Elastic Transaction Platform
Programmer’s Guide

REVISION: FEBRUARY 2015

The contents of this manual may be revised without prior notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the expressed written permission of Heirloom Computing Inc.

Heirloom Computing has made every effort to ensure that this manual is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time.
Preface

This Programmer’s Guide provides guidelines for the usage of Elastic Transaction Platform (ETP).

Elastic Transaction Platform is a robust enterprise sub-system permitting CICS based COBOL applications to migrate and execute within the Java Enterprise Edition (JEE) platform.

Elastic Transaction Platform integrates business applications into a cohesive, flexible, enterprise solution that allows existing business processes to use and exploit Java Enterprise based technologies. Elastic Transaction Platform reduces time, complexity, and risk involved in building and redeploing CICS COBOL business applications. The Elastic Transaction Platform preserves existing business assets, reduces the total cost of ownership and increases flexibility as offered through Enterprise Java Bean (EJB) transactions.

Elastic Transaction Platform provides traditional business sub-system services in the Java Enterprise Application Server environment. These sub-system services will function in conjunction with the popular application servers available on a wide variety of operating platforms. The application infrastructure permits enterprises to integrate business applications, and use an integrated environment built using industry standards.

Trademarks

- IBM is a registered trademark of International Business Machines Inc.
- Oracle and Java are registered trademarks of Oracle and/or its affiliates.
- UNIX is a registered trademark licensed exclusively to X/Open Company Limited.
- Linux is a registered trademark of Linus Torvalds.
- Windows is a registered trademark of Microsoft Corporation.
- Eclipse is a trademark of the Eclipse Foundation Inc.
- Other names may be trademarks of their respective owners.
## Contents

ELASTIC TRANSACTION PLATFORM USAGE ............................................................................. 5
  JOURNALS .................................................................................................................. 5
  JOURNAL PROTOCOLS ............................................................................................... 6
  TEMPLATES ................................................................................................................. 9
  QUEUES ....................................................................................................................... 10
  QUEUE PROTOCOLS .................................................................................................. 10
  SQL .............................................................................................................................. 12
  FILE DEFINITIONS .................................................................................................... 13

BMS FIELD VALIDATION EXTENSIONS ............................................................................ 16
  FMASK ......................................................................................................................... 16
  NMASK ......................................................................................................................... 18
  EVENT .......................................................................................................................... 19
  HTMLINPUT ................................................................................................................ 20

ETP COMMAND SYNTAX ................................................................................................... 21
  ABEND .......................................................................................................................... 22
  ADDRESS ..................................................................................................................... 23
  ADDRESS SET ............................................................................................................. 25
  ASKTIME ...................................................................................................................... 25
  ASSIGN ....................................................................................................................... 26
  BIF DEEDIT ................................................................................................................ 36
  CANCEL ........................................................................................................................ 38
  CHANGE PASSWORD ................................................................................................ 38
  CHANGE TASK .......................................................................................................... 39
  DEFINE COUNTER .................................................................................................... 40
  DELAY ......................................................................................................................... 42
  DELETE ......................................................................................................................... 44
  DELETE COUNTER ..................................................................................................... 45
  DELETEQ.TD .............................................................................................................. 47
  DELETEQ.TS .............................................................................................................. 47
  DOCUMENT CREATE ................................................................................................. 48
  DEQ ............................................................................................................................... 50
  DOCUMENT INSERT ................................................................................................. 51
  DOCUMENT RETRIEVE ............................................................................................... 54
  DOCUMENT SET ......................................................................................................... 55
  DUMP TRANSACTION ............................................................................................... 56
  ENDBR ......................................................................................................................... 58
  ENQ ............................................................................................................................... 59
  ENTER TRACENUM ..................................................................................................... 61
  EXTRACT CERTIFICATE ............................................................................................ 62
  EXTRACT TCPIP ......................................................................................................... 64
  EXTRACT WEB ........................................................................................................... 66
  FORMATTIME ............................................................................................................ 66
  FREEMAIN .................................................................................................................. 69
  GET COUNTER ........................................................................................................... 69
  GETMAIN .................................................................................................................... 72
  JOURNAL ...................................................................................................................... 73
| HANDLE ABEND                        | ................................................................. | 73 |
| HANDLE AID                         | ........................................................................ | 74 |
| HANDLE CONDITION                   | ........................................................................ | 75 |
| IGNORE CONDITION                   | ........................................................................ | 76 |
| LINK                                | ........................................................................ | 76 |
| LOAD                                | ........................................................................ | 78 |
| MONITOR                             | ........................................................................ | 80 |
| POP HANDLE                          | ........................................................................ | 80 |
| POST                                | ........................................................................ | 81 |
| PURGE MESSAGE                      | ........................................................................ | 83 |
| PUSH HANDLE                         | ........................................................................ | 83 |
| QUERY COUNTER                      | ........................................................................ | 84 |
| QUERY SECURITY                     | ........................................................................ | 85 |
| READ                                | ........................................................................ | 87 |
| READNEXT                            | ........................................................................ | 90 |
| READPREV                            | ........................................................................ | 92 |
| READQ.TD                            | ........................................................................ | 95 |
| READQ TS                            | ........................................................................ | 96 |
| RECEIVE MAP                         | ........................................................................ | 98 |
| RECEIVE PARTN                       | ........................................................................ | 99 |
| RELEASE                             | ........................................................................ | 100 |
| RESETBR                             | ........................................................................ | 100 |
| RETRIEVE                            | ........................................................................ | 102 |
| RETURN                              | ........................................................................ | 104 |
| REWIND COUNTER                     | ........................................................................ | 105 |
| REWRITE                             | ........................................................................ | 106 |
| ROUTE                               | ........................................................................ | 108 |
| SEND CONTROL                        | ........................................................................ | 110 |
| SEND MAP                            | ........................................................................ | 111 |
| SEND PAGE                           | ........................................................................ | 113 |
| SEND PARTNSET                       | ........................................................................ | 114 |
| SEND TEXT                           | ........................................................................ | 114 |
| SIGNOFF                             | ........................................................................ | 116 |
| SIGNON                              | ........................................................................ | 116 |
| SPOOLCLOSE                          | ........................................................................ | 118 |
| SPOOLOPEN INPUT                     | ........................................................................ | 119 |
| SPOOLOPEN OUTPUT                   | ........................................................................ | 120 |
| SPOOLREAD                           | ........................................................................ | 122 |
| SPOOLWRITE                          | ........................................................................ | 123 |
| START                               | ........................................................................ | 124 |
| SUSPEND                             | ........................................................................ | 128 |
| START ATTACH                        | ........................................................................ | 128 |
| START BREXIT                        | ........................................................................ | 129 |
| STARTBR                             | ........................................................................ | 130 |
| SUSPEND                             | ........................................................................ | 132 |
| SYNCPOINT                           | ........................................................................ | 133 |
| SYNCPOINT ROLLBACK                  | ........................................................................ | 133 |
| TRACE                               | ........................................................................ | 134 |
| UNLOCK                              | ........................................................................ | 135 |
| UPDATE COUNTER                     | ........................................................................ | 136 |
| UNLOCK                              | ........................................................................ | 138 |
| VERIFY PASSWORD                     | ........................................................................ | 139 |
| WAIT EXTERNAL                      | ........................................................................ | 140 |
Elastic Transaction Platform Usage

When compiling for the Elastic Transaction Platform (ETP) with Elastic COBOL, the compiler option `-out:transaction` must be included. Otherwise, EXEC TRANSACTION or EXEC CICS statements will be not be recognized.

The following resources must be configured so that ETP can be used:

- Journals
- Templates
- Queues
- SQL
- File Definitions

Journals

The journal protocol mechanism is used by several services, including Journal, Operator, and Trace.

A journal protocol provides a structured mechanism for writing to a sequential data store. It may optionally provide asynchronous writes and a reply mechanism. Note that not all journal protocols are available from all environments. For instance, while a Dialog journal may be appropriate for an operator message in an application, writing to the Dialog journal using Journal commands in an EJB would at best, pause services, and at worst, not function at all.

The protocol’s class implementation is defined using the setup value `protocol.journal.name=classname`. The default values for the protocols are available in the table below. A protocol that does not support reply will produce condition EXPIRED:100, while a protocol that does not support wait will ignore it.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Supports</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol.journal.file</td>
<td>N Y</td>
<td>com.heirloomcomputing.etp.transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>journal.BinaryFile</td>
</tr>
<tr>
<td>protocol.journal.textfile</td>
<td>N Y</td>
<td>com.heirloomcomputing.etp.transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>journal.TextFile</td>
</tr>
<tr>
<td>protocol.journal.weblogic</td>
<td>N N</td>
<td>com.heirloomcomputing.etp.transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>journal.WebLogic</td>
</tr>
<tr>
<td>protocol.journal.log4j</td>
<td>N N</td>
<td>com.heirloomcomputing.etp.transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>journal.Log4J</td>
</tr>
<tr>
<td>protocol.journal.sysout</td>
<td>N Y</td>
<td>com.heirloomcomputing.etp.transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>journal.Sysout</td>
</tr>
</tbody>
</table>
### Journal Protocols

All Journal Protocols are in lower-case in setup. Each is followed by a colon (:) and a referent. The referent is further information specific to the protocol.

In general, the Journal service uses the setup entry 'journal.journalname'. The Operator service uses the setup entry 'operator.#' where # is the routing number (2 by default). The Trace service uses the setup entry 'trace'.

**Example:**

```
journal.myjournal=sysout:
operator.2=myjournal
trace=myjournal
```

Protocols intended for logging destinations will interpret certain JTYPEID codes specially. The following JTYPEID codes are interpreted special wherever applicable (case insensitive):

<table>
<thead>
<tr>
<th>JTYPEID</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>*D</td>
<td>Debug</td>
</tr>
<tr>
<td>*I</td>
<td>Information</td>
</tr>
<tr>
<td>*W</td>
<td>Warning</td>
</tr>
<tr>
<td>*E</td>
<td>Error</td>
</tr>
<tr>
<td>*F</td>
<td>Fatal Error</td>
</tr>
<tr>
<td>*T</td>
<td>Trace</td>
</tr>
<tr>
<td>!C</td>
<td>Operator Critical Level</td>
</tr>
<tr>
<td>!I</td>
<td>Operator Immediate Level</td>
</tr>
<tr>
<td>!E</td>
<td>Operator Eventual Level</td>
</tr>
<tr>
<td>!R</td>
<td>Operator Reply Required</td>
</tr>
</tbody>
</table>
file:

The 'binary:' protocol is also available.

The file: protocol appends its entries to a binary file. The referent is the filename.

The format of the binary entry is:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of entry</td>
<td>4 bytes</td>
</tr>
<tr>
<td>User(0) / System(1)</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Task Number</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Transaction ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Terminal ID</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Timestamp (milliseconds)</td>
<td>8 bytes</td>
</tr>
<tr>
<td>JTYPEID</td>
<td>2 bytes</td>
</tr>
<tr>
<td>Length of user prefix</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Length of user data</td>
<td>4 bytes</td>
</tr>
<tr>
<td>User Prefix</td>
<td>Bytes given by length of user prefix</td>
</tr>
<tr>
<td>User Data</td>
<td>Bytes given by length of user data</td>
</tr>
</tbody>
</table>

Example setup (sends output for journal 'mybin' to the file 'testmyfile.bin'):

```
journal.mybin=file:testmyfile.bin
```

textfile:

The 'text:' protocol.

The textfile: protocol appends its entries to a textfile.

The format of the text entry is two lines:

```
[JTYPEID] user-prefix: date-time (EIBTASKN taskn) (EIBTRMID term-id)
(EIBTRNSID trans-id):
user-data
```

Example setup (sends output for journal 'mytxt' to the file 'testmyfile.txt'):

```
journal.mytxt=file:testmyfile.txt
```

weblogic:

The 'bea:' protocol.

This protocol is available only when operating under Oracle WebLogic.

The weblogic: protocol sends its entries to the WebLogic logging mechanism. In the WebLogic console, click the domain name in the navigator, then 'View domain log' on the right for a brief view of the logged messages. The full message file is available as well.

The format is determined by the current format of WebLogic output.

Example setup (sends output for journal 'myweb' to WebLogic catalog 'testmyfile.weblogic'):
journal.myweb=weblogic:testmyfile.weblogic

**log4j:**

The 'apache:' protocol.
The Log4J protocol sends its output to an Apache Log4J logger. The referent is the logger name.

Apache Log4J has its own setup mechanisms and is generally not included by default with the application server. Its jar file (such as log4j-1.2.7.jar) must be added to either the application server itself or to the application. As this is a general service, we recommend adding it to the application server itself if possible.

The Log4J package itself will search for the file log4j.properties.

In addition to this default action, the setup property 'log4.properties' may be set to a configuration file that will be read when the logger is opened. Only use this if Log4J will be included with the application rather than the application server.

**sysout:**

The 'stdout:' protocol.
The sysout protocol sends its output directly to the system output stream. This is often the console for an application server.

Example setup (sends output for journal 'mysysout' to sysout):

```
journal.mysysout=sysout:
```

**syserr:**

The 'stderr:' protocol.
The syserr protocol sends its output directly to the system error stream. This is often the console for an application server.

Example setup (sends output for journal 'mysyserr' to syserr):

```
journal.mysyserr=syserr:
```

**dialog:**

Dialog sends its messages to a dialog box. Dialog does support replies, however it is generally not allowed under an application server environment.

Example setup (sends output for 'mydialog' to a dialog box):

```
journal.mydialog=dialog:
```
queue:

The queue protocol redirects the journal entry as a message through to a queuing protocol. The referent is the symbolic queue name, which in turn must be defined as a queue name in the queuing protocol format.

Example setup (sends output for 'myqueue' to the queue queueimpl):

```plaintext
journal.myqueue=queue:queueimpl
```

Templates

Templates are HTML files where all the information is not necessarily present statically. They are stored as resources, and referenced by their name. Templates may be buffered in memory for faster access.

The template must be setup using the syntax `template.name=resource`, where `resource` is the actual resource filename.

Most text within a template is passed directly through to the destination. However, certain commands (related to SSI) are supported.

```plaintext
<--#include file="filename_OR_template_name"-->
<--#include virtual="filename_OR_template_name"-->
<--#include template="template_name"-->
<--#include resource="filename"-->
```

The include commands include another template file in the processing on the server. The text of the include command itself is removed from the destination stream, replaced by the contents of the resource.

The include template command looks up another template in the setup by `template_name`, referring to the resource indirectly.

The include resource command looks up the resource directly.

The file or virtual command looks up the resource as both template and directly as a resource, using whichever it finds; this is slightly slower than specifying the form directly.

```plaintext
<--#set var="name" value="value"-->
```

A template file has variables, the contents of which may be set and echoed. The DOCUMENT SET and SYMBOLLIST options also set variables.

The set command sets a variable to a value; it will maintain that value within the template until set again. It cannot, however, override the contents of a variable already set by the DOCUMENT SET command.

```plaintext
<--#echo var="name"-->
&name;
```

Echo the contents of the variable to the destination. The variable may have been set using DOCUMENT SET or SYMBOLLIST options or set using the set command.
The & form is more convenient, but be aware that it overlaps with existing entity references of HTML (such as &lt; for a less-than sign). An unrecognized & echo command will be passed through for further interpretation by later stages, such as a web browser, allowing &lt; to work as expected so long as no variable has the same name.

