

# NetBatch Management Programming Manual

## Abstract

This manual describes the Distributed Systems Management (DSM) programmatic interfaces (commands, responses, and event messages in Subsystem Programmatic Interface [SPI] format) to the NetBatch subsystem.

## Product Version

NetBatch D30.00

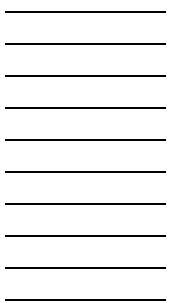
## Supported Release Version Updates (RVUs)

This publication supports D20.00 and all subsequent D-series RVUs, G02.00 and all subsequent G-series RVUs, and H06.03 and all subsequent H-series RVUs until otherwise indicated by its replacement publication.

| Part Number | Published      |
|-------------|----------------|
| 522462-003  | September 2005 |

## Document History

| Part Number | Product Version | Published      |
|-------------|-----------------|----------------|
| 522462-001  | NetBatch D30    | February 2002  |
| 522462-002  | NetBatch D30    | May 2002       |
| 522462-003  | NetBatch D30    | September 2005 |



# NetBatch Management Programming Manual

|                          |                       |                          |                         |                        |
|--------------------------|-----------------------|--------------------------|-------------------------|------------------------|
| <a href="#">Glossary</a> | <a href="#">Index</a> | <a href="#">Examples</a> | <a href="#">Figures</a> | <a href="#">Tables</a> |
|--------------------------|-----------------------|--------------------------|-------------------------|------------------------|

- [What's New in This Manual](#) xi
  - [Manual Information](#) xi
  - [New and Changed Information](#) xi
  - [Changes to the G06.16 Manual](#) xi
  - [Changes to the G06.15 Manual](#) xii
  - [Previous Changes](#) xii
- [About This Manual](#) xvii
  - [Audience](#) xvii
  - [Prerequisites](#) xvii
  - [Organization](#) xvii
  - [Further Reading](#) xviii
  - [Notation Conventions](#) xix
  - [Abbreviations](#) xxiii

## 1. Introduction

- [NetBatch Subsystem Description](#) 1-2
  - [NETBATCH](#) 1-2
  - [BATCHCOM](#) 1-2
  - [BATCHCAL](#) 1-2
  - [NBEXEC](#) 1-2
- [NetBatch Programmatic Interfaces](#) 1-3
- [Objects Managed by the NetBatch Subsystem](#) 1-5
- [Supported Object-Management Functions](#) 1-5
- [Interactive Commands and Corresponding Programmatic Commands](#) 1-6

## 2. Communicating With the NetBatch Subsystem

- [Steps in Communicating With the NetBatch Subsystem](#) 2-1
- [Starting a Scheduler](#) 2-2
  - [Starting a Scheduler Interactively](#) 2-2
  - [Starting a Scheduler Programmatically](#) 2-3
- [Opening a Scheduler](#) 2-3
  - [Opening a Scheduler Interactively](#) 2-3

## **2. Communicating With the NetBatch Subsystem (continued)**

- [Opening a Scheduler Programmatically](#) 2-4
- [Closing a Scheduler](#) 2-5
- [Stopping a Scheduler](#) 2-6
- [Running an EMS Consumer Distributor](#) 2-6

## **3. SPI Programming Considerations for the NetBatch Subsystem**

- [Definition Files](#) 3-2
- [Event-Message Template](#) 3-3
- [Naming Guidelines for Applications](#) 3-3
- [Message Elements for the NetBatch Subsystem](#) 3-4
  - [Commands](#) 3-4
  - [Object Types](#) 3-4
  - [Event Numbers](#) 3-4
  - [Subjects of Event Messages](#) 3-5
  - [Data Lists and Error Lists](#) 3-5
  - [Using SPI to Build Commands and Decode Responses](#) 3-5
  - [Building and Sending a Command Message](#) 3-5
  - [Discontinuing a Command in Progress](#) 3-6
  - [Receiving and Decoding a Response Message](#) 3-7
  - [Error Handling](#) 3-7
  - [Version Compatibility](#) 3-8
  - [Security](#) 3-8
  - [File Names](#) 3-8
  - [Displaying SPI Traffic](#) 3-8
  - [Retrieving and Decoding Event Messages](#) 3-13
  - [Critical Events](#) 3-14
  - [Filters](#) 3-14

## **4. Common Definitions**

- [SPI Standard Definitions](#) 4-2
  - [Information About SPI Standard Definitions Specific to the NetBatch Subsystem](#) 4-4
- [EMS Standard Definitions](#) 4-6
  - [NetBatch-Specific Information About EMS Standard Definitions](#) 4-6
- [NetBatch Definitions](#) 4-8
  - [NetBatch Buffer Declarations](#) 4-12
  - [Private Token and Field Types](#) 4-13

## **4. Common Definitions (continued)**

[Predefined Token and Field Values](#) 4-35

[Simple Tokens](#) 4-40

[Extensible Structured Tokens](#) 4-49

## **5. Commands and Responses**

[Available Commands](#) 5-2

[Symbolic Names of Commands and Objects](#) 5-3

[Sensitive and Nonsensitive Commands](#) 5-5

[Wild-Card Characters](#) 5-8

[Characters Matched by Wild-Card Characters](#) 5-9

[Command Descriptions](#) 5-9

[ABORT SCHEDULER Command](#) 5-9

[ACTIVATE JOB Command](#) 5-10

[ADD ATTACHMENT-SET Command](#) 5-12

[ADD CLASS Command](#) 5-14

[ADD EXECUTOR Command](#) 5-15

[ADD SCHEDULER Command](#) 5-16

[ALTER ATTACHMENT-SET Command](#) 5-18

[ALTER CLASS Command](#) 5-20

[ALTER EXECUTOR Command](#) 5-21

[ALTER JOB Command](#) 5-22

[ALTER SCHEDULER Command](#) 5-26

[DELETE ATTACHMENT-SET Command](#) 5-28

[DELETE CLASS Command](#) 5-30

[DELETE EXECUTOR Command](#) 5-31

[DELETE JOB Command](#) 5-32

[GETVERSION Command](#) 5-34

[INFO ATTACHMENT-SET Command](#) 5-35

[INFO CLASS Command](#) 5-37

[INFO EXECUTOR Command](#) 5-38

[INFO JOB Command](#) 5-39

[INFO SCHEDULER Command](#) 5-42

[RELEASE JOB Command](#) 5-44

[RUNNEXT JOB Command](#) 5-46

[RUNNOW JOB Command](#) 5-48

[SHUTDOWN SCHEDULER Command](#) 5-50

[START EXECUTOR Command](#) 5-51

[START SCHEDULER Command](#) 5-52

## **5. Commands and Responses (continued)**

|   |      |
|---|------|
| <a href="#">STATUS ATTACHMENT-SET Command</a> | 5-53 |
| <a href="#">STATUS EXECUTOR Command</a>       | 5-55 |
| <a href="#">STATUS JOB Command</a>            | 5-56 |
| <a href="#">STATUS SCHEDULER Command</a>      | 5-58 |
| <a href="#">STOP EXECUTOR Command</a>         | 5-60 |
| <a href="#">STOP JOB Command</a>              | 5-61 |
| <a href="#">SUBMIT JOB Command</a>            | 5-63 |
| <a href="#">SUSPEND JOB Command</a>           | 5-67 |
| <a href="#">SWITCHCPU SCHEDULER Command</a>   | 5-69 |
| <a href="#">SWITCHLOG SCHEDULER Command</a>   | 5-70 |

## **6. Event Messages**

|   |     |
|---|-----|
| <a href="#">The NetBatch EMS Interface</a>                                      | 6-1 |
| <a href="#">Event Messages Issued by the NetBatch Subsystem</a>                 | 6-1 |
| <a href="#">Critical Events</a>   | 6-1 |
| <a href="#">Enabling and Disabling NetBatch Event-Message Generation</a>        | 6-3 |
| <a href="#">How EMS Handles NetBatch Event Messages</a>                         | 6-3 |
| <a href="#">Creating an EMS Filter</a>  | 6-3 |
| <a href="#">Step 1: Decide on Scheduler Event Messages</a>                      | 6-4 |
| <a href="#">Step 2: Create EDIT Source File</a>                                 | 6-4 |
| <a href="#">Step 3: Load TACL Definitions Into Memory</a>                       | 6-4 |
| <a href="#">Step 4: Compile Filter Source File</a>                              | 6-5 |
| <a href="#">Step 5: Load the Compiled Filter Object File Into A Distributor</a> | 6-5 |
| <a href="#">Using the Format Template</a>                                       | 6-5 |
| <a href="#">Contents of the Format Template Supplied With NetBatch Software</a> | 6-5 |
| <a href="#">Modifying the Format Template</a>                                   | 6-8 |
| <a href="#">Event-Message Descriptions</a>                                      | 6-9 |

## **7. NetBatch Procedure Calls**

|   |      |
|---|------|
| <a href="#">NB^JOB^SUBMIT Procedure</a> | 7-2  |
| <a href="#">Sample Programs</a>         | 7-10 |
| <a href="#">Sample C Program</a>        | 7-10 |
| <a href="#">Sample COBOL Program</a>    | 7-18 |
| <a href="#">Sample TAL Program</a>      | 7-27 |

## **A. Error Numbers and Error Lists**

|   |     |
|---|-----|
| <a href="#">Notation Used</a>           | A-1 |
| <a href="#">Error-List Descriptions</a> | A-2 |
| <a href="#">512 ZBAT-WRN-SEC-BREACH</a> | A-2 |

## **A. Error Numbers and Error Lists (continued)**

|  |      |
|--|------|
| <a href="#"><u>513 ZBAT-WRN-IN-NE</u></a>              | A-3  |
| <a href="#"><u>514 ZBAT-WRN-EXECPROG-NE</u></a>        | A-3  |
| <a href="#"><u>515 ZBAT-WRN-CLASS-INITIATION</u></a>   | A-4  |
| <a href="#"><u>516 ZBAT-WRN-EXECUTOR-STARTED</u></a>   | A-4  |
| <a href="#"><u>517 ZBAT-WRN-JOB-EXECUTING</u></a>      | A-5  |
| <a href="#"><u>518 ZBAT-WRN-WAITON-SATISFIED</u></a>   | A-5  |
| <a href="#"><u>522 ZBAT-WRN-NOT-NETWORKABLE</u></a>    | A-6  |
| <a href="#"><u>524 ZBAT-WRN-ALTER-TAPEDRIVES</u></a>   | A-7  |
| <a href="#"><u>525 ZBAT-WRN-CPU-DOWN</u></a>           | A-8  |
| <a href="#"><u>526 ZBAT-WRN-EXECUTOR-STOPPED</u></a>   | A-9  |
| <a href="#"><u>527 ZBAT-WRN-R-ACCESS</u></a>           | A-10 |
| <a href="#"><u>528 ZBAT-WRN-W-ACCESS</u></a>           | A-10 |
| <a href="#"><u>529 ZBAT-WRN-E-ACCESS</u></a>           | A-11 |
| <a href="#"><u>530 ZBAT-WRN-P-ACCESS</u></a>           | A-11 |
| <a href="#"><u>531 ZBAT-WRN-CALENDAR-ERROR</u></a>     | A-12 |
| <a href="#"><u>532 ZBAT-WRN-CALENDAR-EXPIRED</u></a>   | A-13 |
| <a href="#"><u>534 ZBAT-WRN-ATT-DELETED</u></a>        | A-14 |
| <a href="#"><u>535 ZBAT-WRN-SECURITY</u></a>           | A-15 |
| <a href="#"><u>536 ZBAT-WRN-DEFAULTS-DEFINE</u></a>    | A-16 |
| <a href="#"><u>540 ZBAT-WRN-RUNNOW-TAPE</u></a>        | A-17 |
| <a href="#"><u>542 ZBAT-WRN-DISALLOW-DEFINE</u></a>    | A-18 |
| <a href="#"><u>544 ZBAT-WRN-SAME-SYSTEM</u></a>        | A-18 |
| <a href="#"><u>547 ZBAT-WRN-SWITCHCPU-DEFERRED</u></a> | A-19 |
| <a href="#"><u>548 ZBAT-WRN-PAST-TIME</u></a>          | A-19 |
| <a href="#"><u>549 ZBAT-INF-PHANDLES-OMITTED</u></a>   | A-20 |
| <a href="#"><u>2048 ZBAT-ERR-ACTIVATE-JOB</u></a>      | A-21 |
| <a href="#"><u>2050 ZBAT-ERR-AFTER-YEAR</u></a>        | A-22 |
| <a href="#"><u>2051 ZBAT-ERR-AFTER-MONTH</u></a>       | A-23 |
| <a href="#"><u>2052 ZBAT-ERR-AFTER-DAY</u></a>         | A-24 |
| <a href="#"><u>2053 ZBAT-ERR-AFTER-HOUR</u></a>        | A-25 |
| <a href="#"><u>2054 ZBAT-ERR-AFTER-MINUTE</u></a>      | A-26 |
| <a href="#"><u>2055 ZBAT-ERR-ALREADY-STARTED</u></a>   | A-26 |
| <a href="#"><u>2056 ZBAT-ERR-AT</u></a>                | A-27 |
| <a href="#"><u>2066 ZBAT-ERR-CALENDAR</u></a>          | A-27 |
| <a href="#"><u>2068 ZBAT-ERR-CALENDAR-FILECODE</u></a> | A-28 |
| <a href="#"><u>2069 ZBAT-ERR-COLD-START</u></a>        | A-28 |
| <a href="#"><u>2071 ZBAT-ERR-CLASS-COUNT</u></a>       | A-29 |
| <a href="#"><u>2073 ZBAT-ERR-CONTEXT</u></a>           | A-30 |

## **A. Error Numbers and Error Lists (continued)**

|  |      |
|--|------|
| <a href="#">2074 ZBAT-ERR-CPU</a>                | A-31 |
| <a href="#">2075 ZBAT-ERR-HOLDAFTER</a>          | A-31 |
| <a href="#">2076 ZBAT-ERR-NO-CPU</a>             | A-32 |
| <a href="#">2077 ZBAT-ERR-DELETE-JOB</a>         | A-32 |
| <a href="#">2078 ZBAT-ERR-EVERY-ZERO-MINUTES</a> | A-33 |
| <a href="#">2079 ZBAT-ERR-EVERY</a>              | A-34 |
| <a href="#">2080 ZBAT-ERR-EVERY-CAL-CRON</a>     | A-35 |
| <a href="#">2082 ZBAT-ERR-EXECUTOR-PROG</a>      | A-35 |
| <a href="#">2058 ZBAT-ERR-WAITON-SELF</a>        | A-36 |
| <a href="#">2086 ZBAT-ERR-EXECUTOR-EXISTS</a>    | A-36 |
| <a href="#">2087 ZBAT-ERR-NO-SUCH-EXECUTOR</a>   | A-37 |
| <a href="#">2090 ZBAT-ERR-EXTRA-TOKEN</a>        | A-37 |
| <a href="#">2091 ZBAT-ERR-HOLD</a>               | A-38 |
| <a href="#">2092 ZBAT-ERR-IFFAILS</a>            | A-38 |
| <a href="#">2093 ZBAT-ERR-IN</a>                 | A-39 |
| <a href="#">2095 ZBAT-ERR-INITIATION</a>         | A-39 |
| <a href="#">2096 ZBAT-ERR-USER-UNDEFINED</a>     | A-40 |
| <a href="#">2098 ZBAT-ERR-JOB-FULL</a>           | A-40 |
| <a href="#">2099 ZBAT-ERR-NO-SUCH-JOB</a>        | A-41 |
| <a href="#">2102 ZBAT-ERR-CLASS-EXISTS</a>       | A-41 |
| <a href="#">2104 ZBAT-ERR-CLASS-IN-USE</a>       | A-42 |
| <a href="#">2105 ZBAT-ERR-NO-SUCH-CLASS</a>      | A-43 |
| <a href="#">2106 ZBAT-ERR-JOBNAME</a>            | A-43 |
| <a href="#">2107 ZBAT-ERR-JOBNAME-EXISTS</a>     | A-44 |
| <a href="#">2108 ZBAT-ERR-JOBNAME-REQUIRED</a>   | A-44 |
| <a href="#">2117 ZBAT-ERR-EMPTY-RESPONSE</a>     | A-45 |
| <a href="#">2118 ZBAT-ERR-MAXPRINTLINES</a>      | A-46 |
| <a href="#">2119 ZBAT-ERR-MAXPRINTPAGES</a>      | A-47 |
| <a href="#">2120 ZBAT-ERR-MAXRESP</a>            | A-48 |
| <a href="#">2121 ZBAT-ERR-MISSING-ATTRIBUTES</a> | A-48 |
| <a href="#">2122 ZBAT-ERR-MISSING-CLASS</a>      | A-49 |
| <a href="#">2123 ZBAT-ERR-MULTIPLE-CONTEXT</a>   | A-49 |
| <a href="#">2124 ZBAT-ERR-MULTIPLE-MAPS</a>      | A-50 |
| <a href="#">2126 ZBAT-ERR-NAME-AND-NUMBER</a>    | A-51 |
| <a href="#">2127 ZBAT-ERR-NAME-OR-NUMBER</a>     | A-52 |
| <a href="#">2128 ZBAT-ERR-NO-SUBMIT</a>          | A-52 |
| <a href="#">2129 ZBAT-ERR-INVALID-COMMAND</a>    | A-53 |
| <a href="#">2131 ZBAT-ERR-NOT-STARTED</a>        | A-53 |



## **A. Error Numbers and Error Lists (continued)**

|   |      |
|---|------|
| <a href="#"><u>2132 ZBAT-ERR-SECURITY</u></a>         | A-54 |
| <a href="#"><u>2133 ZBAT-ERR-SHUTDOWN</u></a>         | A-54 |
| <a href="#"><u>2136 ZBAT-ERR-OUT</u></a>              | A-55 |
| <a href="#"><u>2137 ZBAT-ERR-PRI</u></a>              | A-56 |
| <a href="#"><u>2139 ZBAT-ERR-RESTART</u></a>          | A-56 |
| <a href="#"><u>2140 ZBAT-ERR-STOP-ON-ABEND</u></a>    | A-57 |
| <a href="#"><u>2141 ZBAT-ERR-RUNNEXT</u></a>          | A-58 |
| <a href="#"><u>2142 ZBAT-ERR-RUNNEXT-RUNNOW</u></a>   | A-59 |
| <a href="#"><u>2143 ZBAT-ERR-SWITCHLOG-EDIT</u></a>   | A-60 |
| <a href="#"><u>2144 ZBAT-ERR-SELPRI</u></a>           | A-61 |
| <a href="#"><u>2145 ZBAT-ERR-STARTUP-MESSAGE</u></a>  | A-62 |
| <a href="#"><u>2146 ZBAT-ERR-STOP-JOB</u></a>         | A-62 |
| <a href="#"><u>2148 ZBAT-ERR-SUSPEND-JOB</u></a>      | A-63 |
| <a href="#"><u>2149 ZBAT-ERR-TAPEDRIVES</u></a>       | A-64 |
| <a href="#"><u>2151 ZBAT-ERR-UNKNOWN-OBJECT</u></a>   | A-65 |
| <a href="#"><u>2153 ZBAT-ERR-UNKNOWN-TOKEN</u></a>    | A-65 |
| <a href="#"><u>2154 ZBAT-ERR-VOLUME-REQUIRED</u></a>  | A-66 |
| <a href="#"><u>2155 ZBAT-ERR-VOLUME</u></a>           | A-66 |
| <a href="#"><u>2158 ZBAT-ERR-WAITON-COUNT</u></a>     | A-67 |
| <a href="#"><u>2160 ZBAT-ERR-WAITON-JOBS-DUPL</u></a> | A-67 |
| <a href="#"><u>2167 ZBAT-ERR-SWITCHCPU</u></a>        | A-68 |
| <a href="#"><u>2168 ZBAT-ERR-LOGFILE</u></a>          | A-69 |
| <a href="#"><u>2169 ZBAT-ERR-NOT-C20-FILE</u></a>     | A-70 |
| <a href="#"><u>2170 ZBAT-ERR-DST</u></a>              | A-71 |
| <a href="#"><u>2171 ZBAT-ERR-ATT-EXISTS</u></a>       | A-72 |
| <a href="#"><u>2172 ZBAT-ERR-ATT-DNE</u></a>          | A-72 |
| <a href="#"><u>2173 ZBAT-ERR-ATT-JOB</u></a>          | A-73 |
| <a href="#"><u>2174 ZBAT-ERR-ATT-REQUESTOR</u></a>    | A-74 |
| <a href="#"><u>2175 ZBAT-ERR-ATT</u></a>              | A-75 |
| <a href="#"><u>2177 ZBAT-ERR-ATT-OVERFLOW</u></a>     | A-76 |
| <a href="#"><u>2178 ZBAT-ERR-ATT-UPDATE</u></a>       | A-77 |
| <a href="#"><u>2188 ZBAT-ERR-INTERNAL-ERROR</u></a>   | A-77 |
| <a href="#"><u>2189 ZBAT-ERR-FILE-ERROR</u></a>       | A-78 |
| <a href="#"><u>2191 ZBAT-ERR-NOT-IMPLEMENTED</u></a>  | A-78 |
| <a href="#"><u>2192 ZBAT-ERR-INVALID-SPI</u></a>      | A-79 |
| <a href="#"><u>2193 ZBAT-ERR-NETBATCH-NAME</u></a>    | A-80 |
| <a href="#"><u>2194 ZBAT-ERR-SUSPEND</u></a>          | A-81 |
| <a href="#"><u>2195 ZBAT-ERR-ACTIVATE</u></a>         | A-82 |

## **A. Error Numbers and Error Lists (continued)**

|  |       |
|--|-------|
| <a href="#"><u>2196 ZBAT-ERR-STOP</u></a>                | A-83  |
| <a href="#"><u>2197 ZBAT-ERR-STALL</u></a>               | A-83  |
| <a href="#"><u>2198 ZBAT-ERR-WILDCARD</u></a>            | A-84  |
| <a href="#"><u>2199 ZBAT-ERR-JOB-TOO-MANY-ATT</u></a>    | A-84  |
| <a href="#"><u>2200 ZBAT-ERR-DATE</u></a>                | A-85  |
| <a href="#"><u>2201 ZBAT-ERR-TIME</u></a>                | A-85  |
| <a href="#"><u>2202 ZBAT-ERR-AT-FLAG</u></a>             | A-86  |
| <a href="#"><u>2203 ZBAT-ERR-MISSING-EXECUTOR</u></a>    | A-86  |
| <a href="#"><u>2204 ZBAT-ERR-MISSING-ATT-ID</u></a>      | A-87  |
| <a href="#"><u>2205 ZBAT-ERR-MISSING-RELEASE-MAP</u></a> | A-87  |
| <a href="#"><u>2206 ZBAT-ERR-ATT-ASSIGN</u></a>          | A-88  |
| <a href="#"><u>2207 ZBAT-ERR-ATT-DEFINE</u></a>          | A-89  |
| <a href="#"><u>2208 ZBAT-ERR-ATT-PARAM</u></a>           | A-90  |
| <a href="#"><u>2209 ZBAT-ERR-JOB-DUPL-ATT</u></a>        | A-90  |
| <a href="#"><u>2210 ZBAT-ERR-AFTER-SECOND</u></a>        | A-91  |
| <a href="#"><u>2211 ZBAT-ERR-AFTER-MILLISEC</u></a>      | A-92  |
| <a href="#"><u>2212 ZBAT-ERR-AFTER-MICROSEC</u></a>      | A-93  |
| <a href="#"><u>2213 ZBAT-ERR-CLASS-NAME</u></a>          | A-94  |
| <a href="#"><u>2214 ZBAT-ERR-WAITON-ID</u></a>           | A-95  |
| <a href="#"><u>2215 ZBAT-ERR-EXECUTOR-NAME</u></a>       | A-96  |
| <a href="#"><u>2216 ZBAT-ERR-CLASS-INITIATION</u></a>    | A-96  |
| <a href="#"><u>2217 ZBAT-ERR-VAR-BUF-FULL</u></a>        | A-97  |
| <a href="#"><u>2218 ZBAT-ERR-CRONTAB</u></a>             | A-97  |
| <a href="#"><u>2219 ZBAT-ERR-PURGE-IN-FILE</u></a>       | A-98  |
| <a href="#"><u>2220 ZBAT-ERR-HIGHPIN</u></a>             | A-98  |
| <a href="#"><u>2221 ZBAT-ERR-POSIX</u></a>               | A-99  |
| <a href="#"><u>2222 ZBAT-ERR-SAVEABEND</u></a>           | A-99  |
| <a href="#"><u>2223 ZBAT-ERR-RUND</u></a>                | A-100 |
| <a href="#"><u>2224 ZBAT-ERR-JOBID-ZERO</u></a>          | A-100 |
| <a href="#"><u>2225 ZBAT-ERR-MEM</u></a>                 | A-101 |
| <a href="#"><u>2226 ZBAT-ERR-TIME-LIMIT</u></a>          | A-102 |
| <a href="#"><u>2227 ZBAT-ERR-DESCRIPTION</u></a>         | A-102 |
| <a href="#"><u>2228 ZBAT-ERR-TOO-MANY-SELECTORS</u></a>  | A-103 |
| <a href="#"><u>2229 ZBAT-ERR-NODENAME</u></a>            | A-104 |
| <a href="#"><u>2230 ZBAT-ERR-MAXPRI</u></a>              | A-105 |
| <a href="#"><u>2231 ZBAT-ERR-MAXCONCURRENTJOBS</u></a>   | A-106 |
| <a href="#"><u>2232 ZBAT-ERR-MAXTEMPEXECUTORS</u></a>    | A-107 |
| <a href="#"><u>2233 ZBAT-ERR-EVERY-CATCHUP</u></a>       | A-107 |

## **A. Error Numbers and Error Lists (continued)**

[2234 ZBAT-ERR-EMS](#) A-108

[2235 ZBAT-ERR-PFS](#) A-108

## **B. Token Codes and Token Maps**

[NetBatch Token Codes and Token Types](#) B-2

[NetBatch Token Maps and DDL Definitions](#) B-5

## **C. Sample Programs**

[Sample C Program](#) C-2

[Sample COBOL Program](#) C-15

[Sample TACL Macros](#) C-30

[Sample TAL Program](#) C-41

## **Glossary**

## **Index**

## **Examples**

[Example 7-1. Sample NB^JOB^SUBMIT C Program](#) 7-12

[Example 7-2. Sample NB^JOB^SUBMIT COBOL Program](#) 7-20

[Example 7-3. Sample NB^JOB^SUBMIT TAL Program](#) 7-28

[Example C-1. Sample SPI C Program](#) C-4

[Example C-2. Sample SPI COBOL Program](#) C-17

[Example C-3. Sample SPI TACL Macros](#) C-32

[Example C-4. Sample SPI TAL Program](#) C-42

## **Figures**

[Figure 1-1. NetBatch Subsystem](#) 1-4

[Figure 2-1. Steps in Communicating With the NetBatch Subsystem](#) 2-2

## **Tables**

[Table 1-1. Interactive Commands and Corresponding Programmatic Commands](#) 1-6

[Table 2-1. Open Errors for the Scheduler](#) 2-5

[Table 4-1. SPI Standard Definitions for Header Tokens](#) 4-2

[Table 4-2. SPI Standard Definitions for Other Simple Tokens](#) 4-2

[Table 4-3. SPI Standard Definitions for Special Tokens](#) 4-3

[Table 4-4. SPI Standard Definitions for Token Types](#) 4-3

[Table 4-5. SPI Standard Definitions for Structures](#) 4-3

[Table 4-6. SPI Standard Definitions for Value Names](#) 4-3

**Tables (continued)**

|                             |   |      |
|-----------------------------|---|------|
| <a href="#">Table 4-7.</a>  | <a href="#">EMS Standard Definitions for Header Tokens</a>                      | 4-6  |
| <a href="#">Table 4-8.</a>  | <a href="#">EMS Standard Definition for Data-Portion Token</a>                  | 4-6  |
| <a href="#">Table 4-9.</a>  | <a href="#">NetBatch Buffer Declarations</a>                                    | 4-8  |
| <a href="#">Table 4-10.</a> | <a href="#">NetBatch Private Token and Field Types</a>                          | 4-8  |
| <a href="#">Table 4-11.</a> | <a href="#">NetBatch Token Predefined Token and Field Values</a>                | 4-9  |
| <a href="#">Table 4-12.</a> | <a href="#">NetBatch Simple Tokens</a>  | 4-10 |
| <a href="#">Table 4-13.</a> | <a href="#">NetBatch Extensible Structured Tokens</a>                           | 4-11 |
| <a href="#">Table 5-1.</a>  | <a href="#">Available Commands</a>  | 5-2  |
| <a href="#">Table 5-2.</a>  | <a href="#">Symbolic Names of Commands and Objects</a>                          | 5-3  |
| <a href="#">Table 5-3.</a>  | <a href="#">Sensitive and Nonsensitive Programmatic NetBatch Commands</a>       | 5-5  |
| <a href="#">Table 5-4.</a>  | <a href="#">Commands That Support Wild-Character Matching</a>                   | 5-8  |
| <a href="#">Table 5-5.</a>  | <a href="#">Commands That Do Not Support Wild-Card Character Matching</a>       | 5-8  |
| <a href="#">Table 6-1.</a>  | <a href="#">NetBatch Event Messages</a>   | 6-2  |
| <a href="#">Table 6-2.</a>  | <a href="#">Contents of the Format Template Supplied With NetBatch Software</a> | 6-6  |
| <a href="#">Table B-1.</a>  | <a href="#">NetBatch Token Codes and Token Types</a>                            | B-2  |
| <a href="#">Table B-2.</a>  | <a href="#">NetBatch Token Maps and DDL Definitions</a>                         | B-5  |

---

---

---

---

# What's New in This Manual

## Manual Information

### Abstract

This manual describes the Distributed Systems Management (DSM) programmatic interfaces (commands, responses, and event messages in Subsystem Programmatic Interface [SPI] format) to the NetBatch subsystem.

### Product Version

NetBatch D30.00

### Supported Release Version Updates (RVUs)

This publication supports D20.00 and all subsequent D-series RVUs, G02.00 and all subsequent G-series RVUs, and H06.03 and all subsequent H-series RVUs until otherwise indicated by its replacement publication.

| Part Number | Published      |
|-------------|----------------|
| 522462-003  | September 2005 |

### Document History

| Part Number | Product Version | Published      |
|-------------|-----------------|----------------|
| 522462-001  | NetBatch D30    | February 2002  |
| 522462-002  | NetBatch D30    | May 2002       |
| 522462-003  | NetBatch D30    | September 2005 |

---

## New and Changed Information

- Modified the syntax for COBOL program on page [C-27](#)
- Rebranded the manual with the latest terminology

### Changes to the G06.16 Manual

- Updated the descriptions of [WAIT-TIME](#) on page 7-5 and [START-TIME](#) on page 7-5 to account for a range limit of 32767
- Updated the introductions to [Sample C Program](#) on page 7-10 and [Sample TAL Program](#) on page 7-27 to specify that all instances of WAIT-TIME and START-TIME must be unsigned numerics
- Added messages [548 ZBAT-WRN-PAST-TIME](#) on page A-19 and [549 ZBAT-INF-PHANDLES-OMITTED](#) on page A-20

## Changes to the G06.15 Manual

- Added new event messages to [Section 6, Event Messages](#).
- Added new NetBatch error codes to [Appendix A, Error Numbers and Error Lists](#).

## Previous Changes

The second edition of this manual contained these changes:

### New Commands

The D30 programmatic interfaces support these scheduler commands in addition to scheduler commands supported in the D20 interfaces:

|                 |                |                  |
|-----------------|----------------|------------------|
| ADD SCHEDULER   | INFO SCHEDULER | STATUS SCHEDULER |
| ALTER SCHEDULER |                |                  |

These commands were not available in the D20 interfaces.

### Changed Commands

#### All Commands

Commands previously available only to super-group users are now available to NetBatch supervisors (users with execute access to the NETBATCH program file).

#### Class Commands

The variable-length D30 token ZBAT-TKN-SEL-CLASSNAME replaces the fixed length D20 token ZBAT-TKN-SEL\_CLASSNAME in these commands:

|             |              |            |
|-------------|--------------|------------|
| ADD CLASS   | DELETE CLASS | INFO CLASS |
| ALTER CLASS |              |            |

#### Executor Commands

The variable-length D30 Token ZBAT-TKN-SEL-EXECUTORNAME replaces fixed-length D20 token ZBAT-TKN-SEL-EXECUTORNAME in these commands:

|                 |                |                 |
|-----------------|----------------|-----------------|
| ADD EXECUTOR    | INFO EXECUTOR  | STATUS EXECUTOR |
| ALTER EXECUTOR  | START EXECUTOR | STOP EXECUTOR   |
| DELETE EXECUTOR |                |                 |

#### Job Commands

The variable-length D30 token ZBAT-TKN-SEL-JOBNAME replaces the fixed-length D20 token ZBAT-TKN-SEL-JOBNAME in these commands:

|              |             |             |
|--------------|-------------|-------------|
| ACTIVATE JOB | RELEASE JOB | STOP JOB    |
| ALTER JOB    | RUNNEXT JOB | SUBMIT JOB  |
| DELETE JOB   | RUNNOW JOB  | SUSPEND JOB |
| INFO JOB     |             |             |

The D30 programmatic interfaces introduce new job-selection tokens and support wild-card character matching in these commands:

|              |             |             |
|--------------|-------------|-------------|
| ACTIVATE JOB | INFO JOB    | STOP JOB    |
| ALTER JOB    | RUNNEXT JOB | SUBMIT JOB  |
| DELETE JOB   | RUNNOW JOB  | SUSPEND JOB |

The D30 programmatic interfaces introduce new job-attribute tokens and extend the ZBAT-MAP-DEF-JOB and ZBAT-MAP-DEF-WAITON tokens in these commands:

|           |          |            |
|-----------|----------|------------|
| ALTER JOB | INFO JOB | SUBMIT JOB |
|-----------|----------|------------|

## New Private Token and Field Types

The D30 programmatic interfaces introduce these new private token and field types:

## Changed Private Token

The D30 private token ZBAT-DDL-WAITON-INDICATOR differs from its D20 counterpart (enumerated value ZBAT-ENM-WAITON-RELEASED replaces enumerated value ZBAT-ENM-WAITON-RELEASED-OK).

## New Simple Tokens

The D30 programmatic interfaces introduce these new and simple tokens:

---

**Note.** The variable-length D30 tokens ZBAT-TKN-SEL-CLASSNAME, ZBAT-TKN-SEL-EXECUTORNAME, and ZBAT-TKN-SEL-JOBNAME replace the fixed-length D20 tokens ZBAT-TKN-SEL-CLASS-NAME, ZBAT-TKN-SEL-EXECUTOR-NAME, and ZBAT-TKN-SEL-JOB-NAME.

---

|                            |                           |
|----------------------------|---------------------------|
| ZBAT-TKN-BATCHCTL          | ZBAT-TKN-SEL-NOTADPNAME   |
| ZBAT-TKN-DESCRIPTION       | ZBAT-TKN-SEL-NOTCLASSNAME |
| ZBAT-TKN-EXTSWAP-FILE      | ZBAT-TKN-SEL-NOTINNAME    |
| ZBAT-TKN-LIB-FILE          | ZBAT-TKN-SEL-NOTJOBNAME   |
| ZBAT-TKN-PHANDLE           | ZBAT-TKN-SEL-NOTLIST      |
| ZBAT-TKN-SEL-ADPNAME       | ZBAT-TKN-SEL-NOTUSERNAME  |
| ZBAT-TKN-SEL-CLASSNAME     | ZBAT-TKN-SEL-NOTWAITON    |
| ZBAT-TKN-SEL-EXECUTORNAME  | ZBAT-TKN-SEL-USERNAME     |
| ZBAT-TKN-SEL-INNAME        | ZBAT-TKN-SEL-WAITON       |
| ZBAT-TKN-SEL-JOBNAME       | ZBAT-TKN-SWAP-FILE        |
| ZBAT-TKN-SEL-LIST          | ZBAT-TKN-TERM-FILE        |
| ZBAT-TKN-SEL-NETBATCH-NAME | ZBAT-TKN-TIME-LIMIT       |

## New Predefined Token and Field Values

The D30 programmatic interfaces introduce these new predefined token and field values:

|                                |                             |
|--------------------------------|-----------------------------|
| ZBAT-VAL-EMS-ERROR             | ZBAT-VAL-SCHEDULER-ZSTARTED |
| ZBAT-VAL-EMS-OFF               | ZBAT-VAL-SPECIAL-1          |
| ZBAT-VAL-EMS-ON                | ZBAT-VAL-SPECIAL-2          |
| ZBAT-VAL-EVENT                 | ZBAT-VAL-SPECIAL-3          |
| ZBAT-VAL-EXECUTING             | ZBAT-VAL-SPECIAL-4          |
| ZBAT-VAL-FIRST-LIST            | ZBAT-VAL-SPECIAL-5          |
| ZBAT-VAL-LAST-LIST             | ZBAT-VAL-SPECIAL-6          |
| ZBAT-VAL-READY                 | ZBAT-VAL-SPECIAL-7          |
| ZBAT-VAL-RUNNEXT               | ZBAT-VAL-SPECIAL-8          |
| ZBAT-VAL-RUNNOW                | ZBAT-VAL-SPECIAL-9          |
| ZBAT-VAL-SCHEDULER-ZNOTSTARTED | ZBAT-VAL-SPECIAL-ANY        |
| ZBAT-VAL-SCHEDULER-ZSHUTDOWN   | ZBAT-VAL-SUSPENDED          |
| ZBAT-VAL-TAPE                  | ZBAT-VAL-WAITON-STOP        |
| ZBAT-VAL-TIME                  | ZBAT-VAL-WAITON-STOPABEND   |
| ZBAT-VAL-WAITON-SET            |                             |

## New Extensible Structured Tokens

The D30 programmatic interfaces introduce these new extensible structured tokens:

|                        |                           |
|------------------------|---------------------------|
| ZBAT-MAP-DEF-CRONTAB   | ZBAT-MAP-STATUS-SCHEDULER |
| ZBAT-MAP-DEF-SCHEDULER |                           |

## Changed Extensible Structured Tokens

Some D30 extensible structured tokens differ from their D20 counterparts. These tokens are:

- ZBAT-MAP-DEF-JOB (extended—eleven new ZBAT-DDL-DEF-JOB fields)
- ZBAT-MAP-DEF-WAITON (extended—one new ZBAT-DDL-DEF-WAITON field)
- ZBAT-MAP-DEFINE-ERROR (type of ZNAMETXT field in ZBAT-DDL-DEFINE-ERROR changed)
- ZBAT-MAP-STATUS-JOB (extended—eight new ZBAT-DDL-STATUS-JOB fields)

## New Scheduler-Generated Event Message

The D30- programmatic interfaces introduce this new scheduler-generated event message:

204: ZBAT-EVT-JOB-OVER-LIMIT



## New Errors and Warnings

The D30 programmatic interfaces introduce these new errors and warnings (that is, new ZBAT-DDL-RETCODE values):

- ZBAT-ENM-E-CRONTAB (value is 2218)
- ZBAT-ENM-E-DESCRIPTION (value is 2227)
- ZBAT-ENM-E-EMS (value is 2234)
- ZBAT-ENM-E-EVERY-CATCHUP (value is 2233)
- ZBAT-ENM-E-HIGHPIN (value is 2220)
- ZBAT-ENM-E-JOBID-ZERO (value is 2224)
- ZBAT-ENM-E-MAXCONCURRENTJOBS (value is 2231)
- ZBAT-ENM-E-MAXPRI (value is 2230)
- ZBAT-ENM-E-MAXTEMPEXECUTORS (value is 2232)
- ZBAT-ENM-E-MEM (value is 2225)
- ZBAT-ENM-E-NODENAME (value is 2229)
- ZBAT-ENM-E-PFS (value is 2235)
- ZBAT-ENM-E-POSIX (value is 2221)
- ZBAT-ENM-E-PURGE-IN-FILE (value is 2219)
- ZBAT-ENM-E-RUND (value is 2223)
- ZBAT-ENM-E-SAVEABEND (value is 2222)
- ZBAT-ENM-E-TIME-LIMIT (value is 2226)
- ZBAT-ENM-E-TOO-MANY-SELECTORS (value is 2228)
- ZBAT-ENM-E-VAR-BUF-FULL (value is 2217)
- ZBAT-ENM-W-SAME-SYSTEM (value is 544)
- ZBAT-ENM-W-SWITCHCPU-DEFERRED (value is 547)

## Changed Errors

One D30 ZBAT-DDL-RETCODE value and some D30 ZBAT-ERR- error lists differ from their D20 counterparts. These items are:

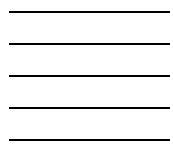
- ZBAT-ENM-E-EVERY-CAL-CRON (value is 2080—replaces the D20 error ZBAT- ENM-E-EVERY-CALENDAR)
- ZBAT-ERR-RUNNEXT (ZBAT-TKN-INT and ZBAT-TKN-SEL-JOBNAME now appear in error list)
- ZBAT-ERR-RUNNEXT-RUNNOW (ZBAT-TKN-INT and ZBAT-TKN-SEL- JOBNAME now appear in error list)

## Deleted Errors

The D30 programmatic interfaces exclude these D20 errors (that is, deleted ZBAT-DDL-RETCODE values):

- ZBAT-ENM-E-WAITON-NOT-ANY (value is 2161)
- ZBAT-ENM-E-WAITON-THIS-JOB (value is 2162)





# About This Manual

This manual describes the Distributed Systems Management (DSM) programmatic interfaces (commands, responses, and event messages in Subsystem Programmatic Interface (SPI) format) to the NetBatch subsystem.

## Audience

The intended audience for this manual includes:

- Experienced C, COBOL, HP Tandem Advanced Command Language (TACL ), and Transaction Application Language (TAL) programmers who write applications that communicate programmatically with the NetBatch subsystem
- Others who need detailed information about the NetBatch programmatic interfaces

## Prerequisites

The audience prerequisites are:

- Familiarity with NonStop <sup>TM</sup> system architecture, HP NonStop operating system, and system operations
- Familiarity with SPI, as described in the *SPI Programming Manual*
- Reading knowledge of the Data Definition Language (DDL)
- Skill in programming in the language used to write the application
- Detailed working knowledge of NetBatch subsystem concepts, facilities, management, and operations as described in the *NetBatch Manual*

## Organization

### Section

[Section 1, Introduction](#)

[Section 2,  
Communicating With the  
NetBatch Subsystem](#)

[Section 3, SPI  
Programming  
Considerations for the  
NetBatch Subsystem](#)

[Section 4, Common  
Definitions](#)

### Description

Introduces the programmatic interface to the NetBatch subsystem.

Explains how to set up communication between a management application and the NetBatch subsystem.

Discusses SPI programming considerations that are specific to the NetBatch subsystem.

Discusses SPI and Event Management Service (EMS) standard definitions and NetBatch definitions used in the token-oriented programmatic interface to the NetBatch subsystem.

| Section   | Description   |
|---|---|
| <a href="#">Section 5, Commands and Responses</a>         | Describes the syntax and semantics of all NetBatch programmatic commands and the responses to those commands.   |
| <a href="#">Section 6, Event Messages</a>                 | Describes the EMS event messages that the NetBatch subsystem can issue and the specific programming considerations for dealing with these event messages in an application.             |
| <a href="#">Section 7, NetBatch Procedure Calls</a>       | Describes the NetBatch procedure call NB^JOB^SUBMIT and contains working C, COBOL, and TAL program examples that use the procedure.   |
| <a href="#">Appendix A, Error Numbers and Error Lists</a> | Lists NetBatch subsystem error numbers (that is, the values whose symbolic names begin with ZBAT-WRN- and ZBAT-ERR-) and describes the error lists associated with the error numbers.   |
| <a href="#">Appendix B, Token Codes and Token Maps</a>    | Lists token codes and token maps specific to the NetBatch subsystem. For each token code, the appendix lists the token type. For each token map, the appendix lists the DDL definition. |
| <a href="#">Appendix C, Sample Programs</a>               | Contains working C, COBOL, and TAL program examples that illustrate programmatic management of the NetBatch subsystem.  |

## Further Reading

This manual contains references to these manuals:

| Section  | Description  |
|--|--|
| <i>Data Definition Language (DDL) Reference Manual</i> | Describes the DDL language syntax and the DDL dictionary database.   |
| <i>DSM Template Services Manual</i>                    | Describes Distributed Systems Management (DSM) Template Services, which support the representation of SPI buffers in display text.             |
| <i>EMS Manual</i>                                      | Describes EMS, a collection of processes, tools, and interfaces that provide event-message collection and distribution in the DSM environment. |
| <i>Guardian Procedure Calls Reference Manual</i>       | Describes the syntax of all system procedure calls.  |
| <i>Guardian Procedure Errors and Messages Manual</i>   | Describes system procedure error codes and error lists, system messages, traps, and the trap error list.                                       |
| <i>Guardian User's Guide</i>                           | Describes basic operating-system tasks.  |
| <i>NetBatch Manual</i>                                 | Describes NetBatch subsystem concepts, facilities, management, and operations.   |

| Section                           | Description  |
|-----------------------------------|--|
| <i>Safeguard Reference Manual</i> | Describes the Safeguard distributed security management facility and the syntax of the commands of the SAFECOM command interpreter.  |
| <i>SPI Programming Manual</i>     | Describes the Subsystem Programmatic Interface (SPI) and explains how to use it in management applications and subsystems you write. |
| <i>TACL Reference Manual</i>      | Describes the syntax, operation, and results of all TACL commands, functions, built-in functions, and built-in variables.            |

## Notation Conventions

### Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

This requirement is described under [Backup DAM Volumes and Physical Disk Drives](#) on page 3-2.

### General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

**UPPERCASE LETTERS.** Uppercase letters indicate keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required. For example:

MAXATTACH

**lowercase italic letters.** Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

*file-name*

**computer type.** Computer type letters within text indicate C and Open System Services (OSS) keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required. For example:

myfile.c

**italic computer type.** *Italic computer type* letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

*pathname*

**[ ] Brackets.** Brackets enclose optional syntax items. For example:

```
TERM [ \system-name. ] $terminal-name
INT[ ERRUPTS ]
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list may be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [  num   ]
   [ -num   ]
   [  text   ]

K [ X | D ] address
```

**{ } Braces.** A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list may be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name }

ALLOWSU { ON | OFF }
```

**| Vertical Line.** A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

**... Ellipsis.** An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]...
[ - ] { 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 }...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char..."
```

**Punctuation.** Parentheses, commas, semicolons, and other symbols not previously described must be entered as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;

LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must enter as shown. For example:

```
"[ repetition-constant-list ]"
```

**Item Spacing.** Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In these example, there are no spaces permitted between the period and any other items:

```
$process-name . #su-name
```

**Line Spacing.** If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE  
  
    [ , attribute-spec ]...
```

**!i and !o.** In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT ( segment-id                !i  
                        , error                      ) ;    !o
```

**!i,o.** In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEDIT ( filenum ) ;                !i,o
```

**!i:i.** In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ ( filename1:length        !i:i  
                        , filename2:length ) ;        !i:i
```

**!o:i.** In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ ( filenum                !i  
                        , [ filename:maxlen ] ) ;    !o:i
```

## Notation for Messages

list summarizes the notation conventions for the presentation of displayed messages in this manual.

**Bold Text.** Bold text in an example indicates user input entered at the terminal. For example:

```
ENTER RUN CODE
```

```
?123
```

```
CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

**Nonitalic text.** Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

**lowercase italic letters.** Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

```
p-register
```

```
process-name
```

**[ ] Brackets.** Brackets enclose items that are sometimes, but not always, displayed. For example:

```
Event number = number [ Subject = first-subject-value ]
```

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list might be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
proc-name trapped [ in SQL | in SQL file system ]
```

**{ } Braces.** A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list might be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
obj-type obj-name state changed to state, caused by  
{ Object | Operator | Service }
```

```
process-name State changed from old-objstate to objstate  
{ Operator Request. }  
{ Unknown. }
```

**| Vertical Line.** A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```



**% Percent Sign.** A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

%005400

%B101111

%H2F

P=%*p-register* E=%*e-register*

## Change Bar Notation

Change bars are used to indicate substantive differences between this edition of the manual and the preceding edition. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

The message types specified in the REPORT clause are different in the COBOL environment and the Common Run-Time Environment (CRE).

The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

## Abbreviations

The glossary of this manual includes abbreviations.



# **1** Introduction

This section introduces NetBatch:

| <b>Topic</b>   | <b>Page</b>         |
|--|---------------------|
| <a href="#">NetBatch Subsystem Description</a>                               | <a href="#">1-2</a> |
| <a href="#">NetBatch Programmatic Interfaces</a>                             | <a href="#">1-3</a> |
| <a href="#">Objects Managed by the NetBatch Subsystem</a>                    | <a href="#">1-5</a> |
| <a href="#">Supported Object-Management Functions</a>                        | <a href="#">1-5</a> |
| <a href="#">Interactive Commands and Corresponding Programmatic Commands</a> | <a href="#">1-6</a> |

# NetBatch Subsystem Description

The NetBatch subsystem lets your organization automate job scheduling, startup, and management on NonStop systems. The subsystem has four core components (in addition to the SPI-compatible programmatic interfaces and the NB^JOB^SUBMIT procedure call described in this manual):

- A scheduler program (NETBATCH)
- An interactive interface (BATCHCOM)
- A calendar-generation program (BATCHCAL)
- An executor program (NBEXEC)

This subsection briefly describes these components. For detailed descriptions, and a comprehensive overview of the NetBatch subsystem, see the *NetBatch Manual*. For a graphical representation of a sample NetBatch subsystem, see [Figure 1-1](#) on page 1-4.

## NETBATCH

NETBATCH is the program file ID of the NetBatch scheduler. The scheduler is a process-pair server that stores job records in its database. It schedules and starts the jobs, tracks and controls their execution, and records details of their termination. It also controls, through its classes and executors, the distribution of jobs among processors in your system.

## BATCHCOM

BATCHCOM is the program file ID of the NetBatch command interpreter. BATCHCOM enables interactive and noninteractive manipulation of NetBatch objects (the scheduler, the scheduler's executors and classes, and attachment sets and jobs).

## BATCHCAL

BATCHCAL is the program file ID of the NetBatch calendar program. The program allows you to generate a calendar file containing a series of dates and times called run times. You can schedule a job to run automatically at those times by using the CALENDAR attribute to assign the file to the job. BATCHCAL also can display run times in a calendar file and reformat an old calendar file to the current format.

## NBEXEC

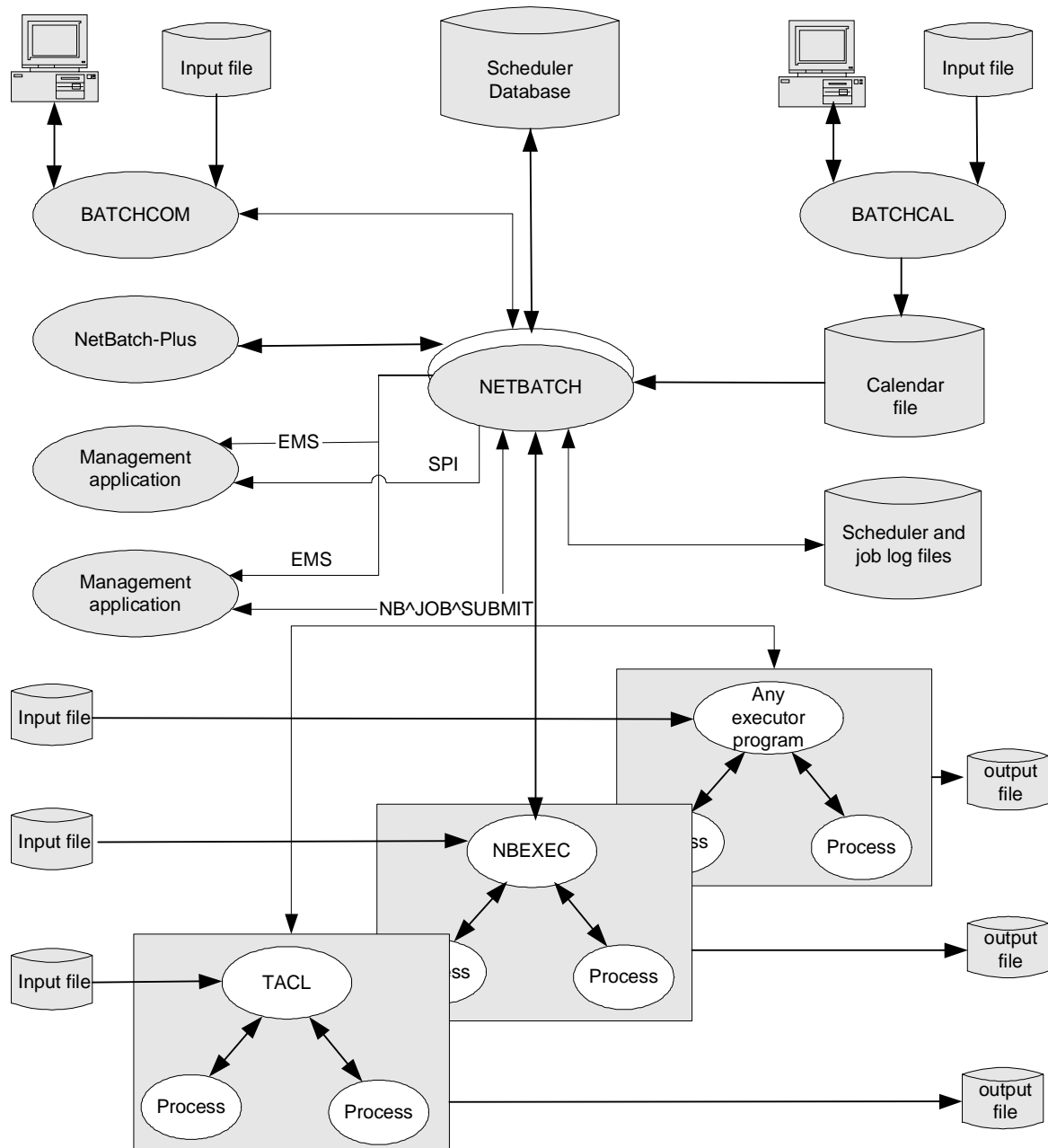
NBEXEC is the file ID of the NetBatch NonStop executor program. Compatible with BPROC (the batch execution process of the obsolete product MIS Batch), NBEXEC executes control file commands, supplies data to started processes, and logs process output. NBEXEC can run as a process pair and offers a simple-but-powerful job control language that includes error-testing and job-recovery facilities.

# NetBatch Programmatic Interfaces

The NetBatch subsystem's SPI-compatible programmatic interfaces enable your applications to:

- Send commands to and receive responses from the subsystem in the form of tokenized messages
- Retrieve events sent by the subsystem in the form of tokenized messages

For general information about SPI and EMS, see the *SPI Programming Manual* and the *EMS Manual*.

**Figure 1-1. NetBatch Subsystem**

VST001.vsd

# Objects Managed by the NetBatch Subsystem

These objects are managed by the NetBatch subsystem:

- An attachment set is a named set of ASSIGNS, DEFINES, and PARAMs.
- A class is a logical entity in the scheduler. A class's purpose is to group jobs and to control their flow to executors and thereby to the executors' processors.
- An executor is a logical entity in the scheduler. An executor's purpose is to link jobs through their classes to a processor. This link enables the scheduler to execute, in the specified processor, the initial process (the executor program) of each job.
- A job is a process or a sequence of processes that performs specified tasks.
- A scheduler is a process-pair server that stores job records in its database, schedules and starts jobs, monitors their execution, and records termination details.

No other subsystems control these objects.

## Supported Object-Management Functions

Your applications can use the programmatic interfaces to the NetBatch subsystem to perform these object-management functions:

- Add attachment sets, classes, and executors to a scheduler
- Alter the attributes of attachment sets, classes, executors, jobs, and schedulers
- Create and initialize scheduler databases
- Delete attachment sets, classes, executors, and jobs from a scheduler
- Display the attributes of attachment sets, classes, executors, jobs, and schedulers
- Display the status of attachment sets, executors, jobs, and schedulers
- Override job dependencies, timing attributes, and selection priority
- Start and stop executors
- Start, shut down, and abort schedulers
- Submit, suspend, activate, and stop jobs
- Switch scheduler processors and scheduler log files

# Interactive Commands and Corresponding Programmatic Commands

[Table 1-1](#) lists the interactive (BATCHCOM) commands that control NetBatch objects and the corresponding programmatic commands and object types. For details of the programmatic commands, see [Section 5, Commands and Responses](#).

**Table 1-1. Interactive Commands and Corresponding Programmatic Commands** (page 1 of 2)

| Interactive Command   | Programmatic Command | Object Type        |
|-----------------------|----------------------|--------------------|
| ABORT SCHEDULER       | ZBAT-CMD-ABORT       | ZBAT-OBJ-SCHEDULER |
| ACTIVATE JOB          | ZBAT-CMD-ACTIVATE    | ZBAT-OBJ-JOB       |
| ADD ATTACHMENT-SET    | ZBAT-CMD-ADD         | ZBAT-OBJ-ATT-SET   |
| ADD CLASS             | ZBAT-CMD-ADD         | ZBAT-OBJ-CLASS     |
| ADD EXECUTOR          | ZBAT-CMD-ADD         | ZBAT-OBJ-EXECUTOR  |
| ADD SCHEDULER         | ZBAT-CMD-ADD         | ZBAT-OBJ-SCHEDULER |
| ALTER ATTACHMENT-SET  | ZBAT-CMD-ALTER       | ZBAT-OBJ-ATT-SET   |
| ALTER CLASS           | ZBAT-CMD-ALTER       | ZBAT-OBJ-CLASS     |
| ALTER EXECUTOR        | ZBAT-CMD-ALTER       | ZBAT-OBJ-EXECUTOR  |
| ALTER JOB             | ZBAT-CMD-ALTER       | ZBAT-OBJ-JOB       |
| ALTER SCHEDULER       | ZBAT-CMD-ALTER       | ZBAT-OBJ-SCHEDULER |
| DELETE ATTACHMENT-SET | ZBAT-CMD-DELETE      | ZBAT-OBJ-ATT-SET   |
| DELETE CLASS          | ZBAT-CMD-DELETE      | ZBAT-OBJ-CLASS     |
| DELETE EXECUTOR       | ZBAT-CMD-DELETE      | ZBAT-OBJ-EXECUTOR  |
| DELETE JOB            | ZBAT-CMD-DELETE      | ZBAT-OBJ-JOB       |
| INFO ATTACHMENT-SET   | ZBAT-CMD-INFO        | ZBAT-OBJ-ATT-SET   |
| INFO CLASS            | ZBAT-CMD-INFO        | ZBAT-OBJ-CLASS     |
| INFO EXECUTOR         | ZBAT-CMD-INFO        | ZBAT-OBJ-EXECUTOR  |
| INFO JOB              | ZBAT-CMD-INFO        | ZBAT-OBJ-JOB       |
| INFO SCHEDULER        | ZBAT-CMD-INFO        | ZBAT-OBJ-SCHEDULER |
| RUNNEXT JOB           | ZBAT-CMD-RUNNEXT     | ZBAT-OBJ-JOB       |
| RUNNOW JOB            | ZBAT-CMD-RUNNOW      | ZBAT-OBJ-JOB       |
| SHUTDOWN SCHEDULER    | ZBAT-CMD-SHUTDOWN    | ZBAT-OBJ-SCHEDULER |
| START EXECUTOR        | ZBAT-CMD-START       | ZBAT-OBJ-EXECUTOR  |
| START SCHEDULER       | ZBAT-CMD-START       | ZBAT-OBJ-SCHEDULER |
| STATUS ATTACHMENT-SET | ZBAT-CMD-STATUS      | ZBAT-OBJ-ATT-SET   |
| STATUS EXECUTOR       | ZBAT-CMD-STATUS      | ZBAT-OBJ-EXECUTOR  |



**Table 1-1. Interactive Commands and Corresponding Programmatic Commands** (page 2 of 2)

| Interactive Command | Programmatic Command | Object Type        |
|---------------------|----------------------|--------------------|
| STATUS JOB          | ZBAT-CMD-STATUS      | ZBAT-OBJ-JOB       |
| STATUS SCHEDULER    | ZBAT-CMD-STATUS      | ZBAT-OBJ-SCHEDULER |
| STOP EXECUTOR       | ZBAT-CMD-STOP        | ZBAT-OBJ-EXECUTOR  |
| STOP JOB            | ZBAT-CMD-STOP        | ZBAT-OBJ-JOB       |
| SUBMIT JOB          | ZBAT-CMD-SUBMIT      | ZBAT-OBJ-JOB       |
| SUSPEND JOB         | ZBAT-CMD-SUSPEND     | ZBAT-OBJ-JOB       |
| SWITCHCPU SCHEDULER | ZBAT-CMD-SWITCHCPU   | ZBAT-OBJ-SCHEDULER |
| SWITCHLOG SCHEDULER | ZBAT-CMD-SWITCHLOG   | ZBAT-OBJ-SCHEDULER |

These commands (not listed in [Table 1-1](#)) are available only in the interactive interface:

|                           |                          |
|---------------------------|--------------------------|
| ALLOW ERRORS              | RELEASE-WAITON           |
| ASSUME <i>object-type</i> | RESET <i>object-type</i> |
| CHANGEUSER                | RUN                      |
| COMMENT                   | SET <i>object-type</i>   |
| DISPLAY-SPI               | SHOW <i>object-type</i>  |
| EXIT                      | STATUS-HISTORY           |
| FC                        | SYSTEM                   |
| HELP                      | VOLUME                   |
| HISTORY                   | !                        |
| OBEY                      | ==                       |
| OPEN                      | ?                        |

This command (not listed in [Table 1-1](#)) is available only in the programmatic interfaces:

ZSPI-CMD-GETVERSION (object type ZSPI-VAL-NULL-OBJECT-TYPE)



# Communicating With the NetBatch Subsystem

This section explains how to set up communication between a management application and the NetBatch subsystem:

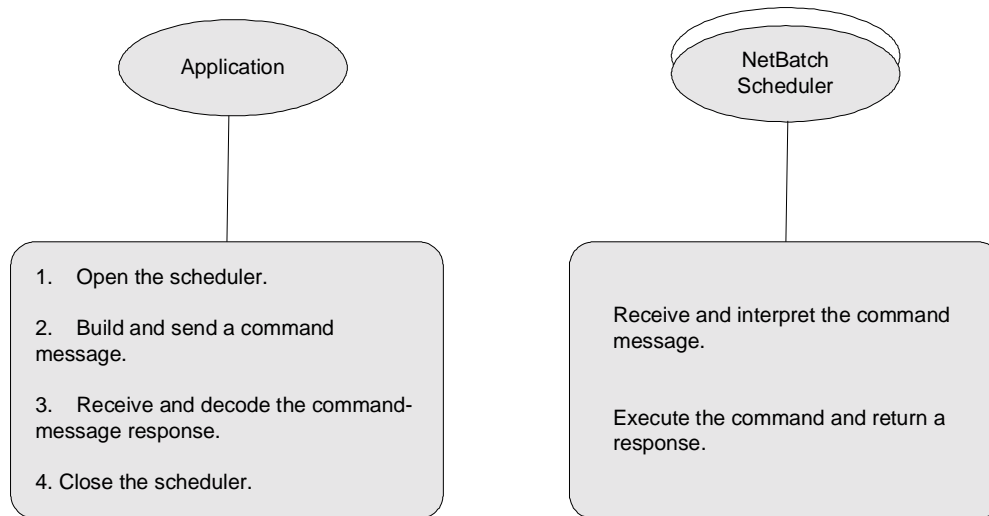
| Topic  | Page                |
|--|---------------------|
| <a href="#">Steps in Communicating With the NetBatch Subsystem</a> | <a href="#">2-1</a> |
| <a href="#">Starting a Scheduler</a>                               | <a href="#">2-2</a> |
| <a href="#">Opening a Scheduler</a>                                | <a href="#">2-3</a> |
| <a href="#">Closing a Scheduler</a>                                | <a href="#">2-5</a> |
| <a href="#">Stopping a Scheduler</a>                               | <a href="#">2-6</a> |
| <a href="#">Running an EMS Consumer Distributor</a>                | <a href="#">2-6</a> |

## Steps in Communicating With the NetBatch Subsystem

[Figure 2-1](#) summarizes the procedure your application must follow when communicating with the NetBatch subsystem.

For descriptions of Steps 1 and 4, see [Opening a Scheduler](#) on page 2-3 and [Closing a Scheduler](#) on page 2-5, respectively.

For descriptions of Steps 2 and 3, see [Building and Sending a Command Message](#) on page 3-5 and [Receiving and Decoding a Response Message](#) on page 3-7, respectively.

**Figure 2-1. Steps in Communicating With the NetBatch Subsystem**

VST002.vsd

## Starting a Scheduler

The NetBatch scheduler must be running before your application can open and communicate with it. You can start a scheduler interactively or programmatically.

### Starting a Scheduler Interactively

To start a scheduler interactively, use the TACL RUN command to run the scheduler program NETBATCH. When NETBATCH is running, use the BATCHCOM commands ADD SCHEDULER (cold starts only) and START SCHEDULER (cold starts and warm starts) to make the scheduler available for use. These examples show interactive cold-start and warm-start procedures:

| <b>Procedure</b> | <b>Example</b>  |
|------------------|---|
| Cold start       | <pre> &gt; NETBATCH /NAME \$SCHD, NOWAIT, PRI 130, CPU 0/ \$DATA7.SCHD !  &gt; BATCHCOM \$SCHD; ADD SCHEDULER, BACKUPCPU 1 Scheduler added  &gt; BATCHCOM \$SCHD; START SCHEDULER Scheduler started           </pre>  |
| Warm start       | <pre> &gt; NETBATCH /NAME \$SCHD, NOWAIT, PRI 149, CPU 2/ \$DATA7.SCHD &gt; BATCHCOM \$SCHD; ALTER SCHEDULER, BACKUPCPU 3 Scheduler altered  &gt; BATCHCOM \$SCHD; START SCHEDULER Scheduler started           </pre> |

For detailed information on the procedures, see the *NetBatch Manual*.

## Starting a Scheduler Programmatically

To start a scheduler programmatically, use the Guardian procedure, `PROCESS_CREATE_` to run the scheduler program `NETBATCH`. When `NETBATCH` is running, use the programmatic commands `ADD SCHEDULER` (cold starts only) and `START SCHEDULER` (cold starts and warm starts) to make the scheduler available for use. For detailed information on the cold-start and warm-start procedures, see the *NetBatch Manual*.

For information on the `PROCESS_CREATE_` procedure, see the *Guardian Procedure Calls Reference Manual*. For information on the programmatic `ADD SCHEDULER` and `START SCHEDULER` commands, see [Section 5, Commands and Responses](#).

## Opening a Scheduler

A process sets up communication with a NetBatch scheduler by sending an open request to the scheduler. The scheduler can handle up to 2000 concurrent opens from executor-program processes, child processes of executor-program processes, and `BATCHCOM`, NetBatch-Plus, or user-written requester processes.

You can open a scheduler interactively or programmatically.

### Opening a Scheduler Interactively

To open a scheduler interactively, use the `BATCHCOM` command `OPEN`. You can execute the command explicitly during a `BATCHCOM` session or implicitly in, for

example, a RUN BATCHCOM command. These examples show explicit and implicit OPEN commands:

### OPEN Command Examples

#### Explicit

```
> BATCHCOM

BATCHCOM - T9190D30 - (31OCT94^01JUN94)

(C)1986 Tandem (C)2004 Hewlett Packard Development
Company, L.P.

1} OPEN $SCHD

NETBATCH SERVER - T9190D30 - (31OCT94-01JUN94)  Time:
29JUL94 ...
```

#### Implicit

```
RUN BATCHCOM command specifies a scheduler:

> BATCHCOM $SCHD

BATCHCOM - T9190D30 - (31OCT94^01JUN94)

(C)1986 Tandem (C)2004 Hewlett Packard Development
Company, L.P.
NETBATCH SERVER - T9190D30 - (31OCT94-01JUN94)  Time:
29JUL94 ...

RUN BATCHCOM command does not specify a scheduler, so the
first scheduler-related command after the session begins
opens $ZBAT by default:

> BATCHCOM

BATCHCOM - T9190D30 - (31OCT94^01JUN94)
(C)1986 Tandem (C)2004 Hewlett Packard Development
Company, L.P.

1} STATUS SCHEDULER

NETBATCH SERVER - T9190D30 - (31OCT94-01JUN94)  Time:
29JUL94 ...

SCHEDULER STATUS

Process : \MELRISK.$ZBAT      Primary : 0,54   Backup :
1,55

...
```

---

**Note.** When BATCHCOM runs as a job's executor-program process, the default scheduler is the scheduler controlling the job.

---

For more information on the OPEN command, see the *NetBatch Manual*.

## Opening a Scheduler Programmatically

To set up programmatic communication with the scheduler, your application must open the scheduler using the mechanism appropriate to your programming language (for example, Guardian procedure FILE\_OPEN\_ for C and TAL, the OPEN verb for COBOL, or the #REQUESTER built-in function for the TACL program). Also, the application must specify the scheduler's process name in the form *\$scheduler-name.#ZSPI*; for example, \$ZBAT.#ZSPI. The qualifier #ZSPI indicates that the requester will be sending and receiving messages in SPI format. For examples of programmatic scheduler opens, see [Appendix C, Sample Programs](#).

When your application opens the scheduler, it must check for file-system errors in addition to the errors listed in [Table 2-1](#). For information on file-system errors, see the *Guardian Procedure Calls Reference Manual*.

**Table 2-1. Open Errors for the Scheduler**

| Error                            | Cause, Effect, and Recovery  |
|----------------------------------|--|
| 11 File or device does not exist | <p><b>Cause.</b> The scheduler did not recognize the process-name qualifier because the qualifier name was not #ZSPI.</p> <p><b>Effect.</b> The open attempt failed.</p> <p><b>Recovery.</b> Change the qualifier name to #ZSPI and retry the command.</p>   |
| 12 Maximum opens exceeded        | <p><b>Cause.</b> The scheduler exceeded its maximum number of opens (2000).</p> <p><b>Effect.</b> The open attempt failed.</p> <p><b>Recovery.</b> Try the open again later.</p>   |
| 16 File not opened               | <p><b>Cause.</b> The scheduler rejected the open attempt because it had not yet completed its own initialization when it received the open request.</p> <p><b>Effect.</b> The open attempt failed.</p> <p><b>Recovery.</b> Try the open again later.</p>   |
| 17 Error on backup open          | <p><b>Cause.</b> The scheduler rejected an attempted backup open because there was no matching primary open, the parameters for the backup open did not match those of the primary open, or the primary process was not running.</p> <p><b>Effect.</b> The open attempt failed.</p> <p><b>Recovery.</b> Open the scheduler by specifying its name.</p> |
| 48 Security violation            | <p><b>Cause.</b> The requester did not have the proper security to communicate with the scheduler.</p> <p><b>Effect.</b> The open attempt failed.</p> <p><b>Recovery.</b> Change the application's security to enable communication with the scheduler and retry the command.</p>  |

## Closing a Scheduler

When your application has finished communicating with the scheduler, it must close the scheduler using the mechanism appropriate to your programming language (for example, Guardian procedure `FILE_CLOSE_` for C and TAL, the `CLOSE` verb for COBOL, or the `#REQUESTER` built-in function for the TACL program). For examples showing programmatic scheduler closure, see [Appendix C, Sample Programs](#).

# Stopping a Scheduler

The NetBatch subsystem provides the ABORT SCHEDULER and SHUTDOWN SCHEDULER commands for stopping schedulers:

| Command            | Function  |
|--------------------|---|
| ABORT SCHEDULER    | Stops all executing and suspended processes associated with jobs, then stops the scheduler.   |
| SHUTDOWN SCHEDULER | Stops suspended processes associated with jobs, then stops the scheduler after allowing all executing processes associated with jobs to finish. |

For more information on these commands, see the *NetBatch Manual* and [Section 5, Commands and Responses](#).

## Running an EMS Consumer Distributor

Before your application can retrieve event messages, you must start an EMS consumer-distributor process, open the process for SPI communication, and specify the source of event messages with an EMS CONTROL command. For more information, see the *EMS Manual*.

To avoid receiving all event messages from all subsystems, load a filter to select only the messages your application is to act upon. You load your filter (written using the EMS filter language EMF) when you start the consumer distributor. For more information, see [Section 6, Event Messages](#).



# SPI Programming Considerations for the NetBatch Subsystem

The *SPI Programming Manual* provides general instructions for formatting commands and decoding responses and event messages for subsystems such as the NetBatch subsystem.

This section provides summary information and discusses SPI programming considerations that are specific to the NetBatch subsystem:

| <b>Topic</b>  | <b>Page</b>         |
|---|---------------------|
| <a href="#">Definition Files</a>                            | <a href="#">3-2</a> |
| <a href="#">Event-Message Template</a>                      | <a href="#">3-3</a> |
| <a href="#">Naming Guidelines for Applications</a>          | <a href="#">3-3</a> |
| <a href="#">Message Elements for the NetBatch Subsystem</a> | <a href="#">3-4</a> |

# Definition Files

The commands, responses, and event messages sent to and received from the NetBatch subsystem consist of items called tokens. Each token contains a particular piece of information, such as a command parameter or a detail of an event. Tokens can be single values or structures consisting of several values. Some tokens, called header tokens, are present in every command and response and in every event message.

Your management applications must declare tokens and related data items for commands, responses, and event messages. HP provides these declarations in definition files. A set of definition files in these languages comes with each NonStop subsystem that supports SPI: C, COBOL, DDL, TACL, and TAL. The C, COBOL, TACL, and TAL files derive from the DDL file.

To use the data declarations defined by a particular subsystem, your application must include the appropriate programming-language definition file associated with that subsystem. The declarations in a COBOL definition file are grouped into sections to enable COBOL programs to declare multiple copies of structures in the definition file. C and TAL programs can load either the entire definition file or just the sections they require. The TACL program always loads the entire definition file. For more information on how applications use definition files, see the *SPI Programming Manual*.

Definition files are named according to this convention:

```
ZSPIDEF.ZsubsysC  
ZSPIDEF.ZsubsysCOB  
ZSPIDEF.ZsubsysDDL  
ZSPIDEF.ZsubsysTACL  
ZSPIDEF.ZsubsysTAL
```

The last characters of each file name indicate the language in which the definitions in the file are coded.

*subsys*

is a three-character code identifying the subsystem or other software component to which the definition belongs. For example, BAT identifies NetBatch definitions.

Definition files are located on the disk volume chosen by your site. (The default disk volume used by the INSTALL system-generation program is \$SYSTEM.)

An application that sends SPI commands to and receives responses from the NetBatch subsystem requires these definition files in the appropriate programming language:

- The SPI (ZSPI) definition file
- The NetBatch (ZBAT) definition file

An application that retrieves event messages issued by the NetBatch subsystem requires the previously listed definition files plus the EMS (ZEMS) definition file.

For example, a management application written in TAL that sends commands to the NetBatch subsystem and retrieves event messages issued by that subsystem requires these SOURCE statements:

```
?SOURCE $vol.ZSPIDEF.ZBATTAL  
?SOURCE $vol.ZSPIDEF.ZEMSTAL  
?SOURCE $vol.ZSPIDEF.ZSPITAL
```

Likewise, a management application written in the C programming language that sends commands to the NetBatch subsystem and retrieves event messages issued by that subsystem requires these include statements:

```
#include "$vol.zspidef.zbatc"  
#include "$vol.zspidef.zemsc"  
#include "$vol.zspidef.zspic"
```

If your application manages other subsystems besides the NetBatch subsystem, it also requires the definition files required by those subsystems.

In NonStop manuals, definition refers to the data declarations in a definition file. For more information on SPI and EMS definitions, see the *SPI Programming Manual* and the *EMS Manual*. For more information on NetBatch definitions, see [Section 4, Common Definitions](#).

## Event-Message Template

The NetBatch subsystem comes with a DSM format template. This template lets the Guardian procedure EMSTEXT display scheduler event-message text in DSM display format. The template specifies which tokens of each message the procedure displays and the message text. Source and object files for the template are provide. You can change the source and recompile it if you want to customize your messages.

The names of the source and object files for the NetBatch template are, respectively, \$vol.ZTEMPL.SBATTMPL and \$vol.ZTEMPL.ZBATTMPL.

For more information on DSM format templates and instructions on how to use and change them, see the *DSM Template Services Manual*. For information specific to the NetBatch template, see [Section 6, Event Messages](#).

## Naming Guidelines for Applications

HP uses names beginning with the letter Z for definitions and component fields of structures in its definition files. To avoid having names you define conflict with names defined by HP, do not begin your names with a Z.

# Message Elements for the NetBatch Subsystem

This subsection provides subsystem-specific information about elements of NetBatch commands, responses, and event messages. For information on these elements, descriptions of elements whose meaning is not subsystem-specific (such as the subsystem ID), and SPI tokens and other definitions, see the *SPI Programming Manual*.

## Commands

The NetBatch subsystem supports these programmatic commands:

|          |            |          |           |
|----------|------------|----------|-----------|
| ABORT    | GETVERSION | SHUTDOWN | SUBMIT    |
| ACTIVATE | INFO       | START    | SUSPEND   |
| ADD      | RELEASE    | STATUS   | SWITCHCPU |
| ALTER    | RUNNEXT    | STOP     | SWITCHLOG |
| DELETE   | RUNNOW     |          |           |

Commands are identified by command numbers with symbolic names of the form ZBAT-CMD-*name* or ZSPI-CMD-*name*, where *name* identifies the command. (The separator character varies with the language. Hyphens appear in this and other DSM manuals because DDL uses hyphens.) For example, the symbolic name of the ALTER command is ZBAT-CMD-ALTER. Symbolic names represent the values that can be assigned to the command-number header token, ZSPI-TKN-COMMAND.

For descriptions of NetBatch subsystem commands and the subsystem's responses to them, see [Section 5, Commands and Responses](#).

## Object Types

The NetBatch subsystem supports these object types:

|         |          |           |
|---------|----------|-----------|
| ATT-SET | EXECUTOR | SCHEDULER |
| CLASS   | JOB      |           |

All commands and responses contain an object-type token in the header. For the NetBatch subsystem, object types are identified in programs by symbolic names of the form ZBAT-OBJ-*name*, where *name* identifies the object type. For example, ZBAT-OBJ-JOB represents the JOB object type. The object-type header token, ZSPI-TKN-OBJECT-TYPE, always has one of these ZBAT values.

A response consists of several response records, which are groups of tokens that give response information about a particular object.

## Event Numbers

All event messages contain a header token identifying the event by number. This number, with the subsystem-ID header token, uniquely identifies the event. Event numbers for NetBatch event messages are identified in programs by symbolic names

of the form ZBAT-EVT-*name*, where *name* identifies the event. For example, the event reporting that a job started is represented by the name ZBAT-EVT-JOB-START.

In NetBatch event messages, the event-number header token (ZEMS-TKN-EVENTNUMBER) can assume any of the set of NetBatch event numbers. For descriptions of NetBatch event messages, see [Section 6, Event Messages](#).

## Subjects of Event Messages

Each NetBatch event message contains a subject token, ZBAT-TKN-SCHEDULER-ID, which immediately follows the ZEMS-TKN-SUBJECT-MARK token. The subject token identifies the scheduler generating the event message. Only one subject token is present in each scheduler event message.

## Data Lists and Error Lists

Responses from the NetBatch subsystem can contain data lists and error lists, as described in the *SPI Programming Manual*. The response buffer might contain one or more data lists if the ZSPI-TKN-MAXRESP token is not zero.

## Using SPI to Build Commands and Decode Responses

The *SPI Programming Manual* provides detailed information on building and sending a command message and on receiving and decoding a response message. These subsections summarize the steps your application must take to perform each of these tasks and discuss NetBatch subsystem-specific programming considerations.

## Building and Sending a Command Message

To build and send a command message to the NetBatch subsystem:

1. Declare a buffer of appropriate size. NetBatch buffer declarations are:

| Buffer Declaration      | Buffer Size                      |
|-------------------------|----------------------------------|
| ZBAT-DDL-MSG-BUFFER     | ZBAT-VAL-BUFLen (2042 bytes)     |
| ZBAT-DDL-MSG-BUFFER-MAX | ZBAT-VAL-BUFLen-MAX (4090 bytes) |
| ZBAT-DDL-MSG-BUFFER-MIN | ZBAT-VAL-BUFLen-MIN (1018 bytes) |

For more information on NetBatch buffer declarations, see [Section 4, Common Definitions](#).

For sending commands, the size of the buffer you select must be passed as a parameter to SSINIT. You also must allocate a buffer of the chosen size in the program's data space or in an extended data segment.

2. Call procedure SSINIT, supplying the buffer, buffer length, subsystem ID, command, and object type. SSINIT initializes the buffer, placing the supplied information in the appropriate fields of the message header.
3. Call procedure SSNULL to initialize to null values the fields of each extensible structured token in the command.

- 
- △ **Caution.** An extensible structured token in a command must always be initialized by SSNULL. Using SSNULL to initialize an extensible structured token ensures every field of the token is initialized to its null value. This action is important because an operation will be performed if a field contains a value other than its null value.

Using SSNULL is important even when your application assigns a value to every field of an extensible structured token. If you do not use SSNULL, the application does not work correctly later if it is compiled with a new version of the definition files that add new fields to the token.

---

4. Call procedure SSPUT or SSPUTTKN to assign values to tokens and to add the tokens to the message.
5. Call procedure SSMOVE or SSMOVETKN—if you are resending a command to retrieve the next response message in a series—to move the context token from the previous response buffer into the command buffer.
6. Send the command message using the mechanism applicable to your programming language (for example, Guardian procedure WRITEREADX for C and TAL, the READ verb for COBOL, or the #REQUESTER built-in function for the TACL program). The NetBatch subsystem receives the message, interprets the command request, executes the command (if there are no command-syntax errors), and returns a response buffer, including one or more error lists (if any errors occurred).

Because the mechanism your application uses to send the command buffer to the NetBatch subsystem is independent of SPI, you can use features available to your programming language, such as nowaited or timed I/O for TAL.

Your application must check for file-system errors when sending the buffer. For file-system error details, see the *Guardian Procedure Errors and Messages Manual*.

## Discontinuing a Command in Progress

The NetBatch subsystem does not respond to requests for discontinuing a command in progress. The subsystem does not support the use of the standard SPI token ZSPI-TKN-ALLOW-TYPE, which in other subsystems allows applications to specify, in a command operating on multiple objects, whether the subsystem is to continue to the next object if it failed on a previous one. The NetBatch subsystem continues to the next object if it fails on the previous object.

## Receiving and Decoding a Response Message

To receive and decode a response message from the NetBatch subsystem:

1. Declare a buffer of appropriate size. NetBatch buffer declarations are:

| Buffer Declaration      | Buffer Size                       |
|-------------------------|-----------------------------------|
| ZBAT-DDL-MSG-BUFFER     | ZBAT-VAL-BUFLLEN (2042 bytes)     |
| ZBAT-DDL-MSG-BUFFER-MAX | ZBAT-VAL-BUFLLEN-MAX (4090 bytes) |
| ZBAT-DDL-MSG-BUFFER-MIN | ZBAT-VAL-BUFLLEN-MIN (1018 bytes) |

For more information on NetBatch buffer declarations, see [Section 4, Common Definitions](#).

2. Read the response message using the mechanism applicable to your programming language (for example, Guardian procedure WRITEREADX for C and TAL, the READ verb for COBOL, or the #REQUESTER built-in function for the TACL program).
3. Call procedure SSGET or SSGETTKN to extract tokens and related information from the buffer.
4. Call procedure SSMOVE or SSMOVETKN—if the buffer contains the ZSPI-TKN-CONTEXT token—to copy the context token into the original command buffer, and resend the command to get the next message in the response or to complete the command.
5. Take action appropriate to the information in the response.

The NetBatch subsystem returns multiple response records in its responses to some commands, as described in the *SPI Programming Manual*.

## Error Handling

Each NetBatch subsystem response includes a return token (ZSPI-TKN-RETCODE) whose value indicates whether an error occurred when the subsystem tried to perform the command. If the command completed with no errors, the value of the return token is zero. Some NetBatch error replies contain information about the error, in addition to the error number in ZSPI-TKN-RETCODE. This additional information is enclosed in an SPI error list (ZSPI-TKN-ERRLIST) and does not use extensible structured tokens.

If the value of ZSPI-TKN-RETCODE is not zero, the command might have failed. Where a nonzero return code is present, the response buffer might contain one or more error lists. Even if the return code is zero, the response might still contain error lists that describe warnings. A warning reports a condition less serious than an error. If no errors occurred but a warning did occur, an error list appears in the response, providing information about the warning condition.

Your applications must always check error lists, regardless of the RETCODE values.

## Version Compatibility

Versions of the NetBatch subsystem that support SPI and EMS are:

|     |                        |
|-----|------------------------|
| SPI | NetBatch D21 and later |
| EMS | NetBatch D20 and later |

To ensure upward compatibility between your applications and later versions of the NetBatch subsystem, the applications must comply with the SPI and EMS programming guidelines in the *SPI Programming Manual* and the *EMS Manual*.

## Security

NetBatch commands are either sensitive or nonsensitive:

- Sensitive commands affect the configuration or state of objects and are usually restricted to NetBatch supervisors (users with execute access to the NETBATCH program file).
- Nonsensitive commands do not affect the configuration or state of objects and are available to all users.

For more information, see [Section 5, Commands and Responses](#).

## File Names

File names specified in NetBatch SPI commands must have the volume, subvolume, and file ID components specified. (If the node is not specified, the scheduler uses the requester's node.)

- The scheduler returns a fully qualified file name in a response message except when the volume is eight characters long (including the dollar sign). In that case, only the volume, subvolume, and file ID components are returned.
- The scheduler rejects file names that do not have the volume, subvolume, and file ID components specified. It also rejects remote file names containing eight-character volume names.

## Displaying SPI Traffic

To display SPI messages that the scheduler receives from and sends to requesters such as BATCHCOM, do one of:

- Include the DISPLAY-SPI parameter in the RUN NETBATCH command. For example:

```
37> NETBATCH /NAME $ZBAT, NOWAIT/ $DATA7.ZBAT DISPLAY-SPI
```

The parameter causes the scheduler to write to its log file the contents of each command buffer received and each response buffer sent.



- Use the BATCHCOM command DISPLAY-SPI. The command makes BATCHCOM display the contents of each command buffer sent and each response buffer received. For more information, see the *NetBatch Manual*.

Message-buffer contents appear in this format:

```
-----
--
|_SPI_BUFFER_ { BEING_SENT_TO | RETURNED_FROM }
|_SCHEDULER_ |
-----
--
        Checksum: zspi-tkn-checksum
        Header Type: zspi-tkn-hdrtype
        Last error: zspi-tkn-lasterr
        Last error code: zspi-tkn-lasterrcode
        Last Position: zspi-tkn-lastposition
        Max Field Version: zspi-tkn-max-field-version
        Maxresp: zspi-tkn-maxresp
        Position: zspi-tkn-position
        Server version: zspi-tkn-server-version
        Subsystem ID: zspi-tkn-ssid
        Used length: zspi-tkn-usedlen
command-name ( command-name-value ) object-type ( object-
type-value )
[ TDT: token-data-type: { VAR | size }; token-code x count
[ size: ] token-value ]...
```

For details of *zspi-tkn*-header tokens, see the *SPI Programming Manual*. Details of other items in the display are:

*command-name*

is a NetBatch command name (that is, the value of *name* in ZBAT-CMD-*name*).

*command-name-value*

is the numeric value of *command-name* (that is, the command number).

*object-type*

is a NetBatch object type (that is, the value of *name* in ZBAT-OBJ-*name*).

*object-type-value*

is the numeric value of *object-type* (that is, the *object-type* number).

*token-data-type*

is the data type of the token (that is, the value of *type* in ZSPI-TDT-*type*).

VAR

indicates the token is a variable-length token. The size of the token's value in bytes is given in *size:token-value*.

*size*

is the token size in bytes (when *size* appears in *token-data-type:size*) or the size of the token's value in bytes (when *size* appears in *size:token-value*). *size* appears with *token-data-type* when the token is not a variable-length token. *size* appears with *token-value* when the token is a variable-length token.

*token-code*

is the value of *code* in ZSPI-TNM-*code* or ZBAT-TNM-*code*.

*count*

is the number of times *token-code* occurs in the message.

*token-value*

is the token value. How the value is displayed depends on the token data type:

- If the type is BYTE, CHAR, DEVICE, FNAME, STRUCT, or SUBVOL and the value is printable, then that value appears. For example, the name TWENTY-FOUR-CHARACTERS-X in ZBAT-TKN-NETBATCH-NAME appears as:

```
TDT: CHAR:24; NETBATCH-NAMEx1 TWENTY-FOUR-CHARACTERS-X
```

If the value is printable and includes spaces, the value is appended with information in this format:

```
? [ count x ] binary-value
```

*count*

specifies the number of occurrences of *binary-value*. *count* appears only when *binary-value* occurs more than once.

*binary-value*

is the binary value of a space (32).

For example, the 19-character and 23-character names NINETEEN-CHARACTERS and TWENTY-THREE-CHARACTERS in the 24-character token ZBAT-TKN-NETBATCH-NAME appear as:

```
TDT: CHAR:24; NETBATCH-NAMEx1 NINETEEN-CHARACTERS ?5x32
```

```
TDT: CHAR:24; NETBATCH-NAMEx1 TWENTY-THREE-CHARACTERS ?32
```

If the value includes printable and unprintable characters, or is unprintable, the display format is:

```
bytes: ? [ count x ] binary-value
[ ? [ count x ] binary-value ]... [ printable-chars ]...
```

*bytes*

is the number of bytes in the value.

*count*

specifies the number of occurrences of *binary-value*. *count* appears only when *binary-value* occurs more than once.

*binary-value*

is the binary value of the character (not necessarily a space).

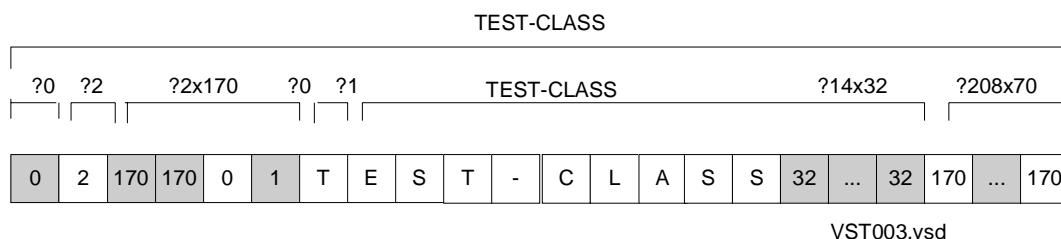
*printable-chars*

represents printable characters.

For example, the value of extensible structured token ZBAT-MAP-DEF-EXECUTOR in an ADD EXECUTOR command specifying class TEST-CLASS appears as:

```
TDT: STRUCT:VAR; DEF-EXECUTORx1
238: ?0 ?2 ?2x170 ?0 ?1 TEST-CLASS ?14x32 ?208x170
```

The interpretation of 238: ?0 ?2 ?2x170 ?0 ?1 TEST-CLASS ?14x32 ?208x170 from the example is:



- If the type is BOOLEAN, ENUM, INT, INT2, or UINT, the actual value appears. For example, a zero value for ZSPI-TKN-RETCODE appears as:

```
TDT: ENUM:2; RETCODEx1 0000
```

- If the type is DATALIST, ENDLIST, or ERRLIST, then DATALIST, END LIST, or ERROR LIST appears as appropriate with details of the tokens in the list. For example, tokens in the data list returned by the scheduler in response to a START EXECUTOR command that starts executor NINETEEN-CHARACTERS appear as:

```
START DATA LIST
  TDT: CHAR:VAR; SEL-EXECUTORNAMEx1
  19:NINETEEN-CHARACTERS
  TDT: ENUM:2; RETCODEx1 0000
END LIST
```

## Example

A scheduler with DISPLAY-SPI logging enabled logs these message-buffer details to its log file in response to a DELETE EXECUTOR command that deletes executor EXEC-01:

```
-----
|_SPI_BUFFER_SENT_TO_SCHEDULER_|
-----
      Checksum: 0
      Header Type: 0
      Last error: 0
      Last error code: tkn 29/4/-506
      Last Position: SSCTL 8 -439
      Max Field Version: 0
      Maxresp: -1
      Position: SSCTL 8 -442
      Server version: 17438 D30
      Subsystem ID: TANDEM.9.D30
      Used length: 69
DELETE (260)EXECUTOR (512)
TDT: CHAR:VAR; SEL-EXECUTORNAMEx1
7:EXEC-01
-----
|_SPI_BUFFER_RETURNED_FROM_SCHEDULER_|
-----
      Checksum: 0
      Header Type: 0
      Last error: 0
      Last error code: tkn 29/4/-506
      Last Position: SSCTL 8 -439
      Max Field Version: 0
      Maxresp: -1
      Position: SSCTL 8 -442
      Server version: 17438 D30
      Subsystem ID: TANDEM.9.D30
      Used length: 86
DELETE (260)EXECUTOR (512)
START DATA LIST
  TDT: CHAR:VAR; SEL-EXECUTORNAMEx1
  7:EXEC-01
  TDT: ENUM:2; RETCODEx1 0000
END LIST
```

## Retrieving and Decoding Event Messages

For detailed information on event-message retrieval and decoding, and for examples of event-message retrieval in the C, COBOL, TACL, and TAL languages, see the *EMS Manual*.

