recovery files. See CBM Recovery fileset. Amend the VOLUME command to reference the subvolume where your recovery files are located.
4) STBUILD updates the Status database with 'today's' run of the BMON. The Status database figures will be set to the number of jobs, segments and units identified during the Prepare process.
5) Start the BMON using the process name configured in the CBM screen. This obey file is utilising BMONU. If required, amend to use BMONL
6) BCOM communicates with the BMON process and builds the schedule

Note 1: *The <PREPARE-NAME> must belong to the named BMON process.*

Note 2: *Following the build of a BMON, the DATACONF assign is stored in the running BMON process. This means thatthe assign does not need to be referenced within any TACL macros, obey files.*

Appendix M MultiBatch Delivered Files

M.1 Files in the Object Subvolume

ALRECON
The ALIASMAP file reconciliation program. See Appendix F MultiBatch Utilities.

AUDLOGHK
The AUDLOG housekeeping program. See Appendix F MultiBatch Utilities.

AUDREPT
The Audit report program. See Appendix F MultiBatch Utilities.

AUDSRV
The program associated with the SRV-AUDSRV Pathway server. The program handles processing from the List Audit Log (ALL) screen.

BCFORMAT
An edit file used to specify the fields that will be returned by BCOM INFO commands. See Appendix F MultiBatch Utilities.

BCNF0998
A configuration file used by the STREAMER program to convert general text event raised by BMON into specific events that allow the WATCHER and TRIGGER programs to maintain the status database.

BCOM
The MultiBatch command interface program that converses with the MultiBatch monitor BMON.

BMONHELP/HLPD
Help text files used when returning details for BCOM help commands.

BMONL/U
The MultiBatch monitor program. BMONL additionally contains privileged code for switching user ids and therefore must be licensed.

CALENDAR
As part of Event Timer this background program monitors the configured schedule and raises events to be acted upon by the TRIGGER program.

CLOCKMON
The background program that monitors the step start times set up on the Configure Monitor Step (CMS) screen. Appropriate events are raised if start times are passed.

CUSTSRV
The program associated with the SRV-CURR-STAT Pathway server. The program handles processing from the Current Status (SOC) screen.

ERRLOGHK
The ERRLOG housekeeping program. See Appendix F MultiBatch Utilities.

ERRSRV
The program associated with the SRV-ERRSRV Pathway server. The program handles processing from the List Error Log (ERL) and Operations Diary(OPD) screens.

ETIMLIB/5/8
The event timer library containing a procedure that can be bound into end user programs to amend the SCHEDULE file. See Appendix F MultiBatch Utilities.

ETIMSRV
The program associated with the EVENT-TIMER Pathway server. See Chapter 8 for a full description of the functionality of Event Timer (EME) and the various screens and parameters this server supports.

ETPRINT
The ETPRINT utility program is used to report on the contents of the Event Timer SCHEDULE file. See Appendix F MultiBatch Utilities.

GUISTAT
This program is associated with the SRV-STATUS-GUI Pathway server. It returns details for the beta version of the MultiBatch Operations GUI.

GETPRMPT
This program can be run from a unit TACL script to accept the response to a prompt as part of the overall MultiBatch job prompting mechanism. See Appendix F MultiBatch Utilities.

HARNCOPY
This edit file contains COBOL copy code layouts used by the MHARNESS program. These files are examples that show how to call LIBUTIL procedures; these provide a programmatic interface to BMON. See Appendix F MultiBatch Utilities.

HELPMBAT
This file contains the text returned when using the MultiBatch Screens’ online help facility (F16).

HELPSRV
The program associated with the SYSTEM-HELP Pathway server. This server returns text from the HELPMBAT file when using the MultiBatch Screens’ online help facility(F16).

INSTALL
This program is used as part of the MultiBatch installation process to tailor files according the parameters set when running the MultiBatch installation macro. See the MultiBatch Installation Guide.

INTEGCAL
This program can be used to change the location of the Event Timer SCHEDULE file in a contingency scenario. See Chapter 8 – Event Timer.