Queues

Queues hold data in sequential order, generally adding items to one end of the queue and retrieving them from the other end.

The Elastic Transaction Platform has a generalized queue mechanism, available for the Transient Data Queue, Temporary Storage Queue and Spool services. A queue is defined using a name and uri, where the uri includes the protocol specifying the implementation of the queue. For example, some queues are available only to the current facility's session and other ones may be available even outside of the Transaction Platform.

Temporary and Transient queues may be written to and read from immediately. Spools must be opened, used, and closed.

The most general form for setup of a Transient or Temporary queue is:

```
queue[.sysid_value].name=uri
[sysid.name=sysid_value]
```

where the uri is protocol:name, protocol specifying the implementation of the queue, and name being passed to that protocol for further interpretation.

A spool queue for output is defined as:

```
spool[.userid[.class[.node]]]=uri
```

and a spool queue for input as:

```
spool[.userid[.class]]=uri
```

where the queue may be as well specified as desired.

Queue Protocols

**session:destination**

The session queue protocol has its scope only within the existing session (conversation or pseudo-conversation), tied to a single terminal. It is never shared between simultaneous tasks and it is safe to use exclusively without the use of ENQ and DEQ.

This is the default queue protocol, if only the destination name is specified without a protocol. An instance of a session queue is returned by TQ commands using a queue not listed in the setup table.
**jms:destination**

The JMS queue uses Java Messaging Service for the queue. It uses any available JMS implementation, such as Oracle MessageQ or MQ-Series / WebSphere MQ. Its scope is defined by the JMS setup for the application server, and it can span tasks, machines, clusters and more. The destination is the JMS destination.

It uses additional table entries, each optional.

- destination.factory=jndi_factory_name
- destination.user=my_user_name
- destination.password=my_password

If no factory is specified, then the default factory is used, if specified:

```
jms.default.factory=jndi_factory_name
```

**jmstqt:destination**

The JMS temporary queue with task scope uses Java Messaging Service for the queue. It uses any available JMS implementation, such as Oracle MessageQ or MQ-Series / WebSphere MQ. The queue is automatically eliminated at the end of the task. The destination is the JMS destination.

It uses additional table entries, each optional.

- destination.factory=jndi_factory_name
- destination.user=my_user_name
- destination.password=my_password

If no factory is specified, then the default factory is used, if specified:

```
jms.default.factory=jndi_factory_name
```

**jmstqs:destination**

The JMS temporary queue with session scope uses Java Messaging Service for the queue. It uses any available JMS implementation, such as Oracle MessageQ or MQ-Series / WebSphere MQ. The queue must be deleted to eliminate its storage. The destination is the JMS destination.

It uses additional table entries, each optional.

- destination.factory=jndi_factory_name
- destination.user=my_user_name
- destination.password=my_password

If no factory is specified, then the default factory is used, if specified:

```
jms.default.factory=jndi_factory_name
```

**null:destination**

The null destination discards its messages.
SQL (Structured Query Language) access within the Elastic Transaction Platform must include setup information telling the platform where to find the database connection.

There are two different setup methods, depending on whether or not the SQL database is setup as a datasource in the application server. It is always preferable to setup the datasource if possible. If not supported by the particular application server, specify the SQL connection attributes on the SQL tab of the Elastic Transaction Platform Deploy Settings within an ETP Deployment Project.

Each SQL database connection has its own name. The default name is 'default' and is used if no SET CONNECTION statement is issued. The default connection is automatically established upon first EXEC SQL command by a transaction. You can connect to other databases and even multiple databases and switch the context between them. Use

```
EXEC SQL
  SET CONNECTION { :host-var | conn-name }
END-EXEC
```

Where `host-var` is a PIC X(16) variable defined in the transaction that holds a database connection name or the unquoted `conn-name` that is a database connection name.
The SQL tab of the ETP Deploy Settings file defines the database connection names used by transactions. The editor inserts into the settings property the file following values:

Setup:

```plaintext
sql.conn-name.datasource=datasource_name
sql.conn-name.user=user_name
sql.conn-name.password=user_password
```

Alternative Setup without datasource:

```plaintext
sql.conn-name.url=url_connection_string
sql.conn-name.driver=class_name
sql.conn-name.user=user_name
sql.conn-name.password=user_password
```

The user and/or password may be omitted if supported by the particular database. The user_name is the user_name for connection to the database. The password is only in server setup files and not visible to end clients.

One special SQL connection is of particular note, the 'file' connection. When ETP file I/O is mapped to a relational database (see Elastic COBOL Getting Started Guide) this connection name is used for the file control service commands (READ/READNEXT/READPREV/STARTBR/RESETBR/ENDBR/WRITE/REWRITE).

## File Definitions

Elastic Transaction Platform file control is done through SQL. That is, file access traditionally accessed in VSAM datasets is now done through JDBC drivers to SQL database tables.

VSAM data must be converted to SQL using any available VSAM to SQL converter. This may include custom programs written in CICS that read the entire dataset and then write it to SQL, as well as third-party utilities used to migrate data en masse.

The resulting SQL tables must include all the relevant information. FILLER fields need not be maintained if they do not contain data necessary to the containing group.

The mapping from the record structure to the SQL tables may vary significantly; an XML file must be available in the project’s resource folder. It must be the same name as the initial VSAM dataset and suffixed with .xml and the setup must include the following entry:

```plaintext
file.dataset_name=jdbc:xml_descriptor_file.xml
```

During execution, when the platform accesses the SQL table, it does so using the descriptions given in the XML file. The XML describes how to find
each data element within the record, including its offset, length, and storage format.

Transcribing dataset descriptive information may be tedious and Elastic COBOL also permits the option to place a $XFD command before the record to be written. The command is $XFD FILE="filename.xml", where the $ character must be in the indicator column. This command will generate correct information for the COBOL portion of the XML file and a good approximation for the SQL portion. Since the SQL portion can have any column names and storage, the file may need to be hand-edited to match the names used by SQL. After the .xml file is produced, the $XFD command is no longer needed and may be removed.

The $XFD command produces the xml descriptive file and places the result in the listing folder. This file must be verified to correct any discrepancies in the actual SQL usage, and the edited file should be moved into the "resources folder". This file is generated to a source folder to prevent accidentally overwriting a hand-edited version of the file.

The root of the XML file is the dataset name. The only entries that matter are the <column> entries. The <group> entries are included for information and for easy usage of group items in the table; the group may be converted to column and its individual columns removed. The column contains the name of the item in the database, the offset, length and type (from com.heirloomcomputing.ecs.api.Datatype) in the record, the sqltype for its storage, and the table name in which it is stored. The offset, length and type should not be changed.

The following sqltype values are recognized:

| A  | ALPHANUMERIC (CHAR) |
| AV | ALPHANUMERIC VARYING (VARCHAR) |
| B  | BINARY/IMAGE (BINARY)   |
| BV | BINARY/IMAGE VARYING (VARBINARY) |
| N  | NUMERIC INTEGER        |
| NZ | ZONED DECIMAL          |
| N1 | SINGLE PRECISION FLOATING POINT |
| N2 | DOUBLE PRECISION FLOATING POINT |

The column entry may also include an attribute named ridfld, which may be set to true in order to mark it as the primary key. It may also be set to duplicates in order to enable duplicate key checks if it is acting as an alternate key. Alternate keys may be set to noduplicates in order to suppress duplicate key checks. (The duplicate key check is computationally expensive for databases and should be skipped if the program logic does not require the DUPKEY condition.)

By default, the file control service uses a SQL connection named 'file' to perform its access.
Example setup:

file.alpha=jdbc:alpha.xml

Example data record for reading or writing:

"$xfd file="alpha.xml"
01 my-record.
  05 surname pic x(20).
  05 first pic x(20).
  05 mi pic x.
  05 ss pic 9(9).
  05 age pic 9(3).

Example XML dataset definition alpha.xml:

<!xml version="1.0" standalone="yes"?>
<dataset name="alpha">
  <!-- This file should be named after your dataset -->
  <group name="my_record" offset="0" length="53" usage="34" sqltype="B" table="alpha">
    <column name="surname" offset="0" length="20" type="32" sqltype="A" table="alpha" />
    <column name="first" offset="20" length="20" type="32" sqltype="A" table="alpha" />
    <column name="mi" offset="40" length="1" type="32" sqltype="A" table="alpha" />
    <column name="ss" offset="41" length="9" type="48" sqltype="NZ" table="alpha" />
    <column name="age" offset="50" length="3" type="48" sqltype="NZ" table="alpha" />
  </group>
</dataset>
BMS Field Validation Extensions

Elastic Transaction Platform provides additional BMS keywords that can be used to provide client-side field formatting and validation. This can help to reduce mainframe processing cycles.

FMASK

The FMASK (Fixed Mask) keyword is typically used for formatting input fields such as phone numbers, dates, SSN, credit-cards etc. It can also be used for formatting custom and/or general purpose field types.

General format

Format 1:

\[
\text{FMASK}=(\text{MASK})
\]

Format 2:

\[
\text{FMASK}=(\text{MASK,CUSTOM-MASK})
\]

Format 3:

\[
\text{FMASK}=(\text{MASK,CUSTOM-MASK,PLACEHOLDER})
\]

General rules

All formats:

1. MASK is a string that may contain mask types. The predefined mask types and their associated regular expressions are:
   - \(9\): [0-9] – a numeric character
   - \(D\): [0-9\.\] – a decimal character
   - \(X\): [A-Za-z0-9] – a strictly defined alphanumeric character
   - \(A\): [A-Za-z] – an alphabetic character
   - \(*\): [.] – any permissible character
   - \(B\): [ ] – a blank space character
   - \(S\): [+-] – a sign character
   - \($\): [$£€] – a common currency sign character
   - \(?\): – any characters following are optional

Examples:

1. \(\text{FMASK}=(\text{AAA-999})\)
   - A field containing 7 characters.
• The first 3 characters entered must be alphabetic.
• The 4th character will automatically be inserted as ‘-’.
• The last 3 characters entered must be numeric.

2. FMASK=((999) 999-9999? x99999)
   • A field representing a phone number containing 21 characters
     (the ‘?’ is a function marker, not a character).
   • All characters before ‘?’ are required.
   • All characters after ‘?’ are optional.

Format 2:
1. CUSTOM-MASK is a regular expression that provides a custom mask for
   any position in the MASK string that is marked with a ‘~’.

Example:
1. FMASK=(99~, [A-C])
   • A field containing 3 characters.
   • The first 2 characters entered must be numeric.
   • The last character ‘~’ identifies a custom mask is to be applied.
   • The CUSTOM-MASK is defined as the regular expression '[A-C]',
     which in this example means that the last character entered must
     be either ‘A’ or ‘B’ or ‘C’.

Format 3:
1. PLACEHOLDER is a string that can be used to indicate to the user the
   format of the input expected. The PLACEHOLDER is displayed in the
   input field but does not affect the actual content of the input field.

Example:
1. FMASK=(99-99-9999,,MM-DD-YYYY)
   • A field representing a date containing 10 characters (including the
     ‘-’ separator characters).
   • The field will accept only numeric input. The separator characters
     ‘-’ are inserted automatically.
   • The PLACEHOLDER is defined as the string ‘MM-DD-YYYY’.
   • NOTE: The 2nd parameter in this example is empty (this is the
     CUSTOM-MASK parameter and has not been set).
NMASK

The NMASK (Numeric Mask) keyword is typically used for formatting integer & decimal input fields such as general purpose numbers and currency. Currency number masks automatically include thousand-separators.

General format

Format 1:

NMASK=()

Format 2:

NMASK=(INT | DEC],[NUM | CUR])

Format 3:

NMASK=(INT | DEC],[NUM | CUR],MINVALUE,MAXVALUE)

General rules

Format 1:

1. This is equivalent to NMASK=(INT,NUM). See Format 2.

Format 2:

1. The 1st argument can be either INT (integer) or DEC (decimal).
2. The 2nd argument can be either NUM (number) or CUR (currency).

Examples:

1. NMASK=(INT,NUM)
   - All characters entered must be numeric.
   - Input field is defined as an integer number (e.g. 12345), with no decimal component and no thousand-separator.
2. NMASK=(DEC,CUR)
   - All characters entered must be numeric or a decimal point.
   - Input field is defined as decimal currency (e.g. 12,345.00).
   - The ‘,’ thousand-separator is automatically inserted.

Format 3:

1. MINVALUE & MAXVALUE can be used to establish formatted minimum & maximum values for the mask.

Example:

1. NMASK=(DEC,CUR,0.0000,999.9999)
• All characters entered must be numeric or a decimal point.
• Input field is defined as decimal currency and will accept any value between 0.0000 and 999.9999
• Format implies a number < 1000 with up to 4 decimals permitted.
• The ‘,’ thousand-separator is automatically inserted.

EVENT

The EVENT keyword is used to perform custom input field validation via JavaScript functions.

General format

Format 1:

EVENT=(myJavaScriptFunction,[FORM | FIELD],%V MSG,[arg1,arg2,...])

General rules

Format 1:

1. ‘myJavaScriptFunction’ is the name of the JavaScript function you want to call. Any JavaScript function can be called. By default, ETP provides 3 functions:
   • validateRange – tests if the field input is within a specified range.
   • validateDate – tests if field input is a valid date.
   • validateState – tests if field input is a valid US state.
2. The 2nd argument can be either FORM (if testing the submit data from the HTML form) or FIELD (if testing the data input to the HTML form).
3. The 3rd argument can be used to pass a custom text message to the function for processing. For example, the text message may be used to display a specific error message in a modal dialog.
4. The %V substitution string in the 3rd argument will be replaced by the contents of the input field.
5. Subsequent arguments (i.e. arg1, arg2, ...) are passed through “as is” to the JavaScript function.

Examples:

1. EVENT=(validateRange,FORM,%V IS NOT A VALID OPTION. PLEASE ENTER 1 THROUGH 5.,1,5)
   • Upon submission of the FORM, the content (%V) is tested to see if it is within the range of 1 through 5 (the range is specified by the last 2 parameters that are passed to the ‘validateRange’ function.
   • If the range is valid, nothing else will happen and processing will continue.
• If the range is invalid, a modal dialog is displayed with the message passed to the ‘validateRange’ function, and the cursor is set to the location of the input field so the input can be corrected.

2. EVENT=(validateDate,FIELD,%V IS NOT A VALID DATE. PLEASE RE-ENTER.,-)
   • Upon exit of the input FIELD, the contents (%V) are tested to see if they are a valid date.
   • If a separator is being used to format the date, then this is passed as the last argument to the ‘validateDate’ function.
   • If the date is valid, nothing else will happen and processing will continue.
   • If the date is invalid, a modal dialog is displayed with the message passed to the ‘validateDate’ function, and the cursor is set to the location of the input field so the input can be corrected.

3. EVENT=(validateState,FIELD,%V IS NOT A VALID STATE. PLEASE RE-ENTER.)
   • Upon exit of the input FIELD, the contents (%V) are tested to see if they are a valid US state.
   • If the state is valid, nothing else will happen and processing will continue.
   • If the state is invalid, a modal dialog is displayed with the message passed to the 'validateState' function, and the cursor is set to the location of the input field so the input can be corrected.

**HTMLINPUT**

The HTMLINPUT keyword is used to inject a string into the input field’s HTML INPUT tag. This can be useful for overriding existing attributes or adding others.