To retrieve and decode event messages from the NetBatch subsystem:

1. Declare a buffer of appropriate size for the EMS GETEVENT command and its response. For recommended buffer sizes, see the *EMS Manual*.
2. Start an EMS consumer distributor and open it specifying the #ZSPI qualifier in the process name.
3. Perform these tasks:
  - a. Format an EMS distributor CONTROL programmatic command to load a filter you have written and to specify the source and destination of event messages, if desired. (You use filters to select the event messages you want your application to process. Filters are discussed further at the end of this section.)
  - b. Send the CONTROL command to the consumer distributor, using the mechanism applicable to your programming language (for example, Guardian procedure WRITEREADX for C and TAL, the READ verb for COBOL, or the #REQUESTER built-in function for the TACL program).
4. Read the response from the distributor, using the mechanism applicable to your programming language.
5. For each event message:
  - a. Format and send a GETEVENT command to the consumer distributor to get the next event message, using the mechanism applicable to your programming language.
  - b. Read the response from the distributor, using the mechanism applicable to your programming language.
  - c. Call procedure SSMOVE or SSMOVETKN to move the context token from the response buffer into the GETEVENT command buffer. The context token will be required later, when your application resends the command to get the next event message.
  - d. Call procedure SSGET or SSGETTKN to retrieve the token containing the event message (ZEMS-TKN-EVENT) from the response buffer.
  - e. Call procedure EMSGET or EMSGETTKN to retrieve the subsystem ID (ZSPI-TKN-SSID) and the event number (ZEMS-TKN-EVENTNUMBER) from the event message. Together, these two tokens identify the event message and determine what information tokens it contains.
  - f. Call procedure EMSGET or EMSGETTKN twice to retrieve the subject of the event message. In the first call, retrieve the subject-mark token (ZEMS-TKN-

SUBJECT) to get the token code and index of the token identifying the subject. Then make another call to retrieve the subject token itself.

- g. Call procedure EMSGET or EMSGETTKN to retrieve the values of other tokens from the event message.
- h. Take action appropriate to the information in the event message.

For information on retrieving tokens from an event message, see the *EMS Manual*.

## Critical Events

Critical events can have serious consequences, such as scheduler failure. The value of event-message token ZEMS-TKN-EMPHASIS determines whether an event is critical. If the value is ZSPI-VAL-TRUE, the event is critical. If the value is ZSPI-VAL-FALSE, the event is noncritical.

For a list of critical and noncritical NetBatch event messages, see [Section 6, Event Messages](#).

## Filters

EMS provides you with the ability to create filters, which let applications select particular event messages from among those reported. Filters select messages to return to an application by examining the values of tokens in the messages. For example, to select only NetBatch event messages, a filter would examine the token that contains the subsystem ID of the issuing subsystem and then let only messages containing the NetBatch subsystem ID pass to the application that requested them.

You can create filter source files and compile them into filter object files by using EMF, the EMS filter language. You can load the resulting filters into consumer, printing, or forwarding distributors by using the EMS program EMSDIST. Also, you can replace filters online.

You can use any of the tokens contained in event messages to select which event messages will be returned to your application. You can create filters that return only critical event messages, all messages with a certain event number, and so on.

For general information on coding, compiling, loading, and replacing filters, see the *EMS Manual*. For information specific to NetBatch subsystem filters, see [Section 6, Event Messages](#).

# 4

## Common Definitions

This section discusses SPI standard definitions, EMS standard definitions, and NetBatch definitions used in the token-oriented programmatic interfaces to the NetBatch subsystem:

| Topic                                    | Page                |
|--|---------------------|
| <a href="#">SPI Standard Definitions</a> | <a href="#">4-2</a> |
| <a href="#">EMS Standard Definitions</a> | <a href="#">4-6</a> |
| <a href="#">NetBatch Definitions</a>     | <a href="#">4-8</a> |

This section provides general information that applies to all uses of the definitions in the interfaces to the NetBatch subsystem. Information on a definition that is specific to a particular command, response, event message, or error list appears in the description of that command, response, event message, or error list.

Definitions in this section appear in DDL format. Definitions of structures use DDL definition statements. For an explanation of DDL, see the *Data Definition Language (DDL) Reference Manual*.

# SPI Standard Definitions

Definitions whose names begin with ZSPI- are SPI standard definitions. These definitions, which are available to all subsystems that support SPI procedures, are found in the ZSPIDEF.ZSPIDDL file and in the corresponding files for other languages.

[Table 4-1](#) through [Table 4-6](#) list the SPI standard definitions used in the NetBatch subsystem's programmatic interfaces. The definitions are for:

- Header tokens
- Other simple tokens
- Special tokens
- Token types
- Structures
- Value names

The tables do not list SPI error numbers or definitions used only in error lists associated with those error numbers, nor do they list all SPI token values an application can use.

For descriptions of SPI standard definitions, see the *SPI Programming Manual*. Information on the definitions that is specific to the NetBatch subsystem follows [Table 4-6](#) on page 4-3.

---

**Table 4-1. SPI Standard Definitions for Header Tokens**

|                       |                            |
|-----------------------|----------------------------|
| ZSPI-TKN-BUFLEN       | ZSPI-TKN-MAX-FIELD-VERSION |
| ZSPI-TKN-CHECKSUM     | ZSPI-TKN-MAXRESP           |
| ZSPI-TKN-COMMAND      | ZSPI-TKN-OBJECT-TYPE       |
| ZSPI-TKN-HDRTYPE      | ZSPI-TKN-POSITION          |
| ZSPI-TKN-LASTERR      | ZSPI-TKN-SERVER-VERSION    |
| ZSPI-TKN-LASTERRCODE  | ZSPI-TKN-SSID              |
| ZSPI-TKN-LASTPOSITION | ZSPI-TKN-USEDLEN           |

---

**Table 4-2. SPI Standard Definitions for Other Simple Tokens**

|                   |                        |
|-------------------|------------------------|
| ZSPI-TKN-COMMENT  | ZSPI-TKN-ERROR         |
| ZSPI-TKN-CONTEXT  | ZSPI-TKN-RESPONSE-TYPE |
| ZSPI-TKN-DATALIST | ZSPI-TKN-RETCODE       |
| ZSPI-TKN-ENDLIST  | ZSPI-TKN-SERVER-BANNER |
| ZSPI-TKN-ERRLIST  |                        |



**Table 4-3. SPI Standard Definitions for Special Tokens**


---

|                       |                           |
|-----------------------|---------------------------|
| ZSPI-TKN-ADDR         | ZSPI-TKN-INITIAL-POSITION |
| ZSPI-TKN-CLEARERR     | ZSPI-TKN-LEN              |
| ZSPI-TKN-COUNT        | ZSPI-TKN-NEXTCODE         |
| ZSPI-TKN-DATA-FLUSH   | ZSPI-TKN-NEXTTOKEN        |
| ZSPI-TKN-DEFAULT-SSID | ZSPI-TKN-OFFSET           |
| ZSPI-TKN-DELETE       | ZSPI-TKN-RESET-BUFFER     |

---

**Table 4-4. SPI Standard Definitions for Token Types**


---

|                     |                    |
|---------------------|--------------------|
| ZSPI-TYP-BOOLEAN    | ZSPI-TYP-INT2      |
| ZSPI-TYP-BYTESTRING | ZSPI-TYP-INT4      |
| ZSPI-TYP-CHAR       | ZSPI-TYP-PHANDLE   |
| ZSPI-TYP-CHAR24     | ZSPI-TYP-STRING    |
| ZSPI-TYP-ENUM       | ZSPI-TYP-TIMESTAMP |
| ZSPI-TYP-INT        | ZSPI-TYP-USERID    |

---

**Table 4-5. SPI Standard Definitions for Structures**


---

|                  |                  |
|------------------|------------------|
| ZSPI-DDL-BOOLEAN | ZSPI-DDL-INT     |
| ZSPI-DDL-BYTE    | ZSPI-DDL-INT2    |
| ZSPI-DDL-CHAR6   | ZSPI-DDL-INT4    |
| ZSPI-DDL-CHAR8   | ZSPI-DDL-PHANDLE |
| ZSPI-DDL-CHAR16  | ZSPI-DDL-UINT    |
| ZSPI-DDL-CHAR24  |                  |

---

**Table 4-6. SPI Standard Definitions for Value Names**


---

|                       |                            |
|-----------------------|----------------------------|
| ZSPI-SSN-ZBAT         | ZSPI-VAL-ERR-WARN-AND-NORM |
| ZSPI-VAL-ERR-AND-WARN | ZSPI-VAL-TANDEM            |

---

## Information About SPI Standard Definitions Specific to the NetBatch Subsystem

The *SPI Programming Manual* fully describes all SPI standard definitions. Information about the definitions that is specific to the NetBatch subsystem is:

ZSPI-SSN-ZBAT

is the NetBatch subsystem number (9).

ZSPI-TKN-COMMAND

contains a command number of a NetBatch command. The value of this token is always one of these enumerated ZBAT-CMD-*name* or ZSPI-CMD-*name* values of ZBAT-DDL-COMMAND:

|           | <i>name</i> |         |          |           |
|-----------|-------------|---------|----------|-----------|
| ZBAT-CMD- | ABORT       | INFO    | SHUTDOWN | SUBMIT    |
|           | ACTIVATE    | RELEASE | START    | SUSPEND   |
|           | ADD         | RUNNEXT | STATUS   | SWITCHCPU |
|           | ALTER       | RUNNOW  | STOP     | SWITCHLOG |
|           | DELETE      |         |          |           |

ZSPI-CMD- GETVERSION

For a description of ZBAT-DDL-COMMAND, see [Private Token and Field Types](#) on page 4-13. For descriptions of NetBatch commands, see [Section 5, Commands and Responses](#).

ZSPI-TKN-ERROR

is the error token. Its value consists of the NetBatch subsystem ID and an error number describing the error. For descriptions of NetBatch error numbers and their associated error lists, see [Appendix A, Error Numbers and Error Lists](#).

ZSPI-TKN-OBJECT-TYPE

contains an object-type number of a NetBatch object. The value of this token is always one of these enumerated ZBAT-OBJ-*name* values of ZBAT-DDL-OBJECT. The object types associated with a command vary with each command.

|           | <i>name</i> |          |     |           |
|-----------|-------------|----------|-----|-----------|
| ZBAT-OBJ- | ATT-SET     | EXECUTOR | JOB | SCHEDULER |
|           | CLASS       |          |     |           |

The object type of command ZSPI-CMD-GETVERSION is ZSPI-VAL-NULL-OBJECT-TYPE.

For a description of ZBAT-DDL-OBJECT, see [Private Token and Field Types](#) on page 4-13. For information on the object types that are valid for specific commands, see [Section 5, Commands and Responses](#).

**ZSPI-TKN-RETCODE**

is the return token that the NetBatch subsystem returns in a response message. If the token contains a value other than zero, an error occurred. In this circumstance, the message also contains an error list describing the error.

**ZSPI-TKN-SERVER-BANNER**

contains the server-banner string for the NetBatch subsystem and is returned only for the GETVERSION command. An example of the NetBatch server-banner string is:

```
NETBATCH SERVER - T9190D30 - (31OCT94-D30)
```

**ZSPI-TKN-SERVER-VERSION**

specifies the server version of the NetBatch subsystem.

**ZSPI-TKN-SSID**

contains ZBAT-VAL-SSID, the subsystem ID of the NetBatch subsystem. ZBAT-VAL-SSID has the structure:

|                          |                         |
|--------------------------|-------------------------|
| Definition ZBAT-VAL-SSID | Tacl SSID.              |
| 02 Z-FILL                | Type Character 8        |
|                          | VALUE ZSPI-VAL-TANDEM.  |
| 02 Z-OWNER               | Redefines Z-FILL        |
|                          | Type ZSPI-DDL-CHAR8.    |
| 02 Z-NUMBER              | Type ZSPI-DDL-INT       |
|                          | VALUE ZSPI-SSN-ZBAT.    |
| 02 Z-VERSION             | Type ZSPI-DDL-UINT      |
|                          | VALUE ZBAT-VAL-VERSION. |
| End                      |                         |

# EMS Standard Definitions

Definitions whose names begin with ZEMS- are EMS standard definitions. These definitions, which are available to all subsystems that support EMS, are found in the ZSPIDEF.ZEMSDDL file and in the corresponding files for other languages.

[Table 4-7](#) and [Table 4-8](#) list the EMS standard definitions used in the NetBatch subsystem’s programmatic interfaces. These definitions comprise:

- Header tokens
- A data-portion token

The tables do not list EMS error numbers or definitions used only in error lists.

For descriptions of EMS standard definitions, see the *EMS Manual*. Information on the definitions that is specific to the NetBatch subsystem follows [Table 4-8](#).

**Table 4-7. EMS Standard Definitions for Header Tokens**

|                   |                      |
|-------------------|----------------------|
| ZEMS-TKN-EMPHASIS | ZEMS-TKN-EVENTNUMBER |
|-------------------|----------------------|

**Table 4-8. EMS Standard Definition for Data-Portion Token**

|                       |
|-----------------------|
| ZEMS-TKN-SUBJECT-MARK |
|-----------------------|

## NetBatch-Specific Information About EMS Standard Definitions

The *EMS Manual* fully describes all EMS standard definitions. Information about the definitions that is specific to the NetBatch subsystem is:

ZEMS-TKN-EVENTNUMBER

is the number the scheduler assigns to an event to identify it. The token can have one of these values:

| ZEMS-TKN-EVENTNUMBER | Event                      |
|----------------------|----------------------------|
| 100                  | ZBAT-EVT-SCHEDULER-START   |
| 101                  | ZBAT-EVT-SCHEDULER-STOP    |
| 102                  | ZBAT-EVT-JOB-START         |
| 200                  | ZBAT-EVT-EXECUTOR-DOWN     |
| 201                  | ZBAT-EVT-EXECUTOR-UP       |
| 202                  | ZBAT-EVT-JOB-NORMAL-STOP   |
| 203                  | ZBAT-EVT-JOB-ABNORMAL-STOP |
| 204                  | ZBAT-EVT-JOB-OVER-LIMIT    |
| 301                  | ZBAT-EVT-JOB-START-ERROR   |
| 500                  | ZBAT-EVT-SCHEDULER-ABENDED |

**ZEMS-TKN-EMPHASIS**

specifies whether an event is critical (as defined in the *EMS Manual*). The token can have one of these values:

| <b>Event</b>                    | <b>Critical?</b> | <b>ZEMS-TKN-EMPHASIS</b> |
|---------------------------------|------------------|--------------------------|
| 100: ZBAT-EVT-SCHEDULER-START   | No               | ZSPI-VAL-FALSE           |
| 101: ZBAT-EVT-SCHEDULER-STOP    | No               | ZSPI-VAL-FALSE           |
| 102: ZBAT-EVT-JOB-START         | No               | ZSPI-VAL-FALSE           |
| 200: ZBAT-EVT-EXECUTOR-DOWN     | No               | ZSPI-VAL-FALSE           |
| 201: ZBAT-EVT-EXECUTOR-UP       | No               | ZSPI-VAL-FALSE           |
| 202: ZBAT-EVT-JOB-NORMAL-STOP   | No               | ZSPI-VAL-FALSE           |
| 203: ZBAT-EVT-JOB-ABNORMAL-STOP | No               | ZSPI-VAL-FALSE           |
| 204: ZBAT-EVT-JOB-OVER-LIMIT    | No               | ZSPI-VAL-FALSE           |
| 301: ZBAT-EVT-JOB-START-ERROR   | No               | ZSPI-VAL-FALSE           |
| 500: ZBAT-EVT-SCHEDULER-ABENDED | Yes              | ZSPI-VAL-TRUE            |

**ZEMS-TKN-SUBJECT-MARK**

marks the token ZBAT-TKN-SCHEDULER-ID in the event message buffer as the subject of the event message.

# NetBatch Definitions

Definitions whose names begin with ZBAT- are NetBatch definitions. They are found in file ZSPIDEF.ZBATDDL and in the corresponding files for other languages.

[Table 4-9](#) through [Table 4-13](#) list NetBatch definitions. These definitions comprise:

- Buffer declarations
- Private token and field types
- Predefined token and field values
- Simple tokens
- Extensible structured tokens

General descriptions of the definitions appear in the subsections following the tables. Information about a definition that is specific to a particular command or response appears in [Section 5, Commands and Responses](#).

---

**Table 4-9. NetBatch Buffer Declarations**

|                         |                         |
|-------------------------|-------------------------|
| ZBAT-DDL-MSG-BUFFER     | ZBAT-DDL-MSG-BUFFER-MIN |
| ZBAT-DDL-MSG-BUFFER-MAX |                         |

---

**Table 4-10. NetBatch Private Token and Field Types** (page 1 of 2)

|                          |                           |
|--------------------------|---------------------------|
| ZBAT-DDL-COMMAND         | ZBAT-DDL-RETCODE          |
| ZBAT-DDL-COMPLETION-CODE | ZBAT-DDL-SCHEDULER-STATE  |
| ZBAT-DDL-DEF-CLASS       | ZBAT-DDL-SPECIAL-REASON   |
| ZBAT-DDL-DEF-CRONTAB     | ZBAT-DDL-STATUS-EXECUTOR  |
| ZBAT-DDL-DEF-EXECUTOR    | ZBAT-DDL-STATUS-JOB       |
| ZBAT-DDL-DEF-JOB         | ZBAT-DDL-STATUS-SCHEDULER |
| ZBAT-DDL-DEF-SCHEDULER   | ZBAT-DDL-WAITON-FOR       |
| ZBAT-DDL-DEF-WAITON      | ZBAT-DDL-WAITON-INDICATOR |
| ZBAT-DDL-DEFINE-ERROR    |                           |
| ZBAT-DDL-EMS             | ZBAT-TYP-CHAR6            |
| ZBAT-DDL-EXECUTOR-LIST   | ZBAT-TYP-COMMAND          |
| ZBAT-DDL-INT2-TRIO       | ZBAT-TYP-COMPLETION-CODE  |
| ZBAT-DDL-JOB-NUMBER      | ZBAT-TYP-INT2-TRIO        |
| ZBAT-DDL-JOB-WHICH-LIST  | ZBAT-TYP-JOB-NUMBER       |
| ZBAT-DDL-LIST            | ZBAT-TYP-LIST             |
| ZBAT-DDL-NETBATCH-NAME   | ZBAT-TYP-NETBATCH-NAME    |
| ZBAT-DDL-OBJECT          | ZBAT-TYP-OBJECT           |

---

**Table 4-10. NetBatch Private Token and Field Types** (page 2 of 2)

|                          |                    |
|--------------------------|--------------------|
| ZBAT-DDL-PAR-RELEASE-JOB | ZBAT-TYP-PC-ERROR0 |
| ZBAT-DDL-PC-ERROR0       | ZBAT-TYP-PC-ERROR1 |
| ZBAT-DDL-PC-ERROR1       | ZBAT-TYP-REASON    |
| ZBAT-DDL-REASON          | ZBAT-TYP-RETCODE   |

**Table 4-11. NetBatch Token Predefined Token and Field Values** (page 1 of 2)

|                               |                                |
|-------------------------------|--------------------------------|
| ZBAT-VAL-BUFLEN               | ZBAT-VAL-READY                 |
| ZBAT-VAL-BUFLEN-MAX           | ZBAT-VAL-RESTART-OFF           |
| ZBAT-VAL-BUFLEN-MIN           | ZBAT-VAL-RESTART-ON            |
| ZBAT-VAL-CALENDAR-EMPTY       | ZBAT-VAL-RUNNEXT               |
| ZBAT-VAL-CALENDAR-ERROR       | ZBAT-VAL-RUNNOW                |
| ZBAT-VAL-EMS-ERROR            | ZBAT-VAL-SCHEDULER-ZNOTSTARTED |
| ZBAT-VAL-EMS-OFF              | ZBAT-VAL-SCHEDULER-ZSHUTDOWN   |
| ZBAT-VAL-EMS-ON               | ZBAT-VAL-SCHEDULER-ZSTARTED    |
| ZBAT-VAL-EVENT                | ZBAT-VAL-SPECIAL-1             |
| ZBAT-VAL-EXECUTING            | ZBAT-VAL-SPECIAL-2             |
| ZBAT-VAL-EXECUTOR-ACTIVE-LIST | ZBAT-VAL-SPECIAL-3             |
| ZBAT-VAL-EXECUTOR-DELETE-LIST | ZBAT-VAL-SPECIAL-4             |
| ZBAT-VAL-EXECUTOR-DOWN-LIST   | ZBAT-VAL-SPECIAL-5             |
| ZBAT-VAL-EXECUTOR-OFF-LIST    | ZBAT-VAL-SPECIAL-6             |
| ZBAT-VAL-EXECUTOR-ON-LIST     | ZBAT-VAL-SPECIAL-7             |
| ZBAT-VAL-EXECUTOR-STOP-LIST   | ZBAT-VAL-SPECIAL-8             |
| ZBAT-VAL-EXTERNAL-SSID        | ZBAT-VAL-SPECIAL-9             |
| ZBAT-VAL-FAIL-AFTER-CREATE    | ZBAT-VAL-SPECIAL-ANY           |
| ZBAT-VAL-FIRST-LIST           | ZBAT-VAL-SSID                  |
| ZBAT-VAL-HOLD-ON              | ZBAT-VAL-STALL                 |
| ZBAT-VAL-JOB-EVENT-LIST       | ZBAT-VAL-SUSPENDED             |
| ZBAT-VAL-JOB-EXECUTING-LIST   | ZBAT-VAL-TAPE                  |
| ZBAT-VAL-JOB-READY-LIST       | ZBAT-VAL-TIME                  |
| ZBAT-VAL-JOB-RUNNEXT-LIST     | ZBAT-VAL-VERSION               |
| ZBAT-VAL-JOB-RUNNOW-LIST      | ZBAT-VAL-WAITON-RELEASED-OK    |
| ZBAT-VAL-JOB-SPECIAL-LIST     | ZBAT-VAL-WAITON-REMOVE         |
| ZBAT-VAL-JOB-SUSPENDED-LIST   | ZBAT-VAL-WAITON-RESET          |
| ZBAT-VAL-JOB-TAPE-LIST        | ZBAT-VAL-WAITON-SET            |

**Table 4-11. NetBatch Token Predefined Token and Field Values** (page 2 of 2)

|                           |                           |
|---------------------------|---------------------------|
| ZBAT-VAL-JOB-TIME-LIST    | ZBAT-VAL-WAITON-STOP      |
| ZBAT-VAL-LAST-LIST        | ZBAT-VAL-WAITON-STOPABEND |
| ZBAT-VAL-NEWPROCESS-ERROR | ZBAT-VAL-WAS-RUNNING      |

**Table 4-12. NetBatch Simple Tokens** (page 1 of 2)

|                           |                            |
|---------------------------|----------------------------|
| ZBAT-TKN-ATT-SET-ASSIGN   | ZBAT-TKN-ATT-SET-TEMPORARY |
| ZBAT-TKN-ATT-SET-DEFINE   | ZBAT-TKN-BATCHCTL          |
| ZBAT-TKN-ATT-SET-ID       | ZBAT-TKN-BYTESTRING        |
| ZBAT-TKN-ATT-SET-PARAM    | ZBAT-TKN-CALENDAR          |
| ZBAT-TKN-ATT-SET-SECURITY | ZBAT-TKN-CHAR6             |
| ZBAT-TKN-COMMAND          | ZBAT-TKN-SEL-CLASSNAME     |
| ZBAT-TKN-COMPLETION-CODE  | ZBAT-TKN-SEL-DEFINE-NAME   |
| ZBAT-TKN-DATA-BASE        | ZBAT-TKN-SEL-EXECUTORNAME  |
| ZBAT-TKN-DESCRIPTION      | ZBAT-TKN-SEL-INNAME        |
| ZBAT-TKN-EXECUTOR-ID      | ZBAT-TKN-SEL-JOB-NUMBER    |
| ZBAT-TKN-EXECUTOR-PROGRAM | ZBAT-TKN-SEL-JOBNAME       |
| ZBAT-TKN-EXTSWAP-FILE     | ZBAT-TKN-SEL-LIST          |
| ZBAT-TKN-FORMATSUBJECT    | ZBAT-TKN-SEL-NETBATCH-NAME |
| ZBAT-TKN-IN-FILE          | ZBAT-TKN-SEL-NOTADPNAME    |
| ZBAT-TKN-INT              | ZBAT-TKN-SEL-NOTCLASSNAME  |
| ZBAT-TKN-INT2             | ZBAT-TKN-SEL-NOTINNAME     |
| ZBAT-TKN-JOB-ID           | ZBAT-TKN-SEL-NOTJOBNAME    |
| ZBAT-TKN-JOB-NAME-ID      | ZBAT-TKN-SEL-NOTLIST       |
| ZBAT-TKN-JOB-NUMBER       | ZBAT-TKN-SEL-NOTUSERNAME   |
| ZBAT-TKN-LIB-FILE         | ZBAT-TKN-SEL-NOTWAITON     |
| ZBAT-TKN-LOG-FILE         | ZBAT-TKN-SEL-PARAM-NAME    |
| ZBAT-TKN-MIN-MAX-ERROR    | ZBAT-TKN-SEL-USERNAME      |
| ZBAT-TKN-NETBATCH-NAME    | ZBAT-TKN-SEL-WAITON        |
| ZBAT-TKN-OBJECT           | ZBAT-TKN-START-TIME        |
| ZBAT-TKN-OUT-FILE         | ZBAT-TKN-STARTUP-MESSAGE   |
| ZBAT-TKN-PC-ERROR0        | ZBAT-TKN-STRING            |
| ZBAT-TKN-PC-ERROR1        | ZBAT-TKN-SWAP-FILE         |
| ZBAT-TKN-PC-ERROR2        | ZBAT-TKN-TERM-FILE         |
| ZBAT-TKN-PHANDLE          | ZBAT-TKN-TERMINATION-INFO  |
| ZBAT-TKN-REASON-NUMBER    | ZBAT-TKN-TEXT              |



**Table 4-12. NetBatch Simple Tokens** (page 2 of 2)

---

|                          |                         |
|--------------------------|-------------------------|
| ZBAT-TKN-RETCODE         | ZBAT-TKN-TIME-LIMIT     |
| ZBAT-TKN-SCHEDULER-ID    | ZBAT-TKN-TOTAL-CPU-TIME |
| ZBAT-TKN-SEL-ADPNAME     | ZBAT-TKN-USERID         |
| ZBAT-TKN-SEL-ASSIGN-NAME | ZBAT-TKN-VOLUME-SUBVOL  |

---

**Table 4-13. NetBatch Extensible Structured Tokens**


---

|                        |                           |
|------------------------|---------------------------|
| ZBAT-MAP-DEF-CLASS     | ZBAT-MAP-DEFINE-ERROR     |
| ZBAT-MAP-DEF-CRONTAB   | ZBAT-MAP-PAR-RELEASE-JOB  |
| ZBAT-MAP-DEF-EXECUTOR  | ZBAT-MAP-STATUS-EXECUTOR  |
| ZBAT-MAP-DEF-JOB       | ZBAT-MAP-STATUS-JOB       |
| ZBAT-MAP-DEF-SCHEDULER | ZBAT-MAP-STATUS-SCHEDULER |
| ZBAT-MAP-DEF-WAITON    |                           |

---

## NetBatch Buffer Declarations

The NetBatch subsystem provides these buffer declarations for message buffers you can use with the SPI procedures. All applications must use one of these declarations for a buffer of the recommended size.

### ZBAT-DDL-MSG-BUFFER

is a message buffer of the size recommended for commands, responses, and event messages. It has the structure:

|                                 |                              |
|---------------------------------|------------------------------|
| Definition ZBAT-DDL-MSG-BUFFER. |                              |
| 02 Z-MSGCODE                    | Type ZSPI-DDL-INT.           |
| 02 Z-BUFLen                     | Type ZSPI-DDL-UINT.          |
| 02 Z-OCCURS                     | Type ZSPI-DDL-UINT.          |
| 02 FILLER                       | Type Binary 8 Unsigned       |
|                                 | Occurs 0 to ZBAT-VAL-BUFLen  |
|                                 | times depending on Z-OCCURS. |
| End                             |                              |

### ZBAT-DDL-MSG-BUFFER-MAX

is a message buffer of the maximum size recommended for commands, responses, and event messages. It has the structure:

|                                     |                                 |
|-------------------------------------|---------------------------------|
| Definition ZBAT-DDL-MSG-BUFFER-MAX. |                                 |
| 02 Z-MSGCODE                        | Type ZSPI-DDL-INT.              |
| 02 Z-BUFLen                         | Type ZSPI-DDL-UINT.             |
| 02 Z-OCCURS                         | Type ZSPI-DDL-UINT.             |
| 02 FILLER                           | Type Binary 8 Unsigned          |
|                                     | Occurs 0 to ZBAT-VAL-BUFLen-MAX |
|                                     | times depending on Z-OCCURS.    |
| End                                 |                                 |

### ZBAT-DDL-MSG-BUFFER-MIN

is a message buffer of the minimum size recommended for commands, responses, and event messages. It has the structure:

|                                     |                                 |
|-------------------------------------|---------------------------------|
| Definition ZBAT-DDL-MSG-BUFFER-MIN. |                                 |
| 02 Z-MSGCODE                        | Type ZSPI-DDL-INT.              |
| 02 Z-BUFLen                         | Type ZSPI-DDL-UINT.             |
| 02 Z-OCCURS                         | Type ZSPI-DDL-UINT.             |
| 02 FILLER                           | Type Binary 8 Unsigned          |
|                                     | Occurs 0 to ZBAT-VAL-BUFLen-MIN |
|                                     | times depending on Z-OCCURS.    |
| End                                 |                                 |

## Private Token and Field Types

A private token is a token defined and used exclusively by the subsystem that defines it. These paragraphs describe the private token and field types defined by the NetBatch subsystem:

ZBAT-DDL-COMMAND

is an enumerated value that specifies a command. It has this structure:

|            |                             |                              |
|------------|-----------------------------|------------------------------|
| Definition | ZBAT-DDL-COMMAND            | Begin                        |
|            |                             | Type Enum                    |
|            |                             | AS "Unknown".                |
| 89         | ZBAT-ENM-COMMAND-ABORT      | Value is ZBAT-CMD-ABORT      |
|            |                             | As "ABORT COMMAND".          |
| 89         | ZBAT-ENM-COMMAND-ACTIVATE   | Value is ZBAT-CMD-ACTIVATE   |
|            |                             | As "ACTIVATE COMMAND".       |
| 89         | ZBAT-ENM-COMMAND-ADD        | Value is ZBAT-CMD-ADD        |
|            |                             | As "ADD COMMAND".            |
| 89         | ZBAT-ENM-COMMAND-ALTER      | Value is ZBAT-CMD-ALTER      |
|            |                             | As "ALTER COMMAND".          |
| 89         | ZBAT-ENM-COMMAND-DELETE     | Value is ZBAT-CMD-DELETE     |
|            |                             | As "DELETE COMMAND".         |
| 89         | ZBAT-ENM-COMMAND-GETVERSION | Value is ZSPI-CMD-GETVERSION |
|            |                             | As "GETVERSION COMMAND".     |
| 89         | ZBAT-ENM-COMMAND-INFO       | Value is ZBAT-CMD-INFO       |
|            |                             | As "INFO COMMAND".           |
| 89         | ZBAT-ENM-COMMAND-RELEASE    | Value is ZBAT-CMD-RELEASE    |
|            |                             | As "RELEASE COMMAND".        |
| 89         | ZBAT-ENM-COMMAND-RUNNEXT    | Value is ZBAT-CMD-RUNNEXT    |
|            |                             | As "RUNNEXT COMMAND".        |
| 89         | ZBAT-ENM-COMMAND-RUNNOW     | Value is ZBAT-CMD-RUNNOW     |
|            |                             | As "RUNNOW COMMAND".         |
| 89         | ZBAT-ENM-COMMAND-SHUTDOWN   | Value is ZBAT-CMD-SHUTDOWN   |
|            |                             | As "SHUTDOWN COMMAND".       |
| 89         | ZBAT-ENM-COMMAND-START      | Value is ZBAT-CMD-START      |
|            |                             | As "START COMMAND".          |
| 89         | ZBAT-ENM-COMMAND-STATUS     | Value is ZBAT-CMD-STATUS     |
|            |                             | As "STATUS COMMAND".         |
| 89         | ZBAT-ENM-COMMAND-STOP       | Value is ZBAT-CMD-STOP       |
|            |                             | As "STOP COMMAND".           |
| 89         | ZBAT-ENM-COMMAND-SUBMIT     | Value is ZBAT-CMD-SUBMIT     |
|            |                             | As "SUBMIT COMMAND".         |
| 89         | ZBAT-ENM-COMMAND-SUSPEND    | Value is ZBAT-CMD-SUSPEND    |
|            |                             | As "SUSPEND COMMAND".        |
| 89         | ZBAT-ENM-COMMAND-SWITCHCPU  | Value is ZBAT-CMD-SWITCHCPU  |
|            |                             | As "SWITCHCPU COMMAND".      |
| 89         | ZBAT-ENM-COMMAND-SWITCHLOG  | Value is ZBAT-CMD-SWITCHLOG  |
|            |                             | As "SWITCHLOG COMMAND".      |
|            | End                         |                              |

## ZBAT-DDL-COMPLETION-CODE

is an enumerated value that indicates the completion code set by a job's executor-program process when the process calls the Guardian procedure ABEND, STOP, or PROCESS\_STOP\_. ZBAT-DDL-COMPLETION-CODE has this structure:

```

Definition ZBAT-DDL-COMPLETION-CODE
Begin
  Type Enum
  AS "Unknown".
  Value is -3
  As "Process terminated; Invalid
  89 ZBAT-ENM-CC-M3      params in STOP/ABEND".
  Value is -2
  89 ZBAT-ENM-CC-M2      As "Process terminated;
  Guardian unable to pass CC".
  Value is -1
  89 ZBAT-ENM-CC-M1      As "-1 TRAP detected".
  Value is 0
  89 ZBAT-ENM-CC-0       As "Normal termination".
  Value is 1
  89 ZBAT-ENM-CC-1       As "Terminated with warning".
  Value is 2
  89 ZBAT-ENM-CC-2       As "Terminated with fatal
  errors".
  Value is 3
  89 ZBAT-ENM-CC-3       As "Premature termination with
  fatal errors".
  Value is 4
  89 ZBAT-ENM-CC-4       As "Process never started".
  Value is 5
  89 ZBAT-ENM-CC-5       As "Process callsabend".
  Value is 6
  89 ZBAT-ENM-CC-6       As "STOP/ABEND issued by an
  external process".
  Value is 7
  89 ZBAT-ENM-CC-7       As "Job requests restart".
End

```

## ZBAT-DDL-DEF-CLASS

defines ZBAT-MAP-DEF-CLASS, an extensible structured token that contains information about a class. For the structure of ZBAT-DDL-DEF-CLASS, see the description of ZBAT-MAP-DEF-CLASS in [Extensible Structured Tokens](#) on page 4-49.

## ZBAT-DDL-DEF-CRONTAB

defines ZBAT-MAP-DEF-CRONTAB, an extensible structured token that contains scheduling information about a recurring job. For the structure of ZBAT-DDL-DEF-CRONTAB, see the description of ZBAT-MAP-DEF-CRONTAB in [Extensible Structured Tokens](#) on page 4-49.

## ZBAT-DDL-DEF-EXECUTOR

defines ZBAT-MAP-DEF-EXECUTOR, an extensible structured token that contains information about an executor. For the structure of ZBAT-DDL-DEF-EXECUTOR, see the description of ZBAT-MAP-DEF-EXECUTOR in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-DEF-JOB**

defines ZBAT-MAP-DEF-JOB, an extensible structured token that contains information about a job. For the structure of ZBAT-DDL-DEF-JOB, see the description of ZBAT-MAP-DEF-JOB in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-DEF-SCHEDULER**

defines ZBAT-MAP-DEF-SCHEDULER, an extensible structured token that contains information about a scheduler. For the structure of ZBAT-DDL-DEF-SCHEDULER, see the description of ZBAT-MAP-DEF-SCHEDULER in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-DEF-WAITON**

defines ZBAT-MAP-DEF-WAITON, an extensible structured token that contains details of a job's dependent relationship with one of its masters. For the structure of ZBAT-DDL-DEF-WAITON, see the description of ZBAT-MAP-DEF-WAITON in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-DEFINE-ERROR**

defines ZBAT-MAP-DEFINE-ERROR, an extensible structured token that contains details of the error detected by the scheduler when the scheduler validated ZBAT-TKN-ATT-SET-DEFINE. For the structure of ZBAT-DDL-DEFINE-ERROR, see the description of ZBAT-MAP-DEFINE-ERROR in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-EMS**

is an enumerated value that indicates the setting of a scheduler's EMS attribute. It has this structure:

|            |                    |                             |
|------------|--------------------|-----------------------------|
| Definition | ZBAT-DDL-EMS       | Begin                       |
|            |                    | Type Enum                   |
|            |                    | AS "Unknown".               |
| 89         | ZBAT-ENM-EMS-OFF   | Value is ZBAT-VAL-EMS-OFF   |
|            |                    | As "EMS OFF ".              |
| 89         | ZBAT-ENM-EMS-ON    | Value is ZBAT-VAL-EMS-ON    |
|            |                    | As "EMS ON".                |
| 89         | ZBAT-ENM-EMS-ERROR | Value is ZBAT-VAL-EMS-ERROR |
|            |                    | As "EMS ERROR ".            |
|            | End                |                             |

## ZBAT-DDL-EXECUTOR-LIST

is an enumerated value that indicates an executor's state. It has the structure:

|            |                               |                               |
|------------|-------------------------------|-------------------------------|
| Definition | ZBAT-DDL-EXECUTOR-LIST        | Begin                         |
|            |                               | Type Enum                     |
|            |                               | AS "Unknown".                 |
| 89         | ZBAT-ENM-EXECUTOR-OFF-LIST    | Value is                      |
|            |                               | ZBAT-VAL-EXECUTOR-OFF-LIST    |
|            |                               | As "OFF-LIST".                |
| 89         | ZBAT-ENM-EXECUTOR-ON-LIST     | Value is                      |
|            |                               | ZBAT-VAL-EXECUTOR-ON-LIST     |
|            |                               | As "ON-LIST".                 |
| 89         | ZBAT-ENM-EXECUTOR-ACTIVE-LIST | Value is                      |
|            |                               | ZBAT-VAL-EXECUTOR-ACTIVE-LIST |
|            |                               | As "ACTIVE-LIST".             |
| 89         | ZBAT-ENM-EXECUTOR-STOP-LIST   | Value is                      |
|            |                               | ZBAT-VAL-EXECUTOR-STOP-LIST   |
|            |                               | As "STOP-LIST".               |
| 89         | ZBAT-ENM-EXECUTOR-DOWN-LIST   | Value is                      |
|            |                               | ZBAT-VAL-EXECUTOR-DOWN-LIST   |
|            |                               | As "DOWN-LIST".               |
| 89         | ZBAT-ENM-EXECUTOR-DELETE-LIST | Value is                      |
|            |                               | ZBAT-VAL-EXECUTOR-DELETE-LIST |
|            |                               | As "DELETE-LIST".             |
|            | End                           |                               |

## ZBAT-DDL-INT2-TRIO

contains three double-integer fields and is the token type used for the ZBAT-TKN-MIN-MAX-ERROR token returned in NetBatch error lists. For a description of ZBAT-TKN-MIN-MAX-ERROR, see [Simple Tokens](#) on page 4-40. ZBAT-DDL-INT2-TRIO has the structure:

|            |                     |                    |
|------------|---------------------|--------------------|
| Definition | ZBAT-DDL-INT2-TRIO. |                    |
| 02         | Z-INT2              | Type ZSPI-DDL-INT2 |
|            |                     | Occurs 3 times.    |
|            | End                 |                    |

## ZBAT-DDL-JOB-NUMBER

contains a job number. It has the structure:

|            |                     |                    |
|------------|---------------------|--------------------|
| Definition | ZBAT-DDL-JOB-NUMBER | Type ZSPI-DDL-INT. |
|------------|---------------------|--------------------|

## ZBAT-DDL-JOB-WHICH-LIST

is an enumerated value that indicates a job's state in the ZINFO-WHICH-LIST field of ZBAT-MAP-DEF-JOB and in the ZWHICH-LIST field of ZBAT-MAP-STATUS-JOB. It has the structure:

```

Definition ZBAT-DDL-JOB-WHICH-LIST

      89 ZBAT-ENM-JOB-RUNNOW-LIST
      89 ZBAT-ENM-JOB-RUNNEXT-LIST
      89 ZBAT-ENM-JOB-READY-LIST
      89 ZBAT-ENM-JOB-SPECIAL-LIST
      89 ZBAT-ENM-JOB-TIME-LIST
      89 ZBAT-ENM-JOB-EVENT-LIST
      89 ZBAT-ENM-JOB-TAPE-LIST
      89 ZBAT-ENM-JOB-EXECUTING-LIST
      89 ZBAT-ENM-JOB-SUSPENDED-LIST

End

Begin
Type Enum
AS "Unknown".
Value is
ZBAT-VAL-JOB-RUNNOW-LIST
As "Run now".
Value is
ZBAT-VAL-JOB-RUNNEXT-LIST
As "Run next".
Value is
ZBAT-VAL-JOB-READY-LIST
As "Ready".
Value is
ZBAT-VAL-JOB-SPECIAL-LIST
As "Special".
Value is
ZBAT-VAL-JOB-TIME-LIST
As "Time".
Value is
ZBAT-VAL-JOB-EVENT-LIST
As "Event".
Value is
ZBAT-VAL-JOB-TAPE-LIST
As "Tape".
Value is
ZBAT-VAL-JOB-EXECUTING-LIST
As "Executing".
Value is
ZBAT-VAL-JOB-SUSPENDED-LIST
As "Suspended".

```

## ZBAT-DDL-LIST

is an enumerated value that indicates a job's state in ZBAT-TKN-SEL-LIST. It has the structure:

|            |                      |                               |
|------------|----------------------|-------------------------------|
| Definition | ZBAT-DDL-LIST        | Begin                         |
|            |                      | Type Enum                     |
|            |                      | AS "Unknown".                 |
| 89         | ZBAT-ENM-RUNNOW      | Value is ZBAT-VAL-RUNNOW      |
|            |                      | As "RUNNOW ".                 |
| 89         | ZBAT-ENM-RUNNEXT     | Value is ZBAT-VAL-RUNNEXT     |
|            |                      | As "RUNNEXT ".                |
| 89         | ZBAT-ENM-READY       | Value is ZBAT-VAL-READY       |
|            |                      | As "READY ".                  |
| 89         | ZBAT-ENM-SPECIAL-1   | Value is ZBAT-VAL-SPECIAL-1   |
|            |                      | As "SPECIAL-1 ".              |
| 89         | ZBAT-ENM-SPECIAL-2   | Value is ZBAT-VAL-SPECIAL-2   |
|            |                      | As "SPECIAL-2 ".              |
| 89         | ZBAT-ENM-SPECIAL-3   | Value is ZBAT-VAL-SPECIAL-3   |
|            |                      | As "SPECIAL-3 ".              |
| 89         | ZBAT-ENM-SPECIAL-4   | Value is ZBAT-VAL-SPECIAL-4   |
|            |                      | As "SPECIAL-4 ".              |
| 89         | ZBAT-ENM-SPECIAL-5   | Value is ZBAT-VAL-SPECIAL-5   |
|            |                      | As "SPECIAL-5 ".              |
| 89         | ZBAT-ENM-SPECIAL-6   | Value is ZBAT-VAL-SPECIAL-6   |
|            |                      | As "SPECIAL-6 ".              |
| 89         | ZBAT-ENM-SPECIAL-7   | Value is ZBAT-VAL-SPECIAL-7   |
|            |                      | As "SPECIAL-7 ".              |
| 89         | ZBAT-ENM-SPECIAL-8   | Value is ZBAT-VAL-SPECIAL-8   |
|            |                      | As "SPECIAL-8 ".              |
| 89         | ZBAT-ENM-SPECIAL-9   | Value is ZBAT-VAL-SPECIAL-9   |
|            |                      | As "SPECIAL-9 ".              |
| 89         | ZBAT-ENM-SPECIAL-ANY | Value is ZBAT-VAL-SPECIAL-ANY |
|            |                      | As "SPECIAL-ANY".             |
| 89         | ZBAT-ENM-TIME        | Value is ZBAT-VAL-TIME        |
|            |                      | As "TIME ".                   |
| 89         | ZBAT-ENM-EVENT       | Value is ZBAT-VAL-EVENT       |
|            |                      | As "EVENT ".                  |
| 89         | ZBAT-ENM-TAPE        | Value is ZBAT-VAL-TAPE        |
|            |                      | As "TAPE ".                   |
| 89         | ZBAT-ENM-EXECUTING   | Value is ZBAT-VAL-EXECUTING   |
|            |                      | As "EXECUTING ".              |
| 89         | ZBAT-ENM-SUSPENDED   | Value is ZBAT-VAL-SUSPENDED   |
|            |                      | As "SUSPENDED ".              |
| End        |                      |                               |

## ZBAT-DDL-NETBATCH-NAME

contains the name of a class, executor, job, or DEFINE. It has the structure:

|            |                        |      |                  |
|------------|------------------------|------|------------------|
| Definition | ZBAT-DDL-NETBATCH-NAME | Type | ZSPI-DDL-CHAR24. |
|------------|------------------------|------|------------------|



## ZBAT-DDL-OBJECT

is an enumerated value that specifies an object type. It has the structure:

|                              |                             |
|------------------------------|-----------------------------|
| Definition ZBAT-DDL-OBJECT   | Begin                       |
|                              | Type Enum                   |
|                              | AS "Unknown".               |
| 89 ZBAT-ENM-ATT-SET          | Value is ZBAT-OBJ-ATT-SET   |
|                              | As "ATTACHMENT SET".        |
| 89 ZBAT-ENM-OBJECT-EXECUTOR  | Value is ZBAT-OBJ-EXECUTOR  |
|                              | As "EXECUTOR OBJECT".       |
| 89 ZBAT-ENM-OBJECT-JOB       | Value is ZBAT-OBJ-JOB       |
|                              | As "JOB OBJECT".            |
| 89 ZBAT-ENM-OBJECT-CLASS     | Value is ZBAT-OBJ-CLASS     |
|                              | As "CLASS OBJECT".          |
| 89 ZBAT-ENM-OBJECT-SCHEDULER | Value is ZBAT-OBJ-SCHEDULER |
|                              | As "SCHEDULER OBJECT".      |
| End                          |                             |

## ZBAT-DDL-PAR-RELEASE-JOB

defines ZBAT-MAP-PAR-RELEASE-JOB, an extensible structured token that a master job uses to release one of its dependents. For the structure of ZBAT-DDL-PAR-RELEASE-JOB, see the description of ZBAT-MAP-PAR-RELEASE-JOB in [Extensible Structured Tokens](#) on page 4-49.

## ZBAT-DDL-PC-ERROR0

is an enumerated value returned in event message ZBAT-EVT-JOB-START-ERROR. The value indicates the error returned to the scheduler by Guardian procedure PROCESS\_CREATE\_. For descriptions of ZBAT-DDL-PC-ERROR0 values, see the explanation of ZBAT-EVT-JOB-START-ERROR in [Section 6, Event Messages](#). ZBAT-DDL-PC-ERROR0 has the structure:

|             |                            |   |
|-------------|----------------------------|---|
| Definition  | ZBAT-DDL-PC-ERROR0         | Begin<br>Type Enum<br>AS "Unknown".<br>Value is 0<br>As "No error. PROCESS_CREATE_<br>successful".<br>Value is 1<br>As "Program file error. See<br>error2".<br>Value is 2<br>As "NETBATCH parameter internal<br>error".<br>Value is 3<br>As "NETBATCH parameter bound<br>internal error".<br>Value is 4<br>As "Library file error. See<br>error2".<br>Value is 5<br>As "Swap file error. See<br>error2".<br>Value is 6<br>As "EXT Swap file error. See<br>error2".<br>Value is 7<br>As "PFS create file error. See<br>error2".<br>Value is 8<br>As "Illegal TERM. See error2".<br>Value is 9<br>As "TERM file error. See<br>error2".<br>Value is 10<br>As "Unable to communicate. See<br>error 2".<br>Value is 11<br>As "Process name error. See<br>error2".<br>Value is 12<br>As "Illegal Prog. See error1".<br>Value is 13<br>As "Illegal Lib. See error1".<br>Value is 14<br>As "Undefined external<br>references.". Value is 15<br>As "No suitable PCB available".<br>Value is 16<br>As "Unable to allocate a map".<br>Value is 17<br>As "Unlicensed privileged<br>file". |
| 89          | ZBAT-ENM-PCERR-OK          |   |
| 89          | ZBAT-ENM-PCERR-FSERR       |   |
| 89          | ZBAT-ENM-PCERR-PAERR       |   |
| 89          | ZBAT-ENM-PCERR-BNERR       |   |
| 89          | ZBAT-ENM-PCERR-LBERR       |   |
| 89          | ZBAT-ENM-PCERR-SWERR       |   |
| 89          | ZBAT-ENM-PCERR-ESERR       |   |
| 89          | ZBAT-ENM-PCERR-PFSERR      |   |
| 89          | ZBAT-ENM-PCERR-ILLTERM     |   |
| 89          | ZBAT-ENM-PCERR-TERMERR     |   |
| 89          | ZBAT-ENM-PCERR-NOMONITOR   |   |
| 89          | ZBAT-ENM-PCERR-BADNAME     |   |
| 89          | ZBAT-ENM-PCERR-ILLPROG     |   |
| 89          | ZBAT-ENM-PCERR-ILLLIB      |   |
| 89          | ZBAT-ENM-PCERR-UNDEFEXT    |   |
| 89          | ZBAT-ENM-PCERR-NOPCB       |   |
| 89          | ZBAT-ENM-PCERR-NOMAP       |   |
| 89          | ZBAT-ENM-PCERR-NOTLICENSED |   |
| (continued) |                            |   |

|                               |   |
|-------------------------------|---|
| 89 ZBAT-ENM-PCERR-LIBCONF     | Value is 18<br>As "Library conflict".                                 |
| 89 ZBAT-ENM-PCERR-PROGEQLIB   | Value is 19<br>As "Program file and library<br>file are same".        |
| 89 ZBAT-ENM-PCERR-ILLSUBTYPE  | Value is 20<br>As "process device subtype<br>illegal".                |
| 89 ZBAT-ENM-PCERR-BACSUBTYPE  | Value is 21<br>As "Backup/Primary have<br>different device subtypes". |
| 89 ZBAT-ENM-PCERR-BACCREATEUN | Value is 22<br>As "Backup creation w/o name".                         |
| 89 ZBAT-ENM-PCERR-CONTEXTERR  | Value is 24<br>As "DEFINE error.See error2".                          |
| 89 ZBAT-ENM-PCERR-BADPFSSIZE  | Value is 27<br>As "the PFS size was out of range".                    |
| 89 ZBAT-ENM-PCERR-UNKNOWN-C   | Value is 28<br>As "C-series node returned error2".                    |
| End                           |   |

## ZBAT-DDL-PC-ERROR1

is an enumerated value returned in event message ZBAT-EVT-JOB-START-ERROR. The value indicates the cause of the error for ZBAT-DDL-PC-ERROR0 values ZBAT-ENM-PCERR-ILLLIB and ZBAT-ENM-PCERR-ILLPROG. For descriptions of ZBAT-DDL-PC-ERROR1 values, see the explanation of ZBAT-EVT-JOB-START-ERROR in [Section 6, Event Messages](#). ZBAT-DDL-PC-ERROR1 has the structure:

|                                   |   |
|-----------------------------------|---|
| Definition ZBAT-DDL-ERROR1        | Begin   |
|                                   | Type Enum   |
|                                   | As "Unknown".   |
| 89 ZBAT-ENM-BADFILE-NOTDISC       | Value is 1  |
| 89 ZBAT-ENM-BADFILE-NOT100        | As "Not a disc file".                                 |
|                                   | Value is 2  |
| 89 ZBAT-ENM-BADFILE-FILSYS        | As "Not file code 100".                               |
|                                   | Value is 3  |
| 89 ZBAT-ENM-BADFILE-TOSVERSION    | As "Not correct file".                                |
|                                   | Value is 4  |
|                                   | As "Requires later version of Guardian".              |
| 89 ZBAT-ENM-BADFILE-NOMAIN        | Value is 5  |
|                                   | As "No main procedure".                               |
| 89 ZBAT-ENM-BADFILE-LIBHASMAIN    | Value is 6  |
|                                   | As "LIB file has main procedure".                     |
| 89 ZBAT-ENM-BADFILE-NODATAPAGES   | Value is 7  |
|                                   | As "Prog has no data pages".                          |
| 89 ZBAT-ENM-BADFILE-PEPINVALID    | Value is 8  |
|                                   | As "Invalid PEP".                                     |
| 89 ZBAT-ENM-BADFILE-INITSEGS      | Value is 9  |
|                                   | As "Header INITSEGS not consistent with size".        |
| 89 ZBAT-ENM-BADFILE-RESIDENTSIZE  | Value is 10   |
|                                   | As "Resident size greater than code area length".     |
| 89 ZBAT-ENM-BADFILE-NOFIXUPS      | Value is 11   |
|                                   | As "File not fixed up by binders".                    |
| 89 ZBAT-ENM-BADFILE-NOUNDEFBLOCKS | Value is 12   |
|                                   | As "File has unidentified data blocks".               |
| 89 ZBAT-ENM-BADFILE-DATACODEREF   | Value is 13   |
|                                   | As "Unresolved data block references in data blocks". |
| 89 ZBAT-ENM-BADFILE-MANYSPPACES   | Value is 14   |
|                                   | As "Too many code spaces in object file".             |
| End                               |   |

## ZBAT-DDL-REASON

is an enumerated value returned in event message ZBAT-EVT-JOB-START-ERROR. The value indicates why the scheduler was unable to start a job's executor-program process. For descriptions of ZBAT-DDL-REASON values, see the explanation of ZBAT-EVT-JOB-START-ERROR in [Section 6, Event Messages](#). ZBAT-DDL-REASON has the structure:

|                               |   |
|-------------------------------|---|
| Definition ZBAT-DDL-REASON    | Begin   |
|                               | Type Enum   |
|                               | As "Unknown".   |
| 89 ZBAT-ENM-PROCESS-CREATE    | Value is<br>ZBAT-REASON-PROCESS-CREATE<br>As "PROCESS_CREATE_ error". |
| 89 ZBAT-ENM-REMOTE-NODE-DOWN  | Value is<br>ZBAT-REASON-REMOTE-NODE-DOWN<br>As "Remote node down".    |
| 89 ZBAT-ENM-ATTACHMENTS-ERROR | Value is<br>ZBAT-REASON-ATTACHMENTS-ERROR<br>As "Attachments error".  |
| 89 ZBAT-ENM-USER-NOT-FOUND    | Value is<br>ZBAT-REASON-USER-NOT-FOUND<br>As "user not found".        |
| 89 ZBAT-ENM-OPEN-FAIL         | Value is<br>ZBAT-REASON-OPEN-FAIL<br>As "open fail".                  |
| 89 ZBAT-ENM-STARTUP-MSG-FAIL  | Value is<br>ZBAT-REASON-STARTUP-MSG-FAIL<br>As "startup msg fail".    |
| 89 ZBAT-ENM-WRITE-ATT-FAIL    | Value is<br>ZBAT-REASON-WRITE-ATT-FAIL<br>As "write attachment fail". |
| 89 ZBAT-ENM-JOB-NOT-EXECUTING | Value is<br>ZBAT-REASON-JOB-NOT-EXECUTING<br>As "job not executing".  |
| 89 ZBAT-ENM-PROCESS-NOT-THERE | Value is<br>ZBAT-REASON-PROCESS-NOT-THERE<br>As "process not there".  |
| 89 ZBAT-ENM-BAD-OUT-FILE      | Value is<br>ZBAT-REASON-BAD-OUT-FILE<br>As "bad out file".            |
| End                           |   |

## ZBAT-DDL-RETCODE

is an enumerated value that indicates an error number. The standard SPI error token ZSPI-TKN-ERROR returns the error number in an error list. For descriptions of NetBatch error numbers and their associated error lists, see [Appendix A, Error Numbers and Error Lists](#). ZBAT-DDL-RETCODE has the structure:

|                                 |                             |
|---------------------------------|-----------------------------|
| Definition ZBAT-DDL-RETCODE     | Begin                       |
|                                 | Type Enum                   |
|                                 | As "Unknown".               |
| 89 ZBAT-ENM-W-SEC-BREACH        | Value is 512                |
|                                 | As "wrn sec breach".        |
| 89 ZBAT-ENM-W-IN-NE             | Value is 513                |
|                                 | As "wrn in ne".             |
| 89 ZBAT-ENM-W-EXECPROG-NE       | Value is 514                |
|                                 | As "wrn execprog ne".       |
| 89 ZBAT-ENM-W-CLASS-INITIATION  | Value is 515                |
|                                 | As "wrn class initiation".  |
| 89 ZBAT-ENM-W-EXECUTOR-STARTED  | Value is 516                |
|                                 | As "wrn executor started".  |
| 89 ZBAT-ENM-W-JOB-EXECUTING     | Value is 517                |
|                                 | As "wrn job executing".     |
| 89 ZBAT-ENM-W-WAITON-SATISFIED  | Value is 518                |
|                                 | As "wrn waiton satisfied".  |
| 89 ZBAT-ENM-W-NO-SUCH-EXECUTOR  | Value is 520                |
|                                 | As "wrn no such executor".  |
| 89 ZBAT-ENM-W-NO-SUCH-JOB       | Value is 521                |
|                                 | As "wrn no such job".       |
| 89 ZBAT-ENM-W-NOT-NETWORKABLE   | Value is 522                |
|                                 | As "wrn not networkable".   |
| 89 ZBAT-ENM-W-NO-SUCH-CLASS     | Value is 523                |
|                                 | As "wrn no such class".     |
| 89 ZBAT-ENM-W-ALTER-TAPEDRIVES  | Value is 524                |
|                                 | As "wrn alter tapedrives".  |
| 89 ZBAT-ENM-W-CPU-DOWN          | Value is 525                |
|                                 | As "wrn cpu down".          |
| 89 ZBAT-ENM-W-EXECUTOR-STOPPED  | Value is 526                |
|                                 | As "wrn executor stopped".  |
| 89 ZBAT-ENM-W-R-ACCESS          | Value is 527                |
|                                 | As "wrn r access".          |
| 89 ZBAT-ENM-W-W-ACCESS          | Value is 528                |
|                                 | As "wrn w access".          |
| 89 ZBAT-ENM-W-E-ACCESS          | Value is 529                |
|                                 | As "wrn e access".          |
| 89 ZBAT-ENM-W-P-ACCESS          | Value is 530                |
|                                 | As "wrn p access".          |
| 89 ZBAT-ENM-W-CALENDAR-ERROR    | Value is 531                |
|                                 | As "wrn calendar error".    |
| 89 ZBAT-ENM-W-CALENDAR-EXPIRED  | Value is 532                |
|                                 | As "wrn calendar expired".  |
| 89 ZBAT-ENM-W-LOG-FILE          | Value is 533                |
|                                 | As "wrn log file".          |
| 89 ZBAT-ENM-W-ATT-DELETED       | Value is 534                |
|                                 | As "wrn att deleted".       |
| 89 ZBAT-ENM-W-SECURITY          | Value is 535                |
|                                 | As "wrn security".          |
| 89 ZBAT-ENM-W-DEFAULTS-DEFINE   | Value is 536                |
|                                 | As "wrn defaults define".   |
| 89 ZBAT-ENM-W-ATT-DNE           | Value is 537                |
|                                 | As "wrn att dne".           |
| 89 ZBAT-ENM-W-ALREADY-SUSPENDED | Value is 538                |
|                                 | As "wrn already suspended". |
| 89 ZBAT-ENM-W-ALREADY-ACTIVATED | Value is 539                |
|                                 | As "wrn already activated". |
| (continued)                     |                             |

|                                  |                              |
|----------------------------------|------------------------------|
| 89 ZBAT-ENM-W-RUNNOW-TAPE        | Value is 540                 |
| 89 ZBAT-ENM-I-NO-SUCH-CMD        | As "wrn runnow tape".        |
| 89 ZBAT-ENM-W-DISALLOW-DEFINE    | Value is 541                 |
| 89 ZBAT-ENM-W-WAITON-NOT-ANY     | As "inf no such cmd".        |
| 89 ZBAT-ENM-W-WAITON-NOT-ANY     | Value is 542                 |
| 89 ZBAT-ENM-W-SAME-SYSTEM        | As "wrn disallow define".    |
| 89 ZBAT-ENM-W-WAITON-THIS-JOB    | Value is 543                 |
| 89 ZBAT-ENM-W-DEFAULT-SUBV-UNSUP | As "wrn waiton not any".     |
| 89 ZBAT-ENM-W-SWITCHCPU-DEFERRED | Value is 544                 |
| 89 ZBAT-ENM-E-BEGIN              | As "wrn same system".        |
| 89 ZBAT-ENM-E-ACTIVATE-JOB       | Value is 545                 |
| 89 ZBAT-ENM-E-AFTER              | As "wrn waiton this job".    |
| 89 ZBAT-ENM-E-AFTER-YEAR         | Value is 546                 |
| 89 ZBAT-ENM-E-AFTER-MONTH        | As "wrn default subv unsup". |
| 89 ZBAT-ENM-E-AFTER-DAY          | Value is 547                 |
| 89 ZBAT-ENM-E-AFTER-HOUR         | As "wrn switchcpu deferred". |
| 89 ZBAT-ENM-E-AFTER-MINUTE       | Value is 2047                |
| 89 ZBAT-ENM-E-ALREADY-STARTED    | As "err begin".              |
| 89 ZBAT-ENM-E-AT                 | Value is 2048                |
| 89 ZBAT-ENM-E-AT-ALLOWED         | As "err activate job".       |
| 89 ZBAT-ENM-E-AT-YEAR            | Value is 2049                |
| 89 ZBAT-ENM-E-AT-MONTH           | As "err after".              |
| 89 ZBAT-ENM-E-AT-DAY             | Value is 2050                |
| 89 ZBAT-ENM-E-AT-HOUR            | As "err after year".         |
| 89 ZBAT-ENM-E-AT-MINUTE          | Value is 2051                |
| 89 ZBAT-ENM-E-BACKUPCPU-NUMBER   | As "err after month".        |
| 89 ZBAT-ENM-E-BACKUPCPU-PRIMARY  | Value is 2052                |
| 89 ZBAT-ENM-E-CALENDAR           | As "err after day".          |
| 89 ZBAT-ENM-E-CALENDAR-FILECODE  | Value is 2053                |
| 89 ZBAT-ENM-E-COLD-START         | As "err after hour".         |
| 89 ZBAT-ENM-E-CLASS-COUNT        | Value is 2054                |
| 89 ZBAT-ENM-E-NO-CLASS-COUNT     | As "err after minute".       |
| 89 ZBAT-ENM-E-CONTEXT            | Value is 2055                |
| 89 ZBAT-ENM-E-CPU                | As "err already started".    |
| (continued)                      | Value is 2056                |
|                                  | As "err at".                 |
|                                  | Value is 2057                |
|                                  | As "err at allowed".         |
|                                  | Value is 2058                |
|                                  | As "err at year".            |
|                                  | Value is 2059                |
|                                  | As "err at month".           |
|                                  | Value is 2060                |
|                                  | As "err at day".             |
|                                  | Value is 2061                |
|                                  | As "err at hour".            |
|                                  | Value is 2062                |
|                                  | As "err at minute".          |
|                                  | Value is 2063                |
|                                  | As "err backupcpu number".   |
|                                  | Value is 2064                |
|                                  | As "err backupcpu primary".  |
|                                  | Value is 2066                |
|                                  | As "err calendar".           |
|                                  | Value is 2068                |
|                                  | As "err calendar filecode".  |
|                                  | Value is 2069                |
|                                  | As "err cold start".         |
|                                  | Value is 2071                |
|                                  | As "err class count".        |
|                                  | Value is 2072                |
|                                  | As "err no class count".     |
|                                  | Value is 2073                |
|                                  | As "err context".            |
|                                  | Value is 2074                |
|                                  | As "err cpu".                |

|                                   |  |
|-----------------------------------|--|
| 89 ZBAT-ENM-E-HOLDAFTER           | Value is 2075<br>As "err holdafter".           |
| 89 ZBAT-ENM-E-NO-CPU              | Value is 2076<br>As "err no cpu".              |
| 89 ZBAT-ENM-E-DELETE-JOB          | Value is 2077<br>As "err delete job".          |
| 89 ZBAT-ENM-E-EVERY-ZERO-MINUTES  | Value is 2078<br>As "err every zero minutes".  |
| 89 ZBAT-ENM-E-EVERY               | Value is 2079<br>As "err every".               |
| 89 ZBAT-ENM-E-EVERY-CAL-CRON      | Value is 2080<br>As "err every cal cron".      |
| 89 ZBAT-ENM-E-EXECUTOR-PROG       | Value is 2082<br>As "err executor prog".       |
| 89 ZBAT-ENM-E-EXECUTOR-ACTIVE     | Value is 2083<br>As "err executor active".     |
| 89 ZBAT-ENM-E-EXECUTOR-NOT-ACTIVE | Value is 2084<br>As "err executor not active". |
| 89 ZBAT-ENM-E-WAITON-SELF         | Value is 2085<br>As "err waiton self".         |
| 89 ZBAT-ENM-E-EXECUTOR-EXISTS     | Value is 2086<br>As "err executor exists".     |
| 89 ZBAT-ENM-E-NO-SUCH-EXECUTOR    | Value is 2087<br>As "err no such executor".    |
| 89 ZBAT-ENM-E-EXECUTOR-FULL       | Value is 2088<br>As "err executor full".       |
| 89 ZBAT-ENM-E-EXTRA-TOKEN         | Value is 2090<br>As "err extra token".         |
| 89 ZBAT-ENM-E-HOLD                | Value is 2091<br>As "err hold".                |
| 89 ZBAT-ENM-E-IFFAILS             | Value is 2092<br>As "err iffails".             |
| 89 ZBAT-ENM-E-IN                  | Value is 2093<br>As "err in".                  |
| 89 ZBAT-ENM-E-IN-REQUIRED         | Value is 2094<br>As "err in required".         |
| 89 ZBAT-ENM-E-INITIATION          | Value is 2095<br>As "err initiation".          |
| 89 ZBAT-ENM-E-USER-UNDEFINED      | Value is 2096<br>As "err user undefined".      |
| 89 ZBAT-ENM-E-JOB-FULL            | Value is 2098<br>As "err job full".            |
| 89 ZBAT-ENM-E-NO-SUCH-JOB         | Value is 2099<br>As "err no such job".         |
| 89 ZBAT-ENM-E-CLASS-EXISTS        | Value is 2102<br>As "err class exists".        |
| 89 ZBAT-ENM-E-CLASS-FULL          | Value is 2103<br>As "err class full".          |
| 89 ZBAT-ENM-E-CLASS-IN-USE        | Value is 2104<br>As "err class in use".        |
| 89 ZBAT-ENM-E-NO-SUCH-CLASS       | Value is 2105<br>As "err no such class".       |
| 89 ZBAT-ENM-E-JOBNAME             | Value is 2106<br>As "err jobname".             |
| 89 ZBAT-ENM-E-JOBNAME-EXISTS      | Value is 2107<br>As "err jobname exists".      |
| 89 ZBAT-ENM-E-JOBNAME-REQUIRED    | Value is 2108<br>As "err jobname required".    |
| 89 ZBAT-ENM-E-JOBNUMBER           | Value is 2110<br>As "err jobnumber".           |
| 89 ZBAT-ENM-E-JOB-TIMING-TYPE     | Value is 2111<br>As "err job timing type".     |
| 89 ZBAT-ENM-E-EMPTY-RESPONSE      | Value is 2117<br>As "err empty response".      |
| (continued)                       |  |



|                                  |                              |
|----------------------------------|------------------------------|
| 89 ZBAT-ENM-E-MAXPRINTLINES      | Value is 2118                |
|                                  | As "err maxprintlines".      |
| 89 ZBAT-ENM-E-MAXPRINTPAGES      | Value is 2119                |
|                                  | As "err maxprintpages".      |
| 89 ZBAT-ENM-E-NO-SUCH-EXECUTOR   | Value is 2087                |
|                                  | As "err no such executor".   |
| 89 ZBAT-ENM-E-EXECUTOR-FULL      | Value is 2088                |
|                                  | As "err executor full".      |
| 89 ZBAT-ENM-E-EXTRA-TOKEN        | Value is 2090                |
|                                  | As "err extra token".        |
| 89 ZBAT-ENM-E-HOLD               | Value is 2091                |
|                                  | As "err hold".               |
| 89 ZBAT-ENM-E-IFFAILS            | Value is 2092                |
|                                  | As "err iffails".            |
| 89 ZBAT-ENM-E-IN                 | Value is 2093                |
|                                  | As "err in".                 |
| 89 ZBAT-ENM-E-IN-REQUIRED        | Value is 2094                |
|                                  | As "err in required".        |
| 89 ZBAT-ENM-E-INITIATION         | Value is 2095                |
|                                  | As "err initiation".         |
| 89 ZBAT-ENM-E-USER-UNDEFINED     | Value is 2096                |
|                                  | As "err user undefined".     |
| 89 ZBAT-ENM-E-JOB-FULL           | Value is 2098                |
|                                  | As "err job full".           |
| 89 ZBAT-ENM-E-NO-SUCH-JOB        | Value is 2099                |
|                                  | As "err no such job".        |
| 89 ZBAT-ENM-E-CLASS-EXISTS       | Value is 2102                |
|                                  | As "err class exists".       |
| 89 ZBAT-ENM-E-CLASS-FULL         | Value is 2103                |
|                                  | As "err class full".         |
| 89 ZBAT-ENM-E-CLASS-IN-USE       | Value is 2104                |
|                                  | As "err class in use".       |
| 89 ZBAT-ENM-E-NO-SUCH-CLASS      | Value is 2105                |
|                                  | As "err no such class".      |
| 89 ZBAT-ENM-E-JOBNAME            | Value is 2106                |
|                                  | As "err jobname".            |
| 89 ZBAT-ENM-E-JOBNAME-EXISTS     | Value is 2107                |
|                                  | As "err jobname exists".     |
| 89 ZBAT-ENM-E-JOBNAME-REQUIRED   | Value is 2108                |
|                                  | As "err jobname required".   |
| 89 ZBAT-ENM-E-JOBNUMBER          | Value is 2110                |
|                                  | As "err jobnumber".          |
| 89 ZBAT-ENM-E-JOB-TIMING-TYPE    | Value is 2111                |
|                                  | As "err job timing type".    |
| 89 ZBAT-ENM-E-EMPTY-RESPONSE     | Value is 2117                |
|                                  | As "err empty response".     |
| 89 ZBAT-ENM-E-MAXPRINTLINES      | Value is 2118                |
|                                  | As "err maxprintlines".      |
| 89 ZBAT-ENM-E-MAXPRINTPAGES      | Value is 2119                |
|                                  | As "err maxprintpages".      |
| 89 ZBAT-ENM-E-MAXRESP            | Value is 2120                |
|                                  | As "err maxresp".            |
| 89 ZBAT-ENM-E-MISSING-ATTRIBUTES | Value is 2121                |
|                                  | As "err missing attributes". |
| 89 ZBAT-ENM-E-MISSING-CLASS      | Value is 2122                |
|                                  | As "err missing class".      |
| 89 ZBAT-ENM-E-MULTIPLE-CONTEXT   | Value is 2123                |
|                                  | As "err multiple context".   |
| 89 ZBAT-ENM-E-MULTIPLE-MAPS      | Value is 2124                |
|                                  | As "err multiple maps".      |
| 89 ZBAT-ENM-E-MULTIPLE-MAXRESP   | Value is 2125                |
|                                  | As "err multiple maxresp".   |
| 89 ZBAT-ENM-E-NAME-AND-NUMBER    | Value is 2126                |
|                                  | As "err name and number".    |
| 89 ZBAT-ENM-E-NAME-OR-NUMBER     | Value is 2127                |
|                                  | As "err name or number".     |
| (continued)                      |                              |

|                                |   |
|--------------------------------|---|
| 89 ZBAT-ENM-E-NO-SUBMIT        | Value is 2128<br>As "err no submit".        |
| 89 ZBAT-ENM-E-INVALID-COMMAND  | Value is 2129<br>As "err invalid command".  |
| 89 ZBAT-ENM-E-NONE-ALLOWED     | Value is 2130<br>As "err none allowed".     |
| 89 ZBAT-ENM-E-NOT-STARTED      | Value is 2131<br>As "err not started".      |
| 89 ZBAT-ENM-E-SECURITY         | Value is 2132<br>As "err security".         |
| 89 ZBAT-ENM-E-SHUTDOWN         | Value is 2133<br>As "err shutdown".         |
| 89 ZBAT-ENM-E-OUT              | Value is 2136<br>As "err out".              |
| 89 ZBAT-ENM-E-PRI              | Value is 2137<br>As "err pri".              |
| 89 ZBAT-ENM-E-RESTART          | Value is 2139<br>As "err restart".          |
| 89 ZBAT-ENM-E-STOP-ON-ABEND    | Value is 2140<br>As "err stop on abend".    |
| 89 ZBAT-ENM-E-RUNNEXT          | Value is 2141<br>As "err runnext".          |
| 89 ZBAT-ENM-E-RUNNEXT-RUNNOW   | Value is 2142<br>As "err runnext runnow".   |
| 89 ZBAT-ENM-E-SWITCHLOG-EDIT   | Value is 2143<br>As "err switchlog edit".   |
| 89 ZBAT-ENM-E-SELPRI           | Value is 2144<br>As "err selpri".           |
| 89 ZBAT-ENM-E-STARTUP-MESSAGE  | Value is 2145<br>As "err startup message".  |
| 89 ZBAT-ENM-E-STOP-JOB         | Value is 2146<br>As "err stop job".         |
| 89 ZBAT-ENM-E-SUBMIT-ALLOWED   | Value is 2147<br>As "err submit allowed".   |
| 89 ZBAT-ENM-E-SUSPEND-JOB      | Value is 2148<br>As "err suspend job".      |
| 89 ZBAT-ENM-E-TAPEDRIVES       | Value is 2149<br>As "err tapedrives".       |
| 89 ZBAT-ENM-E-UNKNOWN-COMMAND  | Value is 2150<br>As "err unknown command".  |
| 89 ZBAT-ENM-E-UNKNOWN-OBJECT   | Value is 2151<br>As "err unknown object".   |
| 89 ZBAT-ENM-E-UNKNOWN-MAP      | Value is 2152<br>As "err unknown map".      |
| 89 ZBAT-ENM-E-UNKNOWN-TOKEN    | Value is 2153<br>As "err unknown token".    |
| 89 ZBAT-ENM-E-VOLUME-REQUIRED  | Value is 2154<br>As "err volume required".  |
| 89 ZBAT-ENM-E-VOLUME           | Value is 2155<br>As "err volume".           |
| 89 ZBAT-ENM-E-WAIT             | Value is 2156<br>As "err wait".             |
| 89 ZBAT-ENM-E-WAITON-FAIL      | Value is 2157<br>As "err waiton fail".      |
| 89 ZBAT-ENM-E-WAITON-COUNT     | Value is 2158<br>As "err waiton count".     |
| 89 ZBAT-ENM-E-WAITON-JOBS-DUPL | Value is 2160<br>As "err waiton jobs dupl". |
| 89 ZBAT-ENM-E-WRONG-SCHEDULER  | Value is 2163<br>As "err wrong scheduler".  |
| 89 ZBAT-ENM-E-SCHEDULER        | Value is 2164<br>As "err scheduler".        |
| 89 ZBAT-ENM-E-BATCHCTL         | Value is 2165<br>As "err batchctl".         |
| 89 ZBAT-ENM-E-WRONG-SSID       | Value is 2166<br>As "err wrong ssid".       |
| (continued)                    |   |

|                                   |  |
|-----------------------------------|--|
| 89 ZBAT-ENM-E-SWITCHCPU           | Value is 2167<br>As "err switchcpu".           |
| 89 ZBAT-ENM-E-LOGFILE             | Value is 2168<br>As "err logfile".             |
| 89 ZBAT-ENM-E-NOT-C20-FILE        | Value is 2169<br>As "err not c20 file".        |
| 89 ZBAT-ENM-E-DST                 | Value is 2170<br>As "err dst".                 |
| 89 ZBAT-ENM-E-ATT-EXISTS          | Value is 2171<br>As "err att exists".          |
| 89 ZBAT-ENM-E-ATT-DNE             | Value is 2172<br>As "err att dne".             |
| 89 ZBAT-ENM-E-ATT-JOB             | Value is 2173<br>As "err att job".             |
| 89 ZBAT-ENM-E-ATT-REQUESTOR       | Value is 2174<br>As "err att requestor".       |
| 89 ZBAT-ENM-E-ATT                 | Value is 2175<br>As "err att".                 |
| 89 ZBAT-ENM-E-ATT-ERR             | Value is 2176<br>As "err att err".             |
| 89 ZBAT-ENM-E-ATT-OVERFLOW        | Value is 2177<br>As "err att overflow".        |
| 89 ZBAT-ENM-E-ATT-UPDATE          | Value is 2178<br>As "err att update".          |
| 89 ZBAT-ENM-E-JOB-MULTIPLE-ATT    | Value is 2179<br>As "err job multiple att".    |
| 89 ZBAT-ENM-E-INTERNAL-ERROR      | Value is 2188<br>As "err internal error".      |
| 89 ZBAT-ENM-E-FILE-ERROR          | Value is 2189<br>As "err file error".          |
| 89 ZBAT-ENM-E-NETBATCH-ID         | Value is 2190<br>As "err netbatch id".         |
| 89 ZBAT-ENM-E-NOT-IMPLEMENTED     | Value is 2191<br>As "err not implemented".     |
| 89 ZBAT-ENM-E-INVALID-SPI         | Value is 2192<br>As "err invalid spi".         |
| 89 ZBAT-ENM-E-NETBATCH-NAME       | Value is 2193<br>As "err netbatch name".       |
| 89 ZBAT-ENM-E-SUSPEND             | Value is 2194<br>As "err suspend".             |
| 89 ZBAT-ENM-E-ACTIVATE            | Value is 2195<br>As "err activate".            |
| 89 ZBAT-ENM-E-STOP                | Value is 2196<br>As "err stop".                |
| 89 ZBAT-ENM-E-STALL               | Value is 2197<br>As "err stall".               |
| 89 ZBAT-ENM-E-WILDCARD            | Value is 2198<br>As "err wildcard".            |
| 89 ZBAT-ENM-E-JOB-TOO-MANY-ATT    | Value is 2199<br>As "err job too many att".    |
| 89 ZBAT-ENM-E-DATE                | Value is 2200<br>As "err date".                |
| 89 ZBAT-ENM-E-TIME                | Value is 2201<br>As "err time".                |
| 89 ZBAT-ENM-E-AT-FLAG             | Value is 2202<br>As "err at flag".             |
| 89 ZBAT-ENM-E-MISSING-EXECUTOR    | Value is 2203<br>As "err missing executor".    |
| 89 ZBAT-ENM-E-MISSING-ATT-ID      | Value is 2204<br>As "err missing att id".      |
| 89 ZBAT-ENM-E-MISSING-RELEASE-MAP | Value is 2205<br>As "err missing release map". |
| 89 ZBAT-ENM-E-ATT-ASSIGN          | Value is 2206<br>As "err att assign".          |
| 89 ZBAT-ENM-E-ATT-DEFINE          | Value is 2207<br>As "err att define".          |
| (continued)                       |  |

|                                  |                              |
|----------------------------------|------------------------------|
| 89 ZBAT-ENM-E-ATT-PARAM          | Value is 2208                |
|                                  | As "err att param".          |
| 89 ZBAT-ENM-E-JOB-DUPL-ATT       | Value is 2209                |
|                                  | As "err job dupl att".       |
| 89 ZBAT-ENM-E-AFTER-SECOND       | Value is 2210                |
|                                  | As "err after second".       |
| 89 ZBAT-ENM-E-AFTER-MILLISEC     | Value is 2211                |
|                                  | As "err after millisec".     |
| 89 ZBAT-ENM-E-AFTER-MICROSEC     | Value is 2212                |
|                                  | As "err after microsec".     |
| 89 ZBAT-ENM-E-CLASS-NAME         | Value is 2213                |
|                                  | As "err class name".         |
| 89 ZBAT-ENM-E-WAITON-ID          | Value is 2214                |
|                                  | As "err waiton id".          |
| 89 ZBAT-ENM-E-EXECUTOR-NAME      | Value is 2215                |
|                                  | As "err executor name".      |
| 89 ZBAT-ENM-E-CLASS-INITIATION   | Value is 2216                |
|                                  | As "err class initiation".   |
| 89 ZBAT-ENM-E-VAR-BUF-FULL       | Value is 2217                |
|                                  | As "err var buf full".       |
| 89 ZBAT-ENM-E-CRONTAB            | Value is 2218                |
|                                  | As "err crontab".            |
| 89 ZBAT-ENM-E-PURGE-IN-FILE      | Value is 2219                |
|                                  | As "err purge in file".      |
| 89 ZBAT-ENM-E-HIGHPIN            | Value is 2220                |
|                                  | As "err highpin".            |
| 89 ZBAT-ENM-E-POSIX              | Value is 2221                |
|                                  | As "err posix".              |
| 89 ZBAT-ENM-E-SAVEABEND          | Value is 2222                |
|                                  | As "err saveabend".          |
| 89 ZBAT-ENM-E-RUND               | Value is 2223                |
|                                  | As "err rund".               |
| 89 ZBAT-ENM-E-JOBID-ZERO         | Value is 2224                |
|                                  | As "err jobid zero".         |
| 89 ZBAT-ENM-E-MEM                | Value is 2225                |
|                                  | As "err mem".                |
| 89 ZBAT-ENM-E-TIME-LIMIT         | Value is 2226                |
|                                  | As "err time limit".         |
| 89 ZBAT-ENM-E-DESCRIPTION        | Value is 2227                |
|                                  | As "err description".        |
| 89 ZBAT-ENM-E-TOO-MANY-SELECTORS | Value is 2228                |
|                                  | As "err too many selectors". |
| 89 ZBAT-ENM-E-NODENAME           | Value is 2229                |
|                                  | As "err nodename".           |
| 89 ZBAT-ENM-E-MAXPRI             | Value is 2230                |
|                                  | As "err maxpri".             |
| 89 ZBAT-ENM-E-MAXCONCURRENTJOBS  | Value is 2231                |
|                                  | As "err maxconcurrentjobs".  |
| 89 ZBAT-ENM-E-MAXTEMPEXECUTORS   | Value is 2232                |
|                                  | As "err maxtempexecutors".   |
| 89 ZBAT-ENM-E-EVERY-CATCHUP      | Value is 2233                |
|                                  | As "err every catchup".      |
| 89 ZBAT-ENM-E-EMS                | Value is 2234                |
|                                  | As "err ems".                |
| 89 ZBAT-ENM-E-PFS                | Value is 2235                |
|                                  | As "err pfs".                |
| 89 ZBAT-ENM-E-NO-HELP            | Value is 2236                |
|                                  | As "err no help".            |
| 89 ZBAT-ENM-E-NBFLAGS            | Value is 2237                |
|                                  | As "err nbflags".            |
| 89 ZBAT-ENM-E-END                | Value is 2238                |
|                                  | As "err end".                |
| End                              |                              |

**ZBAT-DDL-SCHEDULER-STATE**

is an enumerated value that indicates a scheduler's state. It has the structure:

|                                     |  |
|-------------------------------------|--|
| Definition ZBAT-DDL-SCHEDULER-STATE | Begin                                      |
|                                     | Type Enum                                  |
|                                     | AS "Unknown".                              |
| 89 ZBAT-ENM-SCHEDULER-ZNOTSTARTED   | Value is<br>ZBAT-VAL-SCHEDULER-ZNOTSTARTED |
|                                     | As "Scheduler not started".                |
| 89 ZBAT-ENM-SCHEDULER-ZSTARTED      | Value is<br>ZBAT-VAL-SCHEDULER-ZSTARTED    |
|                                     | As "Scheduler started".                    |
| 89 ZBAT-ENM-SCHEDULER-ZSHUTDOWN     | Value is<br>ZBAT-VAL-SCHEDULER-ZSHUTDOWN   |
|                                     | As "Scheduler shutdown".                   |
| End                                 |  |

**ZBAT-DDL-SPECIAL-REASON**

is an enumerated value that indicates the reason a job is in the SPECIAL state. It has the structure:

|  |  |
|--|--|
| Definition ZBAT-DDL-SPECIAL-REASON     | Begin                                  |
|  | Type Enum                              |
|  | AS "Unknown".                          |
| 89 ZBAT-ENM-JOB-HOLD-ON                | Value is<br>ZBAT-VAL-HOLD-ON           |
|  | As "Hold on".                          |
| 89 ZBAT-ENM-JOB-WAS-RUNNING            | Value is<br>ZBAT-VAL-WAS-RUNNING       |
|  | As "Executing".                        |
| 89 ZBAT-ENM-JOB-NEWPROCESS-ERROR       | Value is<br>ZBAT-VAL-NEWPROCESS-ERROR  |
|  | As "Process_create_ error".            |
| 89 ZBAT-ENM-JOB-FAIL-AFTER-CREATE      | Value is<br>ZBAT-VAL-FAIL-AFTER-CREATE |
|  | As "Fail after create".                |
| 89 ZBAT-ENM-JOB-RESTART-ON Value is    | ZBAT-VAL-RESTART-ON                    |
|  | As "Restart on".                       |
| 89 ZBAT-ENM-JOB-RESTART-OFF Value is   | ZBAT-VAL-RESTART-OFF                   |
|  | As "Restart off".                      |
| 89 ZBAT-ENM-JOB-CALENDAR-ERRORValue is | ZBAT-VAL-CALENDAR-ERROR                |
|  | As "Calendar error".                   |
| 89 ZBAT-ENM-JOB-CALENDAR-EMPTYValue is | ZBAT-VAL-CALENDAR-EMPTY                |
|  | As "Calendar empty".                   |
| 89 ZBAT-ENM-JOB-STALL Value is         | ZBAT-VAL-STALL                         |
|  | As "Stall".                            |
| End                                    |  |

**ZBAT-DDL-STATUS-EXECUTOR**

defines ZBAT-MAP-STATUS-EXECUTOR, an extensible structured token that contains executor status information. For the structure of ZBAT-DDL-STATUS-EXECUTOR, see the description of ZBAT-MAP-STATUS-EXECUTOR in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-STATUS-JOB**

defines ZBAT-MAP-STATUS-JOB, an extensible structured token containing job status information. For the structure of ZBAT-DDL-STATUS-JOB, see the description of ZBAT-MAP-STATUS-JOB in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-STATUS-SCHEDULER**

defines ZBAT-MAP-STATUS-SCHEDULER, an extensible structured token containing scheduler status information. For the structure of ZBAT-DDL-STATUS-SCHEDULER, see the description of ZBAT-MAP-STATUS-SCHEDULER in [Extensible Structured Tokens](#) on page 4-49.

**ZBAT-DDL-WAITON-FOR**

is an enumerated value that indicates the type of release a dependent job requires from its master. It has the structure:

|            |                           |                                    |
|------------|---------------------------|------------------------------------|
| Definition | ZBAT-DDL-WAITON-FOR       | Begin                              |
|            |                           | Type Enum                          |
|            |                           | AS "Unknown".                      |
| 89         | ZBAT-ENM-WAITON-SET       | Value is ZBAT-VAL-WAITON-SET       |
|            |                           | AS "Explicit Release".             |
| 89         | ZBAT-ENM-WAITON-STOP      | Value is ZBAT-VAL-WAITON-STOP      |
|            |                           | AS "Implicit Release on stop".     |
| 89         | ZBAT-ENM-WAITON-STOPABEND | Value is ZBAT-VAL-WAITON-STOPABEND |
|            |                           | AS "Implicit Release on abend".    |
|            |                           | End                                |

**ZBAT-DDL-WAITON-INDICATOR**

is an enumerated value that gives information about a job's dependent relationship with its master. It has the structure:

|            |                           |                                      |
|------------|---------------------------|--------------------------------------|
| Definition | ZBAT-DDL-WAITON-INDICATOR | Begin                                |
|            |                           | Type Enum                            |
|            |                           | AS "Unknown".                        |
| 89         | ZBAT-ENM-WAITON-REMOVE    | Value is ZBAT-VAL-WAITON-REMOVE      |
|            |                           | AS "Remove master".                  |
| 89         | ZBAT-ENM-WAITON-RESET     | Value is ZBAT-VAL-WAITON-RESET       |
|            |                           | AS "Set Wait".                       |
| 89         | ZBAT-ENM-WAITON-RELEASED  | Value is ZBAT-VAL-WAITON-RELEASED-OK |
|            |                           | AS "Was Released".                   |
|            |                           | End                                  |

**ZBAT-TYP-CHAR6**

is the token type of ZBAT-TKN-CHAR6 and is defined by the structure ZSPI-DDL-CHAR6 described in the *SPI Programming Manual*. For a description of ZBAT-TKN-CHAR6, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-COMMAND**

is the token type of ZBAT-TKN-COMMAND and is defined by the structure [ZBAT-DDL-COMMAND](#) on page 4-13. For a description of ZBAT-TKN-COMMAND, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-COMPLETION-CODE**

is the token type of ZBAT-TKN-COMPLETION-CODE and is defined by the structure [ZBAT-DDL-COMPLETION-CODE](#) on page 4-14. For a description of ZBAT-TKN-COMPLETION-CODE, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-INT2-TRIO**

is the token type of ZBAT-TKN-MIN-MAX-ERROR and is defined by the structure [ZBAT-DDL-INT2-TRIO](#) on page 4-16. For a description of ZBAT-TKN-MIN-MAX-ERROR, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-JOB-NUMBER**

is the token type of ZBAT-TKN-SEL-JOB-NUMBER and is defined by the structure [ZBAT-DDL-JOB-NUMBER](#) on page 4-16. For a description of ZBAT-TKN-SEL-JOB-NUMBER, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-LIST**

is the token type of ZBAT-TKN-SEL-LIST and is defined by the structure ZBAT-DDL-LIST described earlier in this subsection. For a description of ZBAT-TKN-SEL-LIST, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-NETBATCH-NAME**

is the token type of ZBAT-TKN-SEL-NETBATCH-NAME and is defined by the structure [ZBAT-DDL-NETBATCH-NAME](#) on page 4-18. For a description of ZBAT-TKN-SEL-NETBATCH-NAME, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-OBJECT**

is the token type of ZBAT-TKN-OBJECT and is defined by the structure [ZBAT-DDL-OBJECT](#) on page 4-19. For a description of ZBAT-TKN-OBJECT, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-PC-ERROR0**

is the token type of ZBAT-TKN-PC-ERROR0 and is defined by the structure [ZBAT-DDL-PC-ERROR0](#) on page 4-20. For a description of ZBAT-TKN-PC-ERROR0, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-PC-ERROR1**

is the token type of ZBAT-TKN-PC-ERROR1 and is defined by the structure [ZBAT-DDL-PC-ERROR1](#) on page 4-22. For a description of ZBAT-TKN-PC-ERROR1, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-REASON**

is the token type of ZBAT-TKN-REASON-NUMBER and is defined by the structure [ZBAT-DDL-REASON](#) on page 4-23. For a description of ZBAT-TKN-REASON-NUMBER, see [Simple Tokens](#) on page 4-40.

**ZBAT-TYP-RETCODE**

is the token type of ZBAT-TKN-RETCODE and is defined by the structure [ZBAT-DDL-RETCODE](#) on page 4-24. For a description of ZBAT-TKN-RETCODE, see [Simple Tokens](#) on page 4-40.



## Predefined Token and Field Values

A predefined token or field is a token or field that has a symbolic name in the NetBatch definitions file. The NetBatch subsystem defines these token and field values:

ZBAT-VAL-BUFLLEN

is the recommended buffer size in bytes for commands sent to the NetBatch subsystem.