INTEGHOL
This program is used to ensure consistency in the Event Timer HOLIDAY file in a contingency scenario. See Chapter 8 – Event Timer.

INTEGSHD
This program is used to ensure consistency in the Event Timer SCHEDULE file in a contingency scenario. See Chapter 8 – Event Timer.

LIBUTIL/5/8
Libraries that contain procedures that form the programmatic interface to BMON. See Appendix F MultiBatch Utilities.

MBATCOM
The MultiBatch command line interface that converses with the MultiBatch CALENDAR, CLOCKMON, STREAMER, TRIGGER and WATCHER processes. See Appendix F MultiBatch Utilities.

NWFILS
This EMS Filter edit file is compiled to NWFILT and used by the WATCHER program to select events associated with BMON processing that are then used to update the status database. See Chapter 5 Status Monitoring.

NWFILT
The compiled version of NWFILS.

NHELP

This file contains help text returned by MBATCOM HELP commands.

NHELPD
This file contains detailed help text returned by MBATCOM HELP detail commands.

MHARNESS
This edit file contains example COBOL code that shows the use of LIBUTIL procedures. LIBUTIL provides a programmatic interface to BMON. See Appendix F MultiBatch Utilities.

MIGRATOR
This program allows the relocation of data between MultiBatch database environments, without the need for update of data via the MultiBatch Pathway. See Appendix B Migrator.

MIGRCONP
This is the MultiBatch conditional parameter migration program. The program is used to allow the GLOBAL and LOCAL files to be moved from one environment to another. See Appendix F MultiBatch Utilities.

NOBJCOD/DIR/SYM
These are the SCREEN COBOL object libraries for the MultiBatch Pathway SCREENS.

NTFILS
This EMS Filter edit file is compiled to NTFILT and used by the TRIGGER program to select events associated with BMON processing that are then used to manage Event Timer jobs. See Chapter 8 – Event Timer.

NTFILT
The compiled version of NTFILS.

PINTCOPY
This edit file contains COBOL copy code layouts used by the PINTEG program. These files are examples that show how to call the ETIMLIB procedure to update the SCHEDULE file. See Appendix F MultiBatch Utilities.

PINTEG
This edit file contains example COBOL code that shows how to call the ETIMLIB procedure to update the SCHEDULE file. See Appendix F MultiBatch Utilities.

PREPARE
This program is used in the Prepare Phase, it reads the configuration files and generates BCOM commands to set up the schedule in BMON. See Chapter 4 Scheduling – Prepare Phase.

PSCHED
This utility program provides a report of the contents of the Event Timer SCHEDULE file. See Appendix F MultiBatch Utilities.

RESCHDST
This utility program adjusts job start times, times may be offset from the current values or a new time can be provided. The SWITCHST edit file provides example commands to adjust start times. See Appendix F MultiBatch Utilities.

RESCHED
This utility program resets the Event Timer schedule next run times after a system outage. See Appendix F MultiBatch Utilities.

RUNRST
This edit file provides an example set of commands to run the RESCHDST utility.

SBMON
This edit file provides an example start-up script for BMON including loading the status database. The file should be tailored to match the user’s MultiBatch system as part of the installation process. See the MultiBatch Installation Guide.

SRVADMN

The program associated with the SRV-ADMIN Pathway server. The program handles processing from administration related screens, for example Maintain Data File Configuration(DCC) and Maintain Owner File(OWC). The server also provides logon functionality for the Pathway screen system.

SRVCONF
The program associated with the SRV-CONFIG Pathway server. The program handles input from the screens that are used to set up schedule tasks, for example Maintain STEP Configuration(CST).

SRVDEF
The program associated with the SRV-DEFINE Pathway server. The program handles input from the Maintain Define Class (CDE), Maintain Defines (CDF), List Define Classes (CDL) and List Defines (CLD) screens.

SRVOSS
The program associated with the SRV-CONFIG-OSS Pathway server. The program handles input from screens associated with OSS steps. For example, Maintain OSS STEP Configuration(COS).