**General format**

**Format 1:**

```
HTMLINPUT=STRING
```

**General rules**

**Format 1:**

1. STRING must be a valid HTML INPUT attribute.

**Example:**

1. HTMLINPUT=style="text-align: center"
   • This will override any other settings for this attribute.
The Elastic Transaction Platform service commands supported include all IBM CICS API commands:

<table>
<thead>
<tr>
<th>Service</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in Function</td>
<td>BIF DEEDIT</td>
</tr>
<tr>
<td>Basic Mapping Support</td>
<td>RECEIVE MAP, SEND MAP, SEND TEXT, SEND PAGE, PURGE MESSAGE, ROUTE, RECEIVE PARTN, SEND PARTNSET, SEND CONTROL</td>
</tr>
<tr>
<td>Counter</td>
<td>DEFINE, GET, UPDATE, DELETE, REWIND, QUERY COUNTER and DCOUNTER</td>
</tr>
<tr>
<td>Date/Time</td>
<td>ASKTIME ABTIME, FORMATTIME</td>
</tr>
<tr>
<td>Document</td>
<td>DOCUMENT CREATE, INSERT, RETRIEVE, SET</td>
</tr>
<tr>
<td>Dump Control</td>
<td>DUMP TRANSACTION</td>
</tr>
<tr>
<td>Execute Interface</td>
<td>ADDRESS, HANDLE AID, HANDLE CONDITION, ASSIGN, IGNORE CONDITION, PUSH HANDLE, POP HANDLE, ADDRESS SET</td>
</tr>
<tr>
<td>File Control targeting</td>
<td>READ, WRITE, REWRITE, DELETE, UNLOCK, STARTBR, READNEXT, READPREV, ENDBR, RESETBR</td>
</tr>
<tr>
<td>Interval Control</td>
<td>DELAY, POST, START, START ATTACH, START BREXIT, RETRIEVE, CANCEL, WAIT EVENT</td>
</tr>
<tr>
<td>Journal</td>
<td>JOURNAL, WRITE JOURNALNUM, WAIT JOURNALNUM, WRITE JOURNALNAME, WAIT JOURNAL</td>
</tr>
<tr>
<td>Operator</td>
<td>WRITE OPERATOR</td>
</tr>
<tr>
<td>Program Control</td>
<td>LINK, XCTL, LOAD, RETURN, RELEASE, ABEND, HANDLE ABEND</td>
</tr>
<tr>
<td>Recovery</td>
<td>SYNCPOINT, SYNCPOINT ROLLBACK</td>
</tr>
<tr>
<td>Security</td>
<td>SIGNON, SIGNOFF, VERIFY PASSWORD, QUERY SECURITY, CHANGE PASSWORD</td>
</tr>
<tr>
<td>Spool</td>
<td>SPOOLOPEN, SPOOLREAD, SPOOLWRITE, SPOOLCLOSE</td>
</tr>
<tr>
<td>Storage Control</td>
<td>GETMAIN, FREEMAIN</td>
</tr>
<tr>
<td>Task Control</td>
<td>SUSPEND, ENQ, DEQ</td>
</tr>
<tr>
<td>Task Management</td>
<td>CHANGE TASK</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>EXTRACT TCPIP, excluding EXTRACT CERTIFICATE</td>
</tr>
<tr>
<td>Temporary Storage</td>
<td>WRITEQ TS, READQ TS, DELETEQ TS</td>
</tr>
<tr>
<td>Trace Control</td>
<td>ENTER TRACENUM, ENTER, TRACE, excluding MONITOR</td>
</tr>
<tr>
<td>Transient Data Control</td>
<td>WRITEQ TD, READQ TD, DELETEQ TD</td>
</tr>
<tr>
<td>Web</td>
<td>WEB RECEIVE, WEB SEND, WEB READ HTTPHEADER, WEB STARTBROWSE HTTPHEADER/FORMFIELD/COOKIE, WEB READ/READNEXT HTTPHEADER/FORMFIELD/COOKIE, WEB ENDBROWSE HTTPHEADER/FORMFIELD/COOKIE, WEB WRITE, WEB EXTRACT, WEB RETRIEVE</td>
</tr>
</tbody>
</table>

Restrictions

Application Programming Interface (API) commands are supported except for:
- EXTRACT CERTIFICATE (TCP/IP)
- MONITOR (Trace)
- Batch Data Interchange service is not supported
- Distributed Processing service is not supported (APPC, direct terminal control)
- Sub-system capabilities not supported by the ETP sub-system are the System Programming Interface (SPI) commands and conversational transactions.

**ABEND**

Program controlled user abnormal termination.

**Syntax:**

```
ABEND
    [ABCODE(data-value)]
    [CANCEL]
    [NODUMP]
```

**Service:**

Program Control

**Description:**

An abnormal termination, or abend, propagates from the creation point upwards through the stack until either a programmatic abend handler is found or until the included abend handler is found. The included abend handler creates a program dump so long as NODUMP is not specified.

**Options:**

- **ABCODE(data-value)**
  Data-value is the four-character abnormal termination code, visible in the final abend handling dump and available to the abend handler if present.

- **CANCEL**
  Ignore any handlers specified by HANDLE ABEND commands.

- **NODUMP**
  Request no dump be made at final abend handling.
ADDRESS

Establish a pointer to the address of a system area.

Syntax:

ADDRESS
   [ACEE(pointer-ref)]
   [COMMAREA(pointer-ref)]
   [CSA(pointer-ref)]
   [CWA(pointer-ref)]
   [EIB(pointer-ref)]
   [TCTUA(pointer-ref)]
   [TWA(pointer-ref)]

Service:

Execute Interface Program

Setup:

   cwa.jndi=cwa_jndi_name
   The name of the CWA in JNDI is 'cwa' by default. This allows an alternate
   JNDI storage name.
   cwa.value
   This provides an initial value for the CWA if none is currently present. If the
   CWA is already established, this is ignored.
   cwa.size
   This provides the size of the CWA in bytes. This defaults to 256.
   twa.size
   This provides the size of the TWA in bytes. This defaults to 256.
   eib.size
   This provides the size of the EIB in bytes. This defaults to 256.
   tctua.size
   This provides the size of the TCTUA in bytes. This defaults to 256.

Options:

   ACEE(pointer-ref)
Establishes linkage to the Access Control Environment Element, information pertinent to the security manager. This may be null.

**COMMAREA(pointer-ref)**

Establishes linkage to the commarea. As the COMMAREA is already passed to the program in the PROCEDURE DIVISION USING or initial linkage, this is useful primarily for access from other called programs where the commarea was not passed. This may be null.

**CSA(pointer-ref)**

As the Common System Area (CSA) has been made obsolete by the ASSIGN command, an abend code of ASRD is thrown if access to the CSA is attempted. Always use ASSIGN for access to information originally available via the CSA.

**CWA(pointer-ref)**

Establishes linkage to the Common Work Area (CWA), user storage common to all tasks within application server. It is stored as a JNDI element named 'cwa'. As a direct pointer may not be shared in an application server environment, a copy is taken when first referenced and the value of the memory is restored to JNDI upon the task's end, if modified. As even with direct pointers there would be race conditions, generally do not modify the CWA on other than an occasional basis (such as once per day for a custom date format). This may be null.

**EIB(pointer-ref)**

Establishes linkage to the Execute Interface Block (EIB). As the EIB is already passed to the program in the PROCEDURE DIVISION USING or initial linkage, this is useful primarily for access from other called programs where the EIB was not passed.

Be aware that the method used for handling the EIB means that a variable by the correct name must be present in the program in order to be set by the EXEC command. When called directly from Java, the EIB value must be passed as a parameter; the EIB block itself is not used.

**TCTUA(pointer-ref)**

Establishes linkage to the Terminal Control Table User Area, storage unique to the virtual terminal session. This has little use in command-level programming.

**TWA(pointer-ref)**

Establishes linkage to the Transaction Work Area (TWA), storage unique to the current task. This has little use in command-level programming.

**Conditions:**

None
ADDRESS SET

Set the address of a pointer.

This command is generally not used in new code, as the COBOL SET verb may be used directly.

Syntax:

ADDRESS

[SET(data-area) USING(pointer-ref)]

[SET(pointer-ref) USING(data-area)]

Service:

Execute Interface Program

Setup:

None

Options:

SET(data-area)
Specifies that the data-area’s pointer is modified to be pointing to the using parameter.

SET(pointer-ref)
Specifies that the pointer is modified to be pointing to the using parameter.

USING(pointer-ref)
 Specifies the pointer source to be used by the SET option.

USING(data-area)
Specifies the data-area source’s pointer to be used by the SET option.

Conditions:

None

ASKTIME

Retrieve the current timestamp. The FORMATTIME command may be used to transform the returned timestamp to human readable formats.
ASKTIME also sets EIBDATE and EIBTIME, if present in the calling program.

**Syntax:**

```cobol
ASKTIME
   [ABSTIME(data-area)]
   [MILLISECONDS]
```

**Service:**

Date Time

**Setup:**

None

**Options:**

- **ABSTIME**
  Retrieves current date/time in milliseconds since January 1, 1900 at 00:00.

- **MILLISECONDS**
  Do not artificially decrease accuracy to hundredths of a second for compatibility.

**Conditions:**

None

---

**ASSIGN**

Assign the contents of a transaction platform value to a given data-area. This is used for retrieving internal information safely. All options are for returning data.

**Syntax:**

```cobol
ASSIGN
   [ABCODE(data-area)] [ABDUMP(data-area)]
   [ABPROGRAM(data-area)] [ACTIVITY(data-area)]
   [ACTIVITYID(data-area)] [ALTSCRNHT(data-area)]
   [ALTSCRNWD(data-area)] [APLKYBD(data-area)]
```
Service:

Execute Interface Program

Setup:

None

Options:

**ABCODE(data-area)**
4-character current abend code, blanks if no abend.

**ABDUMP(data-area)**
X"FF" if dump has been produced, X"00" otherwise.

**ABPROGRAM(data-area)**
8-character program name of abending program.

**ACTIVITY(data-area)**
BTS only, until Business Transaction Services supported throws condition INVREQ.

**ACTIVITYID(data-area)**
BTS only, until Business Transaction Services supported throws condition INVREQ.

**ALTSCRNHT(data-area)**
Alternate screen height for terminal.
ALTSCRNWD(data-area)
Alternate screen width for terminal.

APLKYBD(data-area)
X"FF" if APL keyboard feature, X"00" otherwise.

APLTEXT(data-area)
X"FF" if terminal keyboard has APL text feature, X"00" otherwise.

APPLID(data-area)
8-character applid of system owning transaction.

ASRAINTRPT(data-area)
Not supported. If supported, would return 8-character data-area containing ILC and PIC of last abend of type ASRA, ASRB, ASRD or AICA.

ASRAKEY(cvda)
Returns the execution key of the previous ASRA-style interrupt. This could return any of the following cvda values:

CICSEXECKEY
USEREXECKEY
NONCICS
NOTAPPLIC
The current release always returns NOTAPPLIC.

ASRAPSW(data-area)
8-character Program Status Word of last ASRA-style interrupt. The current release always returns binary zeroes.

ASRAREG(data-area)
64-byte data area holding general registers 0…15 of last AREA-style interrupt. The current release always returns binary zeroes, as this register set is mainframe specific.

ASRASPC(cvda)
Returns the space in effect at the time of the previous ASRA-style interrupt. This could return any of the following cvda values:

SUBSPACE
BASESPACE
NOTAPPLIC

The current release always returns NOTAPPLIC.

ASRASTG(cvda)
>Returns the storage addressed at the time of the previous ASRA-style interrupt. This could return any of the following cvda values:

- CICS
- USER
- READONLY
- NOTAPPLIC

The current release always returns NOTAPPLIC.

**BRIDGE(data-area)**

4-character TRANSID of bridge monitor transaction that issued the START BREXIT TRANSID. command. Blanks are returned if this is not applicable.

**BTRANS(data-area)**

X"FF" if terminal support background transparency, X"00" otherwise.

**CMDSEC(data-area)**

X"FF" if command security defined for task, X"00" otherwise.

**COLOR(data-area)**

X"FF" if terminal supports extended color capability, X"00" otherwise.

**CWALEN(data-area)**

Length of the Common Work Area (CWA).

**DEFSCRNHT(data-area)**

Default screen height for terminal.

**DEFSCRNWD(data-area)**

Default screen width for terminal.

**DELIMITER(data-area)**

1-byte data-link control character for 3600. This will be one of:

- X"80" end-of-text
- X"40" end-of-block
- X"20" inter-record separator
- X"10" start of header
- X"08" transparent input

**DESTCOUNT(data-area)**

Following BMS ROUTE, number of different terminal types in route list. Within BMS overflow, relative overflow control number.

**DESTID(data-area)**

BDI only, until Batch Data Interchange is supported throws the condition INVREQ. 8-character outboard destination.
DESTIDLENG(data-area)
BDI only, until Batch Data Interchange is supported throws the condition
INVREQ. Length of destination identifier in DESTID.

DSSCS(data-area)
X"FF" if facility is basic SCS data stream device, X"00" otherwise.

DS3270(data-area)
X"FF" if 3270 data stream device, X"00" otherwise.

EWASUPP(data-area)
X"FF" if Erase Write Alternative supported, X"00" otherwise.

EXTDS(data-area)
X"FF" if terminal supports 3270 extended data stream, X"00" otherwise.

FACILITY(data-area)
4-byte principal facility identifier.

FCI(data-area)
1-byte Facility Control Indicator.

GCHARS(data-area)
Graphic character set global identifier (GCSGID), 1…65,534.

GCODES(data-area)
Code page global identifier (CPGID), 1…65,534.

GMMI(data-area)
X"FF" if good morning message for terminal, X"00" otherwise.

HILIGHT(data-area)
X"FF" if terminal supports extended highlight capability, X"00" otherwise.

INITPARM(data-area)
60-character initialization parameter. If none, binary zeroes.
The setup for the initparm is:
   program.<program_name>.initparm=<initparm_value>

INITPARMLEN(data-area)
Length of INITPARM. If no INITPARM, then zero.

INPARTN(data-area)
1- or 2-character name of most recent input partition.

INVOKINGPROG(data-area)
8-character name of program that used LINK or XCTL to transfer control to
current program. Blanks if at highest level.

KATAKANA(data-area)
X"FF" if facility supports katakana, X"00" otherwise.

**LANGINUSE(data-area)**
3-byte mnemonic code for language in use, corresponds to NATLANGINUSE.

**LDCMNEM(data-area)**
2-byte LDC (Logical Device Code) mnemonic of destination that overflowed.

**LDCNUM(data-area)**
1-byte LDC numeric value of destination that overflowed.

**MAPCOLUMN(data-area)**
Origin column of most recently positioned map.

**MAPHEIGHT(data-area)**
Height of most recently positioned map.

**MAPLINE(data-area)**
Origin line of most recently positioned map.

**MAPWIDTH(data-area)**
Width of most recently positioned map.

**MSRCONTROL(data-area)**
X"FF" if terminal supports MSR (Magnetic Slot Reader), X"00" otherwise.

**NATLANGINUSE(data-area)**
1-byte mnemonic code for language in use, corresponds to LANGINUSE.

**NETNAME(data-area)**
8-character name of logical unit in VTAM network, if applicable.

**NEXTTRANSID(data-area)**
4-character name of next transaction, if available.

**NUMTAB(data-area)**
1-byte number of tabs required to position print element for 2980.

**OPCLASS(data-area)**
24-bits, BMS operator class for routing terminal messages.

**OPERKEYS(data-area)**
Supported for compatibility only, 8 null bytes.

**OPID(data-area)**
3-character operator ID.

**OPSECURITY(data-area)**
3-bytes of zeroes, this field is not used in current systems.
ORGABCODE(data-area)
4-byte original abend code.