ZBAT-VAL-BUFLLEN-MAX

is the maximum buffer size in bytes for commands sent to the NetBatch subsystem.

ZBAT-VAL-BUFLLEN-MIN

is the minimum buffer size in bytes for commands sent to the NetBatch subsystem.

ZBAT-VAL-CALENDAR-EMPTY

indicates a job is in the SPECIAL-8 state (the job's calendar file has no more dates).

ZBAT-VAL-CALENDAR-ERROR

indicates a job is in the SPECIAL-7 state (an error occurred when the scheduler tried to open the job's calendar file).

ZBAT-VAL-EMS-ERROR

indicates the scheduler's EMS attribute is set to ERROR.

ZBAT-VAL-EMS-OFF

indicates the scheduler's EMS attribute is set to OFF.

ZBAT-VAL-EMS-ON

indicates the scheduler's EMS attribute is set to ON.

ZBAT-VAL-EVENT

indicates the EVENT job state.

ZBAT-VAL-EXECUTING

indicates the EXECUTING and OVER LIMIT job states.

ZBAT-VAL-EXECUTOR-ACTIVE-LIST

indicates an executor is in the ACTIVE state.

ZBAT-VAL-EXECUTOR-DELETE-LIST

indicates an executor is in the DELETE state.

ZBAT-VAL-EXECUTOR-DOWN-LIST

indicates an executor is in the DOWN state.

ZBAT-VAL-EXECUTOR-OFF-LIST

indicates an executor is in the OFF state.

ZBAT-VAL-EXECUTOR-ON-LIST

indicates an executor is in the ON state.

ZBAT-VAL-EXECUTOR-STOP-LIST

indicates an executor is in the STOP state.

ZBAT-VAL-EXTERNAL-SSID

is the NetBatch subsystem ID defined as a string; for example, TANDEM.9.17438.

ZBAT-VAL-FAIL-AFTER-CREATE

indicates a job is in the SPECIAL-4 state (the scheduler successfully created a new process for the job's executor program, but the program failed during startup).

ZBAT-VAL-FIRST-LIST

indicates the beginning of a list of token values in the NetBatch DDL definitions file. This value is for HP internal use only.

ZBAT-VAL-HOLD-ON

indicates a job is in the SPECIAL-1 state (the job is on hold).

ZBAT-VAL-JOB-EVENT-LIST

indicates a job is in the EVENT state.

ZBAT-VAL-JOB-EXECUTING-LIST

indicates a job is in the EXECUTING or OVER LIMIT state.

ZBAT-VAL-JOB-READY-LIST

indicates a job is in the READY state.

ZBAT-VAL-JOB-RUNNEXT-LIST

indicates a job is in the RUNNEXT state.

ZBAT-VAL-JOB-RUNNOW-LIST

indicates a job is in the RUNNOW state.

ZBAT-VAL-JOB-SPECIAL-LIST

indicates a job is in the SPECIAL state specified by the ZINFO-SPECIAL-REASON field of ZBAT-MAP-DEF-JOB.

ZBAT-VAL-JOB-SUSPENDED-LIST

indicates a job is in the SUSPENDED state.

ZBAT-VAL-JOB-TAPE-LIST

indicates a job is in the TAPE state.

ZBAT-VAL-JOB-TIME-LIST

indicates a job is in the TIME state.

ZBAT-VAL-LAST-LIST

indicates the end of a list of token values in the NetBatch DDL definitions file. This value is for HP internal use only.

ZBAT-VAL-NEWPROCESS-ERROR

indicates a job is in the SPECIAL-3 state (the scheduler tried to create a new process for the job's executor program, but failed).

ZBAT-VAL-READY

indicates the READY job state.

ZBAT-VAL-RESTART-OFF

indicates a job is in the SPECIAL-6 state (the job has the RESTART OFF attribute and abended but did not restart because its attributes include IFFAILS OFF).

ZBAT-VAL-RESTART-ON

indicates a job is in the SPECIAL-5 state (the job has the RESTART ON attribute and abended but did not restart because its attributes include IFFAILS OFF).

ZBAT-VAL-RUNNEXT

indicates the RUNNEXT job state.

ZBAT-VAL-RUNNOW

indicates the RUNNOW job state.

ZBAT-VAL-SCHEDULER-ZNOTSTARTED

indicates the scheduler's primary process is running, but the backup process is not running. To create the backup process, use the START SCHEDULER command.

ZBAT-VAL-SCHEDULER-ZSHUTDOWN

indicates the scheduler is shutting down.

ZBAT-VAL-SCHEDULER-ZSTARTED

indicates the scheduler's primary and backup processes are running.

ZBAT-VAL-SPECIAL-1

indicates the SPECIAL-1 job state.

ZBAT-VAL-SPECIAL-2

indicates the SPECIAL-2 job state.

ZBAT-VAL-SPECIAL-3

indicates the SPECIAL-3 job state.

ZBAT-VAL-SPECIAL-4

indicates the SPECIAL-4 job state.

ZBAT-VAL-SPECIAL-5

indicates the SPECIAL-5 job state.

ZBAT-VAL-SPECIAL-6

indicates the SPECIAL-6 job state.

ZBAT-VAL-SPECIAL-7

indicates the SPECIAL-7 job state.

ZBAT-VAL-SPECIAL-8

indicates the SPECIAL-8 job state.

ZBAT-VAL-SPECIAL-9

indicates the SPECIAL-9 job state.

ZBAT-VAL-SPECIAL-ANY

indicates any SPECIAL job state.

**ZBAT-VAL-SSID**

is the subsystem ID of the NetBatch subsystem. For information on the structure of ZBAT-VAL-SSID, see [ZSPI-TKN-SSID](#) on page 4-5.

**ZBAT-VAL-STALL**

indicates a job is in the SPECIAL-9 state (the job has the attribute STALL ON and failed while running).

**ZBAT-VAL-SUSPENDED**

indicates the SUSPENDED job state.

**ZBAT-VAL-TAPE**

indicates the TAPE job state.

**ZBAT-VAL-TIME**

indicates the TIME job state.

**ZBAT-VAL-VERSION**

is the product version of the NetBatch subsystem; for example, D30.

**ZBAT-VAL-WAITON-RELEASED-OK**

indicates a job was released by its master.

**ZBAT-VAL-WAITON-REMOVE**

removes a master job from the list of masters specified by the dependent job's WAITON attribute.

**ZBAT-VAL-WAITON-RESET**

indicates a job has yet to be released by its master.

**ZBAT-VAL-WAITON-SET**

indicates a job is waiting for an explicit release from its master.

**ZBAT-VAL-WAITON-STOP**

indicates a job is waiting for an implicit release from its master on normal termination of that master.

**ZBAT-VAL-WAITON-STOPABEND**

indicates a job is waiting for an implicit release from its master on normal or abnormal termination of that master.

**ZBAT-VAL-WAS-RUNNING**

indicates a job is in the SPECIAL-2 state (the job was running when an event other than execution of an ABORT SCHEDULER or SHUTDOWN SCHEDULER command stopped its scheduler's processes).

## Simple Tokens

A simple token has a value consisting of a single field or a fixed structure. The NetBatch subsystem defines these simple tokens:

**ZBAT-TKN-ATT-SET-ASSIGN**

contains a string that specifies the value of an ASSIGN attachment-set attribute. The string's form is a TACL ASSIGN command with the ASSIGN keyword omitted. (For example, when assigning logical file name C to actual file \$A.B.C, the token would contain C, \$A.B.C rather than ASSIGN C, \$A.B.C.) For information on the ASSIGN command, see the *TACL Reference Manual*

**ZBAT-TKN-ATT-SET-DEFINE**

contains a string that specifies the value of a DEFINE attachment-set attribute. The string's form is a TACL ADD DEFINE command with the ADD DEFINE keywords omitted. (For example, when adding a spool DEFINE named =OUT that specifies a location of \$\$, the token would contain =OUT, CLASS SPOOL, LOC \$\$ rather than ADD DEFINE =OUT, CLASS SPOOL, LOC \$\$.) For information on the ADD DEFINE command, see the *TACL Reference Manual*.

**ZBAT-TKN-ATT-SET-ID**

contains a string that specifies an attachment-set ID or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of attachment-set IDs. For information on the form of an attachment-set ID, see the descriptions of the attachment-set commands in [Section 5, Commands and Responses](#).

**ZBAT-TKN-ATT-SET-PARAM**

contains a string that specifies the value of a PARAM attachment-set attribute. The string's form is:

```
STRUCT ci^param;           ! PARAM message
BEGIN                     !
  INT numparams;           ! [1] number of
                           !   parameters
                           !   included in
                           !   this message
  STRING parameters [0:1023]; ! [2] beginning of
                           !   parameters
END;
```

The field *parameters* in the preceding message format comprises *numparams* records of the form (offsets are given in bytes):

```
param[0]          = length n, in bytes, of parameter-name
param[1] FOR n    = parameter-name
param[n+1]        = length v, in bytes, of parameter-value
param[n+2] FOR v  = parameter-value
```

#### ZBAT-TKN-ATT-SET-SECURITY

contains a 16-bit signed integer that specifies the value of an attachment set's SECURITY attribute. The security bits are:

|         |                             |
|---------|-----------------------------|
| <0:3>   | 0                           |
| <4:6>   | ID code allowed for read    |
| <7:9>   | ID code allowed for write   |
| <10:12> | ID code allowed for execute |
| <13:15> | ID code allowed for purge   |

ID code can be one of:

|   |   |
|---|---|
| 0 | Any user (local)                              |
| 1 | Member of owner's group (local)               |
| 2 | Owner (local)                                 |
| 4 | Any user (local or remote)                    |
| 5 | Member of owner's community (local or remote) |
| 6 | Owner (local or remote)                       |
| 7 | Super ID only (local)                         |

The default value is %6666 ("UUUU").

#### ZBAT-TKN-ATT-SET-TEMPORARY

contains a Boolean value that specifies an attachment set's TEMPORARY attribute. The values are:

|                |                         |
|----------------|-------------------------|
| ZSPI-VAL-FALSE | Specifies TEMPORARY OFF |
| ZSPI-VAL-TRUE  | Specifies TEMPORARY ON  |

The default value is ZSPI-VAL-FALSE for attachment sets with user-specified identifiers (that is, named attachment sets). For attachment sets added with scheduler-generated identifiers (that is, numbered attachment sets), the default value is ZSPI-VAL-TRUE.

#### ZBAT-TKN-BATCHCTL

contains the name of the scheduler's configuration file.

#### ZBAT-TKN-BYTESTRING

contains a variable-length string of bytes.

**ZBAT-TKN-CALENDAR**

contains the name of a BATCHCAL file that specifies a job's run times. ZBAT-TKN-CALENDAR, ZBAT-MAP-DEF-CRONTAB, and the ZEVERTY-DAYS, ZEVERTY-HOURS, and ZEVERTY-MINUTES fields of ZBAT-MAP-DEF-JOB are mutually exclusive.

**ZBAT-TKN-CHAR6**

contains a string of six ASCII characters, also addressable as three integers or six individual characters.

**ZBAT-TKN-COMMAND**

contains an enumerated value that specifies a NetBatch command. For a list of the possible values, see the description of [ZBAT-DDL-COMMAND](#) on page 4-13.

**ZBAT-TKN-COMPLETION-CODE**

contains an enumerated value that indicates the completion code set by a job's executor-program process when the process calls the Guardian procedure ABEND, STOP, or PROCESS\_STOP\_. For a list of the possible values, see the description of [ZBAT-DDL-COMPLETION-CODE](#) on page 4-14.

**ZBAT-TKN-DATA-BASE**

contains the location of a scheduler database in *\$volume.subvolume* form.

**ZBAT-TKN-DESCRIPTION**

contains a job description (maximum size is 1000 bytes).

**ZBAT-TKN-EXECUTOR-ID**

contains an executor name.

**ZBAT-TKN-EXECUTOR-PROGRAM**

contains the name of a program file.

**ZBAT-TKN-EXTSWAP-FILE**

contains the name of the swap file for the default extended data segment of a job's executor-program process.

**ZBAT-TKN-FORMATSUBJECT**

contains a 16-bit signed integer that forms part of the DSM template-lookup key. For more information, see the discussion of the MSG edit code in the *DSM Template Services Manual*.

**ZBAT-TKN-IN-FILE**

contains the name of a job input file.



ZBAT-TKN-INT

contains a 16-bit signed integer.

ZBAT-TKN-INT2

contains a 32-bit signed integer.

ZBAT-TKN-JOB-ID

contains a string that specifies the owner and name of a job using an attachment set. The form of the string is *(group-name.user-name)job-name*, where *group-name.user-name* specifies the owner's Guardian user ID and *job-name* the name of the job.

ZBAT-TKN-JOB-NAME-ID

contains a job name.

ZBAT-TKN-JOB-NUMBER

contains a job number.

ZBAT-TKN-LIB-FILE

contains the name of a library file.

ZBAT-TKN-LOG-FILE

contains a scheduler log-file name. For more details, see information on scheduler log files in the *NetBatch Manual*.

ZBAT-TKN-MIN-MAX-ERROR

contains three values in an error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

ZBAT-TKN-NETBATCH-NAME

contains the name of a class, executor, job, or DEFINE.

ZBAT-TKN-OBJECT

contains an enumerated value that specifies a NetBatch object type. For a list of the possible values, see the description of [ZBAT-DDL-OBJECT](#) on page 4-19.

ZBAT-TKN-OUT-FILE

contains the name of an output file.

**ZBAT-TKN-PC-ERROR0**

contains an enumerated value returned in event message ZBAT-EVT-JOB-START-ERROR. The value indicates the error returned to the scheduler by Guardian procedure `PROCESS_CREATE_`. For a list of the possible values, see the description of [ZBAT-DDL-PC-ERROR0](#) on page 4-20. For descriptions of the values, see the explanation of ZBAT-EVT-JOB-START-ERROR in [Section 6, Event Messages](#).

**ZBAT-TKN-PC-ERROR1**

contains an enumerated value returned in event message ZBAT-EVT-JOB-START-ERROR. The value indicates the cause of the error for ZBAT-DDL-PC-ERROR0 values ZBAT-ENM-PCERR-ILLIB and ZBAT-ENM-PCERR-ILLPROG. For a list of the possible values, see the description of [ZBAT-DDL-PC-ERROR1](#) on page 4-22. For descriptions of the values, see the explanation of ZBAT-EVT-JOB-START-ERROR in [Section 6, Event Messages](#).

**ZBAT-TKN-PC-ERROR2**

contains a Guardian procedure error number indicating the cause of the error for these ZBAT-TKN-PC-ERROR0 values:

|                           |                          |
|---------------------------|--------------------------|
| ZBAT-ENM-PCERR-BADNAME    | ZBAT-ENM-PCERR-LBERR     |
| ZBAT-ENM-PCERR-CONTEXTERR | ZBAT-ENM-PCERR-NOMONITOR |
| ZBAT-ENM-PCERR-ESERR      | ZBAT-ENM-PCERR-PFSERR    |
| ZBAT-ENM-PCERR-FSERR      | ZBAT-ENM-PCERR-SWERR     |
| ZBAT-ENM-PCERR-ILLTERM    | ZBAT-ENM-PCERR-TERMERR   |
| ZBAT-ENM-PCERR-LBERR      | ZBAT-ENM-PCERR-UNKNOWN-C |

For error details, see the *Guardian Procedure Errors and Messages Manual*.

**ZBAT-TKN-PHANDLE**

contains the process handle of a job's executor-program process.

**ZBAT-TKN-REASON-NUMBER**

contains an enumerated value returned in event message ZBAT-EVT-JOB-START-ERROR. The value indicates why the scheduler could not start a job's executor program. For a list of possible values, see the [ZBAT-DDL-REASON](#) on page 4-23. For descriptions of values, see the ZBAT-EVT-JOB-START-ERROR description in [Section 6, Event Messages](#).

**ZBAT-TKN-RETCODE**

contains an enumerated value that specifies a NetBatch error number. For a list of the possible values, see the description of [ZBAT-DDL-RETCODE](#) on page 4-24.

**ZBAT-TKN-SCHEDULER-ID**

contains a scheduler name in the form `\node.$process-name`.

**ZBAT-TKN-SEL-ADPNAME**

contains a string that specifies an attachment-set name.

**ZBAT-TKN-SEL-ASSIGN-NAME**

contains a string that specifies an ASSIGN name or, when the string includes either or both of the asterisk (\*) and question mark (?) wild-card characters, a range of ASSIGN names. For information on the form of an ASSIGN name, see the description of the ASSIGN command in the *TACL Reference Manual*

**ZBAT-TKN-SEL-CLASSNAME**

contains a variable-length string that specifies a class name or, when the string includes either or both of the asterisk (\*) and question mark (?) wild-card characters, a range of class names. For information on the form of a class name, see the descriptions of the ADD, ALTER, DELETE, and INFO class commands in [Section 5, Commands and Responses](#).

ZBAT-TKN-SEL-CLASSNAME replaces the fixed-length NetBatch D21 token ZBAT-TKN-SEL-CLASS-NAME.

**ZBAT-TKN-SEL-DEFINE-NAME**

contains a string that specifies a DEFINE name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of DEFINE names. For information on the form of a DEFINE name, see the description of the ADD DEFINE command in the *TACL Reference Manual*.

**ZBAT-TKN-SEL-EXECUTORNAME**

contains a variable-length string specifying an executor name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of executor names. For information on the form of an executor name, see the descriptions of the ADD, ALTER, DELETE, INFO, START, STATUS, and STOP executor commands in [Section 5, Commands and Responses](#).

ZBAT-TKN-SEL-EXECUTORNAME replaces the fixed-length NetBatch D21 token ZBAT-TKN-SEL-EXECUTOR-NAME.

**ZBAT-TKN-SEL-INNAME**

contains a string that specifies an input-file name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of input-file names.

**ZBAT-TKN-SEL-JOB-NUMBER**

contains an integer that specifies a job number.

**ZBAT-TKN-SEL-JOBNAME**

contains a variable-length string that specifies a job name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names. For information on the form of a job name, see the descriptions of the ACTIVATE, ALTER, DELETE, INFO, RUNNEXT, RUNNOW, STATUS, STOP, SUBMIT, and SUSPEND job commands in [Section 5, Commands and Responses](#).

ZBAT-TKN-SEL-JOBNAME replaces the fixed-length NetBatch D21 token ZBAT-TKN-SEL-JOB-NAME.

**ZBAT-TKN-SEL-LIST**

contains an enumerated value that specifies a job state. For a list of the possible values, see the description of [ZBAT-DDL-LIST](#) on page 4-18.

**ZBAT-TKN-SEL-NETBATCH-NAME**

contains the name of a class, executor, job, or DEFINE. Its token type is ZBAT-TYP-NETBATCH-NAME, which is defined by the structure [ZBAT-DDL-NETBATCH-NAME](#) on page 4-18.

**ZBAT-TKN-SEL-NOTADPNAME**

contains a string that specifies an attachment-set name.

**ZBAT-TKN-SEL-NOTCLASSNAME**

contains a string that specifies a class name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of class names.

**ZBAT-TKN-SEL-NOTINNAME**

contains a string that specifies an input-file name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of input-file names.

**ZBAT-TKN-SEL-NOTJOBNAME**

contains a string that specifies a job name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

**ZBAT-TKN-SEL-NOTLIST**

contains an enumerated value that specifies a job state. For a list of the possible values, see the description of [ZBAT-DDL-LIST](#) on page 4-18.

**ZBAT-TKN-SEL-NOTUSERNAME**

contains a string that specifies a user ID or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of user IDs. The user ID can be in *group-name.user-name* or *group-ID,user-ID* form.

**ZBAT-TKN-SEL-NOTWAITON**

contains a string that specifies a master-job name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of master-job names.

**ZBAT-TKN-SEL-PARAM-NAME**

contains a string that specifies a PARAM name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of PARAM names. For information on the form of a PARAM name, see the description of the PARAM command in the *TACL Reference Manual*.

**ZBAT-TKN-SEL-USERNAME**

contains a string that specifies a user ID or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of user IDs. The user ID can be in *group-name.user-name* or *group-ID,user-ID* form.

**ZBAT-TKN-SEL-WAITON**

contains a string that specifies a master-job name or, when the string includes either or both the asterisk (\*) and question mark (?) wild-card characters, a range of master-job names.

**ZBAT-TKN-START-TIME**

contains a 64-bit Julian timestamp indicating the time when the scheduler started the initial process (the executor-program process) of the job.

**ZBAT-TKN-STARTUP-MESSAGE**

contains one or more program parameters sent by the scheduler to a job's executor-program process in the startup message.

**ZBAT-TKN-STRING**

contains a variable-length string of ASCII characters.

**ZBAT-TKN-SWAP-FILE**

contains the name of a file to be used as the swap file for the user data stack segment of a job's executor-program process.

**ZBAT-TKN-TERM-FILE**

contains the name of the home terminal of a job's executor-program process.

**ZBAT-TKN-TERMINATION-INFO**

contains an SPI error number set by a job's executor-program process when the process calls the Guardian procedure ABEND, STOP, or PROCESS\_STOP\_. The number indicates why the process stopped itself. For information about SPI error numbers, see the *SPI Programming Manual*.

**ZBAT-TKN-TEXT**

contains a text string from the process-deletion system message sent to a job's executor-program process by Guardian procedure ABEND, STOP, or PROCESS\_STOP\_. This token is present only if the message includes a text string.

**ZBAT-TKN-TIME-LIMIT**

contains a 32-bit signed integer that specifies a job's time limit.

**ZBAT-TKN-TOTAL-CPU-TIME**

contains a 64-bit fixed-point number indicating, in microseconds, the sum of the processor time taken by all processes of a job.

**ZBAT-TKN-USERID**

contains two bytes of information that identifies a job owner. The first byte contains the owner's group ID. The second byte contains the user ID.

**ZBAT-TKN-VOLUME-SUBVOL**

contains a default node, volume, and subvolume used for qualifying unqualified file references in a job's input file or in ZBAT-TKN-ATT-SET-DEFINE. For more information, see the descriptions of the ADD ATTACHMENT-SET, ALTER ATTACHMENT-SET, ALTER JOB, INFO JOB, and SUBMIT JOB commands in [Section 5, Commands and Responses](#).

## Extensible Structured Tokens

An extensible structured token has a token map and a value consisting of data fields whose number can be extended. This extensibility lets HP add new fields to the structure in later releases. Extensible structured tokens are defined by token maps. The NetBatch subsystem defines these extensible structured tokens:

### ZBAT-MAP-DEF-CLASS

is an extensible structured token that contains information about a class. The token's usage in class commands is:

| Command      | ZBAT-MAP-DEF-CLASS          |
|--------------|-----------------------------|
| ADD CLASS    | Required in command buffer  |
| ALTER CLASS  | Required in command buffer  |
| DELETE CLASS | Not used                    |
| INFO CLASS   | Returned in response buffer |

ZBAT-MAP-DEF-CLASS has the structure:

|  |                                     |
|--|-------------------------------------|
| <pre> Definition ZBAT-DDL-DEF-CLASS. 02 ZINITIATION End </pre> | <pre> Type ZSPI-DDL-BOOLEAN. </pre> |
|--|-------------------------------------|

### ZINITIATION

is a Boolean field that specifies the class's INITIATION attribute. Values are:

|                |                          |
|----------------|--------------------------|
| ZSPI-VAL-FALSE | Specifies INITIATION OFF |
| ZSPI-VAL-TRUE  | Specifies INITIATION ON  |

The default value is ZSPI-VAL-TRUE.

### ZBAT-MAP-DEF-CRONTAB

is an extensible structured token that contains scheduling information about a recurring job. ZBAT-MAP-DEF-CRONTAB, ZBAT-TKN-CALENDAR, and the ZEVEY-DAYS, ZEVEY-HOURS, and ZEVEY-MINUTES fields of ZBAT-MAP-DEF-JOB are mutually exclusive. The token's usage in job commands is:

| Command      | ZBAT-MAP-DEF-CRONTAB        |
|--------------|-----------------------------|
| ACTIVATE JOB | Not used                    |
| ALTER JOB    | Optional in command buffer  |
| DELETE JOB   | Not used                    |
| INFO JOB     | Returned in response buffer |
| RELEASE JOB  | Not used                    |
| RUNNEXT JOB  | Not used                    |
| RUNNOW JOB   | Not used                    |
| STATUS JOB   | Not used                    |
| STOP JOB     | Not used                    |
| SUBMIT JOB   | Optional in command buffer  |
| SUSPEND JOB  | Not used                    |

ZBAT-MAP-DEF-CRONTAB has the structure:

|                                  |                     |
|----------------------------------|---------------------|
| Definition ZBAT-DDL-DEF-CRONTAB. |                     |
| 02 ZMINUTES                      | Type ZSPI-DDL-INT4. |
| 02 ZHOURS                        | Type ZSPI-DDL-INT2. |
| 02 ZDAYS                         | Type ZSPI-DDL-INT2. |
| 02 ZMONTHS                       | Type ZSPI-DDL-INT.  |
| 02 ZWEEKDAYS                     | Type ZSPI-DDL-INT.  |
| End                              |                     |

#### ZMINUTES

is a 64-bit fixed-point number with a bit set to 1 for each minute in an hour that the job is to run. Bits 0 through 59 represent minutes 0 through 59. Bits 60 through 63 are fillers.

#### ZHOURS

is a 32-bit signed integer with a bit set to 1 for each hour in a day that the job is to run. Bits 0 through 23 represent hours 0 through 23. Bits 24 through 31 are fillers.

#### ZDAYS

is a 32-bit signed integer with a bit set to 1 for each day in a month that the job is to run.

#### ZMONTHS

is a 16-bit signed integer with a bit set to 1 for each month in a year that the job is to run. Bit 0 and 13 through 15 are fillers.

#### ZWEEKDAYS

is a 16-bit signed integer with a bit set to 1 for each weekday in a month that the job is to run. Bit 0 represents Sunday. Bit 6 represents Saturday. If all bits are 1, ZDAYS determines the day. If ZDAYS and ZWEEKDAYS bits are not all set to 1, the days selected can match ZDAYS or ZWEEKDAYS.

#### ZBAT-MAP-DEF-EXECUTOR

is an extensible structured token that contains information about an executor. The token's usage in executor commands is:

| Command         | ZBAT-MAP-DEF-EXECUTOR       |
|-----------------|-----------------------------|
| ADD EXECUTOR    | Required in command buffer  |
| ALTER EXECUTOR  | Required in command buffer  |
| DELETE EXECUTOR | Not used                    |
| INFO EXECUTOR   | Returned in response buffer |
| START EXECUTOR  | Not used                    |
| STATUS EXECUTOR | Returned in response buffer |
| STOP EXECUTOR   | Not used                    |



## ZBAT-MAP-DEF-EXECUTOR has the structure:

|                                   |                              |
|-----------------------------------|------------------------------|
| Definition ZBAT-DDL-DEF-EXECUTOR. |                              |
| 02 ZCPU                           | Type ZSPI-DDL-INT.           |
| 02 ZJOBNUMBER                     | Type ZSPI-DDL-INT.           |
| 02 ZCLASS-COUNT                   | Type ZSPI-DDL-INT.           |
| 02 ZCLASSES                       | Occurs 8 times.              |
| 03 ZCLASSNAME                     | Type ZBAT-DDL-NETBATCH-NAME. |
| 03 FILLER                         | Type ZSPI-DDL-INT.           |
| 02 ZCLASS                         | Type ZBAT-DDL-NETBATCH-NAME. |
| End                               |                              |

### ZCPU

specifies the executor's processor attribute. ZCPU can specify any processor configured for the scheduler's node.

### ZJOBNUMBER

is the number of the job using the executor.

### ZCLASS-COUNT

is an integer that indicates the number of classes assigned to the executor. The range for the value of the integer is 1 through 8. The value depends on the number of occurrences of ZCLASSES.

### ZCLASSES

specifies the executor's CLASS attribute. The field is:

#### ZCLASSNAME

is the name of a class and can occur from one to eight times. The default value is the class specified by the scheduler's DEFAULT-CLASS attribute.

### ZCLASS

is the name of the class of the job using the executor.

### ZBAT-MAP-DEF-JOB

is an extensible structured token that contains information about a job. The token's usage in job commands is:

| Command      | ZBAT-MAP-DEF-JOB            |
|--------------|-----------------------------|
| ACTIVATE JOB | Not used                    |
| ALTER JOB    | Required in command buffer  |
| DELETE JOB   | Not used                    |
| INFO JOB     | Returned in response buffer |
| RELEASE JOB  | Not used                    |
| RUNNEXT JOB  | Not used                    |
| RUNNOW JOB   | Not used                    |
| STATUS JOB   | Not used                    |

**Command                    ZBAT-MAP-DEF-JOB (continued)**

|             |                            |
|-------------|----------------------------|
| STOP JOB    | Not used                   |
| SUBMIT JOB  | Required in command buffer |
| SUSPEND JOB | Not used                   |

ZBAT-MAP-DEF-JOB has the structure:

```

Definition ZBAT-DDL-DEF-JOB.
  02 ZCLASSNAME                      Type ZBAT-DDL-NETBATCH-NAME.
  02 ZHOLD                           Type ZSPI-DDL-BOOLEAN.
  02 ZHOLD-AFTER                     Type ZSPI-DDL-BOOLEAN.
  02 ZRESTART                        Type ZSPI-DDL-BOOLEAN.
  02 ZSTOP-ON-ABEND                  Type ZSPI-DDL-BOOLEAN.
  02 ZAT-FLAG                        Type ZSPI-DDL-BOOLEAN.
  02 ZIFFFAILS                       Type ZSPI-DDL-BOOLEAN.
  02 ZPURGE-IN-FILE                  Type ZSPI-DDL-BOOLEAN.
  02 ZSTALL                          Type ZSPI-DDL-BOOLEAN.
  02 ZINFO-NEXT-RUNTIME              Type ZSPI-DDL-INT4.
  02 ZINFO-OUT-SPOOL-NUM             Type ZSPI-DDL-INT.
  02 ZINFO-WHICH-LIST                Type ZBAT-DDL-JOB-WHICH-LIST.
  02 ZINFO-SPECIAL-REASON            Type ZBAT-DDL-SPECIAL-REASON.
  02 ZINFO-TOTAL-CPU-TIME            Type ZSPI-DDL-INT4.
  02 ZINFO-OPEN-ACCESSOR             Type ZSPI-DDL-INT.
  02 ZRECID                          Type ZSPI-DDL-BOOLEAN.
  02 ZEVERTY-DAYS                    Type ZSPI-DDL-INT.
  02 ZEVERTY-HOURS                   Type ZSPI-DDL-INT.
  02 ZEVERTY-MINUTES                 Type ZSPI-DDL-INT.
  02 ZDEFAULT-SECURITY               Type ZSPI-DDL-INT.
  02 ZPRI                            Type ZSPI-DDL-INT.
  02 ZSELPRI                         Type ZSPI-DDL-INT.
  02 ZHIGHPIN                        Type ZSPI-DDL-BOOLEAN.
(continued)

```

```

  02 ZMAXPRINTLINES                  Type ZSPI-DDL-INT2.
  02 ZMAXPRINTPAGES                  Type ZSPI-DDL-INT2.
  02 ZTAPEDRIVES                     Type ZSPI-DDL-INT.
  02 ZDATE.
    03 ZYEAR                         Type ZSPI-DDL-INT.
    03 ZMONTH                        Type ZSPI-DDL-INT.
    03 ZDAY                          Type ZSPI-DDL-INT.
  02 ZTIME.
    03 ZHOUR                         Type ZSPI-DDL-INT.
    03 ZMINUTE                       Type ZSPI-DDL-INT.
    03 ZSECOND                       Type ZSPI-DDL-INT.
    03 ZMILLISECOND                  Type ZSPI-DDL-INT.
    03 ZMICROSECOND                  Type ZSPI-DDL-INT.
  02 ZPOSIX                          Type ZSPI-DDL-INT.
  02 ZSAVEABEND                      Type ZSPI-DDL-BOOLEAN.
  02 ZRUND                           Type ZSPI-DDL-BOOLEAN.
  02 ZJOBID-ZERO                     Type ZSPI-DDL-BOOLEAN.
  02 ZMEM                            Type ZSPI-DDL-INT.
  02 ZPFS                            Type ZSPI-DDL-INT2.
  02 ZNAME                           Type ZSPI-DDL-CHAR8.
  02 ZINFO-TIME-SUBMIT                Type ZSPI-DDL-INT4.
  02 ZINFO-LAST-MOD                  Type ZSPI-DDL-INT4.
  02 ZINFO-LAST-MODUSER               Type ZSPI-DDL-INT.
  02 ZTIME-LIMIT                     Type ZSPI-DDL-INT2.
End

```

**ZCLASSNAME**

is the name of a class and specifies the job's CLASS attribute. The default value is the class specified by the scheduler's DEFAULT-CLASS attribute.

**ZHOLD**

is a Boolean field that specifies the job's HOLD attribute. The values are:

|                |                    |
|----------------|--------------------|
| ZSPI-VAL-FALSE | Specifies HOLD OFF |
| ZSPI-VAL-TRUE  | Specifies HOLD ON. |

The default value is ZSPI-VAL-FALSE.

**ZHOLD-AFTER**

is a Boolean field that specifies the job's HOLDAFTER attribute. The values are:

|                |                         |
|----------------|-------------------------|
| ZSPI-VAL-FALSE | Specifies HOLDAFTER OFF |
| ZSPI-VAL-TRUE  | Specifies HOLDAFTER ON. |

The default value is ZSPI-VAL-FALSE.

**ZRESTART**

is a Boolean field that specifies the job's RESTART attribute. The values are:

|                |                        |
|----------------|------------------------|
| ZSPI-VAL-FALSE | Specifies RESTART OFF. |
| ZSPI-VAL-TRUE  | Specifies RESTART ON.  |

The default value is ZSPI-VAL-FALSE.

**ZSTOP-ON-ABEND**

is a Boolean field that specifies the job's STOP-ON-ABEND attribute. The values are:

|                |                             |
|----------------|-----------------------------|
| ZSPI-VAL-FALSE | Specifies STOP-ON-ABEND OFF |
| ZSPI-VAL-TRUE  | Specifies STOP-ON-ABEND ON  |

The default value is the value of the scheduler's DEFAULT-STOP-ON-ABEND attribute.

**ZAT-FLAG**

is a Boolean field that specifies whether the job has the AFTER attribute or the AT attribute. The values are:

|                |   |
|----------------|---|
| ZSPI-VAL-FALSE | Specifies the AFTER attribute. The job is eligible to run after the time specified by ZINFO-NEXT-RUNTIME. |
| ZSPI-VAL-TRUE  | Specifies the AT attribute. The job runs at the time specified by ZINFO-NEXT-RUNTIME.                     |

The default value is ZSPI-VAL-FALSE.

**ZIFFAILS**

is a Boolean field that specifies the job's IFFAILS attribute. The values are:

|                |                       |
|----------------|-----------------------|
| ZSPI-VAL-FALSE | Specifies IFFAILS OFF |
| ZSPI-VAL-TRUE  | Specifies IFFAILS ON  |

The default value is ZSPI-VAL-FALSE.

**ZPURGE-IN-FILE**

is a Boolean field that specifies the job's PURGE-IN-FILE attribute. The values are:

|                |                             |
|----------------|-----------------------------|
| ZSPI-VAL-FALSE | Specifies PURGE-IN-FILE OFF |
| ZSPI-VAL-TRUE  | Specifies PURGE-IN-FILE ON  |

The default value is ZSPI-VAL-FALSE.

**ZSTALL**

is a Boolean field that specifies the job's STALL attribute. The values are:

|                |                     |
|----------------|---------------------|
| ZSPI-VAL-FALSE | Specifies STALL OFF |
| ZSPI-VAL-TRUE  | Specifies STALL ON  |

The default value is the value of the scheduler's DEFAULT-STALL attribute.

**ZINFO-NEXT-RUNTIME**

indicates the job's next run time in Greenwich-mean-time form.

**ZINFO-OUT-SPOOL-NUM**

contains the spooler job number of the job's log file. This value is supplied only when the job's output goes to a spooler location.

**ZINFO-WHICH-LIST**

is an enumerated value of ZBAT-DDL-JOB-WHICH-LIST that indicates the job's state. See [ZBAT-DDL-JOB-WHICH-LIST](#) on page 4-17.

**ZINFO-SPECIAL-REASON**

is an enumerated value of ZBAT-DDL-SPECIAL-REASON that qualifies the job's state when ZINFO-WHICH-LIST indicates the job is in a SPECIAL state. See [ZBAT-DDL-SPECIAL-REASON](#) on page 4-31.

**ZINFO-TOTAL-CPU-TIME**

indicates, in microseconds, the sum of the processor time taken by all processes of the job.

**ZINFO-OPEN-ACCESSOR**

contains the user ID of the job owner.

**ZRE MID**

is a Boolean field that indicates whether the job was submitted from a requester on the same node as the scheduler or from a remote node. The values are:

- |                       |  |
|-----------------------|--|
| <b>ZSPI-VAL-FALSE</b> | Indicates the job was submitted from a requester on the same node as the scheduler. For a job to run as a local process, the submitter must have local access to the scheduler's node. |
| <b>ZSPI-VAL-TRUE</b>  | Indicates the job was submitted from a requester on a node remote to the node of the scheduler.  |

**ZE VERY-DAYS**

assigns the EVERY attribute to the job and specifies the execution interval in days. ZE VERY-DAYS cannot be specified with ZE VERY-HOURS, ZE VERY-MINUTES, ZBAT-TKN-CALENDAR, and ZBAT-MAP-DEF-CRONTAB. The range is 1 through 365.

To remove the EVERY attribute from the job, set ZE VERY-DAYS, ZE VERY-HOURS, and ZE VERY-MINUTES to zero.

**ZE VERY-HOURS**

assigns the EVERY attribute to the job and specifies the execution interval in hours. ZE VERY-HOURS can be specified with ZE VERY-MINUTES but not with ZE VERY-DAYS, ZBAT-TKN-CALENDAR, or ZBAT-MAP-DEF-CRONTAB. The range is 0 through 168.

To remove the EVERY attribute from the job, set ZE VERY-DAYS, ZE VERY-HOURS, and ZE VERY-MINUTES to zero.

**ZE VERY-MINUTES**

assigns the EVERY attribute to the job and specifies the execution interval in minutes. ZE VERY-MINUTES can be specified with ZE VERY-HOURS, but not with ZE VERY-DAYS, ZBAT-TKN-CALENDAR, or ZBAT-MAP-DEF-CRONTAB. The range is 00 through 59.

To remove the EVERY attribute from the job, set ZE VERY-DAYS, ZE VERY-HOURS, and ZE VERY-MINUTES to zero.

**ZDEFAULT-SECURITY**

specifies a 16-bit signed integer that sets the default security for disk files the job creates. The security bits are:

- |                      |                             |
|----------------------|-----------------------------|
| <b>&lt;0:3&gt;</b>   | 0                           |
| <b>&lt;4:6&gt;</b>   | ID code allowed for read    |
| <b>&lt;7:9&gt;</b>   | ID code allowed for write   |
| <b>&lt;10:12&gt;</b> | ID code allowed for execute |
| <b>&lt;13:15&gt;</b> | ID code allowed for purge   |

ID code can be one of:

- 0 Any user (local)
- 1 Member of owner's group (local)
- 2 Owner (local)
- 4 Any user (local or remote)
- 5 Member of owner's community (local or remote)
- 6 Owner (local or remote)
- 7 Super ID only (local)

#### ZPRI

specifies the job's PRI attribute. The range is 1 through 199. The default value is the value of the scheduler's DEFAULT-PRI attribute.

#### ZSELPRI

specifies the job's SELPRI attribute. The range is 0 through 7. The default value is the value of the scheduler's DEFAULT-SELPRI attribute.

#### ZHIGHPIN

is a Boolean field that specifies the job's HIGHPIN attribute. The values are:

|                |                       |
|----------------|-----------------------|
| ZSPI-VAL-FALSE | Specifies HIGHPIN OFF |
| ZSPI-VAL-TRUE  | Specifies HIGHPIN ON  |

The default value is the value of the scheduler's DEFAULT-HIGHPIN attribute.

#### ZMAXPRINTLINES

specifies the job's MAXPRINTLINES attribute. The value is 0 (for no maximum) or is a number in the range 120 through 65534. The default value is the value of the scheduler's DEFAULT-MAXPRINTLINES attribute.

#### ZMAXPRINTPAGES

specifies the job's MAXPRINTPAGES attribute. The value is 0 (for no maximum) or a number in the range 2 through 65534. The default value is the value of the scheduler's DEFAULT-MAXPRINTPAGES attribute.

#### ZTAPEDRIVES

specifies the job's TAPEDRIVES attribute. The range is 0 through 99. The default value is 0.

#### ZDATE

specifies the date on the scheduler's node on which the job becomes eligible for execution. If ZDATE is present, all its fields must be specified. If omitted, the current date applies. The fields are:

**ZYEAR**

is an integer specifying the Gregorian year. The range is the current year minus one to the year 2525.

**ZMONTH**

is an integer specifying the Gregorian month of ZYEAR. The range is 1 through 12.

**ZDAY**

is an integer specifying the Gregorian day of ZMONTH. The range is 1 through 31.

**ZTIME**

specifies the time on ZDATE when the job becomes eligible for execution. If ZTIME is present, all its fields must be specified. If omitted, the current time applies. The fields are:

**ZHOUR**

is an integer specifying the hour of the day. The range is 0 through 23.

**ZMINUTE**

is an integer specifying the minute of the hour. The range is 0 through 59.

**ZSECOND**

is an integer specifying the second of the minute. The range is 0 through 59.

**ZMILLISECOND**

is an integer specifying the millisecond of the second. The range is 0 through 999.

**ZMICROSECOND**

is an integer specifying the microsecond of the millisecond. The range is 0 through 999.

The scheduler treats 23:59:59:999.999 as midnight. This enables an application to specify AFTER MIDNIGHT without having to increment the date.

**ZPOSIX**

is a word supplied by an HP NonStop Open System Services (OSS) requester and used in job-owner identification. The word has a nonzero value for an OSS job and a zero value for a Guardian job.

**ZSAVEABEND**

is a Boolean field that specifies the job's SAVEABEND attribute. The values are:

|                |                         |
|----------------|-------------------------|
| ZSPI-VAL-FALSE | Specifies SAVEABEND OFF |
| ZSPI-VAL-TRUE  | Specifies SAVEABEND ON  |

The default value is the value of the executor program's saveabend flag (set when compiling or binding the program).

**ZRUND**

is a Boolean field that specifies the job's RUND attribute. The values are:

|                |                    |
|----------------|--------------------|
| ZSPI-VAL-FALSE | Specifies RUND OFF |
| ZSPI-VAL-TRUE  | Specifies RUND ON  |

The default value is ZSPI-VAL-FALSE.

**ZJOBID-ZERO**

is a Boolean field that specifies the job's JOBID-ZERO attribute. The values are:

|                |                          |
|----------------|--------------------------|
| ZSPI-VAL-FALSE | Specifies JOBID-ZERO OFF |
| ZSPI-VAL-TRUE  | Specifies JOBID-ZERO ON  |

The default value is ZSPI-VAL-FALSE.

**ZMEM**

specifies the job's MEM attribute. The range is 1 through 64 2048-byte memory pages. The default value is the value specified in the executor program.

**ZPFS**

specifies the job's PFS attribute. The value is 0 (for the value specified in the executor program) or a number in the range 131,072 bytes (128 kilobytes [KB], or one segment) through 1,048,576 bytes (1024 KB, or eight segments). The default value is the value specified in the executor program.

**ZNAME**

specifies the job's NAME attribute. The value is a unique process name. The default value is a system-generated process name.

**ZINFO-TIME-SUBMIT**

indicates the Greenwich mean time at which the job was submitted to the scheduler.

**ZINFO-LAST-MOD**

indicates the Greenwich mean time at which the job was last altered.



~~ZINFO-LAST-MODUSER~~

is the user ID of the last user to alter the job.

~~ZTIME-LIMIT~~

specifies the job's LIMIT attribute. The range is 0 through 999 hours and 0 through 59 minutes. The default value is 0 hours and 0 minutes.

~~ZBAT-MAP-DEF-SCHEDULER~~

is an extensible structured token that contains information about a scheduler. The token's usage in scheduler commands is:

| <b>Command</b>      | <b>ZBAT-MAP-DEF-SCHEDULER</b> |
|---------------------|-------------------------------|
| ABORT SCHEDULER     | Not used                      |
| ADD SCHEDULER       | Required in command buffer    |
| ALTER SCHEDULER     | Required in command buffer    |
| INFO SCHEDULER      | Returned in response buffer   |
| SHUTDOWN SCHEDULER  | Not used                      |
| START SCHEDULER     | Not used                      |
| STATUS SCHEDULER    | Not used                      |
| SWITCHCPU SCHEDULER | Not used                      |
| SWITCHLOG SCHEDULER | Not used                      |

ZBAT-MAP-DEF-SCHEDULER has the structure:

```

Definition ZBAT-DDL-DEF-SCHEDULER.
02 ZBACKUPCPU2                                Type ZSPI-DDL-INT.
02 ZBACKUPCPU1                                Type ZSPI-DDL-INT.
02 ZMAXCONCURRENTJOBS                        Type ZSPI-DDL-INT.
02 ZMAXTEMPEXECUTORS                         Type ZSPI-DDL-INT.
02 ZTAPEDRIVES                               Type ZSPI-DDL-INT.
02 ZMAXPRI                                   Type ZSPI-DDL-INT.
02 ZINFO-TAPEDRIVES-IN-USE                   Type ZSPI-DDL-INT.
02 ZAT-ALLOWED                              Type ZSPI-DDL-BOOLEAN.
02 ZSUBMIT-ALLOWED                          Type ZSPI-DDL-BOOLEAN.
02 ZEVERY-CATCHUP                           Type ZSPI-DDL-BOOLEAN.
02 ZEMS                                       Type ZBAT-DDL-EMS.
02 ZCLASSNAME                               Type ZBAT-DDL-NETBATCH-NAME.
02 ZPRI                                       Type ZSPI-DDL-INT.
02 ZSELPRI                                  Type ZSPI-DDL-INT.
02 ZMAXPRINTLINES                           Type ZSPI-DDL-INT2.
02 ZMAXPRINTPAGES                           Type ZSPI-DDL-INT2.
02 ZSTOP-ON-ABEND                           Type ZSPI-DDL-BOOLEAN.
02 ZSTALL                                    Type ZSPI-DDL-BOOLEAN.
02 ZHIGHPIN                                 Type ZSPI-DDL-BOOLEAN.
02 ZINITIATION                              Type ZSPI-DDL-BOOLEAN.
02 ZLOCALNAMES                              Type ZSPI-DDL-CHAR8
                                           Occurs 30 times.

End

```

~~ZBACKUPCPU2~~

specifies the second processor preference for the scheduler's backup process. ZBACKUPCPU2, with ZBACKUPCPU1, specifies the scheduler's

BACKUPCPU attribute. The default value of ZBACKUPCPU2 when cold starting the scheduler is \*.

#### ZBACKUPCPU1

specifies the first processor preference for the scheduler's backup process. ZBACKUPCPU1, with ZBACKUPCPU2, specifies the scheduler's BACKUPCPU attribute. The default value of ZBACKUPCPU1 when cold starting the scheduler is the processor of the scheduler's primary process.

#### ZMAXCONCURRENTJOBS

specifies the concurrent-jobs limit for the scheduler. ZMAXCONCURRENTJOBS, with ZMAXTEMPEXECUTORS, specifies the scheduler's MAX-CONCURRENT-JOBS attribute. The default value of ZMAXCONCURRENTJOBS when cold starting the scheduler is 500.

#### ZMAXTEMPEXECUTORS

specifies the temporary-executors limit for the scheduler. ZMAXTEMPEXECUTORS, with ZMAXCONCURRENTJOBS, specifies the scheduler's MAX-CONCURRENT-JOBS attribute. The default value of ZMAXTEMPEXECUTORS when cold starting the scheduler is 500.

#### ZTAPEDRIVES

specifies the scheduler's TAPEDRIVES attribute. The range is 0 through 99. The default value when cold starting the scheduler is 2.

#### ZMAXPRI

specifies the scheduler's MAX-PRI attribute. The range is 1 through 199. The default value when cold starting the scheduler is 199.

#### ZINFO-TAPEDRIVES-IN-USE

is the number of tape drives in use.

#### ZAT-ALLOWED

is a Boolean field that specifies the scheduler's AT-ALLOWED attribute. The values are:

ZSPI-VAL-FALSE

Specifies AT-ALLOWED OFF

ZSPI-VAL-TRUE

Specifies AT-ALLOWED ON

The default value when cold starting the scheduler is ZSPI-VAL-FALSE.

**ZSUBMIT-ALLOWED**

is a Boolean field that specifies the scheduler's SUBMIT-ALLOWED attribute. The values are:

|                |                              |
|----------------|------------------------------|
| ZSPI-VAL-FALSE | Specifies SUBMIT-ALLOWED OFF |
| ZSPI-VAL-TRUE  | Specifies SUBMIT-ALLOWED ON  |

The default value when cold starting the scheduler is ZSPI-VAL-TRUE.

**ZEVERY-CATCHUP**

is a Boolean field that specifies the scheduler's CATCHUP attribute. The values are:

|                |                       |
|----------------|-----------------------|
| ZSPI-VAL-FALSE | Specifies CATCHUP OFF |
| ZSPI-VAL-TRUE  | Specifies CATCHUP ON  |

The default value when cold starting the scheduler is ZSPI-VAL-TRUE.

**ZEMS**

is an enumerated value of ZBAT-DDL-EMS that specifies the scheduler's EMS attribute. See [ZBAT-DDL-EMS](#) on page 4-15.

**ZCLASSNAME**

is the name of a class and specifies the scheduler's DEFAULT-CLASS attribute. The default value when cold starting the scheduler is DEFAULT.

**ZPRI**

specifies the scheduler's DEFAULT-PRI attribute. The range is 1 through 199. The default value when cold starting the scheduler is 120.

**ZSELPRI**

specifies the scheduler's DEFAULT-SELPRI attribute. The range is 0 through 7. The default value when cold starting the scheduler is 3.

**ZMAXPRINTLINES**

specifies the scheduler's DEFAULT-MAXPRINTLINES attribute. The value is 0 (for no maximum) or is a number in the range 120 through 65534. The default value when cold starting the scheduler is 0.

**ZMAXPRINTPAGES**

specifies the scheduler's DEFAULT-MAXPRINTPAGES attribute. The value is 0 (for no maximum) or is a number in the range 2 through 65534. The default value when cold starting the scheduler is 0.

**ZSTOP-ON-ABEND**

is a Boolean field that specifies the scheduler's DEFAULT-STOP-ON-ABEND attribute. The values are:

|                |                                     |
|----------------|-------------------------------------|
| ZSPI-VAL-FALSE | Specifies DEFAULT-STOP-ON-ABEND OFF |
| ZSPI-VAL-TRUE  | Specifies DEFAULT-STOP-ON-ABEND ON  |

The default value when cold starting the scheduler is ZSPI-VAL-FALSE.

**ZSTALL**

is a Boolean field that specifies the scheduler's DEFAULT-STALL attribute. The values are:

|                |                             |
|----------------|-----------------------------|
| ZSPI-VAL-FALSE | Specifies DEFAULT-STALL OFF |
| ZSPI-VAL-TRUE  | Specifies DEFAULT-STALL ON  |

The default value when cold starting the scheduler is ZSPI-VAL-FALSE.

**ZHIGHPIN**

is a Boolean field that specifies the scheduler's DEFAULT-HIGHPIN attribute. The values are:

|                |                               |
|----------------|-------------------------------|
| ZSPI-VAL-FALSE | Specifies DEFAULT-HIGHPIN OFF |
| ZSPI-VAL-TRUE  | Specifies DEFAULT-HIGHPIN ON  |

The default value when cold starting the scheduler is ZSPI-VAL-FALSE.

**ZINITIATION**

is a Boolean field that specifies the scheduler's INITIATION attribute. The values are:

|                |                          |
|----------------|--------------------------|
| ZSPI-VAL-FALSE | Specifies INITIATION OFF |
| ZSPI-VAL-TRUE  | Specifies INITIATION ON  |

The default value when cold starting the scheduler is ZSPI-VAL-TRUE.

**ZLOCALNAMES**

is the name of a node remote to the node of the scheduler. ZLOCALNAMES can occur up to 30 times and specifies the scheduler's LOCALNAMES attribute.

**ZBAT-MAP-DEF-WAITON**

is an extensible structured token containing details of a job's dependent relationship with one of its masters. (A dependent job can have up to eight masters, each specified by one ZBAT-MAP-DEF-WAITON token. Collectively, the masters make up the job's WAITON attribute.) The token's usage in job commands is:

| <b>Command</b> | <b>ZBAT-MAP-DEF-WAITON</b>  |
|----------------|-----------------------------|
| ACTIVATE JOB   | Not used                    |
| ALTER JOB      | Optional in command buffer  |
| DELETE JOB     | Not used                    |
| INFO JOB       | Returned in response buffer |
| RELEASE JOB    | Not used                    |
| RUNNEXT JOB    | Not used                    |
| RUNNOW JOB     | Not used                    |
| STATUS JOB     | Not used                    |
| STOP JOB       | Not used                    |
| SUBMIT JOB     | Optional in command buffer  |
| SUSPEND JOB    | Not used                    |

ZBAT-MAP-DEF-WAITON has the structure:

|                                 |                                 |
|---------------------------------|---------------------------------|
| Definition ZBAT-DDL-DEF-WAITON. |                                 |
| 02 ZMASTER                      | Type ZBAT-DDL-NETBATCH-NAME.    |
| 02 ZINDICATOR                   | Type ZBAT-DDL-WAITON-INDICATOR. |
| 02 ZFOR                         | Type ZBAT-DDL-WAITON-FOR.       |
| End                             |                                 |

#### ZMASTER

is the name of the job's master. Specifying spaces or a null value removes the WAITON attribute from the job.

#### ZINDICATOR

is an enumerated value of ZBAT-DDL-WAITON-INDICATOR that gives information about the job's dependent relationship with ZMASTER. See [ZBAT-DDL-WAITON-INDICATOR](#) on page 4-32.

#### ZFOR

is an enumerated value of ZBAT-DDL-WAITON-FOR that indicates the type of release a dependent job requires from its master. See [ZBAT-DDL-WAITON-FOR](#) on page 4-32.

#### ZBAT-MAP-DEFINE-ERROR

is an extensible structured token that contains details of the error detected by the scheduler when the scheduler validated ZBAT-TKN-ATT-SET-DEFINE. The token's usage in attachment-set commands is:

| <b>Command</b>        | <b>ZBAT-MAP-DEFINE-ERROR</b> |
|-----------------------|------------------------------|
| ADD ATTACHMENT-SET    | Optional in command buffer   |
| ALTER ATTACHMENT-SET  | Optional in command buffer   |
| DELETE ATTACHMENT-SET | Not used                     |
| INFO ATTACHMENT-SET   | Returned in response buffer  |
| STATUS ATTACHMENT-SET | Not used                     |

ZBAT-MAP-DEFINE-ERROR has the structure:

|                                   |                       |
|-----------------------------------|-----------------------|
| Definition ZBAT-DDL-DEFINE-ERROR. |                       |
| 02 ZNAMELEN                       | Type ZSPI-DDL-INT.    |
| 02 ZNAMETXT                       | Type ZSPI-DDL-CHAR24. |
| 02 ZERR                           | Type ZSPI-DDL-INT.    |
| 02 ZATTRLEN                       | Type ZSPI-DDL-INT.    |
| 02 ZATTRTXT                       | Type ZSPI-DDL-CHAR16. |
| 02 ZCLASSLEN                      | Type ZSPI-DDL-INT.    |
| 02 ZCLASSTXT                      | Type ZSPI-DDL-CHAR16. |
| 02 ZCHECKNUM                      | Type ZSPI-DDL-INT.    |
| 02 ZADDR                          | Type ZSPI-DDL-INT.    |
| End                               |                       |

ZNAMELEN

is the number of characters returned in ZNAMETXT.

ZNAMETXT

is the name of the DEFINE in error, left justified and space-filled.

ZERR

is the number of the error returned by Guardian procedure DEFINESETATTR or DEFINEVALIDATEWORK.

ZATTRLEN

is the number of characters returned in ZATTRTXT.

ZATTRTXT

is the DEFINE attribute that is in error, left-justified and space-filled.

ZCLASSLEN

is the number of characters returned in ZCLASSTXT.

ZCLASSTXT

is the class of the DEFINE in error, left-justified and space-filled.

ZCHECKNUM

is a check number returned by Guardian procedure DEFINEVALIDATEWORK.

ZADDR

is the offset in bytes into the string ZBAT-TKN-ATT-SET-DEFINE where the error occurred.

**ZBAT-MAP-PAR-RELEASE-JOB**

is an extensible structured token that a master job uses to release one or more of its dependents. The token's usage in job commands is:

| <b>Command</b> | <b>ZBAT-MAP-PAR-RELEASE-JOB</b> |
|----------------|---------------------------------|
| ACTIVATE JOB   | Not used                        |
| ALTER JOB      | Not used                        |
| DELETE JOB     | Not used                        |
| INFO JOB       | Not used                        |
| RELEASE JOB    | Required in command buffer      |
| RUNNEXT JOB    | Not used                        |
| RUNNOW JOB     | Not used                        |
| STATUS JOB     | Not used                        |
| STOP JOB       | Not used                        |
| SUBMIT JOB     | Not used                        |
| SUSPEND JOB    | Not used                        |

ZBAT-MAP-PAR-RELEASE-JOB has the structure:

|                                      |                              |
|--------------------------------------|------------------------------|
| Definition ZBAT-DDL-PAR-RELEASE-JOB. |                              |
| 02 ZRELEASER                         | Type ZBAT-DDL-NETBATCH-NAME. |
| 02 ZJOBNAME                          | Type ZBAT-DDL-NETBATCH-NAME. |
| End                                  |                              |

**ZRELEASER**

is the name of the master job. It is required only when ZJOBNAME specifies a job or jobs in a scheduler that is not the master-job's scheduler.

**ZJOBNAME**

specifies the name of the dependent job or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of names.

**ZBAT-MAP-STATUS-EXECUTOR**

is an extensible structured token that contains executor status information. The token's usage in executor commands is:

| <b>Command</b>  | <b>ZBAT-MAP-STATUS-EXECUTOR</b> |
|-----------------|---------------------------------|
| ADD EXECUTOR    | Not used                        |
| ALTER EXECUTOR  | Not used                        |
| DELETE EXECUTOR | Not used                        |
| INFO EXECUTOR   | Not used                        |
| START EXECUTOR  | Not used                        |
| STATUS EXECUTOR | Returned in response buffer     |
| STOP EXECUTOR   | Not used                        |

ZBAT-MAP-STATUS-EXECUTOR has the structure:

|                                      |                              |
|--------------------------------------|------------------------------|
| Definition ZBAT-DDL-STATUS-EXECUTOR. |                              |
| 02 ZCPU                              | Type ZSPI-DDL-INT.           |
| 02 ZJOBNUMBER                        | Type ZSPI-DDL-INT.           |
| 02 ZWHICH-LIST                       | Type ZBAT-DDL-EXECUTOR-LIST. |
| 02 ZCLASS                            | Type ZBAT-DDL-NETBATCH-NAME. |
| End                                  |                              |

ZCPU

is the number of the executor's processor.

ZJOBNUMBER

is the number of the job using the executor.

ZWHICH-LIST

is an enumerated value of ZBAT-DDL-EXECUTOR-LIST that indicates the executor's state. For a description of ZBAT-DDL-EXECUTOR-LIST, see [Private Token and Field Types](#) earlier in this subsection.

ZCLASS

is the name of the class of the job using the executor.

ZBAT-MAP-STATUS-JOB

is an extensible structured token that contains job status information. The token's usage in job commands is:

| Command      | ZBAT-MAP-STATUS-JOB          |
|--------------|------------------------------|
| ACTIVATE JOB | Not used.                    |
| ALTER JOB    | Not used.                    |
| DELETE JOB   | Not used.                    |
| INFO JOB     | Not used.                    |
| RELEASE JOB  | Not used.                    |
| RUNNEXT JOB  | Not used.                    |
| RUNNOW JOB   | Not used.                    |
| STATUS JOB   | Returned in response buffer. |
| STOP JOB     | Not used.                    |
| SUBMIT JOB   | Not used.                    |
| SUSPEND JOB  | Not used.                    |



ZBAT-MAP-STATUS-JOB has the structure:

|                                 |                               |
|---------------------------------|-------------------------------|
| Definition ZBAT-DDL-STATUS-JOB. |                               |
| 02 ZOUT-SPOOL-NUM               | Type ZSPI-DDL-INT.            |
| 02 ZSELPRI                      | Type ZSPI-DDL-INT.            |
| 02 ZOPEN-ACCESSOR-DETAIL.       |                               |
| 03 ZGROUP                       | Type ZSPI-DDL-BYTE.           |
| 03 ZUSER                        | Type ZSPI-DDL-BYTE.           |
| 02 ZOPEN-ACCESSOR               | Redefines ZOPEN-ACCESSOR-     |
| DETAIL                          |                               |
|                                 | Type ZSPI-DDL-INT.            |
| 02 ZCLASSNAME                   | Type ZBAT-DDL-NETBATCH-NAME.  |
| 02 ZWHICH-LIST                  | Type ZBAT-DDL-JOB-WHICH-LIST. |
| 02 ZSPECIAL-REASON              | Type ZBAT-DDL-SPECIAL-REASON. |
| 02 ZNEXT-RUNTIME                | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-PREV-RUNTIME           | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-START                  | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-FINISH                 | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-PUT-ON-LIST            | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-USED                   | Type ZSPI-DDL-INT4.           |
| 02 ZREPID                       | Type ZSPI-DDL-BOOLEAN.        |
| 02 ZEXECUTOR                    | Type ZBAT-DDL-NETBATCH-NAME.  |
| 02 ZEXECPHANDLE                 | Type ZSPI-DDL-PHANDLE.        |
| 02 ZTIME-ELAPSEDMAX             | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-CPUMAX                 | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-ELAPSEDTOTAL           | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-CPUTOTAL               | Type ZSPI-DDL-INT4.           |
| 02 ZTIME-SUBMIT                 | Type ZSPI-DDL-INT4.           |
| 02 ZLAST-CC                     | Type ZSPI-DDL-INT.            |
| 02 ZTIMES-RUN                   | Type ZSPI-DDL-INT2.           |
| 02 ZTIME-LIMIT                  | Type ZSPI-DDL-INT2.           |
| End                             |                               |

ZOUT-SPOOL-NUM

contains the spooler job number of the job's log file. This value is only supplied when the job's output goes to a spooler location.

ZSELPRI

is the value of the job's SELPRI attribute.

ZOPEN-ACCESSOR-DETAIL

is the user ID of the job's owner. The fields are:

ZGROUP

is the owner's group number.

ZUSER

is the owner's user number.

ZOPEN-ACCESSOR

redefines ZOPEN-ACCESSOR-DETAIL as a single value comprising the job owner's group number and user number.

**ZCLASSNAME**

is the name of the job's class.

**ZWHICH-LIST**

is an enumerated value of ZBAT-DDL-JOB-WHICH-LIST that indicates the job's state. See [ZBAT-DDL-JOB-WHICH-LIST](#) on page 4-17.

**ZSPECIAL-REASON**

is an enumerated value of ZBAT-DDL-SPECIAL-REASON that qualifies the job's state when ZWHICH-LIST indicates the job is in a SPECIAL state. See [ZBAT-DDL-SPECIAL-REASON](#) on page 4-31.

**ZNEXT-RUNTIME**

indicates the job's next run time in Greenwich-mean-time form.

**ZTIME-PREV-RUNTIME**

records the time the job last ran in Greenwich-mean-time form.

**ZTIME-START**

records the job's start time in Greenwich-mean-time form.

**ZTIME-FINISH**

records the job's finish time in Greenwich-mean-time form. (The finish time is the time when the last running process of the job stops.)

**ZTIME-PUT-ON-LIST**

records the time when ZWHICH-LIST was set in Greenwich-mean-time form.

**ZTIME-USED**

indicates, in microseconds, the total processor time taken by all processes of the job.

**ZREMID**

is a Boolean field that indicates whether the job was submitted from a requester on the same node as the scheduler or from a requester on a different node. The values are:

**ZSPI-VAL-FALSE** Indicates the job was submitted from a requester on the same node as the scheduler

**ZSPI-VAL-TRUE** Indicates the job was submitted from a requester on a node different from that of the scheduler

**ZEXECUTOR**

is the name of the job's executor.

**ZEXECPHANDLE**

is the process handle of the job's executor-program process.

**ZTIME-ELAPSEDMAX**

indicates, in microseconds, the time taken by the longest run of the job.

**ZTIME-CPUMAX**

indicates, in microseconds, the processor time taken by all processes of the longest run of the job.

**ZTIME-ELAPSEDTOTAL**

indicates, in microseconds, the total time taken by all runs of the job.

**ZTIME-CPUTOTAL**

indicates, in microseconds, the total processor time taken by all processes of all runs of the job.

**ZTIME-SUBMIT**

indicates the Greenwich mean time at which the job was submitted to the scheduler.

**ZLAST-CC**

indicates the completion code returned by the job's executor-program process the last time the job ran.

**ZTIMES-RUN**

indicates the number of times the job has run.

**ZTIME-LIMIT**

indicates the time specified by the job's LIMIT attribute.

**ZBAT-MAP-STATUS-SCHEDULER**

is an extensible structured token that contains scheduler status information. The token's usage in scheduler commands is:

| <b>Command</b>     | <b>ZBAT-MAP-STATUS-SCHEDULER</b> |
|--------------------|----------------------------------|
| ABORT SCHEDULER    | Not used                         |
| ADD SCHEDULER      | Not used                         |
| ALTER SCHEDULER    | Not used                         |
| INFO SCHEDULER     | Not used                         |
| SHUTDOWN SCHEDULER | Not used                         |
| START SCHEDULER    | Not used                         |

| Command             | ZBAT-MAP-STATUS-SCHEDULER   |
|---------------------|-----------------------------|
| STATUS SCHEDULER    | Returned in response buffer |
| SWITCHCPU SCHEDULER | Not used                    |
| SWITCHLOG SCHEDULER | Not used                    |

ZBAT-MAP-STATUS-SCHEDULER has the structure:

```

Definition ZBAT-DDL-STATUS-SCHEDULER.
02 ZSTATE                                     Type ZBAT-DDL-SCHEDULER-STATE.
02 ZEXECUTOR.
    03 ZOFF                                     Type ZSPI-DDL-INT.
    03 ZON                                     Type ZSPI-DDL-INT.
    03 ZACTIVE                                Type ZSPI-DDL-INT.
    03 ZSTOP                                  Type ZSPI-DDL-INT.
    03 ZDOWN                                  Type ZSPI-DDL-INT.
    03 ZDELETE                                Type ZSPI-DDL-INT.
02 ZJOB.
    03 ZREADY                                 Type ZSPI-DDL-INT.
    03 ZEXECUTING                             Type ZSPI-DDL-INT.
    03 ZSPECIAL                               Type ZSPI-DDL-INT.
    03 ZTIME                                  Type ZSPI-DDL-INT.
    03 ZEVEN                                    Type ZSPI-DDL-INT.
    03 ZSUSPENDED                             Type ZSPI-DDL-INT.
    03 ZRUNNEXT                               Type ZSPI-DDL-INT.
    03 ZRUNNOW                                Type ZSPI-DDL-INT.
    03 ZTAPE                                  Type ZSPI-DDL-INT.
02 ZJOBCLASS.
    03 ZOFF                                     Type ZSPI-DDL-INT.
    03 ZON                                     Type ZSPI-DDL-INT.
02 ZPROCESS.
    03 ZACTIVE                                Type ZSPI-DDL-INT.
    03 ZSUSPENDED                             Type ZSPI-DDL-INT.
02 ZTAPE.
    03 ZCONFIG                                 Type ZSPI-DDL-INT.
    03 ZTAPEDRIVES-IN-USE                     Type ZSPI-DDL-INT.
02 ZATT-SET-COUNT                             Type ZSPI-DDL-INT.
02 ZINITIATION                               Type ZSPI-DDL-BOOLEAN.
02 ZSUBMIT-ALLOWED                           Type ZSPI-DDL-BOOLEAN.
End

```

#### ZSTATE

is an enumerated value of ZBAT-DDL-SCHEDULER-STATE that indicates the scheduler's state. See [ZBAT-DDL-SCHEDULER-STATE](#) on page 4-31.

#### ZEXECUTOR

indicates the number of executors in each executor state. The fields are:

##### ZOFF

is the number of executors in the OFF state.

##### ZON

is the number of executors in the ON state.

##### ZACTIVE

is the number of executors in the ACTIVE state.

ZSTOP

is the number of executors in the STOP state.

ZDOWN

is the number of executors in the DOWN state.

ZDELETE

is the number of executors in the DELETE state.

ZJOB

indicates the number of jobs in each job state. The fields are:

ZREADY

is the number of jobs in the READY state.

ZEXECUTING

is the number of jobs in the EXECUTING and OVER LIMIT states.

ZSPECIAL

is the number of jobs in the SPECIAL state.

ZTIME

is the number of jobs in the TIME state.

ZEVENT

is the number of jobs in the EVENT state.

ZSUSPENDED

is the number of jobs in the SUSPENDED state.

ZRUNNEXT

is the number of jobs in the RUNNEXT state.

ZRUNNOW

is the number of jobs in the RUNNOW state.

ZTAPE

is the number of jobs in the READY state.

**ZJOBCLASS**

indicates the number of classes with the attribute INITIATION OFF and the number of classes with the attribute INITIATION ON. The fields are:

**ZOFF**

is the number of classes whose INITIATION attribute is set to OFF.

**ZON**

is the number of classes whose INITIATION attribute is set to ON.

**ZPROCESS**

indicates the number of active and suspended job processes. The fields are:

**ZACTIVE**

is the number of active job processes.

**ZSUSPENDED**

is the number of suspended job processes.

**ZTAPE**

indicates the number of tape drives specified by the scheduler's TAPEDRIVES attribute and the number of those drives in use by jobs. The fields are:

**ZCONFIG**

is the number of tape drives specified by the scheduler's TAPEDRIVES attribute.

**ZTAPEDRIVES-IN-USE**

is the number of tape drives in use.

**ZATT-SET-COUNT**

is the number of attachment sets defined in the scheduler.

**ZINITIATION**

is a Boolean field that indicates the value of the scheduler's INITIATION attribute. The values are:

ZSPI-VAL-FALSE

Specifies INITIATION OFF

ZSPI-VAL-TRUE

Specifies INITIATION ON

**ZSUBMIT-ALLOWED**

is a Boolean field that indicates the value of the scheduler's SUBMIT-ALLOWED attribute. The values are:

ZSPI-VAL-FALSE

Specifies SUBMIT-ALLOWED OFF

ZSPI-VAL-TRUE

Specifies SUBMIT-ALLOWED ON





# Commands and Responses

This section describes the syntax and semantics of all NetBatch programmatic commands and the responses to those commands. The command descriptions appear in alphabetic order. For each command, the section gives:

- The command name
- A command description
- A box containing:
  - The symbolic names of the command and the object on which it operates.
  - Lists of the tokens that can be present in the command and response buffers.

The order of the tokens in the lists of tokens in the command buffer and response buffer is not necessarily the order in which they actually appear in a command or response, except:

- The token ZSPI-TKN-DATALIST, if present in a response, always appears at the beginning of a response record.
- The token ZSPI-TKN-ERRLIST, if present in a response, always appears at the beginning of an error list.
- The token ZSPI-TKN-ENDLIST always appears at the end of a response record that begins with the token ZSPI-TKN-DATALIST or at the end of an error list that begins with the token ZSPI-TKN-ERRLIST.

Except for the context token ZSPI-TKN-CONTEXT, the list of tokens in the response buffer represents the tokens that can be present in one reply message that consists of a single or multiple response records. For more information on such responses and on the context token, see the *SPI Programming Manual*.

For each token in the command and response buffers, and each field of an extensible structured token in the command and response buffers, there is an indication of whether the token must be present in the command and of other dependencies. For a description of the notation scheme used, see [Notation Conventions](#) on page -xix.

Error lists are indicated by the token ZSPI-TKN-ERRLIST followed by an ellipsis (...) and the token ZSPI-TKN-ENDLIST. The tokens following token ZSPI-TKN-DATALIST, including ZSPI-TKN-ERRLIST, are indented to show they are in the data list. The value of the token ZSPI-TKN-MAXRESP determines whether data lists are used in the response. For more information on data lists and error lists, see the *SPI Programming Manual*.

- Information about the tokens in the command and response buffers that is not covered in [Section 4, Common Definitions](#).

- Operational notes stating which users can issue the command and any applicable programming considerations.

Preceding the [Command Descriptions](#) on page 5-9 are:

| Topic  | Page                |
|--|---------------------|
| <a href="#">Available Commands</a>                     | <a href="#">5-2</a> |
| <a href="#">Symbolic Names of Commands and Objects</a> | <a href="#">5-3</a> |
| <a href="#">Sensitive and Nonsensitive Commands</a>    | <a href="#">5-5</a> |
| <a href="#">Wild-Card Characters</a>                   | <a href="#">5-8</a> |

In this section, command numbers, object types, tokens, and token values appear in DDL format. Definitions of structures use DDL definition statements. For an explanation of DDL, see the *Data Definition Language (DDL) Reference Manual*.

## Available Commands

[Table 5-1](#) lists the commands available in the NetBatch programmatic interfaces and gives a summary of their functions. For detailed command descriptions, see the *NetBatch Manual*.

**Table 5-1. Available Commands** (page 1 of 2)

| Commands  | Function  |
|---|---|
| ABORT SCHEDULER<br>SHUTDOWN SCHEDULER   | Shut down schedulers  |
| ACTIVATE JOB<br>SUSPEND JOB   | Suspend executing and over-limit jobs and reactivate suspended jobs           |
| ADD ATTACHMENT-SET<br>ADD CLASS<br>ADD EXECUTOR                                       | Add attachment sets, classes, and executors to schedulers                     |
| ADD SCHEDULER   | Create and initialize scheduler databases                                     |
| ALTER ATTACHMENT-SET<br>ALTER CLASS<br>ALTER EXECUTOR<br>ALTER JOB<br>ALTER SCHEDULER | Alter attributes of attachment sets, classes, executors, jobs, and schedulers |
| DELETE ATTACHMENT-SET<br>DELETE CLASS<br>DELETE EXECUTOR<br>DELETE JOB                | Delete attachment sets, classes, executors, and jobs                          |
| GETVERSION  | Return version of the NetBatch subsystem                                      |

**Table 5-1. Available Commands** (page 2 of 2)

| Commands   | Function   |
|--|--|
| INFO ATTACHMENT-SET<br>INFO CLASS<br>INFO EXECUTOR<br>INFO JOB<br>INFO SCHEDULER | List attributes of attachment sets, classes, executors, jobs, and schedulers |
| RELEASE JOB  | Release dependent jobs   |
| RUNNEXT JOB<br>RUNNOW JOB  | Override job dependencies, timing attributes, and selection priorities       |
| START EXECUTOR<br>START SCHEDULER  | Start executors and schedulers   |
| STATUS ATTACHMENT-SET<br>STATUS EXECUTOR<br>STATUS JOB<br>STATUS SCHEDULER       | Display attachment set, executor, job, and scheduler status                  |
| STOP EXECUTOR<br>STOP JOB  | Stop executors and jobs  |
| SUBMIT JOB   | Submit jobs to schedulers  |
| SWITCHCPU SCHEDULER<br>SWITCHLOG SCHEDULER                                       | Switch scheduler processors and log files                                    |

## Symbolic Names of Commands and Objects

[Table 5-2](#) lists NetBatch programmatic commands and object types and gives their symbolic names.

**Table 5-2. Symbolic Names of Commands and Objects** (page 1 of 2)

| Command  | Object Type   | Symbolic Name of Command | Symbolic Name of Object Type  |
|----------|---|--------------------------|---|
| ABORT    | SCHEDULER   | ZBAT-CMD-ABORT           | ZBAT-OBJ-SCHEDULER  |
| ACTIVATE | JOB   | ZBAT-CMD-ACTIVATE        | ZBAT-OBJ-JOB  |
| ADD      | ATTACHMENT-SET<br>CLASS<br>EXECUTOR<br>SCHEDULER        | ZBAT-CMD-ADD             | ZBAT-OBJ-ATT-SET<br>ZBAT-OBJ-CLASS<br>ZBAT-OBJ-EXECUTOR<br>ZBAT-OBJ-SCHEDULER                 |
| ALTER    | ATTACHMENT-SET<br>CLASS<br>EXECUTOR<br>JOB<br>SCHEDULER | ZBAT-CMD-ALTER           | ZBAT-OBJ-ATT-SET<br>ZBAT-OBJ-CLASS<br>ZBAT-OBJ-EXECUTOR<br>ZBAT-OBJ-JOB<br>ZBAT-OBJ-SCHEDULER |

**Table 5-2. Symbolic Names of Commands and Objects** (page 2 of 2)

| <b>Command</b> | <b>Object Type</b>                                      | <b>Symbolic Name of Command</b> | <b>Symbolic Name of Object Type</b>   |
|----------------|---|---------------------------------|---|
| DELETE         | ATTACHMENT-SET<br>CLASS<br>EXECUTOR<br>JOB              | ZBAT-CMD-DELETE                 | ZBAT-OBJ-ATT-SET<br>ZBAT-OBJ-CLASS<br>ZBAT-OBJ-EXECUTOR<br>ZBAT-OBJ-JOB                       |
| GETVERSION     | NULL  | ZSPI-CMD-<br>GETVERSION         | ZSPI-VAL-NULL-<br>OBJECT-TYPE   |
| INFO           | ATTACHMENT-SET<br>CLASS<br>EXECUTOR<br>JOB<br>SCHEDULER | ZBAT-CMD-INFO                   | ZBAT-OBJ-ATT-SET<br>ZBAT-OBJ-CLASS<br>ZBAT-OBJ-EXECUTOR<br>ZBAT-OBJ-JOB<br>ZBAT-OBJ-SCHEDULER |
| RELEASE        | JOB   | ZBAT-CMD-<br>RELEASE            | ZBAT-OBJ-JOB  |
| RUNNEXT        | JOB   | ZBAT-CMD-<br>RUNNEXT            | ZBAT-OBJ-JOB  |
| RUNNOW         | JOB   | ZBAT-CMD-<br>RUNNOW             | ZBAT-OBJ-JOB  |
| SHUTDOWN       | SCHEDULER   | ZBAT-CMD-<br>SHUTDOWN           | ZBAT-OBJ-SCHEDULER  |
| START          | EXECUTOR<br>SCHEDULER                                   | ZBAT-CMD-START                  | ZBAT-OBJ-EXECUTOR<br>ZBAT-OBJ-SCHEDULER   |
| STATUS         | ATTACHMENT-SET<br>EXECUTOR<br>JOB<br>SCHEDULER          | ZBAT-CMD-STATUS                 | ZBAT-OBJ-ATT-SET<br>ZBAT-OBJ-EXECUTOR<br>ZBAT-OBJ-JOB<br>ZBAT-OBJ-SCHEDULER                   |
| STOP           | EXECUTOR<br>JOB   | ZBAT-CMD-STOP                   | ZBAT-OBJ-EXECUTOR<br>ZBAT-OBJ-JOB   |
| SUBMIT         | JOB   | ZBAT-CMD-SUBMIT                 | ZBAT-OBJ-JOB  |
| SUSPEND        | JOB   | ZBAT-CMD-<br>SUSPEND            | ZBAT-OBJ-JOB  |
| SWITCHCPU      | SCHEDULER   | ZBAT-CMD-<br>SWITCHCPU          | ZBAT-OBJ-SCHEDULER  |
| SWITCHLOG      | SCHEDULER   | ZBAT-CMD-<br>SWITCHLOG          | ZBAT-OBJ-SCHEDULER  |

# Sensitive and Nonsensitive Commands

A command's classification depends on its impact on the subsystem and on the security required to use it. There are two types of programmatic NetBatch command:

- Sensitive commands affect the configuration or state of objects and are usually available only to NetBatch supervisors (users with execute access to the NETBATCH program file).
- Nonsensitive commands do not affect the configuration or state of objects and are available to all users.

[Table 5-3](#) lists sensitive and nonsensitive programmatic NetBatch commands and summarizes their availability.

**Table 5-3. Sensitive and Nonsensitive Programmatic NetBatch Commands**  
(page 1 of 4)

| Command              | Sensitive | Nonsensitive | Availability   |
|----------------------|-----------|--------------|--|
| ABORT SCHEDULER      | X         |              | NetBatch supervisors only.   |
| ACTIVATE JOB         | X         |              | All users, but these conditions apply: <ul style="list-style-type: none"> <li>• NetBatch supervisors can reactivate jobs belonging to any user.</li> <li>• Non-NetBatch supervisors can reactivate any job whose input file is a disk file to which they have write access. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can reactivate the job.</li> </ul> |
| ADD ATTACHMENT-SET   | X         |              | All users.   |
| ADD CLASS            | X         |              | NetBatch supervisors only.   |
| ADD EXECUTOR         | X         |              | NetBatch supervisors only.   |
| ADD SCHEDULER        | X         |              | NetBatch supervisors only.   |
| ALTER ATTACHMENT-SET | X         |              | All users, but you must have write access to the set you want to alter.  |
| ALTER CLASS          | X         |              | NetBatch supervisors only.   |
| ALTER EXECUTOR       | X         |              | NetBatch supervisors only.   |

---

**Table 5-3. Sensitive and Nonsensitive Programmatic NetBatch Commands**  
 (page 2 of 4)

| Command               | Sensitive | Nonsensitive | Availability   |
|-----------------------|-----------|--------------|--|
| ALTER JOB             | X         |              | All users, but these conditions apply: <ul style="list-style-type: none"> <li>● You can alter all attributes of a job if the job has a disk input file to which you have write access, or the job's input file is a device or a process or does not exist and you are the job's owner.</li> <li>● NetBatch supervisors can alter all but these attributes of any job: ATTACHMENT-SET, DESCRIPTION, EXECUTOR-PROGRAM, HIGHPIN, IN, JOB-LOG, JOBID-ZERO, LIB, NAME, OUT, PURGE-IN-FILE, RUND, STARTUP, STOP-ON-ABEND, and VOLUME.</li> </ul> |
| ALTER SCHEDULER       | X         |              | NetBatch supervisors only.   |
| DELETE ATTACHMENT-SET | X         |              | All users, but you must have: <ul style="list-style-type: none"> <li>● Purge access to an attachment set to delete the set.</li> <li>● Write access to an attachment set to delete ASSIGNS, DEFINES, and PARAMs from it.</li> </ul>  |
| DELETE CLASS          | X         |              | NetBatch supervisors only.   |
| DELETE EXECUTOR       | X         |              | NetBatch supervisors only.   |
| DELETE JOB            | X         |              | All users, but these conditions apply: <ul style="list-style-type: none"> <li>● NetBatch supervisors can delete jobs belonging to any user.</li> <li>● Non-NetBatch supervisors can delete any job whose input file is a disk file to which they have write access. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can delete the job.</li> </ul>   |
| GETVERSION            |           | X            | All users.   |

---

**Table 5-3. Sensitive and Nonsensitive Programmatic NetBatch Commands**

(page 3 of 4)

| Command                  | Sensitive | Nonsensitive | Availability   |
|--------------------------|-----------|--------------|--|
| INFO<br>ATTACHMENT-SET   |           | X            | All users, but you must have read access to an attachment set to return more than the set's SECURITY and TEMPORARY attributes.   |
| INFO CLASS               |           | X            | All users.   |
| INFO EXECUTOR            |           | X            | All users.   |
| INFO JOB                 |           | X            | All users.   |
| INFO SCHEDULER           |           | X            | All users.   |
| RELEASE JOB              | X         |              | All users.   |
| RUNNEXT JOB              | X         |              | NetBatch supervisors only.   |
| RUNNOW JOB               | X         |              | NetBatch supervisors only.   |
| SHUTDOWN<br>SCHEDULER    | X         |              | NetBatch supervisors only.   |
| START EXECUTOR           | X         |              | NetBatch supervisors only.   |
| START<br>SCHEDULER       | X         |              | NetBatch supervisors only.   |
| STATUS<br>ATTACHMENT-SET |           | X            | All users.   |
| STATUS<br>EXECUTOR       |           | X            | All users.   |
| STATUS JOB               |           | X            | All users.   |
| STATUS<br>SCHEDULER      |           | X            | All users.   |
| STOP EXECUTOR            | X         |              | NetBatch supervisors only.   |
| STOP JOB                 | X         |              | All users, but these conditions apply: <ul style="list-style-type: none"> <li>● NetBatch supervisors can stop jobs belonging to any user.</li> <li>● Non-NetBatch supervisors can stop any job whose input file is a disk file to which they have write access. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can stop the job.</li> </ul> |
| SUBMIT JOB               | X         |              | All users.   |

**Table 5-3. Sensitive and Nonsensitive Programmatic NetBatch Commands**

(page 4 of 4)

| Command             | Sensitive | Nonsensitive | Availability  |
|---------------------|-----------|--------------|---|
| SUSPEND JOB         | X         |              | All users, but these conditions apply: <ul style="list-style-type: none"> <li>● NetBatch supervisors can suspend jobs belonging to any user.</li> <li>● Non-NetBatch supervisors can suspend any job whose input file is a disk file to which they have write access. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can suspend the job.</li> </ul> |
| SWITCHCPU SCHEDULER | X         |              | NetBatch supervisors only.  |
| SWITCHLOG SCHEDULER | X         |              | NetBatch supervisors only.  |

## Wild-Card Characters

[Table 5-4](#) lists the programmatic NetBatch commands that let you specify a range of objects by using the asterisk (\*) or question mark (?) wild-card characters, or both.

**Table 5-4. Commands That Support Wild-Character Matching**

|                       |                |                       |
|-----------------------|----------------|-----------------------|
| ACTIVATE JOB          | INFO EXECUTOR  | STATUS ATTACHMENT-SET |
| ALTER ATTACHMENT-SET  | INFO JOB       | STATUS EXECUTOR       |
| ALTER JOB             | RELEASE JOB    | STATUS JOB            |
| DELETE ATTACHMENT-SET | RUNNEXT JOB    | STOP EXECUTOR         |
| DELETE JOB            | RUNNOW JOB     | STOP JOB              |
| INFO ATTACHMENT-SET   | START EXECUTOR | SUSPEND JOB           |
| INFO CLASS            |                |                       |

[Table 5-5](#) lists the commands that do not support wild-card character matching.

**Table 5-5. Commands That Do Not Support Wild-Card Character Matching**

|                    |                    |                     |
|--------------------|--------------------|---------------------|
| ABORT SCHEDULER    | ALTER EXECUTOR     | START SCHEDULER     |
| ADD ATTACHMENT-SET | DELETE CLASS       | SUBMIT JOB          |
| ADD CLASS          | DELETE EXECUTOR    | SWITCHCPU SCHEDULER |
| ADD EXECUTOR       | GETVERSION         | SWITCHLOG SCHEDULER |
| ALTER CLASS        | SHUTDOWN SCHEDULER |                     |



## Characters Matched by Wild-Card Characters

Wild-card characters:

- \* Match zero or more characters. For example, A\*D matches character strings beginning with A and ending in D (such as ABCD and AD, but not CAD or ADE). You can use multiple asterisks as long as you separate them by at least one other character. For example, \*CD\* matches strings containing CD (such as ABCDEF, XYZCD, and CD21, but not BC3D or DCA). The asterisk on its own matches all strings.
- ? Match a single character. For example, ABC?? matches five-character strings beginning with ABC (such as ABCDE and ABC12, but not ABCDEF).

## Command Descriptions

Descriptions of the NetBatch programmatic commands appear in alphabetic order on these pages.

### ABORT SCHEDULER Command

The ABORT SCHEDULER command immediately stops all executing and suspended processes associated with jobs and shuts down the scheduler.

|                           |                             |
|---------------------------|-----------------------------|
| Command                   |                             |
| ZBAT-CMD-ABORT            |                             |
| Object Type               |                             |
| ZBAT-OBJ-SCHEDULER        |                             |
| Token in Command Buffer   |                             |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING. |
| Tokens in Response Buffer |                             |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.   |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.   |
| ...                       |                             |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.  |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM.   |
| !{ }                      |                             |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.  |

### Token in Command Buffer

For information on the token present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The ABORT SCHEDULER command is a sensitive command available to NetBatch supervisors only.

## ACTIVATE JOB Command

The ACTIVATE JOB command reactivates suspended processes associated with a job.

|                              |  |
|------------------------------|--|
| Command<br>ZBAT-CMD-ACTIVATE |  |
| Object Type<br>ZBAT-OBJ-JOB  |  |
| Tokens in Command Buffer     |  |
| ZBAT-TKN-SEL-JOBNAME         | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTJOBNAME      | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-JOB-NUMBER      | token-type ZBAT-TYP-JOB-NUMBER. ! {A}... |
| ZBAT-TKN-SEL-ADPNAME         | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTADPNAME      | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-CLASSNAME       | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTCLASSNAME    | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-INNAME          | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTINNAME       | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-LIST            | token-type ZBAT-TYP-LIST. ! {A}...       |
| ZBAT-TKN-SEL-NOTLIST         | token-type ZSPI-TYP-INT. ! {A}...        |
| ZBAT-TKN-SEL-USERNAME        | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTUSERNAME     | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-WAITON          | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTWAITON       | token-type ZSPI-TYP-STRING! {A}...       |
| ZSPI-TKN-COMMENT             | token-type ZSPI-TYP-STRING.              |
| ZSPI-TKN-CONTEXT             | token-type ZSPI-TYP-BYTESTRING.          |
| Tokens in Response Buffer    |  |
| ZSPI-TKN-DATALIST            | token-type ZSPI-TYP-LIST.                |
| ZBAT-TKN-SEL-JOBNAME         | token-type ZSPI-TYP-STRING. ! { }        |
| ZBAT-TKN-SEL-JOB-NUMBER      | token-type ZBAT-TYP-JOB-NUMBER. ! { }    |
| ZSPI-TKN-ERRLIST             | token-type ZSPI-TYP-LIST.                |
| ...                          |  |
| ZSPI-TKN-ENDLIST             | token-type ZSPI-TYP-SSCTL.               |
| ZSPI-TKN-RETCODE             | token-type ZSPI-TYP-ENUM. ! { }          |
| ZSPI-TKN-ENDLIST             | token-type ZSPI-TYP-SSCTL.               |
| ZSPI-TKN-CONTEXT             | token-type ZSPI-TYP-BYTESTRING.          |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ACTIVATE JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Notes

- The ACTIVATE JOB command is a sensitive command. It is available to all users, but these conditions apply:
  - You can reactivate any job belonging to any user if you are a NetBatch supervisor.
  - You can reactivate any job whose input file is a disk file to which you have write access whether or not you are a NetBatch supervisor. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can reactivate the job.
- An ACTIVATE JOB command that specifies neither ZBAT-TKN-SEL-USERNAME nor ZBAT-TKN-SEL-NOTUSERNAME acts only on jobs owned by the requesting user.

## ADD ATTACHMENT-SET Command

The ADD ATTACHMENT-SET command adds an attachment set to a scheduler.

|                            |                                      |
|----------------------------|--------------------------------------|
| Command                    |                                      |
| ZBAT-CMD-ADD               |                                      |
| Object Type                |                                      |
| ZBAT-OBJ-ATT-SET           |                                      |
| Tokens in Command Buffer   |                                      |
| ZBAT-TKN-ATT-SET-ID        | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-ATT-SET-ASSIGN    | token-type ZSPI-TYP-BYTESTRING. !... |
| ZBAT-TKN-ATT-SET-DEFINE    | token-type ZSPI-TYP-BYTESTRING. !... |
| ZBAT-TKN-ATT-SET-PARAM     | token-type ZSPI-TYPBYTESTRING. !...  |
| ZBAT-TKN-ATT-SET-SECURITY  | token-type ZSPI-TYP-INT.             |
| ZBAT-TKN-ATT-SET-TEMPORARY | token-type ZSPI-TYP-BOOLEAN.         |
| ZBAT-TKN-VOLUME-SUBVOL     | token-type ZSPI-TYP-BYTESTRING.      |
| ZSPI-TKN-COMMENT           | token-type ZSPI-TYP-STRING.          |
| Tokens in Response Buffer  |                                      |
| ZSPI-TKN-DATALIST          | token-type ZSPI-TYP-LIST.            |
| ZBAT-TKN-ATT-SET-ID        | token-type ZSPI-TYP-STRING.          |
| ZSPI-TKN-ERRLIST           | token-type ZSPI-TYP-LIST.            |
| ...                        |                                      |
| ZSPI-TKN-ENDLIST           | token-type ZSPI-TYP-SSCTL.           |
| ZSPI-TKN-RETCODE           | token-type ZSPI-TYP-ENUM. !{ }       |
| ZSPI-TKN-ENDLIST           | token-type ZSPI-TYP-SSCTL.           |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ADD ATTACHMENT-SET command is:

ZBAT-TKN-ATT-SET-ID

specifies an attachment-set ID in the form:

```
[ ( user-ID ) ] attachment-set-name
```

*user-ID*

specifies the user ID of the attachment-set owner. (*user-ID* must be in *group-name.user-name* or *group-ID,user-ID* form.) The default is the ID of the current user.