SRVPAR
The program associated with the SRV-PARAM Pathway server. The program handles requests associated with the maintenance of scheduling calendars and prepare configurations.

SRVPARS
The program associated with the SRV-CONDITPARAM Pathway server. The program is responsible for the maintenance of the LOCAL and GLOBAL conditional parameter files.

SRVPRMPT
The program associated with the SRV-PROMPT Pathway server. The program is responsible for the maintenance of prompt related data. See the GETPRMPT facility in Appendix F MultiBatch Utilities.

SRVSTAT/U
These programs are associated with the SRV-STATUS Pathway server. The programs are responsible for retrieving data to be displayed on screens associated with the status database. The programs communicate with BMON to alter entity status and SRVSTAT contains privileged code to change the process user id for this purpose. SRVSTATU does not contain the privileged code and does not attempt to change the process user id. See Chapter 5 Status Monitoring.

STATSRV
The program associated with the SRV-BMON-DETAIL Pathway server. The server retrieves entity information for various screens associated with the status database.

STBUILD
This utility program takes output from the PREPARE programs and populates the status database. See Chapter 4 Scheduling – Prepare Phase

STREAMER
The background program that processes events raised by BMON so that they can be processed by the WATCHER program. See Chapter 5 – Status Monitoring.

SWITCHST
This edit file provides example commands for the RESCHDST utility program.

TRIGGER
This background program handles events associated with Event Timer jobs. See Chapter 8 – Event Timer.

UTADDUSR
This utility program adds a user to the MultiBatch database. It is used during the installation process to add the first user, see the MultiBatch Installation Guide and Chapter 2 – Security Management.

UTPARHK
The UTPARHK Utility allows obsolete LOCAL/GLOBAL Conditional parameters to be reported / deleted from the MultiBatch configuration database. See Appendix F MultiBatch Utilities.

UTPCALHK
This utility program provides housekeeping functionality for the scheduling calendars held in the PARAMCAL file. For example, the program can be used to reports calendars where there are no more future dates for running tasks. See Appendix F MultiBatch Utilities.

UTRECOV
This utility program reads and provides a status report from a BMON recovery file. The program could be used for a site-swap where a BMON has been recovered using a replicated recovery file and you wish to obtain a listing showing the BMON schedule status. See Appendix F MultiBatch Utilities.

UTUSERHK
This utility program identifies obsolete Guardian User Ids; these can then be removed from the MultiBatch database using the appropriate screens. See Appendix F MultiBatch Utilities.

WATCHER
This background program processes events, as raised by the STREAMER program on behalf of BMON, to update the schedule status database. The schedule status database can be viewed from various MultiBatch screens accessed from the Status Menu (SOM) screen. See Chapter 5 – Status Monitoring.

WILDCARD
The program associated with the SRV-WILDCARD Pathway server. The server provides details for the List ALIASMAP / Wild Card (CLA) screen; the screen provides fast access to MultiBatch entity specific screens via their alias names.

M.2 Files in the Database Subvolume

ALIASMAP/ALIASMA0
This file and index, map MultiBatch entity names, for example those for jobs, segments and steps, to a longer and more meaningful alias name. MultiBatch provides facilities to use the alias name as a look-up, see List Alias (CLA) in Chapter 3 of this document.

ALTFILE
This FUP obey file sets the alternate key files for the MultiBatch database.

ASSCLASS
This file maps ASSIGN definitions to a MultiBatch Assign Class name. The name can then be used when defining Guardian steps to associate the assigns with a number of steps. See Assign Class Definition (CAS) and List Assign Class (CAL) in Chapter 3 of this document. Assign classes can be referenced when setting up a Guardian step; see the Assign overlay on the CST screen in Chapter 3.

AUDDET
This file contains details of changes to the MultiBatch configuration database. See the AUDREPT utility in Appendix F. Also see Chapter 7 View Audit Logs.

AUDITON/AUDITOFF
These FUP obey files are used to set AUDIT and NO AUDIT respectively on those files in the MultiBatch database that are protected by TMF.