OUTLINE(data-area)
X"FF" if terminal supports field outlining, X"00" otherwise.

PAGENUM(data-area)
Page number for overflow.

PARTNPAGE(data-area)
2-byte name of partition having caused page overflow.

PARTNS(data-area)
X"FF" if terminal supports partitions, X"00" otherwise.

PARTNSET(data-area)
1…6 character application partition set.

PRINSYSID(data-area)
4-character name by which the other system is known locally.

PROCESSS(data-area)
BTS only, until Business Transaction Services supported throws condition INVREQ.

PROCESSTYPE(data-area)
BTS only, until Business Transaction Services supported throws condition INVREQ.

PROGRAM(data-area)
8-character program name of the current program.

PS(data-area)
X"FF" if terminal has programmed symbols capability, X"00" otherwise.

QNAME(data-area)
4-character queue name that initiated the task, if task initiated by ATI.

RESSEC(data-area)
"X" if resource security checking in effect, blank otherwise.

RESTART(data-area)
X"FF" if restart of task has occurred, X"00" otherwise.

RETURNPROG(data-area)
8-character program name of the program to which the current program will return.

SCRNHT(data-area)
Height of 3270 screen.
SCRNWD(data-area)
Width of 3270 screen.

SIGDATA(data-area)
4-byte inbound signal.

SOSI(data-area)
X"FF" is mixed DBCS is supported, X"00" otherwise.

SPJAVAVERSION(data-area)
System Property, the Java version.

SPJAVAVERSION(data-area)
System Property, the Java vendor.

SPJAVAVERSIONURL(data-area)
System Property, the Java vendor URL.

SPJAVAHOME(data-area)
System Property, the Java home.

SPJAVACLASSVERSION(data-area)
System Property, the Java class version.

SPJAVACLASSPATH(data-area)
System Property, the Java classpath.

SPOSNAME(data-area)
System Property, the Operating System name.

SPOSARCH(data-area)
System Property, the Operating System architecture.

SPOSVERSION(data-area)
System Property, the Operating System version.

SPFILESEPARATOR(data-area)
System Property, the file separator character, as in '/' or '\'.

SPPATHSEPARATOR(data-area)
System Property, the path separator character, as in ':' or ';'.

SPLINESEPARATOR(data-area)
System Property, the line separator, as in 13 10.

STARTCODE(data-area)
2-byte indicator specifying how task was started.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>DPL without SYNCONRETURN.</td>
</tr>
<tr>
<td>DS</td>
<td>DPL with SYNCONRETURN.</td>
</tr>
</tbody>
</table>
### Elastixic COBOL Programmer’s Guide

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QD</td>
<td>Transient data queue.</td>
</tr>
<tr>
<td>S</td>
<td>START without data.</td>
</tr>
<tr>
<td>SD</td>
<td>START with data.</td>
</tr>
<tr>
<td>SZ</td>
<td>FEPI START.</td>
</tr>
<tr>
<td>TD</td>
<td>Terminal.</td>
</tr>
<tr>
<td>U</td>
<td>User-attached.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTIONID(data-area)</td>
<td>1-byte station ID for 2980 terminal.</td>
</tr>
<tr>
<td>SYSID(data-area)</td>
<td>4 character system ID.</td>
</tr>
<tr>
<td>TASKPRIORITY(data-area)</td>
<td>Task priority, 0…255.</td>
</tr>
<tr>
<td>TCTUALENG(data-area)</td>
<td>Length of Terminal Control Table User Area (TCTUA).</td>
</tr>
<tr>
<td>TELLERID(data-area)</td>
<td>1-byte teller ID for 2980 terminal.</td>
</tr>
<tr>
<td>TERMCODE(data-area)</td>
<td>2-byte terminal code for terminal associated to task. The first byte is terminal type, the second byte is model number.</td>
</tr>
<tr>
<td>TERMPRIORITY(data-area)</td>
<td>Terminal priority, 0…255. If separate priorities are not supported, 0 is returned.</td>
</tr>
<tr>
<td>TEXTKYBD(data-area)</td>
<td>X“FF” is terminal supports TEXTKYBD, X“00” otherwise.</td>
</tr>
<tr>
<td>TEXTPRINT(data-area)</td>
<td>X“FF” if terminal supports TEXTPRINT, X“00” otherwise.</td>
</tr>
<tr>
<td>TRANPRIORITY(data-area)</td>
<td>Transaction priority, 0…255. If separate priorities are not supported, 0 is returned.</td>
</tr>
<tr>
<td>TWALENG(data-area)</td>
<td>Length of Transaction Work Area (TWA).</td>
</tr>
<tr>
<td>UNATTEND(data-area)</td>
<td>X“FF” is terminal is in unattended mode, X“00” otherwise.</td>
</tr>
<tr>
<td>USERID(data-area)</td>
<td></td>
</tr>
</tbody>
</table>

---

Elastic COBOL Programmer's Guide 35
ID of the user. Traditionally up to 8 characters, but it may be more depending upon user id's allocated by the system. The assign will truncate if necessary.

**USERNAME(data-area)**
Name of the user. Traditionally up to 20 characters, but it may be more depending upon user names allocated by system. The assign will truncate if necessary.

**USERPRIORITY(data-area)**
Operator priority, 0…255. If separate priorities are not supported, 0 is returned.

**VALIDATION(data-area)**
X"FF" is terminal supports validation, X"00" otherwise.

**VERSIONMAJOR(data-area)**
Returns the major release version of the transaction platform.

**VERSIONMINOR(data-area)**
Returns the minor release version of the transaction platform.

**VERSIONSERVICE(data-area)**
Returns the service release version of the transaction platform.

**VERSION(data-area)**
Returns a text representation of the version number, as major.minor.service.

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No signed in user present.</td>
</tr>
<tr>
<td>2</td>
<td>No BMS command in use.</td>
</tr>
<tr>
<td>3</td>
<td>No BDI, Batch Data Interchange in use.</td>
</tr>
<tr>
<td>4</td>
<td>The task was not started using ATI, Automatic Transaction Initiation.</td>
</tr>
<tr>
<td>5</td>
<td>The task has no facility.</td>
</tr>
<tr>
<td>6</td>
<td>Not operating under BTS, Business Transaction Services.</td>
</tr>
<tr>
<td>200</td>
<td>Not valid in DPL, Distributed Program Link.</td>
</tr>
</tbody>
</table>

**NOTAUT H** The requested assignment has not been authorized, generally by the application server itself.

**BIF DEEDIT**

Format alphanumeric data into numeric data. This function is obsolete.
It retains all numeric characters (digits 0 through 9), and right-aligns them with zero-fill on the left. It does not preserve the decimal point or signed, zoned digits.

**Syntax:**

BIF DEEDIT

**Service:**

Built In Function

**Setup:**

None

**Options:**

FIELD
The field is the data that is the alphanumeric source and numeric destination.

LENGTH
The length is a binary number describing the length of the field. If not given, the length of the FIELD variable is used.

**Conditions:**

None

**Example:**

```cobol
01 bif-1 pic x(17).
move "ab1cd2ef3gh4ij5kl" to bif-1
exec transaction
  web send text("BIF FROM: " bif-1)
end-exec
exec transaction
  bif deedit field(bif-1)
end-exec
exec transaction
  web send text("BIF TO: " bif-1)
end-exec
```

**Output:**

**BIF FROM:** ab1cd2ef3gh4ij5kl
**BIF TO:** 00000000000012345
CANCEL

Cancel a previous interval control request. The request must be still waiting to be performed in order to be successfully cancelled.

Syntax:

CANCEL

[REQID(data-value)]
[SYSID(data-value)]
[TRANSID(data-value)]

Service:

Interval Control

Setup:

None

Options:

REQID(data-value)
Specifies request ID (1 to 8 characters) used with START, START ATTACH, POST or DELAY command.

SYSID(data-value)
Specifies the system ID of the request to cancel, if used.

TRANSID(data-value)
Specifies the transaction ID of the request to cancel, if used.

Conditions:

NOTFND  The REQID was not found.

CHANGE PASSWORD

This command is dependent upon the capabilities of the External Security Manager.
Syntax:

CHANGE PASSWORD(data-value)
NEWPASSWORD(data-value)
USERID(data-value)
[ESMREASON(data-area)]
[ESMRESP(data-area)]

Service:

Security

Setup:

None

Options:

ESMREASON(data-area)
Retrieves the external security manager's reason code.

ESMRESP(data-area)
Retrieves the external security manager's response code.

NEWPASSWORD(data-value)
Specifies the new password, up to 8 characters.

PASSWORD(data-value)
Specifies the current password, up to 8 characters.

USERID(data-value)
Specifies the user id, up to 8 characters.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>USERIDERR</td>
<td>The user id is unknown</td>
</tr>
</tbody>
</table>

CHANGE TASK

Change a task's operating priority, yielding the CPU in the process.
Syntax:

CHANGE TASK
[PRIORITY(data-value)]

Service:

Task Management

Setup:

None

Options:

PRIORITY
-1 through 255.
-1 is a no operation, not yielding the CPU.

0 through 255 are priorities. These priorities are scaled to the JVM's levels of priority, which are determined by Thread.MIN_PRIORITY through Thread.MAX_PRIORITY, which is 1 through 10 at the time of this writing.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The PRIORITY is not within -1 through 255.</td>
</tr>
<tr>
<td>100</td>
<td>The PRIORITY option is not supported within the current environment.</td>
</tr>
</tbody>
</table>

DEFINE COUNTER

Define a counter in the counter service. A counter must be defined before its first use. All counters are referenced by name.

Counters are segregated by pools, with each pool having an independent version of any named counters. The pool is selected by the program code.

Syntax:

DEFINE { COUNTER(name) | DCOUNTER(name) } [POOL(name)]
[VALUE(data-value)]
[MINIMUM(data-value)]
[MAXIMUM(data-value)]
Service:

Counter

Setup:

pool.name=redirected_pool_name
If not specified, the pool name is the given name.

counter.jndi=counter_jndi_name
Specify the JNDI name of the external Counter service. The default is 'CounterService'.
The counter dispatches most of its work off to an external service so that the counters will be shared between tasks.

Options:

COUNTER(name)
Specifies the name of the 32-bit signed counter.

DCOUNTER(name)
Specifies the name of the 64-bit unsigned counter.

POOL(name)
Specifies the name of the pool of counters. Each Counter service maintains its own pools of counters.

VALUE(data-value)
Sets the initial value of the counter.

MINIMUM(data-value)
Specifies the minimum value for the counter.

MAXIMUM(data-value)
Specifies the maximum value for the counter.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>202</td>
<td>Counter name already exists, cannot create duplicate counter.</td>
</tr>
<tr>
<td>301</td>
<td>Server error.</td>
</tr>
<tr>
<td>302</td>
<td>No space in the pool to create the counter.</td>
</tr>
<tr>
<td>303</td>
<td>Unexpected error, such as connectivity loss.</td>
</tr>
<tr>
<td>304</td>
<td>Invalid pool.</td>
</tr>
<tr>
<td>305</td>
<td>Cannot connect to server.</td>
</tr>
<tr>
<td>306</td>
<td>Server abend.</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
</tr>
<tr>
<td>308</td>
<td>This condition cannot occur (options table not loadable).</td>
</tr>
<tr>
<td>309</td>
<td>This condition cannot occur (options table error).</td>
</tr>
<tr>
<td>310</td>
<td>This condition cannot occur (options user exit).</td>
</tr>
<tr>
<td>403</td>
<td>The pool name contains invalid characters.</td>
</tr>
<tr>
<td>404</td>
<td>The counter name contains invalid characters.</td>
</tr>
<tr>
<td>406</td>
<td>The increment is invalid; it cannot be larger than the total range of the counter.</td>
</tr>
<tr>
<td>407</td>
<td>The minimum or maximum value is invalid; negative, or maximum less than minimum</td>
</tr>
</tbody>
</table>

**DELAY**

Delay the task for a period of time in a CPU-efficient manner.

As delay loops may be optimized away, and the CPU should be free to perform other activities during processing, a delay may be introduced using this command.

INTERVAL or FOR specify a period of time, whereas TIME or UNTIL specify the final time itself.

Note that most foreground sessions contain timeouts and will be terminated if delayed too far.

**Syntax:**

```
DELAY
[REQID(data-value)]
  [INTERVAL(hhmmss)]
[FOR
  [HOURS(data-value)]
  [MINUTES(data-value)]
  [SECONDS(data-value)]
]
[TIME(hhmmss)]
[UNTIL
  [HOURS(data-value)]
  [MINUTES(data-value)]
  [SECONDS(data-value)]
]
```
**Service:**

Interval Control

**Setup:**

None

**Options:**

**REQID(data-value)**

Specifies a request ID through which this command may be cancelled using the CANCEL command. The request ID is valid only within the same session.

**INTERVAL(hhmmss)**

Specifies an interval of time duration data-value. The format is a single number with two-digits each hour, minute, second.

**TIME(hhmmss)**

Specifies a final time data-value. The format is a single number with two-digits each hour, minute, second.

**FOR**

FOR specifies that HOURS, MINUTES and SECONDS refers to an interval duration.

**UNTIL**

UNTIL specifies that HOURS, MINUTES and SECONDS refers to a final time.

**HOURS(data-value)**

 Specifies the number of hours, and it must be within 0..99.

**MINUTES(data-value)**

Specifies the number of minutes, and it must be within 0..59 if other HOURS or SECONDS is specified.

**SECONDS(data-value)**

Specifies the number of seconds, and it must be within 0..59 if other HOURS or MINUTES is specified.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPIRED</td>
<td>The interval has already expired.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
</tbody>
</table>
DELETE

Delete a record from a given file, interpreted as deleting a row from a database.

Syntax:

DELETE FILE(data-area)
    [TOKEN(data-area) | RIDFLD(data-area) [KEYLENGTH(data-value)]
    [GENERIC [NUMREC(data-area)]]]
    [SYSID(data-area)]
    [NOSUSPEND]
    [RBA | RRN]

Service:

File Control

Setup:

file.<filename>=jdbc:< xml_file_descriptor.xml>[;<sql_connection_name>]

The xml_file_descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XFD FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is ‘file’ by default.

sql.<sql_connection_name>.datasource=<jndi_datasource_name>

The file service uses the SQL connection named ‘file’ by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.
Options:

**FILE(data-area)**
Specifies the name of the dataset, which must be included in setup as a reference to jdbc:filename.xml.