*attachment-set-name*

specifies the name of the attachment set. The name can contain 1 through 24 letters and numbers. It also can contain hyphens but must begin with a letter and end with a letter or number. The name cannot contain spaces.

Omitting ZBAT-TKN-ATT-SET-ID from the command buffer has the same effect as using #CURRENT in the BATCHCOM command ADD ATTACHMENT-SET. The scheduler automatically assigns attachment-set ownership to the current user and generates a number as the set identifier. The scheduler returns the attachment-set ID it generates in the response buffer token ZBAT-TKN-ATT-SET-ID.

ZBAT-TKN-VOLUME-SUBVOL

specifies the default node, volume, and subvolume used for qualifying unqualified file references in ZBAT-TKN-ATT-SET-DEFINE. To avoid unpredictable file-name expansion when your application omits ZBAT-TKN-VOLUME-SUBVOL, fully qualify all file names in ZBAT-TKN-ATT-SET-DEFINE.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ADD ATTACHMENT-SET command is:

ZBAT-TKN-ATT-SET-ID

is the ID of the new attachment set in the form (the ID appears as an ASCII string):

```
[ ( user-ID ) ] attachment-set-ID
```

*user-ID*

specifies the user ID of the attachment-set owner in the form  
*group-name.user-name*.

*attachment-set-ID*

is one of:

*attachment-set-name*

specifies the attachment-set name.

*attachment-set-number*

specifies the attachment-set number.

Examples of attachment-set IDs are (FPP.USER)DAILY and (SUPER.FPP)66.

## Operational Notes

- The ADD ATTACHMENT-SET command is a sensitive command available to all users.
- An ADD ATTACHMENT-SET command will fail if it specifies more ASSIGNS, DEFINES, and PARAMs than will fit in the command buffer. To add an attachment set that contains more ASSIGNS, DEFINES, and PARAMs than will fit in the command buffer, send an initial ADD ATTACHMENT-SET command followed by one or more ALTER ATTACHMENT-SET commands. Specify the TEMPORARY ON attribute in the ADD ATTACHMENT-SET command and the TEMPORARY OFF attribute in the final ALTER ATTACHMENT-SET command. The scheduler deletes the incomplete attachment set if the requester process fails before completing all ALTER ATTACHMENT-SET commands.

## ADD CLASS Command

The ADD CLASS command adds a class to a scheduler.

|                                |                                  |
|--------------------------------|----------------------------------|
| Command                        |                                  |
| ZBAT-CMD-ADD                   |                                  |
| Object Type                    |                                  |
| ZBAT-OBJ-CLASS                 |                                  |
| Tokens in Command Buffer       |                                  |
| ZBAT-TKN-SEL-CLASSNAME         | token-type ZSPI-TYP-STRING. !{ } |
| ZBAT-MAP-DEF-CLASS             |                                  |
| Definition ZBAT-DDL-DEF-CLASS. |                                  |
| 02 ZINITIATION                 | type ZSPI-DDL-BOOLEAN. !{ }      |
| End                            |                                  |
| ZSPI-TKN-COMMENT               | token-type ZSPI-TYP-STRING.      |
| Tokens in Response Buffer      |                                  |
| ZSPI-TKN-DATALIST              | token-type ZSPI-TYP-LIST.        |
| ZBAT-TKN-SEL-CLASSNAME         | token-type ZSPI-TYP-STRING.      |
| ZSPI-TKN-ERRLIST               | token-type ZSPI-TYP-LIST.        |
| ...                            |                                  |
| ZSPI-TKN-ENDLIST               | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-RETCODE               | token-type ZSPI-TYP-ENUM. !{ }   |
| ZSPI-TKN-ENDLIST               | token-type ZSPI-TYP-SSCTL.       |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ADD CLASS command is:

ZBAT-TKN-SEL-CLASSNAME

specifies a class name. The name can contain from 1 through 24 letters and numbers. It also can contain hyphens but must begin with a letter and end with a letter or number. The name cannot contain spaces.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The ADD CLASS command is a sensitive command available to NetBatch supervisors only.

## ADD EXECUTOR Command

The ADD EXECUTOR command adds an executor to a scheduler.

|                                   |                                   |
|-----------------------------------|-----------------------------------|
| Command                           |                                   |
| ZBAT-CMD-ADD                      |                                   |
| Object Type                       |                                   |
| ZBAT-OBJ-EXECUTOR                 |                                   |
| Tokens in Command Buffer          |                                   |
| ZBAT-TKN-SEL-EXECUTORNAME         | token-type ZSPI-TYP-STRING. ! { } |
| ZBAT-MAP-DEF-EXECUTOR             |                                   |
| Definition ZBAT-DDL-DEF-EXECUTOR. |                                   |
| 02 ZCPU                           | type ZSPI-DDL-INT. ! { }          |
| 02 ZJOBNUMBER                     | type ZSPI-DDL-INT.                |
| 02 ZCLASS-COUNT                   | type ZSPI-DDL-INT. ! ZCLASSES     |
| 02 ZCLASSES                       | occurs 8 times.                   |
| 03 ZCLASSNAME                     | type ZBAT-DDL-NETBATCH-NAME.      |
| 03 FILLER                         | type ZSPI-DDL-INT.                |
| 02 ZCLASS                         | type ZBAT-DDL-NETBATCH-NAME.      |
| End                               |                                   |
| ZSPI-TKN-COMMENT                  | token-type ZSPI-TYP-STRING.       |
| Tokens in Response Buffer         |                                   |
| ZSPI-TKN-DATALIST                 | token-type ZSPI-TYP-LIST.         |
| ZBAT-TKN-SEL-EXECUTORNAME         | token-type ZSPI-TYP-STRING.       |
| ZSPI-TKN-ERRLIST                  | token-type ZSPI-TYP-LIST.         |
| ...                               |                                   |
| ZSPI-TKN-ENDLIST                  | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-RETCODE                  | token-type ZSPI-TYP-ENUM. ! { }   |
| ZSPI-TKN-ENDLIST                  | token-type ZSPI-TYP-SSCTL.        |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ADD EXECUTOR command is:

ZBAT-TKN-SEL-EXECUTORNAME

specifies an executor name. The name can contain from 1 through 24 letters and numbers. It also can contain hyphens but must begin with a letter and end with a letter or number. The name cannot contain spaces.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The ADD EXECUTOR command is a sensitive command available to NetBatch supervisors only.

## ADD SCHEDULER Command

The ADD SCHEDULER command purges scheduler files (except BATCHCTL and log files) from the scheduler's database, and creates and initializes a new database. For information on when to use the command, see [Starting a Scheduler](#) on page 2-2.

```

Command
  ZBAT-CMD-ADD

Object Type
  ZBAT-OBJ-SCHEDULER

Tokens in Command Buffer
  ZBAT-MAP-DEF-SCHEDULER
    Definition ZBAT-DDL-DEF-SCHEDULER.
      02 ZBACKUPCPU2                type ZSPI-DDL-INT.
      02 ZBACKUPCPU1                type ZSPI-DDL-INT.
      02 ZMAXCONCURRENTJOBS         type ZSPI-DDL-INT.
      02 ZMAXTEMPEXECUTORS          type ZSPI-DDL-INT.
      02 ZTAPEDRIVES                type ZSPI-DDL-INT.
      02 ZMAXPRI                    type ZSPI-DDL-INT.
      02 ZINFO-TAPEDRIVES-IN-USE    type ZSPI-DDL-INT.
      02 ZAT-ALLOWED               type ZSPI-DDL-BOOLEAN.
      02 ZSUBMIT-ALLOWED           type ZSPI-DDL-BOOLEAN.
      02 ZEVEY-CATCHUP             type ZSPI-DDL-BOOLEAN.
      02 ZEMS                      type ZBAT-DDL-EMS.
      02 ZCLASSNAME                type ZBAT-DDL-NETBATCH-NAME.
      02 ZPRI                      type ZSPI-DDL-INT.
      02 ZSELPRI                   type ZSPI-DDL-INT.
      02 ZMAXPRINTLINES            type ZSPI-DDL-INT2.
      02 ZMAXPRINTPAGES            type ZSPI-DDL-INT2.
      02 ZSTOP-ON-ABEND            type ZSPI-DDL-BOOLEAN.
      02 ZSTALL                    type ZSPI-DDL-BOOLEAN.
      02 ZHIGHPIN                  type ZSPI-DDL-BOOLEAN.
      02 ZINITIATION               type ZSPI-DDL-BOOLEAN.
      02 ZLOCALNAMES               type ZSPI-DDL-CHAR8
      occurs 30 times.

    End

  ZBAT-TKN-EXECUTOR-PROGRAM        token-type ZSPI-TYP-STRING.
  ZBAT-TKN-OUT-FILE                token-type ZSPI-TYP-BYTESTRING.
  ZSPI-TKN-COMMENT                 token-type ZSPI-TYP-STRING.

Tokens in Response Buffer
  ZSPI-TKN-DATALIST                token-type ZSPI-TYP-LIST.
  ZSPI-TKN-ERRLIST                 token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST                 token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE                 token-type ZSPI-TYP-ENUM. ! { }
  ZSPI-TKN-ENDLIST                 token-type ZSPI-TYP-SSCTL.

```

## Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ADD SCHEDULER command is:

ZBAT-TKN-EXECUTOR-PROGRAM

is the name of a program file and specifies the scheduler's DEFAULT-EXECUTOR-PROGRAM attribute. The default is \$SYSTEM.SYSTEM.TACL.



ZBAT-TKN-OUT-FILE

is the name of an output file and specifies the scheduler's DEFAULT-OUT attribute. The default is \$\$.#BATCH.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The ADD SCHEDULER command is a sensitive command available to NetBatch supervisors only.

## ALTER ATTACHMENT-SET Command

The ALTER ATTACHMENT-SET command changes attachment-set attributes.

|                            |                                  |
|----------------------------|----------------------------------|
| Command                    |                                  |
| ZBAT-CMD-ALTER             |                                  |
| Object Type                |                                  |
| ZBAT-OBJ-ATT-SET           |                                  |
| Tokens in Command Buffer   |                                  |
| ZBAT-TKN-ATT-SET-ID        | token-type ZSPI-TYP-STRING. !{ } |
| ZBAT-TKN-ATT-SET-ASSIGN    | token-type ZSPI-TYP-BYTESTRING.  |
| !...                       |                                  |
| ZBAT-TKN-ATT-SET-DEFINE    | token-type ZSPI-TYP-BYTESTRING.  |
| !...                       |                                  |
| ZBAT-TKN-ATT-SET-PARAM     | token-type ZSPI-TYP-BYTESTRING.  |
| !...                       |                                  |
| ZBAT-TKN-ATT-SET-SECURITY  | token-type ZSPI-TYP-INT.         |
| ZBAT-TKN-ATT-SET-TEMPORARY | token-type ZSPI-TYP-BOOLEAN.     |
| ZBAT-TKN-VOLUME-SUBVOL     | token-type ZSPI-TYP-BYTESTRING.  |
| ZSPI-TKN-COMMENT           | token-type ZSPI-TYP-STRING.      |
| ZSPI-TKN-CONTEXT           | token-type ZSPI-TYP-BYTESTRING.  |
| Tokens in Response Buffer  |                                  |
| ZSPI-TKN-DATALIST          | token-type ZSPI-TYP-LIST.        |
| ZBAT-TKN-ATT-SET-ID        | token-type ZSPI-TYP-STRING. !{ } |
| ZSPI-TKN-ERRLIST           | token-type ZSPI-TYP-LIST.        |
| ...                        |                                  |
| ZSPI-TKN-ENDLIST           | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-RETCODE           | token-type ZSPI-TYP-ENUM. !{ }   |
| ZSPI-TKN-ENDLIST           | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-CONTEXT           | token-type ZSPI-TYP-BYTESTRING.  |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ALTER ATTACHMENT-SET command is:

ZBAT-TKN-ATT-SET-ID

specifies an attachment set or a range of attachment sets in the form:

[ ( *user-ID* ) ] *attachment-set-ID*

*user-ID*

specifies a user ID or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of user IDs. (*user-ID* must be in *group-name.user-name* or *group-ID,user-ID* form.) The default is the user ID of the current user.

*attachment-set-ID*

is one of:

*attachment-set-name*

specifies the name of an attachment set owned by *user-ID* or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of names of attachment sets owned by *user-ID*.

*attachment-set-number*

specifies the number of an attachment set owned by *user-ID*.

\*

specifies all attachment sets owned by *user-ID*.

ZBAT-TKN-VOLUME-SUBVOL

specifies the default node, volume, and subvolume used for qualifying unqualified file references in ZBAT-TKN-ATT-SET-DEFINE. To avoid unpredictable file-name expansion when your application omits ZBAT-TKN-VOLUME-SUBVOL, fully qualify all file names in ZBAT-TKN-ATT-SET-DEFINE.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The ALTER ATTACHMENT-SET command is a sensitive command available to all users who have write access to the sets they want to alter.

## ALTER CLASS Command

The ALTER CLASS command changes class attributes.

|                                |                                   |
|--------------------------------|-----------------------------------|
| Command                        |                                   |
| ZBAT-CMD-ALTER                 |                                   |
| Object Type                    |                                   |
| ZBAT-OBJ-CLASS                 |                                   |
| Tokens in Command Buffer       |                                   |
| ZBAT-TKN-SEL-CLASSNAME         | token-type ZSPI-TYP-STRING. ! { } |
| ZBAT-MAP-DEF-CLASS             |                                   |
| Definition ZBAT-DDL-DEF-CLASS. |                                   |
| 02 ZINITIATION                 | type ZSPI-DDL-BOOLEAN. ! { }      |
| End                            |                                   |
| ZSPI-TKN-COMMENT               | token-type ZSPI-TYP-STRING.       |
| Tokens in Response Buffer      |                                   |
| ZSPI-TKN-DATALIST              | token-type ZSPI-TYP-LIST.         |
| ZBAT-TKN-SEL-CLASSNAME         | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-ERRRLIST              | token-type ZSPI-TYP-LIST.         |
| ...                            |                                   |
| ZSPI-TKN-ENDLIST               | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-RETCODE               | token-type ZSPI-TYP-ENUM. ! { }   |
| ZSPI-TKN-ENDLIST               | token-type ZSPI-TYP-SSCTL.        |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ALTER CLASS command is:

ZBAT-TKN-SEL-CLASSNAME  
specifies the name of a class.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Notes

- The ALTER CLASS command is a sensitive command available to NetBatch supervisors only.
- The command does not support wild-card character matching of class names.

## ALTER EXECUTOR Command

The ALTER EXECUTOR command changes executor attributes.

|                                   |                                  |
|-----------------------------------|----------------------------------|
| Command                           |                                  |
| ZBAT-CMD-ALTER                    |                                  |
| Object Type                       |                                  |
| ZBAT-OBJ-EXECUTOR                 |                                  |
| Tokens in Command Buffer          |                                  |
| ZBAT-TKN-SEL-EXECUTORNAME         | token-type ZSPI-TYP-STRING. !{ } |
| ZBAT-MAP-DEF-EXECUTOR             |                                  |
| Definition ZBAT-DDL-DEF-EXECUTOR. |                                  |
| 02 ZCPU                           | type ZSPI-DDL-INT.               |
| 02 ZJOBNUMBER                     | type ZSPI-DDL-INT.               |
| 02 ZCLASS-COUNT                   | type ZSPI-DDL-INT. !ZCLASSES     |
| 02 ZCLASSES                       | occurs 8 times.                  |
| 03 ZCLASSNAME                     | type ZBAT-DDL-NETBATCH-NAME.     |
| 03 FILLER                         | type ZSPI-DDL-INT.               |
| 02 ZCLASS                         | type ZBAT-DDL-NETBATCH-NAME.     |
| End                               |                                  |
| ZSPI-TKN-COMMENT                  | token-type ZSPI-TYP-STRING.      |
| Tokens in Response Buffer         |                                  |
| ZSPI-TKN-DATALIST                 | token-type ZSPI-TYP-LIST.        |
| ZBAT-TKN-SEL-EXECUTORNAME         | token-type ZSPI-TYP-STRING. !{ } |
| ZSPI-TKN-ERRLIST                  | token-type ZSPI-TYP-LIST.        |
| ...                               |                                  |
| ZSPI-TKN-ENDLIST                  | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-RETCODE                  | token-type ZSPI-TYP-ENUM. !{ }   |
| ZSPI-TKN-ENDLIST                  | token-type ZSPI-TYP-SSCTL.       |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ALTER EXECUTOR command is:

ZBAT-TKN-SEL-EXECUTORNAME

specifies the name of an executor.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Notes

- The ALTER EXECUTOR command is a sensitive command available to NetBatch supervisors only.
- The command does not support wild-card character matching of executor names.

## ALTER JOB Command

The ALTER JOB command changes job attributes.

```

Command
  ZBAT-CMD-ALTER

Object Type
  ZBAT-OBJ-JOB

Tokens in Command Buffer
  ZBAT-TKN-SEL-JOBNAME          token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-NOTJOBNAME       token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-JOB-NUMBER       token-type ZBAT-TYP-JOB-NUMBER ! {A} ...
  ZBAT-TKN-SEL-ADPNAME          token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-NOTADPNAME       token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-CLASSNAME        token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-NOTCLASSNAME     token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-INNAME           token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-NOTINNAME        token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-LIST             token-type ZBAT-TYP-LIST. ! {A} ...
  ZBAT-TKN-SEL-NOTLIST          token-type ZSPI-TYP-INT. ! {A} ...
  ZBAT-TKN-SEL-USERNAME         token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-NOTUSERNAME      token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-WAITON           token-type ZSPI-TYP-STRING. ! {A} ...
  ZBAT-TKN-SEL-NOTWAITON        token-type ZSPI-TYP-STRING ! {A} ...

ZBAT-MAP-DEF-JOB
  Definition ZBAT-DDL-DEF-JOB.
    02 ZCLASSNAME                type ZBAT-DDL-NETBATCH-NAME.
    02 ZHOLD                     type ZSPI-DDL-BOOLEAN.
    02 ZHOLD-AFTER               type ZSPI-DDL-BOOLEAN.
    02 ZRESTART                  type ZSPI-DDL-BOOLEAN.
    02 ZSTOP-ON-ABEND            type ZSPI-DDL-BOOLEAN.
    02 ZAT-FLAG                  type ZSPI-DDL-BOOLEAN.
    02 ZIFFAILS                  type ZSPI-DDL-BOOLEAN.
    02 ZPURGE-IN-FILE            type ZSPI-DDL-BOOLEAN.
    02 ZSTALL                    type ZSPI-DDL-BOOLEAN.
    02 ZINFO-NEXT-RUNTIME        type ZSPI-DDL-INT4.
    02 ZINFO-OUT-SPOOL-NUM       type ZSPI-DDL-INT.
    02 ZINFO-WHICH-LIST          type ZBAT-DDL-JOB-WHICH-LIST.
    02 ZINFO-SPECIAL-REASON      type ZBAT-DDL-SPECIAL-REASON.
    02 ZINFO-TOTAL-CPU-TIME      type ZSPI-DDL-INT4.
    02 ZINFO-OPEN-ACCESSOR       type ZSPI-DDL-INT.
    02 ZRE MID                   type ZSPI-DDL-BOOLEAN.
    02 ZEVERY-DAYS              type ZSPI-DDL-INT.
02 ZEVERY-HOURS                 type ZSPI-DDL-INT.
    02 ZEVERY-MINUTES            type ZSPI-DDL-INT.
    02 ZDEFAULT-SECURITY         type ZSPI-DDL-INT.
    02 ZPRI                     type ZSPI-DDL-INT.
    02 ZSELPRI                  type ZSPI-DDL-INT.
    02 ZHIGHPIN                 type ZSPI-DDL-BOOLEAN.
    02 ZMAXPRINTLINES           type ZSPI-DDL-INT2.
    02 ZMAXPRINTPAGES           type ZSPI-DDL-INT2.
    02 ZTAPEDRIVES              type ZSPI-DDL-INT.
    02 ZDATE.
      03 ZYEAR                  type ZSPI-DDL-INT.
      03 ZMONTH                 type ZSPI-DDL-INT.
      03 ZDAY                   type ZSPI-DDL-INT.
    02 ZTIME.
      03 ZHOUR                  type ZSPI-DDL-INT.
      03 ZMINUTE                type ZSPI-DDL-INT.
      03 ZSECOND                type ZSPI-DDL-INT.
      03 ZMILLISECOND           type ZSPI-DDL-INT.
      03 ZMICROSECOND           type ZSPI-DDL-INT.
(continued)

```

```

02 ZPOSIX                                type ZSPI-DDL-INT.
02 ZSAVEABEND                            type ZSPI-DDL-BOOLEAN.
02 ZRUND                                 type ZSPI-DDL-BOOLEAN.
02 ZJOBID-ZERO                           type ZSPI-DDL-BOOLEAN.
02 ZMEM                                  type ZSPI-DDL-INT.
02 ZPFS                                  type ZSPI-DDL-INT2.
02 ZNAM                                  type ZSPI-DDL-CHAR8.
02 ZINFO-TIME-SUBMIT                     type ZSPI-DDL-INT4.
02 ZINFO-LAST-MOD                        type ZSPI-DDL-INT4.
02 ZINFO-LAST-MODUSER                    type ZSPI-DDL-INT.
02 ZTIME-LIMIT                           type ZSPI-DDL-INT2.
End

ZBAT-MAP-DEF-WAITON
  Definition ZBAT-DDL-DEF-WAITON.
    02 ZMASTER                           type ZBAT-DDL-NETBATCH-NAME.
    02 ZINDICATOR                         type ZBAT-DDL-WAITON-INDICATOR.
    02 ZFOR                               type ZBAT-DDL-WAITON-FOR.
  End

ZBAT-TKN-ATT-SET-ID                      token-type ZSPI-TYP-STRING.
ZBAT-TKN-CALENDAR                        token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-DESCRIPTION                     token-type ZSPI-TYP-STRING.
ZBAT-TKN-EXECUTOR-PROGRAM                token-type ZSPI-TYP-STRING.
ZBAT-TKN-EXTSWAP-FILE                    token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-IN-FILE                         token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-LIB-FILE                        token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-OUT-FILE                        token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-STARTUP-MESSAGE                 token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-SWAP-FILE                       token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-TERM-FILE                       token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-VOLUME-SUBVOL                   token-type ZSPI-TYP-BYTESTRING.

ZBAT-MAP-DEF-CRONTAB
  Definition ZBAT-DDL-DEF-CRONTAB.
    02 ZMINUTES                           type ZSPI-DDL-INT4.
    02 ZHOURS                             type ZSPI-DDL-INT2.
    02 ZDAYS                              type ZSPI-DDL-INT2.
    02 ZMONTHS                            type ZSPI-DDL-INT.
    02 ZWEEKDAYS                          type ZSPI-DDL-INT.
  End

ZSPI-TKN-COMMENT                         token-type ZSPI-TYP-STRING.
ZSPI-TKN-CONTEXT                         token-type ZSPI-TYP-BYTESTRING.

Tokens in Response Buffer
ZSPI-TKN-DATALIST                        token-type ZSPI-TYP-LIST.
  ZBAT-TKN-SEL-JOBNAME                   token-type ZSPI-TYP-STRING.
  ZBAT-TKN-SEL-JOB-NUMBER                 token-type ZBAT-TYP-JOB-NUMBER.
  ZSPI-TKN-ERRLIST                       token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST                       token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE                       token-type ZSPI-TYP-ENUM. ! { }
ZSPI-TKN-ENDLIST                         token-type ZSPI-TYP-SSCTL.
ZSPI-TKN-CONTEXT                         token-type ZSPI-TYP-BYTESTRING.

```

## Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ALTER JOB command is:

**ZBAT-TKN-SEL-JOBNAME**

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

**ZBAT-TKN-ATT-SET-ID**

specifies a value for the job's ATTACHMENT-SET attribute in the form:

```
[ ( user-ID ) ] attachment-set-ID
```

*user-ID*

specifies the user ID of the attachment-set owner. (*user-ID* must be in *group-name.user-name* or *group-ID,user-ID* form.) The default is the user ID of the current user.

*attachment-set-ID*

is one of:

*attachment-set-name*

specifies an attachment-set name.

*attachment-set-number*

specifies an attachment-set number.

**ZBAT-TKN-EXECUTOR-PROGRAM**

is the name of a program file and specifies the job's EXECUTOR-PROGRAM attribute.

**ZBAT-TKN-OUT-FILE**

is the name of an output file and specifies the job's OUT attribute.

**ZBAT-TKN-VOLUME-SUBVOL**

contains the default node, volume, and subvolume used for qualifying unqualified file references in the job's input file. The token must specify, at a minimum, a default volume and subvolume. If the node name is not specified, the scheduler uses the node of the requester.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).



## Operational Notes

- The ALTER JOB command is a sensitive command. It is available to all users, but these conditions apply:
  - You can alter all attributes of a job if the job has a disk input file to which you have write access. You can also alter all attributes of a job if the job's input file is a device or a process or does not exist, but you can do this only if you are the job's owner.
  - A NetBatch supervisor can alter all except these attributes of any job:
 

|                  |               |         |
|------------------|---------------|---------|
| ATTACHMENT-SET   | OUT           | STARTUP |
| EXECUTOR-PROGRAM | PURGE-IN-FILE | VOLUME  |
| IN               |               |         |

- An ALTER JOB command that specifies neither ZBAT-TKN-SEL-USERNAME nor ZBAT-TKN-SEL-NOTUSERNAME acts only on jobs owned by the requesting user.
- To remove a master job from the list of masters specified by a dependent job's WAITON attribute, execute an ALTER JOB command with the ZINDICATOR field of ZBAT-MAP-DEF-WAITON set to ZBAT-VAL-WAITON-REMOVE.

To remove a job's WAITON attribute, execute an ALTER JOB command with the ZMASTER field of ZBAT-MAP-DEF-WAITON set to spaces or a null value.

- Attachment sets specified by an ALTER JOB command overwrite all existing attachment sets of the job.

The order in which you specify a job's attachment sets is the order in which the scheduler supplies them to the job. For example, specifying sets C, B, and A in that order makes the scheduler process set C first, set B second, and set A third.

If the name of an ASSIGN, DEFINE, or PARAM from a set conflicts with a name from a set specified earlier, the scheduler overwrites the earlier ASSIGN, DEFINE, or PARAM with the details of the later ASSIGN, DEFINE, or PARAM.

To remove a job's ATTACHMENT-SET attribute, execute an ALTER JOB command with ZBAT-TKN-ATT-SET-ID set to spaces or a null value.

- These fields of ZBAT-MAP-DEF-JOB are returned in that token by the INFO JOB command. The scheduler ignores them in the ALTER JOB command.
 

|                      |                      |
|----------------------|----------------------|
| ZINFO-NEXT-RUNTIME   | ZINFO-TOTAL-CPU-TIME |
| ZINFO-OPEN-ACCESSOR  | ZINFO-WHICH-LIS      |
| ZINFO-OUT-SPOOL-NUM  | ZREMID               |
| ZINFO-SPECIAL-REASON |                      |
- ZBAT-MAP-DEF-CRONTAB, ZBAT-TKN-CALENDAR, and the ZEVERY-DAYS, ZEVERY-HOURS, and ZEVERY-MINUTES fields of ZBAT-MAP-DEF-JOB are mutually exclusive.

## ALTER SCHEDULER Command

The ALTER SCHEDULER command changes scheduler attributes.

```

Command
  ZBAT-CMD-ALTER

Object Type
  ZBAT-OBJ-SCHEDULER

Tokens in Command Buffer
  ZBAT-MAP-DEF-SCHEDULER
    Definition ZBAT-DDL-DEF-SCHEDULER.
      02 ZBACKUPCPU2                                type ZSPI-DDL-INT.
      02 ZBACKUPCPU1                                type ZSPI-DDL-INT.
      02 ZMAXCONCURRENTJOBS                          type ZSPI-DDL-INT.
      02 ZMAXTEMPEXECUTORS                          type ZSPI-DDL-INT.
      02 ZTAPEDRIVES                                type ZSPI-DDL-INT.
      02 ZMAXPRI                                      type ZSPI-DDL-INT.
      02 ZINFO-TAPEDRIVES-IN-USE                     type ZSPI-DDL-INT.
      02 ZAT-ALLOWED                                type ZSPI-DDL-BOOLEAN.
      02 ZSUBMIT-ALLOWED                            type ZSPI-DDL-BOOLEAN.
      02 ZEVERY-CATCHUP                             type ZSPI-DDL-BOOLEAN.
      02 ZEMS                                          type ZBAT-DDL-EMS.
      02 ZCLASSNAME                                  type ZBAT-DDL-NETBATCH-NAME.
      02 ZPRI                                          type ZSPI-DDL-INT.
      02 ZSELPRI                                      type ZSPI-DDL-INT.
      02 ZMAXPRINTLINES                             type ZSPI-DDL-INT2.
      02 ZMAXPRINTPAGES                             type ZSPI-DDL-INT2.
      02 ZSTOP-ON-ABEND                             type ZSPI-DDL-BOOLEAN.
      02 ZSTALL                                       type ZSPI-DDL-BOOLEAN.
      02 ZHIGHPIN                                    type ZSPI-DDL-BOOLEAN.
      02 ZINITIATION                                type ZSPI-DDL-BOOLEAN.
      02 ZLOCALNAMES                                type ZSPI-DDL-CHAR8
                                                    occurs 30 times.

    End

  ZBAT-TKN-EXECUTOR-PROGRAM                        token-type ZSPI-TYP-STRING.
  ZBAT-TKN-OUT-FILE                                token-type ZSPI-TYP-BYTESTRING.
  ZSPI-TKN-COMMENT                                token-type ZSPI-TYP-STRING.

Tokens in Response Buffer
  ZSPI-TKN-DATALIST                                token-type ZSPI-TYP-LIST.
  ZSPI-TKN-ERRLIST                                token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST                                token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE                                token-type ZSPI-TYP-ENUM. ! { }
  ZSPI-TKN-ENDLIST                                token-type ZSPI-TYP-SSCTL.

```

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the ALTER SCHEDULER command is:

ZBAT-TKN-EXECUTOR-PROGRAM

is the name of a program file and specifies the scheduler's DEFAULT-EXECUTOR-PROGRAM attribute.

ZBAT-TKN-OUT-FILE

is the name of an output file and specifies the scheduler's DEFAULT-OUT attribute.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The ALTER SCHEDULER command is a sensitive command available to NetBatch supervisors only.

## DELETE ATTACHMENT-SET Command

The DELETE ATTACHMENT-SET command deletes attachment sets from a scheduler. The command also deletes specified ASSIGNS, DEFINES (except =\_DEFAULTS), and PARAMs from attachment sets.

|                           |                                  |
|---------------------------|----------------------------------|
| Command                   |                                  |
| ZBAT-CMD-DELETE           |                                  |
| Object Type               |                                  |
| ZBAT-OBJ-ATT-SET          |                                  |
| Tokens in Command Buffer  |                                  |
| ZBAT-TKN-ATT-SET-ID       | token-type ZSPI-TYP-STRING.!!{ } |
| ZBAT-TKN-SEL-ASSIGN-NAME  | token-type ZSPI-TYP-STRING.!!... |
| ZBAT-TKN-SEL-DEFINE-NAME  | token-type ZSPI-TYP-STRING.!!... |
| ZBAT-TKN-SEL-PARAM-NAME   | token-type ZSPI-TYP-STRING.!!... |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.      |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.  |
| Tokens in Response Buffer |                                  |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.        |
| ZBAT-TKN-ATT-SET-ID       | token-type ZSPI-TYP-STRING.!!{ } |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.        |
| ...                       |                                  |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM.!!{ }   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.  |

## Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the DELETE ATTACHMENT-SET command is:

ZBAT-TKN-ATT-SET-ID

specifies an attachment set or a range of attachment sets in the form:

[ ( *user-ID* ) ] *attachment-set-ID*

*user-ID*

specifies a user ID or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of user IDs. (*user-ID* must be in *group-name.user-name* or *group-ID,user-ID* form.) The default is the user ID of the current user.

*attachment-set-ID*

is one of:

*attachment-set-name*

specifies the name of an attachment set owned by *user-ID* or, when specified with either or both the asterisk (\*) and question mark (?) wild-

card characters, a range of names of attachment sets owned by *user-ID*.

*attachment-set-number*

specifies the number of an attachment set owned by *user-ID*.

\*

specifies all attachment sets owned by *user-ID*.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The DELETE ATTACHMENT-SET command is a sensitive command. It is available to all users, but these conditions apply:

- You must have purge access to an attachment set to delete the set.
- You must have write access to an attachment set to delete ASSIGNS, DEFINES, and PARAMs from it.

## DELETE CLASS Command

The DELETE CLASS command deletes classes from a scheduler.

|                           |                                   |
|---------------------------|-----------------------------------|
| Command                   |                                   |
| ZBAT-CMD-DELETE           |                                   |
| Object Type               |                                   |
| ZBAT-OBJ-CLASS            |                                   |
| Tokens in Command Buffer  |                                   |
| ZBAT-TKN-SEL-CLASSNAME    | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.       |
| Tokens in Response Buffer |                                   |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.         |
| ZBAT-TKN-SEL-CLASSNAME    | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.         |
| ...                       |                                   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.        |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the DELETE CLASS command is:

ZBAT-TKN-SEL-CLASSNAME

specifies the name of a class.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Notes

- The DELETE CLASS command is a sensitive command available to NetBatch supervisors only.
- The command does not support wild-card character matching of class names.

## DELETE EXECUTOR Command

The DELETE EXECUTOR command deletes executors from a scheduler.

|                           |                                   |
|---------------------------|-----------------------------------|
| Command                   |                                   |
| ZBAT-CMD-DELETE           |                                   |
| Object Type               |                                   |
| ZBAT-OBJ-EXECUTOR         |                                   |
| Tokens in Command Buffer  |                                   |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.       |
| Tokens in Response Buffer |                                   |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.         |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.         |
| ...                       |                                   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.        |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the DELETE EXECUTOR command is:

ZBAT-TKN-SEL-EXECUTORNAME  
specifies the name of an executor.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Notes

- The DELETE EXECUTOR command is a sensitive command available to NetBatch supervisors only.
- The command does not support wild-card character matching of executor names.

## DELETE JOB Command

The DELETE JOB command deletes from a scheduler a job that is not executing, over limit, or suspended.

|                           |   |
|---------------------------|---|
| Command                   |   |
| ZBAT-CMD-DELETE           |   |
| Object Type               |   |
| ZBAT-OBJ-JOB              |   |
| Tokens in Command Buffer  |   |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-NOTJOBNAME   | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER. ! {A} ... |
| ZBAT-TKN-SEL-ADPNAME      | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-NOTADPNAME   | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-CLASSNAME    | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-NOTCLASSNAME | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-INNAME       | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-NOTINNAME    | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-LIST         | token-type ZBAT-TYP-LIST. ! {A} ...       |
| ZBAT-TKN-SEL-NOTLIST      | token-type ZSPI-TYP-INT. ! {A} ...        |
| ZBAT-TKN-SEL-USERNAME     | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-NOTUSERNAME  | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-WAITON       | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZBAT-TKN-SEL-NOTWAITON    | token-type ZSPI-TYP-STRING. ! {A} ...     |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.               |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.           |
| Tokens in Response Buffer |   |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.                 |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING.               |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.           |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.                 |
| ...                       |   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.                |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }           |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.                |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.           |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the DELETE JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).



## Operational Note

The DELETE JOB command is a sensitive command. It is available to all users, but these conditions apply:

- You can delete jobs belonging to any user if you are a NetBatch supervisor.
- You can delete any job whose input file is a disk file to which you have write access whether or not you are a NetBatch supervisor. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can delete the job.

## GETVERSION Command

The GETVERSION command returns the NetBatch server version in token ZSPI-TKN-SERVER-VERSION in the SPI header and the server ID in token ZSPI-TKN-SERVER-BANNER.

|                           |                                 |
|---------------------------|---------------------------------|
| Command                   |                                 |
| ZSPI-CMD-GETVERSION       |                                 |
| Object Type               |                                 |
| ZSPI-VAL-NULL-OBJECT-TYPE |                                 |
| Tokens in Command Buffer  |                                 |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.     |
| Tokens in Response Buffer |                                 |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.       |
| ZSPI-TKN-SERVER-BANNER    | token-type ZSPI-TYP-CHAR50.     |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.       |
| ...                       |                                 |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.      |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { } |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.      |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#). Information about the tokens specific to the GETVERSION command is:

ZSPI-TKN-SERVER-BANNER

is a 50-character string that contains the standard version ID of the NetBatch subsystem, the NetBatch release date, and the NetBatch compilation date.

### Operational Note

The GETVERSION command is a nonsensitive command available to all users.

## INFO ATTACHMENT-SET Command

The INFO ATTACHMENT-SET command lists the attributes of attachment sets.

|                            |                                       |
|----------------------------|---------------------------------------|
| Command                    |                                       |
| ZBAT-CMD-INFO              |                                       |
| Object Type                |                                       |
| ZBAT-OBJ-ATT-SET           |                                       |
| Tokens in Command Buffer   |                                       |
| ZBAT-TKN-ATT-SET-ID        | token-type ZSPI-TYP-STRING. ! { }     |
| ZBAT-TKN-SEL-ASSIGN-NAME   | token-type ZSPI-TYP-STRING. ! ...     |
| ZBAT-TKN-SEL-DEFINE-NAME   | token-type ZSPI-TYP-STRING. ! ...     |
| ZBAT-TKN-SEL-PARAM-NAME    | token-type ZSPI-TYP-STRING. ! ...     |
| ZSPI-TKN-COMMENT           | token-type ZSPI-TYP-STRING.           |
| ZSPI-TKN-CONTEXT           | token-type ZSPI-TYP-BYTESTRING.       |
| Tokens in Response Buffer  |                                       |
| ZSPI-TKN-DATALIST          | token-type ZSPI-TYP-LIST.             |
| ZBAT-TKN-ATT-SET-ID        | token-type ZSPI-TYP-STRING. ! { }     |
| ZBAT-TKN-ATT-SET-ASSIGN    | token-type ZSPI-TYP-BYTESTRING. ! ... |
| ZBAT-TKN-ATT-SET-DEFINE    | token-type ZSPI-TYP-BYTESTRING. ! ... |
| ZBAT-TKN-ATT-SET-PARAM     | token-type ZSPI-TYP-BYTESTRING. ! ... |
| ZBAT-TKN-ATT-SET-SECURITY  | token-type ZSPI-TYP-INT.              |
| ZBAT-TKN-ATT-SET-TEMPORARY | token-type ZSPI-TYP-BOOLEAN.          |
| ZSPI-TKN-ERRLIST           | token-type ZSPI-TYP-LIST.             |
| ...                        |                                       |
| ZSPI-TKN-ENDLIST           | token-type ZSPI-TYP-SSCTL.            |
| ZSPI-TKN-RETCODE           | token-type ZSPI-TYP-ENUM. ! { }       |
| ZSPI-TKN-ENDLIST           | token-type ZSPI-TYP-SSCTL.            |
| ZSPI-TKN-CONTEXT           | token-type ZSPI-TYP-BYTESTRING.       |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the INFO ATTACHMENT-SET command is:

ZBAT-TKN-ATT-SET-ID

specifies an attachment set or a range of attachment sets in the form:

[ ( *user-ID* ) ] *attachment-set-ID*

*user-ID*

specifies a user ID or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of user IDs. (*user-ID* must be in *group-name.user-name* or *group-ID,user-ID* form.) The default is the user ID of the current user.

*attachment-set-ID*

is one of these:

*attachment-set-name*

specifies the name of an attachment set owned by *user-ID* or, when specified with either or both the asterisk (\*) and question mark (?) wildcard characters, a range of names of attachment sets owned by *user-ID*.

*attachment-set-number*

specifies the number of an attachment set owned by *user-ID*.

\*

specifies all attachment sets owned by *user-ID*.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Notes

- The INFO ATTACHMENT-SET command is a nonsensitive command available to all users. You must have read access to an attachment set to return more than the set's SECURITY and TEMPORARY attributes.
- The scheduler lists the SECURITY and TEMPORARY attributes for all attachment sets it returns, regardless of the sets' security.

## INFO CLASS Command

The INFO CLASS command lists the attributes of classes.

```

Command
  ZBAT-CMD-INFO

Object Type
  ZBAT-OBJ-CLASS

Tokens in Command Buffer
  ZBAT-TKN-SEL-CLASSNAME          token-type ZSPI-TYP-STRING. ! { }
  ZSPI-TKN-COMMENT                token-type ZSPI-TYP-STRING.
  ZSPI-TKN-CONTEXT                token-type ZSPI-TYP-BYTESTRING.

Tokens in Response Buffer
  ZSPI-TKN-DATALIST              token-type ZSPI-TYP-LIST.
  ZBAT-TKN-SEL-CLASSNAME         token-type ZSPI-TYP-STRING. ! { }

  ZBAT-MAP-DEF-CLASS
    Definition ZBAT-DDL-DEF-CLASS.
      02 ZINITIATION              type ZSPI-DDL-BOOLEAN.
    End

  ZSPI-TKN-ERRLIST               token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST               token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE               token-type ZSPI-TYP-ENUM. ! { }
  ZSPI-TKN-ENDLIST               token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-CONTEXT               token-type ZSPI-TYP-BYTESTRING.

```

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the INFO CLASS command is:

ZBAT-TKN-SEL-CLASSNAME

specifies a class name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of class names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The INFO CLASS command is a nonsensitive command available to all users.

## INFO EXECUTOR Command

The INFO EXECUTOR command lists the attributes of executors.

```

Command
  ZBAT-CMD-INFO

Object Type
  ZBAT-OBJ-EXECUTOR

Tokens in Command Buffer
  ZBAT-TKN-SEL-EXECUTORNAME      token-type ZSPI-TYP-STRING. !{}
  ZSPI-TKN-COMMENT               token-type ZSPI-TYP-STRING.
  ZSPI-TKN-CONTEXT               token-type ZSPI-TYP-
  BYTESTRING.

Tokens in Response Buffer
  ZSPI-TKN-DATALIST              token-type ZSPI-TYP-LIST.
  ZBAT-TKN-SEL-EXECUTORNAME      token-type ZSPI-TYP-STRING. !{}

  ZBAT-MAP-DEF-EXECUTOR
  Definition ZBAT-DDL-DEF-EXECUTOR.
    02 ZCPU                      type ZSPI-DDL-INT.
    02 ZJOBNUMBER                type ZSPI-DDL-INT.
    02 ZCLASS-COUNT              type ZSPI-DDL-INT.
    02 ZCLASSES                  occurs 8 times.
      03 ZCLASSNAME              type ZBAT-DDL-NETBATCH-NAME.
      03 FILLER                  type ZSPI-DDL-INT.
    02 ZCLASS                    type ZBAT-DDL-NETBATCH-NAME.
  End

  ZSPI-TKN-ERRLIST               token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST               token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE               token-type ZSPI-TYP-ENUM. !{}
  ZSPI-TKN-ENDLIST               token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-CONTEXT               token-type ZSPI-TYP-
  BYTESTRING.

```

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the INFO EXECUTOR command is:

ZBAT-TKN-SEL-EXECUTORNAME

specifies an executor name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of executor names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The INFO EXECUTOR command is a nonsensitive command available to all users.

## INFO JOB Command

The INFO JOB command lists the attributes of a job.

```

Command
  ZBAT-CMD-INFO

Object Type
  ZBAT-OBJ-JOB

Tokens in Command Buffer
  ZBAT-TKN-SEL-JOBNAME                token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-NOTJOBNAME             token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-JOB-NUMBER             token-type ZBAT-TYP-JOB-NUMBER.
  ! {A}
  ZBAT-TKN-SEL-ADPNAME                token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-NOTADPNAME             token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-CLASSNAME              token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-NOTCLASSNAME           token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-INNAME                 token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-NOTINNAME              token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-LIST                   token-type ZBAT-TYP-LIST. ! {A}...
  ZBAT-TKN-SEL-NOTLIST                token-type ZSPI-TYP-INT. ! {A}...
  ZBAT-TKN-SEL-USERNAME                token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-NOTUSERNAME             token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-WAITON                  token-type ZSPI-TYP-STRING. ! {A}...
  ZBAT-TKN-SEL-NOTWAITON               token-type ZSPI-TYP-STRING! {A}...
  ZSPI-TKN-COMMENT                     token-type ZSPI-TYP-STRING.
  ZSPI-TKN-CONTEXT                     token-type ZSPI-TYP-BYTESTRING.

Tokens in Response Buffer
  ZSPI-TKN-DATALIST                   token-type ZSPI-TYP-LIST.
  ZBAT-TKN-SEL-JOBNAME                 token-type ZSPI-TYP-STRING.
  ZBAT-TKN-SEL-JOB-NUMBER              token-type ZBAT-TYP-JOB-NUMBER.

  ZBAT-MAP-DEF-JOB
    Definition ZBAT-DDL-DEF-JOB.
      02 ZCLASSNAME                     type ZBAT-DDL-NETBATCH-NAME.
      02 ZHOLD                           type ZSPI-DDL-BOOLEAN.
      02 ZHOLD-AFTER                     type ZSPI-DDL-BOOLEAN.
      02 ZRESTART                       type ZSPI-DDL-BOOLEAN.
      02 ZSTOP-ON-ABEND                  type ZSPI-DDL-BOOLEAN.
      02 ZAT-FLAG                        type ZSPI-DDL-BOOLEAN.
      02 ZIFFAILS                        type ZSPI-DDL-BOOLEAN.
      02 ZPURGE-IN-FILE                  type ZSPI-DDL-BOOLEAN.
      02 ZSTALL                          type ZSPI-DDL-BOOLEAN.
      02 ZINFO-NEXT-RUNTIME               type ZSPI-DDL-INT4.
      02 ZINFO-OUT-SPOOL-NUM             type ZSPI-DDL-INT.
      02 ZINFO-WHICH-LIST                 type ZBAT-DDL-JOB-WHICH-LIST.
      02 ZINFO-SPECIAL-REASON            type ZBAT-DDL-SPECIAL-REASON.
      02 ZINFO-TOTAL-CPU-TIME             type ZSPI-DDL-INT4.
      02 ZINFO-OPEN-ACCESSOR             type ZSPI-DDL-INT.
      02 ZRE MID                         type ZSPI-DDL-BOOLEAN.
      02 ZE VERY-DAYS                     type ZSPI-DDL-INT.
      02 ZE VERY-HOURS                     type ZSPI-DDL-INT.
      02 ZE VERY-MINUTES                  type ZSPI-DDL-INT.
      02 ZDEFAULT-SECURITY               type ZSPI-DDL-INT.
      02 ZPRI                             type ZSPI-DDL-INT.
      02 ZSELPRI                         type ZSPI-DDL-INT.
      02 ZHIGHPIN                        type ZSPI-DDL-BOOLEAN.
      02 ZMAXPRINTLINES                  type ZSPI-DDL-INT2.
      02 ZMAXPRINTPAGES                  type ZSPI-DDL-INT2.
      02 ZTAPEDRIVES                     type ZSPI-DDL-INT.

```

(continued)

```

02 ZDATE.
03 ZYEAR                                type ZSPI-DDL-INT.
03 ZMONTH                              type ZSPI-DDL-INT.
03 ZDAY                                type ZSPI-DDL-INT.
02 ZTIME.
03 ZHOUR                                type ZSPI-DDL-INT.
03 ZMINUTE                             type ZSPI-DDL-INT.
03 ZSECOND                             type ZSPI-DDL-INT.
03 ZMILLISECOND                        type ZSPI-DDL-INT.
03 ZMICROSECOND                       type ZSPI-DDL-INT.
02 ZPOSIX                              type ZSPI-DDL-INT.
02 ZSAVEABEND                          type ZSPI-DDL-BOOLEAN.
02 ZRUND                               type ZSPI-DDL-BOOLEAN.
02 ZJOBID-ZERO                         type ZSPI-DDL-BOOLEAN.
02 ZMEM                                type ZSPI-DDL-INT.
02 ZPFS                                 type ZSPI-DDL-INT2.
02 ZNAME                               type ZSPI-DDL-CHAR8.
02 ZINFO-TIME-SUBMIT                   type ZSPI-DDL-INT4.
02 ZINFO-LAST-MOD                     type ZSPI-DDL-INT4.
02 ZINFO-LAST-MODUSER                 type ZSPI-DDL-INT.
02 ZTIME-LIMIT                         type ZSPI-DDL-INT2.
End

ZBAT-MAP-DEF-WAITON
Definition ZBAT-DDL-DEF-WAITON.
02 ZMASTER                            type ZBAT-DDL-NETBATCH-NAME.
02 ZINDICATOR                         type ZBAT-DDL-WAITON-INDICATOR.
02 ZFOR                               type ZBAT-DDL-WAITON-FOR.
End

ZBAT-TKN-ATT-SET-ID                    token-type ZSPI-TYP-STRING.
ZBAT-TKN-CALENDAR                     token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-DESCRIPTION                   token-type ZSPI-TYP-STRING.
ZBAT-TKN-EXECUTOR-PROGRAM              token-type ZSPI-TYP-STRING.
ZBAT-TKN-EXTSWAP-FILE                  token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-IN-FILE                       token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-LIB-FILE                      token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-OUT-FILE                      token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-STARTUP-MESSAGE               token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-SWAP-FILE                     token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-TERM-FILE                     token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-VOLUME-SUBVOL                 token-type ZSPI-TYP-BYTESTRING.

ZBAT-MAP-DEF-CRONTAB
Definition ZBAT-DDL-DEF-CRONTAB.
02 ZMINUTES                           type ZSPI-DDL-INT4.
02 ZHOURS                             type ZSPI-DDL-INT2.
02 ZDAYS                              type ZSPI-DDL-INT2.
02 ZMONTHS                            type ZSPI-DDL-INT.
02 ZWEEKDAYS                          type ZSPI-DDL-INT.
End

ZSPI-TKN-ERRLIST                       token-type ZSPI-TYP-LIST.
...
ZSPI-TKN-ENDLIST                       token-type ZSPI-TYP-SSCTL.
ZSPI-TKN-RETCODE                       token-type ZSPI-TYP-ENUM. ! {}
ZSPI-TKN-ENDLIST                       token-type ZSPI-TYP-SSCTL
ZSPI-TKN-CONTEXT                       token-type ZSPI-TYP-BYTESTRING.

```

## Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the INFO JOB command is:



ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the INFO JOB command is:

ZBAT-TKN-EXECUTOR-PROGRAM

specifies the value of the job's EXECUTOR-PROGRAM attribute.

ZBAT-TKN-OUT-FILE

specifies the value of the job's OUT attribute.

ZBAT-TKN-VOLUME-SUBVOL

specifies the default node, volume, and subvolume used for qualifying unqualified file references in the job's input file.

## Operational Notes

- The INFO JOB command is a nonsensitive command available to all users.
- The order in which the scheduler returns a job's attachment sets is the order in which the scheduler supplies them to the job.

## INFO SCHEDULER Command

The INFO SCHEDULER command lists scheduler attributes.

```

Command
  ZBAT-CMD-INFO

Object Type
  ZBAT-OBJ-SCHEDULER

Tokens in Command Buffer
  ZSPI-TKN-COMMENT                                token-type ZSPI-TYP-STRING.

Tokens in Response Buffer
  ZSPI-TKN-DATALIST                                token-type ZSPI-TYP-LIST.

  ZBAT-MAP-DEF-SCHEDULER
  Definition ZBAT-DDL-DEF-SCHEDULER.
    02 ZBACKUPCPU2                                type ZSPI-DDL-INT.
    02 ZBACKUPCPU1                                type ZSPI-DDL-INT.
    02 ZMAXCONCURRENTJOBS                         type ZSPI-DDL-INT.
    02 ZMAXTEMPEXECUTORS                         type ZSPI-DDL-INT.
    02 ZTAPEDRIVES                                type ZSPI-DDL-INT.
    02 ZMAXPRI                                     type ZSPI-DDL-INT.
    02 ZINFO-TAPEDRIVES-IN-USE                     type ZSPI-DDL-INT.
    02 ZAT-ALLOWED                                type ZSPI-DDL-BOOLEAN.
    02 ZSUBMIT-ALLOWED                             type ZSPI-DDL-BOOLEAN.
    02 ZEVERY-CATCHUP                             type ZSPI-DDL-BOOLEAN.
    02 ZEMS                                         type ZBAT-DDL-EMS.
    02 ZCLASSNAME                                  type ZBAT-DDL-NETBATCH-NAME.
    02 ZPRI                                         type ZSPI-DDL-INT.
    02 ZSELPRI                                     type ZSPI-DDL-INT.
    02 ZMAXPRINTLINES                             type ZSPI-DDL-INT2.
    02 ZMAXPRINTPAGES                             type ZSPI-DDL-INT2.
    02 ZSTOP-ON-ABEND                             type ZSPI-DDL-BOOLEAN.
    02 ZSTALL                                       type ZSPI-DDL-BOOLEAN.
    02 ZHIGHPIN                                    type ZSPI-DDL-BOOLEAN.
    02 ZINITIATION                                type ZSPI-DDL-BOOLEAN.
    02 ZLOCALNAMES                                type ZSPI-DDL-CHAR8
                                                occurs 30 times.

  End
  ZBAT-TKN-EXECUTOR-PROGRAM                        token-type ZSPI-TYP-STRING.
  ZBAT-TKN-OUT-FILE                                token-type ZSPI-TYP-
BYTESTRING.
  ZSPI-TKN-ERRLIST                                token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST                                token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE                                token-type ZSPI-TYP-ENUM. ! {}
  ZSPI-TKN-ENDLIST                                token-type ZSPI-TYP-SSCTL.

```

### Tokens in Command Buffer

For information on the token present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the INFO SCHEDULER command is:

ZBAT-TKN-EXECUTOR-PROGRAM

specifies the value of the scheduler's DEFAULT-EXECUTOR-PROGRAM attribute.

ZBAT-TKN-OUT-FILE

specifies the value of the scheduler's DEFAULT-OUT attribute.

## Operational Note

The INFO SCHEDULER command is a nonsensitive command available to all users.

## RELEASE JOB Command

The RELEASE JOB command enables a master job to release one or more of its dependents.

|                           |                                    |
|---------------------------|------------------------------------|
| Command                   |                                    |
| ZBAT-CMD-RELEASE          |                                    |
| Object Type               |                                    |
| ZBAT-OBJ-JOB              |                                    |
| Tokens in Command Buffer  |                                    |
| ZBAT-MAP-PAR-RELEASE-JOB  |                                    |
| Definition                | ZBAT-DDL-PAR-RELEASE-JOB.          |
| 02 ZRELEASER              | type ZBAT-DDL-NETBATCH-NAME.       |
| 02 ZJOBNAME               | type ZBAT-DDL-NETBATCH-NAME. ! { } |
| End                       |                                    |
| ZSPI-TKN-RESPONSE-TYPE    | token-type ZSPI-TYP-ENUM.          |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.        |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.    |
| Tokens in Response Buffer |                                    |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.          |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING.        |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.    |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.          |
| ...                       |                                    |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.         |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }    |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.         |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.    |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the RELEASE JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies the name of a released job.

ZBAT-TKN-SEL-JOB-NUMBER

specifies the number of job ZBAT-TKN-SEL-JOBNAME.

**ZSPI-TKN-RESPONSE-TYPE**

is the standard SPI response-type token that lets the requester specify whether the server (that is, the scheduler) is to return all responses. The values are:

|                            |   |
|----------------------------|---|
| ZSPI-VAL-ERR-AND-WARN      | Causes the scheduler to return a single response record indicating whether the command was successful. ZSPI-TKN-DATALIST tokens returned in the response record are ZSPI-TKN-ERRLIST, ZSPI-TKN-ENDLIST, and ZSPI-TKN-RETCODE.   |
| ZSPI-VAL-ERR-WARN-AND-NORM | Causes the scheduler to return a response record for each job it releases from the set of jobs specified in the command. ZSPI-TKN-DATALIST tokens returned in each response record are ZBAT-TKN-SEL-JOBNAME, ZBAT-TKN-SEL-JOB-NUMBER, ZSPI-TKN-ERRLIST, ZSPI-TKN-ENDLIST, and ZSPI-TKN-RETCODE. |

If you specify ZSPI-VAL-ERR-WARN-AND-NORM, the requester must receive all responses from and return all context tokens to the scheduler. If the requester does not do this, the command might not release all possible jobs.

The default value is ZSPI-VAL-ERR-AND-WARN.

## Operational Notes

- The RELEASE JOB command is a sensitive command available to all users.
- The process sending the command (that is, the requester process) must be a NetBatch job.
- The requester process (the master job) must supply its own job name in the ZRELEASER field of ZBAT-MAP-PAR-RELEASE-JOB if the dependent job is running in a different scheduler from that of the master job. For the requester process to get its own job name, it must:
  1. Call Guardian procedure `PROCESS_GETINFO_` to return its job ID and the process handle of its ancestor (that is, the process handle of its GMOM, or scheduler). The procedure's *jobid* and *gmom's-processhandle* fields return this information.
  2. Call Guardian procedure `PROCESS_GETINFO_` again, this time supplying the process handle obtained at Step 1 in the procedure's *processhandle* field. The procedure returns the scheduler's name in the *proc-fname:maxlen* field.
  3. Execute a `STATUS JOB job-ID` command on the scheduler identified at Step 2, where *job-ID* is the job ID returned at Step 1. The command returns the job name in response-buffer token ZBAT-TKN-SEL-JOBNAME.

## RUNNEXT JOB Command

The RUNNEXT JOB command makes the scheduler run a job immediately when an executor associated with the job's class is available. The command overrides the job's dependencies, timing attributes, and selection priority.

|                           |                                      |
|---------------------------|--------------------------------------|
| Command                   |                                      |
| ZBAT-CMD-RUNNEXT          |                                      |
| Object Type               |                                      |
| ZBAT-OBJ-JOB              |                                      |
| Tokens in Command Buffer  |                                      |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTJOBNAME   | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.      |
| ! {A}...                  |                                      |
| ZBAT-TKN-SEL-ADPNAME      | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTADPNAME   | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-CLASSNAME    | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTCLASSNAME | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-INNAME       | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTINNAME    | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-LIST         | token-type ZBAT-TYP-LIST. ! {A}...   |
| ZBAT-TKN-SEL-NOTLIST      | token-type ZSPI-TYP-INT. ! {A}...    |
| ZBAT-TKN-SEL-USERNAME     | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTUSERNAME  | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-WAITON       | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTWAITON    | token-type ZSPI-TYP-STRING. ! {A}... |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.          |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.      |
| Tokens in Response Buffer |                                      |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.            |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.      |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.            |
| ...                       |                                      |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.           |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }      |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.           |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.      |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the RUNNEXT JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The RUNNEXT JOB command is a sensitive command available to NetBatch supervisors only.

## RUNNOW JOB Command

The RUNNOW JOB command makes the scheduler run a job immediately. The command overrides job dependencies, timing attributes, and selection priority and causes the creation of a temporary executor.

|                           |                                      |
|---------------------------|--------------------------------------|
| Command                   |                                      |
| ZBAT-CMD-RUNNOW           |                                      |
| Object Type               |                                      |
| ZBAT-OBJ-JOB              |                                      |
| Tokens in Command Buffer  |                                      |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTJOBNAME   | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.      |
| ! {A}...                  |                                      |
| ZBAT-TKN-SEL-ADPNAME      | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTADPNAME   | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-CLASSNAME    | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTCLASSNAME | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-INNAME       | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTINNAME    | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-LIST         | token-type ZBAT-TYP-LIST. ! {A}...   |
| ZBAT-TKN-SEL-NOTLIST      | token-type ZSPI-TYP-INT. ! {A}...    |
| ZBAT-TKN-SEL-USERNAME     | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTUSERNAME  | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-WAITON       | token-type ZSPI-TYP-STRING. ! {A}... |
| ZBAT-TKN-SEL-NOTWAITON    | token-type ZSPI-TYP-STRING. ! {A}... |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.          |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.      |
| Tokens in Response Buffer |                                      |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.            |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.      |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.            |
| ...                       |                                      |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.           |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }      |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.           |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.      |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the RUNNOW JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).



## Operational Note

The RUNNOW JOB command is a sensitive command available to NetBatch supervisors only.

## SHUTDOWN SCHEDULER Command

The SHUTDOWN SCHEDULER command shuts down a scheduler. The command allows executing and over-limit jobs to finish before shutdown, but it immediately stops suspended jobs and their processes.

|                           |                                |
|---------------------------|--------------------------------|
| Command                   |                                |
| ZBAT-CMD-SHUTDOWN         |                                |
| Object Type               |                                |
| ZBAT-OBJ-SCHEDULER        |                                |
| Tokens in Command Buffer  |                                |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.    |
| Tokens in Response Buffer |                                |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.      |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.      |
| ...                       |                                |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.     |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. !{ } |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.     |

### Tokens in Command Buffer

For information on the token present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The SHUTDOWN SCHEDULER command is a sensitive command available to NetBatch supervisors only.

## START EXECUTOR Command

The START EXECUTOR command starts executors whose state is OFF or STOP, thus making them available for use by jobs.

|                           |                                   |
|---------------------------|-----------------------------------|
| Command                   |                                   |
| ZBAT-CMD-START            |                                   |
| Object Type               |                                   |
| ZBAT-OBJ-EXECUTOR         |                                   |
| Tokens in Command Buffer  |                                   |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.       |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.   |
| Tokens in Response Buffer |                                   |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.         |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.         |
| ...                       |                                   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }   |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.   |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the START EXECUTOR command is:

ZBAT-TKN-SEL-EXECUTORNAME

specifies an executor name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of executor names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The START EXECUTOR command is a sensitive command available to NetBatch supervisors only.

## START SCHEDULER Command

The START SCHEDULER command makes available for use a scheduler you are cold starting or warm starting.

|                           |                                 |
|---------------------------|---------------------------------|
| Command                   |                                 |
| ZBAT-CMD-START            |                                 |
| Object Type               |                                 |
| ZBAT-OBJ-SCHEDULER        |                                 |
| Tokens in Command Buffer  |                                 |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.     |
| Tokens in Response Buffer |                                 |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.       |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.       |
| ...                       |                                 |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.      |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { } |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.      |

### Tokens in Command Buffer

For information on the token present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The START SCHEDULER command is a sensitive command available to NetBatch supervisors only.

## STATUS ATTACHMENT-SET Command

The STATUS ATTACHMENT-SET command lists attachment sets and the names and owners of jobs using those sets.

|                                 |                                  |
|---------------------------------|----------------------------------|
| Command<br>ZBAT-CMD-STATUS      |                                  |
| Object Type<br>ZBAT-OBJ-ATT-SET |                                  |
| Tokens in Command Buffer        |                                  |
| ZBAT-TKN-ATT-SET-ID             | token-type ZSPI-TYP-STRING. !{ } |
| ZSPI-TKN-COMMENT                | token-type ZSPI-TYP-STRING.      |
| ZSPI-TKN-CONTEXT                | token-type ZSPI-TYP-BYTESTRING.  |
| Tokens in Response Buffer       |                                  |
| ZSPI-TKN-DATALIST               | token-type ZSPI-TYP-LIST.        |
| ZBAT-TKN-ATT-SET-ID             | token-type ZSPI-TYP-STRING. !{ } |
| ZBAT-TKN-JOB-ID                 | token-type ZSPI-TYP-STRING.      |
| ZSPI-TKN-ERRLIST                | token-type ZSPI-TYP-LIST.        |
| ...                             |                                  |
| ZSPI-TKN-ENDLIST                | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-RETCODE                | token-type ZSPI-TYP-ENUM. !{ }   |
| ZSPI-TKN-ENDLIST                | token-type ZSPI-TYP-SSCTL.       |
| ZSPI-TKN-CONTEXT                | token-type ZSPI-TYP-BYTESTRING.  |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the STATUS ATTACHMENT-SET command is:

ZBAT-TKN-ATT-SET-ID

specifies an attachment set or a range of attachment sets in the form:

[ ( *user-ID* ) ] *attachment-set-ID*

*user-ID*

specifies a user ID or, when specified with either or both of the asterisk (\*) and question mark (?) wild-card characters, a range of user IDs. (*user-ID* must be in *group-name.user-name* or *group-ID, user-ID* form.) The default is the user ID of the current user.

*attachment-set-ID*

is one of:

*attachment-set-name*

specifies the name of an attachment set owned by *user-ID* or, when specified with either or both of the asterisk (\*) and question mark (?) wild-card characters, a range of names of attachment sets owned by *user-ID*.

*attachment-set-number*

specifies the number of an attachment set owned by *user-ID*.

\*

specifies all attachment sets owned by *user-ID*.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The STATUS ATTACHMENT-SET command is a nonsensitive command available to all users.

## STATUS EXECUTOR Command

The STATUS EXECUTOR command returns executors' names, processors, and states. If an executor is in use by a job, the command also returns the job's number and the name of its class.

|                                      |                                   |
|--------------------------------------|-----------------------------------|
| Command                              |                                   |
| ZBAT-CMD-STATUS                      |                                   |
| Object Type                          |                                   |
| ZBAT-OBJ-EXECUTOR                    |                                   |
| Tokens in Command Buffer             |                                   |
| ZBAT-TKN-SEL-EXECUTORNAME            | token-type ZSPI-TYP-STRING. ! { } |
| ZSPI-TKN-COMMENT                     | token-type ZSPI-TYP-STRING.       |
| ZSPI-TKN-CONTEXT                     | token-type ZSPI-TYP-BYTESTRING.   |
| Tokens in Response Buffer            |                                   |
| ZSPI-TKN-DATALIST                    | token-type ZSPI-TYP-LIST.         |
| ZBAT-TKN-SEL-EXECUTORNAM             | token-type ZSPI-TYP-STRING. ! { } |
| ZBAT-MAP-STATUS-EXECUTOR             |                                   |
| Definition ZBAT-DDL-STATUS-EXECUTOR. |                                   |
| 02 ZCPU                              | type ZSPI-DDL-INT.                |
| 02 ZJOBNUMBER                        | type ZSPI-DDL-INT.                |
| 02 ZWHICH-LIST                       | type ZBAT-DDL-EXECUTOR-LIST.      |
| 02 ZCLASS                            | type ZBAT-DDL-NETBATCH-NAME.      |
| End                                  |                                   |
| ZSPI-TKN-ERRRLIST                    | token-type ZSPI-TYP-LIST.         |
| ...                                  |                                   |
| ZSPI-TKN-ENDLIST                     | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-RETCODE                     | token-type ZSPI-TYP-ENUM. ! { }   |
| ZSPI-TKN-ENDLIST                     | token-type ZSPI-TYP-SSCTL.        |
| ZSPI-TKN-CONTEXT                     | token-type ZSPI-TYP-BYTESTRING.   |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the STATUS EXECUTOR command is:

ZBAT-TKN-SEL-EXECUTORNAME

specifies an executor name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of executor names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The STATUS EXECUTOR command is a nonsensitive command available to all users.

## STATUS JOB Command

The STATUS JOB command returns a job's number, name, owner, state, class, selection priority or spooler log file number, and an indicator specifying whether the job was submitted from a node local to or remote from the scheduler. The command also returns these information when applicable to the job: next and previous run times; start and finish times; the time when the job's state changed; the name of the executor in use by the job; the process handle of the job's executor-program process; and run statistics.

```

Command
ZBAT-CMD-STATUS

Object Type
ZBAT-OBJ-JOB

Tokens in Command Buffer
ZBAT-TKN-SEL-JOBNAME           token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-NOTJOBNAME        token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-JOB-NUMBER        token-type ZBAT-TYP-JOB-NUMBER. ! {A}...
ZBAT-TKN-SEL-ADPNAME           token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-NOTADPNAME        token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-CLASSNAME         token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-NOTCLASSNAME      token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-INNAME            token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-NOTINNAME         token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-LIST              token-type ZBAT-TYP-LIST. ! {A}...
ZBAT-TKN-SEL-NOTLIST           token-type ZSPI-TYP-INT. ! {A}...
ZBAT-TKN-SEL-USERNAME          token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-NOTUSERNAME       token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-WAITON            token-type ZSPI-TYP-STRING. ! {A}...
ZBAT-TKN-SEL-NOTWAITON         token-type ZSPI-TYP-STRING. ! {A}...
ZSPI-TKN-COMMENT               token-type ZSPI-TYP-STRING.
ZSPI-TKN-CONTEXT               token-type ZSPI-TYP-BYTESTRING.

Tokens in Response Buffer
ZSPI-TKN-DATALIST              token-type ZSPI-TYP-LIST.
  ZBAT-TKN-SEL-JOBNAME          token-type ZSPI-TYP-STRING.
  ZBAT-TKN-SEL-JOB-NUMBER       token-type ZBAT-TYP-JOB-NUMBER.

ZBAT-MAP-STATUS-JOB
  Definition ZBAT-DDL-STATUS-JOB.
    02 ZOUT-SPOOL-NUM           type ZSPI-DDL-INT.
    02 ZSELPRI                  type ZSPI-DDL-INT.
    02 ZOPEN-ACCESSOR-DETAIL.   type ZSPI-DDL-BYTE.
      03 ZGROUP                  type ZSPI-DDL-BYTE.
      03 ZUSER                    type ZSPI-DDL-BYTE.
    02 ZOPEN-ACCESSOR           redefines ZOPEN-ACCESSOR-DETAIL
                                type ZSPI-DDL-INT.
    02 ZCLASSNAME                type ZBAT-DDL-NETBATCH-NAME.
    02 ZWHICH-LIST               type ZBAT-DDL-JOB-WHICH-LIST.
    02 ZSPECIAL-REASON          type ZBAT-DDL-SPECIAL-REASON.
    02 ZNEXT-RUNTIME            type ZSPI-DDL-INT4.
    02 ZTIME-PREV-RUNTIME       type ZSPI-DDL-INT4.
    02 ZTIME-START              type ZSPI-DDL-INT4.
    02 ZTIME-FINISH             type ZSPI-DDL-INT4.
    02 ZTIME-PUT-ON-LIST        type ZSPI-DDL-INT4.
  (continued)

```



|                       |                                 |
|-----------------------|---------------------------------|
| 02 ZTIME-USED         | type ZSPI-DDL-INT4.             |
| 02 ZREMIID            | type ZSPI-DDL-BOOLEAN.          |
| 02 ZEXECUTOR          | type ZBAT-DDL-NETBATCH-NAME.    |
| 02 ZEXECPHANDLE       | type ZSPI-DDL-PHANDLE.          |
| 02 ZTIME-ELAPSEDMAX   | type ZSPI-DDL-INT4.             |
| 02 ZTIME-CPUMAX       | type ZSPI-DDL-INT4.             |
| 02 ZTIME-ELAPSEDTOTAL | type ZSPI-DDL-INT4.             |
| 02 ZTIME-CPUTOTAL     | type ZSPI-DDL-INT4.             |
| 02 ZTIME-SUBMIT       | type ZSPI-DDL-INT4.             |
| 02 ZLAST-CC           | type ZSPI-DDL-INT.              |
| 02 ZTIMES-RUN         | type ZSPI-DDL-INT2.             |
| 02 ZTIME-LIMIT        | type ZSPI-DDL-INT2.             |
| End                   |                                 |
| ZBAT-TKN-PHANDLE      | token-type ZSPI-TYP-PHANDLE!{}  |
| ZSPI-TKN-ERRLIST      | token-type ZSPI-TYP-LIST.       |
| ...                   |                                 |
| ZSPI-TKN-ENDLIST      | token-type ZSPI-TYP-SSCTL.      |
| ZSPI-TKN-RETCODE      | token-type ZSPI-TYP-ENUM.!{}    |
| ZSPI-TKN-ENDLIST      | token-type ZSPI-TYP-SSCTL.      |
| ZSPI-TKN-CONTEXT      | token-type ZSPI-TYP-BYTESTRING. |

## Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the STATUS JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The STATUS JOB command is a nonsensitive command available to all users.

## STATUS SCHEDULER Command

The STATUS SCHEDULER command displays information about a scheduler.

```

Command
  ZBAT-CMD-STATUS

Object Type
  ZBAT-OBJ-SCHEDULER

Tokens in Command Buffer
  ZSPI-TKN-COMMENT                                token-type ZSPI-TYP-STRING.

Tokens in Response Buffer
  ZSPI-TKN-DATALIST                                token-type ZSPI-TYP-LIST.

  ZBAT-MAP-STATUS-SCHEDULER
    Definition ZBAT-DDL-STATUS-SCHEDULER
      02 ZSTATE                                     type ZBAT-DDL-SCHEDULER-STATE.
      02 ZEXECUTOR.
        03 ZOFF                                     type ZSPI-DDL-INT.
        03 ZON                                      type ZSPI-DDL-INT.
        03 ZACTIVE                                  type ZSPI-DDL-INT.
        03 ZSTOP                                    type ZSPI-DDL-INT.
        03 ZDOWN                                    type ZSPI-DDL-INT.
        03 ZDELETE                                  type ZSPI-DDL-INT.
      02 ZJOB.
        03 ZREADY                                   type ZSPI-DDL-INT.
        03 ZEXECUTING                              type ZSPI-DDL-INT.
        03 ZSPECIAL                                type ZSPI-DDL-INT.
        03 ZTIME                                    type ZSPI-DDL-INT.
        03 ZEVENT                                   type ZSPI-DDL-INT.
        03 ZSUSPENDED                              type ZSPI-DDL-INT.
        03 ZRUNNEXT                                type ZSPI-DDL-INT.
        03 ZRUNNOW                                  type ZSPI-DDL-INT.
        03 ZTAPE                                    type ZSPI-DDL-INT.
      02 ZJOBCLASS.
        03 ZOFF                                     type ZSPI-DDL-INT.
        03 ZON                                      type ZSPI-DDL-INT.
      02 ZPROCESS.
        03 ZACTIVE                                  type ZSPI-DDL-INT.
        03 ZSUSPENDED                              type ZSPI-DDL-INT.
      02 ZTAPE.
        03 ZCONFIG                                  type ZSPI-DDL-INT.
        03 ZTAPEDRIVES-IN-USE                      type ZSPI-DDL-INT.
      02 ZATT-SET-COUNT                             type ZSPI-DDL-INT.
      02 ZINITIATION                               type ZSPI-DDL-BOOLEAN.
      02 ZSUBMIT-ALLOWED                           type ZSPI-DDL-BOOLEAN.
    End

  ZBAT-TKN-BATCHCTL                                token-type ZSPI-TYP-STRING.
  ZBAT-TKN-LOG-FILE                                token-type ZSPI-TYP-STRING.
  ZSPI-TKN-ERRLIST                                 token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST                                 token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE                                 token-type ZSPI-TYP-ENUM. ! {}
  ZSPI-TKN-ENDLIST                                 token-type ZSPI-TYP-SSCTL.

```

## Tokens in Command Buffer

For information on the token present in the command buffer, see [Section 4, Common Definitions](#).

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Note

The STATUS SCHEDULER command is a nonsensitive command available to all users.

## STOP EXECUTOR Command

The STOP EXECUTOR command stops executors whose state is ACTIVE or ON, thus making them unavailable for use by jobs.

|                           |                                |
|---------------------------|--------------------------------|
| Command                   |                                |
| ZBAT-CMD-STOP             |                                |
| Object Type               |                                |
| ZBAT-OBJ-EXECUTOR         |                                |
| Tokens in Command Buffer  |                                |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type ZSPI-TYP-STRING.    |
| !{ }                      |                                |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.    |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-           |
| BYTESTRING.               |                                |
| Tokens in Response Buffer |                                |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.      |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type ZSPI-TYP-STRING.    |
| !{ }                      |                                |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.      |
| ...                       |                                |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.     |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. !{ } |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.     |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-           |
| BYTESTRING.               |                                |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the STOP EXECUTOR command is:

ZBAT-TKN-SEL-EXECUTORNAME

specifies an executor name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of executor names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The STOP EXECUTOR command is a sensitive command available to NetBatch supervisors only.

## STOP JOB Command

The STOP JOB command stops executing or suspended processes associated with a job.

|                           |                                       |
|---------------------------|---------------------------------------|
| Command                   |                                       |
| ZBAT-CMD-STOP             |                                       |
| Object Type               |                                       |
| ZBAT-OBJ-JOB              |                                       |
| Tokens in Command Buffer  |                                       |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-NOTJOBNAME   | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.       |
| ! {A} ...                 |                                       |
| ZBAT-TKN-SEL-ADPNAME      | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-NOTADPNAME   | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-CLASSNAME    | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-NOTCLASSNAME | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-INNAME       | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-NOTINNAME    | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-LIST         | token-type ZBAT-TYP-LIST. ! {A} ...   |
| ZBAT-TKN-SEL-NOTLIST      | token-type ZSPI-TYP-INT. ! {A} ...    |
| ZBAT-TKN-SEL-USERNAME     | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-NOTUSERNAME  | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-WAITON       | token-type ZSPI-TYP-STRING. ! {A} ... |
| ZBAT-TKN-SEL-NOTWAITON    | token-type ZSPI-TYP-STRING! {A} ...   |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.           |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.       |
| Tokens in Response Buffer |                                       |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.             |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING.           |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.       |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.             |
| ...                       |                                       |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.            |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }       |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.            |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.       |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the STOP JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Notes

- The STOP JOB command is a sensitive command. It is available to all users, but these conditions apply:
  - You can stop jobs belonging to any user if you are a NetBatch supervisor.
  - You can stop any job whose input file is a disk file to which you have write access whether or not you are a NetBatch supervisor. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can stop the job.
- A STOP JOB command that specifies neither ZBAT-TKN-SEL-USERNAME nor ZBAT-TKN-SEL-NOTUSERNAME acts only on jobs owned by the requesting user.

## SUBMIT JOB Command

The SUBMIT JOB command submits a job to a scheduler.

```

Command
  ZBAT-CMD-SUBMIT

Object Type
  ZBAT-OBJ-JOB

Tokens in Command Buffer
  ZBAT-TKN-SEL-JOBNAME                                token-type ZSPI-TYP-STRING.!!{}

  ZBAT-MAP-DEF-JOB
    Definition ZBAT-DDL-DEF-JOB.
      02 ZCLASSNAME                                     type ZBAT-DDL-NETBATCH-NAME.
      02 ZHOLD                                           type ZSPI-DDL-BOOLEAN.
      02 ZHOLD-AFTER                                     type ZSPI-DDL-BOOLEAN.
      02 ZRESTART                                       type ZSPI-DDL-BOOLEAN.
      02 ZSTOP-ON-ABEND                                 type ZSPI-DDL-BOOLEAN.
      02 ZAT-FLAG                                       type ZSPI-DDL-BOOLEAN.
      02 ZIFFAILS                                       type ZSPI-DDL-BOOLEAN.
      02 ZPURGE-IN-FILE                                 type ZSPI-DDL-BOOLEAN.
      02 ZSTALL                                         type ZSPI-DDL-BOOLEAN.
      02 ZINFO-NEXT-RUNTIME                             type ZSPI-DDL-INT4.
      02 ZINFO-OUT-SPOOL-NUM                           type ZSPI-DDL-INT.
      02 ZINFO-WHICH-LIST                              type ZBAT-DDL-JOB-WHICH-LIST.
      02 ZINFO-SPECIAL-REASON                          type ZBAT-DDL-SPECIAL-REASON.
      02 ZINFO-TOTAL-CPU-TIME                          type ZSPI-DDL-INT4.
      02 ZINFO-OPEN-ACCESSOR                           type ZSPI-DDL-INT.
      02 ZREMIID                                        type ZSPI-DDL-BOOLEAN.
      02 ZEVEY-DAYS                                     type ZSPI-DDL-INT.
      02 ZEVEY-HOURS                                   type ZSPI-DDL-INT.
      02 ZEVEY-MINUTES                                 type ZSPI-DDL-INT.
      02 ZDEFAULT-SECURITY                             type ZSPI-DDL-INT.
      02 ZPRI                                           type ZSPI-DDL-INT.
      02 ZSELPRI                                       type ZSPI-DDL-INT.
      02 ZHIGHPIN                                       type ZSPI-DDL-BOOLEAN.
      02 ZMAXPRINTLINES                                type ZSPI-DDL-INT2.
      02 ZMAXPRINTPAGES                                type ZSPI-DDL-INT2.
      02 ZTAPEDRIVES                                   type ZSPI-DDL-INT.
      02 ZDATE.
        03 ZYEAR                                       type ZSPI-DDL-INT.
        03 ZMONTH                                      type ZSPI-DDL-INT.
        03 ZDAY                                        type ZSPI-DDL-INT.
      02 ZTIME.
        03 ZHOUR                                       type ZSPI-DDL-INT.
        03 ZMINUTE                                    type ZSPI-DDL-INT.
        03 ZSECOND                                    type ZSPI-DDL-INT.
        03 ZMILLISECOND                              type ZSPI-DDL-INT.
        03 ZMICROSECOND                              type ZSPI-DDL-INT.

(continued)

```

```

02 ZPOSIX                                type ZSPI-DDL-INT.
02 ZSAVEABEND                           type ZSPI-DDL-BOOLEAN.
02 ZRUND                                type ZSPI-DDL-BOOLEAN.
02 ZJOBID-ZERO                           type ZSPI-DDL-BOOLEAN.
02 ZMEM                                  type ZSPI-DDL-INT.
02 ZPFS                                  type ZSPI-DDL-INT2.
02 ZNAME                                  type ZSPI-DDL-CHAR8.
02 ZINFO-TIME-SUBMIT                     type ZSPI-DDL-INT4.
02 ZINFO-LAST-MOD                        type ZSPI-DDL-INT4.
02 ZINFO-LAST-MODUSER                    type ZSPI-DDL-INT.
02 ZTIME-LIMIT                           type ZSPI-DDL-INT2.
End

ZBAT-MAP-DEF-WAITON
  Definition ZBAT-DDL-DEF-WAITON.
    02 ZMASTER                           type ZBAT-DDL-NETBATCH-NAME.
    02 ZINDICATOR                         type ZBAT-DDL-WAITON-INDICATOR.
    02 ZFOR                               type ZBAT-DDL-WAITON-FOR.
  End

ZBAT-TKN-ATT-SET-ID                      token-type ZSPI-TYP-STRING.
ZBAT-TKN-CALENDAR                        token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-DESCRIPTION                     token-type ZSPI-TYP-STRING.
ZBAT-TKN-EXECUTOR-PROGRAM                token-type ZSPI-TYP-STRING.
ZBAT-TKN-EXTSWAP-FILE                    token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-IN-FILE                         token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-LIB-FILE                        token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-OUT-FILE                        token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-STARTUP-MESSAGE                  token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-SWAP-FILE                       token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-TERM-FILE                       token-type ZSPI-TYP-BYTESTRING.
ZBAT-TKN-VOLUME-SUBVOL                   token-type ZSPI-TYP-BYTESTRING.!!{}

ZBAT-MAP-DEF-CRONTAB
  Definition ZBAT-DDL-DEF-CRONTAB.
    02 ZMINUTES                           type ZSPI-DDL-INT4.
    02 ZHOURS                             type ZSPI-DDL-INT2.
    02 ZDAYS                              type ZSPI-DDL-INT2.
    02 ZMONTHS                            type ZSPI-DDL-INT.
    02 ZWEEKDAYS                          type ZSPI-DDL-INT.
  End

ZSPI-TKN-COMMENT                         token-type ZSPI-TYP-STRING.

Tokens in Response Buffer
ZSPI-TKN-DATALIST                        token-type ZSPI-TYP-LIST.
  ZBAT-TKN-SEL-JOBNAME                    token-type ZSPI-TYP-STRING.
  ZBAT-TKN-SEL-JOB-NUMBER                  token-type ZBAT-TYP-JOB-NUMBER.
  ZSPI-TKN-ERRLIST                         token-type ZSPI-TYP-LIST.
  ...
  ZSPI-TKN-ENDLIST                         token-type ZSPI-TYP-SSCTL.
  ZSPI-TKN-RETCODE                         token-type ZSPI-TYP-ENUM.!!{}
ZSPI-TKN-ENDLIST                         token-type ZSPI-TYP-SSCTL.

```



## Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the SUBMIT JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name. The name can contain 1 through 24 letters and numbers. It also can contain hyphens but must begin with a letter and end with a letter or number. The name cannot contain spaces.

ZBAT-TKN-ATT-SET-ID

specifies a value for the job's ATTACHMENT-SET attribute in the form:

```
[ ( user-ID ) ] attachment-set-ID
```

*user-ID*

specifies the user ID of the attachment-set owner. (*user-ID* must be in *group-name.user-name* or *group-ID,user-ID* form.) The default is the user ID of the current user.

*attachment-set-ID*

is one of:

*attachment-set-name*

specifies an attachment-set name.

*attachment-set-number*

specifies an attachment-set number.

A job can have up to three attachment sets, so ZBAT-TKN-ATT-SET-ID can appear in the command up to three times.

ZBAT-TKN-EXECUTOR-PROGRAM

is the name of a program file and specifies the job's EXECUTOR-PROGRAM attribute. The default is the value of the scheduler's DEFAULT-EXECUTOR-PROGRAM attribute.

ZBAT-TKN-OUT-FILE

is the name of an output file and specifies the job's OUT attribute. The default is the value of the scheduler's DEFAULT-OUT attribute.

ZBAT-TKN-VOLUME-SUBVOL

specifies the default node, volume, and subvolume used for qualifying unqualified file references in the job's input file. The token must specify, at a minimum, a

default volume and subvolume. If the node name is not specified, the scheduler uses the node of the requester.

## Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

## Operational Notes

- The SUBMIT JOB command is a sensitive command available to all users.
- The owner of a job is the owner of the requester process that submits the job.
- The order in which you specify a job's attachment sets is the order in which the scheduler supplies them to the job. For example, specifying sets C, B, and A in that order makes the scheduler process set C first, set B second, and set A third.
- If the name of an ASSIGN, DEFINE, or PARAM from a set conflicts with a name from a set specified earlier, the scheduler overwrites the earlier ASSIGN, DEFINE, or PARAM with the details of the later ASSIGN, DEFINE, or PARAM.
- These fields of ZBAT-MAP-DEF-JOB are returned in that token by the INFO JOB command. The scheduler ignores them in the SUBMIT JOB command.
 

|                      |                      |
|----------------------|----------------------|
| ZINFO-NEXT-RUNTIME   | ZINFO-TOTAL-CPU-TIME |
| ZINFO-OPEN-ACCESSOR  | ZINFO-WHICH-LIST     |
| ZINFO-OUT-SPOOL-NUM  | ZRE MID              |
| ZINFO-SPECIAL-REASON |                      |
- ZBAT-MAP-DEF-CRONTAB, ZBAT-TKN-CALENDAR, and the ZEVERY-DAYS, ZEVERY-HOURS, and ZEVERY-MINUTES fields of ZBAT-MAP-DEF-JOB are mutually exclusive.

## SUSPEND JOB Command

The SUSPEND JOB command suspends executing processes associated with a job.

|                           |  |
|---------------------------|--|
| Command                   |  |
| ZBAT-CMD-SUSPEND          |  |
| Object Type               |  |
| ZBAT-OBJ-JOB              |  |
| Tokens in Command Buffer  |  |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTJOBNAME   | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER. ! {A}... |
| ZBAT-TKN-SEL-ADPNAME      | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTADPNAME   | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-CLASSNAME    | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTCLASSNAME | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-INNAME       | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTINNAME    | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-LIST         | token-type ZBAT-TYP-LIST. ! {A}...       |
| ZBAT-TKN-SEL-NOTLIST      | token-type ZSPI-TYP-INT. ! {A}...        |
| ZBAT-TKN-SEL-USERNAME     | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTUSERNAME  | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-WAITON       | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZBAT-TKN-SEL-NOTWAITON    | token-type ZSPI-TYP-STRING. ! {A}...     |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.              |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.          |
| Tokens in Response Buffer |  |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.                |
| ZBAT-TKN-SEL-JOBNAME      | token-type ZSPI-TYP-STRING.              |
| ZBAT-TKN-SEL-JOB-NUMBER   | token-type ZBAT-TYP-JOB-NUMBER.          |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.                |
| ...                       |  |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.               |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { }          |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.               |
| ZSPI-TKN-CONTEXT          | token-type ZSPI-TYP-BYTESTRING.          |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#). Information on the tokens specific to the SUSPEND JOB command is:

ZBAT-TKN-SEL-JOBNAME

specifies a job name or, when specified with either or both the asterisk (\*) and question mark (?) wild-card characters, a range of job names.

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Notes

- The SUSPEND JOB command is a sensitive command. It is available to all users, but these conditions apply:

- You can suspend jobs belonging to any user if you are a NetBatch supervisor.
- You can suspend any job whose input file is a disk file to which you have write access whether or not you are a NetBatch supervisor. If the input file does not exist or is a device or process, only the owner and NetBatch supervisors can suspend the job.
- A SUSPEND JOB command that specifies neither ZBAT-TKN-SEL-USERNAME nor ZBAT-TKN-SEL-NOTUSERNAME acts only on jobs owned by the requesting user.

## SWITCHCPU SCHEDULER Command

The SWITCHCPU SCHEDULER command makes a scheduler's primary process run in the processor of its backup process, and the backup process run in the processor of its primary process.

|                           |                                |
|---------------------------|--------------------------------|
| Command                   |                                |
| ZBAT-CMD-SWITCHCPU        |                                |
| Object Type               |                                |
| ZBAT-OBJ-SCHEDULER        |                                |
| Tokens in Command Buffer  |                                |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.    |
| Tokens in Response Buffer |                                |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.      |
| ZSPI-TKN-ERRRLIST         | token-type ZSPI-TYP-LIST.      |
| ...                       |                                |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.     |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. !{ } |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.     |

### Tokens in Command Buffer

For information on the token present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The SWITCHCPU SCHEDULER command is a sensitive command available to NetBatch supervisors only.

## SWITCHLOG SCHEDULER Command

The SWITCHLOG SCHEDULER command closes the current scheduler log file and opens another.

|                           |                                 |
|---------------------------|---------------------------------|
| Command                   |                                 |
| ZBAT-CMD-SWITCHLOG        |                                 |
| Object Type               |                                 |
| ZBAT-OBJ-SCHEDULER        |                                 |
| Tokens in Command Buffer  |                                 |
| ZBAT-TKN-LOG-FILE         | token-type ZSPI-TYP-STRING.     |
| ZSPI-TKN-COMMENT          | token-type ZSPI-TYP-STRING.     |
| Tokens in Response Buffer |                                 |
| ZSPI-TKN-DATALIST         | token-type ZSPI-TYP-LIST.       |
| ZBAT-TKN-LOG-FILE         | token-type ZSPI-TYP-STRING.     |
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST.       |
| ...                       |                                 |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.      |
| ZSPI-TKN-RETCODE          | token-type ZSPI-TYP-ENUM. ! { } |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL.      |

### Tokens in Command Buffer

For information on the tokens present in the command buffer, see [Section 4, Common Definitions](#).

### Tokens in Response Buffer

For information on the tokens present in the response buffer, see [Section 4, Common Definitions](#).

### Operational Note

The SWITCHLOG SCHEDULER command is a sensitive command available to NetBatch supervisors only.

# 6 Event Messages

This section describes the EMS event messages that the NetBatch subsystem can issue and the specific programming considerations for dealing with these event messages in an application. For general information about EMS, see the *EMS Manual*.

In this section, event-message tokens and their values appear in DDL format. For an explanation of DDL, see the *Data Definition Language (DDL) Reference Manual*.

| Topic  | Page                |
|--|---------------------|
| <a href="#">The NetBatch EMS Interface</a>                               | <a href="#">6-1</a> |
| <a href="#">Enabling and Disabling NetBatch Event-Message Generation</a> | <a href="#">6-3</a> |
| <a href="#">How EMS Handles NetBatch Event Messages</a>                  | <a href="#">6-3</a> |
| <a href="#">Creating an EMS Filter</a>                                   | <a href="#">6-3</a> |
| <a href="#">Using the Format Template</a>                                | <a href="#">6-5</a> |
| <a href="#">Event-Message Descriptions</a>                               | <a href="#">6-9</a> |

## The NetBatch EMS Interface

The NetBatch subsystem (version D20 or later) has an EMS interface that it uses to send messages about certain events to the EMS collector \$0. The messages are in SPI format as described in the *SPI Programming Manual*.

## Event Messages Issued by the NetBatch Subsystem

The NETBATCH scheduler process issues the NetBatch subsystem's event messages. [Table 6-1](#) on page 6-2 lists and describes the messages. Detailed descriptions of the messages appear later in this section.