AUDLOG/AUDLOG0
This file and index contain header details for changes to the MultiBatch configuration database. See Chapter 7 View Audit Logs.

BMONCONF/BMONCON0
This file and index contain BMON configuration data. See Maintain BMON Configuration Screen (CBM) in Chapter 3.

BMONSTAT/BMONSTA0
This file and index contain information on the progress of a BMON schedule at the BMON level as reported on the status screens; see Chapter 5 – Status Monitoring. These files are updated by the WATCHER process; see Chapter 5 – Status Monitoring WATCHER environment.

CALFIL
This file is provided as an example calendar file to be used with the Event timer; see Chapter 8 Event Timer.

CURRDEP
This file is created during a BMON schedule build and contains the schedules job and segment dependencies. See Chapter 4 Scheduling – Prepare Phase. The file is used by MultiBatch schedule status reporting; see Chapter 5 – Status Monitoring.

DATACONF
This file allows MultiBatch programs to find the location and name of the various database files. See Appendix C Datafile Configuration. This file is updated as part of the installation process.

DEFCONF/DEFCONF0
This file contains defines definitions as created on the Maintain Defines (CDF) screen. The defines can be referenced when setting up a step on either the Maintain STEP Configuration(CST) or Maintain OSS STEP Configuration (COS) screens.

DEFCLASS
This file contains define class definitions as created on the Maintain Define Class (CDE) screen. The classes can be referenced when setting up a step on either the Maintain STEP Configuration (CST) or Maintain OSS STEP Configuration (COS) screens.

ERRLOG/ERRLOG0
This file and index are written by the WATCHER process when an error condition is reported by BMON, see Chapter 5 – Status Monitoring WATCHER environment. See Chapter 6 Error Resolution for the

view this data.

GLOBAL
This file contains the Conditional Processing global parameter information. See Appendix D Conditional Processing.

HOLIDAY
This file contains the holiday information for the Event Timer AT Schedule(EAT) functionality. See Chapter 8 Event Timer.

JOBCONF/JOBCONF0
This file and index contain the job information entered on the Maintain JOB Configuration (CJB) screen. See Chapter 3 Configure MultiBatch.

JOBDEP/JOBDEP0
This file and index contain the job dependencies entered on the Maintain JOB Configuration (CJB) dependencies screen overlay. See Chapter 3 Configure MultiBatch.

JOBSTAT/JOBSTAT0
This file and index written during prepare processing hold information about the status of scheduled jobs; see Chapter 4 Scheduling – Prepare Phase. Whilst a schedule is running and job statuses change, JOBSTAT is updated by WATCHER; see Chapter 5 – Status Monitoring.

LOCAL/LOCAL0
This file and index contain contains the Conditional Processing local parameter information. See Appendix D Conditional Processing.

LOGGEDON
This file holds details of which Pathway terminals are currently displaying Status Monitoring screens. WATCHER uses this information to initiate screen updates when a schedule progresses.

MIGDEFS
This edit file is an example of a translation file used by the MIGRATOR program. See Appendix B Migrator.

OBMNSTAT/OJOBSTAT/OSEGSTAT/OSTPSTAT
This files are a historical record of the data in the equivalent status files. They are populated by the STBUILD program, see Chapter 4 Scheduling – Prepare Phase.

PARAMCAL/PARAMCA0
This file and index contain the scheduling holiday information where a set of holiday dates can be associated with a with parameter name. See Calendar Maintenance Menu (PAM) and Chapter 4 Scheduling.

PARCLASS
This file contains parameter class definitions as created on the Param Class Definition (CPA) screen. A class can contain a number of parameter name/value pairs and the class name can be referenced when setting up a step on the Maintain STEP Configuration(CST) screen.

PREPCONF
This file contains the data entered on the Maintain Prepare Configuration (PRC) screen. See Chapter 4 Scheduling – Prepare Phase.

PRMCLASS
This file contains the data entered on the Maintain Prompt Class (CPC) screen. A set of prompts from a class can be associated with step by giving the prompt class name on the Maintain Prompt (CPM) screen. See Appendix F.7 GETPRMPT

SCHEDULE/SCHEDUL0
This file and index contain the AT, EVERY and CRONTAB scheduling details for the Event Timer sub-system. See Chapter 8 Event Timer.