**TOKEN(data-area)**
Specifies token used by READ UPDATE in order to associate this delete with a prior read.

**RIDFLD(data-area)**
Specifies the record ID field, the key field.

**KEYLENGTH(data-value)**
Specifies the length of the key.

**GENERIC**
The key is generic, specified to the length of keylength.

**NUMREC(data-area)**
Returns the number of deleted records.

**SYSID(data-area)**
Specifies the system ID.

**NOSUSPEND**
Specifies that no waiting is to occur if records are locked.

**RBA**
Use RBA as the key field name.

**RRN**
Use RRN as the key field name.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILENOTFOUND</td>
<td>The file was not found.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>An invalid request was issued.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output file error occurred during the operation</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The operation could not find the record.</td>
</tr>
</tbody>
</table>

**DELETE COUNTER**

Delete the named counter.
Syntax:

DELETE { COUNTER(name) | D COUNTER(name) } [POOL(name)]

Service:

Counter

Setup:

pool.name=redirected_pool_name
If not specified, the pool name is the given name.

counter.jndi=counter_jndi_name
Specify the JNDI name of the Counter service. The default is 'CounterService'.

Options:

COUNTER(name)
Specifies the name of the 32-bit counter.

DCOUNTER(name)
Specifies the name of the 64-bit counter.

POOL(name)
Specifies the name of the pool of counters. Each Counter service maintains its own pools of counters.

Conditions:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
<tr>
<td>201</td>
<td>Named counter invalid.</td>
</tr>
<tr>
<td>301</td>
<td>Server error.</td>
</tr>
<tr>
<td>303</td>
<td>Unexpected error, such as connectivity loss.</td>
</tr>
<tr>
<td>304</td>
<td>Invalid pool.</td>
</tr>
<tr>
<td>305</td>
<td>Cannot connect to server.</td>
</tr>
<tr>
<td>306</td>
<td>Server abend.</td>
</tr>
<tr>
<td>308</td>
<td>This condition cannot occur (options table not loadable).</td>
</tr>
<tr>
<td>309</td>
<td>This condition cannot occur (options table error).</td>
</tr>
<tr>
<td>310</td>
<td>This condition cannot occur (options user exit).</td>
</tr>
<tr>
<td>403</td>
<td>The pool name contains invalid characters.</td>
</tr>
</tbody>
</table>
DELETEQ TD

Delete all data within a transient data queue.

Syntax:

DELETEQ TD QUEUE(name)

[SYSID(name)]

Service:

Transient Data Control

Setup:

queue[.sysid_value].name=uri

[sysid.name=sysid_value]

Options

QUEUE(name)

Specifies the name of the transient data queue. The queue must be defined.

SYSID(name)

Specifies the system ID. This is used to determine which queue definition is used.

Conditions

| SYSIDERR | An unknown sysid is being used. |
| QIDERR   | 1. There is no such queue definition. |
|          | 2. The queue could not be obtained |
| INVREQ   | The queue could not be deleted. For instance, JMS queues (being externally available) cannot be deleted. |

DELETEQ TS

Delete all data within a temporary storage queue.

Syntax:

DELETEQ TS {QUEUE(name) | QNAME(name)}
Service:
Temporary Storage Control

Setup:

queue[.sysid_value].name[.main | .auxiliary]=uri
[sysid.name=sysid_value]

Options

QUEUE(name)
Specifies the name of the transient data queue. The queue must be defined.

QNAME(name)
See QUEUE.

SYSID(name)
Specifies the system ID. This is used to determine which queue definition is used.

Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIDERR</td>
<td>An unknown sysid is being used</td>
</tr>
<tr>
<td>QIDERR</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

DOCUMENT CREATE

Create a new document.
The document is stored in an internal format, manipulated with DOCUMENT commands, and either sent to the web using WEB SEND or retrieved in final form using DOCUMENT RETRIEVE.

Syntax:

DOCUMENT CREATE DOCTOKEN(data-area)
{{ TEXT(data-area) | BINARY(data-area) | FROM(data-area)) [LENGTH(data-value)]} |}

TEMPLATE(name)
FROMDOC(data-area)
[DOCSIZE(data-area)]
[HOSTCODEPAGE(name)]
[SYMBOLLIST(data-area) [LISTLENGTH(data-value)]]

Service:

Document

Setup:

    template.name=resource

Options:

    DOCTOKEN(data-area)
    Retrieves a 16-byte document token. The document token must then be passed to other DOCUMENT commands to refer to the same document.

    TEXT(data-area)
    Specifies text to be inserted.

    BINARY(data-area)
    Specifies binary data to be inserted.

    FROM(data-area)
    Specifies a binary image of a previously retrieved document or template.

    LENGTH(data-value)
    Specifies the length of the TEXT, BINARY or FROM data.

    TEMPLATE(name)
    Specifies a template. The template must be defined.

    FROMDOC(data-area)
    Specifies that contents of another document are to be inserted. The data-area is the 16-byte retrieved DOCTOKEN.

    DOCSIZE(data-area)
    Retrieves the estimated byte size of the document.

    HOSTCODEPAGE(name)
    Specifies the host's code page.

    SYMBOLLIST(data-area)
    Specifies a list of symbols and their values in URL-encoded form. Each list agreement is a name, an equals sign (=), and a value. Each list element is separated from each other list element by an ampersand (&).
The value must not contain a plain ampersand, as it is used for element separation, and it must not contain a plain percentage character (%), as it is used for embedding values. The % sign must be followed by two hexadecimal characters, which form a single character representing the given ASCII value. So, %26 may be used to embed an ampersand (&), and %25 may be used to embed a percentage character (%). A plus sign represents a space, so a plus sign must be embedded as %2B. All other characters are themselves.

Example:

SYMBOLLIST('first=George&last=Washington&fullname=George+Washington').

LISTLENGTH(data-value)

Specifies the length of the SYMBOLLIST.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>The document being received FROM is in an invalid format.</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The FROMDOC is not found.</td>
</tr>
<tr>
<td></td>
<td>The TEMPLATE is not found.</td>
</tr>
<tr>
<td></td>
<td>The HOSTCODEPAGE is not found.</td>
</tr>
<tr>
<td>SYMBOLERR</td>
<td>The symbol name is invalid.</td>
</tr>
<tr>
<td>TEMPLATERR</td>
<td>The template is invalid.</td>
</tr>
</tbody>
</table>

DEQ

Release a resource from exclusive use that had been previously allocated by ENQ.

This command is cooperative, so any other tasks wanting the same resource must also allocate it for exclusive use using the same mechanism.

The default implementation uses JNDI. A resource is locked to a single task by binding an object to the JNDI name 'lts_resource_<resourcename>', where <resourcename> is the name of the resource to its given length. The resource is unlocked by unbinding the object. (Nesting the enq/deq is left to the individual task when locked between tasks.) The scope of the resource allocation is therefore the scope of JNDI in use, either a single application server or clustered application servers using clustered JNDI.

Syntax:

DEQ RESOURCE(data-area) [LENGTH(data-value)]
[UOW | TASK | MAXLIFETIME(cvda)]

Service:

Task Control

Options:

RESOURCE(data-value)
Specifies the name of the resource as a data-area. This is always the content of the specified resource, not the address of the resource. LENGTH should always be specified, but the implicit length of the resource is used if LENGTH is left unspecified.

LENGTH(data-value)
Specifies the length of the resource name. If left unspecified, an implicit length of the resource data-area is used, not the address of the resource. This must be between 1 and 255. It must be same as when allocated by ENQ, or the resource will not have the same resource name as intended.

UOW
Specifies that the resource is to live until the end of the unit of work. This is the default.

TASK
Specifies that the resource is to live until the end of the task.

MAXLIFETIME(cvda)
Specifies the maximum lifetime of the resource, either the CVDA for UOW or TASK. The CVDA LUW is accepted as a synonym for UOW. Specify the lifetime in this way if not constant, otherwise use the UOW or TASK tag directly.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>MAXLIFETIME was set to an invalid CVDA.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LENGERR</th>
<th>LENGTH was not between 1 and 255.</th>
</tr>
</thead>
</table>

DOCUMENT INSERT

Insert document elements into a document.
Syntax:

DOCUMENT INSERT DOCTOKEN(data-area)

{{ TEXT(data-area) | BINARY(data-area) | FROM(data-area)} [LENGTH(data-value)]} |

SYMBOL(name)
TEMPLATE(name)
FROMDOC(data-area)
BOOKMARK(name)
[DOCSIZE(data-area)]
[HOSTCODEPAGE(name)]
[AT(name)]
[TO(name)]

Service:

Document

Setup:

template.name=resource

Options:

DOCTOKEN(data-area)
Specifies the document into which the element is inserted; this is obtained originally from a DOCUMENT CREATE.

TEXT(data-area)
Specifies text to be inserted.

BINARY(data-area)
Specifies binary data to be inserted.

FROM(data-area)
Specifies a binary image of a previously retrieved document or template.

LENGTH(data-value)
Specifies the length of the TEXT, BINARY or FROM data.

SYMBOL(name)
Specifies a symbol. The symbol's current content value is inserted, not the symbol itself.

TEMPLATE(name)
Specifies a template. The template must be defined.

**FROMDOC(data-area)**

Specifies that contents of another document are to be inserted. The data-area is the 16-byte retrieved DOCTOKEN.

**BOOKMARK(name)**

Specifies that a bookmark is to be inserted. The bookmark may be used to note the current location for later inserts.

**DOCSIZE(data-area)**

Retrieves the estimated byte size of the document.

**HOSTCODEPAGE(name)**

Specifies the host's code page.

**AT(name)**

If TO is not specified, then the insertion occurs immediately after the bookmark name.

If TO is specified, then the contents between AT and TO bookmarks are removed, being replaced by this insertion element.

**TO(name)**

If AT is not specified, then the insertion occurs immediately before the bookmark name.

If AT is specified, then the contents between AT and TO bookmarks are removed, being replaced by this insertion element.

**Conditions:**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DUPREC</strong></td>
<td>A duplicate bookmark is being defined.</td>
</tr>
<tr>
<td><strong>INVREQ</strong></td>
<td><strong>RESP2</strong></td>
</tr>
<tr>
<td>0</td>
<td>The bookmark TO is incorrectly before the bookmark AT.</td>
</tr>
<tr>
<td>1</td>
<td>The document being received FROM is in an invalid format.</td>
</tr>
<tr>
<td>2</td>
<td>The bookmark is invalid.</td>
</tr>
<tr>
<td><strong>NOTFND</strong></td>
<td><strong>RESP2</strong></td>
</tr>
<tr>
<td>1</td>
<td>The DOCUMENT is not found</td>
</tr>
<tr>
<td>2</td>
<td>The FROMDOC is not found.</td>
</tr>
<tr>
<td>3</td>
<td>The TEMPLATE is not found</td>
</tr>
<tr>
<td>4</td>
<td>The SYMBOL is not found.</td>
</tr>
<tr>
<td>5</td>
<td>The AT bookmark is not found.</td>
</tr>
<tr>
<td>6</td>
<td>The TO bookmark is not found.</td>
</tr>
<tr>
<td>7</td>
<td>The HOSTCODEPAGE is not found.</td>
</tr>
</tbody>
</table>
DOCUMENT RETRIEVE

Retrieve a copy of a document.

Syntax:

```
DOCUMENT RETRIEVE DOCTOKEN(data-area)
  INTO(data-area)
  LENGTH(data-area)
  MAXLENGTH(data-value)
  CLNTCODEPAGE(name)
  DATAONLY
```

Service:

Document

Setup:

None

Options:

```
DOCTOKEN(data-area)
  Specifies the document token, the document to retrieve.
INTO(data-area)
  Returns the data itself.
LENGTH(data-area)
  Returns the length of the data.
MAXLENGTH(data-value)
  Specifies the maximum length of data expected.
CLNTCODEPAGE(name)
  Specifies the client code page.
DATAONLY
  Return only data, not meta-data such as internal bookmarks.
```
Conditions:

<table>
<thead>
<tr>
<th>LENGERR</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MAXLENGTH is less than or equal to zero.</td>
</tr>
<tr>
<td>2</td>
<td>The document is truncated because the buffer is too small.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NOTFND</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The document is not found.</td>
</tr>
<tr>
<td>7</td>
<td>The client codepage is invalid.</td>
</tr>
</tbody>
</table>

**DOCUMENT SET**

Add or set symbols within the specified document.

**Syntax:**

```cobol
DOCUMENT SET
  DOCTOKEN(data-area)
  {
    { SYMBOL(name) [ NAMELENGTH(data-value) ] VALUE(data-area) } |
    { SYMBOOLLIST(data-area) } |
    }
  LENGTH(data-value)
```

**Service:**

Document

**Options:**

**DOCTOKEN(data-area)**

Specifies a document token, created using earlier document commands.

**SYMBOL(name)**

Specifies a symbol name. The name is whitespace trimmed when used.

**NAMELENGTH(data-value)**

Specifies the length of the symbol name. (The default is the whitespace trimmed length.)

**SYMBOOLLIST(data-area)**

Specifies a list of symbols. The list is of the form name=value, separated by semicolons (;). The values are restricted, such that a percent sign followed
by two hexadecimal digits is an ASCII character, ampersand is not allowed (replaced by %26), and pluses (+) represent spaces (replaced by %2B).

**LENGTH(data-value)**
Specifies the length of the symbol value or of the symbol list.

**VALUE(data-area)**
Specifies the value of the symbol.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Response Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTFND</td>
<td>RESP2</td>
<td>The document does not exist.</td>
</tr>
<tr>
<td>SYMBOLERR</td>
<td>RESP2</td>
<td>Neither SYMBOL nor SYMBOLLIST was specified, or if specified were invalid.</td>
</tr>
</tbody>
</table>

**DUMP TRANSACTION**

Dump information about the transaction programmatically. A dump occurs automatically upon an abend.

The dump is distributed among the various active services, so which options are actually recognized depends upon the implementation of the other services.

**Syntax:**

```cobol
DUMP TRANSACTION DUMPCODE(name)
    [COMPLETE]
    [FROM(data-area) [LENGTH(data-value) | FLENGTH(data-value)]
    [TRT]
    [SEGMENTLIST(data-area) LENGTHLIST(data-area) NUMSEGMENTS(data-area)]
    [TASK]
    [STORAGE]
    [PROGRAM]
    [TERMINAL]
    [TABLES]
    [DCT]
    [FCT]
    [PCT]
```
Service:

Dump Control

Setup:

None

Options:

DUMPCODE(name)
Specify a dumpcode signature for the dump.

COMPLETE
Include all information in the dump.

FROM(data-area)
Specify a data-area to be dumped along with the remainder of the dump.

LENGTH(data-value)
The length of the FROM.

FLENGTH(data-value)
The length of the FROM.

SEGMENTLIST(data-area)
LENGTHLIST(data-area)
NUMSEGMENTS(data-area)
Segment dumping is not currently supported.
This specifies that there are NUMSEGMENTS segments, each listed in
SEGMENTLIST with a length specified by LENGTHLIST.

TASK
Storage associated with task.

STORAGE
Dump storage.

PROGRAM
Program storage areas.

TERMINAL
Storage associated with terminal.

**TABLES - DCT, FCT, PCT, PPT, SIT, TCT.**

**DCT**
Destination Control Table.

**FCT**
File Control Table.

**PCT**
Process Control Table.

**PPT**
Processing Program Table.

**SIT**
System Initialization Table.

**TCT**
Terminal Control Table.

---

**ENDBR**

End the browsing operation on a file. The browse must previously have been started using STARTBR.

**Syntax:**

```cobol
ENDBR
{FILE(data-value) | DATASET(data-value)}
[REQID(data-value)]
[SYSID(data-value)]
```

**Service:**

File Control

**Setup:**

See file setup under READ.