## Critical Events

The scheduler generates two classes of events:

- Critical events are errors and warnings for which the results could be serious, such as scheduler failure.
- Noncritical events are usually informational in nature, such as a job stopping without error.

The value of event-message token ZEMS-TKN-EMPHASIS determines whether an event is critical. If the value is ZSPI-VAL-TRUE, the event is critical. If the value is ZSPI-VAL-FALSE, the event is noncritical.

**Table 6-1. NetBatch Event Messages** (page 1 of 2)

| <b>Message Number</b> | <b>Symbolic Name</b>       | <b>Cause</b>  | <b>Emphasis</b> |
|-----------------------|----------------------------|---|-----------------|
| 100                   | ZBAT-EVT-SCHEDULER-START   | A START SCHEDULER command started the scheduler.  | Noncritical     |
| 101                   | ZBAT-EVT-SCHEDULER-STOP    | An ABORT SCHEDULER or SHUTDOWN SCHEDULER command shut down the scheduler.   | Noncritical     |
| 102                   | ZBAT-EVT-JOB-START         | The scheduler started the initial process (the executor program) of the job.  | Noncritical     |
| 200                   | ZBAT-EVT-EXECUTOR-DOWN     | The executor's processor went down.   | Noncritical     |
| 201                   | ZBAT-EVT-EXECUTOR-UP       | The executor's processor became available after being down.   | Noncritical     |
| 202                   | ZBAT-EVT-JOB-NORMAL-STOP   | The job stopped without error.  | Noncritical     |
| 203                   | ZBAT-EVT-JOB-ABNORMAL-STOP | <p>One of these events occurred:<br/>The job's executor-program process did one of these:</p> <ul style="list-style-type: none"> <li>● Abended by calling the system procedure <code>PROCESS_STOP_</code> (with the <code>ABEND</code> option specified) or <code>ABEND</code>.</li> <li>● Stopped with completion code -3, -2, -1, 2, 3, 4, 5, 6, or 7.</li> <li>● The scheduler stopped the job and all its processes because the job had the attribute <code>STOP-ON-ABEND ON</code> and did one of these:</li> <li>● Terminated because of processor</li> </ul> | Noncritical     |
| 204                   | ZBAT-EVT-JOB-OVER-LIMIT    | The job exceeded its execution time limit as specified by the <code>LIMIT</code> attribute.   | Noncritical     |
| 301                   | ZBAT-EVT-JOB-START-ERROR   | The scheduler failed to start the job's executor-program process.   | Noncritical     |



**Table 6-1. NetBatch Event Messages** (page 2 of 2)

| Message Number | Symbolic Name                 | Cause   | Emphasis |
|----------------|-------------------------------|---|----------|
| 500            | ZBAT-EVT-SCHEDULER-ABENDED    | The scheduler called the ABEND or PROCESS_STOP_ (with ABEND option) system procedure because of a system resource or scheduler problem. | Critical |
| 501            | ZBAT-EVT-LOGON-FAILURE        | The scheduler could not log on to the log file owner's user ID.   | Critical |
| 502            | ZBAT-EVT-LOGFILE-CREATE-ERROR | The scheduler could not create the scheduler log file.  | Critical |

## Enabling and Disabling NetBatch Event-Message Generation

You can enable NetBatch event-message generation when starting a scheduler or while the scheduler is running.

- To enable event-message generation when starting a scheduler, include the EMS parameter in the RUN NETBATCH command. For example:

```
5> NETBATCH /NAME $ZBAT, NOWAIT/ $NB.ZBAT EMS
6> BATCHCOM $ZBAT; INFO SCHEDULER, EMS
SCHEDULER ATTRIBUTES
ems : On
```

- To enable event-message generation while the scheduler is running, use the ALTER SCHEDULER command to set the scheduler's EMS attribute to ON or ERROR.
- To disable event-message generation, use the ALTER SCHEDULER command to set the EMS attribute to OFF.

## How EMS Handles NetBatch Event Messages

With event-message generation enabled, the scheduler sends a message to the EMS collector (\$0) when an event listed in [Table 6-1](#) on page 6-2 occurs. On receiving the message, EMS stores it in a disk file called the EMS event log. EMS distributors then collect and filter messages from the log and route the selected messages to the appropriate destination. Selection of scheduler event messages from a distributor is by an EMS filter loaded into the distributor. For more information, see the *EMS Manual*.

## Creating an EMS Filter

You can create an EMS filter that selects all scheduler event messages or specific messages only. To write and compile the filter, use EMF, the EMS filter language.

After writing and compiling the filter, load its compiled object file into a distributor such as the consumer distributor by using the EMS program EMSDIST. The filter then tests the value of tokens in event messages and filters the specified messages. You can write a filter to select scheduler event messages based on any token in the message.

This procedure describes how to write a filter to select specific NetBatch event messages, including the steps to compile the filter object file and load the file into a distributor. For more information, see the *EMS Manual*.

## Step 1: Decide on Scheduler Event Messages

Decide which scheduler event messages you want the filter to select:

```
Decision:Select all event messages from all schedulers.
```

## Step 2: Create EDIT Source File

Create an EDIT source file and enter the filter specifications using the EMF filter language:

```
> EDIT FILTSRC !
CURRENT FILE IS $NB.TRASH.FILTSRC
*ADD
  1      FILTER ZBATMSG;
  2      BEGIN
  3          IF ZSPI^TKN^SSID = SSID(ZBAT^VAL^EXTERNAL^SSID)
THEN
  4          PASS
  5          ELSE
  6          FAIL;
  7      END;
  8      //
*EXIT
>
```

## Step 3: Load TACL Definitions Into Memory

Load into memory from these files the standard TACL definitions for SPI, EMS, and scheduler tokens:

- \$vol.ZSPIDEF.ZSPITACL
- \$vol.ZSPIDEF.ZEMSTACL
- \$vol.ZSPIDEF.ZBATTACL

```
> #PUSH X
> #LOAD /KEEP 1, LOADED X/ $vol.ZSPIDEF.ZBATTACL
> #LOAD /KEEP 1, LOADED X/ $vol.ZSPIDEF.ZEMSTACL
> #LOAD /KEEP 1, LOADED X/ $vol.ZSPIDEF.ZSPITACL
> #POP X
>
```

## Step 4: Compile Filter Source File

Compile the filter source file by using the EMS filter compiler:

```
> EMF /IN FILTSRC/ FILTOBJ
      EMS Filter Compiler - T9634D20 - (01JUN93) - (19JUN90)
      (C)1987 Tandem (C)2004 Hewlett Packard Development
Company, L.P.
Source file: [1] $NB.TRASH.FILTSRC 1994-07-28 10:51:38
1 FILTER ZBATMSG;
2 BEGIN
3     IF ZSPI^TKN^SSID = SSID(ZBAT^VAL^EXTERNAL^SSID) THEN
4         PASS
5     ELSE
6         FAIL;
7 END;
8
```

## Step 5: Load the Compiled Filter Object File Into A Distributor

Load the compiled filter object file into a consumer, printing, or forwarding distributor by using the EMS program EMSDIST:

```
> EMSDIST TYPE PRINTING, COLLECTOR $0, FILTER FILTOBJ,
TEXTOUT
$MYTERM
```

## Using the Format Template

The NetBatch subsystem comes with a DSM format template. This template enables the Guardian procedure EMSTEXT to display scheduler event-message text in DSM display format. The template also specifies which tokens of each message the procedure will display and the message text. Your system administrator loads the template's source file (\$VOL.ZTEMPL.SBATTMPL) and object file (\$VOL.ZTEMPL.ZBATTMPL) when installing NetBatch software.

## Contents of the Format Template Supplied With NetBatch Software

[Table 6-2](#) on page 6-6 shows the statements in the scheduler's format-template source file (\$VOL.ZTEMPL.SBATTMPL) for each scheduler event message. The table also gives a sample of each message as formatted by the template's object file.

**Table 6-2. Contents of the Format Template Supplied With NetBatch Software**  
(page 1 of 3)

| <b>Message Number</b> | <b>Symbolic Name</b>  |
|-----------------------|---|
| 100                   | ZBAT-EVT-SCHEDULER-START  |
|                       | <p>Source statements: MSG: ZEMS-TKN-EVENTNUMBER, ZBAT-EVT-SCHEDULER-START</p> <p>"&lt;1&gt;&lt;2&gt;:&lt;3&gt; SCHEDULER &lt;4&gt; STARTED"</p> <p>1: ZEMS-TKN-SUBJECT-MARK</p> <p>2: ZSPI-TKN-NEXTTOKEN, MSG(ZBAT-TKN-FORMATSUBJECT)</p> <p>3: ZEMS-TKN-EVENTNUMBER, I</p> <p>== *nnnn*</p> <p>4: ZBAT-TKN-SCHEDULER-ID</p> <p>Sample message: 94-07-28 14:31:23 \MELBDEV.\$ZBAT<br/>TANDEM.BAT.D30 000100</p> <p>\MELBDEV.\$ZBAT:100 SCHEDULER</p> <p>\MELBDEV.\$ZBAT STARTED</p>   |
| 101                   | ZBAT-EVT-SCHEDULER-STOP   |
|                       | <p>Source statements: MSG: ZEMS-TKN-EVENTNUMBER, ZBAT-EVT-SCHEDULER-STOP</p> <p>"&lt;1&gt;&lt;2&gt;:&lt;3&gt; SCHEDULER &lt;4&gt; was stopped by &lt;7&gt;, log &lt;5&gt;, database &lt;6&gt;"</p> <p>1: ZEMS-TKN-SUBJECT-MARK</p> <p>2: ZSPI-TKN-NEXTTOKEN, MSG(ZBAT-TKN-FORMATSUBJECT)</p> <p>3: ZEMS-TKN-EVENTNUMBER, I</p> <p>== *nnnn*</p> <p>4: ZBAT-TKN-SCHEDULER-ID</p> <p>5: ZBAT-TKN-LOG-FILE</p> <p>6: ZBAT-TKN-DATA-BASE</p> <p>7: ZBAT-TKN-TEXT</p> <p>Sample message: 94-07-28 13:58:08 \MELBDEV.\$ZBAT<br/>TANDEM.BAT.D30 000101</p> <p>\MELBDEV.\$ZBAT:101 SCHEDULER</p> <p>\MELBDEV.\$ZBAT was stopped by</p> <p>SHUTDOWN SCHEDULER, log</p> <p>\MELBDEV.\$TRASH.ZBAT.LOGAAD, database</p> <p>\$TRASH.ZBAT</p> |
| 102                   | ZBAT-EVT-JOB-START  |

---

**Table 6-2. Contents of the Format Template Supplied With NetBatch Software**  
 (page 2 of 3)
**Message****Number      Symbolic Name**

Source statements:      MSG: ZEMS-TKN-EVENTNUMBER, ZBAT-EVT-JOB-START  
                               " <1><2>:<3> JOB <4>(<5>) started,  
                               program <6>, user ID <7>"  
                               1: ZEMS-TKN-SUBJECT-MARK  
                               2: ZSPI-TKN-NEXTTOKEN, MSG(ZBAT-TKN-FORMATSUBJECT)  
                               3: ZEMS-TKN-EVENTNUMBER, I  
                               == \*nnnn\*  
                               4: ZBAT-TKN-JOB-NAME-ID  
                               5: ZBAT-TKN-JOB-NUMBER, I  
                               6: ZBAT-TKN-EXECUTOR-PROGRAM  
                               7: ZBAT-TKN-USERID

Sample message:      94-07-28 12:51:06 \MELBDEV.\$ZBAT  
                               TANDEM.BAT.D30                000102

                              \MELBDEV.\$ZBAT:102 JOB X(1) started,  
                               program

                              \MELBDEV.\$SYSTEM.SYSTEM.TACL, user ID  
                               255255

## 200      ZBAT-EVT-EXECUTOR-DOWN

Source statements:      MSG: ZEMS-TKN-EVENTNUMBER, ZBAT-EVT-EXECUTOR-DOWN  
                               " <1><2>:<3> EXECUTOR <4> is down"  
                               1: ZEMS-TKN-SUBJECT-MARK  
                               2: ZSPI-TKN-NEXTTOKEN, MSG(ZBAT-TKN-FORMATSUBJECT)  
                               3: ZEMS-TKN-EVENTNUMBER, I  
                               == \*nnnn\*  
                               4: ZBAT-TKN-EXECUTOR-ID

Sample message:      94-07-28 14:48:09 \MELBDEV.\$ZBAT  
                               TANDEM.BAT.D30                000200

                              \MELBDEV.\$ZBAT:200 EXECUTOR EXEC1 is down

## 201      ZBAT-EVT-EXECUTOR-UP

Source statements:      MSG: ZEMS-TKN-EVENTNUMBER, ZBAT-EVT-EXECUTOR-UP  
                               " <1><2>:<3> EXECUTOR <4> is now up"  
                               1: ZEMS-TKN-SUBJECT-MARK  
                               2: ZSPI-TKN-NEXTTOKEN, MSG(ZBAT-TKN-FORMATSUBJECT)  
                               3: ZEMS-TKN-EVENTNUMBER, I  
                               == \*nnnn\*  
                               4: ZBAT-TKN-EXECUTOR-ID

---

**Table 6-2. Contents of the Format Template Supplied With NetBatch Software**  
(page 3 of 3)

| Message Number | Symbolic Name  |
|----------------|--|
|                | <div>Sample message: 94-07-28 14:50:20 \MELBDEV.\$ZBAT<br/>TANDEM.BAT.D30 000201<br/><br/>\MELBDEV.\$ZBAT:201 EXECUTOR EXEC1 is now<br/>up</div>   |
| 202            | <div>ZBAT-EVT-JOB-NORMAL-STOP</div> <div>Source statements: MSG: ZEMS-TKN-EVENTNUMBER, ZBAT-EVT-JOB-NORMAL-STOP<br/>" &lt;1&gt;&lt;2&gt;:&lt;3&gt; JOB &lt;4&gt;(&lt;5&gt;) has stopped<br/>with &lt;10&gt;"<br/>" &lt;*cr&gt;Job started at &lt;8&gt;, CPU usage<br/>&lt;9&gt; microsecs, program &lt;6&gt;, userid &lt;7&gt;&amp;<br/>."<br/>1: ZEMS-TKN-SUBJECT-MARK<br/>2: ZSPI-TKN-NEXTTOKEN, MSG(ZBAT-TKN-FORMATSUBJECT)<br/>3: ZEMS-TKN-EVENTNUMBER, I<br/>== *nnnn*<br/>4: ZBAT-TKN-JOB-NAME-ID<br/>5: ZBAT-TKN-JOB-NUMBER<br/>6: ZBAT-TKN-EXECUTOR-PROGRAM<br/>7: ZBAT-TKN-USERID<br/>8: ZBAT-TKN-START-TIME , TIME<br/>("H2:M2:S2")<br/>9: ZBAT-TKN-TOTAL-CPU-TIME , I<br/>10: ZBAT-TKN-COMPLETION-CODE</div> <div>Sample message: 94-07-28 13:22:21 \MELBDEV.\$ZBAT<br/>TANDEM.BAT.D30 000202<br/><br/>\MELBDEV.\$ZBAT:202 JOB Y(2) has stopped<br/>with<br/>Normal termination Job<br/>started at 13:21:12, CPU usage 13708<br/>microsecs, program<br/><br/>\MELBDEV.\$SYSTEM.SYSTEM.DELAY, userid<br/>255255.</div> |

## Modifying the Format Template

To modify the scheduler’s DSM format template, follow the template-modification procedure described in the *DSM Template Services Manual*.

# Event-Message Descriptions

This subsection describes event messages generated by the NetBatch subsystem. The messages appear in ascending order by event number. Each description contains:

- A header containing the event-message number and text.
- A box containing a list of the tokens that can appear in the event message. The two types of tokens are:
  - An unconditional token is a token that is always present in an event message; for example, ZEMS-TKN-EVENTNUMBER.
  - A conditional token is a token that is present in the event message only in certain cases. For example, conditional token ZBAT-TKN-TEXT appears in message ZBAT-EVT-JOB-NORMAL-STOP only when the ABEND, STOP, or PROCESS\_STOP\_ system procedure called by a job's executor-program process includes a text string in the process-deletion system message.

The box also contains the text version of the event message as displayed by Guardian procedure EMSTEXT.

- Information on the event-message tokens that [Section 4, Common Definitions](#) does not cover.
- Information about the variable items in the event-message text.
- An explanation of the cause of the event message.
- Details of the effect of the event message.
- A recommendation about the action required when the event occurs.

## 100 ZBAT-EVT-SCHEDULER-START

### Unconditional Tokens

ZSPI-TKN-SSID  
ZEMS-TKN-EVENTNUMBER  
ZEMS-TKN-EMPHASIS  
ZEMS-TKN-SUBJECT-MARK  
ZBAT-TKN-SCHEDULER-ID

token-type ZSPI-TYP-SSID  
token-type ZSPI-TYP-ENUM.  
token-type ZSPI-TYP-BOOLEAN.  
token-type ZSPI-TYP-MARK.  
token-type ZSPI-TYP-STRING.

### Event-Message Text

SCHEDULER \node.\$process-name STARTED

## Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

## Event-Message Text

*\node.\$process-name*

specifies the node and process name of the scheduler.

**Cause.** A START SCHEDULER command started the scheduler.

**Effect.** The scheduler is now running as a process-pair server.

**Recovery.** Informational message only; no corrective action required.

## 101 ZBAT-EVT-SCHEDULER-STOP

|   |                              |
|---|------------------------------|
| Unconditional Tokens  |                              |
| ZSPI-TKN-SSID   | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER  | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS   | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK   | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-SCHEDULER-ID   | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-LOG-FILE   | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-DATA-BASE  | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-TEXT   | token-type ZSPI-TYP-STRING.  |
| Event-Message Text  |                              |
| SCHEDULER <i>\node.\$process-name</i> was stopped by <i>command</i> , log <i>log-file-name</i> , database <i>schd-database-subvol</i> |                              |

## Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

## Event-Message Text

*\node.\$process-name*

specifies the node and process name of the scheduler.

*command*

specifies which of the commands ABORT SCHEDULER and SHUTDOWN SCHEDULER shut down the scheduler.

*log-file-name*

specifies the name of the scheduler's log file.

*schd-database-subvol*

specifies the location of the scheduler's database.

**Cause.** An ABORT SCHEDULER or SHUTDOWN SCHEDULER command shut down the scheduler.

**Effect.** The scheduler is no longer running.



**Recovery.** Informational message only; no corrective action required.

## 102 ZBAT-EVT-JOB-START

|  |                              |
|--|------------------------------|
| Unconditional Tokens   |                              |
| ZSPI-TKN-SSID  | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER   | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS  | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK  | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-JOB-NUMBER  | token-type ZSPI-TYP-INT.     |
| ZBAT-TKN-JOB-NAME-ID   | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-USERID  | token-type ZSPI-TYP-USERID.  |
| ZBAT-TKN-EXECUTOR-PROGRAM  | token-type ZSPI-TYP-STRING.  |
| Event-Message Text   |                              |
| JOB <i>job-name</i> ( <i>job-number</i> ) started, program <i>program-file-name</i> , user ID <i>group-ID</i> <i>user-ID</i> |                              |

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

### Event-Message Text

*job-name*

specifies the name of the job.

*job-number*

specifies the number of the job.

*program-file-name*

specifies the program file name of the job's executor-program process.

*group-ID user-ID*

specifies the user ID of the job owner.

**Cause.** The scheduler started the initial process (the executor program) of the job.

**Effect.** The job's executor program started successfully.

**Recovery.** Informational message only; no corrective action required.

## 200 ZBAT-EVT-EXECUTOR-DOWN

|                                       |                              |
|---------------------------------------|------------------------------|
| Unconditional Tokens                  |                              |
| ZSPI-TKN-SSID                         | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER                  | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS                     | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK                 | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-EXECUTOR-ID                  | token-type ZSPI-TYP-STRING.  |
| Event-Message Text                    |                              |
| EXECUTOR <i>executor-name</i> is down |                              |

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

### Event-Message Text

*executor-name*

specifies the name of the executor.

**Cause.** The executor's processor went down.

**Effect.** The executor is no longer available for use by jobs. A job that abended because the executor's processor went down will make the scheduler generate event message 203. For recovery action, see event message [203 ZBAT-EVT-JOB-ABNORMAL-STOP](#) on page 6-15.

**Recovery.** Alter the executor's processor attribute to specify an available processor when both these conditions exist:

- The executor's classes are unique to the executor.
- You want the scheduler to continue selecting jobs from the executor's classes.

You can reassign the executor to its original processor when the processor becomes available.

## 201 ZBAT-EVT-EXECUTOR-UP

|   |                              |
|---|------------------------------|
| Unconditional Tokens                    |                              |
| ZSPI-TKN-SSID                           | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER                    | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS                       | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK                   | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-EXECUTOR-ID                    | token-type ZSPI-TYP-STRING.  |
| Event-Message Text                      |                              |
| EXECUTOR <i>executor-name</i> is now up |                              |

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

### Event-Message Text

*executor-name*

specifies the name of the executor.

**Cause.** The executor's processor became available after being down.

**Effect.** The executor is now available for use by jobs.

**Recovery.** Reassign the executor to its original processor if you altered the executor's processor attribute when the processor went down.

## 202 ZBAT-EVT-JOB-NORMAL-STOP

|   |                                      |
|---|--------------------------------------|
| Unconditional Tokens  |                                      |
| ZSPI-TKN-SSID   | token-type ZSPI-TYP-SSID.            |
| ZEMS-TKN-EVENTNUMBER  | token-type ZSPI-TYP-ENUM.            |
| ZEMS-TKN-EMPHASIS   | token-type ZSPI-TYP-BOOLEAN.         |
| ZEMS-TKN-SUBJECT-MARK   | token-type ZSPI-TYP-MARK.            |
| ZBAT-TKN-JOB-NUMBER   | token-type ZSPI-TYP-INT.             |
| ZBAT-TKN-JOB-NAME-ID  | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-USERID   | token-type ZSPI-TYP-USERID.          |
| ZBAT-TKN-EXECUTOR-PROGRAM   | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-START-TIME   | token-type ZSPI-TYP-TIMESTAMP.       |
| ZBAT-TKN-TOTAL-CPU-TIME   | token-type ZSPI-TYP-INT4.            |
| ZBAT-TKN-COMPLETION-CODE  | token-type ZBAT-TYP-COMPLETION-CODE. |
| ZBAT-TKN-TERMINATION-INFO   | token-type ZSPI-TYP-INT.             |
| Conditional Token   |                                      |
| ZBAT-TKN-TEXT   | token-type ZSPI-TYP-STRING.          |
| Event-Message Text  |                                      |
| JOB <i>job-name</i> ( <i>job-number</i> ) has stopped with <i>completion-text</i> . Job started at <i>time</i> , CPU usage <i>number</i> microsecs, program <i>program-file-name</i> , user ID <i>group-ID</i> <i>user-ID</i> . |                                      |

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#). Information on the tokens specific to this message is:

**ZBAT-TKN-COMPLETION-CODE**

is the completion code set by the job's executor-program process when it calls the ABEND, STOP, or PROCESS\_STOP\_ system procedure. This token has one of these values:

| <b>Value</b>  | <b>Description</b>                                     |
|---------------|--|
| ZBAT-ENM-CC-0 | Normal, voluntary termination with no errors           |
| ZBAT-ENM-CC-1 | Normal, voluntary termination with warning diagnostics |

For information on completion codes, see the *Guardian Procedure Calls Reference Manual*.

**Conditional Token**

For information on the conditional token present in the event message, see [Section 4, Common Definitions](#).

**Event-Message Text**

*job-name*

specifies the name of the job.

*job-number*

specifies the number of the job.

*completion-text*

indicates how the job terminated. *completion-text* is one of:

| <b>Value</b>            | <b>Description</b>                                     |
|-------------------------|--|
| Normal termination      | Normal, voluntary termination with no errors           |
| Terminated with warning | Normal, voluntary termination with warning diagnostics |

*time*

specifies when the scheduler started the job's executor-program process.

*number*

specifies the total processor time taken by all processes of the job.

*program-file-name*

specifies the program file name of the job's executor-program process.

*group-ID user-ID*

specifies the user ID of the job owner.

**Cause.** The job stopped without error.

**Effect.** None

**Recovery.** Informational message only; no corrective action required.

## 203 ZBAT-EVT-JOB-ABNORMAL-STOP

|                           |                                      |
|---------------------------|--------------------------------------|
| Unconditional Tokens      |                                      |
| ZSPI-TKN-SSID             | token-type ZSPI-TYP-SSID.            |
| ZEMS-TKN-EVENTNUMBER      | token-type ZSPI-TYP-ENUM.            |
| ZEMS-TKN-EMPHASIS         | token-type ZSPI-TYP-BOOLEAN.         |
| ZEMS-TKN-SUBJECT-MARK     | token-type ZSPI-TYP-MARK.            |
| ZBAT-TKN-JOB-NUMBER       | token-type ZSPI-TYP-INT.             |
| ZBAT-TKN-JOB-NAME-ID      | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-USERID           | token-type ZSPI-TYP-USERID.          |
| ZBAT-TKN-EXECUTOR-PROGRAM | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-START-TIME       | token-type ZSPI-TYP-TIMESTAMP.       |
| ZBAT-TKN-TOTAL-CPU-TIME   | token-type ZSPI-TYP-INT4.            |
| ZBAT-TKN-COMPLETION-CODE  | token-type ZBAT-TYP-COMPLETION-CODE. |
| ZBAT-TKN-TERMINATION-INFO | token-type ZSPI-TYP-INT.             |
| Conditional Token         |                                      |
| ZBAT-TKN-TEXT             | token-type ZSPI-TYP-STRING.          |

Event-Message Text  
 JOB *job-name* (*job-number*) has ABORTED with *completion-text*. Job started at *time*, CPU usage *number* microsecs, program *program-file-name*, user ID *group-ID* *user-ID*.

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#). Information on the tokens specific to this message is:

#### ZBAT-TKN-COMPLETION-CODE

is the completion code set by the job's executor-program process when it calls the ABEND, STOP, or PROCESS\_STOP\_ system procedure. This token can have one of these values:

| Value          | Description  |
|----------------|--|
| ZBAT-ENM-CC-M3 | The process terminated itself, but passed invalid parameters to ABEND, STOP, or PROCESS_DELETE_ (M3 corresponds to completion code -3).  |
| ZBAT-ENM-CC-M2 | The process terminated itself, but the operating system could not pass completion code and termination information to that process because of a resource problem (M2 corresponds to completion code -2). |
| ZBAT-ENM-CC-M1 | Trap detected (M1 corresponds to completion code -1).  |
| ZBAT-ENM-CC-0  | Normal, voluntary termination with no errors.  |
| ZBAT-ENM-CC-1  | Normal, voluntary termination with warning diagnostics.  |
| ZBAT-ENM-CC-2  | Abnormal, voluntary termination with fatal errors or diagnostics.  |

| Value         | Description  |
|---------------|--|
| ZBAT-ENM-CC-3 | Abnormal, voluntary, but premature termination with fatal errors or diagnostics.   |
| ZBAT-ENM-CC-4 | The process did not start.   |
| ZBAT-ENM-CC-5 | The process called ABEND or PROCESS_STOP_.   |
| ZBAT-ENM-CC-6 | <p>An external, authorized process issued an ABEND, STOP, or PROCESS_STOP_ to delete the process. The scheduler includes information about the terminator process in ZBAT-TKN-TEXT in the form:</p> <pre>\node.\$pname (group-ID,user-ID)  \node.\$term.#qual1.qual2 \node.\$pname</pre> <p>is the node and name of the terminator process.</p> <pre>(group-ID,user-ID)</pre> <p>is the group ID and user ID of the owner of the terminator process.</p> <pre>\node.\$term.#qual1.qual2</pre> <p>is the home terminal of the terminator process.</p> |
| ZBAT-ENM-CC-7 | The process sent a restart request to the scheduler.   |

For information on completion codes, see the *Guardian Procedure Calls Reference Manual*.

## Conditional Token

For information on the conditional token present in the event message, see [Section 4, Common Definitions](#).

## Event-Message Text

*job-name*

specifies the name of the job.

*job-number*

specifies the number of the job.

*completion-text*

indicates how the job terminated. *completion-text* is one of:

| <b>Value</b>                                     | <b>Description</b>   |
|--|--|
| -1 TRAP detected                                 | Trap detected.<br>Completion code -1.  |
| Job requests restart                             | The process sent a restart request to the scheduler.<br>Completion code 7.   |
| Normal termination                               | Normal, voluntary termination with no errors.<br>Completion code 0.  |
| Premature termination with fatal errors          | Abnormal, voluntary, but premature termination with fatal errors or diagnostics.<br>Completion code 3.   |
| Process callsabend                               | The process called ABEND or PROCESS_STOP_.<br>Completion code 5.   |
| Process never started                            | The process did not start.<br>Completion code 4.   |
| Process terminated; Guardian unable to pass CC   | The process terminated itself, but the operating system could not pass completion code and termination information to that process because of a resource problem.<br>Completion code -2. |
| Process terminated; Invalid params in STOP/ABEND | The process terminated itself, but passed invalid parameters to ABEND, STOP, or PROCESS_DELETE_.<br>Completion code -3.  |
| STOP/ABEND issued by an external process         | An external, authorized process issued an ABEND, STOP, or PROCESS_STOP_ to delete the process.<br>Completion code 6.   |
| Terminated with fatal errors                     | Abnormal, voluntary termination with fatal errors or diagnostics.<br>Completion code 2.  |
| Terminated with warning                          | Normal, voluntary termination with warning diagnostics.<br>Completion code 1.  |

*time*

specifies when the scheduler started the job's executor-program process.

*number*

specifies the total processor time taken by all processes of the job.

*program-file-name*

specifies the program file name of the job's executor-program process.

*group-ID user-ID*

specifies the user ID of the job owner.

**Cause.** One of these events occurred:

- The job's executor-program process did one of:
  - Abended by calling the system procedure PROCESS\_STOP\_ (with the ABEND option specified) or ABEND
  - Stopped with completion code -3, -2, -1, 2, 3, 4, 5, 6, or 7
- The scheduler stopped the job and all its processes because the job had the attribute STOP-ON-ABEND ON and did one of:
  - Terminated because of processor failure
  - Abended with any completion code
  - Stopped with completion code -3, -2, -1, 1, 2, 3, 4, 5, 6, or 7
- A process external to the scheduler stopped the job's executor-program process.

**Effect.** The job terminated.

**Recovery.** Perform whatever action the job owner requests.

## 204 ZBAT-EVT-JOB-OVER-LIMIT

### Unconditional Tokens

|                           |                              |
|---------------------------|------------------------------|
| ZSPI-TKN-SSID             | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER      | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS         | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK     | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-JOB-NUMBER       | token-type ZSPI-TYP-INT.     |
| ZBAT-TKN-JOB-NAME-ID      | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-USERID           | token-type ZSPI-TYP-USERID.  |
| ZBAT-TKN-EXECUTOR-PROGRAM | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-TIME-LIMIT       | token-type ZSPI-TYP-INT2.    |

### Event-Message Text

JOB *job-name* (*job-number*) ran longer than *limit* minutes Program *program-file-name*, user ID *group-ID user-ID*

## Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

## Event-Message Text

*job-name*

specifies the name of the job.



*job-number*

specifies the number of the job.

*limit*

specifies the job's execution time limit as specified by the LIMIT attribute.

*program-file-name*

specifies the program file name of the job's executor-program process.

*group-ID user-ID*

specifies the user ID of the job owner.

**Cause.** The job exceeded its execution time limit as specified by the LIMIT attribute but continued to run.

**Effect.** The job ran longer than ZBAT-TKN-TIME-LIMIT specified.

**Recovery.** Perform whatever action the job owner requests.

## 301 ZBAT-EVT-JOB-START-ERROR

|   |                                      |
|---|--------------------------------------|
| Unconditional Tokens  |                                      |
| ZSPI-TKN-SSID   | token-type ZSPI-TYP-SSID.            |
| ZEMS-TKN-EVENTNUMBER  | token-type ZSPI-TYP-ENUM.            |
| ZEMS-TKN-EMPHASIS   | token-type ZSPI-TYP-BOOLEAN.         |
| ZEMS-TKN-SUBJECT-MARK   | token-type ZSPI-TYP-MARK.            |
| ZBAT-TKN-JOB-NUMBER   | token-type ZSPI-TYP-INT.             |
| ZBAT-TKN-JOB-NAME-ID  | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-USERID   | token-type ZSPI-TYP-USERID.          |
| ZBAT-TKN-EXECUTOR-PROGRAM   | token-type ZSPI-TYP-STRING.          |
| ZBAT-TKN-COMPLETION-CODE  | token-type ZBAT-TYP-COMPLETION-CODE. |
| ZBAT-TKN-REASON-NUMBER  | token-type ZBAT-TYP-REASON.          |
| Conditional Tokens  |                                      |
| ZBAT-TKN-PC-ERROR0  | token-type ZBAT-TYP-PC-ERROR0.       |
| ZBAT-TKN-PC-ERROR1  | token-type ZBAT-TYP-PC-ERROR1.       |
| ZBAT-TKN-PC-ERROR2  | token-type ZSPI-TYP-INT.             |
| Event-Message Text  |                                      |
| JOB <i>job-name</i> ( <i>job-number</i> ) failed because of <i>error-condition</i> . Program <i>program-file-name</i> , user ID <i>group-ID user-ID</i> . |                                      |

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#). Information on the tokens specific to this message is:

ZBAT-TKN-COMPLETION-CODE

is the completion code set by the job's executor-program process when it calls the ABEND, STOP, or PROCESS\_STOP\_ system procedure. This token has the value ZBAT-ENM-CC-4 (process did not start).

*ZBAT-TKN-REASON-NUMBER*

indicates why the scheduler was unable to start the job's executor-program process. This token has one of these values:

| <b>Value</b>               | <b>Description</b>   |
|----------------------------|--|
| ZBAT-ENM-ATTACHMENTS-ERROR | The scheduler could not load DEFINES from the job's attachment set into the program file space of the executor-program process.  |
| ZBAT-ENM-BAD-OUT-FILE      | The scheduler could not load a DEFINE specified by the job's OUT attribute for one of these reasons: <ul style="list-style-type: none"> <li>● The job has no ATTACHMENT-SET attribute that specifies the DEFINE.</li> <li>● The DEFINE exists but is not a map or spool DEFINE.</li> </ul> |
| ZBAT-ENM-OPEN-FAIL         | The scheduler was unable to open and send the startup message to the executor-program process.   |
| ZBAT-ENM-PROCESS-CREATE    | A PROCESS_CREATE_ system procedure error occurred when the scheduler tried to start the executor-program process.  |
| ZBAT-ENM-PROCESS-NOT-THERE | The scheduler started the executor-program process, which stopped before the scheduler could send the startup message.   |
| ZBAT-ENM-REMOTE-NODE-DOWN  | The scheduler could not start the executor-program process on the specified remote node because the node was not available.  |
| ZBAT-ENM-STARTUP-MSG-FAIL  | The scheduler opened the executor-program process but failed to complete the startup message sequence.   |
| ZBAT-ENM-USER-NOT-FOUND    | The scheduler could not start the executor-program process because the ID of the job's owner is invalid or frozen by the Safeguard program.  |

## Conditional Tokens

For information on the conditional tokens present in the event message, see [Section 4, Common Definitions](#). Information on the tokens specific to this message is:

*ZBAT-TKN-PC-ERROR0*

indicates the error returned to the scheduler by the PROCESS\_CREATE\_ system procedure call. This token has one of these values:

| <b>Value</b>               | <b>Description</b>   |
|----------------------------|--|
| ZBAT-ENM-PCERR-BACCREATEUN | Backup creation specified, but caller unnamed.<br>PROCESS_CREATE_ error 22.  |
| ZBAT-ENM-PCERR-BACSUBTYPE  | Process device subtype specified in the backup process is different than that specified in the primary process.<br>PROCESS_CREATE_ error 21. |
| ZBAT-ENM-PCERR-BADNAME     | Process name error. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 11.  |
| ZBAT-ENM-PCERR-BADPFSSIZE  | Invalid PFS (program file space) size in program file.<br>PROCESS_CREATE_ error 27.  |
| ZBAT-ENM-PCERR-BNERR       | NETBATCH bounds error.<br>PROCESS_CREATE_ error 3.   |
| ZBAT-ENM-PCERR-CONTEXTERR  | DEFINE error. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 24.  |
| ZBAT-ENM-PCERR-ESERR       | File-system error on extended swap file. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 6.                                    |
| ZBAT-ENM-PCERR-FSERR       | File-system error on program file For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 1.   |
| ZBAT-ENM-PCERR-ILLLIB      | Invalid library file format. For details, see ZBAT-TKN-PC-ERROR1.<br>PROCESS_CREATE_ error 13.   |
| ZBAT-ENM-PCERR-ILLPROG     | Invalid program file format. For details, see ZBAT-TKN-PC-ERROR1.<br>PROCESS_CREATE_ error 12.   |
| ZBAT-ENM-PCERR-ILLSUBTYPE  | Program file has an illegal process device subtype.<br>PROCESS_CREATE_ error 20.   |
| ZBAT-ENM-PCERR-ILLTERM     | Invalid home terminal (device does not exist or is wrong device type). For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 8.      |
| ZBAT-ENM-PCERR-LBERR       | File-system error on library file. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 4.  |
| ZBAT-ENM-PCERR-LIBCONF     | Library conflict.<br>PROCESS_CREATE_ error 18.   |
| ZBAT-ENM-PCERR-NOMAP       | Unable to allocate virtual address space.<br>PROCESS_CREATE_ error 16.   |

| <b>Value</b>               | <b>Description</b>   |
|----------------------------|--|
| ZBAT-ENM-PCERR-NOMONITOR   | Unable to communicate with system-monitor process. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 10.         |
| ZBAT-ENM-PCERR-NOPCB       | No process control block available or no PIN less than 255 available.<br>PROCESS_CREATE_ error 15.                           |
| ZBAT-ENM-PCERR-NOTLICENSED | Unlicensed privileged program.<br>PROCESS_CREATE_ error 17.  |
| ZBAT-ENM-PCERR-OK          | No error. Process created or creation initiated with the NOWAIT option.<br>PROCESS_CREATE_ error 0.                          |
| ZBAT-ENM-PCERR-PAERR       | NETBATCH parameter error.<br>PROCESS_CREATE_ error 2.  |
| ZBAT-ENM-PCERR-PFSERR      | File-system error during PFS (program file space) creation. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 7. |
| ZBAT-ENM-PCERR-PROGEQLIB   | Program file and library file are the same file.<br>PROCESS_CREATE_ error 19.  |
| ZBAT-ENM-PCERR-SWERR       | File-system error on swap file. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 5.                             |
| ZBAT-ENM-PCERR-TERMERR     | I/O error to home terminal. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 9.                                 |
| ZBAT-ENM-PCERR-UNDEFEXT    | The process has undefined externals, but the scheduler started it anyway.<br>PROCESS_CREATE_ error 14.                       |
| ZBAT-ENM-PCERR-UNKNOWN-C   | A remote node returned an unrecognized error number. For details, see ZBAT-TKN-PC-ERROR2.<br>PROCESS_CREATE_ error 28.       |

For more information on PROCESS\_CREATE\_ errors, see the *Guardian Procedure Calls Reference Manual*.

#### ZBAT-TKN-PC-ERROR1

indicates the cause of the error for ZBAT-TKN-PC-ERROR0 values ZBAT-ENM-PCERR-ILLLIB and ZBAT-ENM-PCERR-ILLPROG. The token value is one of

| Value                         | Description   |
|-------------------------------|---|
| ZBAT-ENM-BADFILE-DATACODEREF  | The file has data blocks with unresolved references in the program or library file. |
| ZBAT-ENM-BADFILE-FILSYS       | The file does not have the correct file structure.                                  |
| ZBAT-ENM-BADFILE-INITSEGS     | The file header INITSEGS is not consistent with its size.                           |
| ZBAT-ENM-BADFILE-LIBHASMAIN   | The library file has a main procedure.  |
| ZBAT-ENM-BADFILE-MANYSPPACES  | The file has too many code spaces.  |
| ZBAT-ENM-BADFILE-NODATAPAGES  | The program file does not have any data pages.                                      |
| ZBAT-ENM-BADFILE-NOFIXUPS     | The file was not a Binder-prepared file.  |
| ZBAT-ENM-BADFILE-NOMAIN       | The program file does not have a main procedure.                                    |
| ZBAT-ENM-BADFILE-NOT100       | The file does not have a file code of 100.  |
| ZBAT-ENM-BADFILE-NOTDISC      | The file is not a disk file.  |
| ZBAT-ENM-BADFILE-PEPINVALID   | The file has an invalid procedure entry point.                                      |
| ZBAT-ENM-BADFILE-RESIDENTSIZE | The file resident size is greater than the code area length.                        |
| ZBAT-ENM-BADFILE-TOSVERSION   | The file requires a later operating system version.                                 |
| ZBAT-ENM-BADFILE-UNDEFBLOCKS  | The file has undefined data blocks.   |

#### ZBAT-TKN-PC-ERROR2

is a system procedure error number indicating the cause of the error for these ZBAT-TKN-PC-ERROR0 values:

|                           |                          |
|---------------------------|--------------------------|
| ZBAT-ENM-PCERR-BADNAME    | ZBAT-ENM-PCERR-NOMONITOR |
| ZBAT-ENM-PCERR-CONTEXTERR | ZBAT-ENM-PCERR-PFSERR    |
| ZBAT-ENM-PCERR-ESERR      | ZBAT-ENM-PCERR-SWERR     |
| ZBAT-ENM-PCERR-FSERR      | ZBAT-ENM-PCERR-TERMERR   |
| ZBAT-ENM-PCERR-ILLTERM    | ZBAT-ENM-PCERR-UNKNOWN-C |
| ZBAT-ENM-PCERR-LBERR      |                          |

For more information on system procedure errors, see the *Guardian Procedure Errors and Messages Manual*.

## Event-Message Text

*job-name*

specifies the name of the job.

*job-number*

specifies the number of the job.

*error-condition*

indicates why the scheduler could not start the job's executor-program process.  
*error-condition* is one of:

| Value                    | Description  |
|--------------------------|--|
| Attachments error        | The scheduler could not load DEFINES from the job's attachment set into the program file space of the executor-program process.  |
| Bad out file             | The scheduler could not load a DEFINE specified by the job's OUT attribute for one of these reasons: <ul style="list-style-type: none"> <li>● The job has no ATTACHMENT-SET attribute that specifies the DEFINE.</li> <li>● The DEFINE exists but is not a map or spool DEFINE.</li> </ul> |
| Open fail                | The scheduler could not open and send the startup message to the executor-program process.   |
| Process not there        | The scheduler started the executor-program process, which stopped before the scheduler could send the startup message.   |
| PROCESS_CREATE_<br>error | A PROCESS_CREATE_ system procedure error occurred when the scheduler tried to start the executor-program process.  |
| Remote node down         | The scheduler could not start the executor-program process on the specified remote node because the node was not available.  |
| Startup msg fail         | The scheduler opened the executor-program process but failed to complete the startup message sequence.   |
| User not found           | The scheduler could not start the executor-program process because the Guardian ID of the job's owner is invalid or frozen by Safeguard.   |

*program-file-name*

specifies the program file name of the job's executor-program process.

*group-ID user-ID*

specifies the user ID of the job owner.

**Cause.** The scheduler failed to start the job's executor-program process.

**Effect.** The job failed.

**Recovery.** Correct the cause of the error, then perform whatever actions the job owner requests.

## 500 ZBAT-EVT-SCHEDULER-ABENDED

|  |                              |
|--|------------------------------|
| Unconditional Tokens   |                              |
| ZSPI-TKN-SSID  | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER   | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS  | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK  | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-SCHEDULER-ID  | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-LOG-FILE  | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-DATA-BASE   | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-TEXT  | token-type ZSPI-TYP-STRING.  |
| Event-Message Text   |                              |
| SCHEDULER \node.\$process-name has abended because of error-condition. Log log-file-name, database schd-database-subvol. |                              |

For recovery details, see the *NetBatch Management Programming Manual*.

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

### Event-Message Text

*\node.\$process-name*

specifies the node and process name of the scheduler.

*error-condition*

indicates why the scheduler abended.

*log-file-name*

specifies the name of the scheduler's log file.

*schd-database-subvol*

specifies the location of the scheduler's database.

**Cause.** The scheduler called the ABEND or PROCESS\_STOP\_ (with ABEND option) system procedure because of a system resource or scheduler problem.

**Effect.** The scheduler process terminated.

**Recovery.** When the error indicates a system resource problem (for example, disk full), correct the cause of the problem, then warm start the scheduler. If the error indicates a scheduler problem:

1. Check the scheduler's log file for diagnostic messages.

2. Save the scheduler's saveabend, database, and program files.
3. Warm start the scheduler.
4. Report the problem to your HP support representative. Provide the representative with a copy of the scheduler's log, saveabend, database, and program files.

## 501 ZBAT-EVT-LOGON-FAILURE

|   |                              |
|---|------------------------------|
| Unconditional Tokens  |                              |
| ZSPI-TKN-SSID   | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER  | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS   | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK                                       | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-LOGON-ERROR  | token-type ZSPI-TYP-INT.     |
| ZBAT-TKN-TEXT   | token-type ZSPI-TYP-STRING.  |
| Event-Message Text  |                              |
| Logon failure occurred with status = <i>status-number</i> . |                              |

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

### Event-Message Text

*status-number*

specifies the status number returned by USER\_AUTHENTICATE\_.

**Cause.** The scheduler could not log on to the log file owner's user ID.

**Effect.** The logon status is returned, and the scheduler process ultimately abends.

**Recovery.** The status indicates the type of error that occurred. For details on the error, see USER\_AUTHENTICATE\_ in the *Guardian Procedure Calls Reference Manual* and take corrective action accordingly.



## 502 ZBAT-EVT-LOGFILE-CREATE-ERROR

|                       |                              |
|-----------------------|------------------------------|
| Unconditional Tokens  |                              |
| ZSPI-TKN-SSID         | token-type ZSPI-TYP-SSID.    |
| ZEMS-TKN-EVENTNUMBER  | token-type ZSPI-TYP-ENUM.    |
| ZEMS-TKN-EMPHASIS     | token-type ZSPI-TYP-BOOLEAN. |
| ZEMS-TKN-SUBJECT-MARK | token-type ZSPI-TYP-MARK.    |
| ZBAT-TKN-LOG-FILE     | token-type ZSPI-TYP-STRING.  |
| ZBAT-TKN-CREATE-ERROR | token-type ZSPI-TYP-INT.     |
| ZBAT-TKN-TEXT         | token-type ZSPI-TYP-STRING.  |

Event-Message Text  
 Logfile *log-file-name* could not be created due to file system error *err-num*

### Unconditional Tokens

For information on the unconditional tokens present in the event message, see [Section 4, Common Definitions](#).

### Event-Message Text

*log-file-name*

is the name of the scheduler's log file.

*err-num*

is the file-system error that occurred while creating the scheduler log file.

**Cause.** The scheduler could not create the scheduler log file.

**Effect.** A file-system error is returned, and the scheduler process ultimately abends.

**Recovery.** The file-system error indicates why the log file could not be created. For details, see the *Guardian Procedure Calls Reference Manual* and take corrective action accordingly.



# **7** NetBatch Procedure Calls

This section describes the NetBatch procedure call NB^JOB^SUBMIT and contains working C, COBOL, and TAL program examples that use the procedure.

| <b>Topic</b>                            | <b>Page</b>          |
|---|----------------------|
| <a href="#">NB^JOB^SUBMIT Procedure</a> | <a href="#">7-2</a>  |
| <a href="#">Sample Programs</a>         | <a href="#">7-10</a> |

NB^JOB^SUBMIT syntax appears in TAL format, which is the same format used for Guardian procedure calls in the *Guardian Procedure Calls Reference Manual*. For notation conventions, see “Notation Conventions” in that manual.

# NB^JOB^SUBMIT Procedure

NB^JOB^SUBMIT is a TAL procedure defined in the NetBatch library file BATCHLIB. The procedure enables job submission and alteration from user-written programs and has functionality similar to that of the SUBMIT JOB and ALTER JOB commands.

```
{ error := } NB^JOB^SUBMIT ( submit-rec!i
{ CALL      }                ,base-date!i
                                ,masters!i
                                ,purge-test-flag!i
                                ,time-rec!i
                                ,rt-time!o
                                ,alter-flag!i
                                ,open-t!io
                                ,[ nb-job-num ]!io
                                ,[ att-set ]      ) ;!i
```

*error* returned value

INT

returns a number indicating the result of the submit or alter operation. The number can be zero (operation successful) or a NetBatch error number. For details of NetBatch errors, see [Appendix A, Error Numbers and Error Lists](#).

*submit-rec* input

STRUCT:ref:\*

specifies the job's name, scheduler, start time, defaults, and these attributes:

|                  |               |               |
|------------------|---------------|---------------|
| CLASS            | MAXPRINTLINES | STALL         |
| EXECUTOR-PROGRAM | MAXPRINTPAGES | STARTUP       |
| HIGHPIN          | OUT           | STOP-ON-ABEND |
| HOLD             | PRIORITY      | TAPEDRIVES    |
| HOLDAFTER        | RESTART       | WAIT          |
| IN               | SELPRI        |               |

*submit-rec* has the structure:

```

DEF SUBMIT-REC.
03 FILLER                                Type Character 12.
03 ACTUAL-JOB-NAME                      Type Character 24.
03 FILLER                                Type Binary 16.
03 JOB-DETAILS.
05 SCHEDULER                            Type Character 18.
05 JCLASS                                Type Character 24.
05 EXECUTOR-PROGRAM                     Type Character 36.
05 FILLER                                Type Binary 16.
05 PRINT-LINES.
07 B                                     Type Binary 16.
05 PRINT-PAGES.
07 B                                     Type Binary 16.
05 TAPE-DRIVES.
07 B                                     Type Binary 16.
05 SELPRI                                Type Character 1.
05 STALL                                 Type Character 1.
05 PRIORITY.
07 B                                     Type Binary 16.
05 WAIT-TIME.
07 B                                     Type Binary 16.
05 START-TIME.
07 B                                     Type Binary 16.
05 START-UP                             Type Character 40.
05 IN-FILE                              Type Character 36.
05 OUT-FILE                              Type Character 36.
05 DEFAULTS                             Type Character 26.
05 FILLER                                Type Character 1.
05 HOLD-FLAG                             Type Character 1.
05 RESTART-FLAG                         Type Character 1.
05 JOB-TEXT                              Type Character 32.
05 FILLER                                Type Character 1.
05 STOP-ON-ABEND                        Type Character 1.
05 HIGHPIN                              Type Character 1.
05 FILLER                                Type Character 6.
05 HOLD-AFTER-FLAG                      Type Character 1.
END

```

#### ACTUAL-JOB-NAME

is the space-filled name of the job. This field is mandatory.

#### JOB-DETAILS

specifies the job's scheduler, start time, defaults, and attributes (as listed in the introduction to *submit-rec*). Character fields can be all spaces or space-filled. Spaces are not allowed in binary fields. The effect of an all-spaces character field depends on whether *alter-flag* specifies submit or alter:

- If it specifies submit, the job adopts a scheduler-supplied default where available. For example, if JCLASS is all spaces, the job adopts the class specified by the scheduler's DEFAULT-CLASS attribute.
- If it specifies alter, an all-spaces character field has no effect on the job. For example, if JCLASS is all spaces, the job's class is not changed.

**SCHEDULER**

is the name of the job's scheduler in [ \node . ] \$process-name form.  
This field is mandatory.

**JCLASS**

specifies the value (a space-filled class name) of the job's CLASS attribute.  
This field can be all spaces.

**EXECUTOR-PROGRAM**

specifies the value (a space-filled program-file name) of the job's EXECUTOR-PROGRAM attribute. This field can be all spaces.

**PRINT-LINES**

specifies the value of the job's MAXPRINTLINES attribute. The value is a number in the range 120 through 65534, 0 for no maximum, or -1 for the value of the scheduler's DEFAULT-MAXPRINTLINES attribute.

**PRINT-PAGES**

specifies the value of the job's MAXPRINTPAGES attribute. The value is a number in the range 2 through 65534, 0 for no maximum, or -1 for the value of the scheduler's DEFAULT-MAXPRINTPAGES attribute.

**TAPE-DRIVES**

specifies the value of the job's TAPEDRIVES attribute. The value is a number in the range 0 through 99.

**SELPRI**

specifies the value of the job's SELPRI attribute. The value is a number in the range 0 through 7. This field can be all spaces.

**STALL**

is one of these uppercase values specifying the job's STALL attribute. This field can be all spaces.

N STALL OFF  
Y STALL ON

**PRIORITY**

specifies the value of the job's PRI attribute. The value is a number in the range 1 through 199, or 0 for the value of the scheduler's DEFAULT-PRI attribute.

**WAIT-TIME**

specifies the value of the job's WAIT attribute. To specify the wait time, subtract 32768—hexadecimal 8000—from HHMM, where HH specifies 00 through 23 hours and MM specifies 00 through 59 minutes. (For example, 32768 - 1000 specifies a 10-hour wait time.) To specify no wait time, set WAIT-TIME to zero.

NB^JOB^SUBMIT adds WAIT-TIME to *base-date* after *base-date* is modified by START-TIME.

**START-TIME**

specifies the job's start time. To specify the time, subtract 32768—hexadecimal 8000—from HHMM, where HH specifies 00 through 23 hours and MM specifies 00 through 59 minutes. (For example, 32768 - 1000 specifies a 10 a.m. start time.) To specify no start time (that is, accept *base-date*), set START-TIME to zero.

START-TIME replaces the time component of *base-date*.

NB^JOB^SUBMIT adds WAIT-TIME to *base-date* after *base-date* is modified by START-TIME.

**START-UP**

specifies the value (one or more space-filled program parameters) of the job's STARTUP attribute. This field can be all spaces.

**IN-FILE**

specifies the value (a space-filled input-file name) of the job's IN attribute. This field can be all spaces.

**OUT-FILE**

specifies the value (a space-filled output-file name) of the job's OUT attribute. This field can be all spaces.

**DEFAULTS**

specifies a default node, volume, and subvolume (in [ \node. ] \$volume.subvolume form) for use when expanding partial file names. This field is mandatory.

**HOLD-FLAG**

is one of these uppercase values specifying the job's HOLD attribute. This field can be all spaces.

N HOLD OFF

Y HOLD ON

**RESTART-FLAG**

is one of these uppercase values specifying the job's RESTART attribute. This field can be all spaces.

N RESTART OFF  
Y RESTART ON

**JOB-TEXT**

is a space-filled comment such as a job description. The scheduler ignores JOB-TEXT. This field can be all spaces.

**STOP-ON-ABEND**

is one of these uppercase values specifying the job's STOP-ON-ABEND attribute. This field can be all spaces.

N STOP-ON-ABEND OFF  
Y STOP-ON-ABEND ON

**HIGHPIN**

is one of these uppercase values specifying the job's HIGHPIN attribute. This field can be all spaces.

N HIGHPIN OFF  
Y HIGHPIN ON

**HOLD-AFTER-FLAG**

is one of these uppercase values specifying the job's HOLDAFTER attribute. This field can be all spaces.

N HOLDAFTER OFF  
Y HOLDAFTER ON

*base-date*

input

INT:ref:3

is a 48-bit timestamp specifying a reference date and time used to calculate the job's next run time. For the current date and time, specify 65535 (hexadecimal FFFF or HIGH-VALUES).

START-TIME replaces the time component of *base-date*. NB^JOB^SUBMIT adds WAIT-TIME to *base-date* after *base-date* is modified by START-TIME.

*masters*

input

STRING:ref:192

is an 8 x 24-character array that specifies the job's WAITON attribute. To set the attribute, specify the space-filled names of the job's masters. To remove the attribute from the job, specify all spaces.



*purge-test-flag*

input

INT:value

is one of these values specifying the job's PURGE-IN-FILE attribute:

- 1 PURGE-IN-FILE ON
- 0 PURGE-IN-FILE OFF

Setting the value of *purge-test-flag* to 1 disables NB^JOB^SUBMIT's submit and alter functions and returns the job's *rt-time*.

*time-rec*

input

STRUCT:ref:\*

specifies the job's AT, AFTER, CALENDAR, EVERY, and IFFAILS attributes.  
*time-rec* has the structure:

```

DEF TIME-REC.
03 T-TYPE                      Type Binary 16.
03 T-EVERY.
    05 EDAYS                   Type Binary 16.
    05 HHHMM                   Type Binary 16.
    05 FILLER                   Type Character 32.
03 CALENDAR                    Redefines T-EVERY
                                Type Character 36.
END

```

T-TYPE

is one of these values specifying which of the AT, AFTER, CALENDAR, EVERY, and IFFAILS attributes apply to the job. This field is mandatory.

- |                         |                            |
|-------------------------|----------------------------|
| -1 AT                   | 5 AFTER, CALENDAR, IFFAILS |
| 0 AFTER                 | 10 AT, EVERY               |
| 2 AFTER, EVERY          | 11 AT, EVERY, IFFAILS      |
| 3 AFTER, EVERY, IFFAILS | 12 AT, CALENDAR            |
| 4 AFTER, CALENDAR       | 13 AT, CALENDAR, IFFAILS   |

A job with the AT attribute runs at the time specified by *base-date*. If submission of the job fails because the scheduler has the attribute AT-ALLOWED OFF, NB^JOB^SUBMIT resubmits the job with the AFTER attribute.

A job with the AFTER attribute becomes eligible to run after the time specified by *base-date*.

T-EVERY

specifies the EVERY attribute and the value of that attribute. T-EVERY is mandatory if T-TYPE is 2, 3, 10, or 11. The fields are:

## EDAYS

is a number in the range 1 through 365 that specifies the execution interval in days. Zero specifies a null value. EDAYS is mandatory if HHHMM has a null value.

## HHHMM

specifies the execution interval in hours and minutes, where HHH specifies hours in the range 000 through 168 and MM specifies minutes in the range 00 through 59. Zero specifies a null value. HHHMM is mandatory if EDAYS has a null value.

## CALENDAR

specifies the value (a BATCHCAL calendar file name) of the job's CALENDAR attribute. CALENDAR is mandatory if T-TYPE is 4, 5, 12, or 13.

*rt-time*

output

INT:ref:3

is a 48-bit timestamp indicating the job's next run time as calculated by NB^JOB^SUBMIT. *rt-time* reflects *base-date* after *base-date* is modified by START-TIME and WAIT-TIME.

*alter-flag*

input

INT:value

specifies whether NB^JOB^SUBMIT is to submit or alter the job specified by *submit-rec*. A zero value specifies submit and a nonzero value specifies alter.

*open-t*

input:output

INT:ref:41

is an array used by NB^JOB^SUBMIT to hold details of up to eight schedulers that it opens. *open-t* must be initialized to zeroes (LOW-VALUES) by the calling process before the first NB^JOB^SUBMIT call and must not be modified by the process between the first and subsequent NB^JOB^SUBMIT calls. See [Consideration](#) on page 7-9.

*nb-job-num*

input:output

INT:value

is a job number in the range 1 through 9999 returned to the procedure by the scheduler following job submission or alteration. The number identifies the job submitted or altered. The procedure also can use *nb-job-num* to pass a job number to the scheduler when altering a job.

*att-set*

input

STRING:ref:43

specifies the value (an attachment-set ID) of the job's ATTACHMENT-SET attribute. To remove the attribute from the job, specify spaces.

**Consideration**

The first call to NB^JOB^SUBMIT opens the scheduler specified by the SCHEDULER field of JOB-DETAILS in *submit-rec*. The data passed in *open-t* must not be modified after that call so that subsequent calls do not have to reopen the scheduler. *open-t* must be global. For COBOL, *open-t* must be declared in the main program where it is initialized the first time to LOW-VALUES. Other programs that call NB^JOB^SUBMIT must be called with *open-t* in the USING parameter list of the CALL verb.

# Sample Programs

This subsection contains source code for sample C, COBOL, and TAL programs that illustrate use of the NB^JOB^SUBMIT procedure.

## Sample C Program

[Example 7-1](#) on page 7-12 contains the source code for a sample C program that demonstrates the use of NB^JOB^SUBMIT. The program uses NB^JOB^SUBMIT to submit a job to a scheduler, displays some of the job's details, and handles error conditions. All instances of WAIT-TIME or START-TIME must be unsigned numerics. The source code for the program is available in the file NBSPIEX in the NetBatch installation subvolume.

To run the program, you need a D21 or later scheduler named \$ZBAT running on the node where the program will run.

### Step 1: Copy the source code

Copy the source code for the sample C program from file NBSPIEX to a new EDIT file:

```
> EDIT; GET $SYSGEN.ZNETBTCH.NBSPIEX 1834/2121 PUT
TEMP.NBCSRC
TEXT EDITOR - T9601D20 - (01JUN93)
CURRENT FILE IS $DATA7.TEMP.NBCSRC
```

### Step 2: Change the SYSTEM.SYSTEM References

Change SYSTEM.SYSTEM references in the new file to specify the volume and subvolume containing the NetBatch library file BATCHLIB:

```
*LIST BOTH /SYSTEM.SYSTEM.BAT/
1841      #pragma SEARCH "$SYSTEM.SYSTEM.BATCHLIB"
*CHANGE /SYSTEM.SYSTEM.BAT/SYSTEM.SYS00.BAT/ ALL
1841      #pragma SEARCH "$SYSTEM.SYS00.BATCHLIB"
```

### Step 3: Change class name

Change class name CLASS-A to that of an existing class in \$ZBAT if class CLASS-A does not exist in that scheduler. Otherwise, add class CLASS-A to \$ZBAT.

```
*CHANGE /"CLASS-A                                "/"OPERATIONS
"/ ALL
2023      strncpy (submit_rec.job_details.jclass,
"OPERATIONS                                ",24);
```

## Step 4: Change Job Name

Change job name MASTER-A to your own choice of name if MASTER-A conflicts with an existing production job in \$ZBAT. Otherwise, delete job MASTER-A from \$ZBAT.

```
*CHANGE / "MASTER-A                                " / "NBJS-C-JOB
"/ ALL
2014          strncpy (submit_rec.actual_job_name, "NBJS-C-
JOB          ",24
);
```

## Step 5: Compile the Source File

End the EDIT session and compile the source file:

```
*EXIT
> C /IN TEMP.NBCSRC/ TEMP.NBCOBJ; SUPPRESS
```

## Step 6: Test the Program

Test the program by running the compiled object:

```
> RUN TEMP.NBCOBJ
Error 513 encountered.
Job submitted to $ZBAT using NB^JOB^SUBMIT
Job number : 2
Job name   : NBJS-C-JOB
```

## Step 7: Delete the Submitted Job

```
> BATCHCOM $ZBAT; DELETE JOB NBJS-C-JOB
Job NBJS-C-JOB Jobnumber 2 deleted
```

---

**Example 7-1. Sample NB^JOB^SUBMIT C Program**

```
#pragma INSPECT,SYMBOLS
#pragma NOMAP
#pragma NOLMAP
#pragma RUNNABLE
#pragma NOXMEM
/* NOXMEM is mandatory because NB^JOB^SUBMIT cannot access extended memory */
#pragma HEAP 20 pages
#pragma SEARCH "$SYSTEM.SYSTEM.BATCHLIB"

/* OVERVIEW:
 *-----*
 * This is an example program to be included in the SPI
 *
 * Manual to illustrate the programmatic use of NB^JOB^SUBMIT
 *
 * to submit a job to a NetBatch Scheduler.
 *
 *
 * It performs two functions:
 *
 *   a) Submits a job to NetBatch,
 *
 *   b) Interprets any errors returned by the Scheduler
 *
 *   and displays those details to the user.
 *-----*
 */
#include <memoryh>  nolist
#include <stdioh>    nolist
#include <stringh>   nolist
#include <stdlibh>   nolist
#include <talh>      nolist
#include <cextdecs>  nolist

/* START STRUCTURE DEFINITIONS FOR NB^JOB^SUBMIT */

/* Submit record */
struct SUBMIT_REC_DEF
{
    char          filler_0[12];
    char          actual_job_name[24];
    short         filler_1;
    struct
    {
        char      scheduler[18];
        char      jclass[24];
        char      executor_program[36];
        short     filler_2;
        struct
        {
            short  b;
        } print_lines;
    } struct
    (continued)
```

---

```

    {
        short          b;
    } print_pages;
    struct
    {
        short          b;
    } tape_drives;
    char          selpri;
    char          stall;
    struct
    {
        short          b;
    } priority;
    struct
    {
        short          b;
    } wait_time;
    struct
    {
        short          b;
    } start_time;
    char          start_up[40];
    char          in_file[36];
    char          out_file[36];
    char          defaults[26];
    char          filler_3;
    char          hold_flag;
    char          restart_flag;
    char          job_text[32];
    char          filler_4;
    char          stop_on_abend;
    char          highpin;
    char          filler_5[6];
    char          hold_after_flag;
    } job_details;
} submit_rec;
/* Base date */
struct BASE_DATE_DEF
{
    short          after_date[3];
} base_date;
/* Masters */
struct MASTERS_DEF
{
    char          master_jobs[8][24];
} masters;
/* Purge-test flag */
short          purge_test_flag;
/* Time rec */
struct TIME_REC_DEF
(continued)

```

```

{
    short                t_type;
    union
    {
        struct
        {
            short        edays;
            short        hhmm;
            char          filler_0[32];
        } t_every;
        char            calendar[36];
    } u_t_every;
} time_rec;

/* Run time */
struct RT_TIME_DEF
{
    short                run_time[3];
} rt_time;
/* Alter flag */
short                  alter_flag;
/* Open table */
struct OPEN_TABLE_DEF
{
    struct
    {
        short            chan;
        short            vers;
        char             sched[6];
    } table_entry[8];
    short                xsum;
} open_table;

/* Job number */
short                  nb_job_num;

/* Att set */
char                   att_set[43];

/* END STRUCTURE DEFINITIONS FOR NB^JOB^SUBMIT */

#pragma page
/* Interface declarations for TAL procedures */

_tal _variable _alias ("NB^JOB^SUBMIT") short nb_job_submit
(int *,                /* SUBMIT REC */
 int *,                /* BASE DATE */
 char *,               /* MASTERS */
 int,                  /* PURGE-TEST FLAG */
 int *,               /* TIME REC */
(continued)

```



```

        int *,      /* RT TIME          */
        int ,      /* ALTER FLAG        */
        int *,      /* OPEN TABLE       */
        int *,      /* NB JOB NUM        */
        char *      /* ATT SET           */
    );

#pragma page
/*
 *-----
 * nb_init
 * Use:
 *     Data setup
 * Effects:
 *     Set up the job attributes required to submit a job
 *-----
 */
void nb_init(void)
{
    /* Initialize Submit rec */

    strncpy (submit_rec.actual_job_name, "MASTER-A", 24);
    strcpy  (submit_rec.job_details.defaults, getenv("DEFAULTS"));
    strncpy (submit_rec.job_details.executor_program, "$SYSTEM.SYSTEM.TACL", 36);
    submit_rec.job_details.highpin = 'N';
    submit_rec.job_details.hold_after_flag = 'Y';
    submit_rec.job_details.hold_flag = 'Y';
    strcpy  (submit_rec.job_details.in_file, getenv("DEFAULTS"));
    strcat  (submit_rec.job_details.in_file, ".INFILE");
    strncpy (submit_rec.job_details.out_file, "$S.#MASTERA", 36);
    strncpy (submit_rec.job_details.jclass, "CLASS-A", 24);
    submit_rec.job_details.print_lines.b = 0;
    submit_rec.job_details.print_pages.b = 0;
    submit_rec.job_details.tape_drives.b = 0;
    submit_rec.job_details.priority.b = 0;
    submit_rec.job_details.start_time.b = 0;
    submit_rec.job_details.wait_time.b = 0;
    submit_rec.job_details.selpri = ' ';
    submit_rec.job_details.stall = ' ';
    setmem (submit_rec.job_details.start_up, 40, ' ');
    setmem (submit_rec.job_details.job_text, 32, ' ');
    submit_rec.job_details.restart_flag = ' ';
    submit_rec.job_details.stop_on_abend = ' ';
    strncpy (submit_rec.job_details.scheduler, "$ZBAT", 18);

    /* Now initialize the other parameters */

    TIMESTAMP ((short *) &base_date.after_date);
    (continued)

```

```

setmem ((char *) &masters.master_jobs, 192, ' ');
    purge_test_flag = 0;
    time_rec.t_type = 0;
    alter_flag      = 0;

    memset (&open_table.table_entry, 0, 80);

    open_table.xsum = 0;
    nb_job_num      = 0;
}

#pragma page
/*
 *-----
 * nb_submit
 * Use:
 *     Submit the job
 * Effects:
 *     Submits a job to a scheduler
 *     Check for errors and print
 *     Print some job information
 *-----
 */

void nb_submit(void)
{
    short error;

    /* Submit the job */

    error = nb_job_submit ((short *) &submit_rec,
                           (short *) &base_date,
                           (char *)  &masters,
                           (short)   purge_test_flag,
                           (short *) &time_rec,
                           (short *) &rt_time,
                           (short)   alter_flag,
                           (short *) &open_table,
                           (short *) &nb_job_num);

    /* Check for errors and print */

    if (error != 0)
    {
        printf("\n");
        printf("    Error %d encountered.\n");
        printf("\n");
    }
    (continued)

```

```
/* Print job information */

    if (nb_job_num != 0)
    {
        printf("    Job submitted to $ZBAT using NB^JOB^SUBMIT\n\n");
        printf("        Job number : %d\n", nb_job_num);
        printf("        Job name   : %s\n",
submit_rec.actual_job_name);
    }
}

#pragma page
/*
*-----
* main
* Use:
*     Calls all required functions to complete task
* Effects:
*     Initializes the NB^JOB^SUBMIT data structures
*     Submits job
*     Status on job and print details
*-----
*/

main()
{
    nb_init();

    nb_submit();
}
```

## Sample COBOL Program

[Example 7-2](#) on page 7-20 contains the source code for a sample COBOL program that demonstrates the use of NB^JOB^SUBMIT. The program uses NB^JOB^SUBMIT to submit a job to a scheduler, displays some of the job's details, and handles error conditions. The source code for the program is available in the file NBSPIEX in the NetBatch installation subvolume.

To run the sample program, you need a D21 or later scheduler named \$ZBAT running on the node where the program will run.

### Step 1: Copy the Source Code

Copy the source code for the sample COBOL program from file NBSPIEX to a new EDIT file:

```
> EDIT; GET $SYSGEN.ZNETBTCH.NBSPIEX 2129/2395 PUT
TEMP.NBCOBSRC
TEXT EDITOR - T9601D20 - (01JUN93)
CURRENT FILE IS $DATA7.TEMP.NBCOBSRC
```

### Step 2: Change SYSTEM.SYSTEM References

Change SYSTEM.SYSTEM references in the new file to specify the volume and subvolume containing the NetBatch library file BATCHLIB and the COBOL library and external-declaration files COBOLLIB and COBOLEX0:

```
*LIST BOTH /SYSTEM.SYSTEM.BAT/
2131      ?SEARCH $SYSTEM.SYSTEM.BATCHLIB
*CHANGE /SYSTEM.SYSTEM.BAT/SYSTEM.SYS00.BAT/ ALL
2131      ?SEARCH $SYSTEM.SYS00.BATCHLIB
*LIST BOTH /SYSTEM.SYSTEM.COB/
2130      ?SEARCH $SYSTEM.SYSTEM.COBOLLIB
2132      ?CONSULT $SYSTEM.SYSTEM.COBOLEX0
*CHANGE /SYSTEM.SYSTEM.COB/SYSTEM.SYS00.COB/ ALL
```

### Step 3: Change Class Name

Change class name CLASS-A to that of an existing class in \$ZBAT if class CLASS-A does not exist in that scheduler. Otherwise, add class CLASS-A to \$ZBAT.

```
*CHANGE /CLASS-A/OPERATIONS/ ALL
2294      MOVE "OPERATIONS" TO JCLASS.
```

### Step 4: Change Job Name

Change job name MASTER-A to your own choice of name if MASTER-A conflicts with an existing production job in \$ZBAT. Otherwise, delete job MASTER-A from \$ZBAT.

```
*CHANGE /MASTER-A/NBJS-COBOL85-JOB/ ALL
2285      MOVE "NBJS-COBOL85-JOB" TO ACTUAL-JOB-NAME.
```

## Step 5: Compile Source File

End the EDIT session and compile the source file:

```
*EXIT  
> COBOL85 /IN TEMP.NBCOBSRC/ TEMP.NBCOBOBJ; SUPPRESS
```

## Step 6: Test the Program

Test the program by running the compiled object:

```
> RUN TEMP.NBCOBOBJ  
Error 00513 encountered.  
Job submitted to $ZBAT using NB^JOB^SUBMIT  
Job number : 3  
Job name : NBJ5-COBOL85-JOB
```

## Step 7: Delete the Submitted Job

```
> BATCHCOM $ZBAT; DELETE JOB NBJ5-COBOL85-JOB  
Job NBJ5-COBOL85-JOB Jobnumber 3 deleted
```

---

**Example 7-2. Sample NB^JOB^SUBMIT COBOL Program**

```

?ENV COMMON
?SEARCH $SYSTEM.SYSTEM.COBOLLIB
?SEARCH $SYSTEM.SYSTEM.BATCHLIB
?CONSULT $SYSTEM.SYSTEM.COBOLEX0
?SYMBOLS, INSPECT, SAVE STARTUP

IDENTIFICATION DIVISION.
PROGRAM-ID.                                NBCOBSRC.
DATE-WRITTEN.                             November 1993.

* OVERVIEW:
*=====
*
*
* This is an example program to be included in the SPI
*
* Manual to illustrate the programmatic use of NB^JOB^SUBMIT
*
* to submit a job to a NetBatch Scheduler
*
*
*
* It performs two functions:
*
*   a) Submit a job to NetBatch,
*
*   b) Interpret any errors returned by the Scheduler
*
*       and display those details to the user.
*
*
*=====
/
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.

DATA DIVISION.
WORKING-STORAGE SECTION.

01 SUBMIT-REC.
   03 FILLER                                PIC X(12).
   03 ACTUAL-JOB-NAME                       PIC X(24).
   03 FILLER                                NATIVE-2.
   03 JOB-DETAILS.
      05 SCHEDULER                          PIC X(18).
      05 JCLASS                             PIC X(24).
      05 EXECUTOR-PROGRAM                   PIC X(36).
      05 FILLER                             NATIVE-2.
      05 PRINT-LINES.
         07 B                               NATIVE-2.
      05 PRINT-PAGES.
         07 B                               NATIVE-2.
      05 TAPE-DRIVES.
         07 B                               NATIVE-2.
      05 SELPRI                             PIC X(1).
      05 STALL                              PIC X(1).
(continued)

```

---

```

05 PRIORITY.
07 B NATIVE-2.
05 WAIT-TIME.
07 B NATIVE-2.

05 START-TIME.
07 B NATIVE-2.
05 START-UP PIC X(40).
05 IN-FILE PIC X(36).
05 OUT-FILE PIC X(36).
05 DEFAULTS PIC X(26).
05 FILLER PIC X(1).
05 HOLD-FLAG PIC X(1).
05 RESTART-FLAG PIC X(1).
05 JOB-TEXT PIC X(32).
05 FILLER PIC X(1).
05 STOP-ON-ABEND PIC X(1).
05 HIGHPIN PIC X(1).
05 FILLER PIC X(6).
05 HOLD-AFTER-FLAG PIC X(1).

01 BASE-DATE.
03 BD-T1 NATIVE-2.
03 BD-T2 NATIVE-2.
03 BD-T3 NATIVE-2.

01 MASTERS-ARRAY.
03 MASTERS PIC X(24) OCCURS 8 TIMES.

01 PURGE-TEST-FLAG NATIVE-2.

01 TIME-REC.
03 T-TYPE NATIVE-2.
03 T-EVERY.
05 EDAYS NATIVE-2.
05 HHMM NATIVE-2.
05 FILLER PIC X(32).
03 CALENDAR REDEFINES T-EVERY PIC X(36).

01 RT-TIME.
03 RT-T1 NATIVE-2.
03 RT-T2 NATIVE-2.
03 RT-T3 NATIVE-2.

01 ALTER-FLAG NATIVE-2.

01 OPEN-T.
03 X OCCURS 8 TIMES.
05 CHAN NATIVE-2.
05 VERS NATIVE-2.
05 SCHED PIC X(6).
03 XSUM NATIVE-2.

01 NB-JOB-NUM NATIVE-2 VALUE 0.
(continued)

```

```

05 START-TIME.
    07 B
    05 START-UP
    05 IN-FILE
    05 OUT-FILE
    05 DEFAULTS
    05 FILLER
    05 HOLD-FLAG
    05 RESTART-FLAG
    05 JOB-TEXT
    05 FILLER
    05 STOP-ON-ABEND
    05 HIGHPIN
    05 FILLER
    05 HOLD-AFTER-FLAG
    NATIVE-2.
    PIC X(40).
    PIC X(36).
    PIC X(36).
    PIC X(26).
    PIC X(1).
    PIC X(1).
    PIC X(1).
    PIC X(32).
    PIC X(1).
    PIC X(1).
    PIC X(1).
    PIC X(6).
    PIC X(1).

01 BASE-DATE.
    03 BD-T1
    03 BD-T2
    03 BD-T3
    NATIVE-2.
    NATIVE-2.
    NATIVE-2.

01 MASTERS-ARRAY.
    03 MASTERS
    PIC X(24) OCCURS 8 TIMES.

01 PURGE-TEST-FLAG
    NATIVE-2.

01 TIME-REC.
    03 T-TYPE
    03 T-EVERY.
    05 EDAYS
    05 HHMM
    05 FILLER
    03 CALENDAR REDEFINES T-EVERY
    NATIVE-2.
    NATIVE-2.
    PIC X(32).
    PIC X(36).

01 RT-TIME.
    03 RT-T1
    03 RT-T2
    03 RT-T3
    NATIVE-2.
    NATIVE-2.
    NATIVE-2.

01 ALTER-FLAG
    NATIVE-2.

01 OPEN-T.
    03 X OCCURS 8 TIMES.
    05 CHAN
    05 VERS
    05 SCHED
    03 XSUM
    NATIVE-2.
    NATIVE-2.
    PIC X(6).
    NATIVE-2.

01 NB-JOB-NUM
    NATIVE-2 VALUE 0.

(continued)

```



```

/
PROCEDURE DIVISION.

A000-MAINLINE SECTION.

    PERFORM B000-SETUP.

    PERFORM C000-SUBMIT.

    STOP RUN.

A000-EXIT.
EXIT.

/
B000-SETUP SECTION.

*===== *
*
*
*   This section:
*
*   a) Gets the StartUp text from the system to get the
*       default Volume/subvolume,
*
*   b) Sets up the attributes for the job to be submitted.
*
*
*===== *

* Get Startup Text
  ENTER "GETSTARTUPTEXT" USING W02-PORION
                                W02-STARTUP-TEXT
                                GIVING W02-RESULT.

  IF W02-RESULT = -1
    MOVE W02-RESULT TO W02-ERROR-NUM
    MOVE "GETSTARTUPTEXT FAILED" TO W02-ERROR-TEXT
    ENTER TAL "PROCESS_STOP_" USING OMITTED
                                OMITTED
                                OMITTED
                                2
                                W02-ERROR-NUM
                                OMITTED
                                W02-ERROR-TEXT

  END-IF.

(continued)

```

```

* Set up the Submit fields
  MOVE "MASTER-A"          TO ACTUAL-JOB-NAME.
  MOVE W02-STARTUP-TEXT    TO DEFAULTS.
  MOVE "$SYSTEM.SYSTEM.TACL" TO EXECUTOR-PROGRAM.
  MOVE "N"                 TO HIGHPIN.
  MOVE "Y"                 TO HOLD-AFTER-FLAG,
                          HOLD-FLAG.
  STRING W02-STARTUP-TEXT DELIMITED BY SPACES,
        ".INFILE"        DELIMITED BY SIZE
                          INTO IN-FILE.
  MOVE "CLASS-A"           TO JCLASS.
  MOVE "$S.#MASTERA"       TO OUT-FILE.
  MOVE 0                   TO B OF PRINT-LINES,
                          B OF PRINT-PAGES,
                          B OF TAPE-DRIVES.

  MOVE LOW-VALUES          TO PRIORITY,
                          START-TIME,
                          WAIT-TIME.

  MOVE SPACES              TO SELPRI,
                          STALL,
                          START-UP,
                          JOB-TEXT,
                          RESTART-FLAG,
                          STOP-ON-ABEND.

  MOVE "$ZBAT"             TO SCHEDULER.

* Set up the Base date
  MOVE HIGH-VALUES         TO BASE-DATE.

* Set up the Masters array
  MOVE LOW-VALUES          TO MASTERS-ARRAY.

* Set up the Purge/Test flag
  MOVE 0                   TO PURGE-TEST-FLAG.

* Set up Time rec
  MOVE 0                   TO T-TYPE.

* Set up the Alter flag for Submit
  MOVE 0                   TO ALTER-FLAG.

* Set up Open T
  MOVE LOW-VALUES          TO OPEN-T.

* Set up Attachment set
  MOVE SPACES              TO ATT-SET.
  B000-EXIT.
  EXIT.
/
C000-SUBMIT SECTION.

  (continued)

```

```

*=====*
*
*
*   This section:
*
*
*   a) Submits the job to NetBatch,
*   b) Interprets any errors returned by the scheduler.
*
*=====*

* Submit the job.
* Note: RT-TIME and NB-JOB-NUM are returned values for Submit
      ENTER TAL "NB^JOB^SUBMIT" USING SUBMIT-REC
                                   BASE-DATE
                                   MASTERS-ARRAY
                                   PURGE-TEST-FLAG
                                   TIME-REC
                                   RT-TIME
                                   ALTER-FLAG
                                   OPEN-T
                                   NB-JOB-NUM
                                   ATT-SET
                                   GIVING W02-ERROR-NUM.

      IF W02-ERROR-NUM NOT = 0
          DISPLAY " "
          DISPLAY "      Error " W02-ERROR-NUM " encountered."
      END-IF.

      IF NB-JOB-NUM NOT = 0
          PERFORM D000-RESULTS
      END-IF.

C000-EXIT.
EXIT.

/
D000-RESULTS SECTION.

*=====*
*
*
*   This section:
*
*
*   a) Displays the job name and number on the screen if
*       the submit was successful.
*
*=====*

* Display job number and name
      MOVE NB-JOB-NUM TO W02-DISPLAY-NUM.

(continued)

```

```
    DISPLAY " ".  
    DISPLAY "      Job submitted to $ZBAT using NB^JOB^SUBMIT"  
    DISPLAY " ".  
    DISPLAY "                  Job number : " W02-DISPLAY-NUM.  
    DISPLAY "                  Job name   : " ACTUAL-JOB-NAME.  
    DISPLAY " ".  
  
D000-EXIT.  
EXIT.
```

## Sample TAL Program

[Example 7-3](#) on page 7-28 contains the source code for a sample TAL program that demonstrates the use of NB^JOB^SUBMIT. The program uses NB^JOB^SUBMIT to submit a job to a scheduler, displays some of the job's details, and handles error conditions. All instances of WAIT-TIME or START-TIME must be unsigned numerics. The source code for the program is available in the file NBSPIEX in the NetBatch installation subvolume.

To run the sample program, you need a D21 or later scheduler named \$ZBAT running on the node where the program will run.

### Step 1: Copy the Source Code

Copy the source code for the sample TAL program from file NBSPIEX to a new EDIT file:

```
> EDIT; GET NBSPIEX 2403/2756 PUT TEMP.NBTALSRC
TEXT EDITOR - T9601D20 - (01JUN93)
CURRENT FILE IS $DATA7.TEMP.NBTALSRC
```

### Step 2: Change SYSTEM.SYSTEM References

Change SYSTEM.SYSTEM references in the new file to specify the volume and subvolume containing the NetBatch library file BATCHLIB:

```
*LIST BOTH /SYSTEM.SYSTEM.BAT/
2608      ?SEARCH $SYSTEM.SYSTEM.BATCHLIB
*CHANGE /SYSTEM.SYSTEM.BAT/SYSTEM.SYS00.BAT/ ALL
2608      ?SEARCH $SYSTEM.SYS00.BATCHLIB
```

### Step 3: Change Class Name

Change class name CLASS-A to that of an existing class in \$ZBAT if class CLASS-A does not exist in that scheduler. Otherwise, add class CLASS-A to \$ZBAT.

```
*CHANGE /"CLASS-A                                " /"OPERATIONS
"/ ALL
2681      submit^rec.job^details.jclass          ' : = '
"OPERATIONS                                " ;
```

### Step 4: Change Job Name

Change job name MASTER-A to your own choice of name if MASTER-A conflicts with an existing production job in \$ZBAT. Otherwise, delete job MASTER-A from \$ZBAT.

```
*CHANGE /"MASTER-A                                " /"NBJS-TAL-JOB
"/ ALL
2667      submit^rec.actual^job^name              ' : = '
"NBJS-TAL-JOB                                " ;
```

## Step 5: Compile Source File

End the EDIT session and compile the source file:

```
*EXIT
23> TAL /IN TEMP.NBTALSRC/ TEMP.NBTALOBJ; SUPPRESS
```

## Step 6: Test the Program

Test the program by running the compiled object:

```
> RUN TEMP.NBTALOBJ
Error 0513 encountered.
Job submitted to $ZBAT using NB^JOB^SUBMIT
Job number : 0004
Job name   : NBJS-TAL-JOB
```

## Step 7: Delete the Submitted Job

```
> BATCHCOM $ZBAT; DELETE JOB NBJS-TAL-JOB
Job NBJS-TAL-JOB Jobnumber 4 deleted
```

---

### Example 7-3. Sample NB^JOB^SUBMIT TAL Program

```
?NOMAP ,NOLMAP ,NOGMAP ,NOCODE
?INSPECT ,SYMBOLS

! OVERVIEW:
!-----!
! This is an example program to be included in the SPI
! Manual to illustrate the programmatic use of NB^JOB^SUBMIT!
! to submit a job to a NetBatch Scheduler
!
! It performs two functions:
!   a) Submits a job to NetBatch,
!   b) Interprets any errors returned by the Scheduler
!      and displays those details to the user.
!-----!
! STRUCTURE DEFINITIONS USED BY NB^JOB^SUBMIT !
--Startup message
STRUCT .startup^msg;
BEGIN
  INT msgcode;
  STRUCT default;
  BEGIN
    INT vol[0:3];
    INT subvol[0:3];
  END;
STRUCT infile;
BEGIN
  INT vol[0:3];
  INT subvol[0:3];
(continued)
```

---

```

        INT fname[0:3];
    END;
    STRUCT outfile;
    BEGIN
        INT vol[0:3];
        INT subvol[0:3];
        INT fname[0:3];
    END;
    STRING param[0:529];
    END;
    --Submit record
STRUCT .submit^rec;
    BEGIN
        FILLER 12;
        STRUCT actual^job^name;
            BEGIN STRING BYTE [0:23]; END;
        FILLER 2;

        STRUCT job^details;
            BEGIN
                STRUCT scheduler;
                    BEGIN STRING BYTE [0:17]; END;
                STRUCT jclass;
                    BEGIN STRING BYTE [0:23]; END;
                STRUCT executor^program;
                    BEGIN STRING BYTE [0:35]; END;
                FILLER 2;
                STRUCT print^lines;
                    BEGIN
                        INT b;
                    END;
                STRUCT print^pages;
                    BEGIN
                        INT b;
                    END;
                STRUCT tape^drives;
                    BEGIN
                        INT b;
                    END;
                STRING selpri;
                STRING stall;
                STRUCT priority;
                    BEGIN
                        INT b;
                    END;
                STRUCT wait^time;
                    BEGIN
                        INT b;
                    END;

                STRUCT start^time;
                    BEGIN
                        INT b;
                    END;
                STRUCT start^up;
                    BEGIN STRING BYTE [0:39]; END;
            END;
    END;
(continued)

```

```

    STRUCT in^file;
        BEGIN STRING BYTE [0:35]; END;
    STRUCT out^file;
        BEGIN STRING BYTE [0:35]; END;
    STRUCT defaults;
        BEGIN STRING BYTE [0:25]; END;
    FILLER 1;
    STRING hold^flag;
    STRING restart^flag;
    STRUCT job^text;
        BEGIN STRING BYTE [0:31]; END;
    FILLER 1;
    STRING stop^on^abend;
    STRING highpin;
    FILLER 6;

    STRING hold^after^flag;
    END;
END;

STRUCT .base^date;
    BEGIN
        INT after^date[0:2];
    END;

STRUCT .masters;
    BEGIN
        STRUCT master^jobs[0:7];
            BEGIN STRING BYTE [0:23]; END;
    END;

INT purge^test^flag;

STRUCT .time^rec;
    BEGIN
        INT t^type;
        STRUCT t^every;
            BEGIN
                INT edays;
                INT hhmm;
                FILLER 32;
            END;
        STRUCT calendar = t^every;
            BEGIN STRING BYTE [0:35]; END;
    END;

STRUCT .rt^time;
    BEGIN
        INT run^time[0:2];
    END;

INT alter^flag;

STRUCT .open^table;
    BEGIN
        STRUCT table^entry[0:7];

```

(continued)



```

        BEGIN
        INT chan;
        INT vers;
        STRUCT sched;
            BEGIN STRING BYTE [0:5]; END;
        END;
        INT xsum;
        END;

INT nb^job^num;

STRUCT .att^set;
    BEGIN STRING BYTE [0:42]; END;

! END STRUCTURE DEFINITIONS USED BY NB^JOB^SUBMIT !

--String pointers to structures
STRING .open^table^ptr := @open^table '<<' 1;
STRING .masters^ptr := @masters '<<' 1;

--Global declarations
LITERAL maxfilewords = 20;
LITERAL maxfilebytes = 40;

INT      error;
INT      out_chan;
INT      .ptr;
INT      work^int[0:11];
INT      work^length[0:maxfilewords];

STRING   work^area = work^length[1];
STRING   .err^rename[0:30] := "OLDFILENAME_TO_FILENAME_ Failed";
STRING   .err^open[0:16] := "FILE_OPEN_ Failed";
STRING   .text[0:78];
STRING   out^buffer[0:78];
STRING   .ascii^num[0:3];

--Defines
DEFINE stopwitherror(etxt, etxt^len) =
    CALL PROCESS_STOP_ (,,,2,error,,etxt:etxt^len) #;
    --Declaration for NB^JOB^SUBMIT procedure
    INT PROC nb^job^submit (submit^rec, base^date, masters, purge^test^flag,
        time^rec, rt^time, alter^flag, open^table,
nb^job^num,
        att^set) variable;
        INT      .submit^rec,
            .base^date;
        STRING   .masters;
        INT      purge^test^flag,
            .time^rec,
            .rt^time,
            alter^flag,
(continued)

```

```

        .open^table,
        .nb^job^num;
STRING .att^set;

EXTERNAL;

        ?NOLIST

--System procedures library
?SOURCE $SYSTEM.SYSTEM.EXTDECS0

--Location of NB^JOB^SUBMIT
?SEARCH $SYSTEM.SYSTEM.BATCHLIB
?LIST

!-----!
!
!   This proc is invoked by the INITIALIZER system routine.!
!   It:                                     !
!
!   a)  Gets the system Startup message           !
!   b)  Stores the message details in a structure !
!-----!

PROC startup^proc (ruchb, start^data, msg, msg^length, match) VARIABLE;

INT .ruchb,
    .start^data,
    .msg,
    msg^length,
    match;

BEGIN
    startup^msg.msgcode ':= ' msg[0] FOR msg^length/2;
END;

!-----!
!
!   This is the main proc of this program.      !
!   It:                                     !
!
!   a)  Invokes the INITIALIZER system procedure !
!   b)  Opens the output file (terminal)         !
!   c)  Sets up the structures required by NB^JOB^SUBMIT !
!   d)  Submits a job via NB^JOB^SUBMIT          !
!   e)  Interprets any errors returned by the scheduler !
!       and displays them on the screen          !
!   f)  Display the job name and number on the screen !
!   g)  Closes the output file (terminal)        !
!-----!

--Main procedure
PROC submit^job MAIN;
BEGIN
    (continued)

```

```

-- Get the Startup message and process it
CALL INITIALIZER (!ruch!, !start^data!, STARTUP^PROC);
-- Convert C-Series output filename to D-Series format
IF (error := OLDFILENAME_TO_FILENAME_ (startup^msg.outfile.vol,
                                      work^area:maxfilebytes,
                                      work^length) ) THEN

    stopwitherror(err^rename, 31);

-- Open the OUT file
IF (error := FILE_OPEN_ (work^area:work^length, out_chan)) THEN
    stopwitherror(err^open, 17);

-- Initialize SUBMIT^REC
submit^rec.actual^job^name           ':= ' "MASTER-A           ";

work^int ':= ' startup^msg.default FOR 8 WORDS & "           ";
IF (error := OLDFILENAME_TO_FILENAME_ (work^int,
                                      work^area:maxfilebytes,
                                      work^length) ) THEN

    stopwitherror(err^rename, 31);
submit^rec.job^details.defaults      ':= ' work^area FOR work^length
BYTES;
submit^rec.job^details.executor^program ':= '
"$SYSTEM.SYSTEM.TACL                ";
submit^rec.job^details.highpin        ':= ' "N";
submit^rec.job^details.hold^after^flag ':= ' "Y";
submit^rec.job^details.hold^flag      ':= ' "Y";
submit^rec.job^details.in^file        ':= ' work^area FOR work^length
BYTES & ".INFILE";
submit^rec.job^details.out^file        ':= ' "$S.#MASTERA";
submit^rec.job^details.jclass          ':= ' "CLASS-A          ";
submit^rec.job^details.print^lines.b   := 0;
submit^rec.job^details.print^pages.b   := 0;
submit^rec.job^details.tape^drives.b   := 0;
submit^rec.job^details.priority.b      := 0;
submit^rec.job^details.start^time.b    := 0;
submit^rec.job^details.wait^time.b     := 0;
submit^rec.job^details.selpri          ':= ' " ";
submit^rec.job^details.stall           ':= ' " ";
submit^rec.job^details.start^up        ':= ' [40 * [" "]];
submit^rec.job^details.job^text        ':= ' [32 * [" "]];
submit^rec.job^details.restart^flag    ':= ' " ";
submit^rec.job^details.stop^on^abend   ':= ' " ";
submit^rec.job^details.scheduler       ':= ' "$ZBAT           ";

-- Now initialize the other parameters
CALL TIMESTAMP (base^date);
masters^ptr      ':= ' $LEN(masters) * [" "];
purge^test^flag  := 0;
time^rec.t^type  := 0;
(continued)

```

```

alter^flag      := 0;
open^table^ptr  := $LEN(OPEN^table) * [0];
nb^job^num      := 0;

-- Submit the job
error := NB^JOB^SUBMIT (submit^rec,
                        base^date,
                        masters,
                        purge^test^flag,
                        time^rec,
                        rt^time,
                        alter^flag,
                        open^table,
                        nb^job^num);

-- Process any errors and display them
IF error THEN
  BEGIN
    CALL NUMOUT(ascii^num, error, 10, 4);

    out^buffer := " " -> @ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);

    out^buffer := " Error " & ascii^num FOR 4 & " encountered." ->
@ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);
  END;
-- Display job information on the screen
IF nb^job^num <> 0 THEN
  BEGIN
    CALL NUMOUT(ascii^num, nb^job^num, 10, 4);

    out^buffer := " " -> @ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);

    out^buffer := " Job submitted to $ZBAT using NB^JOB^SUBMIT" ->
@ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);

    out^buffer := " " -> @ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);

    out^buffer := " Job number : "
& ascii^num FOR 4 -> @ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);

```

(continued)

```

    out^buffer := " Job name : "
& submit^rec.actual^job^name FOR 24 BYTES ->
@ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);

    out^buffer := " " -> @ptr;
    CALL WRITE (out_chan, out^buffer, @ptr '-' @out^buffer);
  END;

  CALL CLOSE (out_chan);

END;

```

# **A** Error Numbers and Error Lists

This appendix lists NetBatch subsystem error numbers (that is, the values whose symbolic names begin with ZBAT-WRN- and ZBAT-ERR-) and describes the error lists associated with the error numbers. These error numbers can occur as values of the return token ZSPI-TKN-RETCODE and as part of the value of the error token ZSPI-TKN-ERROR.