SECCLASS
This file defines for a named class the access rights to each of the MultiBatch facilities. Security classes are

ALLFACIL class to allow initial usage of the MultiBatch system. See Chapter 2 Security Management.

SECPROFL
The file defines the security classes that are allocated to a MultiBatch user. See Chapter 2 Security Management.

SEGCONF/SEGCONF0
This file and index contain the job information entered on Maintain SEGMENT Configuration (CSE). See Chapter 3 Configure MultiBatch.

SEGDEP0/SEGDEP
This file and index contain the segment dependencies entered on the Maintain SEGMENT Configuration (CSE) dependencies screen overlay. See Chapter 3 Configure MultiBatch.

SEGSTAT/SEGSTAT0
This file and index written during prepare processing hold information about the status of scheduled segment; see Chapter 4 Scheduling – Prepare Phase. Whilst a schedule is running and segment statuses change, SEGSTAT is updated by WATCHER; see Chapter 5 – Status Monitoring.

STEPASS
The file contains the ASSIGN definitions as set up on the Maintain STEP Configuration(CST) ASSIGN overlay. See Chapter 3 Configure MultiBatch.

STEPCONF/STEPCON0
This file and index contain the step information entered on the Maintain STEP Configuration(CST) and Maintain OSS STEP Configuration(COS) screens. See Chapter 3 Configure MultiBatch.

STEPDEF
The file contains the DEFINE definitions as set up on the Maintain STEP Configuration(CST) and Maintain OSS STEP Configuration(COS) screens DEFINE overlays. See Chapter 3 Configure MultiBatch. These defines reference definitions in the **DEFCONF** file.

STEPMON
The file contains the data enter on the Configure Monitor Step (CMS) screen and used by the CLOCKMON process. See Chapter 5 Start Time Monitoring (CLOCKMON) and Chapter 3 Configure MultiBatch.

STEPOSE
The file contains the OSS Environment definitions as set up on the Maintain OSS STEP Configuration(COS) screen Environment overlays. See Chapter 3 Configure MultiBatch.

STEPOSEC
The file contains the OSS Environment Class definitions as set up on the Maintain Environment Class (COE) screen. These are referenced on the Maintain OSS STEP Configuration(COS) Environment Class overlay. See Chapter 3 Configure MultiBatch.

STEPOSF
The file contains the OSS FDs as set up on the Maintain OSS STEP Configuration(COS) screen FD overlays. See Chapter 3 Configure MultiBatch.

STEPOSFC
The file contains the OSS FDs as set up on the Maintain File Descriptor Class (COF) screen. These are referenced on the Maintain OSS STEP Configuration(COS) FD Class overlay. See Chapter 3 Configure MultiBatch.

STEPOSS
This file contains the OSS step specific data that is in addition to that held on **STEPCONF**. This data is entered on the Maintain OSS STEP Configuration(COS) FD Class overlay. See Chapter 3 Configure MultiBatch.

STEPPAR
The file contains the PARAMETER definitions as set up on the Maintain STEP Configuration(CST) PARAM overlay. See Chapter 3 Configure MultiBatch.

STEPPROM
This file contains prompt details as set up on the Maintain Prompt (CPM) screen. See the GETPRMPT facility in Appendix F MultiBatch Utilities.

STEPSTAT/STEPSTA0
This file and index written during prepare processing hold information about the status of scheduled step, see Chapter 4 Scheduling – Prepare Phase. Whilst a schedule is running and step statuses change, STEPSTAT is updated by WATCHER; see Chapter 5 – Status Monitoring.

STPROMPT/STPROMP0
This file and index contain prompt status details as updated by the GETPRMPT functionality. See the GETPRMPT facility in Appendix F MultiBatch Utilities.

SYSOWNER/SYSOWNE0
This file and index contain ownership details for a BMON. See Chapter 2 Security Management.