**Options**

```cobol
FILE(data-value)
```
Specifies the filename. The filename must be setup to point to an .xml file descriptor.

**DATASET(data-value)**
Specifies the filename, just as does FILE. FILE is preferred.

**REQID(data-value)**
Specifies a previously specified request identifier to control multiple browses.

**SYSID(data-value)**
Specifies the system ID, 1 to 4 characters, of the browse operation.

### Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILENOTFOUND</td>
<td>The file was not found.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>Logical error.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>Input/Output error.</td>
</tr>
<tr>
<td>ISCINVREQ</td>
<td>Remote system error.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Resource security check failed.</td>
</tr>
<tr>
<td>SYSIDERR</td>
<td>System ID error.</td>
</tr>
</tbody>
</table>

### ENQ

Acquire a resource for exclusive use.

This command is cooperative, so any other tasks wanting the same resource must also allocate it for exclusive use using the same mechanism.

As this command by definition forces tasks to wait or serialize their actions, it should be avoided wherever possible.

The default implementation uses JNDI. A resource is locked to a single task by binding an object to the JNDI name 'lts_resource_<resourcename>', where <resourcename> is the name of the resource to its given length. The resource is unlocked by unbinding the object. (Nesting the enq/deq is left to the individual task when locked between tasks.) The scope of the resource allocation is therefore the scope of JNDI in use, either a single application server or clustered application servers using clustered JNDI.

### Syntax:

```
ENQ RESOURCE(data-area) [LENGTH(data-value)]
   [UOW | TASK | MAXLIFETIME(cvda)]
   [NOSUSPEND]
   [RETRIES(data-value)]
   [RETRYINTERVAL(data-value)]
```
Service:

Task Control

Options:

**RESOURCE(data-value)**

Specifies the name of the resource as a data-area. This is always the content of the specified resource, not the address of the resource. LENGTH should always be specified, but the implicit length of the resource is used if LENGTH is left unspecified.

**LENGTH(data-value)**

Specifies the length of the resource name. If left unspecified, an implicit length of the resource data-area is used, not the address of the resource. This must be between 1 and 255.

**UOW**

Specifies that the resource is to live until the end of the unit of work. This is the default.

**TASK**

Specifies that the resource is to live until the end of the task.

**MAXLIFETIME(cvda)**

Specifies the maximum lifetime of the resource, either the CVDA for UOW or TASK. The CVDA LUW is accepted as a synonym for UOW. Specify the lifetime in this way if not constant, otherwise use the UOW or TASK tag directly.

**NOSUSPEND**

Do not suspend the task waiting for the resource; issue ENQBUSY instead.

**RETRIES(data-value)**

Specifies a maximum number of retries to obtain the resource if the implementation may use it; it is ignored otherwise.

**RETRYINTERVAL(data-value)**

Specifies an interval between retries to obtain the resource if the implementation may use it; it is ignored otherwise. The interval is in microseconds, so a RETRYINTERVAL(1000) would delay for one second before retrying.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENQBUSY</td>
<td>The resource is busy and cannot be enqueued. This is only issued if not instructed to wait for the resource indefinitely. Unlike most conditions, it is ignored by default, so always code a RESP or HANDLE to catch</td>
</tr>
</tbody>
</table>
ENTER TRACENUM

Append a record to the trace journal.

Syntax:

ENTER TRACENUM(data-value)
  FROM (data-area) [FROMLENGTH(data-area)]
  [RESOURCE(name)]
  [EXCEPTION]

Service:

Trace Control

Options:

TRACENUM
Trace identifier, a number within 0 through 199.

FROM
Contents of the trace entry.

FROMLENGTH
Length of the trace entry contents, within 0 through 4000.

RESOURCE
Resource field of trace entry.

EXCEPTION
Write entry even if user trace flag is off. The text '*EXCU' is included in the trace entry.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>TRACENUM is not within 0 through 199.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
EXTRACT CERTIFICATE

(THIS COMMAND IS UNSUPPORTED)

Extract information about the client X.509 certificate received during SSL handshake, if present.

Syntax:

EXTRACT CERTIFICATE(ptr-ref)
[LENGTH(data-area)]
[SERIALNUM(ptr-ref)]
[SERIALNUMLEN(data-area)]
[USERID(ptr-ref)]
[OWNER | ISSUER]
[COMMONNAME(ptr-ref)]
[COMMONNAMLEN(data-area)]
[COUNTRY(ptr-ref)]
[STATE(ptr-ref)]
[STATELEN(data-area)]
[LOCALITY(ptr-ref)]
[LOCALITYLEN(data-area)]
[ORGANIZATION(ptr-ref)]
[ORGANIZATLEN(data-area)]
[ORGUNIT(ptr-ref)]
[ORGUNITLEN(data-area)]

Service:

TCP/IP

Setup:

None
Options:

CERTIFICATE
Retrieve the full certificate binary.

LENGTH
Retrieve the length of the full certificate binary.

SERIALNUM
Retrieve the pointer to the serial number.

SERIALNUMLEN
Retrieve the length of the serial number.

USERID
Retrieve the user identification.

OWNER
The extracted values of this command relate to the owner of the certificate. This is the default, if ISSUER is not specified.

ISSUER
The extracted values of this command relate to the issuer of the certificate.

COMMONNAME
Retrieve the common name from the client certificate.

COMMONNAMELEN
Retrieve the length of the common name retrieved.

COUNTRY
Retrieve the pointer to the country.

STATE
Retrieve the pointer to the state or province.

STATELEN
Retrieve the length of the state or province returned.

LOCALITY
Retrieve the locality.

LOCALITYLEN
Retrieve the length of the locality.

ORGANIZATION
Retrieve the organization.

ORGANIZATIONLEN
Retrieve the length of the organization.
ORGUNIT
Retrieve the organizational unit.

ORGUNITLEN
Retrieve the length of the organizational unit.

Conditions:

| INVREQ | Occurs for the following reasons:
| | The common is not in a web session,
| | Not in an HTTP request,
| | There is no certificate,
| | Or there is an error in retrieving the certificate. |

| LENGERR | The extracted contents exceed the corresponding length. |

EXTRACT TCPIP

Extract information about the current TCP/IP connection.
The current implementation is available only when executing as a web session.

Syntax:

```
EXTRACT TCPIP
   [CLIENTNAME(data-area) CNAMELENGTH(data-area)]
   [SERVERNAME(data-area) SNAMELENGTH(data-area)]
   [CLIENTADDR(data-area) CADDRLENGTH(data-area)]
   [CLIENTADDRNU(data-area)]
   [SERVERADDR(data-area) SADDRLENGTH(data-area)]
   [SERVERADDRNU(data-area)]
   [SSLTYPE(cvda)]
   [TCPIPSERVICE(data-area)]
   [PORTNUMBER(data-area)]
   [PORTNUMNU(data-area)]
```

Service:

TCP/IP
Setup:

None

Options:

CLIENTNAME
Extract the client host name, address in text form if unknown.

CNAMELENGTH
Pass the length of the CLIENTNAME buffer, receive the length of contents.

SERVERNAME
Extract the server host name, address in text form if unknown.

SNAMELENGTH
Pass the length of the SERVERNAME buffer, receive the length of contents.

CLIENTADDR
Receive the client address in text form.

CADDRLENGTH
Pass the length of the CLIENTADDR buffer, receive the length of contents.

CLIENTADDRNU
Receive the client address in four-byte form, suitable for PIC X(2) COMP-X.

SERVERADDR
Receive the server address in text form.

SADDRLENGTH
Pass the length of the SERVERADDR buffer, receive the length of contents.

SERVERADDRNU
Receive the server address in four-byte form, suitable for PIC X(2) COMP-X.

SSLTYPE
The CVDA value of SSL, NOSSL, or CLIENTAUTH, checkable against DFHVALUE(SSL), DFHVALUE(NOSSL) or DFHVALUE(CLIENTAUTH) in program code.

NOSSL is returned when connected without SSL, as in the http protocol.
SSL is returned when connected using SSL, as in the https protocol.
CLIENTAUTH is returned when connected using SSL and the client has client authorization, a client certificate.

TCPIPSERVICE
Extract the TCP/IP service name, such as 'HTTP/1.1'.

PORTNUMBER
Extract the port number of the TCP/IP service in text form.

**PORTNUMNU**

Extract the port number of the TCP/IP service in numeric form.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>The session is not a web session.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length extracted is longer than the length allowed by the corresponding LENGTH parameter.</td>
</tr>
</tbody>
</table>

**EXTRACT WEB**

See WEB EXTRACT.

**FORMATTIME**

Format a given absolute date/time timestamp into human readable formats.

**Syntax:**

```
FORMATTIME [ABSTIME(data-area)]
    [DATE(data-area)]
    [FULLDATE(data-area)]
    [DATEFORM(data-area)]
    [DATESEP | DATESEP(data-value)]
    [DAYCOUNT(data-area)]
    [DAYOFMONTH(data-area)]
    [DAYOFWEEK(data-area)]
    [DDMMYY(data-area)]
    [DDMMYYYY(data-area)]
    [MMDDYY(data-area)]
    [MMDDYYYY(data-area)]
    [MONTHOFYEAR(data-area)]
    [TIME(data-area) [TIMESEP | TIMESEP(data-value)]]
    [YEAR(data-area)]
    [YYDDD(data-area)]
    [YYDDMM(data-area)]
    [YYMMDD(data-area)]
```

Elastic COBOL Programmer's Guide  66
Service:

Date Time

Setup:

init.dateform

This should be set to one of the following values:
YYMMDD (default)
DDMMYY
MMDDYY

Options:

In all options where date formatting is returned, M is month digit, D is date digit, Y is year digit.

ABSTIME(data-area)

Data-area is a numeric, specifying the number of milliseconds since January 1, 1900 at 00:00, as returned by ASKTIME. This is the value used as the basis for other formatting options. If not specified, the current date/time timestamp is used.

DATE(data-area)

Retrieve date in form specified by init.datform, where the date separator is specified by DATESEP (blank otherwise).

FULLDATE(data-area)

DATEFORM(data-area)

DATESEP(data-value)

If data-value is specified, it is the date separator; if data-value is not specified, '/' is used.

Data-value must be one character in length.

DAYCOUNT(data-area)

Retrieve number of days since January 1, 1900.

DAYOFMONTH(data-area)

Retrieve number of day in the month.

DAYOFWEEK(data-area)
Retrieve number of day in the week.

**DDMMYY(data-area)**
Retrieve the date in DDMMYY format.

**DDMMYYYY(data-area)**
Retrieve the date in DDMMYYYY format.

**MMDDYY(data-area)**
Retrieve the date in MMDDYY format.

**MMDDYYYY(data-area)**
Retrieve the date in MMDDYYYY format.

**MONTHOFYEAR(data-area)**
Retrieve the month of the year.

**TIME(data-area)**
Retrieve the time, where the separator in hh:mm:ss is specified by TIMESEP (blank otherwise).

**TIMESEP(data-value)**
If data-value is specified, it is the time separator; if data-value is not specified, ':' is used.
Data-value must be one character in length.

**YEAR(data-area)**
Retrieve the year.

**YYDDD(data-area)**
Retrieve the date in YYDDD format.

**YYDDMM(data-area)**
Retrieve the date in YYDDMM format.

**YYMMDD(data-area)**
Retrieve the date in YYMMDD format.

**YYYYDDD(data-area)**
Retrieve the date in YYYYDDD format.

**YYYYDDMM(data-area)**
Retrieve the date YYYYDDMM format.

**YYYYMMDD(data-area)**
Retrieve the date in YYYYMMDD format.

**Conditions:**
None
FREEMAIN

Free memory allocated by GETMAIN.
In this environment, memory is garbage collected automatically, so FREEMAIN does nothing. If additional pointers point to the memory, it will remain allocated. Additionally, if no pointers are pointing to the memory it is removed automatically.

Syntax:

FREEMAIN
{DATA(data-area) | DATAPINTER(ptr-value)}

Setup:
None

Service:
Storage Control

Options:

DATA(data-area)
Specifies the main storage to be released.

DATAPINTER(ptr-value)
Specifies the main storage to be released.

Conditions:
None

GET COUNTER

Get the next value from the counter service.

Syntax:

GET { COUNTER(name) | DCOUNTER(name) }
{POOL(name)}
VALUE(data-area)
[INCREMENT(data-value) [REDUCE]]

[WRAP]

[COMPAREMIN(data-value)]

[COMPAREMAX(data-value)]

Service:

Counter

Setup:

pool.name=redirected_pool_name

If not specified, the pool name is the given name.

counter.jndi=counter_jndi_name

Specify the JNDI name of the Counter service. The default is 'CounterService'.

Options:

COUNTER(name)

Specifies the name of the 32-bit counter.

DCOUNTER(name)

Specifies the name of the 64-bit counter.

POOL(name)

Specifies the name of the pool of counters. Each Counter service maintains its own pools of counters.

VALUE(data-area)

Retrieves the value of the counter.

INCREMENT(data-value)

Specifies an alternate increment value, rather than one (1).

REDUCE

Specifies that the increment is to be reduced if the increment would force the counter to be at its limit. The result would then be the maximum value plus one (1).

WRAP

Automatically performs a wrap rather than notify that the counter is its limit.

COMPAREMIN(data-value)
Specifies a minimum value against which to compare the value. If the value is greater than or equal, then no action is taken; if less than the minimum, a condition is raised.

If COMPAREMIN>COMPAREMAX, then either condition must be true rather than both.

Success of the command is conditional upon the comparison.

**COMPAREMAX(data-value)**

Specifies a maximum value against which to compare the value. If the value is less than or equal, then no action is taken; if less than the minimum, a condition is raised.

If COMPAREMIN>COMPAREMAX, then either condition must be true rather than both.

Success of the command is conditional upon the comparison.

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Named counter invalid.</td>
</tr>
<tr>
<td>301</td>
<td>Server error.</td>
</tr>
<tr>
<td>303</td>
<td>Unexpected error, such as connectivity loss.</td>
</tr>
<tr>
<td>304</td>
<td>Invalid pool.</td>
</tr>
<tr>
<td>305</td>
<td>Cannot connect to server.</td>
</tr>
<tr>
<td>306</td>
<td>Server abend.</td>
</tr>
<tr>
<td>308</td>
<td>This condition cannot occur (options table not loadable).</td>
</tr>
<tr>
<td>309</td>
<td>This condition cannot occur (options table error).</td>
</tr>
<tr>
<td>310</td>
<td>This condition cannot occur (options user exit).</td>
</tr>
<tr>
<td>403</td>
<td>The pool name contains invalid characters.</td>
</tr>
<tr>
<td>404</td>
<td>The counter name contains invalid characters.</td>
</tr>
<tr>
<td>406</td>
<td>The increment is invalid; it cannot be larger than the total range of the counter.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LENGERR</th>
<th>LENGERR occurs only for COUNTER commands, not DCOUNTER commands.</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>The value has become negative.</td>
</tr>
<tr>
<td>002</td>
<td>The value is too large by one bit.</td>
</tr>
<tr>
<td>003</td>
<td>The value is too large by more than one bit.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUPPRESSED</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>The maximum value has already been assigned. It must be reset or retrieved using WRAP.</td>
</tr>
<tr>
<td>103</td>
<td>The value is not within COMPAREMIN and COMPAREMAX, or beyond the limits if only one of COMPAREMIN and COMPAREMAX is used</td>
</tr>
</tbody>
</table>
GETMAIN

Allocate main storage, returning a pointer to it. FREEMAIN is then used to free the memory afterwards.