- For an explanation of error lists and information about retrieving them from the buffer, see the *SPI Programming Manual*.
- For general information on how the NetBatch subsystem handles errors, see [Section 3, SPI Programming Considerations for the NetBatch Subsystem](#).
- For error-handling information on specific commands and their responses, see [Section 5, Commands and Responses](#).

In this appendix, all tokens and their values appear in DDL format. For a quick explanation of DDL as it applies to SPI, see the summary of DDL for SPI in the *SPI Programming Manual*.

## Notation Used

For each error-list description on these pages, a box lists all the tokens that can appear in the error list, including the tokens that start and end all error lists—ZSPI-TKN-ERRLIST and ZSPI-TKN-ENDLIST. Except for ZSPI-TKN-ERRLIST and ZSPI-TKN-ENDLIST, the order of the tokens in the box is not necessarily the order in which they will actually occur.

The notation used in the box for simple tokens is a shorthand version of the essential information given in the DDL TOKEN-CODE statement.

Following the box, there is a description of each token in the error list except ZSPI-TKN-ERRLIST and ZSPI-TKN-ENDLIST. ZSPI-TKN-ERRLIST and ZSPI-TKN-ENDLIST serve the same function in all error lists, so they are not described here. For more information about these tokens, see the *SPI Programming Manual*.

# Error-List Descriptions

The descriptions in this section are in ascending order by error number (that is, in ascending order by ZBAT-WRN- and ZBAT-ERR- values).

## 512 ZBAT-WRN-SEC-BREACH

ZSPI-TKN-ERRLIST

token-type ZSPI-TYP-LIST

ZSPI-TKN-ERROR

token-type ZSPI-TYP-ERROR

ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-SEC-BREACH (512). This token is always present in the error list.

**Cause.** ZBAT-TKN-IN-FILE specified a job input file not secured against write or purge access.

**Effect.** The command executed successfully. Users with write access to the input file can alter any attribute of or delete the job using the file. With the file as a medium, these users also can assume your application's level of security. As a result, they could modify the input file to purge other users' files, change user passwords, and so on.

**Recovery.** Secure the file against write and purge access by using the Safeguard distributed security management facility or the Guardian standard security system. For information on Safeguard security, see the *Safeguard Reference Manual*. For information on Guardian security, see the *Guardian User's Guide*.

## 513 ZBAT-WRN-IN-NE

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-IN-NE (513). This token is always present in the error list.

**Cause.** ZBAT-TKN-IN-FILE specified a nonexistent job input file.

**Effect.** The command executed successfully, but the job will abend when it runs if its executor program requires an input file.

**Recovery.** Not applicable unless the executor program requires an input file. In that case, use the ALTER JOB command to specify an existing input file. Alternatively, create the specified file before the job runs.

## 514 ZBAT-WRN-EXECPROG-NE

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-EXECPROG-NE (514). This token is always present in the error list.

**Cause.** ZBAT-TKN-EXECUTOR-PROGRAM specified a nonexistent program file.

**Effect.** The command executed successfully, but the job fails when it runs if the specified file does not exist at that time. The scheduler puts the job in the SPECIAL-3 state on failure. (The STATUS JOB command indicates a SPECIAL-3 state by returning ZBAT-VAL-NEWPROCESS-ERROR in the ZSPECIAL-REASON field of ZBAT-MAP-STATUS-JOB.)

**Recovery.** Use the ALTER JOB command to specify an existing program file or create the specified file before the job runs.

## 515 ZBAT-WRN-CLASS-INITIATION

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-CLASS-INITIATION (515). This token is always present in the error list.

**Cause.** An ALTER CLASS command omitted the ZINITIATION field of ZBAT-MAP-DEF-CLASS or specified ZINITIATION without a valid Boolean value.

**Effect.** The command failed.

**Recovery.** Set the ZINITIATION field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 516 ZBAT-WRN-EXECUTOR-STARTED

|                           |            |                 |
|---------------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST          | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR            | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST          | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-EXECUTOR-STARTED (516). This token is always present in the error list.

ZBAT-TKN-SEL-EXECUTORNAME

is the name of the executor specified by ZBAT-TKN-SEL-EXECUTORNAME in the START EXECUTOR command.

**Cause.** The START EXECUTOR command specified a started executor instead of a stopped executor.

**Effect.** None

**Recovery.** Not applicable if the command specified the intended executor. If not, change ZBAT-TKN-SEL-EXECUTORNAME to specify the intended executor and retry the command.



## 517 ZBAT-WRN-JOB-EXECUTING

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-JOB-EXECUTING (517). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the ALTER JOB command.

**Cause.** An ALTER JOB command operated on an executing, over-limit, or suspended job.

**Effect.** The command executed successfully, but only these attributes affect the job: HOLDAFTER, IFFAILS, PURGE-IN-FILE, RESTART, STALL, and STOP-ON-ABEND. If the job is recurrent, all altered attributes apply the next time the job runs. (A recurrent job has the CALENDAR or EVERY attribute.)

**Recovery.** Informational message only; no corrective action is needed.

## 518 ZBAT-WRN-WAITON-SATISFIED

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-WAITON-SATISFIED (518). This token is always present in the error list.

**Cause.** The RELEASE JOB command released an already released dependent job.

**Effect.** None

**Recovery.** Not applicable if the command specified the intended job. If not, change the ZJOBNAME field of ZBAT-MAP-PAR-RELEASE-JOB to specify the intended job and retry the command.

## 522 ZBAT-WRN-NOT-NETWORKABLE

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-NOT-NETWORKABLE (522). This token is always present in the error list.

ZBAT-TKN-STRING

is the name of the invalid remote volume.

**Cause.** The remote volume specified in a file name contains eight or more characters (including \$). Remote volume names cannot contain more than seven characters (including \$).

**Effect.** The command failed.

**Recovery.** Specify a valid remote volume and retry the command. An alternative—if the job's executor program is the TACL program—is to omit volume references completely and include the TACL VOLUME command in the job's input file.

## 524 ZBAT-WRN-ALTER-TAPEDRIVES

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-ALTER-TAPEDRIVES (524). This token is always present in the error list.

**Cause.** The ALTER JOB command specified, for an executing, over-limit, or suspended job, more tape drives than were available.

**Effect.** The command executed successfully, but the scheduler assigned only spare tape drives to the job, not the required number. The scheduler reserves the spare drives for the job (thus preventing other jobs from using them), but the drives shortfall remains.

**Recovery.** Use the SUSPEND JOB command to suspend the job if it is executing or over limit. Next, use the STATUS SCHEDULER command to monitor tape drive availability. When the number of drives available equals the drive shortfall, retry the ALTER JOB command, specifying the job's full drive requirement in the ZTAPEDRIVES field of ZBAT-MAP-DEF-JOB.

## 525 ZBAT-WRN-CPU-DOWN

|                           |            |                 |
|---------------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST          | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR            | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST          | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-CPU-DOWN (525). This token is always present in the error list.

#### ZBAT-TKN-SEL-EXECUTORNAME

is the name of the executor specified by ZBAT-TKN-SEL-EXECUTORNAME in the START EXECUTOR command.

**Cause.** The START EXECUTOR command specified an executor whose processor is down.

**Effect.** The command executed successfully, but the executor's state went from OFF to DOWN, rather than to ON.

**Recovery.** Not applicable. The executor's state changes automatically to ON when the processor comes up.

## 526 ZBAT-WRN-EXECUTOR-STOPPED

|                           |            |                 |
|---------------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST          | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR            | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST          | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-EXECUTOR-STOPPED (526). This token is always present in the error list.

ZBAT-TKN-SEL-EXECUTORNAME

is the name of the executor specified by ZBAT-TKN-SEL-EXECUTORNAME in the STOP EXECUTOR command.

**Cause.** The STOP EXECUTOR command specified a stopped executor or an executor in the STOP or DELETE state.

**Effect.** None

**Recovery.** Not applicable if the command specified the intended executor. If not, change ZBAT-TKN-SEL-EXECUTORNAME to specify the intended executor and retry the command.

## 527 ZBAT-WRN-R-ACCESS

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-R-ACCESS (527). This token is always present in the error list.

**Cause.** ZBAT-TKN-IN-FILE specified a job input file secured against read access.

**Effect.** The command executed successfully, but the job will abend when it runs if its executor program requires an input file.

**Recovery.** Not applicable unless the executor program requires an input file. In that case, ask the owner of the file to resecure it for read access before the job runs. Alternatively, use the ALTER JOB command to specify an input file secured for read access.

## 528 ZBAT-WRN-W-ACCESS

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-W-ACCESS (528). This token is always present in the error list.

**Cause.** ZBAT-TKN-OUT-FILE specified a nonexistent job output file or a job output file secured against write access.

**Effect.** The command executed successfully, but the job will abend when it runs if its executor program is incapable of creating or writing to the output file.

**Recovery.** Create the output file or ask the file's owner to resecure it for write access before the job runs.

## 529 ZBAT-WRN-E-ACCESS

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-E-ACCESS (529). This token is always present in the error list.

**Cause.** ZBAT-TKN-EXECUTOR-PROGRAM specified a program file secured against execute access. The job requires execute access to the file to start.

**Effect.** The command executed successfully, but the job will fail when the scheduler tries to start it. The scheduler will put the job in the SPECIAL-3 state on failure. (The STATUS JOB command indicates a SPECIAL-3 state by returning ZBAT-VAL-NEWPROCESS-ERROR in the ZSPECIAL-REASON field of ZBAT-MAP-STATUS-JOB.)

**Recovery.** Ask the owner of the file to resecure it for execute access before the job starts. Alternatively, use the ALTER JOB command to specify a program file secured for execute access.

## 530 ZBAT-WRN-P-ACCESS

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-P-ACCESS (530). This token is always present in the error list.

**Cause.** The ZPURGE-IN-FILE field of ZBAT-MAP-DEF-JOB set the PURGE-IN-FILE ON attribute for a job whose input-file security prevents purge access.

**Effect.** The command executed successfully. However, the scheduler does not purge the job's input file when it deletes the job.

**Recovery.** Not applicable unless you want the scheduler to purge the input file. In that case, ask the owner of the file to resecure it for purge access before the scheduler deletes the job.

## 531 ZBAT-WRN-CALENDAR-ERROR

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-CALENDAR-ERROR (531). This token is always present in the error list.

#### ZBAT-TKN-INT

contains a file-system error number.

**Cause.** The scheduler could not open the file specified by ZBAT-TKN-CALENDAR because of a file-system error.

**Effect.** The command executed successfully, but the scheduler put the job in the SPECIAL-7 state. (The STATUS JOB command indicates a SPECIAL-7 state by returning ZBAT-VAL-CALENDAR-ERROR in the ZSPECIAL-REASON field of ZBAT-MAP-STATUS-JOB.)

**Recovery.** To resolve the SPECIAL-7 state and make the job ready to run:

1. Correct the condition indicated by the file-system error number in ZBAT-TKN-INT. For information about the cause of the error, see the descriptions of file-system errors in the *Guardian Procedure Errors and Messages Manual*.
2. Check that the job has the correct calendar file assigned to it by using the INFO JOB command.
3. Alter the job's HOLD attribute to HOLD OFF by using the ALTER JOB command.



## 532 ZBAT-WRN-CALENDAR-EXPIRED

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-CALENDAR-EXPIRED (532). This token is always present in the error list.

**Cause.** The calendar file specified by ZBAT-TKN-CALENDAR does not contain any future times.

**Effect.** The command executed successfully, but the scheduler put the job in the SPECIAL-8 state. (The STATUS JOB command indicates a SPECIAL-8 state by returning ZBAT-VAL-CALENDAR-EMPTY in the ZSPECIAL-REASON field of ZBAT-MAP-STATUS-JOB.)

**Recovery.** To resolve the SPECIAL-8 state and make the job ready to run:

1. Regenerate the specified calendar file with future times. Alternatively, use the ALTER JOB command to alter ZBAT-TKN-CALENDAR to specify another file containing future times.
2. Alter the job's HOLD attribute to HOLD OFF by using the ALTER JOB command.

## 534 ZBAT-WRN-ATT-DELETED

|                  |                            |
|------------------|----------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-ATT-DELETED (534). This token is always present in the error list.

#### ZBAT-TKN-STRING

is the ID of the deleted attachment set.

**Cause.** The scheduler automatically deleted the attachment set specified by ZBAT-TKN-STRING because the set had the TEMPORARY ON attribute and was not in use by any jobs.

**Effect.** The scheduler deleted the attachment set.

**Recovery.** Informational message only; no corrective action is needed.

## 535 ZBAT-WRN-SECURITY

|                  |                            |
|------------------|----------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-SECURITY (535). This token is always present in the error list.

#### ZBAT-TKN-STRING

is the ID of the attachment set specified by ZBAT-TKN-ATT-SET-ID in the attachment-set command.

**Cause.** ZBAT-TKN-ATT-SET-ID specified an attachment set secured against read access.

**Effect.** The command executed successfully, but the scheduler did not return details of the specified attachment set.

**Recovery.** Not applicable unless you require details of the specified attachment set. In that case, ask the owner of the set to resecure it for read access and retry the command.

## 536 ZBAT-WRN-DEFAULTS-DEFINE

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-DEFAULTS-DEFINE (536). This token is always present in the error list.

**Cause.** ZBAT-TKN-SEL-DEFINE-NAME of the DELETE ATTACHMENT-SET command specified the defaults DEFINE =\_DEFAULTS. =\_DEFAULTS is a permanent DEFINE that cannot be deleted.

**Effect.** The command executed successfully, but the scheduler did not delete DEFINE =\_DEFAULTS.

**Recovery.** Informational message only; no corrective action is needed.

## 540 ZBAT-WRN-RUNNOW-TAPE

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-INT           | token-type | ZSPI-TYP-INT           |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-RUNNOW-TAPE (540). This token is always present in the error list.

#### ZBAT-TKN-INT

is the number of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the RUNNOW JOB command.

#### ZBAT-TKN-NETBATCH-NAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the RUNNOW JOB command.

**Cause.** The RUNNOW JOB command operated on a job whose TAPEDRIVES attribute specified more drives than are available.

**Effect.** The command executed successfully, but the job does not run until the required drives become available.

**Recovery.** Informational message only; no corrective action is needed.

## 542 ZBAT-WRN-DISALLOW-DEFINE

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-DISALLOW-DEFINE (542). This token is always present in the error list.

**Cause.** ZBAT-TKN-SEL-DEFINE-NAME of the attachment-set command specified a DEFINE whose name begins with =\_ZBAT. HP reserves DEFINE names beginning with =\_ZBAT for its own use.

**Effect.** The command executed successfully, but the scheduler rejected the DEFINE.

**Recovery.** Change ZBAT-TKN-SEL-DEFINE-NAME to specify a valid DEFINE name (first character an equals sign (=); second character a letter) and retry the command.

## 544 ZBAT-WRN-SAME-SYSTEM

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-SAME-SYSTEM (544). This token is always present in the error list.

**Cause.** One or more of the EXTSWAP, LIB, and SWAP job attributes specified a file on a node different from that of the executor program. The attributes must specify files on the same node as the executor program.

**Effect.** The command executed successfully, but the executor program fails during startup.

**Recovery.** Change ZBAT-TKN-EXTSWAP-FILE, ZBAT-TKN-LIB-FILE, and ZBAT-TKN-SWAP-FILE to specify files on the same node as the executor program and retry the command.

## 547 ZBAT-WRN-SWITCHCPU-DEFERRED

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-SWITCHCPU-DEFERRED (547). This token is always present in the error list.

**Cause.** A SWITCHCPU SCHEDULER command operated on a scheduler that was sending startup messages to jobs.

**Effect.** The command executed successfully, but the processor switch does not occur until the jobs have read their startup messages. The scheduler does not start more jobs while the processor switch is pending.

**Recovery.** Informational message only; no corrective action is needed.

## 548 ZBAT-WRN-PAST-TIME

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-WRN-PAST-TIME (548). This token is always present in the error list.

**Cause.** A job was submitted or altered to run with an AT or AFTER time in the past.

**Effect.** The job runs.

**Recovery.** Informational message only; no corrective action is needed.

## 549 ZBAT-INF-PHANDLES-OMITTED

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-INF-PHANDLES-OMITTED (549). This token is always present in the error list.

**Cause.** In response to a STATUS command, the *process-handle-count* process handles started by a NetBatch job could not be accommodated in the Reply buffer due to resource limitations.

**Effect.** None. The command completes successfully but displays this informational message.

**Recovery.** To find out the job whose process handles have been omitted:

1. Set the Batchcom command Display-Spi to ON.
2. Issue the STATUS JOB command. From the SPI information, determine the job that returns the informational message.

This is the job for which process handles have been omitted.

3. Set the Batchcom command Display-Spi to OFF.
4. To display the maximum possible process handles for the job, use STATUS JOB *job-number/job-name*, DETAIL.



## 2048 ZBAT-ERR-ACTIVATE-JOB

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZSPI-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ACTIVATE-JOB (2048). This token is always present in the error list.

#### ZBAT-TKN-NETBATCH-NAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the ACTIVATE JOB command.

**Cause.** An ACTIVATE JOB command specified a nonsuspended job. The command operates only on suspended jobs.

**Effect.** None

**Recovery.** Not applicable if the command specified the intended job. If not, change ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER to specify the intended job and retry the command.

## 2050 ZBAT-ERR-AFTER-YEAR

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-YEAR (2050). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZYEAR field of ZDATE in ZBAT-MAP-DEF-JOB specified a year greater than 2525 or less than the current year minus one.

**Effect.** The command failed.

**Recovery.** Change ZYEAR to specify a year less than or equal to 2525 but greater than or equal to the current year minus one and retry the command.

## 2051 ZBAT-ERR-AFTER-MONTH

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-YTP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-MONTH (2051). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMONTH field of ZDATE in ZBAT-MAP-DEF-JOB specified a month value outside the allowable range 1 through 12.

**Effect.** The command failed.

**Recovery.** Change ZMONTH to specify a month value in the range 1 through 12 and retry the command.

## 2052 ZBAT-ERR-AFTER-DAY

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-DAY (2052). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZDAY field of ZDATE in ZBAT-MAP-DEF-JOB specified a day value outside the allowable range 1 through 31 or a value in that range but not applicable to the month or year.

**Effect.** The command failed.

**Recovery.** Change ZDAY to specify a day value in the range 1 through 31 and retry the command.

## 2053 ZBAT-ERR-AFTER-HOUR

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-HOUR (2053). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZHOUR field of ZTIME in ZBAT-MAP-DEF-JOB specified an hour value outside the allowable range 0 through 23.

**Effect.** The command failed.

**Recovery.** Change ZHOUR to specify an hour value in the range 0 through 23 and retry the command.

## 2054 ZBAT-ERR-AFTER-MINUTE

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-MINUTE (2054). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMINUTE field of ZTIME in ZBAT-MAP-DEF-JOB specified a minute value outside the allowable range 0 through 59.

**Effect.** The command failed.

**Recovery.** Change ZMINUTE to specify a minute value in the range 0 through 59 and retry the command.

## 2055 ZBAT-ERR-ALREADY-STARTED

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ALREADY-STARTED (2055). This token is always present in the error list.

**Cause.** The START SCHEDULER command operated on a started scheduler.

**Effect.** None

**Recovery.** Informational message only; no corrective action is needed.

## 2056 ZBAT-ERR-AT

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AT (2056). This token is always present in the error list.

**Cause.** The scheduler has the attribute AT-ALLOWED OFF. This attribute prevents use of the RUNNOW JOB command and submission of jobs with the AT attribute.

**Effect.** The command failed.

**Recovery.** Use the ALTER SCHEDULER command to set the value of the AT-ALLOWED attribute to ON and retry the command that failed. Alternatively, use the RUNNEXT JOB command instead of RUNNOW JOB or submit the job with the AFTER attribute.

## 2066 ZBAT-ERR-CALENDAR

|                  |                            |
|------------------|----------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CALENDAR (2066). This token is always present in the error list.

ZBAT-TKN-STRING

is the invalid file name specified by ZBAT-TKN-CALENDAR.

**Cause.** ZBAT-TKN-CALENDAR specified an invalid file name.

**Effect.** The command failed.

**Recovery.** Specify a valid file name in ZBAT-TKN-CALENDAR and retry the command.

## 2068 ZBAT-ERR-CALENDAR-FILECODE

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CALENDAR-FILECODE (2068). This token is always present in the error list.

**Cause.** ZBAT-TKN-CALENDAR specified a file not generated by BATCHCAL.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-CALENDAR to specify a file generated by BATCHCAL and retry the command.

## 2069 ZBAT-ERR-COLD-START

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-COLD-START (2069). This token is always present in the error list.

**Cause.** File JOB was missing from the scheduler's database during a warm start.

**Effect.** The warm start failed.

**Recovery.** Cold start the scheduler.



## 2071 ZBAT-ERR-CLASS-COUNT

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CLASS-COUNT (2071). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZCLASS-COUNT field of ZBAT-MAP-DEF-EXECUTOR specified less than one class or more than eight classes for an executor.

**Effect.** The command failed.

**Recovery.** Ensure the ZCLASSES fields of ZBAT-MAP-DEF-EXECUTOR specify at least one class, but no more than eight classes. Then change ZCLASS-COUNT to specify the number of classes specified by the fields and retry the command.

## 2073 ZBAT-ERR-CONTEXT

ZSPI-TKN-ERRLIST

token-type ZSPI-TYP-LIST

ZSPI-TKN-ERROR

token-type ZSPI-TYP-ERROR

ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CONTEXT (2073). This token is always present in the error list.

**Cause.** A command changed the value of a context token. Commands must ignore context-token values, sending the tokens back to the servers with the messages that ask for the next responses.

**Effect.** The command failed.

**Recovery.** Change the command to ignore the context-token value and retry the command.

## 2074 ZBAT-ERR-CPU

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CPU (2074). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZCPU field of ZBAT-MAP-DEF-EXECUTOR specified a nonexistent processor on the scheduler's node.

**Effect.** The command failed.

**Recovery.** Change ZCPU to specify a processor configured for the scheduler's node and retry the command.

## 2075 ZBAT-ERR-HOLDAFTER

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-HOLDAFTER (2075). This token is always present in the error list.

**Cause.** The ZHOLD-AFTER field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's HOLDAFTER attribute.

**Effect.** The command failed.

**Recovery.** Set the ZHOLD-AFTER field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2076 ZBAT-ERR-NO-CPU

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NO-CPU (2076). This token is always present in the error list.

**Cause.** The ADD EXECUTOR command did not specify a processor in the ZCPU field of ZBAT-MAP-DEF-EXECUTOR.

**Effect.** The command failed.

**Recovery.** Specify a processor in the ZCPU field and retry the command.

## 2077 ZBAT-ERR-DELETE-JOB

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-DELETE-JOB (2077). This token is always present in the error list.

**Cause.** The DELETE JOB command operated on an executing, over-limit, or suspended job. The command operates only on jobs whose states are EVENT, READY, RUNNEXT, RUNNOW, SPECIAL-*n*, TAPE, or TIME.

**Effect.** The command failed.

**Recovery.** Use the STOP JOB command to stop the job. Alternatively, retry the DELETE JOB command, specifying a job whose state is EVENT, READY, RUNNEXT, RUNNOW, SPECIAL-*n*, TAPE, or TIME.

## 2078 ZBAT-ERR-EVERY-ZERO-MINUTES

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EVERY-ZERO-MINUTES (2078). This token is always present in the error list.

**Cause.** The ZEVERY-DAYS field of ZBAT-MAP-DEF-JOB had a null value, and the ZEVERY-HOURS and ZEVERY-MINUTES fields specified zero hours and zero minutes.

**Effect.** The command failed.

**Recovery.** Set ZEVERY-DAYS to a number in the range 1 through 365 and retry the command. Alternatively, set ZEVERY-HOURS and ZEVERY-MINUTES to specify a value greater than zero hours and zero minutes and retry the command. (Valid values for ZEVERY-HOURS are numbers in the range 0 through 168. Vvalid values for ZEVERY-MINUTES are numbers in the range 0 through 59.)

## 2079 ZBAT-ERR-EVERY

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EVERY (2079). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZEVERY-DAYS, ZEVERY-HOURS, or ZEVERY-MINUTES field of ZBAT-MAP-DEF-JOB specified a value outside the allowable range. The allowable range for ZEVERY-DAYS is 1 through 365, for ZEVERY-HOURS 0 through 168, and for ZEVERY-MINUTES 0 through 59.

**Effect.** The command failed.

**Recovery.** Specify a valid value for ZEVERY-DAYS or valid values for ZEVERY-HOURS and ZEVERY-MINUTES and retry the command.

## 2080 ZBAT-ERR-EVERY-CAL-CRON

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EVERY-CAL-CRON (2080). This token is always present in the error list.

**Cause.** The command specified two or more of the CALENDAR attribute, the EVERY attribute, and ZBAT-MAP-DEF-CRONTAB. CALENDAR, EVERY, and ZBAT-MAP-DEF-CRONTAB are mutually exclusive.

**Effect.** The command failed.

**Recovery.** Specify only one of the CALENDAR attribute, the EVERY attribute, and ZBAT-MAP-DEF-CRONTAB and retry the command.

## 2082 ZBAT-ERR-EXECUTOR-PROG

|                  |                            |
|------------------|----------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EXECUTOR-PROG (2082). This token is always present in the error list.

ZBAT-TKN-STRING

is the invalid file name specified by ZBAT-TKN-EXECUTOR-PROGRAM.

**Cause.** ZBAT-TKN-EXECUTOR-PROGRAM specified an invalid file name.

**Effect.** The command failed.

**Recovery.** Specify a valid file name in ZBAT-TKN-EXECUTOR-PROGRAM and retry the command.

## 2058 ZBAT-ERR-WAITON-SELF

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-WAITON-SELF (2085). This token is always present in the error list.

**Cause.** The ZMASTER field of ZBAT-MAP-DEF-WAITON specified, directly or indirectly, the same job as the dependent job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER. A dependent job cannot depend on itself.

**Effect.** The command failed.

**Recovery.** Change ZMASTER to specify the intended master job and retry the command. Alternatively, remove ZBAT-MAP-DEF-WAITON and retry the command.

## 2086 ZBAT-ERR-EXECUTOR-EXISTS

|                           |                            |
|---------------------------|----------------------------|
| ZSPI-TKN-ERRLIST          | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR            | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST          | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EXECUTOR-EXISTS (2086). This token is always present in the error list.

ZBAT-TKN-SEL-EXECUTORNAME

is the name of the executor specified by ZBAT-TKN-SEL-EXECUTORNAME in the ADD EXECUTOR command.

**Cause.** The ADD EXECUTOR command specified the name of an existing executor.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-EXECUTORNAME to specify a unique executor name and retry the command.



## 2087 ZBAT-ERR-NO-SUCH-EXECUTOR

|                           |            |                 |
|---------------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST          | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR            | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-EXECUTORNAME | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST          | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NO-SUCH-EXECUTOR (2087). This token is always present in the error list.

ZBAT-TKN-SEL-EXECUTORNAME

is the name of the nonexistent executor specified in the command.

**Cause.** ZBAT-TKN-SEL-EXECUTORNAME specified a nonexistent executor.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-EXECUTORNAME to specify an existing executor and retry the command.

## 2090 ZBAT-ERR-EXTRA-TOKEN

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EXTRA-TOKEN (2090). This token is always present in the error list.

**Cause.** A requester sent too many tokens in a command.

**Effect.** The command failed.

**Recovery.** Change the requester to send only allowed tokens.

## 2091 ZBAT-ERR-HOLD

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-HOLD (2091). This token is always present in the error list.

**Cause.** The ZHOLD field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's HOLD attribute.

**Effect.** The command failed.

**Recovery.** Set the ZHOLD field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2092 ZBAT-ERR-IFFAILS

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-IFFAILS (2092). This token is always present in the error list.

**Cause.** The ZIFFAILS field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's IFFAILS attribute.

**Effect.** The command failed.

**Recovery.** Set the ZIFFAILS field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2093 ZBAT-ERR-IN

|                  |                            |
|------------------|----------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-IN (2093). This token is always present in the error list.

ZBAT-TKN-STRING

is the invalid file name specified by ZBAT-TKN-IN-FILE.

**Cause.** ZBAT-TKN-IN-FILE specified an invalid file name.

**Effect.** The command failed.

**Recovery.** Specify a valid file name in ZBAT-TKN-IN-FILE and retry the command.

## 2095 ZBAT-ERR-INITIATION

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-INITIATION (2095). This token is always present in the error list.

**Cause.** The ZINITIATION field of ZBAT-MAP-DEF-CLASS specified an invalid value for the class's INITIATION attribute.

**Effect.** The command failed.

**Recovery.** Set the ZINITIATION field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2096 ZBAT-ERR-USER-UNDEFINED

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-USER-UNDEFINED (2096). This token is always present in the error list.

**Cause.** The owner of the application issuing the command has an invalid user ID.

**Effect.** The command failed.

**Recovery.** Log on with a valid user ID and retry the command.

## 2098 ZBAT-ERR-JOB-FULL

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-JOB-FULL (2098). This token is always present in the error list.

**Cause.** The scheduler attempted to add a job to its database in response to a SUBMIT JOB command. The attempt failed because either the JOB or CHKQUE file that records job details was full. This error occurs when one of the files contains the maximum number of job records (9999).

**Effect.** The command failed.

**Recovery.** Retry the command after deleting unwanted jobs from the database by using the DELETE JOB and STOP JOB commands.

## 2099 ZBAT-ERR-NO-SUCH-JOB

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NO-SUCH-JOB (2099). This token is always present in the error list.

**Cause.** ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER specified a nonexistent job.

**Effect.** The command failed.

**Recovery.** Not applicable if the command specified the intended job. If not, change ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER to specify the intended job and retry the command.

## 2102 ZBAT-ERR-CLASS-EXISTS

|                        |                            |
|------------------------|----------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-CLASSNAME | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CLASS-EXISTS (2102). This token is always present in the error list.

ZBAT-TKN-SEL-CLASSNAME

is the name of the class specified by ZBAT-TKN-SEL-CLASSNAME in the ADD CLASS command.

**Cause.** The ADD CLASS command specified the name of an existing class.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-CLASSNAME to specify a unique class name and retry the command.

## 2104 ZBAT-ERR-CLASS-IN-USE

|                        |            |                 |
|------------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-CLASSNAME | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CLASS-IN-USE (2104). This token is always present in the error list.

#### ZBAT-TKN-SEL-CLASSNAME

is the name of the class specified by ZBAT-TKN-SEL-CLASSNAME in the DELETE CLASS command.

**Cause.** ZBAT-TKN-SEL-CLASSNAME in the DELETE CLASS command specified a class assigned to one or more executors. The command only deletes a class that is not associated with any executors.

**Effect.** The command failed.

**Recovery.** Dissociate the class from its executors by using the ALTER EXECUTOR command, then retry the DELETE CLASS command.

## 2105 ZBAT-ERR-NO-SUCH-CLASS

|                        |                            |
|------------------------|----------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-CLASSNAME | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NO-SUCH-CLASS (2105). This token is always present in the error list.

ZBAT-TKN-SEL-CLASSNAME

is the name of the nonexistent class specified in the command.

**Cause.** ZBAT-TKN-SEL-CLASSNAME specified a nonexistent class.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-CLASSNAME to specify an existing class and retry the command.

## 2106 ZBAT-ERR-JOBNAME

|                        |                                   |
|------------------------|-----------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type ZSPI-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL         |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-JOBNAME (2106). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the invalid job name specified by ZBAT-TKN-SEL-JOBNAME.

**Cause.** ZBAT-TKN-SEL-JOBNAME specified an invalid job name.

**Effect.** The command failed.

**Recovery.** Specify a valid job name in ZBAT-TKN-SEL-JOBNAME and retry the command.

## 2107 ZBAT-ERR-JOBNAME-EXISTS

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZSPI-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-JOBNAME-EXISTS (2107). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME in the SUBMIT JOB command.

**Cause.** The SUBMIT JOB command specified the name of an existing job.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-JOBNAME to specify a unique job name and retry the command.

## 2108 ZBAT-ERR-JOBNAME-REQUIRED

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-JOBNAME-REQUIRED (2108). This token is always present in the error list.

**Cause.** The command did not specify ZBAT-TKN-SEL-JOBNAME, a required syntax item.

**Effect.** The command failed.

**Recovery.** Specify ZBAT-TKN-SEL-JOBNAME and retry the command.



## 2117 ZBAT-ERR-EMPTY-RESPONSE

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EMPTY-RESPONSE (2117). This token is always present in the error list.

**Cause.** The command used wild-card characters to specify a range of attachment set, class, or executor names. No names matched the wild-card specification, or names that matched were of records secured against read access.

**Effect.** None

**Recovery.** Informational message only; no corrective action is needed.

## 2118 ZBAT-ERR-MAXPRINTLINES

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MAXPRINTLINES (2118). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMAXPRINTLINES field of ZBAT-MAP-DEF-JOB specified a maximum number of print lines outside the allowable range 120 through 65534.

**Effect.** The command failed.

**Recovery.** Change ZMAXPRINTLINES to specify a *maximum-print-lines* value in the range 120 through 65534 and retry the command. To specify no maximum, specify a zero value.

## 2119 ZBAT-ERR-MAXPRINTPAGES

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MAXPRINTPAGES (2119). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMAXPRINTPAGES field of ZBAT-MAP-DEF-JOB specified a maximum number of print pages outside the allowable range 2 through 65534.

**Effect.** The command failed.

**Recovery.** Change ZMAXPRINTPAGES to specify a *maximum-print-pages* value in the range 2 through 65534 and retry the command. To specify no maximum, specify a zero value.

## 2120 ZBAT-ERR-MAXRESP

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MAXRESP (2120). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The application sent a maximum-response token with an invalid value.

**Effect.** The command failed.

**Recovery.** Change the application to send the token with a value in the range indicated by ZBAT-TKN-MIN-MAX-ERROR.

## 2121 ZBAT-ERR-MISSING-ATTRIBUTES

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MISSING-ATTRIBUTES (2121). This token is always present in the error list.

**Cause.** The command did not specify the required attributes.

**Effect.** The command failed.

**Recovery.** Specify the required attributes and retry the command.

## 2122 ZBAT-ERR-MISSING-CLASS

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MISSING-CLASS (2122). This token is always present in the error list.

**Cause.** The class command omitted ZBAT-TKN-SEL-CLASSNAME (a required token) or specified ZBAT-TKN-SEL-CLASSNAME without a valid value.

**Effect.** The command failed.

**Recovery.** Set ZBAT-TKN-SEL-CLASSNAME to a valid value and retry the command.

## 2123 ZBAT-ERR-MULTIPLE-CONTEXT

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MULTIPLE-CONTEXT (2123). This token is always present in the error list.

**Cause.** An application sent two or more context tokens in a command that should have contained only one context token.

**Effect.** The command failed.

**Recovery.** Change the application to send one context token.

## 2124 ZBAT-ERR-MULTIPLE-MAPS

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MULTIPLE-MAPS (2124). This token is always present in the error list.

**Cause.** The application sent two or more map tokens in a command that should have contained only one map token.

**Effect.** The command failed.

**Recovery.** Change the application to send one map token.

## 2126 ZBAT-ERR-NAME-AND-NUMBER

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-INT           | token-type | ZSPI-TYP-INT           |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-TYP-NETBATCH-NAME |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NAME-AND-NUMBER (2126). This token is always present in the error list.

ZBAT-TKN-INT

is the number of the job specified in the command by ZBAT-TKN-SEL-JOB-NUMBER.

ZBAT-TKN-NETBATCH-NAME

is the name of the job specified in the command by ZBAT-TKN-SEL-JOBNAME.

ZBAT-TKN-NETBATCH-NAME

is the name of the job that actually corresponds to ZBAT-TKN-SEL-JOB-NUMBER.

**Cause.** ZBAT-TKN-SEL-JOBNAME and ZBAT-TKN-SEL-JOB-NUMBER specified different jobs.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-JOBNAME to specify the actual job that corresponds to ZBAT-TKN-SEL-JOB-NUMBER and retry the command. Alternatively, send only ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER.

## 2127 ZBAT-ERR-NAME-OR-NUMBER

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NAME-OR-NUMBER (2127). This token is always present in the error list.

**Cause.** The job command did not specify at least one of ZBAT-TKN-SEL-JOBNAME and ZBAT-TKN-SEL-JOB-NUMBER.

**Effect.** The command failed.

**Recovery.** Specify ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the command and retry the command.

## 2128 ZBAT-ERR-NO-SUBMIT

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NO-SUBMIT (2128). This token is always present in the error list.

**Cause.** The SUBMIT JOB command submitted a job to a scheduler that has the attribute SUBMIT-ALLOWED OFF. The attribute prevents job submission.

**Effect.** The command failed.

**Recovery.** Use the ALTER SCHEDULER command to alter the scheduler's SUBMIT-ALLOWED attribute to SUBMIT-ALLOWED ON, then retry the SUBMIT JOB command.



## 2129 ZBAT-ERR-INVALID-COMMAND

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-TYP-NETBATCH-NAME |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-INVALID-COMMAND (2129). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the command sent.

ZBAT-TKN-NETBATCH-NAME

is the name of the object that corresponds to the specified command.

**Cause.** The command specified for the named object is invalid.

**Effect.** The command failed.

**Recovery.** Specify a valid command for the object or a valid object for the command and retry the command.

## 2131 ZBAT-ERR-NOT-STARTED

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST.  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR. |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL. |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NOT-STARTED (2131). This token is always present in the error list.

**Cause.** The command operated on a scheduler that has not started.

**Effect.** The command failed.

**Recovery.** Use the START SCHEDULER command to make the scheduler available for use and retry the failed command.

## 2132 ZBAT-ERR-SECURITY

|                        |                                   |
|------------------------|-----------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type ZBAT-TYP-NETBATCH-NAME |
| ZBAT-TKN-STRING        | token-type ZBAT-TYP-STRING        |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL         |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SECURITY (2132). This token is always present in the error list.

#### ZBAT-TKN-NETBATCH-NAME

is the name of the job specified in the command by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER.

#### ZBAT-TKN-STRING

is the ID of the attachment set specified in the command by ZBAT-TKN-ATT-SET-ID.

**Cause.** The command specified an attachment set or job to which the application has no access.

**Effect.** The command failed.

**Recovery.** Informational message only; no corrective action is needed.

## 2133 ZBAT-ERR-SHUTDOWN

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SHUTDOWN (2133). This token is always present in the error list.

**Cause.** The command operated on a scheduler that was shutting down.

**Effect.** The command failed.

**Recovery.** Informational message only; no corrective action is needed.

## 2136 ZBAT-ERR-OUT

|                     |                            |
|---------------------|----------------------------|
| ZSPI-TKN-ERRLIST    | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR      | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-TKN-STRING | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST    | token-type ZSPI-TYP-SSCTL  |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-OUT (2136). This token is always present in the error list.

#### ZBAT-TKN-STRING

is the invalid file name specified by ZBAT-TKN-OUT-FILE.

**Cause.** ZBAT-TKN-OUT-FILE specified an invalid file name.

**Effect.** The command failed.

**Recovery.** Specify a valid file name in ZBAT-TKN-OUT-FILE and retry the command.

## 2137 ZBAT-ERR-PRI

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-PRI (2137). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZPRI field of ZBAT-MAP-DEF-JOB specified an execution priority outside the allowable range 1 through 199.

**Effect.** The command failed.

**Recovery.** Change ZPRI to specify an execution priority in the range 1 through 199 and retry the command.

## 2139 ZBAT-ERR-RESTART

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-RESTART (2139). This token is always present in the error list.

**Cause.** The ZRESTART field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's RESTART attribute.

**Effect.** The command failed.

**Recovery.** Set the ZRESTART field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2140 ZBAT-ERR-STOP-ON-ABEND

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-STOP-ON-ABEND (2140). This token is always present in the error list.

**Cause.** The ZSTOP-ON-ABEND field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's STOP-ON-ABEND attribute.

**Effect.** The command failed.

**Recovery.** Set the ZSTOP-ON-ABEND field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2141 ZBAT-ERR-RUNNEXT

|                      |            |                 |
|----------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST     | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR       | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-INT         | token-type | ZBAT-TYP-INT    |
| ZBAT-TKN-SEL-JOBNAME | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST     | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-RUNNEXT (2141). This token is always present in the error list.

#### ZBAT-TKN-INT

is the number of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the RUNNEXT JOB command.

#### ZBAT-TKN-SEL-JOBNAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the RUNNEXT JOB command.

**Cause.** The RUNNEXT JOB command specified a job whose state was EXECUTING, OVER LIMIT, RUNNEXT, SPECIAL-*n*, or SUSPENDED. The command operates only on jobs in the EVENT, READY, RUNNOW, TAPE, or TIME states.

**Effect.** The command failed.

**Recovery.** Not applicable if the command specified the intended job. If not, change ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER to specify the intended job and retry the command.

## 2142 ZBAT-ERR-RUNNEXT-RUNNOW

|                      |            |                 |
|----------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST     | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR       | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-INT         | token-type | ZBAT-TYP-INT    |
| ZBAT-TKN-SEL-JOBNAME | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST     | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-RUNNEXT-RUNNOW (2142). This token is always present in the error list.

#### ZBAT-TKN-INT

is the number of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the RUNNOW JOB command.

#### ZBAT-TKN-SEL-JOBNAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the RUNNOW JOB command.

**Cause.** The RUNNOW JOB command specified a job whose state was EXECUTING, OVER LIMIT, RUNNOW, SPECIAL-*n*, or SUSPENDED. The command operates only on jobs in the EVENT, READY, RUNNEXT, TAPE, or TIME states.

**Effect.** The command failed.

**Recovery.** Not applicable if the command specified the intended job. If not, change ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER to specify the intended job and retry the command.

## 2143 ZBAT-ERR-SWITCHLOG-EDIT

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SWITCHLOG-EDIT (2143). This token is always present in the error list.

#### ZBAT-TKN-STRING

is the name of the file specified in the SWITCHLOG SCHEDULER command.

**Cause.** The SWITCHLOG SCHEDULER command specified an EDIT file as the scheduler's log file. EDIT files cannot be log files.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-LOG-FILE to specify a valid scheduler log file and retry the command. The log file can be a device; a process; an unstructured, relative, or entry-sequenced disk file that is not an EDIT file; or a nonexistent disk file.



## 2144 ZBAT-ERR-SELPRI

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZBAT-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SELPRI (2144). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZSELPRI field of ZBAT-MAP-DEF-JOB specified a selection priority outside the allowable range 0 through 7.

**Effect.** The command failed.

**Recovery.** Change ZSELPRI to specify a selection priority in the range 0 through 7 and retry the command.

## 2145 ZBAT-ERR-STARTUP-MESSAGE

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-STARTUP-MESSAGE (2145). This token is always present in the error list.

ZBAT-TKN-STRING

contains the value specified by ZBAT-TKN-STARTUP-MESSAGE.

**Cause.** ZBAT-TKN-STARTUP-MESSAGE specified a startup message containing more than 961 characters or omitted the message altogether.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-STARTUP-MESSAGE to specify a valid startup message containing no more than 961 characters and retry the command.

## 2146 ZBAT-ERR-STOP-JOB

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-STOP-JOB (2146). This token is always present in the error list.

**Cause.** The STOP JOB command specified a job whose state was EVENT, READY, RUNNEXT, RUNNOW, SPECIAL-*n*, TAPE, or TIME. The command operates only on executing, over-limit, or suspended jobs.

**Effect.** The command failed.

**Recovery.** Use the DELETE JOB command to delete the job. Alternatively, retry the STOP JOB command, specifying a job whose state is EXECUTING, OVER LIMIT, or SUSPENDED.

## 2148 ZBAT-ERR-SUSPEND-JOB

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SUSPEND-JOB (2148). This token is always present in the error list.

**Cause.** The SUSPEND JOB command specified a job that was not executing or over limit. The command operates only on executing and over-limit jobs.

**Effect.** The command failed.

**Recovery.** Not applicable if the command specified the intended job. If not, change ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER to specify the intended job and retry the command.

## 2149 ZBAT-ERR-TAPEDRIVES

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZBAT-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-TAPEDRIVES (2149). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZTAPEDRIVES field of ZBAT-MAP-DEF-JOB specified a number outside the allowable range 0 through 99.

**Effect.** The command failed.

**Recovery.** Change ZTAPEDRIVES to specify a number in the range 0 through 99 and retry the command.

## 2151 ZBAT-ERR-UNKNOWN-OBJECT

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZBAT-TKN-INT     | token-type | ZBAT-TYP-INT   |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-UNKNOWN-OBJECT (2151). This token is always present in the error list.

ZBAT-TKN-INT

contains the invalid object specified in the command.

**Cause.** ZBAT-OBJ-*object* specified an invalid object. Valid values for *object* are ATT-SET, CLASS, EXECUTOR, JOB, and SCHEDULER.

**Effect.** The command failed.

**Recovery.** Change ZBAT-OBJ-*object* to specify a valid value and retry the command.

## 2153 ZBAT-ERR-UNKNOWN-TOKEN

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZBAT-TKN-INT2    | token-type | ZBAT-TYP-INT2  |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-UNKNOWN-TOKEN (2153). This token is always present in the error list.

ZBAT-TKN-INT2

is the token not recognized by the scheduler.

**Cause.** An application sent a token not recognized by the scheduler.

**Effect.** The command failed.

**Recovery.** Change the application to send the correct token.

## 2154 ZBAT-ERR-VOLUME-REQUIRED

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-VOLUME-REQUIRED (2154). This token is always present in the error list.

**Cause.** The command did not specify the required token ZBAT-TKN-VOLUME-SUBVOL.

**Effect.** The command failed.

**Recovery.** Specify ZBAT-TKN-VOLUME-SUBVOL and retry the command.

## 2155 ZBAT-ERR-VOLUME

|                        |            |                     |
|------------------------|------------|---------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST       |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR      |
| ZBAT-TKN-VOLUME-SUBVOL | token-type | ZBAT-TYP-BYTESTRING |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL      |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-VOLUME (2155). This token is always present in the error list.

ZBAT-TKN-VOLUME-SUBVOL

is the invalid volume and subvolume specified in the command by ZBAT-TKN-VOLUME-SUBVOL.

**Cause.** ZBAT-TKN-VOLUME-SUBVOL specified an invalid volume or subvolume or both.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-VOLUME-SUBVOL to specify a valid volume and subvolume and retry the command.

## 2158 ZBAT-ERR-WAITON-COUNT

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-WAITON-COUNT (2158). This token is always present in the error list.

**Cause.** ZBAT-MAP-DEF-WAITON specified more than eight master jobs. A dependent job can have no more than eight masters.

**Effect.** The command failed.

**Recovery.** Specify no more than eight master jobs and retry the command.

## 2160 ZBAT-ERR-WAITON-JOBS-DUPL

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-WAITON-JOBS-DUPL (2160). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the duplicate job name.

**Cause.** ZBAT-MAP-DEF-WAITON specified duplicate job names.

**Effect.** The command failed.

**Recovery.** Specify unique job names in ZBAT-MAP-DEF-WAITON and retry the command.

## 2167 ZBAT-ERR-SWITCHCPU

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SWITCHCPU (2167). This token is always present in the error list.

**Cause.** The scheduler attempted to switch processors in response to a SWITCHCPU SCHEDULER command. The attempt failed because the scheduler was running without a backup in the only available processor on its node.

**Effect.** The command failed.

**Recovery.** Not applicable. The scheduler automatically creates its backup when another processor becomes available.



## 2168 ZBAT-ERR-LOGFILE

|                  |                            |
|------------------|----------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-INT     | token-type ZBAT-TYP-INT    |
| ZBAT-TKN-STRING  | token-type ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-LOGFILE (2168). This token is always present in the error list.

#### ZBAT-TKN-INT

contains a file-system error number.

#### ZBAT-TKN-STRING

is the name of the file specified in the SWITCHLOG SCHEDULER command.

**Cause.** The scheduler attempted to switch log files in response to a SWITCHLOG SCHEDULER command. The attempt was unsuccessful because of a file-system error.

**Effect.** The command failed.

**Recovery.** Correct the file-system error condition indicated for ZBAT-TKN-STRING by ZBAT-TKN-INT and retry the command. For information on the cause of the error, see the descriptions of file-system errors in the *Guardian Procedure Errors and Messages Manual*.

## 2169 ZBAT-ERR-NOT-C20-FILE

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NOT-C20-FILE (2169). This token is always present in the error list.

ZBAT-TKN-STRING

is the name of the pre-C20 scheduler database file.

**Cause.** The C20 or later version of the scheduler that the application tried to warm start had in its database a file created by a version of the scheduler earlier than C20.

**Effect.** The warm start failed.

**Recovery.** Run the UPDATENB file conversion program supplied with the C20 version of the NetBatch product and retry the warm start. For information about running the program, see the software release document (softdoc) for NetBatch product version C20.

## 2170 ZBAT-ERR-DST

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZBAT-TKN-INT     | token-type ZBAT-TYP-INT   |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-DST (2170). This token is always present in the error list.

#### ZBAT-TKN-INT

contains a CONVERTTIMESTAMP-procedure error number.

**Cause.** The command specified a job run time in a daylight-saving time (DST) transition period, resulting in CONVERTTIMESTAMP-procedure error ZBAT-TKN-INT.

**Effect.** The command failed.

**Recovery.** Specify a run time outside the DST transition period and retry the command. Alternatively, retry the command after correcting the CONVERTTIMESTAMP error condition. For information on the cause of the error, see the description of the CONVERTTIMESTAMP procedure in the *Guardian Procedure Calls Reference Manual*.

## 2171 ZBAT-ERR-ATT-EXISTS

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-EXISTS (2171). This token is always present in the error list.

ZBAT-TKN-STRING

is the attachment-set name specified in the ADD ATTACHMENT-SET command.

**Cause.** The ADD ATTACHMENT-SET command specified the name of an existing attachment set.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-ATT-SET-ID to specify a unique attachment-set name and retry the command.

## 2172 ZBAT-ERR-ATT-DNE

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-DNE (2172). This token is always present in the error list.

ZBAT-TKN-STRING

is the name of the nonexistent attachment set specified in the command.

**Cause.** ZBAT-TKN-ATT-SET-ID specified a nonexistent attachment set.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-ATT-SET-ID to specify an existing attachment set and retry the command.

## 2173 ZBAT-ERR-ATT-JOB

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-JOB (2173). This token is always present in the error list.

ZBAT-TKN-STRING

is the name of the attachment set specified in the command.

**Cause.** The DELETE ATTACHMENT-SET command specified an attachment set in use by one or more jobs.

**Effect.** The command failed.

**Recovery.** Use the ALTER JOB command to dissociate the attachment set from the jobs using it and retry the DELETE ATTACHMENT-SET command. (To list jobs using the set, use the STATUS ATTACHMENT-SET command.)

## 2174 ZBAT-ERR-ATT-REQUESTOR

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-REQUESTOR (2174). This token is always present in the error list.

#### ZBAT-TKN-STRING

is the name of the attachment set whose BATCHCOM creator is still running.

#### ZBAT-TKN-STRING

is the ID of the BATCHCOM process that created the attachment set.

**Cause.** The DELETE ATTACHMENT-SET command specified an attachment set created by a BATCHCOM process that is still running. The scheduler prevents deletion of attachment sets in that circumstance.

**Effect.** The command failed.

**Recovery.** Stop the set's BATCHCOM creator and retry the command.

## 2175 ZBAT-ERR-ATT

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT (2175). This token is always present in the error list.

#### ZBAT-TKN-STRING

is the invalid attachment-set ID specified in the command.

**Cause.** A requester specified an invalid attachment-set ID.

**Effect.** The command failed.

**Recovery.** Change the requester to specify a valid attachment-set ID and retry the command.

## 2177 ZBAT-ERR-ATT-OVERFLOW

|                          |            |                 |
|--------------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST         | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR           | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-ASSIGN-NAME | token-type | ZSPI-TYP-STRING |
| ZBAT-TKN-SEL-PARAM-NAM   | token-type | ZSPI-TYP-STRING |
| ZBAT-TKN-STRING          | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST         | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-OVERFLOW (2177). This token is always present in the error list.

#### ZBAT-TKN-SEL-ASSIGN-NAME

is the name of the ASSIGN that caused the storage overflow.

#### ZBAT-TKN-SEL-PARAM-NAME

is the name of the PARAM that caused the storage overflow.

#### ZBAT-TKN-STRING

is the name of the attachment set specified in the command.

**Cause.** An internal storage overflow momentarily prevented the scheduler from updating the attachment-set record.

**Effect.** The command failed.

**Recovery.** Retry the command.



## 2178 ZBAT-ERR-ATT-UPDATE

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-UPDATE (2178). This token is always present in the error list.

**Cause.** The attachment-set command referred to a set the scheduler was updating in response to another command.

**Effect.** The command failed.

**Recovery.** Retry the command.

## 2188 ZBAT-ERR-INTERNAL-ERROR

|                  |                            |
|------------------|----------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type ZBAT-TYP-STRING |
| ZBAT-TKN-STRING  | token-type ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-INTERNAL-ERROR (2188). This token is always present in the error list.

ZBAT-TKN-STRING

is the contents of the program-counter register.

ZBAT-TKN-STRING

is the contents of the environment register.

**Cause.** The scheduler abended while processing the request.

**Effect.** The scheduler abended.

**Recovery.** Warm start the scheduler, retry the command, and report the error and program-counter and environment register values to your HP representative.

## 2189 ZBAT-ERR-FILE-ERROR

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-INT     | token-type | ZBAT-TYP-INT    |
| ZBAT-TKN-STRING  | token-type | ZBAT-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-FILE-ERROR (2189). This token is always present in the error list.

#### ZBAT-TKN-INT

contains a file-system error number.

#### ZBAT-TKN-STRING

is the name of the file on which the file-system error occurred.

**Cause.** A file-system error on file ZBAT-TKN-STRING prevented command execution.

**Effect.** The command failed.

**Recovery.** Correct the condition indicated by the file-system error number in ZBAT-TKN-INT and retry the command. For information on the cause of the error, see the descriptions of file-system errors in the *Guardian Procedure Errors and Messages Manual*.

## 2191 ZBAT-ERR-NOT-IMPLEMENTED

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NOT-IMPLEMENTED (2191). This token is always present in the error list.

**Cause.** The requested function is not available in your version of NetBatch.

**Effect.** None

**Recovery.** Informational message only; no corrective action is needed.

## 2192 ZBAT-ERR-INVALID-SPI

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZBAT-TKN-INT     | token-type | ZBAT-TYP-INT   |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-INVALID-SPI (2192). This token is always present in the error list.

ZBAT-TKN-INT

contains an SPI error number.

**Cause.** The application sent an invalid SPI request to the scheduler.

**Effect.** The command failed.

**Recovery.** Determine the cause of the SPI error by using the *Guardian Procedure Errors and Messages Manual* and by checking the documentation for SPI procedures SSGET and SSGETTKN in the *SPI Programming Manual*. Change the application and retry the command if the error comes from the application. Report the error to your HP representative if it comes from the scheduler.

## 2193 ZBAT-ERR-NETBATCH-NAME

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NETBATCH-NAME (2193). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the invalid class or executor name specified in the command.

**Cause.** ZBAT-TKN-SEL-CLASSNAME or ZBAT-TKN-SEL-EXECUTORNAME specified an invalid name.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-CLASSNAME or ZBAT-TKN-SEL-EXECUTORNAME to specify a valid name and retry the command.

## 2194 ZBAT-ERR-SUSPEND

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SUSPEND (2194). This token is always present in the error list.

#### ZBAT-TKN-NETBATCH-NAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the SUSPEND JOB command.

**Cause.** One of these file-system errors occurred when the scheduler tried to suspend job ZBAT-TKN-NETBATCH-NAME in response to a SUSPEND JOB command:

| File-System Error Number | Description  |
|--------------------------|--|
| 11                       | Process does not exist                             |
| 48                       | Security violation                                 |
| 201                      | Unable to communicate with the process's processor |

**Effect.** The command failed.

**Recovery.** Use the job's log file to determine which file-system error occurred, correct the error condition, and retry the command. For information on the cause of the error, see the descriptions of file-system errors in the *Guardian Procedure Errors and Messages Manual*.

## 2195 ZBAT-ERR-ACTIVATE

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZBAT-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ACTIVATE (2195). This token is always present in the error list.

#### ZBAT-TKN-NETBATCH-NAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the ACTIVATE JOB command.

**Cause.** One of these file-system errors occurred when the scheduler tried to reactivate job ZBAT-TKN-NETBATCH-NAME in response to an ACTIVATE JOB command:

| File-System Error Number | Description  |
|--------------------------|--|
| 11                       | Process does not exist                             |
| 48                       | Security violation                                 |
| 201                      | Unable to communicate with the process's processor |

**Effect.** The command failed.

**Recovery.** Use the job's log file to determine which file-system error occurred, correct the error condition, and retry the command. For information on the cause of the error, see the descriptions of file-system errors in the *Guardian Procedure Errors and Messages Manual*.

## 2196 ZBAT-ERR-STOP

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-NETBATCH-NAME | token-type ZBAT-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-STOP (2196). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the name of the job specified by ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER in the STOP JOB command.

**Cause.** A file-system error occurred when the scheduler tried to stop job ZBAT-TKN-NETBATCH-NAME in response to a STOP JOB command

**Effect.** The command failed.

**Recovery.** Use the job's log file to determine which file-system error occurred, correct the error condition, and retry the command. For information on the cause of the error, see the descriptions of file-system errors in the *Guardian Procedure Errors and Messages Manual*.

## 2197 ZBAT-ERR-STALL

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-STALL (2197). This token is always present in the error list.

**Cause.** The ZSTALL field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's STALL attribute.

**Effect.** The command failed.

**Recovery.** Set the ZSTALL field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2198 ZBAT-ERR-WILDCARD

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-WILDCARD (2198). This token is always present in the error list.

**Cause.** The RUNNEXT JOB or RUNNOW JOB command used wild-card characters to specify a range of job names. These commands do not support wild-card character searching in your version of the NetBatch product.

**Effect.** The command failed.

**Recovery.** Change ZBAT-TKN-SEL-JOBNAME or ZBAT-TKN-SEL-JOB-NUMBER to specify the full name or number of a single job and retry the command.

## 2199 ZBAT-ERR-JOB-TOO-MANY-ATT

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-JOB-TOO-MANY-ATT (2199). This token is always present in the error list.

**Cause.** The command specified more than three attachment sets for the job. Your NetBatch scheduler allows only three attachment sets per job.

**Effect.** The command failed.

**Recovery.** Specify no more than three attachment sets for the job and retry the command.



## 2200 ZBAT-ERR-DATE

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-DATE (2200). This token is always present in the error list.

**Cause.** The ZDATE field of ZBAT-MAP-DEF-JOB specified values for some but not all of ZYEAR, ZMONTH, and ZDAY. The field must specify values for each of ZYEAR, ZMONTH, and ZDAY.

**Effect.** The command failed.

**Recovery.** Change ZDATE to specify values for each of ZYEAR, ZMONTH, and ZDAY and retry the command.

## 2201 ZBAT-ERR-TIME

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-TIME (2201). This token is always present in the error list.

**Cause.** The ZTIME field of ZBAT-MAP-DEF-JOB specified values for some but not all of ZHOUR, ZMINUTE, ZSECOND, ZMILLISECOND, and ZMICROSECOND. The field must specify values for each of ZHOUR, ZMINUTE, ZSECOND, ZMILLISECOND, and ZMICROSECOND.

**Effect.** The command failed.

**Recovery.** Change ZTIME to specify values for each of ZHOUR, ZMINUTE, ZSECOND, ZMILLISECOND, and ZMICROSECOND and retry the command.

## 2202 ZBAT-ERR-AT-FLAG

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AT-FLAG (2202). This token is always present in the error list.

**Cause.** The ZAT-FLAG field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's AT flag.

**Effect.** The command failed.

**Recovery.** Set the ZAT-FLAG field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2203 ZBAT-ERR-MISSING-EXECUTOR

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MISSING-EXECUTOR (2203). This token is always present in the error list.

**Cause.** The executor command omitted ZBAT-TKN-SEL-EXECUTORNAME (a required token) or specified ZBAT-TKN-SEL-EXECUTORNAME without a valid value.

**Effect.** The command failed.

**Recovery.** Set ZBAT-TKN-SEL-EXECUTORNAME to a valid value and retry the command.

## 2204 ZBAT-ERR-MISSING-ATT-ID

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MISSING-ATT-ID (2204). This token is always present in the error list.

**Cause.** The attachment-set command omitted ZBAT-TKN-ATT-SET-ID (a required token) or specified ZBAT-TKN-ATT-SET-ID without a valid value.

**Effect.** The command failed.

**Recovery.** Set ZBAT-TKN-ATT-SET-ID to a valid value and retry the command.

## 2205 ZBAT-ERR-MISSING-RELEASE-MAP

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MISSING-RELEASE-MAP (2205). This token is always present in the error list.

**Cause.** The RELEASE JOB command omitted ZBAT-MAP-PAR-RELEASE-JOB (a required token) or specified ZBAT-MAP-PAR-RELEASE-JOB without valid values.

**Effect.** The command failed.

**Recovery.** Specify ZBAT-MAP-PAR-RELEASE-JOB with valid values and retry the command.

## 2206 ZBAT-ERR-ATT-ASSIGN

|                          |                            |
|--------------------------|----------------------------|
| ZSPI-TKN-ERRLIST         | token-type ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR           | token-type ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-ASSIGN-NAME | token-type ZSPI-TYP-STRING |
| ZBAT-TKN-STRING          | token-type ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST         | token-type ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-ASSIGN (2206). This token is always present in the error list.

ZBAT-TKN-SEL-ASSIGN-NAME

is the invalid ASSIGN name.

ZBAT-TKN-STRING

is the ID of the attachment set specified in the command.

**Cause.** ZBAT-TKN-SEL-ASSIGN-NAME specified an invalid ASSIGN name.

**Effect.** The command failed.

**Recovery.** Specify a valid ASSIGN name in ZBAT-TKN-SEL-ASSIGN-NAME and retry the command.

## 2207 ZBAT-ERR-ATT-DEFINE

|                       |            |                       |
|-----------------------|------------|-----------------------|
| ZSPI-TKN-ERRLIST      | token-type | ZSPI-TYP-LIST         |
| ZSPI-TKN-ERROR        | token-type | ZSPI-TYP-ERROR        |
| ZBAT-MAP-DEFINE-ERROR | token-type | ZBAT-DDL-DEFINE-ERROR |
| ZBAT-TKN-STRING       | token-type | ZSPI-TYP-STRING       |
| ZSPI-TKN-ENDLIST      | token-type | ZSPI-TYP-SSCTL        |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-DEFINE (2207). This token is always present in the error list.

#### ZBAT-MAP-DEFINE-ERROR

is an extensible structured token containing details of the error detected by the scheduler when the scheduler validated ZBAT-TKN-ATT-SET-DEFINE. For information on the structure of ZBAT-MAP-DEFINE-ERROR and descriptions of its fields, see [Section 4, Common Definitions](#).

#### ZBAT-TKN-STRING

is the ID of the attachment set specified in the command.

**Cause.** A DEFINE error occurred.

**Effect.** The command failed.

**Recovery.** Correct the condition indicated by the DEFINE error number in ZBAT-MAP-DEFINE-ERROR and retry the command. For information on the cause of the error, see the descriptions of DEFINE errors in the *Guardian Procedure Errors and Messages Manual*.

## 2208 ZBAT-ERR-ATT-PARAM

|                         |            |                 |
|-------------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST        | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR          | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-SEL-PARAM-NAME | token-type | ZSPI-TYP-STRING |
| ZBAT-TKN-STRING         | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST        | token-type | ZSPI-TYP-SSCTL  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-ATT-PARAM (2208). This token is always present in the error list.

ZBAT-TKN-SEL-PARAM-NAME

is the invalid PARAM name.

ZBAT-TKN-STRING

is the ID of the attachment set specified in the command.

**Cause.** ZBAT-TKN-SEL-PARAM-NAME specified an invalid PARAM name.

**Effect.** The command failed.

**Recovery.** Specify a valid PARAM name in ZBAT-TKN-SEL-PARAM-NAME and retry the command.

## 2209 ZBAT-ERR-JOB-DUPL-ATT

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

ZZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-JOB-DUPL-ATT (2209). This token is always present in the error list.

**Cause.** The command specified a duplicate attachment-set name.

**Effect.** The command failed.

**Recovery.** Remove the duplicate attachment-set name and retry the command.

## 2210 ZBAT-ERR-AFTER-SECOND

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-SECOND (2210). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZSECOND field of ZTIME in ZBAT-MAP-DEF-JOB specified a seconds value outside the allowable range 0 through 59.

**Effect.** The command failed.

**Recovery.** Change ZSECOND to specify a seconds value in the range 0 through 59 and retry the command.

## 2211 ZBAT-ERR-AFTER-MILLISEC

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-MILLISEC (2211). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMILLISECOND field of ZTIME in ZBAT-MAP-DEF-JOB specified a milliseconds value outside the allowable range 0 through 999.

**Effect.** The command failed.

**Recovery.** Change ZMILLISECOND to specify a milliseconds value in the range 0 through 999 and retry the command.



## 2212 ZBAT-ERR-AFTER-MICROSEC

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-AFTER-MICROSEC (2212). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMICROSECOND field of ZTIME in ZBAT-MAP-DEF-JOB specified a microseconds value outside the allowable range 0 through 999.

**Effect.** The command failed.

**Recovery.** Change ZMICROSECOND to specify a microseconds value in the range 0 through 999 and retry the command.

## 2213 ZBAT-ERR-CLASS-NAME

|                        |                             |
|------------------------|-----------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST.   |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR.  |
| ZBAT-TKN-SEL-CLASSNAME | token-type ZSPI-TYP-STRING. |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL.  |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CLASS-NAME (2213). This token is always present in the error list.

ZBAT-TKN-SEL-CLASSNAME

is the invalid class name.

**Cause.** ZBAT-TKN-SEL-CLASSNAME specified an invalid class name.

**Effect.** The command failed.

**Recovery.** Specify a valid class name in ZBAT-TKN-SEL-CLASSNAME and retry the command.

## 2214 ZBAT-ERR-WAITON-ID

|                        |            |                        |
|------------------------|------------|------------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type | ZSPI-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL         |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-WAITON-ID (2214). This token is always present in the error list.

#### ZBAT-TKN-NETBATCH-NAME

is the invalid job name specified by the ZMASTER field of ZBAT-MAP-DEF-WAITON.

**Cause.** The ZMASTER field of ZBAT-MAP-DEF-WAITON specified an invalid job name.

**Effect.** The command failed.

**Recovery.** Change the ZMASTER field to specify a valid job name and retry the command.

## 2215 ZBAT-ERR-EXECUTOR-NAME

|                        |                                   |
|------------------------|-----------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST          |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR         |
| ZBAT-TKN-NETBATCH-NAME | token-type ZSPI-TYP-NETBATCH-NAME |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL         |

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EXECUTOR-NAME (2215). This token is always present in the error list.

ZBAT-TKN-NETBATCH-NAME

is the invalid executor name.

**Cause.** ZBAT-TKN-SEL-EXECUTORNAME specified an invalid executor name.

**Effect.** The command failed.

**Recovery.** Specify a valid executor name in ZBAT-TKN-SEL-EXECUTORNAME and retry the command.

## 2216 ZBAT-ERR-CLASS-INITIATION

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CLASS-INITIATION (2216). This token is always present in the error list.

**Cause.** The ALTER CLASS command did not specify the INITIATION attribute.

**Effect.** The command failed.

**Recovery.** Change the command to specify the INITIATION attribute and retry the command.

## 2217 ZBAT-ERR-VAR-BUF-FULL

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-VAR-BUF-FULL (2217). This token is always present in the error list.

**Cause.** The accumulated size of these items exceeds the scheduler's internal-storage capacity:

|                           |                                 |
|---------------------------|---------------------------------|
| ZBAT-TKN-DESCRIPTION      | ZBAT-TKN-OUT-FILE               |
| ZBAT-TKN-EXECUTOR-PROGRAM | ZBAT-TKN-STARTUP-MESSAGE        |
| ZBAT-TKN-EXTSWAP-FILE     | ZBAT-TKN-SWAP-FILE              |
| ZBAT-TKN-IN-FILE          | ZBAT-TKN-TERM-FILE              |
| ZBAT-TKN-LIB-FILE         | ZBAT-TKN-VOLUME-SUBVOL          |
| ZBAT-TKN-LOG-FILE         | ZNAME field of ZBAT-MAP-DEF-JOB |

**Effect.** The command failed.

**Recovery.** Reduce the size of or delete ZBAT-TKN-DESCRIPTION and retry the command. If the error recurs, reduce the size of or delete one by one and in order of increasing importance other items listed in the cause of this message (starting with ZBAT-TKN-STARTUP-MESSAGE) and retry the command.

## 2218 ZBAT-ERR-CRONTAB

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-CRONTAB (2218). This token is always present in the error list.

**Cause.** One or more ZBAT-MAP-DEF-CRONTAB fields specified an invalid value.

**Effect.** The command failed.

**Recovery.** Specify valid values for all ZBAT-MAP-DEF-CRONTAB fields and retry the command.

## 2219 ZBAT-ERR-PURGE-IN-FILE

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-PURGE-IN-FILE (2219). This token is always present in the error list.

**Cause.** The ZPURGE-IN-FILE field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's PURGE-IN-FILE attribute.

**Effect.** The command failed.

**Recovery.** Set the ZPURGE-IN-FILE field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2220 ZBAT-ERR-HIGHPIN

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-HIGHPIN (2220). This token is always present in the error list.

**Cause.** The ZHIGHPIN field of ZBAT-MAP-DEF-JOB or ZBAT-MAP-DEF-SCHEDULER specified an invalid value.

**Effect.** The command failed.

**Recovery.** Set the ZHIGHPIN field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2221 ZBAT-ERR-POSIX

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-POSIX (2221). This token is always present in the error list.

**Cause.** The ZPOSIX field of ZBAT-MAP-DEF-JOB specified an invalid value.

**Effect.** The command failed.

**Recovery.** Retry the command, using procedure SSNULL to initialize to null values the fields of ZBAT-MAP-DEF-JOB.

## 2222 ZBAT-ERR-SAVEABEND

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-SAVEABEND (2222). This token is always present in the error list.

**Cause.** The ZSAVEABEND field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's SAVEABEND attribute.

**Effect.** The command failed.

**Recovery.** Set the ZSAVEABEND field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2223 ZBAT-ERR-RUND

ZSPI-TKN-ERRLIST  
ZSPI-TKN-ERROR  
ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
token-type ZSPI-TYP-ERROR  
token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-RUND (2223). This token is always present in the error list.

**Cause.** The ZRUND field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's RUND attribute.

**Effect.** The command failed.

**Recovery.** Set the ZRUND field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2224 ZBAT-ERR-JOBID-ZERO

ZSPI-TKN-ERRLIST  
ZSPI-TKN-ERROR  
ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
token-type ZSPI-TYP-ERROR  
token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-JOBID-ZERO (2224). This token is always present in the error list.

**Cause.** The ZJOBID-ZERO field of ZBAT-MAP-DEF-JOB specified an invalid value for the job's JOBID-ZERO attribute.

**Effect.** The command failed.

**Recovery.** Set the ZJOBID-ZERO field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.



## 2225 ZBAT-ERR-MEM

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MEM (2225). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMEM field of ZBAT-MAP-DEF-JOB specified a number of memory pages outside the allowable range 0 through 64.

**Effect.** The command failed.

**Recovery.** Change ZMEM to specify a number of memory pages in the range 0 through 64 and retry the command.

## 2226 ZBAT-ERR-TIME-LIMIT

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZSPI-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-TIME-LIMIT (2226). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZTIME-LIMIT field of ZBAT-MAP-DEF-JOB specified a time limit outside the allowable range 0 through 999 hours and 0 through 59 minutes.

**Effect.** The command failed.

**Recovery.** Change ZTIME-LIMIT to specify a time limit in the range 0 through 999 hours and 0 through 59 minutes and retry the command.

## 2227 ZBAT-ERR-DESCRIPTION

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-DESCRIPTION (2227). This token is always present in the error list.

**Cause.** The size of ZBAT-TKN-DESCRIPTION exceeded 1000 bytes.

**Effect.** The command failed.

**Recovery.** Reduce the size of ZBAT-TKN-DESCRIPTION to 1000 bytes or less and retry the command.

## 2228 ZBAT-ERR-TOO-MANY-SELECTORS

|                  |                           |
|------------------|---------------------------|
| ZSPI-TKN-ERRLIST | token-type ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type ZSPI-TYP-SSCTL |

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-TOO-MANY-SELECTORS (2228). This token is always present in the error list.

**Cause.** The accumulated size of the specified job-selection tokens exceeded 600 bytes. Job-selection tokens are:

|                            |                           |
|----------------------------|---------------------------|
| ZBAT-TKN-SEL-ADPNAME       | ZBAT-TKN-SEL-NOTADPNAME   |
| ZBAT-TKN-SEL-ASSIGN-NAME   | ZBAT-TKN-SEL-NOTCLASSNAME |
| ZBAT-TKN-SEL-CLASSNAME     | ZBAT-TKN-SEL-NOTCLASSNAME |
| ZBAT-TKN-SEL-DEFINE-NAME   | ZBAT-TKN-SEL-NOTJOBNAME   |
| ZBAT-TKN-SEL-EXECUTORNAME  | ZBAT-TKN-SEL-NOTLIST      |
| ZBAT-TKN-SEL-INNAME        | ZBAT-TKN-SEL-NOTUSERNAME  |
| ZBAT-TKN-SEL-JOB-NUMBER    | ZBAT-TKN-SEL-NOTWAITON    |
| ZBAT-TKN-SEL-JOBNAME       | ZBAT-TKN-SEL-PARAM-NAME   |
| ZBAT-TKN-SEL-LIST          | ZBAT-TKN-SEL-USERNAME     |
| ZBAT-TKN-SEL-NETBATCH-NAME | ZBAT-TKN-SEL-WAITON       |

**Effect.** The command failed.

**Recovery.** Reduce the size of or delete job-selection tokens and retry the command.

## 2229 ZBAT-ERR-NODENAME

|                  |            |                 |
|------------------|------------|-----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST   |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR  |
| ZBAT-TKN-STRING  | token-type | ZSPI-TYP-STRING |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL  |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-NODENAME (2229). This token is always present in the error list.

#### ZBAT-TKN-STRING

is the name of the invalid remote node.

**Cause.** The ZLOCALNAMES field of ZBAT-MAP-DEF-SCHEDULER specified an invalid remote node.

**Effect.** The command failed.

**Recovery.** Change ZLOCALNAMES to specify a valid remote node and retry the command.

## 2230 ZBAT-ERR-MAXPRI

|                        |                               |
|------------------------|-------------------------------|
| ZSPI-TKN-ERRLIST       | token-type ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type ZBAT-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MAXPRI (2230). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMAXPRI field of ZBAT-MAP-DEF-SCHEDULER specified a maximum priority outside the allowable range 1 through 199.

**Effect.** The command failed.

**Recovery.** Change ZMAXPRI to specify a maximum priority in the range 1 through 199 and retry the command.

## 2231 ZBAT-ERR-MAXCONCURRENTJOBS

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZBAT-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MAXCONCURRENTJOBS (2231). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMAXCONCURRENTJOBS field of ZBAT-MAP-DEF-SCHEDULER specified a concurrent-jobs limit outside the allowable range 0 through 500.

**Effect.** The command failed.

**Recovery.** Change ZMAXCONCURRENTJOBS to specify a concurrent-jobs limit in the range 0 through 500 and retry the command.

## 2232 ZBAT-ERR-MAXTEMPEXECUTORS

|                        |            |                    |
|------------------------|------------|--------------------|
| ZSPI-TKN-ERRLIST       | token-type | ZSPI-TYP-LIST      |
| ZSPI-TKN-ERROR         | token-type | ZSPI-TYP-ERROR     |
| ZBAT-TKN-MIN-MAX-ERROR | token-type | ZBAT-TYP-INT2-TRIO |
| ZSPI-TKN-ENDLIST       | token-type | ZSPI-TYP-SSCTL     |

### Tokens

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-MAXTEMPEXECUTORS (2232). This token is always present in the error list.

#### ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZMAXTEMPEXECUTORS field of ZBAT-MAP-DEF-SCHEDULER specified a temporary-executors limit outside the allowable range 0 through 500.

**Effect.** The command failed.

**Recovery.** Change ZMAXTEMPEXECUTORS to specify a temporary-executors limit in the range 0 through 500 and retry the command.

## 2233 ZBAT-ERR-EVERY-CATCHUP

|                  |            |                |
|------------------|------------|----------------|
| ZSPI-TKN-ERRLIST | token-type | ZSPI-TYP-LIST  |
| ZSPI-TKN-ERROR   | token-type | ZSPI-TYP-ERROR |
| ZSPI-TKN-ENDLIST | token-type | ZSPI-TYP-SSCTL |

### Token

#### ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EVERY-CATCHUP (2233). This token is always present in the error list.

**Cause.** The ZEVEY-CATCHUP field of ZBAT-MAP-DEF-SCHEDULER specified an invalid value for the scheduler's CATCHUP attribute.

**Effect.** The command failed.

**Recovery.** Set the ZEVEY-CATCHUP field to a valid Boolean value (for example, ZSPI-VAL-TRUE or ZSPI-VAL-FALSE) and retry the command.

## 2234 ZBAT-ERR-EMS

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZSPI-TYP-SSCTL

### Token

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-EMS (2234). This token is always present in the error list.

**Cause.** The ZEMS field of ZBAT-MAP-DEF-SCHEDULER specified an invalid value for the scheduler's EMS attribute.

**Effect.** The command failed.

**Recovery.** Set the ZEMS field to a valid ZBAT-DDL-EMS value and retry the command.

## 2235 ZBAT-ERR-PFS

ZSPI-TKN-ERRLIST  
 ZSPI-TKN-ERROR  
 ZBAT-TKN-MIN-MAX-ERROR  
 ZSPI-TKN-ENDLIST

token-type ZSPI-TYP-LIST  
 token-type ZSPI-TYP-ERROR  
 token-type ZBAT-TYP-INT2-TRIO  
 token-type ZSPI-TYP-SSCTL

### Tokens

ZSPI-TKN-ERROR

is the standard SPI error token. Its value consists of the NetBatch subsystem ID and the error number ZBAT-ERR-PFS (2235). This token is always present in the error list.

ZBAT-TKN-MIN-MAX-ERROR

contains three values in this error list. The first double integer contains the minimum allowable value for the token or field. The second double integer contains the maximum allowable value for the token or field. The third double integer contains the value specified by the application.

**Cause.** The ZPFS field of ZBAT-MAP-DEF-JOB specified a nonzero process-file-segment size outside the allowable range 131,072 through 1,048,576 bytes.

**Effect.** The command failed.

**Recovery.** Change ZPFS to specify a process-file-segment size of zero bytes or 131,072 through 1,048,576 bytes and retry the command.



# **B** Token Codes and Token Maps

This appendix lists token information specific to the NetBatch subsystem, including:

- Token codes specific to the NetBatch subsystem and the token type for each token code
- Token maps specific to the NetBatch subsystem and the DDL definition for each token map

# NetBatch Token Codes and Token Types

[Table B-1](#) lists token codes specific to the NetBatch subsystem (that is, token codes whose names begin with ZBAT-TKN-). For each code, the table shows the token type.

You can derive the symbolic name of the token number of a token code from the name of the token code by replacing -TKN- with -TNM-. For example, token code ZBAT-TKN-OBJECT has the token number ZBAT-TNM-OBJECT.

**Table B-1. NetBatch Token Codes and Token Types** (page 1 of 3)

| Token Code                 | TokenType                |
|----------------------------|--------------------------|
| ZBAT-TKN-ATT-SET-ASSIGN    | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-ATT-SET-DEFINE    | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-ATT-SET-ID        | ZSPI-TYP-STRING          |
| ZBAT-TKN-ATT-SET-PARAM     | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-ATT-SET-SECURITY  | ZSPI-TYP-INT             |
| ZBAT-TKN-ATT-SET-TEMPORARY | ZSPI-TYP-BOOLEAN         |
| ZBAT-TKN-BATCHCTL          | ZSPI-TYP-STRING          |
| ZBAT-TKN-BYTESTRING        | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-CALENDAR          | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-CHAR6             | ZBAT-TYP-CHAR6           |
| ZBAT-TKN-COMMAND           | ZBAT-TYP-COMMAND         |
| ZBAT-TKN-COMPLETION-CODE   | ZBAT-TYP-COMPLETION-CODE |
| ZBAT-TKN-DATA-BASE         | ZSPI-TYP-STRING          |
| ZBAT-TKN-DESCRIPTION       | ZSPI-TYP-STRING          |
| ZBAT-TKN-EXECUTOR-ID       | ZSPI-TYP-STRING          |
| ZBAT-TKN-EXECUTOR-PROGRAM  | ZSPI-TYP-STRING          |
| ZBAT-TKN-EXTSWAP-FILE      | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-FORMATSUBJECT     | ZSPI-TYP-INT             |
| ZBAT-TKN-IN-FILE           | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-INT               | ZSPI-TYP-INT             |
| ZBAT-TKN-INT2              | ZSPI-TYP-INT2            |
| ZBAT-TKN-JOB-ID            | ZSPI-TYP-STRING          |
| ZBAT-TKN-JOB-NAME-ID       | ZSPI-TYP-STRING          |
| ZBAT-TKN-JOB-NUMBER        | ZSPI-TYP-INT             |
| ZBAT-TKN-LIB-FILE          | ZSPI-TYP-BYTESTRING      |
| ZBAT-TKN-LOG-FILE          | ZSPI-TYP-STRING          |
| ZBAT-TKN-MIN-MAX-ERROR     | ZBAT-TYP-INT2-TRIO       |
| ZBAT-TKN-NETBATCH-NAME     | ZSPI-TYP-CHAR24          |

**Table B-1. NetBatch Token Codes and Token Types** (page 2 of 3)

| <b>Token Code</b>          | <b>TokenType</b>       |
|----------------------------|------------------------|
| ZBAT-TKN-OBJECT            | ZBAT-TYP-OBJECT        |
| ZBAT-TKN-OUT-FILE          | ZSPI-TYP-BYTESTRING    |
| ZBAT-TKN-PC-ERROR0         | ZBAT-TYP-PC-ERROR0     |
| ZBAT-TKN-PC-ERROR1         | ZBAT-TYP-PC-ERROR1     |
| ZBAT-TKN-PC-ERROR2         | ZSPI-TYP-INT           |
| ZBAT-TKN-PHANDLE           | ZSPI-TYP-PHANDLE       |
| ZBAT-TKN-REASON-NUMBER     | ZBAT-TYP-REASON        |
| ZBAT-TKN-RETCODE           | ZBAT-TYP-RETCODE       |
| ZBAT-TKN-SCHEDULER-ID      | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-ADPNAME       | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-ASSIGN-NAME   | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-CLASSNAME     | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-DEFINE-NAME   | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-EXECUTORNAME  | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-INNAME        | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-JOB-NUMBER    | ZBAT-TYP-JOB-NUMBER    |
| ZBAT-TKN-SEL-JOBNAME       | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-LIST          | ZBAT-TYP-LIST          |
| ZBAT-TKN-SEL-NETBATCH-NAME | ZBAT-TYP-NETBATCH-NAME |
| ZBAT-TKN-SEL-NOTADPNAME    | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-NOTCLASSNAME  | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-NOTINNAME     | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-NOTJOBNAME    | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-NOTLIST       | ZSPI-TYP-INT           |
| ZBAT-TKN-SEL-NOTUSERNAME   | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-NOTWAITON     | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-PARAM-NAME    | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-USERNAME      | ZSPI-TYP-STRING        |
| ZBAT-TKN-SEL-WAITON        | ZSPI-TYP-STRING        |
| ZBAT-TKN-START-TIME        | ZSPI-TYP-TIMESTAMP     |
| ZBAT-TKN-STARTUP-MESSAGE   | ZSPI-TYP-BYTESTRING    |
| ZBAT-TKN-STRING            | ZSPI-TYP-STRING        |
| ZBAT-TKN-SWAP-FILE         | ZSPI-TYP-BYTESTRING    |
| ZBAT-TKN-TERM-FILE         | ZSPI-TYP-BYTESTRING    |

**Table B-1. NetBatch Token Codes and Token Types** (page 3 of 3)

| <b>Token Code</b>         | <b>TokenType</b>    |
|---------------------------|---------------------|
| ZBAT-TKN-TERMINATION-INFO | ZSPI-TYP-INT        |
| ZBAT-TKN-TEXT             | ZSPI-TYP-STRING     |
| ZBAT-TKN-TIME-LIMIT       | ZSPI-TYP-INT2       |
| ZBAT-TKN-TOTAL-CPU-TIME   | ZSPI-TYP-INT4       |
| ZBAT-TKN-USERID           | ZSPI-TYP-USERID     |
| ZBAT-TKN-VOLUME-SUBVOL    | ZSPI-TYP-BYTESTRING |

## NetBatch Token Maps and DDL Definitions

[Table B-2](#) lists token maps specific to the NetBatch subsystem (that is, token maps whose names begin with ZBAT-MAP-). For each token map, the table shows the DDL definition.

You can derive the symbolic name of the token number of a token map from the name of the token map by replacing -MAP- with -TNM-. For example, token map ZBAT-MAP-DEF-JOB has the token number ZBAT-TNM-DEF-JOB.

**Table B-2. NetBatch Token Maps and DDL Definitions** (page 1 of 8)

| Token Map             | DDL Definition   |
|-----------------------|--|
| ZBAT-MAP-DEF-CLASS    | Definition ZBAT-DDL-DEF-CLASS.<br>02 ZINITIATION<br>End<br>Type ZSPI-DDL-BOOLEAN   |
| ZBAT-MAP-DEF-CRONTAB  | Definition ZBAT-DDL-DEF-CRONTAB.<br>02 ZMINUTES<br>02 ZHOURS<br>02 ZDAYS<br>02 ZMONTHS<br>02 ZWEEKDAYS<br>End<br>Type ZSPI-DDL-INT4<br>Type ZSPI-DDL-INT2<br>Type ZSPI-DDL-INT2<br>Type ZSPI-DDL-INT<br>Type ZSPI-DDL-INT  |
| ZBAT-MAP-DEF-EXECUTOR | Definition ZBAT-DDL-DEF-EXECUTOR.<br>02 ZCPU<br>02 ZJOBNUMBER<br>02 ZCLASS-COUNT<br>02 ZCLASSES<br>03 ZCLASSNAME<br>03 FILLER<br>02 ZCLASS<br>End<br>Type ZSPI-DDL-INT<br>Type ZSPI-DDL-INT<br>Type ZSPI-DDL-INT<br>Occurs 8 times<br>Type BAT-DDL-NETBATCH-NAME<br>Type ZSPI-DDL-INT<br>Type ZBAT-DDL-NETBATCH-NAME |

**Table B-2. NetBatch Token Maps and DDL Definitions** (page 2 of 8)

| <b>Token Map</b> | <b>DDL Definition</b>   |
|------------------|---|
| ZBAT-MAP-DEF-JOB | Definition ZBAT-DDL-DEF-JOB.                                    |
|                  | 02 ZCLASSNAME                      Type ZBAT-DDL-NETBATCH-NAME  |
|                  | 02 ZHOLD                              Type ZSPI-DDL-BOOLEAN     |
|                  | 02 ZHOLD-AFTER                      Type ZSPI-DDL-BOOLEAN       |
|                  | 02 ZRESTART                          Type ZSPI-DDL-BOOLEAN      |
|                  | 02 ZSTOP-ON-ABEND                  Type ZSPI-DDL-BOOLEAN        |
|                  | 02 ZAT-FLAG                          Type ZSPI-DDL-BOOLEAN      |
|                  | 02 ZIFFAILS                          Type ZSPI-DDL-BOOLEAN      |
|                  | 02 ZPURGE-IN-FILE                  Type ZSPI-DDL-BOOLEAN        |
|                  | 02 ZSTALL                            Type ZSPI-DDL-BOOLEAN      |
|                  | 02 ZINFO-NEXT-RUNTIME              Type ZSPI-DDL-INT4           |
|                  | 02 ZINFO-OUT-SPOOL-NUM            Type ZSPI-DDL-INT             |
|                  | 02 ZINFO-WHICH-LIST                Type ZBAT-DDL-JOB-WHICH-LIST |
|                  | 02 ZINFO-SPECIAL-REASON          Type ZBAT-DDL-SPECIAL-REASON   |
|                  | 02 ZINFO-TOTAL-CPU-TIME            Type ZSPI-DDL-INT4           |
|                  | 02 ZINFO-OPEN-ACCESSOR            Type ZSPI-DDL-INT             |
|                  | 02 ZREMID                            Type ZSPI-DDL-BOOLEAN      |
|                  | 02 ZEVERTY-DAYS                    Type ZSPI-DDL-INT            |
|                  | 02 ZEVERTY-HOURS                   Type ZSPI-DDL-INT            |
|                  | 02 ZEVERTY-MINUTES                Type ZSPI-DDL-INT             |
|                  | 02 ZDEFAULT-SECURITY               Type ZSPI-DDL-INT            |
|                  | 02 ZPRI                               Type ZSPI-DDL-INT         |
|                  | 02 ZSELPRI                           Type ZSPI-DDL-INT          |
|                  | 02 ZHIGHPIN                         Type ZSPI-DDL-BOOLEAN       |
|                  | 02 ZMAXPRINTLINES                  Type ZSPI-DDL-INT2           |
|                  | 02 ZMAXPRINTPAGES                  Type ZSPI-DDL-INT2           |
|                  | 02 ZTAPEDRIVES                     Type ZSPI-DDL-INT            |

**Table B-2. NetBatch Token Maps and DDL Definitions** (page 3 of 8)

| <b>Token Map</b> | <b>DDL Definition</b>                              |
|------------------|--|
|                  | 02 ZDATE.  |
|                  | 03 ZYEAR                      Type ZSPI-DDL-INT    |
|                  | 03 ZMONTH                    Type ZSPI-DDL-INT     |
|                  | 03 ZDAY                      Type ZSPI-DDL-INT     |
|                  | 02 ZTIME.  |
|                  | 03 ZHOUR                     Type ZSPI-DDL-INT     |
|                  | 03 ZMINUTE                  Type ZSPI-DDL-INT      |
|                  | 03 ZSECOND                  Type ZSPI-DDL-INT      |
|                  | 03 ZMILLISECOND            Type ZSPI-DDL-INT       |
|                  | 03 ZMICROSECOND            Type ZSPI-DDL-INT       |
|                  | 02 ZPOSIX                    Type ZSPI-DDL-INT     |
|                  | 02 ZSAVEABEND              Type ZSPI-DDL-BOOLEAN   |
|                  | 02 ZRUND                     Type ZSPI-DDL-BOOLEAN |
|                  | 02 ZJOBID-ZERO              Type ZSPI-DDL-BOOLEAN  |
|                  | 02 ZMEM                      Type ZSPI-DDL-INT     |
|                  | 02 ZPFS                      Type ZSPI-DDL-INT2    |
|                  | 02 ZNAME                     Type ZSPI-DDL-CHAR8   |
|                  | 02 ZINFO-TIME-SUBMIT        Type ZSPI-DDL-INT4     |
|                  | 02 ZINFO-LAST-MOD          Type ZSPI-DDL-INT4      |
|                  | 02 ZINFO-LAST-MODUSER      Type ZSPI-DDL-INT       |
|                  | 02 ZTIME-LIMIT                                     |
|                  | End                          Type ZSPI-DDL-INT2    |

**Table B-2. NetBatch Token Maps and DDL Definitions** (page 4 of 8)

| <b>Token Map</b>       | <b>DDL Definition</b>   |
|------------------------|---|
| ZBAT-MAP-DEF-SCHEDULER | Definition ZBAT-DDL-DEF-SCHEDULER.<br>02 ZBACKUPCPU2                   Type ZSPI-DDL-INT<br>02 ZBACKUPCPU1                   Type ZSPI-DDL-INT<br>02 ZMAXCONCURRENTJOBS           Type ZSPI-DDL-INT<br>02 ZMAXTEMPEXECUTORS           Type ZSPI-DDL-INT<br>02 ZTAPEDRIVES                   Type ZSPI-DDL-INT<br>02 ZMAXPR                         Type ZSPI-DDL-INT<br>02 ZINFO-TAPEDRIVES-IN-USE      Type ZSPI-DDL-INT<br>02 ZAT-ALLOWED                  Type ZSPI-DDL-BOOLEAN<br>02 ZSUBMIT-ALLOWED              Type ZSPI-DDL-BOOLEAN<br>02 ZEVERY-CATCHUP               Type ZSPI-DDL-BOOLEAN<br>02 ZEMS                          Type ZSPI-DDL-E<br>02 ZCLASSNAME                    Type ZSPI-DDL-NETBATCH-NAME<br>02 ZPRI                          Type ZSPI-DDL-INT<br>02 ZSELPRI                        Type ZSPI-DDL-INT2<br>02 ZMAXPRINTLINES                Type ZSPI-DDL-INT2<br>02 ZMAXPRINTPAGES                Type ZSPI-DDL-BOOLEAN<br>02 ZSTOP-ON-ABEND                Type ZSPI-DDL-BOOLEAN<br>02 ZSTALL                         Type ZSPI-DDL-BOOLEAN<br>02 ZHIGHPIN                       Type ZSPI-DDL-BOOLEAN<br>02 ZINITIATION                    Type ZSPI-DDL-CHAR8<br>02 ZLOCALNAMES                   Occurs 30 times<br>End |
| ZBAT-MAP-DEF-WAITON    | Definition ZBAT-DDL-DEF-WAITON.<br>02 ZMASTER                       Type ZBAT-DDL-NETBATCH-NAME<br>02 ZINDICATOR                    Type ZBAT-DDL-WAITON-INDICATOR<br>02 ZFOR                            Type ZBAT-DDL-WAITON-FOR<br>End  |



**Table B-2. NetBatch Token Maps and DDL Definitions** (page 5 of 8)

| <b>Token Map</b>             | <b>DDL Definition</b>  |
|------------------------------|--|
| ZBAT-MAP-<br>DEFINE-ERROR    | Definition ZBAT-DDL-DEFINE-<br>ERROR.<br>02 ZNAMELEN                      Type ZSPI-DDL-INT.<br>02 ZNAMETXT                      Type ZSPI-DDL-CHAR24.<br>02 ZERR                            Type ZSPI-DDL-INT.<br>02 ZATTRLEN                      Type ZSPI-DDL-INT.<br>02 ZATTRTXT                      Type ZSPI-DDL-CHAR16.<br>02 ZCLASSLEN                      Type ZSPI-DDL-INT.<br>02 ZCLASSTXT                      Type ZSPI-DDL-CHAR16.<br>02 ZCHECKNUM                      Type ZSPI-DDL-INT.<br>02 ZADDR                          Type ZSPI-DDL-INT.<br>End |
| ZBAT-MAP-PAR-<br>RELEASE-JOB | Definition ZBAT-DDL-PAR-<br>RELEASE-JOB.<br>02 ZRELEASER                      Type ZBAT-DDL-NETBATCH-NAME.<br>02 ZJOBNAME                        Type ZBAT-DDL-NETBATCH-NAME.<br>End   |

**Table B-2. NetBatch Token Maps and DDL Definitions** (page 6 of 8)

| <b>Token Map</b>                 | <b>DDL Definition</b>  |
|----------------------------------|--|
| ZBAT-MAP-<br>STATUS-<br>EXECUTOR | Definition ZBAT-DDL-STATUS-<br>EXECUTOR.<br>02 ZCPU<br>02 ZJOBNUMBER<br>02 ZWHICH-LIST<br>02 ZCLASS<br>End |
|                                  | Type ZSPI-DDL-INT.<br>Type ZSPI-DDL-INT.<br>Type ZSPI-DDL-EXECUTOR-LIST.<br>Type ZSPI-DDL-NETBATCH-NAME.   |

**Table B-2. NetBatch Token Maps and DDL Definitions** (page 7 of 8)

| <b>Token Map</b>    | <b>DDL Definition</b>                                       |
|---------------------|---|
| ZBAT-MAP-STATUS-JOB | Definition ZBAT-DDL-STATUS-JOB.                             |
|                     | 02 ZOUT-SPOOL-NUM           Type ZSPI-DDL-INT.              |
|                     | 02 ZSELPRI                   Type ZSPI-DDL-INT.             |
|                     | 02 ZOPEN-ACCESSOR-DETAIL.   Type ZSPI-DDL-BYTE.             |
|                     | 03 ZGROUP                   Type ZSPI-DDL-BYTE.             |
|                     | 03 ZUSER                    Redefines ZOPEN-ACCESSOR-DETAIL |
|                     | 02 ZOPEN-ACCESSOR          Type ZSPI-DDL-INT.               |
|                     | 02 ZCLASSNAME              Type ZSPI-DDL-NETBATCH-NAME.     |
|                     | 02 ZWHICH-LIST              Type ZSPI-DDL-JOB-WHICH-LIST.   |
|                     | 02 ZSPECIAL-REASON         Type ZSPI-DDL-SPECIAL-REASON.    |
|                     | 02 ZNEXT-RUNTIME           Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-PREV-RUNTIME      Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-START             Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-FINISH            Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-PUT-ON-LIST       Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-USED              Type ZSPI-DDL-INT4.              |
|                     | 02 ZRE MID                  Type ZSPI-DDL-BOOLEAN.          |
|                     | 02 ZEXECUTOR               Type ZSPI-DDL-NETBATCH-NAME.     |
|                     | 02 ZEXECPHANDLE            Type ZSPI-DDL-PHANDLE.           |
|                     | 02 ZTIME-ELAPSED MAX       Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-CPUMAX            Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-                   Type ZSPI-DDL-INT4.             |
|                     | 02 ZTIME-CPUTOTAL          Type ZSPI-DDL-INT4.              |
|                     | 02 ZTIME-SUBMIT            Type ZSPI-DDL-INT4.              |
|                     | 02 ZLAST-CC                Type ZSPI-DDL-INT.               |
|                     | 02 ZTIMES-RUN              Type ZSPI-DDL-INT2.              |
|                     | 02 ZTIME-LIMIT             Type ZSPI-DDL-INT2.              |
|                     | End   |

**Table B-2. NetBatch Token Maps and DDL Definitions** (page 8 of 8)

| <b>Token Map</b>          | <b>DDL Definition</b>  |
|---------------------------|--|
| ZBAT-MAP-STATUS-SCHEDULER | Definition ZBAT-DDL-STATUS-SCHEDULER.<br>02 ZSTATE Type ZBAT-DDL-SCHEDULAR-STATE.<br>02 ZEXECUTOR.<br>03 ZOFF Type ZSPI-DDL-INT.<br>03 ZON Type ZSPI-DDL-INT.<br>03 ZACTIVE Type ZSPI-DDL-INT.<br>03 ZSTOP Type ZSPI-DDL-INT.<br>03 ZDOWN Type ZSPI-DDL-INT.<br>03 ZDELETE Type ZSPI-DDL-INT.<br>02 ZJOB.<br>03 ZREADY Type ZSPI-DDL-INT.<br>03 ZEXECUTING Type ZSPI-DDL-INT.<br>03 ZSPECIAL Type ZSPI-DDL-INT.<br>03 ZTIME Type ZSPI-DDL-INT.<br>03 ZEVENT Type ZSPI-DDL-INT.<br>03 ZSUSPENDED Type ZSPI-DDL-INT.<br>03 ZRUNNEXT Type ZSPI-DDL-INT.<br>03 ZRUNNOW Type ZSPI-DDL-INT.<br>03 ZTAPE Type ZSPI-DDL-INT.<br>02 ZJOBCLASS.<br>03 ZOFF Type ZSPI-DDL-INT.<br>03 ZON Type ZSPI-DDL-INT.<br>02 ZPROCESS.<br>03 ZACTIVE Type ZSPI-DDL-INT.<br>03 ZSUSPENDED Type ZSPI-DDL-INT.<br>02 ZTAPE.<br>03 ZCONFIG Type ZSPI-DDL-INT.<br>03 ZTAPEDRIVES-IN-USE Type ZSPI-DDL-INT.<br>02 ZATT-SET-COUNT Type ZSPI-DDL-INT.<br>02 ZINITIATION Type ZSPI-DDL-BOOLEAN.<br>02 ZSUBMIT-ALLOWED Type ZSPI-DDL-BOOLEAN.<br>End |

# Sample Programs

This appendix contains source code for sample C, COBOL, TACL, and TAL programs that illustrate programmatic management of the NetBatch subsystem. The sample programs include source code for:

- The statements that include the standard and subsystem-specific DDL definitions required by the NetBatch subsystem
- The commands and procedures to open a NetBatch scheduler
- Programmatic SUBMIT JOB and STATUS JOB commands
- Error handling

| <b>Topic</b>                         | <b>Page</b>          |
|--------------------------------------|----------------------|
| <a href="#">Sample C Program</a>     | <a href="#">C-2</a>  |
| <a href="#">Sample COBOL Program</a> | <a href="#">C-15</a> |
| <a href="#">Sample TACL Macros</a>   | <a href="#">C-30</a> |
| <a href="#">Sample TAL Program</a>   | <a href="#">C-41</a> |

# Sample C Program

[Example C-1](#) on page C-4 contains the source code for a sample C program. The program submits a job to a scheduler, executes a STATUS JOB command on the submitted job, and displays some job details. The source code for the program is available in the file NBSPIEX in the NetBatch installation subvolume.

The prerequisite to completing the procedure is a D21 or later scheduler named \$ZBAT running on the node where the program will run.

## Step 1: Copy the Source Code

Copy the source code for the sample C program from file NBSPIEX to a new EDIT file:

```
> EDIT; GET $SYSGEN.ZNETBTCH.NBSPIEX 7/537 PUT TEMP.CSRC
TEXT EDITOR - T9601D20 - (01JUN93)
CURRENT FILE IS $DATA7.TEMP.CSRC
```

## Step 2: Change ISV.ZSPIDEF References

Change ISV.ZSPIDEF references in the new file to specify the volume and subvolume containing the source-definition files ZSPIC and ZBATC:

```
*LIST BOTH /ISV.ZSPIDEF/
 39      #include "$ISV.ZSPIDEF.ZSPIC" nolist
 40      #include "$ISV.ZSPIDEF.ZBATC" nolist
*CHANGE /ISV/SYSGEN/ ALL
 39      #include "$SYSGEN.ZSPIDEF.ZSPIC" nolist
 40      #include "$SYSGEN.ZSPIDEF.ZBATC" nolist
```

## Step 3: Change Class Name

Change class name CLASS-A to that of an existing class in \$ZBAT if class CLASS-A does not exist in that scheduler. Otherwise, add class CLASS-A to \$ZBAT.

```
*CHANGE /"CLASS-A                                "/" OPERATIONS
"/ ALL
 329      cptr = strncpy (JOB.zclassname.u_z_c.z_c,
"OPERATIONS                                ",24);
```

## Step 4: Change Job Name

Change job name MASTER-A to your own choice of name if MASTER-A conflicts with an existing production job in \$ZBAT. Otherwise, delete job MASTER-A from \$ZBAT.

```
*CHANGE /"MASTER-A                                "/" C-JOB
"/ ALL
 242      cptr = strncpy (jobname.u_z_c.z_c,
"C-JOB                                ",24);
```

## Step 5: Compile Source File

End the EDIT session and compile the source file:

```
*EXIT  
> C /IN TEMP.CSRC/ TEMP.COBJ; SUPPRESS
```

## Step 6: Test the Program

Test the program by running the compiled object:

```
> RUN TEMP.COBJ  
Job submitted to $ZBAT using SPI  
Job number : 5  
Job name    : C-JOB  
Job status from $ZBAT using SPI  
Selpri      : 3  
Class       : OPERATIONS
```

## Step 7: Delete the Submitted Job

```
> BATCHCOM $ZBAT; DELETE JOB C-JOB  
Job C-JOB Jobnumber 5 deleted
```

---

**Example C-1. Sample SPI C Program**

```
#pragma INSPECT,SYMBOLS
#pragma NOMAP
#pragma NOLMAP
#pragma RUNNABLE
#pragma XMEM
#pragma HEAP 20 pages

/ *
 *#####
 *#
#
 *# This C source code compiles into a sample program that
#
 *# demonstrates the subsystem programmatic interface (SPI) to the
#
 *# NetBatch scheduler. The program performs two functions:
#
 *#     * Submits a job to scheduler $ZBAT
#
 *#     * Executes a STATUS JOB command on the submitted job
#
 *#     and displays some of the job's details
#
 *#
#
 *#####
 */

#include <stdioh>    nolist
#include <stringh>   nolist
#include <stdlibh>    nolist
#include <memoryh>    nolist
#include <talh>       nolist
#include <cextdecs(SSINIT, SSNUL, SSPUTTKN, SSPUT, SSGETTKN, SSGET)> nolist
#include <cextdecs(WRITEREADX, DEBUG, FILE_OPEN_, FILE_CLOSE_)> nolist
#include <cextdecs(FILE_GETINFO_, PROCESS_STOP_)> nolist

/ * Local ZSPIDEF volume */

#include "$ISV.ZSPIDEF.ZSPIC" nolist
#include "$ISV.ZSPIDEF.ZBATC (zbat_val_version, zbat_val_ssid,
zbat_tkn_ems,\
                                constants, error_constants,\
                                zbat_ddl_job_which_list,\
                                zbat_ddl_special_reason, zbat_ddl_def_job,\
                                zbat_ddl_status_job, zbat_ddl_msg_buffer,\
                                zbat_map_def_job, zbat_map_status_job,\
                                zbat_ddl_netbatch_name, zbat_tkn_spi)" nolist

#define MAXFILEBYTES 100
#define MAXFILESIZE 12

(continued)
```

---



```

/* These DEFINES are used for zbat structs to shorten names */

#define BATBUFDEF zbat_ddl_msg_buffer_def

/* Use this struct for variable-length string tokens */
typedef struct bytestr_buf {
    int    len;                /* Length or count of string in bytes */
    char   str[MAXFILEBYTES]; /* Data */
} BYTESTR;

BATBUFDEF *spi_buff;          /* Global SPI buffer */

/* Declare the ssids using the typedefs from the DDL output */

zbat_val_ssids_def zbat_val_ssids;

zbat_ddl_def_job_def JOB;      /* Job definition structure */
zbat_ddl_status_job_def STATUS; /* Job status structure */
BYTESTR work_area;             /* Working bytestring buffer */

/* Error text for the system-procedure-call errors */

char    essget[13]      = "SSGET Failed";
char    essput[13]      = "SSPUT Failed";
char    essinit[14]     = "SSINIT Failed";
char    essnull[14]     = "SSNULL Failed";
char    essgettkn[16]   = "SSGETTKN Failed";
char    essputtkn[16]   = "SSPUTTKN Failed";
char    efileopen[18]   = "FILE_OPEN_ Failed";
char    ewritereadx[18] = "WRITEREADX Failed";
char    emalloc[34]     = "MALLOC() Failed to obtain memory";
char    eretcode[35]    = "RETCODE indicated a NetBatch error";

short   schd_chan;       /* Channel for scheduler */
int     spi_err;         /* SPI error value */

#pragma page

/*

(continued)

```

```

*-----
* stopwitherror(error, err_msg)
* Use:
*     Error handling
* Effects:
*     Forces the program to abend with CC and prints message passed
*-----
*/

void stopwitherror(const short errnum,
                  char * err_msg)
{
    PROCESS_STOP_ (,,,2,(short)errnum,,(char *) err_msg,(short
)strlen(err_msg));
}

#pragma page

/*
*-----
* open_scheduler()
* Use:
*     Part of initialization
* Effects:
*     Opens scheduler for SPI I/O if successful
*     Stops program otherwise
*-----
*/

void open_scheduler(void)
{
    int    status = 0;
    char   schname[12] = "$ZBAT.#ZSPI";

    /* Open scheduler with #ZSPI for SPI I/O */ /* */
}

void open_scheduler(void)
{
    (continued)
}

```

```

    int    status = 0;
    char   schname[12] = "$ZBAT.#ZSPI";

    /* Open scheduler with #ZSPI for SPI I/O */

    status = FILE_OPEN_ ((char *) schname, (short) strlen(schname), (short
*) &schd_chan);
    if (status != 0)
        stopwitherror(status, (char *) &efileopen);
}

#pragma page

/*
-----
* send_spi
* Use:
*     Performs WRITEREADX to scheduler (SPI buffer)
* Effects:
*     Returns RETCODE if successful
*     Stops program if unsuccessful
*-----
*/

int send_spi(void)
{
    int    retcode;
    int    ccval= 0;
    int    error;
    short  used_len;

    used_len = (spi_buff->z_occurs)+6;
    ccval = WRITEREADX(    (short)      schd_chan,
                          (char *)      spi_buff,
                          (short)      used_len,
                          (short)      (ZBAT_VAL_BUFLen+6)
                          );

    if (ccval != CCE) {
        FILE_GETINFO_(    (short)      schd_chan,
                          (short *) &error
                          );
        stopwitherror(error, (char *) &ewritereadx);
    }
}

(continued)

```

```

    /* Get the RETCODE token returned in the SPI buffer */

    spi_err =  SSGETTKN ( (short *)      spi_buff,
                        (long)          ZSPI_TKN_RETCODE,
                        (char *)        &retcode,
                        1
                    );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essgettkn);

    /* Return the SPI RETCODE */

    return (retcode);
}

#pragma page

/*
 *-----
 * submit_job
 * Use:
 *     Submits a job to the scheduler
 * Effects:
 *     Sets SPI buffer to job submit details if successful
 *     Stops program otherwise
 *-----
 */

void submit_job(void)
{
    int      retcode = 0;
    char      *cptr;

    /* Initialize the SPI buffer */

    spi_err = SSINIT (
        (short *) spi_buff          /* SPI buffer
    */
        , (ZBAT_VAL_BUFLen+6)      /* SPI buffer length
    */
        , (short *) &zbat_val_ssid /* SSID - subsystem ID
    */
        , ZSPI_VAL_CMDHDR          /* SPI buffer type
    */
        , ZBAT_CMD_SUBMIT          /* Command number
    */
        , ZBAT_OBJ_JOB             /* Object type
    */
        , 0                        /* Max-resp
    */
        ,                          /* Server-vrsn
    */
        , ZSPI_VAL_TRUE            /* Checksum enabled
    */
    );
    (continued)

```

```

        );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essinit);

    /* Insert the job-name token */

    strcpy (work_area.str, "MASTER-A");
    work_area.len = strlen(work_area.str);

    spi_err = SSPUTTKN (
buffer    (short *)      spi_buff,          /* SPI
*/          (long)        ZBAT_TKN_SEL_JOBNAME, /* Token ID
*/          (char *)      &work_area        /* Token
value */
    );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essputtkn);

    /* Insert the executor-program token */

    strcpy (work_area.str, "$SYSTEM.SYSTEM.TACL");
    work_area.len = strlen(work_area.str);

    spi_err = SSPUTTKN (
ID        (short *)      spi_buff,          /* SPI buffer */
*/          (long)        ZBAT_TKN_EXECUTOR_PROGRAM, /* Token
*/          (char *)      &work_area        /* Token value */
    );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essputtkn);

    /* Insert the volume-subvol token */

    strcpy (work_area.str, getenv("DEFAULTS"));
    work_area.len = strlen(work_area.str);

    spi_err = SSPUTTKN (
buffer    (short *)      spi_buff,          /* SPI
*/          (long)        ZBAT_TKN_VOLUME_SUBVOL, /* Token ID
*/          (char *)      &work_area        /* Token
value */
    (continued)

```

```

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essputtkn);

    /* Insert the in-file token */

    strcpy (work_area.str, getenv("DEFAULTS"));
    strcat (work_area.str, ".INFILE");
    work_area.len = strlen(work_area.str);

    spi_err = SSPUTTKN (
        (short *)      spi_buff,          /* SPI buffer */
        (long)         ZBAT_TKN_IN_FILE, /* Token ID */
        (char *)       &work_area        /* Token value */
    );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essputtkn);

    /* Insert the out-file token */

    strcpy (work_area.str, "$S.#MASTERA");
    work_area.len = strlen(work_area.str);

    spi_err = SSPUTTKN (
        (short *)      spi_buff,          /* SPI buffer
*/
        (long)         ZBAT_TKN_OUT_FILE, /* Token ID
*/
        (char *)       &work_area        /* Token value
*/
    );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essputtkn);

    /* Set up ZBAT-MAP-DEF-JOB */
    /* Initialize the JOB_MAP for JOB SUBMIT to nulls */

    spi_err = SSNULL ((short *) zbat_map_def_job, (char *) &JOB );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essnull);

    /* Move a set of constant values into the job structure for submission
*/
    cptr = strncpy (JOB.zclassname.u_z_c.z_c, "CLASS-A
",24);
    JOB.zhold          = ZSPI_VAL_TRUE;
    JOB.zhold_after    = ZSPI_VAL_TRUE;
    JOB.zdefault_security = 04444; /* Octal 4444 = NNNN */
    (continued)

```

```

    spi_err = SSPUT ( (short *) spi_buff, (short *) zbat_map_def_job, (char
*) &JOB );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essput);

    /* Do WriteRead to scheduler */

    retcode = send_spi();
    if (retcode != 0)
        stopwitherror(retcode, (char *) &eretcode);
}

#pragma page
/*
-----
* status_job
* Use:
*     Perform status job on the job number passed
* Effects:
*     Sets SPI buffer to status job details if successful
*     Stops program otherwise
*-----
*/

void status_job(const int jobnum)
{
    int         retcode = 0;

    /* Initialize the SPI buffer          */

    spi_err = SSINIT (
        (short *) spi_buff                /* SPI buffer
    */
        , (ZBAT_VAL_BUFLen+6)            /* SPI buffer length
    */
        , (short *) &zbat_val_ssid        /* SSID - subsystem ID
    */
        , ZSPI_VAL_CMDHDR                 /* SPI buffer type
    */
        , ZBAT_CMD_STATUS                 /* Command number
    */
        , ZBAT_OBJ_JOB                    /* Object type
    */
        , 0                               /* Max-resp
    */
        ,                                  /* Server-vrsn
    */
        , ZSPI_VAL_TRUE                   /* Checksum enabled
    */

    );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essinit);
    /* Insert the job number into zbat_tkn_sel_job_number */
    (continued)

```

```

        spi_err = SSPUTTKN (
                                (short *) spi_buff,          /* SPI
buffer */
                                (long)   ZBAT_TKN_SEL_JOB_NUMBER, /* Token ID
*/
                                (char *)  &jobnum            /* Token
value */
                                );

        if (spi_err != ZSPI_ERR_OK)
            stopwitherror(spi_err, (char *) &essputtkn);

        /* Do WriteRead to scheduler */

        retcode = send_spi();
        if (retcode != 0)
            stopwitherror(retcode, (char *) &eretcode);
    }

#pragma page

/*
*-----
* closedown
* Use:
*     Part of finalization
* Effects:
*     Close scheduler for SPI I/O
*-----
*/

void closedown(void)
{
    FILE_CLOSE_ (schd_chan);
}

#pragma page

/*
*-----
* main
* Use:
*     Calls all required functions to complete task
* Effects:
*     Opens scheduler for SPI
*     Submits job
*     Status on job and print details
*     Closes scheduler for SPI
(continued)

```



```

*
*-----
*/

main()
{
    char    *cptr;
    int     jobnumber;
    zbat_ddl_netbatch_name_def  jobname;    /* NetBatch definition for job
name */

    /* Initialize the subsystem IDs */

    cptr = strncpy(zbat_val_ssid.u_z_fill.z_fill, ZSPI_VAL_TANDEM, 8);
    zbat_val_ssid.z_number = ZSPI_SSN_ZBAT;
    zbat_val_ssid.z_version = ZBAT_VAL_VERSION;

    /* malloc some memory for SPI buffer */

    spi_buff = (BATBUFDEF *) malloc (sizeof(BATBUFDEF));
    if (spi_buff == NULL)
        exit(EXIT_FAILURE);

    /* Open scheduler with #ZSPI */

    open_scheduler();

    /* Submit job to scheduler */

    submit_job();

    printf("\n");
    printf("    Job submitted to $ZBAT using SPI\n\n");

    /* Get the job-number token returned in the SPI buffer */

    spi_err =  SSGETTKN ( (short *)  spi_buff,
                        (long)      ZBAT_TKN_SEL_JOB_NUMBER,
                        (char *)    &jobnumber,
                        1
                    );

    if (spi_err != ZSPI_ERR_OK)
        stopwitherror(spi_err, (char *) &essgettkn);

    printf("                Job number : %d\n", jobnumber);

    (continued)

```

```

/* Get the job-name token returned in the SPI buffer */
spi_err = SSGETTKN ( (short *)      spi_buff,
                    (long)          ZBAT_TKN_SEL_JOBNAME,
                    (char *)        &work_area,
                    1
                );

if (spi_err != ZSPI_ERR_OK)
    stopwitherror(spi_err, (char *) &essgettkn);

cptr = strncpy (jobname.u_z_c.z_c, work_area.str, work_area.len);

printf("                Job name      : %s\n\n", jobname.u_z_c.z_c);
/* Perform a status job command on the job number and print details */
status_job(jobnumber);

printf("    Job status from $ZBAT using SPI\n\n");

/* Get the zbat_map_status_job map returned in the SPI buffer */
spi_err = SSGET ( (short *) spi_buff,
                 (short *) zbat_map_status_job,
                 (char *) &STATUS,
                 1
                );

if (spi_err != ZSPI_ERR_OK)
    stopwitherror(spi_err, (char *) &essget);

printf("                Selpri      : %d \n", STATUS.zselpri);
printf("                Class       : %s \n\n",
STATUS.zclassname.u_z_c.z_c);

/* Close scheduler for SPI comms */

closedown();
}

```

# Sample COBOL Program

[Example C-2](#) on page C-17 contains the source code for a sample COBOL program. The program submits a job to a scheduler, executes a STATUS JOB command on the submitted job, and displays some job details. The source code for the program is available in the file NBSPIEX in the NetBatch installation subvolume.

To run the sample program, follow these steps. The prerequisite to completing the procedure is a D21 or later scheduler named \$ZBAT running on the node where the program will run.

## Step 1: Copy the Source Code

Copy the source code for the sample COBOL program from file NBSPIEX to a new EDIT file:

```
> EDIT; GET $SYSGEN.ZNETBTCH.NBSPIEX 545/1093 PUT TEMP.COBSRC
TEXT EDITOR - T9601D20 - (01JUN93)
CURRENT FILE IS $DATA7.TEMP.COBSRC
```

## Step 2: Change ISV.ZSPIDEF References

Change ISV.ZSPIDEF references in the new file to specify the volume and subvolume containing the source-definition files ZSPICOB and ZBATCOB:

```
*LIST BOTH /ISV.ZSPIDEF/
  589      COPY ZBAT-DDL-MSG-BUFFER OF $ISV.ZSPIDEF.ZBATCOB.
  600      ?SOURCE $ISV.ZSPIDEF.ZBATCOB (ZBAT-TKN-SEL-JOB-
NUMBER,
  610      ?SOURCE $ISV.ZSPIDEF.ZSPICOB
*CHANGE /ISV/SYSGEN/ ALL
  589      COPY ZBAT-DDL-MSG-BUFFER OF
$SYSGEN.ZSPIDEF.ZBATCOB.
  600      ?SOURCE $SYSGEN.ZSPIDEF.ZBATCOB (ZBAT-TKN-SEL-JOB-
NUMBER,
  610      ?SOURCE $SYSGEN.ZSPIDEF.ZSPICOB
```

## Step 3: Change the SYSTEM.SYSTEM References

Change SYSTEM.SYSTEM references in the new file to specify the volume and subvolume containing the COBOL library and external-declaration files COBOLLIB and COBOLEX0:

```
*LIST BOTH /SYSTEM.SYSTEM.COB/
  547      ?SEARCH $SYSTEM.SYSTEM.COBOLLIB
  548      ?CONSULT $SYSTEM.SYSTEM.COBOLEX0
*CHANGE /SYSTEM.SYSTEM.COB/SYSTEM.SYS00.COB/ ALL
  547      ?SEARCH $SYSTEM.SYS00.COBOLLIB
  548      ?CONSULT $SYSTEM.SYS00.COBOLEX0
```

## Step 4: Change Class Name

Change class name CLASS-A to that of an existing class in \$ZBAT if class CLASS-A does not exist in that scheduler. Otherwise, add class CLASS-A to \$ZBAT.

```
*CHANGE /"CLASS-A"/"OPERATIONS"/ ALL
      904          MOVE "OPERATIONS"    TO ZCLASSNAME OF ZBAT-DDL-
DEF-JOB.
```

## Step 5: Change Job Name

Change job name MASTER-A to your own choice of name if MASTER-A conflicts with an existing production job in \$ZBAT. Otherwise, delete job MASTER-A from \$ZBAT.

```
*CHANGE /"MASTER-A"/"COBOL85-JOB"/ ALL
      813          STRING "COBOL85-JOB" DELIMITED BY SIZE
```

## Step 6: Compile the Source File

End the EDIT session and compile the source file:

```
*EXIT
> COBOL85 /IN TEMP.COBSRC/ TEMP.COBOBJ; SUPPRESS
```

## Step 7: Test the Program

Test the program by running the compiled object:

```
> RUN TEMP.COBOBJ
  Job submitted to $ZBAT using SPI
                Job number :      6
                Job name   : COBOL85-JOB
  Job status from $ZBAT using SPI
                Selpri    :      3
                Class     : OPERATIONS
```

## Step 8: Delete the Submitted Job

```
> BATCHCOM $ZBAT; DELETE JOB COBOL85-JOB
Job COBOL85-JOB Jobnumber 6 deleted
```

---

**Example C-2. Sample SPI COBOL Program**

```

?ENV COMMON
?COMPACT, SYMBOLS, INSPECT, SAVE STARTUP
?SEARCH $SYSTEM.SYSTEM.COBOLLIB
?CONSULT $SYSTEM.SYSTEM.COBOLEX0

?MAIN SPIEXCOB

    IDENTIFICATION DIVISION.

        PROGRAM-ID.          SPIEXCOB.
        DATE-WRITTEN.        November 1993.
        DATE-COMPILED.

    *
    * OVERVIEW:
    *#####
    *#
    #
    *# This COBOL source code compiles into a sample program
    #
    *# that demonstrates the subsystem programmatic interface (SPI)
    #
    *# to the NetBatch scheduler. The program performs two functions:
    #
    *#     * Submits a job to scheduler $ZBAT
    #
    *#     * Executes a STATUS JOB command on the submitted job
    #
    *#     and displays some of the job's details
    #
    *#####
    /
    ENVIRONMENT DIVISION.
    CONFIGURATION SECTION.
    SOURCE-COMPUTER.    T16.
    OBJECT-COMPUTER.    T16.
    INPUT-OUTPUT SECTION.
    FILE-CONTROL.

        SELECT SCHED-FILE
            ASSIGN TO "$ZBAT.#ZSPI"
            FILE STATUS IS W03-SCHED-STATUS.

/DATA DIVISION
DATA DIVISION.
FILE SECTION.

    FD SCHED-FILE
        LABEL RECORDS ARE OMITTED
        RECORD VARYING 6 TO 2048.

    COPY ZBAT-DDL-MSG-BUFFER OF $ISV.ZSPIDEF.ZBATCOB.

WORKING-STORAGE SECTION.
(continued)

```

---

```

*****
*   Working-storage constants
*
*****
/
?NOLIST
?SOURCE $ISV.ZSPIDEF.ZBATCOB ( ZBAT-TKN-SPI ,
?                               ZBAT-TKN-EMS ,
?
?                               ZBAT-DDL-NETBATCH-NAME ,
?                               ZBAT-MAP-STATUS-JOB ,
?                               ZBAT-DDL-STATUS-JOB ,
?
?                               ZBAT-VAL-SSID ,
?                               ZBAT-MAP-DEF-JOB ,
?
?                               ZBAT-DDL-DEF-JOB ,
?                               CONSTANTS )
/
?SOURCE $ISV.ZSPIDEF.ZSPICOB
?LIST
/
01  W02-WORK-AREA.
03  W02-PORTION                PIC X(30).
03  W02-TEXT                   PIC X(50).
03  W02-RESULT                 PIC S9(4) COMP.
03  W02-TRUE                   NATIVE-2 VALUE -1.

01  W03-WORK-AREA.
03  W03-DISPLAY-NUM           PIC ZZZ9.
03  W03-SCHED-STATUS          PIC XX VALUE "00".
03  W03-ERROR-TEXT           PIC X(75).

01  W05-SPI-WORK-AREA.
03  W05-STATUS                NATIVE-2.
03  W05-DATA-STRUCT.
05  W05-DATA-LEN              NATIVE-2.
05  W05-DATA-AREA             PIC X(30).
03  W05-TOKEN-RETCODE         NATIVE-2.
03  W05-TOKEN-JOBNUM          NATIVE-2.
03  W05-ASCII                 PIC X(5).

01  W07-ERROR-MESSAGES.
03  FILLER                    PIC X(75) VALUE
    "ERROR ON SCHEDULER $ZBAT.#ZSPI, FILE STATUS =:  ".
03  FILLER                    PIC X(75) VALUE
    "SSGET Failed                                     ".
03  FILLER                    PIC X(75) VALUE
    "SSINIT Failed                                    ".
(continued)

```

```

03  FILLER                                PIC X(75) VALUE
    "SSPUT Failed                        ".
03  FILLER                                PIC X(75) VALUE
    "GETSTARTUPTEXT Failed              ".
03  FILLER                                PIC X(75) VALUE
    "SSNULL Failed                     ".
03  FILLER                                PIC X(75) VALUE
    "NUMIN Failed                       ".
03  FILLER                                PIC X(75) VALUE
    "RETCODE indicated a NetBatch error ".

01  W07-ERROR-MESSAGES-ARRAY REDEFINES W07-ERROR-MESSAGES.
    03  W07-ERROR-TEXT          PIC X(75) OCCURS 8 TIMES.

PROCEDURE DIVISION.

/M A I N L I N E      S P I E X C O B

A000-MAINLINE SECTION.

    PERFORM A000-INIT.

    PERFORM A000-MAIN.

    CLOSE SCHED-FILE.

A000-EXIT.
    STOP RUN.

/
*****
A000-INIT SECTION.

#####
*#  This section:
#
*#  a) Gets the volume info from startup text
#
*#  b) Opens the scheduler
#
#####
*  Get startup text
    MOVE "VOLUME" TO W02-PORTION.
    MOVE SPACES TO W02-TEXT.
    MOVE 0 TO W02-RESULT.
    ENTER "GETSTARTUPTEXT" USING W02-PORTION,
                                W02-TEXT
                                GIVING W02-RESULT.

    IF W02-RESULT = -1
        MOVE 0 TO W05-STATUS
        CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(5)
    END-IF.
(continued)

```

```

* Open the scheduler
  OPEN I-O SCHED-FILE.
  IF W03-SCHED-STATUS NOT = "00"
    MOVE 0 TO W05-STATUS
    MOVE SPACES TO W03-ERROR-TEXT
    STRING "OPEN " DELIMITED BY SIZE
           W07-ERROR-TEXT(1) DELIMITED BY ":"
           " " DELIMITED BY SIZE
           W03-SCHED-STATUS DELIMITED BY SIZE
           INTO W03-ERROR-TEXT
    END-STRING
    CALL "SPIEXABN" USING W05-STATUS, W03-ERROR-TEXT
  END-IF.

A000-A90-EXIT.
  EXIT.

/
*****
A000-MAIN SECTION.

*#####
*# This section:
*#
*# a) Calls the routine to submit a job
*#
*# b) Retrieves job name and number tokens and displays on
*# the screen
*#
*# c) Calls the routine to get job status
*#
*# d) Displays job status details on the screen
*#
*#####

* Submit job
  PERFORM B000-SUBMIT-JOB.
* Get the job-number token and display on screen
  ENTER TAL "SSGET" USING ZBAT-DDL-MSG-BUFFER,
                      ZBAT-TKN-SEL-JOB-NUMBER,
                      W05-TOKEN-JOBNUM,
                      1
                      GIVING W05-STATUS.
  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(2)
  END-IF.

  MOVE W05-TOKEN-JOBNUM TO W03-DISPLAY-NUM.
  DISPLAY " ".
  DISPLAY " Job submitted to $ZBAT using SPI".
  DISPLAY " ".
  DISPLAY " Job number : " W03-DISPLAY-NUM.
  (continued)

```



```

* Get the job-name token and display on screen
  ENTER TAL "SSGET" USING ZBAT-DDL-MSG-BUFFER,
                        ZBAT-TKN-SEL-JOBNAME,
                        ZBAT-DDL-NETBATCH-NAME,
                        1
                        GIVING W05-STATUS.
  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(2)

  END-IF.
  DISPLAY "                Job name      :  " ZBAT-DDL-NETBATCH-NAME.

* Get the status of the job
  PERFORM B100-STATUS-JOB.

  ENTER TAL "SSGET" USING ZBAT-DDL-MSG-BUFFER,
                        ZBAT-MAP-STATUS-JOB,
                        ZBAT-DDL-STATUS-JOB,
                        1
                        GIVING W05-STATUS.
  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(2)
  END-IF.

* Output the job status
  MOVE ZSELPRI OF ZBAT-DDL-STATUS-JOB TO W03-DISPLAY-NUM.
  DISPLAY " ".
  DISPLAY "      Job status from $ZBAT using SPI".
  DISPLAY " ".
  DISPLAY "                Selpri      :  " W03-DISPLAY-NUM.
  DISPLAY "                Class       :  " ZCLASSNAME OF ZBAT-DDL-
STATUS-JOB.
  DISPLAY " ".

  A000-A99-EXIT.
  EXIT.

/
*****
  B000-SUBMIT-JOB  SECTION.

*#####
*#
#
*#  This section:
#
*#  a) Initializes the SPI buffer
#
*#  b) Inserts tokens necessary to submit the job
#
*#  c) Submits the job
#
*#
#
*#####
  (continued)

```

```

* Initialize the SPI buffer
  ENTER TAL "SSINIT" USING ZBAT-DDL-MSG-BUFFER,
                           ZBAT-VAL-BUFLen,
                           ZBAT-VAL-SSID,
                           ZSPI-VAL-CMDHDR,
                           ZBAT-CMD-SUBMIT,
                           ZBAT-OBJ-JOB,
                           0,
                           OMITTED,

  W02-TRUE

                           GIVING W05-STATUS.

  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(3)
  END-IF.

  DISPLAY "                Job name      :  " ZBAT-DDL-NETBATCH-NAME.

* Get the status of the job
  PERFORM B100-STATUS-JOB.

  ENTER TAL "SSGET" USING ZBAT-DDL-MSG-BUFFER,
                           ZBAT-MAP-STATUS-JOB,
                           ZBAT-DDL-STATUS-JOB,
                           1,
                           GIVING W05-STATUS.

  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(2)
  END-IF.

* Output the job status
  MOVE ZSELPRI OF ZBAT-DDL-STATUS-JOB TO W03-DISPLAY-NUM.
  DISPLAY " ".
  DISPLAY "   Job status from $ZBAT using SPI".
  DISPLAY " ".
  DISPLAY "                Selpri       :  " W03-DISPLAY-NUM.
  DISPLAY "                Class       :  " ZCLASSNAME OF ZBAT-DDL-
STATUS-JOB.
  DISPLAY " ".

  A000-A99-EXIT.
  EXIT.

/
*****
B000-SUBMIT-JOB  SECTION.

(continued)

```

```

#####
*# This section:
#
*# a) Initializes the SPI buffer
#
*# b) Inserts tokens necessary to submit the job
#
*# c) Submits the job
#
#####

* Initialize the SPI buffer
  ENTER TAL "SSINIT" USING ZBAT-DDL-MSG-BUFFER,
                           ZBAT-VAL-BUFLLEN,
                           ZBAT-VAL-SSID,
                           ZSPI-VAL-CMDHDR,
                           ZBAT-CMD-SUBMIT,
                           ZBAT-OBJ-JOB,
                           0,
                           OMITTED,
                           W02-TRUE
                           GIVING W05-STATUS.

  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(3)
  END-IF.

* Insert job-name token

  MOVE 1 TO W05-DATA-LEN.
  MOVE SPACES TO W05-DATA-AREA.
  STRING "MASTER-A" DELIMITED BY SIZE
                        INTO W05-DATA-AREA
                        WITH POINTER W05-DATA-LEN

  END-STRING.
  SUBTRACT 1 FROM W05-DATA-LEN.

  ENTER TAL "SSPUT" USING ZBAT-DDL-MSG-BUFFER,
                           ZBAT-TKN-SEL-JOBNAME,
                           W05-DATA-STRUCT

                           GIVING W05-STATUS.

  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(4)
  END-IF.

* Insert executor-program token
  MOVE 1 TO W05-DATA-LEN.
  MOVE SPACES TO W05-DATA-AREA.
  STRING "$SYSTEM.SYSTEM.TACL" DELIMITED BY SIZE
                                INTO W05-DATA-AREA
                                WITH POINTER W05-DATA-LEN

  END-STRING.
  SUBTRACT 1 FROM W05-DATA-LEN.
(continued)

```

```

ENTER TAL "SSPUT" USING ZBAT-DDL-MSG-BUFFER,
                        ZBAT-TKN-EXECUTOR-PROGRAM,
                        W05-DATA-STRUCT
                        GIVING W05-STATUS.
IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(4)
END-IF.

* Insert volume-subvol token
MOVE 1      TO W05-DATA-LEN.
MOVE SPACES TO W05-DATA-AREA.
STRING W02-TEXT DELIMITED BY SPACE
                        INTO W05-DATA-AREA
                        WITH POINTER W05-DATA-LEN

END-STRING.
SUBTRACT 1 FROM W05-DATA-LEN.

ENTER TAL "SSPUT" USING ZBAT-DDL-MSG-BUFFER,
                        ZBAT-TKN-VOLUME-SUBVOL,
                        W05-DATA-STRUCT
                        GIVING W05-STATUS.

IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(4)
END-IF.

* Insert in-file token
MOVE 1      TO W05-DATA-LEN.
MOVE SPACES TO W05-DATA-AREA.
STRING W02-TEXT DELIMITED BY SPACE
        ".INFILE" DELIMITED BY SIZE
                        INTO W05-DATA-AREA
                        WITH POINTER W05-DATA-LEN

END-STRING.
SUBTRACT 1 FROM W05-DATA-LEN.

ENTER TAL "SSPUT" USING ZBAT-DDL-MSG-BUFFER,
                        ZBAT-TKN-IN-FILE,
                        W05-DATA-STRUCT
                        GIVING W05-STATUS.

IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(4)
END-IF.

* Insert out-file token
MOVE 1      TO W05-DATA-LEN.
MOVE SPACES TO W05-DATA-AREA.
STRING "$S.#MASTERA" DELIMITED BY SIZE
(continued)

```

```

                                INTO W05-DATA-AREA
                                WITH POINTER W05-DATA-LEN
END-STRING.
SUBTRACT 1 FROM W05-DATA-LEN.
ENTER TAL "SSPUT" USING ZBAT-DDL-MSG-BUFFER,
                        ZBAT-TKN-OUT-FILE,
                        W05-DATA-STRUCT
                        GIVING W05-STATUS.
IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(4)
END-IF.

* Initialize the job structure
ENTER TAL "SSNULL" USING ZBAT-MAP-DEF-JOB,
                        ZBAT-DDL-DEF-JOB
                        GIVING W05-STATUS.
IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(6)
END-IF.
MOVE "CLASS-A" TO ZCLASSNAME OF ZBAT-DDL-DEF-JOB.

* High values denotes a 'true' condition
MOVE HIGH-VALUES TO ZHOLD,
                        ZHOLD-AFTER.

* Default security - base 8 = octal
MOVE "4444 " TO W05-ASCII.
ENTER TAL "NUMIN" USING W05-ASCII,
                        ZDEFAULT-SECURITY,
                        8,
                        W05-STATUS.
IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(7)
END-IF.

* Put job structure details into the SPI buffer
ENTER TAL "SSPUT" USING ZBAT-DDL-MSG-BUFFER,
                        ZBAT-MAP-DEF-JOB,
                        ZBAT-DDL-DEF-JOB
                        GIVING W05-STATUS.
IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(4)
END-IF.

* Submit the job
PERFORM C000-SEND-SPI.

(continued)

```

```

      B000-B99-EXIT.
      EXIT.
/
*****
      B100-STATUS-JOB SECTION.
*****
*# This section:
*#
*# a) Initializes the SPI buffer
*#
*# b) Inserts tokens necessary to get job status
*#
*# c) Gets job status information
*#
*****
* Initialize the SPI buffer
      ENTER TAL "SSINIT" USING ZBAT-DDL-MSG-BUFFER,
                                ZBAT-VAL-BUFLLEN,
                                ZBAT-VAL-SSID,
                                ZSPI-VAL-CMDHDR,
                                ZBAT-CMD-STATUS,
                                ZBAT-OBJ-JOB,
                                0,
                                OMITTED,
                                W02-TRUE
                                GIVING W05-STATUS.
      IF W05-STATUS NOT = 0
          CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(3)
      END-IF.

* Insert job number token
      ENTER TAL "SSPUT" USING ZBAT-DDL-MSG-BUFFER,
                                ZBAT-TKN-SEL-JOB-NUMBER,
                                W05-TOKEN-JOBNUM
                                GIVING W05-STATUS.
      IF W05-STATUS NOT = 0
          CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(4)
      END-IF.

* Get job status
      PERFORM C000-SEND-SPI.

      B100-B99-EXIT.
      EXIT.
/
*****
      C000-SEND-SPI SECTION.
*****
*# This section:
*#
*# a) Sends requests to the scheduler
*#
*# b) Interprets the responses from the scheduler
*#
*# c) Checks for errors from the scheduler
*#
*****
      (continued)

```

```

* Write the SPI buffer to the scheduler
  READ SCHED-FILE
    WITH PROMPT ZBAT-DDL-MSG-BUFFER
  END-READ.

  IF W03-SCHED-STATUS NOT = "00"
    MOVE 0 TO W05-STATUS
    MOVE SPACES TO W03-ERROR-TEXT
    STRING "READ "          DELIMITED BY SIZE

                          W07-ERROR-TEXT(1) DELIMITED BY ":"
                          " "              DELIMITED BY SIZE
    W03-SCHED-STATUS      DELIMITED BY SIZE
                          INTO W03-ERROR-TEXT

    END-STRING
    CALL "SPIEXABN" USING W05-STATUS, W03-ERROR-TEXT
  END-IF.

* Read tokens returned in SPI buffer
  ENTER TAL "SSGET" USING ZBAT-DDL-MSG-BUFFER,
                      ZSPI-TKN-RETCODE,
                      W05-TOKEN-RETCODE,
                      1
                      GIVING W05-STATUS.

  IF W05-STATUS NOT = 0
    CALL "SPIEXABN" USING W05-STATUS, W07-ERROR-TEXT(2)
  END-IF.

* Check for errors from scheduler
  IF W05-TOKEN-RETCODE NOT = 0
    CALL "SPIEXABN" USING W05-TOKEN-RETCODE, W07-ERROR-TEXT(8)
  END-IF.

C000-C99-EXIT.
EXIT.

END PROGRAM SPIEXCOB.

*****
*                               * END PROGRAM SPIEXCOB *
*****
(continued)

```

```

/
*****
*           * START PROGRAM SPIEXABN *
*
*****

IDENTIFICATION DIVISION.

PROGRAM-ID.          SPIEXABN.
DATE-WRITTEN.        16/11/93.
DATE-COMPILED.

*
* OVERVIEW:
*#####
*#
*# This program is the error handling routine for SPIEXCOB.
*#
*# It is passed an error number and error text, and then
*#
*# abends the program.
*#
*#
*#
*#####

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.

DATA DIVISION.
FILE SECTION.

WORKING-STORAGE SECTION.

LINKAGE SECTION.

01  W01-ERROR-CODE          NATIVE-2.
01  W01-ERROR-TEXT         PIC X(75).

PROCEDURE DIVISION USING  W01-ERROR-CODE,
                          W01-ERROR-TEXT.

*M A I N L I N E      S P I E X A B N
*****
A000-MAINLINE SECTION.

        PERFORM A000-PROCESS-ERROR.

A000-EXIT.
        STOP RUN.

(continued)

```



```
*****
A000-PROCESS-ERROR  SECTION.

    ENTER TAL "PROCESS_STOP_" USING OMITTED,
                                   OMITTED,
                                   OMITTED,
                                   2,
                                   W01-ERROR-CODE,
                                   OMITTED,
                                   W01-ERROR-TEXT.

A000-A00-EXIT.
EXIT.
*****
```

# Sample TACL Macros

[Example C-3](#) on page C-32 contains the source code for sample TACL macros. The macros submit a job to a scheduler, execute a STATUS JOB command on the submitted job, and display some of the job's details. The source code for the macros is available in the file NBSPIEX in the NetBatch installation subvolume.

To run the sample macros, follow these steps. The prerequisite to completing the procedure is a D21 or later scheduler named \$ZBAT running on the node where the macros will run.

## Step 1: Copy the Source Code

Copy the source code for the sample TACL macros from file NBSPIEX to a new EDIT file:

```
> EDIT; GET $SYSGEN.ZNETBTCH.NBSPIEX 1101/1517 PUT
TEMP.TACLSRC
TEXT EDITOR - T9601D20 - (01JUN93)
CURRENT FILE IS $DATA7.TEMP.TACLSRC
```

## Step 2: Change ISV.ZSPIDEF References

Change ISV.ZSPIDEF references in the new file to specify the volume and subvolume containing the source-definition files ZSPISEGF and ZBATSEGF:

```
*LIST BOTH /ISV.ZSPIDEF/
1116      ATTACHSEG SHARED $ISV.ZSPIDEF.ZBATSEGF :zbat
1120      ATTACHSEG SHARED $ISV.ZSPIDEF.ZSPISEGF :spi
*CHANGE /ISV/SYSGEN/ ALL
1116      ATTACHSEG SHARED $SYSGEN.ZSPIDEF.ZBATSEGF :zbat
1120      ATTACHSEG SHARED $SYSGEN.ZSPIDEF.ZSPISEGF :spi
```

## Step 3: Change Class Name

Change class name CLASS-A to that of an existing class in \$ZBAT if class CLASS-A does not exist in that scheduler. Otherwise, add class CLASS-A to \$ZBAT.

```
*CHANGE /CLASS-A/OPERATIONS/ ALL
1351      #SET zbat^job^def:zclassname:z^c      OPERATIONS
```

## Step 4: Change Job Name

Change job name MASTER-A to your own choice of name if MASTER-A conflicts with an existing production job in \$ZBAT. Otherwise, delete job MASTER-A from \$ZBAT.

```
*CHANGE /MASTER-A/TACL-JOB/ ALL
1260      #SET data^struct:data^area TACL-JOB
```

## Step 5: Load Macros From Source File

End the EDIT session and load the macros from the source file you created at Step 1:

```
*EXIT
> LOAD /KEEP 1/ TEMP.TACLSRC
Loaded from $DATA7.TEMP.TACLSRC:
SPIEXTACL SUBMIT^JOB STATUS^JOB SEND^SPI WAIT^FOR^IT
```

## Step 6: Test the Macros

Test the macros by invoking SPIEXTACL:

```
> SPIEXTACL
  Job submitted to $ZBAT using SPI
                Job number : 7
                Job name   : TACL-JOB
  Job status from $ZBAT using SPI
                Selpri     : 3
                Class      : OPERATIONS
```

## Step 7: Delete the Submitted Job

```
> BATCHCOM $ZBAT; DELETE JOB TACL-JOB
Job TACL-JOB Jobnumber 7 deleted
```

---

**Example C-3. Sample SPI TACL Macros**

```
?SECTION spiextacl ROUTINE

== ##### ==
== #
# ==
== # This TACL macro demonstrates the subsystem programmatic
# ==
== # interface (SPI) to the NetBatch scheduler. The macro performs
# ==
== # two functions:
# ==
== # * Submits a job to scheduler $ZBAT
# ==
== # * Executes a STATUS JOB command on the submitted job
# ==
== # and displays some of the job's details
# ==
== #
# ==
== ##### ==

== Load the TACL segment files if they aren't already loaded
[#IF NOT [#VARIABLEINFO /EXISTENCE/ :zbat] |THEN|
    ATTACHSEG SHARED $ISV.ZSPIDEF.ZBATSEGF :zbat
    #SET #USELIST :zbat [#USELIST]
]
[#IF NOT [#VARIABLEINFO /EXISTENCE/ :spi ] |THEN|
    ATTACHSEG SHARED $ISV.ZSPIDEF.ZSPISEGF :spi
    #SET #USELIST :spi [#USELIST]
]

#FRAME

== Push the variables to be used by routines
#PUSH schd
#SET schd $ZBAT
#PUSH zbat^ss^status, ss^error
#PUSH zbat^error^var, zbat^data^var, zbat^prompt^var
#PUSH zbat^all^done, zbat^wait^for^it^result, zbat^return^error
#PUSH work^area

== Define the structures used by routines

[#DEF zbat^spi^buffer STRUCT LIKE zbat^ddl^msg^buffer;]
[#DEF zbat^job^def STRUCT LIKE zbat^ddl^def^job;]

[#DEF zbat^job^status STRUCT LIKE zbat^ddl^status^job;]

[#DEF zbat^retcode STRUCT
    BEGIN
        INT retcode;
    END;
]
[#DEF zbat^jobnumber STRUCT
    BEGIN
        (continued)
    ]
]
```

---

```

    INT jobnumber;
    END;
]

== Define work-area data structure
[#DEF data^struct STRUCT
    BEGIN
    INT data^len;
    STRUCT data^area;
    BEGIN
    CHAR BYTE(0:23);
    END;
    END;
]

== Submit a job
#SET ss^error [submit^job]
[#IF ss^error |THEN|
    #RETURN
]

== Get job-number token and display on screen
#SETMANY zbat^ss^status, [#SSGETV /INDEX 1/ zbat^spi^buffer
                                zbat^tkn^sel^job^number
                                zbat^jobnumber
                                ]
[#IF zbat^ss^status |THEN|
    #OUTPUT Termination Info: [zbat^ss^status]
    #OUTPUT SSGETV Failed
    #RETURN
]

#OUTPUT
#OUTPUT /COLUMN 4/ Job submitted to [schd] using SPI
#OUTPUT
#OUTPUT /COLUMN 17/ Job number : [zbat^jobnumber:jobnumber]

== Get job-name token and display on screen
#SETMANY zbat^ss^status, [#SSGETV /INDEX 1/ zbat^spi^buffer
                                zbat^tkn^sel^jobname
                                data^struct
                                ]
[#IF zbat^ss^status |THEN|
    #OUTPUT Termination Info: [zbat^ss^status]
    #OUTPUT SSGETV Failed
    #RETURN
]
]
(continued)

```

```

#OUTPUT /COLUMN 17/ Job name      : [data^struct:data^area]

== Get the status of the job
#SET ss^error [status^job]
[#IF ss^error | THEN|
    #RETURN
]

#SETMANY zbat^ss^status, [#SSGETV /INDEX 1/ zbat^spi^buffer
                                zbat^map^status^job
                                zbat^job^status

                                ]
[#IF zbat^ss^status | THEN|
    #OUTPUT Termination Info: [zbat^ss^status]
    #OUTPUT SSGETV Failed
    #RETURN
]

== Display the status of the job on screen
#OUTPUT
#OUTPUT /COLUMN 4/ Job status from [sched] using SPI
#OUTPUT
#OUTPUT /COLUMN 17/ Selpri      : [zbat^job^status:zselpri]
#OUTPUT /COLUMN 17/ Class      : [zbat^job^status:zclassname:z^c]
#OUTPUT

== Close the scheduler
#SET req^error [#REQUESTER /WAIT [zbat^val^buflen]/
                                CLOSE zbat^error^var
                                ]
[#IF req^error | THEN|
    #OUTPUT Termination Info: [req^error]
    #OUTPUT REQUESTER close error
]

#UNFRAME

=====
?SECTION submit^job ROUTINE

== ##### ==
==
==
== This routine:
==
== a) Initializes the SPI buffer
==
== b) Inserts tokens necessary to submit the job
==
== c) Submits the job
==
== ##### ==

(continued)

```

```

== Initialize the SPI buffer
#SET zbat^ss^status [#SSINIT zbat^spi^buffer
                        [zbat^val^ssid]
                        [zbat^cmd^submit]
                        /OBJECT [zbat^obj^job], CHECKSUM -1/
]
[#IF zbat^ss^status THEN|
  #OUTPUT Termination Info: [zbat^ss^status]
  #OUTPUT SSINIT Failed
  #RESULT -1

  #UNFRAME

  #RETURN
]
== Insert job-name token
#SET data^struct:data^area MASTER-A
#SET work^area [data^struct:data^area]
#SET data^struct:data^len [#CHARCOUNT work^area]
#SET zbat^ss^status [#SSPUTV zbat^spi^buffer
                        zbat^tkn^sel^jobname
                        data^struct
]
[#IF zbat^ss^status THEN|
  #OUTPUT Termination Info: [zbat^ss^status]
  #OUTPUT SSPUTV Failed
  #RESULT -1

  #UNFRAME

  #RETURN
]
== Insert executor-program token
#SET data^struct:data^area $SYSTEM.SYSTEM.TACL
#SET work^area [data^struct:data^area]
#SET data^struct:data^len [#CHARCOUNT work^area]
#SET zbat^ss^status [#SSPUTV zbat^spi^buffer
                        zbat^tkn^executor^program
                        data^struct
]
[#IF zbat^ss^status THEN|
  #OUTPUT Termination Info: [zbat^ss^status]
  #OUTPUT SSPUTV Failed
  #RESULT -1
  #RETURN
]

== Insert volume-subvol token
#SET data^struct:data^area [#DEFAULTS]
#SET work^area [data^struct:data^area]
(continued)

```

```

#SET data^struct:data^len [#CHARCOUNT work^area]
#SET zbat^ss^status [#SSPUTV zbat^spi^buffer
                        zbat^tkn^volume^subvol
                        data^struct
                    ]
[#IF zbat^ss^status | THEN|
    #OUTPUT Termination Info: [zbat^ss^status]
    #OUTPUT SSPUTV Failed
    #RESULT -1
    #RETURN
]

== Insert in-file token
#PUSH infile^name
#SET work^area
#SET infile^name .INFILE
#CHARINS work^area 1 [#DEFAULTS]
#CHARINSV work^area ([#CHARCOUNT work^area]+1) infile^name
#SET data^struct:data^area [work^area]
#SET data^struct:data^len [#CHARCOUNT work^area]
#SET zbat^ss^status [#SSPUTV zbat^spi^buffer
                        zbat^tkn^in^file
                        data^struct
                    ]
[#IF zbat^ss^status | THEN|
    #OUTPUT Termination Info: [zbat^ss^status]
    #OUTPUT SSPUTV Failed
    #RESULT -1
    #RETURN
]

== Insert out-file token
#SET data^struct:data^area $.#MASTERA
#SET work^area [data^struct:data^area]
#SET data^struct:data^len [#CHARCOUNT work^area]
#SET zbat^ss^status [#SSPUTV zbat^spi^buffer
                        zbat^tkn^out^file
                        data^struct
                    ]
[#IF zbat^ss^status | THEN|
    #OUTPUT Termination Info: [zbat^ss^status]
    #OUTPUT SSPUTV Failed
    #RESULT -1
    #RETURN
]

== Initialize the job structure
#SET zbat^ss^status [#SSNULL zbat^map^def^job
                        zbat^job^def
                    ]

(continued)

```



```

[#IF zbat^ss^status |THEN|
  #OUTPUT Termination Info: [zbat^ss^status]
  #OUTPUT SSNULL Failed
  #RESULT -1
  #RETURN
]

== Set job attributes
#SET zbat^job^def:zclassname:z^c      CLASS-A
#SET zbat^job^def:zhold                -1          == True
#SET zbat^job^def:zhold^after          -1          == True
#SET zbat^job^def:zdefault^security 2340          == Security "NNNN"

== Put job structure details into SPI buffer
#SET zbat^ss^status [#SSPUTV zbat^spi^buffer
                    zbat^map^def^job
                    zbat^job^def
                    ]

[#IF zbat^ss^status |THEN|
  #OUTPUT Termination Info: [zbat^ss^status]
  #OUTPUT SSPUTV Failed
  #RESULT -1
  #RETURN
]

== Submit the job
#SET ss^error [send^spi]
[#IF ss^error |THEN|
  #RESULT -1
|ELSE|
  #RESULT 0
]

=====
?SECTION status^job ROUTINE
== ##### ==
== This routine:
==
== a) Initializes the SPI buffer
==
== b) Inserts tokens necessary to get job status
==
== c) Gets job status information
==
== ##### ==

== Initialize the SPI buffer
#SET zbat^ss^status [#SSINIT zbat^spi^buffer
                    [zbat^val^ssid]
                    [zbat^cmd^status]
                    /OBJECT [zbat^obj^job], CHECKSUM -1/
                    ]
(continued)

```

```

[#IF zbat^ss^status |THEN|
  #OUTPUT Termination Info: [zbat^ss^status]
  #OUTPUT SSINIT Failed
  #RESULT -1
  #RETURN
]
== Insert job-number token
#SET zbat^ss^status [#SSPUTV zbat^spi^buffer
                        zbat^tkn^sel^job^number
                        zbat^jobnumber
]
[#IF zbat^ss^status |THEN|
  #OUTPUT Termination Info: [zbat^ss^status]
  #OUTPUT SSPUTV Failed
  #RESULT -1
  #RETURN
]

== Get job status
#SET zbat^all^done 0
[#LOOP |DO|
  wait^for^it
  |UNTIL| zbat^all^done
]
[#IF zbat^wait^for^it^result |THEN|
  #OUTPUT Termination Info: [zbat^wait^for^it^result]
  #OUTPUT REQUESTER read error
  #RESULT -1
  #RETURN
]

#RESULT 0

=====
?SECTION send^spi ROUTINE

== ##### ==
== This routine:
==
== a) Sends requests to the scheduler
==
== b) Interprets the responses from the scheduler
==
== c) Checks for errors from the scheduler
==
== ##### ==

#PUSH req^error

== Write the SPI buffer to the scheduler
  (continued)

```

```

#SET req^error [#REQUESTER /WAIT [zbat^val^buflen]/ READ
                                [schd].#ZSPI
                                zbat^error^var
                                zbat^data^var
                                zbat^prompt^var

]
[#IF req^error |THEN|
    #OUTPUT Termination Info: [req^error]
    #OUTPUT REQUESTER open/read error
    #RESULT -1
    #RETURN
]

== Wait for response from the scheduler
#SET zbat^all^done 0

[#LOOP |DO|
    wait^for^it
    |UNTIL| zbat^all^done
]
[#IF zbat^wait^for^it^result |THEN|
    #OUTPUT Termination Info: [zbat^wait^for^it^result]
    #OUTPUT REQUESTER read error
    #RESULT -1
    #RETURN
]

== Read tokens returned in SPI buffer
#SETMANY zbat^ss^status, [#SSGETV /INDEX 1/ zbat^spi^buffer
                                zspi^tkn^retcode
                                zbat^retcode
                                ]
[#IF zbat^ss^status |THEN|
    #OUTPUT Termination Info: [zbat^ss^status]
    #OUTPUT SSGETV Failed
    #RESULT -1
    #RETURN
]

== Check for scheduler errors
[#IF zbat^retcode:retcode |THEN|
    #OUTPUT Termination Info: [zbat^retcode:retcode]
    #OUTPUT RETCODE indicated a NetBatch error
    #RESULT -1
    #RETURN
]

#RESULT 0
(continued)

```

```

=====
?SECTION wait^for^it MACRO

== ##### ==
==
== This macro:
==
== a) Interprets the responses from the scheduler
==
== ##### ==

#SET zbat^wait^for^it^result 0
== Get the data returned from the scheduler
[#CASE [#VARIABLEINFO /VARIABLE/ [#WAIT zbat^error^var
                                         zbat^data^var
                                         zbat^prompt^var ]]

    |zbat^error^var|    #OUTPUT Termination Info: [zbat^error^var]
                        #OUTPUT Scheduler [schd].#ZSPI read error
                        #SET zbat^all^done 1
                        #SET zbat^wait^for^it^result [zbat^error^var]

    |zbat^data^var|    #EXTRACTV zbat^data^var zbat^spi^buffer
                        #SET zbat^all^done 1

    |zbat^prompt^var|  #APPENDV zbat^prompt^var zbat^spi^buffer
]

```

# Sample TAL Program

[Example C-4](#) on page C-42 contains the source code for a sample TAL program. The program submits a job to a scheduler, executes a STATUS JOB command on the submitted job, and displays some job details. The source code for the program is available in the file NBSPIEX in the NetBatch installation subvolume.

To run the sample program, follow these steps. The prerequisite to completing the procedure is a D21 or later scheduler named \$ZBAT running on the node where the program will run.

## Step 1: Copy the Source Code

Copy the source code for the sample TAL program from file NBSPIEX to a new EDIT file:

```
> EDIT; GET $SYSGEN.ZNETBTCH.NBSPIEX 1525/1826 PUT
TEMP.TALSRC
TEXT EDITOR - T9601D20 - (01JUN93)
CURRENT FILE IS $DATA7.TEMP.TALSRC
```

## Step 2: Change ISV.ZSPIDEF References

Change ISV.ZSPIDEF references in the new file to specify the volume and subvolume containing the source-definition files ZSPITAL and ZBATTAL:

```
*LIST BOTH /ISV.ZSPIDEF/
1529      ?NOLIST, SOURCE $ISV.ZSPIDEF.ZSPITAL
1532      ?NOLIST, SOURCE $ISV.ZSPIDEF.ZBATTAL
*CHANGE /ISV/SYSGEN/ ALL
1529      ?NOLIST, SOURCE $SYSGEN.ZSPIDEF.ZSPITAL
1532      ?NOLIST, SOURCE $SYSGEN.ZSPIDEF.ZBATTAL
```

## Step 3: Change Class Name

Change class name CLASS-A to that of an existing class in \$ZBAT if class CLASS-A does not exist in that scheduler. Otherwise, add class CLASS-A to \$ZBAT.

```
*CHANGE /"CLASS-A" /"OPERATIONS"
"/ ALL
1674      job.zclassname      ' := ' "OPERATIONS
"; -- Class name
```

## Step 4: Change Job Name

Change job name MASTER-A to your own choice of name if MASTER-A conflicts with an existing production job in \$ZBAT. Otherwise, delete job MASTER-A from \$ZBAT.

```
*CHANGE /"MASTER-A"/"TAL-JOB"/ ALL
1639      work^area ' := ' "TAL-JOB" -> @ptr;
```

## Step 5: Compile the Source File

End the EDIT session and compile the source file:

```
*EXIT
> TAL /IN TEMP.TALSRC/ TEMP.TALOBJ; SUPPRESS
```

## Step 6: Test the Program

Test the program by running the compiled object:

```
> RUN TEMP.TALOBJ
Job submitted to $ZBAT using SPI
      Job number : 008
      Job name   : TAL-JOB
Job status from $ZBAT using SPI
      Selpri    : 3
      Class     : OPERATIONS
```

## Step 7: Delete the Submitted Job

```
> BATCHCOM $ZBAT; DELETE JOB TAL-JOB
Job TAL-JOB Jobnumber 8 deleted
```

---

### Example C-4. Sample SPI TAL Program

```
?SYMBOLS, INSPECT
?HIGHPIN, HIGHREQUESTERS

-- SPI standard definitions
?NOLIST, SOURCE $ISV.ZSPIDEF.ZSPITAL

-- NetBatch TAL definitions
?NOLIST, SOURCE $ISV.ZSPIDEF.ZBATTAL

?LIST
-- Defines
DEFINE stopwitherror(t,l) =
    CALL PROCESS_STOP_(,,,2,error,,t:l) #;

-- Startup-message offsets
LITERAL _defaults = 1;
LITERAL _in       = 1+8;
LITERAL _out      = 1+8+12;
LITERAL _text     = (1+8+12+12)*2;
--
LITERAL spi_occurs = 2;
LITERAL max_file_length = 24;
LITERAL maxfilewords = 20;
LITERAL maxfilebytes = 40;
--
(continued)
```

---

```

-- Global declarations
INT      .spi^buf[0:((zbat^val^buflen^min+6)/2)];
INT      .zbat^val^ssid^init[0:5] := [zspi^val^tandem, zspi^ssn^zbat,
                                      zbat^val^version];
INT      work^length[0:Maxfilewords]; -- Common work buffer
STRING   work^area = work^length[1];
INT      .temp[0:11];
INT      .sm[0:66];                  -- Startup message buffer word
pointer   .ssm := @sm '<<' 1;        -- Startup message buffer byte
pointer
INT      recv_chan;                  -- Channel for $receive
INT      out_chan;                   -- Channel for OUT file
INT      schd_chan;                  -- Channel for Scheduler

-- Error text for system-procedure-call errors
STRING   .essget[0:11]               := "SSGET Failed";
STRING   .essput[0:11]               := "SSPUT Failed";
STRING   .essinit[0:12]              := "SSINIT Failed";
STRING   .essnull[0:12]              := "SSNULL Failed";
STRING   .efile_open_[0:16]          := "FILE_OPEN_ Failed";
STRING   .ewrite_readx_[0:16]        := "WRITEREADX Failed";
STRING   .eretcod[0:33]              := "RETCODE indicated a NetBatch error";
--
-- Structures definition
STRUCT   .job(zbat^ddl^def^job^def); -- Job definition structure
--
-- SYSTEM procedures library
?NOLIST, SOURCE $system.system.extdecs0
?LIST
-----

-- SEND^SPI function procedure.
-- This function procedure sends a formatted SPI buffer to the
-- scheduler and returns the ERROR or WARNING message (RETCODE)
-- from the scheduler.
-----

INT PROC SEND^SPI;

BEGIN
INT      error;
INT      retcode;

-- Write the SPI buffer to the scheduler
CALL WRITEREADX(schd_chan
                , spi^buf
                , (spi^buf[spi_occurs]+6)
                , zbat^val^buflen^min);

IF <> THEN
BEGIN
(continued)

```

```

    CALL FILE_GETINFO_(schd_chan, error);
    stopwitherror(ewrite_readx_, 16);
END;

-- Read tokens returned in SPI buffer
IF error := SSGETTKN(spi^buf
                    , zspi^tkn^retcode
                    , retcode
                    , 1) THEN
    stopwitherror(essget, 12);

RETURN retcode;
END;
-----

-- SUBMIT^JOB function procedure.
-- This function procedure formats the SPI buffer for NetBatch
-- and submits a job to $ZBAT. The job only has the IN, OUT,
-- EXECUTOR-PROGRAM, CLASS, VOLUME, HOLD, and HOLDAFTER
-- attributes.
-----

INT  PROC SUBMIT^JOB;

BEGIN
INT      error;                                -- General error variables
INT      job^map[0:zbat^map^def^job^wln] := zbat^map^def^job;
STRING   .ptr;                                -- Work pointer

-- Initialize the SPI buffer
if (error := SSINIT( spi^buf
                    , (zbat^val^buflen^min+6)
                    , zbat^val^ssid^init
                    , zspi^val^cmdhdr
                    , zbat^cmd^submit           -- Submit command token
                    , zbat^obj^job             -- Job object token
                    , 0                         -- MAXRESP
                    ,
                    , zspi^val^true)) then      -- Enable checksum
    stopwitherror(essinit, 13);

-- Insert job-name token
work^area := "MASTER-A" -> @ptr;
work^length := @ptr '-' @work^area;
IF (error := SSPUTTKN(spi^buf, zbat^tkn^sel^jobname, work^length)) THEN

--    CALL PROCESS_STOP_(, , 2!cc!, error!ti!, , essput:12);
    stopwitherror(essput, 12);

-- Insert executor-program token
work^area := "$SYSTEM.SYSTEM.TACL" -> @ptr;
    (continued)

```



```

work^length := @ptr '-' @work^area;
IF (error := SSPUTTKN(spi^buf, zbat^tkn^executor^program, work^length))
THEN
    stopwitherror(essput, 12);

-- Insertvolume-subvol token
temp := sm[_defaults] for 8 & "INFILE ";-- Get <vol>.<subvol> from
startup-message
error :=
    OLDFILENAME_TO_FILENAME_(temp, work^area:maxfilebytes,
                             work^length);-- C-series filename to D-
series format
work^length := work^length - 7;
IF (error := SSPUTTKN(spi^buf, ZBAT^TKN^VOLUME^SUBVOL, work^length)) THEN
    stopwitherror(essput, 12);

! Insert in-file token
work^length := work^length + 7;                -- Use data already in buffer
IF (error := SSPUTTKN(spi^buf, ZBAT^TKN^IN^FILE, work^length)) THEN
    stopwitherror(essput, 12);

! Insert out-file token
work^area := '$S.#MASTERA' -> @ptr;
work^length := @ptr '-' @work^area;
IF (error := SSPUTTKN(spi^buf, ZBAT^TKN^OUT^FILE, work^length)) THEN
    stopwitherror(essput, 12);

-- Initialize the ZBAT-MAP-DEF-JOB structure
IF (error := SSNULL(job^map, job)) THEN
    stopwitherror(essnull, 13);
job.zclassname      := ' "CLASS-A                "; -- Class name
job.zhold            :=                          -- Hold ON
job.zhold^after      :=  zspi^val^true;          -- Holdafter ON
job.zdefault^security := %4444;                  -- Security NNNN

-- Put the job definition structure details into the SPI buffer
IF error :=  SSPUT (spi^buf                -- SPI buffer
                  , job^map                -- Token map
                  , job)                   -- Token value
THEN
    stopwitherror(essput, 12);

    RETURN SEND^SPI;
END;                                     -- SUBMIT^JOB
-----

-- The STATUS^JOB function procedure.
-- This function procedure formats a SPI buffer for the scheduler
-- to get the status of the job submitted. This procedure uses
-- the job number returned by the scheduler as a result of the
-- SUBMIT command to identify the job.
-----

INT PROC STATUS^JOB(JNUM);
    (continued)

```

```

INT      .jnum;

BEGIN
INT      error;                                -- General error variable

! Initialize the SPI buffer
IF (error := SSINIT( spi^buf
                    , (zbat^val^buflen^min+6)
                    , zbat^val^ssid^init
                    , zspi^val^cmdhdr
                    , zbat^cmd^status          -- Status command token
                    , zbat^obj^job             -- Job object token
                    , 0                        -- MAXRESP
                    , zspi^val^true)) THEN      -- Enable checksum
    stopwitherror(essinit, 13);

! Insert job-number token returned by the SUBMIT^JOB proc
IF (error := SSPUTTKN(spi^buf, zbat^tkn^sel^job^number, jnum)) THEN
    stopwitherror(essput, 12);

RETURN SEND^SPI;
END;                                           -- STATUS^JOB
-----
--
-- The MAIN PROCEDURE.
-- Here the above procedures are called to communicate to the
-- scheduler $ZBAT and finally print out the data returned by
-- the scheduler through the SPI tokens.
-----
--
PROC request MAIN;

BEGIN
INT      .scheduler^name[0:10] := ["$ZBAT.#ZSPI"];

INT      error;
INT      jobnumber;
STRING   .ascii^num[0:$LEN(zbat^ddl^netbatch^name^def)];
INT      version;
INT      .ptr;                                -- Working pointer
INT      .job^status^map[0:zbat^map^status^job^wln] := zbat^map^status^job;
STRING   buffer[0:79];

STRUCT   .job^stat(zbat^ddl^status^job^def);    -- Job status definition
structure

-- Read startup message
work^area := " $RECEIVE";
CALL FILE_OPEN_(work^area:8, recv_chan,,,,,1);
(continued)

```

```

    CALL READX(recv_chan, sm, ($OCCURS(sm)*$LEN(sm)));
    CALL FILE_CLOSE_(recv_chan);

-- Convert C-series file name to D-series format
error := OLDFILENAME_TO_FILENAME_(sm[_out], work^area:maxfilebytes,
                                   work^length);
-- Open OUT file
    IF error OR (error := FILE_OPEN_(work^area:work^length, out_chan))
THEN
    stopwitherror(efile_open_, 16);

-- Open the scheduler
    IF (error := FILE_OPEN_(scheduler^name:11, schd_chan)) THEN
        stopwitherror(efile_open_, 16);

-- Format the SPI structure and submit the job
error := submit^job;

-- Check if only warnings are returned
    IF NOT (error = 0) THEN
        stopwitherror(eretcod, 34)
    ELSE
        BEGIN
            IF error := SSGETTKN(spi^buf -- Check if there is a job number in
the buffer
                                   , zbat^tkn^sel^job^number
                                   , jobnumber
                                   , 1) THEN
--          CALL PROCESS_STOP_(,,,2!cc!,error!ti!,,essget:12);
                stopwitherror(essget, 12);

-- Job submission output header message

                buffer :=' " " -> @ptr;
                CALL WRITE(out_chan, buffer, @ptr '-' @buffer);

                buffer :=' "    Job submitted to $ZBAT using SPI" & %H0A0D -> @ptr;
                CALL WRITE(out_chan, buffer, @ptr '-' @buffer);

-- Output job number to screen
                CALL NUMOUT(ascii^num, jobnumber, 10, 3);
                buffer :=' "          Job number : " & ascii^num FOR 3 -
>@ptr;
                CALL WRITE(out_chan, buffer, @ptr '-' @buffer);

                IF error := SSGETTKN(spi^buf -- Get Job name from SPI buffer
                                   , zbat^tkn^sel^jobname
                                   , ascii^num
                                   , 1) THEN
                    stopwitherror(essget, 12);

(continued)

```

```

        buffer ':=' "                               Job name      : " & ascii^num for 12 &
                                                %H0A0D  & %H0A0D ->
@ptr;
    CALL WRITE(out_chan, buffer, @ptr '-' @buffer);
    END;

-- Format the SPI buffer and get the status of the job
    IF (error := status^job(jobnumber)) THEN
        stopwitherror(eretcod, 34)
    ELSE

BEGIN
    IF error := SSGET(spi^buf                    -- Read tokens returned
                    , job^status^map
                    , job^stat
                    , 1) THEN
        stopwitherror(essget, 12);

-- Job status output header message
    buffer ':=' "      Job status from $ZBAT using SPI" & %H0A0D -> @ptr;
    CALL WRITE(out_chan, buffer, @ptr '-' @buffer);

-- Output job selpri
    CALL NUMOUT(ascii^num, job^stat.zselpri, 10, 2);
    buffer ':=' "                Selpri      : " &
                ascii^num[1] FOR 1 -> @ptr;
    CALL WRITE(out_chan, buffer, @ptr '-' @buffer);

-- Output job class
    buffer ':=' "                                Class      : " &
                job^stat.zclassname FOR 24  &
                %H0A0D-> @ptr;
    CALL WRITE(out_chan, buffer, @ptr '-' @buffer);
    END;

    CALL CLOSE(schd_chan,);
    CALL CLOSE(out_chan,);
END;

```

---

---

---

---

# Glossary

**abend.** An acronym for abnormal end (of a process).

**ASSIGN.** A parameter that assigns the name of an actual file to a logical file name in a program. It also can specify the file's creation and open attributes. For more information, see the *TACL Reference Manual* and the *TACL Programming Guide*.

**attachment set.** A named set of ASSIGNS, DEFINES, and PARAMs.

**attribute.** A characteristic of an entity (for example, the selection priority of a job). In an SPI interface, an attribute of an object is usually expressed as a simple token or a field in an extensible structured token. The attributes of a token are its length, count, address, and offset. Programs can get these through special SSGET operations.

**BATCHCAL.** The file ID of the NetBatch calendar program. The program enables you to generate a run calendar, display run times, and reformat an old calendar file to the current format.

**BATCHCOM.** The file ID of the NetBatch command interpreter program. BATCHCOM enables interactive and noninteractive manipulation of the scheduler; the scheduler's executors, classes, and attachment sets; and jobs.

**BPROC.** The former file ID of NBEXEC, the NetBatch executor program. See [NBEXEC](#).

**buffer.** A sequence of memory locations used for temporary storage of data. For instance, data to be sent in an interprocess message is encoded in a buffer from which it is copied by the file system. The data is delivered to a buffer addressable by the recipient. See also [message](#); [SPI buffer](#).

**built-in.** A primitive function or variable in the TACL program. Names of built-ins always begin with a pound sign (#).

**calendar.** See [run calendar](#).

**class.** A logical entity in the scheduler. A class's purpose is to group jobs and to control their flow to executors and thereby to the executors' processors. Classes are the job-queuing mechanisms of the scheduler. You can assign a class to multiple executors to give its jobs opportunities to execute in different processors.

**collector.** An EMS process that accepts event messages from subsystems and logs them in the event log. See also [distributor](#).

**command.** A demand for action by or information from a subsystem, or the operation demanded by an operator or application. A command is typically conveyed as an interprocess message from an application to a subsystem.

**command message.** An SPI message, containing a command, that is sent from an application program to a subsystem. See [SPI message](#).

**command number.** A number representing a particular command to a subsystem. Each subsystem or management process with a token-oriented programmatic interface can have its own set of command numbers, represented in DDL by constants and in programs by TAL LITERAL or DEFINE declarations, COBOL level-01 variables, C #define directives, or TACL text variables. The command number is a header token in command and response messages.

**completion code.** A status code returned by a process to its creator. The code indicates whether the process terminated successfully or otherwise.

**conditional token.** A token that is sometimes, but not always, present in a particular event message. See also [unconditional token](#).

**consumer distributor.** An EMS distributor process that returns on request selected event messages to management applications. See also [forwarding distributor](#); [printing distributor](#).

**context, context information.** The information required by a subsystem to process a command that requires more than one interchange of command and response messages. Continuation of a response in multiple response messages from the subsystem requires the subsystem to send the context information to the application program. The application program must send that information back to the subsystem in a new command message, so that the subsystem can continue with the response. See [context token](#).

**context token.** A token indicating (by its presence or absence) whether more response messages are to come.

- If present in a response message, the response is continued in another response message. To get the next message, the application reissues the original command with the context token is included in the new command message.
- If absent from a response message, the application knows that the series of response messages is complete.

The contents of the context token enable the subsystem to find its place and issue the next response message. In a response, the context token is a type of response-control token—the only response-control token that can be present in a response as well as in a command. In event-message distribution, the GETEVENT command returns the context token with the next event message. The context token identifies the next event message, so the backup distributor process can recover if the primary process goes down. The requester must send the context token back to the distributor on the next GETEVENT call. (See [context, context information](#).)

**continuation.** The packaging of a response in multiple response messages. The subsystem uses a context token to indicate that the response is continued to another message. Each response message can contain multiple response records, but a single response record cannot span two response messages.

**control and inquiry.** Those aspects of object management related to the state or configuration of an object. Such aspects include actions that affect the state or configuration of an object, inquiries about the object, and commands pertaining to the session environment (for example, commands that set default values for the session). See also [event management](#).

**critical event.** An event designated as critical to system or network operations. Each subsystem determines what set of events generated by that subsystem should be critical. The subsystem identifies whether an event is critical or noncritical by setting the value of the emphasis token in the event message. See also [noncritical event](#).

**current position.** The SPI-buffer location of the token whose code, value, or attribute has just been retrieved. Scans for the next token code (with the operations ZSPI-TKN-NEXTCODE and ZSPI-TKN-NEXTTOKEN) begin at the current position but always return a code beyond the current position. See also [next position](#); [initial position](#).

**current token.** The token in the current position. See current position.

**data list.** A grouping of tokens used to separate response records in a response message, or to enclose a single response record if the program so requests. A data list consists of a list token denoting a data list (different from the token that starts an error list or a generic list), followed by a response record and an end-list token. See [response record](#).

**data-portion token.** A token in the body of an SPI message as opposed to the header of the message. Data-portion tokens are placed in the buffer using SSPUT or SSPUTTKN. Some data-portion tokens can occur multiple times in the buffer, and most can be enclosed in lists. Programs can set the current position to these tokens and retrieve their values using the NEXTCODE and NEXTTOKEN operations. See also [header](#).

**DEFINE.** A named set of attributes and associated values. In a DEFINE (as with an ASSIGN), you can specify information that jobs communicate to processes they start. The NetBatch product supports all DEFINE types.

**definition.** One of the declarations provided by HP for use in applications that call the SPI procedures. These definitions are provided in definition files.

**definition files.** A set of files containing declarations for use in applications that call SPI procedures. SPI has a standard definition file for DDL and one for each programming language supporting SPI. The latter files are derived from the DDL definition file. Similarly, each subsystem with a token-oriented programmatic interface has one definition file for DDL and one for each programming language. Some subsystems such as data communications subsystems have extra, shared definition files. See *also* [SPI standard definitions](#); [EMS standard definitions](#); [subsystem definitions](#).

**dependency.** A relationship between two jobs that prevents one of the jobs (the dependent job) from executing before the other job (the master job) releases it. See *also* [dependent job](#); [master job](#).

**Distributed Systems Management (DSM).** Software tools that aid management of NonStop S-series systems and Expand networks. These tools include SPI, EMS, Subsystem Control Facility (SCF), Distributed Name Service (DNS), ViewPoint console application, DSM Template Services, and token-oriented programmatic interfaces to the management processes for various NonStop subsystems, as well as various tools that provide management services and help in the development of management applications.

**dependent job.** A job with the WAITON attribute. Execution of such a job depends on its release by each job specified by the attribute. See *also* [dependency](#); [master job](#).

**distribution subvolume (DSV).** See [DSV](#).

**distributor.** An EMS process that distributes event messages from event logs to requesting management applications, to Guardian console message destinations, to a collector on another node, or to printers, devices, or files. See *also* [consumer distributor](#); [forwarding distributor](#); [printing distributor](#).

**downward compatibility.** The ability of a requester to operate properly with a server of a lower revision level. In this case, the requester is downward-compatible with the server, and the server is upward-compatible with the requester. See *also* [upward compatibility](#).

**DSM Template Services.** A software facility that is used to produce display text from tokenized SPI messages. This facility is used most commonly to generate operator console messages from EMS event messages, but it also can be used to generate labels from SPI token values.

**DSV.** A subvolume containing product files restored from a site update tape (SUT).

**EMS.** A software facility providing event-message collection, logging, and distribution facilities for the NonStop OS. It provides for different descriptions of events for people and for programs, lets an operator or application select conveniently from event-message data, and allows for flexible distribution of event messages in a system or network. It has programmatic interfaces based on SPI for both event reporting and event retrieval. See [event message](#).



**EMS standard definitions.** The set of declarations provided by EMS for use in event management regardless of the subsystem. Any application that retrieves tokens from event messages needs the EMS standard definitions. Names of EMS standard definitions start with ZEMS. See also [definition](#); [definition files](#); [SPI standard definitions](#).

**end-list token.** A syntax token that ends a list. SPI defines a single end-list token, whose token code is ZSPI-TKN-ENDLIST. See also [list token](#); [syntax token](#).

**enumerated type.** A 16-bit signed data type that has one of a specified list of values with designated meanings. The enumerated type is one of the standard token data types defined by SPI. The list of acceptable values for the data type and what those values mean varies depending on the token number. The list is defined by the subsystem.

**error.** A condition that causes a command or other operation to fail. See also [warning](#).

**error list.** A grouping of tokens used in a response record to provide error and warning information. An error list consists of a list token that denotes an error list (different from the token that starts a data list or a generic list), followed by an error token, other tokens explaining the error (optional), and an end-list token. Error lists can be nested in other error lists. The return token cannot be included in an error list. See [return token](#).

**error number.** A value that can be assigned to a return token, or to the last field of an error token, to identify an error that occurred. SPI defines a small set of error numbers, but most error numbers are defined by subsystems.

**error token.** A response token that indicates the reason an error occurred in performing a command. NonStop subsystems enclose each error token in an error list, which can also contain additional information about the error. A response record must contain a return token, and also can contain error lists to explain the error further. The token code for the error token is ZSPI-TKN-ERROR. Its value is a structure consisting of the subsystem ID and an error number identifying the error. See [error list](#), [error number](#), and [return token](#).

**event.** A significant change in a condition in the system or network. Events can be operational errors, notifications of limits exceeded, requests for action, and so on.

**event log.** A file or set of files maintained by EMS to store event messages generated by subsystems.

**event management.** The reporting and logging of important events that occur in a system or network, the distribution and retrieval of information concerning those events, and the actions taken by operations personnel or software in response to the events. See also [control and inquiry](#).

**Event Management Service (EMS).** See [EMS](#).

**event message.** (1) In programmatic interfaces based on SPI, a special type of SPI message that describes an event occurring in the system. (2) In the ViewPoint console application, the displayed form of such a message, shown as one text line.

**executor.** A logical entity in the scheduler. An executor's purpose is to link jobs through their classes to a processor. This link enables the scheduler to execute, in the specified processor, the initial process (the executor program) of each job. Executors act as gateways between classes and processors. When started, an executor allows one job at a time from the classes to run in its processor. No other jobs can use the executor until the job finishes. Stopping an executor prevents jobs from using it to gain access to its processor. The number of started executors determines how many jobs can run together. For example, a scheduler with 10 started executors can run up to 10 jobs concurrently.

**executor program.** A program file started as the initial process of a job by a NetBatch scheduler. (The process executes the commands contained in the job's input file.)

**explicit command.** A command entered by a user or specified in an input file.

**extensible structure.** A structure declared for the value of an extensible structured token. See also [extensible structured token](#); [fixed structure](#).

**extensible structured token.** A token consisting of a token code and a value that is an extensible structure. HP can extend extensible structures by adding new fields at the ends of the structures. Such structures are typically used to indicate the attributes of an object being operated on and to return status and statistics information in responses. They can also be used for other purposes. The token is referenced by a token map that describes the structure to SPI so SPI can provide compatibility between different versions of the structure. See also [simple token](#); [structure](#); [structured token](#).