Syntax:

GETMAIN SET(ptr-ref)

{{FLENGTH(data-value) [BELOW]} | (LENGTH(data-value))}

[INITIMG(data-value)]

[SHARED]

[NOSUSPEND]

[USERDATAKEY | CICSDATAKEY]

Setup:

None

Service:

Storage Control

Options:

SET(ptr-ref)

SET retrieves the block of allocated storage.

FLENGTH(data-value)

FLENGTH determines the length of allocated storage. FLENGTH is preferred over LENGTH.

BELOW

The BELOW option is to specify the location of allocated memory. This option has no meaning with the environment and is ignored.

LENGTH(data-value)

LENGTH determines the length of allocated storage. FLENGTH is preferred over LENGTH.

INITIMG(data-value)

Specify a one-character initialization for allocated memory. By default, memory is initialized to low-values / nulls.

SHARED
The SHARED option allocates the memory for shared storage. This is not possible within the environment, so it always raises the condition UNSUPPORTED_OPTION.

**NOSUSPEND**

Do not suspend the task if memory is not available. This continuously waits and retries the allocation.

**USERDATAKEY**

This specifies the location of allocated memory and has no meaning within the environment. It is ignored.

**CICSDATAKEY**

This specifies the location of allocated memory and has no meaning within the environment. It is ignored.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LENGERR</td>
<td>The FLENGTH &lt; 1, or LENGTH = 0.</td>
</tr>
<tr>
<td>NOSTG</td>
<td>The storage requested is not currently available.</td>
</tr>
<tr>
<td>UNSUPPORTED_OPTION</td>
<td>This is always raised by SHARED.</td>
</tr>
</tbody>
</table>

**JOURNAL**

This command is obsolete, replaced by WRITE JOURNALNAME. See WRITE JOURNALNUM for information on the obsolete syntax.

**HANDLE ABEND**

Handle an abnormal termination by executing procedural code or another program.

**Syntax:**

```
HANDLE ABEND
   {CANCEL | RESET | PROGRAM(name) | LABEL(label)}
```
Service:

Program Control

Options:

CANCEL
Specifies that any previous handle for abend should be cancelled.
RESET
Specifies that a previously cancelled abend should be reinstated.
LABEL(label)
Specifies a paragraph or section label to execute upon abend.
PROGRAM(name)
Specifies a program name to link and execute upon abend.

Conditions:

| PGMIDERR | The program could not be linked. This may be delayed until the actual abend occurs. |

HANDLE AID

Handle Attention Identifier instructs the program to automatically transfer control to the given label when the Attention Identifier key is recognized as pressed in the RECEIVE MAP command.

Syntax:

HANDLE
    ANYKEY(label)
    CLEAR(label)
    CLRPARTN(label)
    ENTER(label)
    LIGHTPEN(label)
    OPERID(label)
    PA1…PA3(label)
    PF1…PA24(label)
    TRIGGER(label)
Service:

Execute Interface Program

Setup:

None

Options:

key(label)
Label specifies a paragraph or section name. If label is omitted, then the default handler is restored.

Conditions:

None

HANDLE CONDITION

Handle conditions explicitly using custom program logic rather than the default condition handler (generally the termination dump). These handlers may be suspended and restored using PUSH HANDLE and POP HANDLE.

Syntax:

HANDLE CONDITION {condition(label)}…

Service:

Execute Interface Program

Setup:

None

Options:

condition(label)
Condition specifies a condition name, such as INVREQ, QIDERR, ERROR, etc.
Label specifies a paragraph or section name. If label is omitted, then the default handler is restored.
Conditions:

None

 IGNORE CONDITION

Ignore conditions explicitly rather than executing the default condition handler (generally the termination dump) or other custom handler. These handlers may be suspended and restored using PUSH HANDLE and POP HANDLE.

Syntax:

IGNORE CONDITION {condition}…

Service:

Execute Interface Program

Setup:

None

Options:

condition

Condition specifies a condition name, such as INVREQ, QIDERR, ERROR, etc.

Conditions:

None

 LINK

Link to another program, and then return continuing the existing program.

Syntax:

LINK PROGRAM(name)

[COMMAREA(data-area) [LENGTH(data-area)] [DATALENGTH(data-value)]

[INPUTMSG(data-area) [INPUTMSGLEN(data-value)]]
Service:

Program Control

Setup:

pathname=program Uri

program_name must be a valid program reference, either a class name or URI.

It may be a class-name/program-name, or remote:jndi-name.

pct.name.sysid=jndi_name

This establishes the sysid for a program, when one is not specified in the link.

pct.name.transid=transid_name

This establishes the default transid for a program, when one is not specified in the link.

sysid.name=jndi_name

This establishes a JNDI name for the sysid, possibly referencing a remote EJB.

xlt.name=program_name

program_name must be a valid name in the program table.

Options:

PROGRAM(name)
The program name to be loaded.

COMMAREA(data-area)
The COMMAREA to pass to the linked program.

LENGTH(data-area)
The length of the COMMAREA.

DATALENGTH(data-value)
The data length of the COMMAREA to pass to the other program. This must be less than or equal to the LENGTH.

INPUTMSG(data-area)
Specifies data to be received by the other program’s RECEIVE.
**INPUTMSGLEN(data-value)**

Specifies the length of the INPUTMSG.

**SYSID(name)**

 Specifies the system on which the program is to be linked. This means that the program name's table entry begins with the system id followed by a period, then the normal program name.

**SYNCONRETURN**

 Only meaningful for Distributed Program Link, specifies that the linked program is to take a syncpoint upon completion.

**TRANSID(name)**

 Specifies the transaction name to link.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>TRANSID is all blanks.</td>
</tr>
</tbody>
</table>
| LENGERR   | 11: COMMAREA length not between 0 and 32767  
           | 12: DATALENGTH < 0  
           | 13: DATALENGTH > LENGTH  
           | 27: INPUTMSGLEN not between 0 and 32767 |
| PGMIDERR  | 1: The program name has no table entry.  
           | 3: The program could not be loaded. |

**LOAD**

Load a program or resource into memory.

As the target environment provides a strong separation between program data and program code, whether ENTRY or SET is used to obtain the return reference determines the type of load. If ENTRY, program code is returned; if SET, then program data is returned.

**Syntax:**

```cobol
LOAD {PROGRAM(name) | RESOURCE(name)}
   [SET(pointer-ref)]
   [ENTRY(procedure-pointer-ref)]
```
[LENGTH(data-area) | FLENGTH(data-area)]
[HOLD]

**Service:**

Program Control

**Setup:**

```
rct.name=resource_name
```
resource_name must be the name of a data file in the resources directory, or a classname representing a java.langRunnable class, such as an Elastic COBOL program.

```
pct.name=program_uri
```
program_uri must be a valid program reference, either a classname or URI.

**Options:**

**PROGRAM(name)**
The program name to be loaded.

**RESOURCE(name)**
The resource name to be loaded. This is a synonym for PROGRAM, preferred for data. It is only referenced if PROGRAM is not present.

**ENTRY(procedure-pointer-ref)**
This returns a reference to a Java executable class. This will use the resource table name, if available, otherwise the program table name.

**LENGTH(data-area)**
Only used for data, returns the length of the data area set.

**FLENGTH(data-area)**
Only used for data, returns the length of the data area set.

**SET(pointer-ref)**
Returns the memory containing the data represented by the program or resource name.

**HOLD**
Requests that the data area not be released when the task is ended. As this system uses a garbage collection scheme for memory allocation, this option has no effect. The memory is released automatically whenever no references are made to it.
Conditions:

<table>
<thead>
<tr>
<th>PGMIDERR</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The program name has no table entry.</td>
</tr>
<tr>
<td>3</td>
<td>The program could not be loaded.</td>
</tr>
</tbody>
</table>

**MONITOR**

(This command is unsupported)

MONITOR provides performance information.

**Syntax:**

MONITOR

POINT(data-value) [DATA1(data-area) DATA2(data-area)
ENTRYNAME(data-area)]

**Service:**

Trace Control

**POP HANDLE**

Restore condition handlers from the stack, previously pushed there by PUSH HANDLE. This is useful if desiring the suspension of handling for a routine and it should be paired with PUSH HANDLE.

**Syntax:**

POP HANDLE

**Service:**

Execute Interface Program

**Setup:**

None

**Options:**

None
Conditions:

| INVREQ  | Indicates that there was no matching PUSH HANDLE. |

POST

Post a message to an area of memory after a specified delay. The task continues processing while the post waits to perform its action.

INTERVAL or AFTER specify a period of time, whereas TIME or AT specify the final time itself.

Note that most foreground sessions contain timeouts and will be terminated if delayed too far.

Syntax:

```
POST
  SET(pointer-ref)
  [REQID(data-value)]
    [INTERVAL(hhmmss)]
  [AFTER
    [HOURS(data-value)]
    [MINUTES(data-value)]
    [SECONDS(data-value)]
  ]
  [TIME(hhmmss)]
  [AT
    [HOURS(data-value)]
    [MINUTES(data-value)]
    [SECONDS(data-value)]
  ]
```

Service:

Interval Control

Setup:

None
Options:

**SET(pointer-ref)**
Specifies a pointer (USAGE POINTER) data item which is set to a four-byte memory control area. When the delay is finished and the post is done, the first and third byte are altered to X"40" and X"80". This exact same pointer item may be passed to the WAIT EVENT command.

**REQID(data-value)**
Specifies a request ID through which this command may be cancelled using the CANCEL command. The request ID is valid only within the same session.

**INTERVAL(hhmmss)**
Specifies an interval of time duration data-value. The format is a single number with two-digits each hour, minute, second.

**TIME(hhmmss)**
Specifies a final time data-value. The format is a single number with two-digits each hour, minute, second.

**AFTER**
AFTER specifies that HOURS, MINUTES and SECONDS refers to an interval duration.

**AT**
AT specifies that HOURS, MINUTES and SECONDS refers to a final time.

**HOURS(data-value)**
Specifies the number of hours, and it must be within 0..99.

**MINUTES(data-value)**
Specifies the number of minutes, and it must be within 0..59 if other HOURS or SECONDS is specified.

**SECONDS(data-value)**
Specifies the number of seconds, and it must be within 0..59 if other HOURS or MINUTES is specified.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPIRED</td>
<td>The interval has already expired.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
<tr>
<td></td>
<td>4 Hours are invalid.</td>
</tr>
<tr>
<td></td>
<td>5 Minutes are invalid.</td>
</tr>
<tr>
<td></td>
<td>6 Seconds are invalid.</td>
</tr>
</tbody>
</table>
PURGE MESSAGE

(THIS COMMAND IS UNSUPPORTED)

Purge the current BMS logical message.

Syntax:

```
PURGE MESSAGE
```

Service:

Basic Mapping Support

Setup:

None

Options:

None

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>TSIOERR</td>
<td>Temporary storage input/output error.</td>
</tr>
</tbody>
</table>

PUSH HANDLE

Push the current condition handlers onto the stack. This is useful if desiring the suspension of handling for a routine and it should be paired with POP HANDLE.

Syntax:

```
PUSH HANDLE
```

Service:

Execute Interface Program

Setup:

None
Options:

None

Conditions:

None

QUERY COUNTER

Query attributes of the named counter.

Syntax:

UPDATE { COUNTER(name) | DCOUNTER(name) } [POOL(name)]
   [VALUE(data-area)]
   [MINIMUM(data-area)]
   [MAXIMUM(data-area)]

Service:

Counter

Setup:

pool.name=redirected_pool_name
If not specified, the pool name is the given name.

counter.jndi=counter_jndi_name
Specify the JNDI name of the Counter service. The default is
‘CounterService’.

Options:

COUNTER(name)
Specifies the name of the 32-bit counter.

DCOUNTER(name)
Specifies the name of the 64-bit counter.

POOL(name)
Specifies the name of the pool of counters. Each Counter service maintains
its own pools of counters.
VALUE(data-area)
Retrieves the current value of the counter. The counter's value is only retrieved, not updated, by this action.

MINIMUM(data-area)
Retrieves the minimum value of the counter, as given upon its definition.

MAXIMUM(data-area)
Retrieves the maximum value of the counter, as given upon its definition.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
<tr>
<td>201</td>
<td>Named counter invalid.</td>
</tr>
<tr>
<td>301</td>
<td>Server error.</td>
</tr>
<tr>
<td>303</td>
<td>Unexpected error, such as connectivity loss.</td>
</tr>
<tr>
<td>304</td>
<td>Invalid pool.</td>
</tr>
<tr>
<td>305</td>
<td>Cannot connect to server.</td>
</tr>
<tr>
<td>306</td>
<td>Server abend.</td>
</tr>
<tr>
<td>308</td>
<td>This condition cannot occur (options table not loadable).</td>
</tr>
<tr>
<td>309</td>
<td>This condition cannot occur (options table error).</td>
</tr>
<tr>
<td>310</td>
<td>This condition cannot occur (options user exit).</td>
</tr>
<tr>
<td>403</td>
<td>The pool name contains invalid characters.</td>
</tr>
<tr>
<td>404</td>
<td>The counter name contains invalid characters</td>
</tr>
</tbody>
</table>

LENGERR | LENGERR occurs only for COUNTER commands, not DCOUNTER commands. |
| 001 | The value has become negative. |
| 002 | The value is too large by one bit. |
| 003 | The value is too large by more than one bit. |

QUERY SECURITY
This command is dependent upon the capabilities of the External Security Manager.

Syntax:

QUERY SECURITY
   {RESTYPE(data-value) | RESCLASS(data-value) RESIDLENGTH(data-value)}
   RESID(data-value)
   [LOGMESSAGE(cvda)]
   [READ(cvda)]
Service:

Security

Setup:

None

Options:

**ALTER(cvda)**
Retrieves whether user has alter authority for resource, returning ALTERABLE or NOTALTERABLE.

**CONTROL(cvda)**
Retrieves whether user has control authority for resource, returning CTRLABLE or NOTCTRLABLE.

**LOGMESSAGE(cvda)**
Specifies whether security violation messages are logged, passing LOG or NOLOG.

**READ(cvda)**
Retrieves whether user has read authority for resource, returning READABLE or NOTREADABLE.

**RESCLASS(data-value)**
Specifies 8-character resource class. SPCOMMAND refers to an ETP resource.

**RESID(data-value)**
Specifies the resource ID to check.

**RESIDLENGTH(data-value)**
Specifies the length of the resource ID to check.

**RESTYPE(data-value)**
Specifies the type of resource, 1 to 12 characters, from the list:

- **DB2ENTRY**
- **SPCOMMAND**
- **FILE**
- **TDQUEUE**
- **JOURNALNAME**
- **TRANSACTION**
- **JOURNALNUM**
- **TRANSATTACH**
- **PROGRAM**
- **TSQUEUE**
- **PSB**
**UPDATE(cvda)**

Retrieves whether user has update authority for resource, returning UPDATABLE or NOTUPDATABLE.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td></td>
</tr>
<tr>
<td>LENGERR</td>
<td></td>
</tr>
<tr>
<td>NOTFND</td>
<td></td>
</tr>
<tr>
<td>QIDERR</td>
<td></td>
</tr>
</tbody>
</table>

**READ**

Read a single record from the file.