**filter.** A file containing a list of criteria against which incoming event messages can be compared. The filter allows messages that satisfy the criteria to pass through it to the application. Messages that do not satisfy the criteria cannot pass through the filter.

**fixed structure.** A structure declared for the value of a simple token that includes several fields. Fields cannot be added to fixed structures. See also [extensible structure](#).

**forwarding distributor.** An EMS distributor process that sends selected event messages to an EMS collector on another node. See also [consumer distributor](#); [printing distributor](#).

**GETVERSION command.** An information command that reports to the requester the server version of the subsystem server and possibly additional version information about objects defined by the subsystem. All NonStop subsystems with a programmatic command interface based on SPI have a GETVERSION command.

**group manager.** A user whose Guardian user ID is *n*,255 (for example, 205,255). See also [non-super-group user](#); [super ID](#); [super-group user](#).

**header.** See [SPI message header](#).

**header token.** A special token type containing information about an SPI message. Header tokens are common to all or most messages of a specific type and differ from other tokens in several ways: they exist in the buffer at initialization; their values are usually set by SSINIT; they occur only once in a buffer; they are never enclosed in a list; they cannot be moved to another buffer with SSMOVE; and programs cannot position to them or retrieve their values using a NEXTCODE or NEXTTOKEN operation. Programs retrieve header-token values by passing appropriate token codes to SSGET and can change some header-token values by passing the token codes to SSPUT. Examples of command header tokens are the command, object type, maximum-response, server-version, maximum-field-version, and checksum tokens. Examples of event-message header tokens are the event number, the event generation time, the logging time, the maximum-field-version token, and the checksum token.

**header type.** A header token in an SPI message that indicates whether the message is a command or response message, or an event message.

**high PIN.** A process identification number in the range 256 through 65535. See also [low PIN](#).

**implicit command.** A default command effective in the absence of an explicit command.

**information token.** A response token that conveys information requested by a command, as opposed to one that serves a syntactical purpose such as delimiting a list, indicates response continuation, identifies how a command completed, or identifies an error. Object-selector tokens, attribute tokens, status tokens, and statistics tokens are types of information tokens.

**initial position.** The location in an SPI buffer just prior to the first token that is not a header token. See also [current position](#); [next position](#).

**initialize.** To prepare a data structure to have values assigned to it. For example, the SPI SSINIT procedure initializes the buffer by building the message header. The SSNULL procedure initializes an extensible structured token by assigning null values to the fields of the structure.

**input file.** A file containing information an executor program needs to execute a job. For example, the input file for an NBEXEC process contains NBEXEC commands. The input file for a COBOL compiler process contains the program source.

**interactive session.** See [session](#).

**job.** A process or a sequence of processes that performs specified tasks. All NetBatch jobs have an executor program and, depending on the program, an input file. The input file contains commands executed by the executor program, which the NetBatch scheduler starts as the job's initial process. The executor-program process can start other processes after the initial process has been started.

**list.** In a SPI message, a group of tokens that defines a context for scanning the buffer and extracting tokens with the SSGET procedure. A list construct imposes hierarchy in the buffer. To retrieve the tokens from a list, the application must first position to the start of the list by retrieving the initial list token, retrieve tokens from the list, then pop out of the list to the next higher level of tokens by retrieving the end-list token. SPI defines three types of lists: data lists, error lists, and generic lists.

**list token.** A syntax token that begins a list. SPI defines three different tokens to begin a list, depending on the type of list: the data-list token, the error-list token, and the generic-list token. See also [end-list token](#); [syntax token](#).

**low PIN.** A PIN in the range 0 through 254. (Technically, PIN 255 also is a low PIN although it is never assigned to a running process.) See also [high PIN](#).

**macro.** A sequence of TACL commands and built-in functions that can contain dummy arguments, thus providing a means for simple argument substitution. No validity checking of the arguments is performed. When the macro name is given to the TACL program, the program substitutes the expansion of the command sequence for the name, replacing any dummy arguments with parameter values supplied when the macro was invoked. See also [routine](#).

**management application.** A program or set of programs that issues commands to subsystems, retrieves event messages, or performs both functions, to aid in managing a computer system or a network of systems. A management application is a requester with respect to the subsystems to which it sends commands. The subsystems are servers with respect to the management application.

**management interface.** An interactive or programmatic interface through which one can manage a subsystem and its objects. In some subsystems, a specific process is dedicated to the management interface. In other subsystems, the process that provides the management interface also performs other functions.

**management process.** The process through which an application issues commands to a subsystem. A management process can be part of a subsystem, or it can be associated with more than one subsystem. In the latter case, the management process is logically part of each of the subsystems. PATHMON is an example of a management process.

**master job.** A job specified by a dependent job's WAITON attribute. Execution of the dependent job depends on its release by the master job. See also [dependency](#); [dependent job](#).

**maximum field version.** In an SPI message, the latest version associated with any non-null field of any extensible structured token in the message. The maximum field version of the SPI message is contained in a header token. It corresponds to the version of the oldest server or requester that can successfully process the message.

**message.** A block of information, usually in the form of a structure, that is sent from one process to another. See also [SPI message](#).

**message buffer.** A sequence of memory locations used for the contents of an interprocess message. See also [buffer](#); [SPI buffer](#).

**message code.** The contents of the first word of an interprocess message. A message code of -28 identifies the message as an SPI message.

**NBEXEC.** The file ID of the NetBatch *exec*utor program. Formerly BPROC (the batch execution process of the obsolete MIS Batch product), NBEXEC executes control file commands, supplies data to started processes, and logs process output.

**NB^JOB^SUBMIT.** A TAL procedure call defined in the NetBatch library file BATCHLIB. The procedure enables the programmatic submission and alteration of jobs from user-written programs. It provides functionality similar to that of the BATCHCOM commands SUBMIT JOB and ALTER JOB.

**NETBATCH.** The file ID of the NetBatch scheduler program. See also [scheduler](#).

**NetBatch supervisor.** Any user with execute access to the NETBATCH program file.

**NetBatch-Plus.** A Pathway application that provides a screen-driven interface to the NetBatch product.

**next position.** The location at which a subsequent operation occurs. In SPI, the next position is the location in the SPI buffer from which SSGET normally retrieves a token value or token attribute. (An exception is the special calling mode for requesting an attribute of the current token.) See also [current position](#); [initial position](#).

**non-super-group user.** A user whose Guardian user ID is 1 through 254,<sub>n</sub> (for example, 205,70). See also [group manager](#); [super ID](#); [super-group user](#).

**noncritical event.** An event that is not considered critical to the operation of the system or network. Each subsystem determines what set of events generated by that subsystem should be designated as critical. The others are noncritical. The subsystem identifies whether an event is critical or noncritical by setting the value of the emphasis token in the event message. See also [critical event](#).

**noninteractive session.** See [session](#).

**nonsensitive command.** A subsystem command that can be issued by any user or program with access to the subsystem—that is, a command on which the subsystem imposes no further security restrictions. See also [sensitive command](#).

**null object type.** A place-holder object type that management applications can use in programmatic commands that do not require explicit specification of a particular object type.

**null value.** A value indicating that a program has made no explicit assignment to a variable or field. For SPI, a field of a structure has a null value if the application has made no explicit assignment to that field after calling the SSNULL procedure to initialize the structure.

**object.** (1) In SPI, an entity subject to independent reference and control by a subsystem: for example, in NetBatch, an attachment set, class, executor, job, or scheduler. An object typically has a name and a type known to the controlling subsystem. (2) In DDL, an item in a dictionary. DDL assigns each object a unique object number for identification.

**object type.** The category of objects to which a specific object belongs. A subsystem identifies a set of object types for the objects it manages. The operator interface to the subsystem might have keywords to identify the types. The programmatic interface would have object-type numbers suitable for passing to the SSINIT procedure.

**object-name token.** A parameter or response token that identifies, by name, a particular object of a given object type. An object-name token is a type of object-selector token. See [object-selector token](#).

**object-selector token.** A token (of the object type given in the command) that identifies one or more specific objects to operate on. Typically, the value of such a token is either some form of object name or an object number. An object-name token is a type of object-selector token. See [object-name token](#).

**object-type number.** A number representing an object type managed by a subsystem. Each subsystem with a token-oriented programmatic interface can have its own set of object-type numbers, represented in DDL by constants and in programs by TAL LITERAL or DEFINE declarations, C #define directives, COBOL level-01 variables, or TACL text variables. (In some cases, as with the data communications subsystems, object-type numbers are shared by several subsystems.) The object-type number is a header token in commands and responses. See [object type](#).

**owner.** (1) For a disk file, the user or program that created the file, or a user or program to whom the creator has given the file with the FUP GIVE command. (2) For a process, the user or program that created the process or, if the PROGID option was specified in the FUP SECURE command for the code file, the user or program that owns the code file. (3) For a token or other definition, the subsystem that provided the definition. (4) For a subsystem, the company or organization that provides the subsystem, or the eight-character string identifying that company.

**PARAM.** A parameter that supplies a user-defined value to a process requesting that value at creation time. For more information, see the *TACL Reference Manual* and the *TACL Programming Guide*.



**parameter token.** (1) In control and inquiry, a token supplying parameter information for a command. Most tokens in a command message are parameter tokens. Depending on the subsystem, they can include attribute tokens, object-selector or object-name tokens, and subsystem-control tokens. See also [syntax token](#). (2) In event management, a token representing a parameter passed by an application to an event-message filter. Such tokens are kept in a parameter buffer. For more information, see the *EMS Manual*.

**PIN.** A unique, system-assigned identifier of a process running in a processor. See also [high PIN](#); [low PIN](#).

**predefined value.** A commonly used value that is given a name in a definition file.

**printing distributor.** An EMS distributor process that sends selected event messages to printers, devices, or files. See also [consumer distributor](#); [forwarding distributor](#).

**private token type.** A token type defined by, and specific to, a particular subsystem. A private token type is built from standard SPI token data types although it might have additional semantic connotations for the subsystem. See [token type](#).

**procedural interface.** A means of getting system or application program services through procedure calls. Also, the set of procedures through which services are obtained. For instance, an application has a procedural interface to SPI. That interface comprises the procedures SSINIT, SSNULL, SSPUT, SSPUTTKN, SSGET, SSGETTKN, SSMOVE, and SSMOVETKN.

**processor.** A computer component whose parts include circuits controlling the interpretation and execution of instructions.

**process identification number (PIN).** See [PIN](#).

**programmatic command.** A command issued by a program rather than by a human user.

**programmatic interface.** A means for a program to communicate with another program. On a NonStop system, a programmatic interface typically includes a message format, a set of message formats, or a set of procedures (such as the SPI procedures) to build and decode messages; definitions of message elements (commands, data types, objects, parameters, response data, errors, and so on); rules for communication between the requester and the server; and software to receive and respond to messages defined for the interface.

**requester version.** The software revision level of the definition files used in the compilation of a requester. Each subsystem has its own definitions, so the requester version can differ in requests to different subsystems.

**response.** The information or confirmation supplied by a subsystem in reaction to a command. A response is typically conveyed as one or more interprocess messages (response messages) from a subsystem to an application.

**response message.** An SPI message that is sent from a subsystem to an application program in reaction to a command message. See [SPI message](#); [command message](#).

**response record.** A set of response tokens, usually describing the results of performing a command on one object. A response can consist of multiple response records, distributed among one or more response messages. A response message always contains a whole number of response records (that is, a response record cannot be split between two messages). If there are multiple response records in a response message, each response record is enclosed in a data list. See *also* [data list](#). Each response record must contain a return token; see *also* [return token](#).

**response token.** A token returned as an element of a response. Response tokens include information tokens (which contain response data of interest to the application), syntax tokens (such as list tokens), one special response-control token (the context token), the return token, and error tokens.

**response-control token.** A parameter token or response token that influences or reflects how a subsystem packages its response to a command. Response-control tokens are defined by SPI rather than by subsystems. They include the maximum-response token, the response-type token, and the context token.

**return token.** The response token that indicates whether a command was successful and why it failed if it did. Every response record in a response from a NonStop subsystem contains a return token. A response record also can contain error lists that include error tokens. The token code for the return token is ZSPI-TKN-RETCODE. Its value consists of a single integer field. See *also* [error token](#).

**routine.** A sequence of TACL commands and built-in functions that can perform complex argument interpretation. A routine can interpret an item in its argument string by applying the information received from the execution of previous arguments or by applying the information received from the results of any function or program executed before the argument item is read. Routines construct their own expansions by using the built-in function #RESULT. When the routine name is given to the TACL program, the arguments that the routine uses with #RESULT are substituted for the name. A routine can have a null expansion. See *also* [macro](#).

**run calendar.** A disk file generated from user-supplied source data by the BATCHCAL program. The file contains a series of dates and times called run times. You can schedule a job to run automatically at those times by using the CALENDAR attribute to assign the file to the job.

**run time.** The date and time when a job runs.



**scheduler.** A process-pair server that stores job records in its database, schedules and starts jobs, monitors their execution, and records job termination details. The scheduler's program file is NETBATCH and its command interface BATCHCOM. The scheduler queues jobs according to the jobs' scheduling criteria and dependencies. It starts the jobs' executor programs, monitors job execution, and acts as a home terminal for the jobs' processes. If any of the job's processes fail, the scheduler also can stop, restart, or reschedule a job if the job's attributes specify such action. A scheduler records events such as the creation of executor-program processes in a log file. The scheduler has an EMS interface through which it sends information about certain scheduler-related and job-related events to an EMS collector. Each scheduler has its own database whose files record information about the scheduler, its classes and executors, attachment sets, and jobs. See also [NETBATCH](#).

**sensitive command.** A command available to a restricted set of users because the subsystem restricts access to the command. See also [nonsensitive command](#).

**server version.** The software release version of the server to which a requester using SPI (such as a management application) is sending a command. If the server version is older than the maximum field version in a request, the server rejects the request. SPI puts the maximum field version into the command buffer. The server puts its own version into each response buffer. See [maximum field version](#).

**session.** The period during which two entities can exchange data. (1) For a management application, the period during which an application can issue commands to a subsystem. (2) For a command interpreter, the period during which a user can issue commands to the command interpreter. (3) For the ViewPoint console application, the period between the user's invoking the application and exiting the application.

**simple token.** A token consisting of a token code and a value that is either a single elementary field, such as an integer or a character string, or a fixed (nonextensible) structure. See also [extensible structured token](#).

**special operation.** An operation, such as a control operation or an operation that gets information from the buffer (rather than the header), performed by the SSGET procedure or the SSPUT procedure. Special operations include getting the length or number of occurrences of a token, changing the current position, clearing the last-error information, or deleting a token from the buffer. A program directs SSGET or SSPUT to perform a special operation by passing to the procedure one of a set of special SPI token codes. These special token codes do not represent tokens in the buffer but simply direct SSGET or SSPUT to perform the indicated operations.

**SPI.** A set of procedures and associated definition files used to define common message-based interfaces for communication between requesters and servers—for instance, in a management application. It includes procedures to build and decode specially formatted messages (as described under SPI message); definition files in TAL, C, COBOL, and TACL format for inclusion in programs, macros, and routines using the interface procedures; and definition files in DDL for programmers writing their own subsystems.

**SPI buffer.** A sequence of memory locations used for a message produced by the SPI procedures. See *also* [buffer](#); [message](#).

**SPI control code.** A special token code, passed to one of the SPI procedures, that directs SPI to perform a specified action on the buffer (such as a positioning operation). The ZSPI-TKN-DATAFLUSH, ZSPI-TKN-DELETE, and ZSPI-TKN-CLEARERR token codes for SSPUT are examples of SPI control codes. An SPI control code is a type of special operation. See [special operation](#).

**SPI definitions.** See [SPI standard definitions](#).

**SPI error number.** A number that indicates whether a call to an SPI procedure completed successfully and why it failed if it did. This number is returned in the status parameter on calls to the SPI procedures. The SPI error number does not reflect the success or failure of a command. It applies only to errors in the building and decoding of a message in an SPI buffer.

**SPI message.** A message specially formatted by the SPI procedures for communication between a management application and a subsystem, or between one subsystem and another. An SPI message consists of a collection of tokens. To retrieve a token from the message, the application passes a token code to SPI, which scans for the appropriate token and returns its value to the application. An SPI message is a single block of information sent at one time as one interprocess message. The two types of SPI messages are distinguished by two different SPI message header types: command and response messages, and event messages. See [header type](#).

**SPI message header.** The initial part of an SPI message. The first word of the header always contains the value -28. The remainder of the header contains descriptive information about the SPI message, most of which is accessible as header tokens. The tokens in the header differ according to the header type. The header of a message that contains a command or response differs from the header of an event message. An application can use SSGET or EMSGET calls to retrieve the values of header tokens, and can use SSPUT calls to change the values of some. However, there are basic differences between header tokens and other tokens. See [header](#).

**SPI procedures.** Guardian procedures that build and decode buffers for use in system and network management and in other applications. These procedures are SSINIT, SSNULL, SSPUT, SSPUTTKN, SSGET, SSGETTKN, SSMOVE, and SSMOVETKN.

**SPI standard definitions.** Declarations available for use with SPI procedures. There also are subsystem-specific declarations for each subsystem, and some declarations that apply to multiple subsystems. An application using SPI needs the SPI standard definitions and the subsystem definitions for all subsystems with which it communicates. Names of SPI standard definitions start with ZSPI. See *also* [definition](#); [definition files](#); [EMS standard definitions](#); [subsystem definitions](#).

**structure.** A data item with multiple fields, possibly of different types. This type of data item corresponds to a DEF in the DDL language, to a STRUCT in the TAL and TACL languages, and to a RECORD in the COBOL language.

**structured token.** A token whose value is a structure. Some structured tokens are simple tokens with fixed structures—for example, the error token, ZSPI-TKN-ERROR. Other structured tokens are extensible structured tokens. See [structure](#); [simple token](#); [extensible structured token](#).

**subject.** In event management, a device, process, or other named entity about which a given event message is concerned.

**Subsystem Programmatic Interface (SPI).** See [SPI](#).

**subsystem.** A program or set of processes that manages a cohesive set of objects. Each subsystem has a process (in some cases, this process is the entire subsystem) through which applications can request services by issuing commands defined by that subsystem. See [management process](#).

**subsystem definitions.** The set of declarations available for use with a particular subsystem that supports a token-oriented programmatic interface. See *also* [definition files](#); [EMS standard definitions](#); [SPI standard definitions](#).

**subsystem ID.** A data structure that uniquely identifies a subsystem (including whether it is a NonStop subsystem or a subsystem you write). It consists of the name of the owner of the subsystem (the company that provides it), a subsystem number that denotes the subsystem within the scope of its owner, and a subsystem version number. The subsystem ID is an argument to most of the SPI procedures.

**subsystem number.** An integer that identifies a subsystem in the context of its owner. The subsystem owner, the subsystem number, and the subsystem version number make up the subsystem ID that uniquely identifies a subsystem.

**subsystem owner.** A value identifying the company that supplies a particular subsystem. It consists of a name of up to eight characters, blank-filled on the right. The owner for all subsystems supplied by HP is TANDEM. The subsystem owner, the subsystem number, and the subsystem version number make up the subsystem ID that uniquely identifies the subsystem.

**subsystem version number.** A 16-bit integer representing the software release version of a subsystem. The subsystem version number is a field of the subsystem ID. If its value is null (zero), the subsystem ID refers to any and all versions of the subsystem. See [version number](#).

**subsystem-control token.** A parameter token that influences how a subsystem performs a command. For instance, in the START PATHWAY programmatic command, the parameter ZPWY-TKN-DEF-PATHWAY is a subsystem-control token because it determines whether a cold start or a cool start will be performed. Similarly, the SPI token ZSPI-TKN-ALLOW-TYPE is a subsystem-control token. It determines under what conditions a subsystem will continue command processing on the next object in a sequence if errors or warnings occur. See also [response-control token](#).

**super ID.** The Guardian user ID 255,255. See also [group manager](#); [non-super-group user](#); [super-group user](#).

**super-group user.** A user whose Guardian user ID is 255,*n* (for example 255,13). See also [group manager](#); [non-super-group user](#); [super ID](#).

**supervisor, NetBatch.** See [NetBatch supervisor](#).

**symbolic name.** A name used in programs to refer to a value or a variable. HP provides definition files that declare symbolic names for values, token codes, token maps, extensible structures, and other related variables used in management applications.

**syntax token.** A token whose function is not to provide information for a command or response but to bracket or group other tokens. Its use is similar to that of a punctuation symbol. The tokens that begin and end lists (the list tokens) are syntax tokens. See also [parameter token](#); [information token](#).

**TACL.** The standard command interpreter for the HP NonStop operation system.

**TAL.** A high-level, block-structured language that works with NonStop OS hardware to provide optimal object-program performance.

**Tandem Advanced Command Language (TACL).** See [TACL](#).

**target subvolumes (TSVs).** See [TSVs](#).

**time attributes.** The job attributes AFTER, AT, CALENDAR, EVERY, and WAIT. The attributes determine the run time of a job.

**token.** (1) In SPI, a distinguishable unit in an SPI message. Programs place tokens in an SPI buffer using the SSPUT procedure (except for header tokens, which are a special case), and retrieve them from the buffer with the SSGET procedure. A token has two parts: an identifying code, or token code, and a token value. For command and response messages, a token normally represents a parameter to a command, an item of information in a response, or control information for the subsystem. For event messages, a token normally represents an item of information about an event or about the event message itself. (2) In the TACL environment, an entity recognized by the #ARGUMENT built-in function when parsing an argument string passed to a routine.

**token code.** (1) In SPI, a 32-bit value that, as the first part of a token, allows any token to be identified and located in an SPI message. A token code consists of a token type (16 bits) and a token number (16 bits). (2) In the TAL, C, TACL, and COBOL languages, names are used to represent token codes (ZSPI-TKN-SSID, for example). (3) In DDL, a special definition (using the TOKEN-CODE statement) that the DDL compiler will translate into an SPI token code. See also [token map](#).

**token data type.** The part of the token code that defines the type of value (such as an integer or a file name) allowed for a token.

**token length.** The part of a token code that indicates the length in bytes of the corresponding token value. A token length of 255 indicates that the token value has variable length or a length greater than 254. In this case, the first word of the token value contains the (noninclusive) byte length of the rest of the token value.

**token map.** (1) In SPI, a structure that contains decoding information for an extensible structured token. Also, a variable name used to reference an extensible structured token. The token map includes a token code and a description of the token value: its fields, the null values of those fields, and the versions of the fields. A token map defines a structure that might change in some later code version (by the addition of new fields at the end), and the information in the map allows SPI to provide compatibility between different structure versions. (2) In DDL, a special definition (using the TOKEN-MAP statement) that the DDL compiler will translate into an SPI token map.

**token number.** The number used by a subsystem to identify each type of token that it defines. The token type and the token number together form the token code.

**token type.** (1) In SPI, a combination of the token data type and token length; part of the token code. (2) In DDL, a special definition (using the TOKEN-TYPE statement) that the DDL compiler translates into an SPI token type.

**token value.** The value assigned to a token.

**token-oriented.** Said of a programmatic interface that conveys information as a series of code-value pairs accessed by code rather than by address or ordinal position. SPI is used by application programs and subsystems to provide a token-oriented programmatic interface.

**Transaction Application Language (TAL).** See [TAL](#).

**TSVs.** Subvolumes containing product files updated from a distribution subvolume by DSM/SCM, the software configuration manager.

**unconditional token.** A token that is always present in a particular event message. See *also* [conditional token](#).

**upward compatibility.** The ability of a requester to operate with a server of a higher revision level. In this case, the requester is upward-compatible with the server, and the server is downward-compatible with the requester. See *also* [downward compatibility](#).

**version compatibility.** The ability of a requester and server of different revision levels to operate together.

**version number.** A 16-bit integer representation of a software release version. For NonStop subsystems, the version number consists of an uppercase alphabetic character in its left half and a number in its right half.

**ViewPoint console application.** An extensible interactive application for operators. It allows a system or a network to be controlled from a single terminal. It includes several block-mode display screens for events, a block-mode display for system or network status, a conversational TACL screen, and a facility called Define Process to maintain sessions with multiple subsystems at the same time.

**warning.** A condition, encountered in performing a command or other operation, that can be significant but does not cause the command or operation to fail. A warning is less serious than an error. See *also* [error](#).

**wild-card character.** A character matching a series of characters in a string. Used in compare operations to mask characters that are not significant. The NetBatch wild-card characters are \* (asterisk) and ? (question mark).

# Index

## Numbers

204 (event message ZBAT-EVT-JOB-OVER-LIMIT), description of [6-18/6-19](#)  
23:59:59:999.999', specifies midnight [4-57](#)  
301 (event message ZBAT-EVT-JOB-START-ERROR), description of [6-19/6-25](#)

## A

ABEND procedure [4-42](#), [4-48](#), [6-2](#), [6-3](#), [6-15](#), [6-18](#), [6-19](#)  
Abend, definition of [Glossary-1](#)  
ACTIVE executor state [4-35](#)  
ADD SCHEDULER command  
    interactive scheduler startup, use in [2-2](#)  
AFTER job attribute  
    See ZAT-FLAG field  
ASSIGN [Glossary-1](#)  
ASSIGN attachment-set attribute  
    See ZBAT-TKN-, ATT-SET-ASSIGN  
AT job attribute  
    See ZAT-FLAG field  
Attachment set  
    attributes  
        ASSIGN (ZBAT-TKN-ATT-SET-ASSIGN) [4-40](#)  
        DEFINE (ZBAT-TKN-ATT-SET-DEFINE) [4-40](#)  
        PARAM (ZBAT-TKN-ATT-SET-PARAM) [4-40](#)  
        SECURITY (ZBAT-TKN-ATT-SET-SECURITY) [4-41](#)  
        TEMPORARY (ZBAT-TKN-ATT-SET-TEMPORARY) [4-41](#)  
    definition of [Glossary-1](#)  
ATTACHMENT-SET job attribute  
    See ZBAT-TKN-, ATT-SET-ID  
Attributes  
    AFTER

Attributes (continued)

    See ZAT-FLAG field  
ASSIGN  
    See ZBAT-TKN-, ATT-SET-ASSIGN  
AT  
    See ZAT-FLAG field  
ATTACHMENT-SET  
    See ZBAT-TKN-, ATT-SET-ID  
AT-ALLOWED  
    See ZAT-ALLOWED field  
BACKUPCPU  
    See ZBACKUPCPU1 field;  
    ZBACKUPCPU2 field  
CALENDAR  
    See ZBAT-TKN-, CALENDAR  
CATCHUP  
    See ZEVERY-CATCHUP field  
CLASS (executor)  
    See ZCLASSNAME field, of  
    ZCLASSES in ZBAT-MAP-DEF-EXECUTOR  
CLASS (job)  
    See ZCLASSNAME field, of ZBAT-MAP-DEF-JOB  
CPU  
    See ZCPU field, of ZBAT-MAP-DEF-EXECUTOR  
DEFAULT-CLASS  
    See ZCLASSNAME field, of ZBAT-MAP-DEF-SCHEDULER  
DEFAULT-EXECUTOR-PROGRAM  
    See ZBAT-TKN-, EXECUTOR-PROGRAM  
DEFAULT-HIGHPIN  
    See ZHIGHPIN field, of ZBAT-MAP-DEF-SCHEDULER  
DEFAULT-MAXPRINTLINES  
    See ZMAXPRINTLINES field, of ZBAT-MAP-DEF-SCHEDULER



## Attributes (continued)

DEFAULT-MAXPRINTPAGES

See ZMAXPRINTPAGES field, of  
ZBAT-MAP-DEF-SCHEDULER

DEFAULT-OUT

See ZBAT-TKN-, OUT-FILE

DEFAULT-PRI

See ZPRI field, of ZBAT-MAP-DEF-  
SCHEDULER

DEFAULT-SELPRI

See ZSELPRI field, of ZBAT-MAP-  
DEF-SCHEDULER

DEFAULT-STALL

See ZSTALL field, of ZBAT-MAP-  
DEF-SCHEDULER

DEFAULT-STOP-ON-ABEND

See ZSTOP-ON-ABEND field, of  
ZBAT-MAP-DEF-SCHEDULER

DEFINE

See ZBAT-TKN-, ATT-SET-DEFINE

definition of [Glossary-1](#)

DESCRIPTION

See ZBAT-TKN-, DESCRIPTION

EMS

See ZEMS field

EVERY

See ZEVERY-DAYS field; ZEVERY-  
HOURS field; ZEVERY-MINUTES  
field

EXECUTOR-PROGRAM

See ZBAT-TKN-, EXECUTOR-  
PROGRAM

EXTSWAP

See ZBAT-TKN-, EXTSWAP-FILE

HIGHPIN

See ZHIGHPIN field, of ZBAT-MAP-  
DEF-JOB

HOLD

See ZHOLD field

HOLDAFTER

See ZHOLD-AFTER field

## Attributes (continued)

IFFAILS

See ZIFFAILS field

IN

See ZBAT-TKN-, IN-FILE

INITIATION (class)

See ZINITIATION field, of ZBAT-  
MAP-DEF-CLASS

INITIATION (scheduler)

See ZINITIATION field, of ZBAT-  
MAP-DEF-SCHEDULER

JOBID-ZERO

See ZJOBID-ZERO field

LIB

See ZBAT-TKN-, LIB-FILE

LIMIT

See ZBAT-TKN-, TIME-LIMIT

LOCALNAMES

See ZLOCALNAMES field

MAXPRINTLINES

See ZMAXPRINTLINES field, of  
ZBAT-MAP-DEF-JOB

MAXPRINTPAGES

See ZMAXPRINTPAGES field, of  
ZBAT-MAP-DEF-JOB

MAX-CONCURRENT-JOBS

See ZMAXCONCURRENTJOBS  
field; ZMAXTEMPEXECUTORS  
field

MAX-PRI

See ZMAXPRI field

MEM

See ZMEM field

NAME

See ZNAME field

OUT

See ZBAT-TKN-, OUT-FILE

PARAM

See ZBAT-TKN-, ATT-SET-PARAM

PFS



## Attributes (continued)

See ZPFS field

## PRI

See ZPRI field, of ZBAT-MAP-DEF-JOB

## PURGE-IN-FILE

See ZPURGE-IN-FILE field

## RESTART

See ZRESTART field

## RUND

See ZRUND field

## SAVEABEND

See ZSAVEABEND field

## SECURITY

See ZBAT-TKN-, ATT-SET-SECURITY

## SELPRI

See ZSELPRI field, of ZBAT-MAP-DEF-JOB

## STALL

See ZSTALL field, of ZBAT-MAP-DEF-JOB

## STARTUP

See ZBAT-TKN-, STARTUP-MESSAGE

## STOP-ON-ABEND

See ZSTOP-ON-ABEND field, of ZBAT-MAP-DEF-JOB

## SUBMIT-ALLOWED

See ZSUBMIT-ALLOWED field

## SWAP

See ZBAT-TKN-, SWAP-FILE

## TAPEDRIVES (job)

See ZTAPEDRIVES field, of ZBAT-MAP-DEF-JOB

## TAPEDRIVES (scheduler)

See ZTAPEDRIVES field, of ZBAT-MAP-DEF-SCHEDULER

## TEMPORARY

See ZBAT-TKN-, ATT-SET-TEMPORARY

## Attributes (continued)

## TERM

See ZBAT-TKN-, TERM-FILE

## WAITON

See ZBAT-DDL-, WAITON-INDICATOR; ZBAT-MAP-, DEF-WAITON

## AT-ALLOWED scheduler attribute

See ZAT-ALLOWED field

**B**

## BACKUPCPU scheduler attribute

See ZBACKUPCPU1 field;  
ZBACKUPCPU2 field

## BATCHCAL

core component of NetBatch  
product [1-2](#)

definition of [1-2](#), [Glossary-1](#)

## BATCHCOM

core component of NetBatch  
product [1-2](#)

definition of [1-2](#), [Glossary-1](#)

interactive commands versus

programmatic commands, table of [1-6](#)

scheduler startup, usage in [2-2](#)

## BATCHCTL

See ZBAT-TKN-, BATCHCTL

## BPROC

definition of [Glossary-1](#)

## BPROC, definition of

See also NBEXEC

## Buffers

declarations for [4-8](#)

definition of [Glossary-1](#)

## Built-in

ARGUMENT [Glossary-17](#)

definition of [Glossary-1](#)

in macro [Glossary-8](#)

in routine [Glossary-12](#)

RESULT [Glossary-12](#)

# C

## C programming language

### define directives

- representing command numbers [Glossary-2](#)
- representing object-type numbers [Glossary-10](#)

### sample programs, SPI [C-2](#)

## Calendar

### See Run calendar

## CALENDAR job attribute

### See ZBAT-TKN-, CALENDAR

## CATCHUP scheduler attribute

### See ZEVERY-CATCHUP field

## Class

### attribute, INITIATION (ZINITIATION field of ZBAT-MAP-DEF-CLASS) [4-49](#)

### definition of [Glossary-1](#)

### states

### See ZJOBCLASS field

## class [Glossary-1](#)

## CLASS executor attribute

### See ZCLASSNAME field, of ZCLASSES in ZBAT-MAP-DEF-EXECUTOR

## CLASS job attribute

### See ZCLASSNAME field, of ZBAT-MAP-DEF-JOB

## COBOL programming language

### level-01 variables

- represent command numbers [Glossary-2](#)
- represent object-type numbers [Glossary-10](#)

### sample programs, SPI [C-15](#)

## Codes, token, and token types, table of [B-2/B-4](#)

## Cold start, scheduler [2-2](#)

## Collector (EMS)

### definition of [Glossary-1](#)

### \$0 [6-3](#)

## Command message, definition of [Glossary-1](#)

## Command number, definition of [Glossary-2](#)

## Commands

### definition of [Glossary-1](#)

### descriptions of ADD CLASS [5-14](#)

### discontinuing [3-6](#)

### GETEVENT, in EMS [Glossary-2](#)

### interactive versus programmatic, table of [1-6](#)

## Common definitions, SPI [4-2/4-5](#)

## Compiling filters [6-3](#)

## Completion codes, definition of [Glossary-2](#)

## Components (core), of NetBatch product [1-2](#)

## Conditional token, definition of [Glossary-2](#)

## Configuration file, scheduler

### See ZBAT-TKN-, BATCHCTL

## Consumer distributor

### definition of [Glossary-2](#)

### loading filters [6-4](#)

### See also Distributor

## Context token, definition of [Glossary-2](#)

## Context, definition of [Glossary-2](#)

## Continuation, definition of [Glossary-3](#)

## Control and inquiry, definition of [Glossary-3](#)

## CONVERTTIMESTAMP procedure [A-71](#)

## Core components of NetBatch product [1-2](#)

## CPU executor attribute

### See ZCPU field, of ZBAT-MAP-DEF-EXECUTOR

## critical [Glossary-3](#)

## Critical events

### definition of [3-14](#), [Glossary-3](#)

### flagged by ZEMS-TKN-EMPHASIS [4-7](#), [6-1](#)

### for NetBatch subsystem [6-1](#)

## Current position, definition of [Glossary-3](#)

## Current token, definition of [Glossary-3](#)

# D

Data Definition Language (DDL)

See DDL

Data lists, definition of [Glossary-3](#)

Data-portion token, definition of [Glossary-3](#)

DDL (Data Definition Language), definitions and token maps, table of [B-5/B-12](#)

Decoding response messages [3-7](#)

DEFAULT-CLASS scheduler attribute

See ZCLASSNAME field, of ZBAT-MAP-DEF-SCHEDULER

DEFAULT-EXECUTOR-PROGRAM scheduler attribute

See ZBAT-TKN-, EXECUTOR-PROGRAM

DEFAULT-HIGHPIN scheduler attribute

See ZHIGHPIN field, of ZBAT-MAP-DEF-SCHEDULER

DEFAULT-MAXPRINTLINES scheduler attribute

See ZMAXPRINTLINES field, of ZBAT-MAP-DEF-SCHEDULER

DEFAULT-MAXPRINTPAGES scheduler attribute

See ZMAXPRINTPAGES field, of ZBAT-MAP-DEF-SCHEDULER

DEFAULT-OUT scheduler attribute

See ZBAT-TKN-, OUT-FILE

DEFAULT-PRI scheduler attribute

See ZPRI field, of ZBAT-MAP-DEF-SCHEDULER

DEFAULT-SELPRI scheduler attribute

See ZSELPRI field, of ZBAT-MAP-DEF-SCHEDULER

DEFAULT-STALL scheduler attribute

See ZSTALL field, of ZBAT-MAP-DEF-SCHEDULER

DEFAULT-STOP-ON-ABEND scheduler attribute

See ZSTOP-ON-ABEND field, of ZBAT-MAP-DEF-SCHEDULER

DEFINE attachment-set attribute

See ZBAT-TKN-, ATT-SET-DEFINE

DEFINESETATTR procedure [4-64](#)

DEFINEVALIDATEWORK procedure [4-64](#)

DEFINE, definition of [Glossary-3](#)

Definition files, definition of [Glossary-4](#)

Definitions

DDL, and token maps, table of [B-5/B-12](#)

definition of [Glossary-3](#)

ZSPI-, standard definitions for SPI [4-2/4-5](#)

DELETE executor state [4-36](#)

Dependency, job

See Job, dependent

DESCRIPTION job attribute

See ZBAT-TKN-, DESCRIPTION

Distributed Systems Management (DSM)

See DSM

Distribution subvolume (DSV)

See DSV

Distributor

consumer

definition of [Glossary-2](#)

loading filters [6-4](#)

definition of [Glossary-4](#)

forwarding, definition of [Glossary-6](#)

function of [6-3](#)

printing, definition of [Glossary-11](#)

DOWN executor state [4-36](#)

Downward compatibility, definition of [Glossary-4](#)

See also Upward compatibility; Version compatibility

DSM Template Services, definition of [Glossary-4](#)

See also DSM (Distributed Systems Management)

DSM (Distributed Systems Management)

definition of [Glossary-4](#)

format template [6-5](#)

DSM (Distributed Systems Management) (continued)  
    programmatic interfaces, list of [-xvii](#)  
    See also DSM Template Services  
    template-lookup key [4-42](#)  
DSV (distribution subvolume), definition of [Glossary-4](#)

## E

Edit code, MSG [4-42](#)  
EMS parameter, in RUN NETBATCH command [6-3](#)  
EMS scheduler attribute  
    See ZEMS field  
EMS (Event Management Service)  
    collector (\$0) [6-3](#)  
    definition of [Glossary-4](#)  
    EMS parameter, NETBATCH program [6-3](#)  
    EMSTEXT procedure [6-5](#)  
    event messages, descriptions of  
        204, ZBAT-EVT-JOB-OVER-LIMIT [6-18/6-19](#)  
        301, ZBAT-EVT-JOB-START-ERROR [6-19/6-25](#)  
    event-message generation, enabling and disabling [6-3](#)  
    event-message handling [6-3](#)  
    event-message selection [6-3](#)  
    filter creation [6-3](#)  
    format template [6-5](#)  
    GETEVENT command [Glossary-2](#)  
    See also Event messages  
    standard definitions, definition of [Glossary-5](#)  
EMSTEXT procedure [6-5](#)  
End-list token, definition of [Glossary-5](#)  
Enumerated type, definition of [Glossary-5](#)  
Error lists  
    definition of [Glossary-5](#)  
    descriptions of [A-1/A-108](#)

Error numbers  
    definition of [Glossary-5](#)  
    list of [A-1/A-108](#)  
Error token, definition of [Glossary-5](#)  
Errors  
    definition of [Glossary-5](#)  
    notation used to describe [A-1](#)  
    open, for scheduler [2-5](#)  
    order of tokens in [A-1](#)  
    See also Warnings  
EVENT job state [4-35](#), [4-36](#)  
Event log  
    definition of [Glossary-5](#)  
    message distribution by distributor [6-3](#), [Glossary-4](#)  
    message logging by collector [6-3](#), [Glossary-1](#)  
Event Management Service (EMS)  
    See EMS  
Event management, definition of [Glossary-5](#)  
Event messages  
    critical [3-14](#), [6-1](#)  
    definition of [Glossary-6](#)  
    descriptions of  
        204, ZBAT-EVT-JOB-OVER-LIMIT [6-18/6-19](#)  
        301, ZBAT-EVT-JOB-START-ERROR [6-19/6-25](#)  
    generation of, enabling and disabling [6-3](#)  
    noncritical [3-14](#), [6-1](#)  
    See also EMS  
Event, definition of [Glossary-5](#)  
    See also Event messages  
EVERY job attribute  
    See ZEVERTY-DAYS field; ZEVERTY-HOURS field; ZEVERTY-MINUTES field [4-55](#)  
EXECUTING job state [4-35](#), [4-36](#)

**Executor**

## attributes

CLASS (ZCLASSES field of ZBAT-MAP-DEF-EXECUTOR) [4-51](#)

CPU (ZCPU field of ZBAT-MAP-DEF-EXECUTOR) [4-51](#)

definition of [Glossary-6](#)

## states

ACTIVE [4-35](#)

DELETE [4-36](#)

DOWN [4-36](#)

OFF [4-36](#)

ON [4-36](#)

STOP [4-36](#)

Executor program, definition of [Glossary-6](#)

EXECUTOR-PROGRAM job attribute

See ZBAT-TKN-, EXECUTOR-PROGRAM

Explicit command, definition of [Glossary-6](#)

Extensible structured tokens, definition of [Glossary-6](#)

Extensible Structured Tokens, table of [4-11](#)

Extensible structure, definition of [Glossary-6](#)

EXTSWAP job attribute

See ZBAT-TKN-, EXTSWAP-FILE

**F****Filters**

compiling [6-3](#)

definition of [Glossary-6](#)

installing [6-3](#)

loading [6-3](#)

Fixed structure, definition of [Glossary-6](#)

Format template, EMS [6-5](#)

Forwarding distributor

definition of [Glossary-6](#)

Forwarding distributor, definition of

See also Distributor

**G**

GETEVENT command, EMS [Glossary-2](#)

GETVERSION command, definition of [Glossary-6](#)

Group manager, definition of [Glossary-6](#)

**H****Header**

See SPI message header

**Header tokens**

defined by SPI [4-2](#)

definition of [Glossary-6](#)

Header type, definition of [Glossary-7](#)

**HIGH PIN**

See also HIGHPIN job attribute; Low PIN

**High PIN**

definition of [Glossary-7](#)

**HIGHPIN job attribute**

See ZHIGHPIN field, of ZBAT-MAP-DEF-JOB

**HOLD job attribute**

See ZHOLD field

**HOLDAFTER job attribute**

See ZHOLD-AFTER field

**I****IFFAILS job attribute**

See ZIFFAILS field

Implicit command, definition of [Glossary-7](#)

**IN job attribute**

See ZBAT-TKN-, IN-FILE

Information token, definition of [Glossary-7](#)

Initial position, definition of [Glossary-7](#)

Initialize, definition of [Glossary-7](#)

**INITIATION class attribute**

See ZINITIATION field, of ZBAT-MAP-DEF-CLASS

## INITIATION scheduler attribute

See ZINITIATION field, of ZBAT-MAP-DEF-SCHEDULER

## Installation subvolumes (ISVs)

See ISVs

Installing filters [6-3](#)

Interactive commands versus programmatic commands, table of [1-6](#)

## Interactive session

See Session

ISVs (installation subvolumes), definition of [Glossary-18](#)

**J**

## Job

## attributes

AFTER (ZAT-FLAG field of ZBAT-MAP-DEF-JOB) [4-53](#)

AT (ZAT-FLAG field of ZBAT-MAP-DEF-JOB) [4-53](#)

ATTACHMENT-SET (ZBAT-TKN-ATT-SET-ID) [4-40](#)

CALENDAR (ZBAT-TKN-CALENDAR) [4-42](#)

CLASS (ZCLASSNAME field of ZBAT-MAP-DEF-JOB) [4-53](#)

DESCRIPTION (ZBAT-TKN-DESCRIPTION) [4-42](#)

EVERY (ZEVERY-DAYS, ZEVERY-HOURS, and ZEVERY-MINUTES fields of ZBAT-MAP-DEF-JOB) [4-55](#)

EXECUTOR-PROGRAM (ZBAT-TKN-EXECUTOR-PROGRAM) [4-42](#)

EXTSWAP (ZBAT-TKN-EXTSWAP-FILE) [4-42](#)

HIGHPIN (ZHIGHPIN field of ZBAT-MAP-DEF-JOB) [4-56](#)

HOLD (ZHOLD field of ZBAT-MAP-DEF-JOB) [4-53](#)

## Job (continued)

HOLDAFTER (ZHOLDAFTER field of ZBAT-MAP-DEF-JOB) [4-53](#)

IFFAILS (ZIFFAILS field of ZBAT-MAP-DEF-JOB) [4-54](#)

IN (ZBAT-TKN-IN-FILE) [4-42](#)

JOBID-ZERO (ZJOBID-ZERO field of ZBAT-MAP-DEF-JOB) [4-58](#)

LIB (ZBAT-TKN-LIB-FILE) [4-43](#)

LIMIT (ZBAT-TKN-TIME-LIMIT) [4-48](#)

## MAXPRINTLINES

(ZMAXPRINTLINES field of ZBAT-MAP-DEF-JOB) [4-56](#)

## MAXPRINTPAGES

(ZMAXPRINTPAGES field of ZBAT-MAP-DEF-JOB) [4-56](#)

MEM (ZMEM field of ZBAT-MAP-DEF-JOB) [4-58](#)

NAME (ZNAME field of ZBAT-MAP-DEF-JOB) [4-58](#)

OUT (ZBAT-TKN-OUT-FILE) [4-43](#)

PFS (ZPFS field of ZBAT-MAP-DEF-JOB) [4-58](#)

PRI (ZPRI field of ZBAT-MAP-DEF-JOB) [4-56](#)

PURGE-IN-FILE (ZPURGE-IN-FILE field of ZBAT-MAP-DEF-JOB) [4-54](#)

RESTART (ZRESTART field of ZBAT-MAP-DEF-JOB) [4-53](#)

RUND (ZRUND field of ZBAT-MAP-DEF-JOB) [4-58](#)

SAVEABEND (ZSAVEABEND field of ZBAT-MAP-DEF-JOB) [4-58](#)

SELPRI (ZSELPRI field of ZBAT-MAP-DEF-JOB) [4-56](#)

STALL (ZSTALL field of ZBAT-MAP-DEF-JOB) [4-54](#)

STARTUP (ZBAT-TKN-STARTUP-MESSAGE) [4-47](#)

STOP-ON-ABEND (ZSTOP-ON-ABEND field of ZBAT-MAP-DEF-JOB) [4-53](#)

## Job (continued)

SWAP (ZBAT-TKN-SWAP-FILE) [4-47](#)

TAPEDRIVES (ZTAPEDRIVES field of ZBAT-MAP-DEF-JOB) [4-56](#)

TERM (ZBAT-TKN-TERM-FILE) [4-47](#)

WAITON (ZBAT-DDL-WAITON-FOR, ZBAT-DDL-WAITON-INDICATOR, ZBAT-MAP-DEF-WAITON) [4-32](#), [4-62](#)

definition of [Glossary-8](#)

dependency, definition of [Glossary-4](#)

See also ZBAT-DDL-, WAITON-FOR; ZBAT-DDL-, WAITON-INDICATOR; ZBAT-MAP-, DEF-WAITON

dependent, definition of [Glossary-4](#)

See also ZBAT-DDL-, WAITON-FOR; ZBAT-DDL-, WAITON-INDICATOR; ZBAT-MAP-, DEF-WAITON

input file, definition of [Glossary-7](#)

See also ZBAT-TKN-, IN-FILE

master, definition of [Glossary-8](#)

See also ZBAT-DDL-, WAITON-FOR; ZBAT-DDL-, WAITON-INDICATOR; ZBAT-MAP-, DEF-WAITON

owner [4-43](#), [4-48](#), [4-54](#), [4-67](#)

run time, definition of [Glossary-12](#)

See also ZDATE field; ZTIME field

states

EVENT [4-35](#), [4-36](#)

EXECUTING [4-35](#), [4-36](#)

READY [4-36](#), [4-37](#)

RUNNEXT [4-36](#), [4-37](#)

RUNNOW [4-37](#)

SPECIAL-1 [4-36](#), [4-38](#)

SPECIAL-2 [4-38](#), [4-40](#)

SPECIAL-3 [4-37](#), [4-38](#)

SPECIAL-4 [4-36](#), [4-38](#)

## Job (continued)

SPECIAL-5 [4-37](#), [4-38](#)

SPECIAL-6 [4-37](#), [4-38](#)

SPECIAL-7 [4-35](#), [4-38](#)

SPECIAL-8 [4-35](#), [4-38](#)

SPECIAL-9 [4-38](#), [4-39](#)

SUSPENDED [4-37](#), [4-39](#)

TAPE [4-37](#), [4-39](#)

TIME [4-37](#), [4-39](#)

JOBID-ZERO job attribute

See ZJOBID-ZERO field

## L

Letter Z, disallowed at beginning of names [3-3](#)

LIB job attribute

See ZBAT-TKN-, LIB-FILE

LIMIT job attribute

See ZBAT-TKN-, TIME-LIMIT

List

definition of [Glossary-8](#)

of errors [A-1](#)

List token, definition of [Glossary-8](#)

Loading filters [6-3](#)

LOCALNAMES scheduler attribute

See ZLOCALNAMES field

Low PIN, definition of [Glossary-8](#)

See also High PIN

## M

Macro, definition of [Glossary-8](#)

Management application, definition of [Glossary-8](#)

Management interface, definition of [Glossary-8](#)

Management process, definition of [Glossary-8](#)

Maximum field version, definition of [Glossary-9](#)



**MAXPRINTLINES** job attribute

See ZMAXPRINTLINES field, of ZBAT-MAP-DEF-JOB

**MAXPRINTPAGES** job attribute

See ZMAXPRINTPAGES field, of ZBAT-MAP-DEF-JOB

**MAX-CONCURRENT-JOBS** scheduler attribute

See ZMAXCONCURRENTJOBS field;  
ZMAXTEMPEXECUTORS field [4-60](#)

**MAX-PRI** scheduler attribute

See ZMAXPRI field

**MEM** job attribute

See ZMEM field

**Message** buffer

definition of [Glossary-9](#)

**Message** buffer, definition of

See also Buffers

**Message** code, definition of [Glossary-9](#)**Messages**

definition of [Glossary-9](#)

event

See Event messages

**Midnight**, specifying [4-57](#)**MSG** edit code [4-42](#)**N****NAME** job attribute

See ZNAME field

**Names**

guidelines for, in applications [3-3](#)

of ASSIGNS [4-45](#)

of attachment sets [4-40](#)

of built-ins [Glossary-1](#)

of classes [4-45](#)

of DEFINES [4-45](#)

of EMS standard definitions [Glossary-5](#)

of executors [4-45](#)

of executor-program processes

See ZNAME field

**Names** (continued)

of jobs [4-46](#)

of object types [4-19](#)

of PARAMs [4-47](#)

of SPI standard definitions [Glossary-15](#)

symbolic, of error numbers [A-1](#)

ZBAT- [B-2](#), [B-5](#)

ZSPI- [4-2](#)

**NBEXEC**

BPROC [1-2](#)

core component of NetBatch  
product [1-2](#)

definition of [1-2](#), [Glossary-9](#)

**NB^JOB^SUBMIT**

core component of NetBatch  
product [1-2](#)

definition of [Glossary-9](#)

**NetBatch** product

buffer declarations, table of [4-8](#)

core components [1-2](#)

event-management interface [1-3](#)

interactive commands versus

programmatic commands [1-6](#)

objects managed by [1-5](#)

object-management functions,  
programmatic [1-5](#)

overview [1-2](#)

predefined token and field values, table  
of [4-8](#)

private token and field types, table  
of [4-8](#)

programmatic interface [1-3](#)

programs

calendar (BATCHCAL) [1-2](#)

command interpreter, interactive  
(BATCHCOM) [1-2](#)

NonStop executor (NBEXEC) [1-2](#)

scheduler (NETBATCH) [1-2](#)

See also NETBATCH program

simple tokens, table of [4-8](#)



## NetBatch product (continued)

## subsystem

- communicating with, steps in [2-1](#)
- error lists issued by [A-1](#)
- error numbers issued by [A-1](#)
- overview [1-2](#)
- tokens, codes and types, table of [B-2/B-4](#)
- tokens, maps and DDL definitions, table of [B-5/B-12](#)

## NETBATCH program

- core component of NetBatch product [1-2](#)
- definition of [1-2](#), [Glossary-9](#)
- EMS parameter [6-3](#)
- running interactively [2-2](#)
- See also NetBatch product

NetBatch supervisor, definition of [Glossary-9](#)NetBatch-Plus, definition of [Glossary-9](#)Next position, definition of [Glossary-9](#)

## Noncritical events

- definition of [3-14](#), [Glossary-9](#)
- flagged by ZEMS-TKN-EMPHASIS [4-7](#), [6-1](#)
- for NetBatch subsystem [6-1](#)

## Noninteractive session

See Session

Nonsensitive commands, definition of [Glossary-9](#)Non-super-group user, definition of [Glossary-9](#)Null object type, definition of [Glossary-10](#)Null value, definition of [Glossary-10](#)

## Numbers

- of errors, in NetBatch subsystem responses [A-1](#)
- of Guardian procedure errors [4-44](#)
- of object types [4-19](#)

## O

## Object types

- definition of [Glossary-10](#)
- names of [4-19](#)
- numbers of [4-19](#)

## Objects

- definition of [Glossary-10](#)
- managed by NetBatch subsystem [1-5](#)

Object-name token, definition of [Glossary-10](#)Object-selector token, definition of [Glossary-10](#)Object-type token, definition of [Glossary-10](#)OFF executor state [4-36](#)ON executor state [4-36](#)Open errors, for scheduler [2-5](#)

## Options, run

See Run options

## OUT job attribute

See ZBAT-TKN-, OUT-FILE

## Owner

- definition of [Glossary-10](#)
- of job [4-43](#), [4-48](#), [4-54](#), [4-67](#)
- of subsystem [Glossary-15](#)

## P

## PARAM attachment-set attribute

See ZBAT-TKN-, ATT-SET-PARAM

Parameter token, definition of [Glossary-11](#)PARAM, definition of [Glossary-10](#)

## PFS job attribute

See ZPFS field

## PIN

definition of [Glossary-11](#)

## PIN, definition of

See also High PIN; Low PIN

## Predefined token and field values

table of [4-8](#)

Predefined value, definition of [Glossary-11](#)

PRI job attribute

See ZPRI field, of ZBAT-MAP-DEF-JOB

Printing distributor

definition of [Glossary-11](#)

Printing distributor, definition of

See also Distributor

Private token and field types, table of [4-8](#)

Private token type, definition

of [Glossary-11](#)

Procedural interface, definition

of [Glossary-11](#)

Procedure calls

Guardian

ABEND [4-42](#), [4-48](#), [6-2](#), [6-3](#), [6-15](#),  
[6-18](#), [6-19](#)

CONVERTTIMESTAMP [A-71](#)

DEFINESETATTR [4-64](#)

DEFINEVALIDATEWORK [4-64](#)

EMSTEXT [6-5](#)

PROCESS\_CREATE\_ [4-20](#), [4-44](#),  
[6-20](#), [6-24](#)

PROCESS\_STOP\_ [4-42](#), [4-48](#),  
[6-2](#), [6-3](#), [6-15](#), [6-18](#), [6-19](#)

STOP [4-42](#), [4-48](#), [6-15](#), [6-19](#)

SPI

SSGET [A-79](#)

SSGETTKN [A-79](#)

SSNULL [A-99](#)

Process identification number

See PIN

PROCESS\_CREATE\_ procedure [4-20](#),  
[4-44](#), [6-20](#), [6-24](#)

PROCESS\_STOP\_ procedure [4-42](#), [4-48](#),  
[6-2](#), [6-3](#), [6-15](#), [6-18](#), [6-19](#)

Programmatic commands

definition of [Glossary-11](#)

versus interactive commands, table  
of [1-6](#)

Programmatic interface, definition

of [Glossary-11](#)

PURGE-IN-FILE job attribute

See ZPURGE-IN-FILE field

## R

READY job state [4-36](#), [4-37](#)

Receiving a response message [3-7](#)

Requester version, definition  
of [Glossary-11](#)

Response message, definition  
of [Glossary-12](#)

Response records, definition  
of [Glossary-12](#)

Response token, definition of [Glossary-12](#)

Responses

definition of [Glossary-11](#)

receiving and decoding [3-7](#)

Response-control token, definition  
of [Glossary-12](#)

RESTART job attribute

See ZRESTART field

Return token, definition of [Glossary-12](#)

Routine, definition of [Glossary-12](#)

Run calendar

definition of [Glossary-12](#)

Run calendar, definition of

See also ZBAT-TKN-, CALENDAR

RUN NETBATCH command

EMS parameter [6-3](#)

programmatic and interactive scheduler  
startup, use in [2-2](#)

Run options, for executor-program  
processes

CPU

See ZCPU field, of ZBAT-MAP-  
DEF-EXECUTOR

EXTSWAP

See ZBAT-TKN-, EXTSWAP-FILE

HIGHPIN

See ZHIGHPIN field, of ZBAT-MAP-  
DEF-JOB

IN

See ZBAT-TKN-, IN-FILE

JOBID

See ZJOBID-ZERO field

## processes (continued)

## LIB

See ZBAT-TKN-, LIB-FILE

## MEM

See ZMEM field

## NAME

See ZNAME field

## OUT

See ZBAT-TKN-, OUT-FILE

## PFS

See ZPFS field

## PRI

See ZPRI field, of ZBAT-MAP-DEF-JOB

SAVEABEND (INSPECT run-option value)

See ZSAVEABEND field

## SWAP

See ZBAT-TKN-, SWAP-FILE

## TERM

See ZBAT-TKN-, TERM-FILE

Run time, job, definition of [Glossary-12](#)

See also ZDATE field; ZTIME field [Glossary-12](#)

RUND job attribute

See ZRUND field

RUNNEXT job state [4-36](#), [4-37](#)

RUNNOW job state [4-37](#)

## S

Sample programs, SPI

TACL [C-32/C-40](#)

SAVEABEND job attribute

See ZSAVEABEND field

## Scheduler

attributes

AT-ALLOWED (ZAT-ALLOWED field of ZBAT-MAP-DEF-SCHEDULER) [4-60](#)

## Scheduler (continued)

BACKUPCPU (ZBACKUPCPU1 and ZBACKUPCPU2 fields of ZBAT-MAP-DEF-SCHEDULER) [4-59](#)

CATCHUP (ZEVERY-CATCHUP field of ZBAT-MAP-DEF-SCHEDULER) [4-61](#)

DEFAULT-CLASS (ZCLASSNAME field of ZBAT-MAP-DEF-SCHEDULER) [4-61](#)

DEFAULT-EXECUTOR-PROGRAM (ZBAT-TKN-EXECUTOR-PROGRAM) [4-42](#)

DEFAULT-HIGHPIN (ZHIGHPIN field of ZBAT-MAP-DEF-SCHEDULER) [4-62](#)

DEFAULT-MAXPRINTLINES (ZMAXPRINTLINES field of ZBAT-MAP-DEF-SCHEDULER) [4-61](#)

DEFAULT-MAXPRINTPAGES (ZMAXPRINTPAGES field of ZBAT-MAP-DEF-SCHEDULER) [4-61](#)

DEFAULT-OUT (ZBAT-TKN-OUT-FILE) [4-43](#)

DEFAULT-PRI (ZPRI field of ZBAT-MAP-DEF-SCHEDULER) [4-61](#)

DEFAULT-SELPRI (ZSELPRI field of ZBAT-MAP-DEF-SCHEDULER) [4-61](#)

DEFAULT-STALL (ZSTALL field of ZBAT-MAP-DEF-SCHEDULER) [4-62](#)

DEFAULT-STOP-ON-ABEND (ZSTOP-ON-ABEND field of ZBAT-MAP-DEF-SCHEDULER) [4-62](#)

EMS (ZEMS field of ZBAT-MAP-DEF-SCHEDULER) [4-61](#)

INITIATION (ZINITIATION field of ZBAT-MAP-DEF-SCHEDULER) [4-62](#)

LOCALNAMES (ZLOCALNAMES field of ZBAT-MAP-DEF-SCHEDULER) [4-62](#)

## Scheduler (continued)

MAX-CONCURRENT-JOBS  
(ZMAXCONCURRENTJOBS and  
ZMAXTEMPEXECUTORS fields of  
ZBAT-MAP-DEF-  
SCHEDULER) [4-60](#)

MAX-PRI (ZMAXPRI field of ZBAT-  
MAP-DEF-SCHEDULER) [4-60](#)

SUBMIT-ALLOWED (ZSUBMIT-  
ALLOWED field of ZBAT-MAP-DEF-  
SCHEDULER) [4-61](#)

TAPEDRIVES (ZTAPEDRIVES field  
of ZBAT-MAP-DEF-  
SCHEDULER) [4-60](#)

configuration file

See ZBAT-TKN-, BATCHCTL

definition of [1-2](#), [Glossary-13](#)

See also NETBATCH program

startup [2-2](#)

SECURITY attachment-set attribute

See ZBAT-TKN-, ATT-SET-SECURITY

SELPRI job attribute

See ZSELPRI field, of ZBAT-MAP-DEF-  
JOB

Sensitive commands, definition  
of [Glossary-13](#)

Server version, definition of [Glossary-13](#)

Session, definition of [Glossary-13](#)

Set, attachment

See Attachment set

Simple tokens

defined by NetBatch, table of [4-8](#)

defined by SPI [4-2](#)

definition of [Glossary-13](#)

Special operation, definition of [Glossary-13](#)

SPECIAL-1 job state [4-36](#), [4-38](#)

SPECIAL-2 job state [4-38](#), [4-40](#)

SPECIAL-3 job state [4-37](#), [4-38](#)

SPECIAL-4 job state [4-36](#), [4-38](#)

SPECIAL-5 job state [4-37](#), [4-38](#)

SPECIAL-6 job state [4-37](#), [4-38](#)

SPECIAL-7 job state [4-35](#), [4-38](#)

SPECIAL-8 job state [4-35](#), [4-38](#)

SPECIAL-9 job state [4-38](#), [4-39](#)

SPI (Subsystem Programmatic Interface)

buffer, definition of [Glossary-14](#)

control code, definition of [Glossary-14](#)

definition of [Glossary-14](#)

definitions, definition of [Glossary-14](#)

error number, definition of [Glossary-14](#)

message header, definition  
of [Glossary-14](#)

message, definition of [Glossary-14](#)

procedures, definition of [Glossary-14](#)

programming considerations for  
NetBatch subsystem [3-1/3-14](#)

See also NB^JOB^SUBMIT

standard definitions

definition of [Glossary-15](#)

tables of [4-2/4-3](#)

SSGET procedure [A-79](#)

SSGETTKN procedure [A-79](#)

SSNULL procedure [A-99](#)

STALL job attribute

See ZSTALL field, of ZBAT-MAP-DEF-  
JOB

Standard definitions, SPI [4-2/4-5](#)

START SCHEDULER command

interactive scheduler startup, use in [2-2](#)

STARTUP job attribute

See ZBAT-TKN-, STARTUP-MESSAGE

Startup, scheduler [2-2](#)

STOP EXECUTOR command, tokens in  
command buffer [5-60](#)

STOP executor state [4-36](#)

STOP procedure [4-42](#), [4-48](#), [6-15](#), [6-19](#)

STOP-ON-ABEND job attribute

See ZSTOP-ON-ABEND field, of ZBAT-  
MAP-DEF-JOB

Structured token, definition of [Glossary-15](#)

Structures

defined by NetBatch [4-8](#)

definition of [Glossary-15](#)

Subjects, definition of [Glossary-15](#)

SUBMIT-ALLOWED scheduler attribute  
See ZSUBMIT-ALLOWED field

Subsystem  
definition of [Glossary-15](#)  
definitions, definition of [Glossary-15](#)  
ID, definition of [Glossary-15](#)  
number, definition of [Glossary-15](#)  
owner  
definition of [Glossary-15](#)  
for NetBatch  
subsystem [Glossary-15](#)  
version number, definition  
of [Glossary-16](#)

Subsystem Programmatic Interface (SPI)  
See EMS interface; NB^JOB^SUBMIT;  
SPI

Subsystem-control token, definition  
of [Glossary-16](#)

Supervisor, NetBatch, definition  
of [Glossary-9](#)

Super-group user, definition of [Glossary-16](#)

Super-ID, definition of [Glossary-16](#)

SUSPENDED job state [4-37](#), [4-39](#)

SWAP job attribute  
See ZBAT-TKN-, SWAP-FILE

Symbolic names  
definition of [Glossary-16](#)  
of error numbers [A-1](#)

Syntax token, definition of [Glossary-16](#)

## T

TACL programming language  
definition of [Glossary-16](#)  
RUN command  
EMS parameter [6-3](#)  
interactive scheduler startup, use  
in [2-2](#)  
sample macros, SPI [C-30](#)  
text variables

TACL programming language (continued)  
representing command  
numbers [Glossary-2](#)  
representing object-type  
numbers [Glossary-10](#)

TAL programming language  
definition of [Glossary-16](#)  
FILE\_OPEN\_ procedure [2-4](#)  
LITERAL or DEFINE declarations  
representing command  
numbers [Glossary-2](#)  
representing object-type  
numbers [Glossary-10](#)  
procedure NB^JOB^SUBMIT  
See NB^JOB^SUBMIT  
sample programs, SPI [C-41](#)

TAPE job state [4-37](#), [4-39](#)

TAPEDRIVES job attribute  
See ZTAPEDRIVES field, of ZBAT-  
MAP-DEF-JOB

TAPEDRIVES scheduler attribute  
See ZTAPEDRIVES field, of ZBAT-  
MAP-DEF-SCHEDULER

Template services  
See DSM Template Services

Template, format (EMS) [6-5](#)

Template-lookup key [4-42](#)

TEMPORARY attachment-set attribute  
See ZBAT-TKN-, ATT-SET-  
TEMPORARY

TERM job attribute  
See ZBAT-TKN-, TERM-FILE

Time attributes, definition of [Glossary-16](#)

TIME job state [4-37](#), [4-39](#)

Tokens  
codes and types, table of [B-2/B-4](#)  
code, definition of [Glossary-17](#)  
conditional, definition of [Glossary-2](#)  
context, definition of [Glossary-2](#)  
current, definition of [Glossary-3](#)  
data type, definition of [Glossary-17](#)

## Tokens (continued)

data-portion, definition of [Glossary-3](#)  
 DDL definitions, of token maps, table of [B-5/B-12](#)  
 defined by SPI [4-2/4-5](#)  
 definition of [Glossary-17](#)  
 end-list, definition of [Glossary-5](#)  
 error, definition of [Glossary-5](#)  
 extensible structured, definition of [Glossary-6](#)  
 header [4-2](#)  
     definition of [Glossary-6](#)  
 information, definition of [Glossary-7](#)  
 length, definition of [Glossary-17](#)  
 list, definition of [Glossary-8](#)  
 maps and DDL definitions, table of [B-5/B-12](#)  
 map, definition of [Glossary-17](#)  
 number, definition of [Glossary-17](#)  
 object-name, definition of [Glossary-10](#)  
 object-selector, definition of [Glossary-10](#)  
 order of in error lists [A-1](#)  
 parameter, definition of [Glossary-11](#)  
 response, definition of [Glossary-12](#)  
 response-control, definition of [Glossary-12](#)  
 return, definition of [Glossary-12](#)  
 simple, definition of [Glossary-13](#)  
 structured, definition of [Glossary-15](#)  
 subsystem-control, definition of [Glossary-16](#)  
 syntax, definition of [Glossary-16](#)  
 types  
     defined by NetBatch [4-8](#)  
     defined by SPI [4-2/4-3](#)  
     definition of [Glossary-17](#)  
 types and codes, table of [B-2/B-4](#)  
 unconditional, definition of [Glossary-18](#)  
 value, definition of [Glossary-17](#)

## Tokens (continued)

ZBAT simple [4-8](#)  
 ZSPI simple [4-2](#)  
 Token-oriented, definition of [Glossary-17](#)  
 Types, token, and token codes, table of [B-2/B-4](#)

## U

Unconditional token, definition of [Glossary-18](#)  
 Upward compatibility, definition of [Glossary-18](#)  
     See also Downward compatibility;  
     Version compatibility

## V

Value names, defined by NetBatch [4-8](#)  
 Version compatibility, definition of [Glossary-18](#)  
 Version number, definition of [Glossary-18](#)  
 ViewPoint console application, definition of [Glossary-18](#)

## W

WAITON job attribute  
     See ZBAT-DDL-, WAITON-INDICATOR;  
     ZBAT-MAP-, DEF-WAITON  
 Warm start, scheduler [2-2](#)  
 Warnings  
     definition of [Glossary-18](#)  
     See also Errors  
 Wild-card characters  
     characters matched [Glossary-18](#)  
     definition of [Glossary-18](#)

## Z

Z (letter), disallowed at beginning of names [3-3](#)



- ZACTIVE field
  - of ZEXECUTOR in ZBAT-MAP-STATUS-SCHEDULER [4-70](#)
  - of ZPROCESS in ZBAT-MAP-STATUS-SCHEDULER [4-72](#)
- ZADDR field [4-64](#)
- ZATTRLEN field [4-64](#)
- ZATTRTXT field [4-64](#)
- ZATT-SET-COUNT field [4-72](#)
- ZAT-ALLOWED field [4-60](#)
- ZAT-FLAG field [4-53](#)
- ZBACKUPCPU1 field [4-60](#)
  - See also ZBACKUPCPU2 field
- ZBACKUPCPU2 field [4-59](#)
  - See also ZBACKUPCPU1 field
- ZBAT-DDL-
  - DEFINE-ERROR [4-64](#)
  - definitions, tables of [4-8](#), [B-5/B-12](#)
  - DEF-CLASS [4-49](#)
  - DEF-CRONTAB [4-50](#)
  - DEF-JOB [4-53](#)
  - DEF-WAITON [4-63](#)
  - OBJECT [4-19](#)
  - PAR-RELEASE-JOB [4-19](#)
  - PC-ERROR0 [4-20](#)
  - PC-ERROR1 [4-22](#)
  - RETCODE [4-24](#)
  - SCHEDULER-STATE [4-31](#)
  - SPECIAL-REASON [4-31](#)
  - STATUS-EXECUTOR [4-32](#)
  - STATUS-JOB [4-32](#), [4-67](#)
  - STATUS-SCHEDULER [4-32](#)
  - WAITON-FOR [4-32](#)
  - WAITON-INDICATOR [4-32](#)
- ZBAT-ERR-
  - ACTIVATE [A-82](#)
  - AFTER-DAY [A-24](#)
  - AFTER-MICROSEC [A-93](#)
  - AFTER-MINUTE [A-26](#)
  - AFTER-MONTH [A-23](#)
- ZBAT-ERR- (continued)
  - AFTER-SECOND [A-91](#)
  - ALREADY-STARTED [A-26](#)
  - AT [A-27](#)
  - ATT-DEFINE [A-89](#)
  - ATT-JOB [A-73](#)
  - ATT-OVERFLOW [A-76](#)
  - ATT-PARAM [A-90](#)
  - ATT-REQUESTOR [A-74](#)
  - ATT-UPDATE [A-77](#)
  - AT-FLAG [A-86](#)
  - CALENDAR-FILECODE [A-28](#)
  - CLASS-COUNT [A-29/A-30](#)
  - CLASS-EXISTS [A-41](#)
  - CLASS-IN-USE [A-42/A-43](#)
  - CONTEXT [A-30/A-31](#)
  - CRONTAB [A-97](#)
  - DESCRIPTION [A-102](#)
  - EMPTY-RESPONSE [A-45](#)
  - EMS [A-108](#)
  - EVERY [A-34](#)
  - EVERY-CATCHUP [A-107/A-108](#)
  - EXECUTOR-EXISTS [A-36/A-37](#)
  - EXECUTOR-PROG [A-35](#)
  - EXTRA-TOKEN [A-37/A-38](#)
  - FILE-ERROR [A-78](#)
  - HOLDAFTER [A-31/A-32](#)
  - IN [A-39](#)
  - INTERNAL-ERROR [A-77](#)
  - INVALID-COMMAND [A-53](#)
  - INVALID-SPI [A-79](#)
  - JOBID-ZERO [A-100](#)
  - JOBNAME [A-43](#)
  - JOBNAME-EXISTS [A-44](#)
  - JOBNAME-REQUIRED [A-44](#)
  - JOB-DUPL-ATT [A-90](#)
  - JOB-TOO-MANY-ATT [A-84](#)
  - MAXCONCURRENTJOBS [A-106](#)
  - MAXPRI [A-105](#)

## ZBAT-ERR- (continued)

MAXPRINTPAGES [A-47](#)  
MAXTEMPEXECUTORS [A-107](#)  
MEM [A-101/A-102](#)  
MISSING-EXECUTOR [A-86](#)  
MULTIPLE-CONTEXT [A-49](#)  
NETBATCH-NAME [A-80/A-81](#)  
NODENAME [A-104/A-105](#)  
NOT-IMPLEMENTED [A-78](#)  
NO-SUBMIT [A-52](#)  
NO-SUCH-EXECUTOR [A-37](#)  
NO-SUCH-JOB [A-41](#)  
PFS [A-108](#)  
POSIX [A-99](#)  
PRI [A-56](#)  
PURGE-IN-FILE [A-98](#)  
RESTART [A-56](#)  
RUND [A-100](#)  
RUNNEXT [A-58](#)  
RUNNEXT-RUNNOW [A-59](#)  
SAVEABEND [A-99](#)  
SECURITY [A-54](#)  
SELPRI [A-61](#)  
SHUTDOWN [A-54](#)  
STOP [A-83](#)  
SUSPEND [A-81](#)  
SWITCHCPU [A-68](#)  
TIME [A-85](#)  
TIME-LIMIT [A-102](#)  
UNKNOWN-OBJECT [A-65](#)  
USER-UNDEFINED [A-40](#)  
VOLUME [A-66](#)  
VOLUME-REQUIRED [A-66](#)  
WAITON-COUNT [A-67](#)  
WAITON-ID [A-95/A-96](#)  
WAITON-JOBS-DUPL [A-67](#)  
WAITON-SELF [A-36](#)  
WILDCARD [A-84](#)

## ZBAT-EVT-

JOB-OVER-LIMIT event message,  
description of [6-18/6-19](#)  
JOB-START-ERROR event message,  
description of [6-19/6-25](#)

## ZBAT-MAP-

## DEFINE-ERROR

description [4-64/4-65](#)  
structure [4-64](#)  
definitions, table of [4-11](#), [B-5/B-12](#)

## DEF-CLASS

description [4-49](#)  
structure [4-49](#)

## DEF-CRONTAB

description [4-50](#)  
structure [4-50](#), [4-50](#)

## DEF-EXECUTOR

description [4-51](#)  
structure [4-51](#)

## DEF-JOB

description [4-53/4-57](#)  
structure [4-53](#)

## DEF-SCHEDULER

description [4-59/4-62](#)  
structure [4-59](#)

## DEF-WAITON

description [4-63](#)  
structure [4-63](#)

## PAR-RELEASE-JOB

description [4-65](#)  
structure [4-65](#)

## STATUS-EXECUTOR

description [4-66](#)  
structure [4-66](#)

## STATUS-JOB

description [4-67/4-69](#)  
structure [4-67](#)

## STATUS-SCHEDULER

description [4-70](#)



## ZBAT-MAP- (continued)

structure [4-70](#)

## ZBAT-OBJ-

values of [4-19](#)

## ZBAT-TKN-

ATT-SET-ASSIGN [4-40](#)

ATT-SET-DEFINE [4-40](#)

ATT-SET-ID

general description of [4-40](#)

in ADD ATTACHMENT-SET  
command [5-12](#)

in ALTER ATTACHMENT-SET  
command [5-18](#)

ATT-SET-PARAM [4-40](#)

ATT-SET-SECURITY [4-41](#)

ATT-SET-TEMPORARY [4-41](#)

BATCHCTL [4-41](#)

BYTESTRING [4-41](#)

CALENDAR [4-42](#)

CHAR6 [4-42](#)

COMMAND [4-42](#)

COMPLETION-CODE, general  
description of [4-42](#)

DATA-BASE [4-42](#)

definitions, table of [4-8/4-11](#), [B-2/B-4](#)

DESCRIPTION [4-42](#)

EXECUTOR-ID [4-42](#)

EXECUTOR-PROGRAM, general  
description of [4-42](#)

EXTSWAP-FILE [4-42](#)

FORMATSUBJECT [4-42](#)

INT [4-43](#)

INT2 [4-43](#)

IN-FILE [4-42](#)

JOB-ID [4-43](#)

JOB-NAME-ID [4-43](#)

JOB-NUMBER [4-43](#)

LIB-FILE [4-43](#)

LOG-FILE [4-43](#)

MIN-MAX-ERROR [4-43](#)

## ZBAT-TKN- (continued)

NETBATCH-NAME [4-43](#)

OBJECT [4-43](#)

OUT-FILE, general description of [4-43](#)

PC-ERROR0

general description of [4-44](#)

in ZBAT-EVT-JOB-START-ERROR  
message [6-20](#)

PC-ERROR1

general description of [4-44](#)

in ZBAT-EVT-JOB-START-ERROR  
message [6-22](#)

PC-ERROR2

general description of [4-44](#)

in ZBAT-EVT-JOB-START-ERROR  
message [6-23](#)

PHANDLE [4-44](#)

REASON-NUMBER

general description of [4-44](#)

in ZBAT-EVT-JOB-START-ERROR  
message [6-20](#)

RETCODE [4-44](#)

SCHEDULER-ID [4-44](#)

SEL-ADPNAME [4-45](#)

SEL-ASSIGN-NAME [4-45](#)

SEL-CLASSNAME

general description of [4-45](#)

in ALTER CLASS command [5-20](#)

SEL-CLASS-NAME

See ZBAT-TKN, SEL-CLASSNAME

SEL-DEFINE-NAME [4-45](#)

SEL-EXECUTORNAME, general  
description of [4-45](#)

SEL-EXECUTOR-NAME

See ZBAT-TKN, SEL-  
EXECUTORNAME

SEL-INNAME [4-45](#)

SEL-JOBNAME

general description of [4-46](#)

in RUNNEXT JOB command [5-46](#)

## ZBAT-TKN- (continued)

in STATUS JOB command [5-57](#)  
 SEL-JOB-NAME  
     See ZBAT-TKN, SEL-JOBNAME  
 SEL-JOB-NUMBER, general  
 description of [4-45](#)  
 SEL-LIST [4-46](#)  
 SEL-NETBATCH-NAME [4-46](#)  
 SEL-NOTADPNAME [4-46](#)  
 SEL-NOTCLASSNAME [4-46](#)  
 SEL-NOTINNAME [4-46](#)  
 SEL-NOTJOBNAME [4-46](#)  
 SEL-NOTLIST [4-46](#)  
 SEL-NOTUSERNAME [4-47](#)  
 SEL-NOTWAITON [4-47](#)  
 SEL-PARAM-NAME [4-47](#)  
 SEL-USERNAME [4-47](#)  
 SEL-WAITON [4-47](#)  
 STARTUP-MESSAGE [4-47](#)  
 START-TIME [4-47](#)  
 STRING [4-47](#)  
 SWAP-FILE [4-47](#)  
 TERMINATION-INFO [4-48](#)  
 TERM-FILE [4-47](#)  
 TEXT [4-48](#)  
 TIME-LIMIT [4-48](#)  
 TOTAL-CPU-TIME [4-48](#)  
 USERID [4-48](#)  
 VOLUME-SUBVOL  
     general description of [4-48](#)  
     in ALTER ATTACHMENT-SET  
     command [5-19](#)

## ZBAT-TYP-

CHAR6 [4-33](#)  
 COMMAND [4-33](#)  
 COMPLETION-CODE [4-33](#)  
 definitions, table of [4-8](#)  
 INT2-TRIO [4-33](#)  
 JOB-NUMBER [4-33](#)

## ZBAT-TYP- (continued)

LIST [4-33](#)  
 NETBATCH-NAME [4-33](#)  
 OBJECT [4-33](#)  
 PC-ERROR0 [4-34](#)  
 PC-ERROR1 [4-34](#)  
 REASON [4-34](#)  
 RETCODE [4-34](#)

## ZBAT-VAL-

BUFLEN [4-35](#)  
 BUFLLEN-MAX [4-35](#)  
 BUFLLEN-MIN [4-35](#)  
 CALENDAR-EMPTY [4-35](#)  
 CALENDAR-ERROR [4-35](#)  
 definitions, table of [4-8](#)  
 EMS-ERROR [4-35](#)  
 EMS-OFF [4-35](#)  
 EMS-ON [4-35](#)  
 EVENT [4-35](#)  
 EXECUTING [4-35](#)  
 EXECUTOR-ACTIVE-LIST [4-35](#)  
 EXECUTOR-DELETE-LIST [4-36](#)  
 EXECUTOR-DOWN-LIST [4-36](#)  
 EXECUTOR-OFF-LIST [4-36](#)  
 EXECUTOR-ON-LIST [4-36](#)  
 EXECUTOR-STOP-LIST [4-36](#)  
 EXTERNAL-SSID [4-36](#)  
 FAIL-AFTER-CREATE [4-36](#)  
 FIRST-LIST [4-36](#)

See also ZBAT-VAL-, LAST-LIST

HOLD-ON [4-36](#)  
 JOB-EVENT-LIST [4-36](#)  
 JOB-EXECUTING-LIST [4-36](#)  
 JOB-READY-LIST [4-36](#)  
 JOB-RUNNEXT-LIST [4-36](#)  
 JOB-RUNNOW-LIST [4-37](#)  
 JOB-SPECIAL-LIST [4-37](#)  
 JOB-SUSPENDED-LIST [4-37](#)  
 JOB-TAPE-LIST [4-37](#)

## ZBAT-VAL- (continued)

JOB-TIME-LIST [4-37](#)LAST-LIST [4-37](#)

See also ZBAT-VAL-, FIRST-LIST

NEWPROCESS-ERROR [4-37](#)READY [4-37](#)RESTART-OFF [4-37](#)RESTART-ON [4-37](#)RUNNEXT [4-37](#)RUNNOW [4-37](#)SCHEDULER-ZNOTSTARTED [4-38](#)SCHEDULER-ZSHUTDOWN [4-38](#)SCHEDULER-ZSTARTED [4-38](#)SPECIAL-1 [4-38](#)SPECIAL-2 [4-38](#)SPECIAL-3 [4-38](#)SPECIAL-4 [4-38](#)SPECIAL-5 [4-38](#)SPECIAL-6 [4-38](#)SPECIAL-7 [4-38](#)SPECIAL-8 [4-38](#)SPECIAL-9 [4-38](#)SPECIAL-ANY [4-38](#)SSID [4-39](#)STALL [4-39](#)SUSPENDED [4-39](#)TAPE [4-39](#)TIME [4-39](#)VERSION [4-39](#)WAITON-RELEASED-OK [4-39](#)WAITON-REMOVE [4-39](#)WAITON-RESET [4-39](#)WAITON-SET [4-39](#)WAITON-STOP [4-39](#)WAITON-STOPABEND [4-39](#)WAS-RUNNING [4-40](#)

## ZBAT-WRN-

DISALLOW-DEFINE [A-18](#)EXECUTOR-STARTED [A-4](#)

## ZBAT-WRN- (continued)

IN-NE [A-3](#)JOB-EXECUTING [A-5](#)NOT-NETWORKABLE [A-6](#)R-ACCESS [A-10](#)SAME-SYSTEM [A-18](#)WAITON-SATISFIED [A-5/A-6](#)ZCHECKNUM field [4-64](#)

ZCLASS field

of ZBAT-MAP-DEF-EXECUTOR [4-51](#)of ZBAT-MAP-STATUS-  
EXECUTOR [4-66](#)ZCLASSLEN field [4-64](#)

ZCLASSNAME field

of ZBAT-MAP-DEF-JOB [4-53](#)of ZBAT-MAP-DEF-SCHEDULER [4-61](#)of ZBAT-MAP-STATUS-JOB [4-68](#)of ZCLASSES in ZBAT-MAP-DEF-  
EXECUTOR [4-51](#)ZCLASSTXT field [4-64](#)ZCLASS-COUNT field [4-51](#)ZCONFIG field [4-72](#)

ZCPU field

of ZBAT-MAP-DEF-EXECUTOR [4-51](#)of ZBAT-MAP-STATUS-  
EXECUTOR [4-66](#)ZDATE field [4-56](#)ZDAY field [4-57](#)ZDAYS field [4-50](#)ZDEFAULT-SECURITY field [4-55](#)ZDELETE field [4-71](#)ZDOWN field [4-71](#)ZEMS field [4-61](#)

ZEMS-TKN-

EMPHASIS [3-14](#)ZERR field [4-64](#)ZEVENT field [4-71](#)ZEVERY-CATCHUP field [4-61](#)ZEVERY-DAYS field [4-55](#)ZEVERY-HOURS field [4-55](#)ZEVERY-MINUTES field [4-55](#)

- ZEXECPHANDLE field [4-69](#)
- ZEXECUTING field [4-71](#)
- ZEXECUTOR field
  - of ZBAT-MAP-STATUS-JOB [4-68](#)
  - of ZBAT-MAP-STATUS-SCHEDULER [4-70](#)
- ZFOR field [4-63](#)
- ZGROUP field [4-67](#)
- ZHIGHPIN field
  - of ZBAT-MAP-DEF-JOB [4-56](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-62](#)
- ZHOLD field [4-53](#)
- ZHOLD-AFTER field [4-53](#)
- ZHOUR field [4-57](#)
- ZHOURS field [4-50](#)
- ZIFFAILS field [4-54](#)
- ZINDICATOR field [4-63](#)
- ZINFO-LAST-MOD field [4-58](#)
- ZINFO-LAST-MODUSER field [4-59](#)
- ZINFO-NEXT-RUNTIME field [4-54](#)
- ZINFO-OPEN-ACCESSOR field [4-54](#)
- ZINFO-OUT-SPOOL-NUM field [4-54](#)
- ZINFO-SPECIAL-REASON field [4-54](#)
- ZINFO-TAPEDRIVES-IN-USE field [4-60](#)
  - See also ZTAPEDRIVES-IN-USE field
- ZINFO-TIME-SUBMIT field [4-58](#)
- ZINFO-TOTAL-CPU-TIME field [4-54](#)
- ZINFO-WHICH-LIST field [4-54](#)
- ZINITIATION field
  - of ZBAT-MAP-DEF-CLASS [4-49](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-62](#)
  - of ZBAT-MAP-STATUS-SCHEDULER [4-72](#)
- ZJOB field [4-71](#)
- ZJOBCLASS field [4-72](#)
- ZJOBID-ZERO field [4-58](#)
- ZJOBNAME field [4-65](#)
- ZJOBNUMBER field
  - of ZBAT-MAP-DEF-EXECUTOR [4-51](#)
  - of ZBAT-MAP-STATUS-EXECUTOR [4-66](#)
- ZLAST-CC field [4-69](#)
- ZLOCALNAMES field [4-62](#)
- ZMASTER field [4-63](#)
- ZMAXCONCURRENTJOBS field [4-60](#)
  - See also ZMAXTEMPEXECUTORS field
- ZMAXPRI field [4-60](#)
- ZMAXPRINTLINES field
  - of ZBAT-MAP-DEF-JOB [4-56](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-61](#)
- ZMAXPRINTPAGES field
  - of ZBAT-MAP-DEF-JOB [4-56](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-61](#)
- ZMAXTEMPEXECUTORS field [4-60](#)
  - See also ZMAXCONCURRENTJOBS field
- ZMEM field [4-58](#)
- ZMICROSECOND field [4-57](#)
- ZMILLISECOND field [4-57](#)
- ZMINUTE field [4-57](#)
- ZMINUTES field [4-50](#)
- ZMONTH field [4-57](#)
- ZMONTHS field [4-50](#)
- ZNAME field [4-58](#)
- ZNAMELEN field [4-64](#)
- ZNAMETXT field [4-64](#)
- ZNEXT-RUNTIME field [4-68](#)
- ZOFF field
  - of ZEXECUTOR in ZBAT-MAP-STATUS-SCHEDULER [4-70](#)
  - of ZJOBCLASS in ZBAT-MAP-STATUS-SCHEDULER [4-72](#)
- ZON field
  - of ZEXECUTOR in ZBAT-MAP-STATUS-SCHEDULER [4-70](#)
  - of ZJOBCLASS in ZBAT-MAP-STATUS-SCHEDULER [4-72](#)
- ZOPEN-ACCESSOR field [4-67](#)
- ZOPEN-ACCESSOR-DETAIL field [4-67](#)
- ZOUT-SPOOL-NUM field [4-67](#)
- ZPFS field [4-58](#)
- ZPOSIX field [4-57](#)

- ZPRI field
  - of ZBAT-MAP-DEF-JOB [4-56](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-61](#)
- ZPROCESS field [4-72](#)
- ZPURGE-IN-FILE field [4-54](#)
- ZREADY field [4-71](#)
- ZRELEASER field [4-65](#)
- ZRE MID field
  - of ZBAT-MAP-DEF-JOB [4-55](#)
  - of ZBAT-MAP-STATUS-JOB [4-68](#)
- ZRESTART field [4-53](#)
- ZRUND field [4-58](#)
- ZRUNNEXT field [4-71](#)
- ZRUNNOW field [4-71](#)
- ZSAVEABEND field [4-58](#)
- ZSECOND field [4-57](#)
- ZSELPRI field
  - of ZBAT-MAP-DEF-JOB [4-56](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-61](#)
  - of ZBAT-MAP-STATUS-JOB [4-67](#)
- ZSPECIAL field [4-71](#)
- ZSPECIAL-REASON field [4-68](#)
- ZSPI- definitions [4-2/4-5](#)
- ZSPI-DDL-CHAR6 [4-33](#)
- ZSPI-TKN-
  - ADDR [4-2](#)
  - BUFLEN [4-2](#)
  - COMMENT [4-2](#)
  - definitions, tables of [4-2](#)
  - ENDLIST [A-1](#)
  - ERRLIST [A-1](#)
  - ERROR [4-24](#)
- ZSPI-TYP-
  - definitions, table of [4-3](#)
- ZSPI-VAL-
  - FALSE
    - STALL attribute value [4-54](#)
    - ZEMS-TKN-EMPHASIS
      - value [3-14](#), [4-7](#), [6-1](#)
  - TRUE
- ZSPI-VAL- (continued)
  - ZEMS-TKN-EMPHASIS
    - value [3-14](#), [4-7](#), [6-1](#)
- ZSTALL field
  - of ZBAT-MAP-DEF-JOB [4-54](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-62](#)
- ZSTATE field [4-70](#)
- ZSTOP field [4-71](#)
- ZSTOP-ON-ABEND field
  - of ZBAT-MAP-DEF-JOB [4-53](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-62](#)
- ZSUBMIT-ALLOWED field
  - of ZBAT-MAP-DEF-SCHEDULER [4-61](#)
  - of ZBAT-MAP-STATUS-SCHEDULER [4-73](#)
- ZSUSPENDED field
  - of ZJOB in ZBAT-MAP-STATUS-SCHEDULER [4-71](#)
  - of ZPROCESS in ZBAT-MAP-STATUS-SCHEDULER [4-72](#)
- ZTAPE field
  - of ZBAT-MAP-STATUS-SCHEDULER [4-72](#)
  - of ZJOB in ZBAT-MAP-STATUS-SCHEDULER [4-71](#)
- ZTAPEDRIVES field
  - of ZBAT-MAP-DEF-JOB [4-56](#)
  - of ZBAT-MAP-DEF-SCHEDULER [4-60](#)
- ZTAPEDRIVES-IN-USE field [4-72](#)
  - See also ZINFO-TAPEDRIVES-IN-USE field
- ZTIME field
  - of ZBAT-MAP-DEF-JOB [4-57](#)
  - of ZJOB in ZBAT-MAP-STATUS-SCHEDULER [4-71](#)
- ZTIMES-RUN field [4-69](#)
- ZTIME-CPUMAX field [4-69](#)
- ZTIME-CPUTOTAL field [4-69](#)
- ZTIME-ELAPSEDMAX field [4-69](#)
- ZTIME-ELAPSEDTOTAL field [4-69](#)
- ZTIME-FINISH field [4-68](#)

ZTIME-LIMIT field [4-59](#), [4-69](#)  
ZTIME-PREV-RUNTIME field [4-68](#)  
ZTIME-PUT-ON-LIST field [4-68](#)  
ZTIME-START field [4-68](#)  
ZTIME-SUBMIT field [4-69](#)  
ZTIME-USED field [4-68](#)  
ZUSER field [4-67](#)  
ZWEEKDAYS field [4-50](#)  
ZWHICH-LIST field  
    of ZBAT-MAP-STATUS-  
    EXECUTOR [4-66](#)  
    of ZBAT-MAP-STATUS-JOB [4-68](#)  
ZYEAR field [4-57](#)

## Special Characters

\$0, EMS collector [6-3](#)