**Syntax:**

```
READ {FILE(data-value) | DATASET(data-value))
  [UPDATE [TOKEN(data-area)]
  | UNCOMMITTED
  | CONSISTENT
  | REPEATABLE
  ]
  {INTO(data-area) | SET(pointer-ref)}
RIDFLD(data-area) [KEYLENGTH(data-value) [GENERIC]]
SYSID(data-value) LENGTH(data-area) | LENGTH(data-area)]
[RBA | RRN | DEBKEY | DEBREC]
[EQUAL | GTEQ]
[NOSUSPEND]
```

**Service:**

File Control

**Setup:**

```
file.<filename>=jdbc:< xml_file_descriptor.xml>[:<sql_connection_name>]
```

The xml_file_descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XFD FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is 'file' by default.
sql.<sql_connection_name>.datasource=<jndi_datasource_name>
The file service uses the SQL connection named 'file' by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

Options:

**CONSISTENT**
Specifies data integrity, the record is read in a consistent manner with other data clients.

**DATASET(data-value)**
Specifies the filename the same as FILE, but FILE is preferred.

**DEBKEY**
Deblocking by key in BDAM, this option is not used.

**DEBREC**
Deblocking by record in BDAM, this option is not used.

**EQUAL**
Specifies that the record's key must be equal to the given record ID field.

**FILE(data-value)**
Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE('abc') would refer to setup entry 'file.abc=jdbc:abc.xml'.

**GENERIC**
Specifies that the record ID field is generic, matching any with the given contents equal up the length of keylength.

**GTEQ**
Specifies that the record's key must be greater than or equal to the given record ID field.

**INTO(data-area)**
Specifies the data-area into which the record is read.

**KEYLENGTH(data-value)**
Specifies the length of the record ID key field.

**LENGTH(data-area)**
Specifies the length of the into field. Upon return, this may indicate the length of the actual retrieved record. As a mapping is occurring between records and SQL, the return length will generally be the maximum record length.

**NOSUSPEND**
The read does not suspend on locks.
RBA
Specifies that the key is by relative byte address.

REPEATABLE
Specifies data integrity, the record read is repeatable.

RIDFLD(data-area)
Specifies the record ID field, the key used to search for the record. This data item is mapped to the column marked with a ridfld attribute in the XML file descriptor.

RRN
Specifies that the key is by relative record number.

SET(pointer-ref)
Retrieves the data record into a data area pointed to by the pointer ref.

SYSID(data-value)
Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

TOKEN(data-area)
Specifies a token to be used later in a rewrite command.

UNCOMMITTED
Specifies data integrity, the record read is uncommitted.

UPDATE
Specifies that the record is to be read for update, necessary for the rewrite command.

Conditions:

<table>
<thead>
<tr>
<th>KEY</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPKEY</td>
<td>The record has a duplicate key that has not yet been read.</td>
</tr>
<tr>
<td>FILENOTFOUND</td>
<td>The file itself could not be found.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>A logical error occurred.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error occurred, such as an invalid SQL query.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length of the record read exceeds the given INTO buffer's LENGTH.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The record was not found, such as an invalid record ID field.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The file is not open.</td>
</tr>
</tbody>
</table>
**READNEXT**

Read the next record from the file during a browse operation. This must be bracketed by STARTBR and ENDBR commands.

**Syntax:**

```
READNEXT {FILE(data-value) | DATASET(data-value)}
   [UPDATE [TOKEN(data-area)]
   | UNCOMMITTED
   | CONSISTENT
   | REPEATABLE
   ]
 {INTO(data-area) | SET(pointer-ref)}
 RIDFLD(data-area) [KEYLENGTH(data-value) [REQID(data-value)]]
 [SYSID(data-value) LENGTH(data-area) | LENGTH(data-area)]
 [RBA | RRN]
 [NOSUSPEND]
```

**Service:**

File Control

**Setup:**

```
file.<filename>=jdbc:< xml_file_descriptor.xml>[<sql_connection_name>]
```

The xml_file_descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XFD FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is 'file’ by default.

```
sql.<sql_connection_name>.datasource=<jndi_datasource_name>
```

The file service uses the SQL connection named ‘file’ by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

**Options:**

**CONSISTENT**
Specifies data integrity, the record is read in a consistent manner with other data clients.

**DATASET**(data-value)
Specifies the filename the same as FILE, but FILE is preferred.

**FILE**(data-value)
Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE('abc') would refer to setup entry 'file.abc=jdbc:abc.xml'.

**INTO**(data-area)
Specifies the data-area into which the record is read.

**KEYLENGTH**(data-value)
Specifies the length of the record ID key field.

**LENGTH**(data-area)
Specifies the length of the into field. Upon return, this may indicate the length of the actual retrieved record. As a mapping is occurring between records and SQL, the return length will generally be the maximum record length.

**NOSUSPEND**
The read does not suspend on locks.

**RBA**
Specifies that the key is by relative byte address.

**REPEATABLE**
Specifies data integrity, the record read is repeatable.

**REQID**(data-value)
Specifies a browser's request ID.

**RIDFLD**(data-area)
Specifies the record ID field, the key used to search for the record. This data item is mapped to the column marked with an ridfld attribute in the XML file descriptor.

**RRN**
Specifies that the key is by relative record number.

**SET**(pointer-ref)
Retrieves the data record into a data area pointed to by the pointer ref.

**SYSID**(data-value)
Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

**TOKEN**(data-area)
Specifies a token to be used later in a rewrite command.

**UNCOMMITTED**

Specifies data integrity, the record read is uncommitted.

**UPDATE**

Specifies that the record is to be read for update, necessary for the rewrite command.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPKEY</td>
<td>The record has a duplicate key that has not yet been read.</td>
</tr>
<tr>
<td>ENDFILE</td>
<td>The end of the browse request's result set has been reached.</td>
</tr>
<tr>
<td>FILENOTFOUND</td>
<td>The file itself could not be found.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>A logical error occurred.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error occurred, such as an invalid SQL query.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length of the record read exceeds the given INTO buffer's LENGTH.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The record was not found, such as an invalid record ID field.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The file is not open.</td>
</tr>
<tr>
<td>RECORDBUSY</td>
<td>The record was locked.</td>
</tr>
</tbody>
</table>

**READPREV**

Read the previous record from the file during a browse operation. This must be bracketed by STARTBR and ENDBR commands.

Note, not all SQL JDBC drivers support the operation required to browse previous records. An IOERR with a message about an operation not being supported by JDBC 2.0 will be given if the JDBC driver does not have sufficient support for this command.

**Syntax:**

```
READPREV (FILE(data-value) | DATASET(data-value))
[UPDATE [TOKEN(data-area)]
| UNCOMMITTED
| CONSISTENT
| REPEATABLE
```


} {INTO(data-area) | SET(pointer-ref)}
RIDFLD(data-area) [KEYLENGTH(data-value) [REQID(data-value)]]
[SYSID(data-value) LENGTH(data-area) | LENGTH(data-area)]
[RBA | RRN]
[NOSUSPEND]

Service:

File Control

Setup:

file.<filename>=jdbc:< xml_file_descriptor.xml>[;<sql_connection_name>]
The xml_file_descriptor is an XML file described in the file setup, containing
information relating the database columns to the record positions. It must be
placed in the resources folder. Placing a $XFD FILE="filename.xml"
command before a record in the data division will generate a starting
template in the listing folder. The sql_connection_name is 'file' by default.

sql.<sql_connection_name>.datasource=<jndi_datasource_name>
The file service uses the SQL connection named 'file' by default, but it may
be overridden in the file setup. It must be setup in the same manner as all
SQL connections.

Options:

CONSISTENT
Specifies data integrity, the record is read in a consistent manner with other
data clients.

DATASET(data-value)
Specifies the filename the same as FILE, but FILE is preferred.

FILE(data-value)
Specifies the filename. This filename is referred to in the setup via
file.<filename>. For example, FILE('abc') would refer to setup entry
'file.abc=jdbc:abc.xml'.

INTO(data-area)
Specifies the data-area into which the record is read.

KEYLENGTH(data-value)
Specifies the length of the record ID key field.

LENGTH(data-area)
Specifies the length of the into field. Upon return, this may indicate the length of the actual retrieved record. As a mapping is occurring between records and SQL, the return length will generally be the maximum record length.

**NOSUSPEND**
The read does not suspend on locks.

**RBA**
Specifies that the key is by relative byte address.

**REPEATABLE**
Specifies data integrity, the record read is repeatable.

**REQID(data-value)**
Specifies a browser’s request ID.

**RIDFLD(data-area)**
Specifies the record ID field, the key used to search for the record. This data item is mapped to the column marked with an ridfld attribute in the XML file descriptor.

**RRN**
Specifies that the key is by relative record number.

**SET(pointer-ref)**
Retrieves the data record into a data area pointed to by the pointer ref.

**SYSID(data-value)**
Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

**TOKEN(data-area)**
Specifies a token to be used later in a rewrite command.

**UNCOMMITTED**
Specifies data integrity, the record read is uncommitted.

**UPDATE**
Specifies that the record is to be read for update, necessary for the rewrite command.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPKEY</td>
<td>The record has a duplicate key that has not yet been read.</td>
</tr>
<tr>
<td>ENDFILE</td>
<td>The end of the browse request’s result set has been reached.</td>
</tr>
<tr>
<td>FILENOTFOUND</td>
<td>The file itself could not be found.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>A logical error occurred.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error occurred, such as an invalid SQL query.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length of the record read exceeds the given INTO buffer's LENGTH.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The record was not found, such as an invalid record ID field.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The file is not open.</td>
</tr>
<tr>
<td>RECORDBUSY</td>
<td>The record was locked.</td>
</tr>
</tbody>
</table>

### READQ TD

Read data from a transient data queue.

**Syntax:**

```
READQ TD QUEUE(name)
    [INTO(data-area) | SET(pointer-ref)]
    [LENGTH(data-area)]
    [SYSID(name)]
    [NOSUSPEND]
```

**Service:**

Transient Data Control

**Setup:**

```
queue.sysid_value.name=uri
[syid.name=sysid_value]
```

**Options**

**QUEUE(name)**

Specifies the name of the transient data queue. The queue must be defined.

**INTO(data-area)**

Specifies the data area into which the queue data is copied.

**SET(pointer-ref)**

Returns a pointer to the read data record.
LENGTH(data-value)
Specifies the length of the INTO buffer, if provided. Returns the length of the returned data.

SYSID(name)
Specifies the system ID. This is used to determine which queue definition is used.

Conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIDERR</td>
<td>An unknown sysid is being used.</td>
</tr>
<tr>
<td>QIDERR</td>
<td>There is no such queue definition.</td>
</tr>
<tr>
<td>QZERO</td>
<td>The end of the queue has been reached.</td>
</tr>
<tr>
<td>QBUSY</td>
<td>The queue was busy and could not be read, only possible if NOSUSPEND is specified.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>The queue consumer could not be created or started, for reasons such as being an output only queue.</td>
</tr>
<tr>
<td>IOERR</td>
<td>The receive from queue responded with an error.</td>
</tr>
</tbody>
</table>

READQ TS
Read data from a temporary storage data queue.

Syntax:

READQ TS (QUEUE(name) | QNAME(name))
[INTO(data-area) | SET(pointer-ref)]
[LENGTH(data-area)]
[SYSID(name)]
[NOSUSPEND]
[NUMITEMS(data-area)]
[NEXT | ITEM(data-value)]

Service:
Temporary Storage Control
Setup:

```cobol
queue(sysid_value).name(main | auxiliary)=uri
[sysid.name=sysid_value]
```

Options

**QUEUE(name)**
Specifies the name of the transient data queue. The queue must be defined.

**QNAME(name)**
See QUEUE.

**INTO(data-area)**
Specifies the data area into which the queue data is copied.

**SET(pointer-ref)**
Returns a pointer to the read data record.

**LENGTH(data-value)**
Specifies the length of the INTO buffer, if provided. Returns the length of the returned data.

**SYSID(name)**
Specifies the system ID. This is used to determine which queue definition is used.

**NUMITEMS(data-area)**
Returns the number of items in the queue.

**ITEM(data-area)**
Specifies the queue item number to read.

**NEXT**
Specifies that the next item in the queue is to be read. This position counter is shared at the scope of the queue, which may not be the individual task.

Conditions

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIDERR</td>
<td>An unknown sysid is being used.</td>
</tr>
<tr>
<td>QIDERR</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>There is no such queue definition.</td>
</tr>
<tr>
<td>2</td>
<td>The queue could not be obtained.</td>
</tr>
<tr>
<td>QBUSY</td>
<td>The queue was busy and could not be read, only possible if NOSUSPEND is specified.</td>
</tr>
<tr>
<td>ITEMERR</td>
<td>The requested item is not present in the queue.</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length of the read data is longer than the allocated LENGTH. This is only possible when INTO is used rather than SET.</td>
</tr>
</tbody>
</table>

## RECEIVE MAP

Receive the presentation map's input.

### Syntax:

```
RECEIVE MAP(name)
   [MAPSET(name)]
   [INTO(data-area) | SET(pointer-ref)]
   [TERMINAL [ASIS] [INPARTN(data-value)]
   | FROM(data-area) [LENGTH(data-value)]
```

### Service:

Basic Mapping Support

### Setup:

None

### Options:

- **ASIS**
- **FROM(data-area)**
- **INPARTN(data-value)**
- **INTO(data-area)**
- **LENGTH(data-value)**
- **MAP(name)**
  The name is a literal data-value.
- **MAPSET(name)**
The name is a literal data-value.

**SET(pointer-ref)**

**TERMINAL**

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOC</td>
</tr>
<tr>
<td>EODS</td>
</tr>
<tr>
<td>INVMPSZ</td>
</tr>
<tr>
<td>INVPARTN</td>
</tr>
<tr>
<td>INVREQ</td>
</tr>
<tr>
<td>MAPFAIL</td>
</tr>
<tr>
<td>PARTNFAIL</td>
</tr>
<tr>
<td>RDATT</td>
</tr>
<tr>
<td>UNEXPIN</td>
</tr>
</tbody>
</table>

**RECEIVE PARTN**

**Syntax:**

```
RECEIVE PARTN(data-area)
  {INTO(data-area) | SET(pointer-ref)}
  [LENGTH(data-value)]
  [ASIS]
```

**Service:**

Basic Mapping Support

**Setup:**

None

**Options:**

- **ASIS**
- **INTO(data-area)**
- **LENGTH(data-value)**
- **PARTN(data-area)**
- **SET(pointer-ref)**
**RELEASE**

Release a loaded program resource.

This command does nothing in this system, as this system employs garbage collection for automatic release of resources when no longer referenced. The resource is available when any pointers or procedure-pointers referencing it have been altered to point to another item, including null.

It is good practice to null such pointers in all cases as in other systems they would become invalid after release.

**Syntax:**

```
RELEASE PROGRAM(name)
```

**Service:**

Program Control

**Setup:**

None

**Options:**

**PROGRAM(name)**

The program name is the one to be released.

**Conditions:**

None

**RESETBR**

Reset a browse operation. This is similar to an ENDBR followed by a STARTBR.
**Syntax:**

```
RESETBR
   {FILE(data-value) | DATASET(data-value)}
   [KEYLENGTH(data-value) [GENERIC]]
   [REQID(data-value)]
   [SYSID(data-value)]
   [GTEQ | EQUAL]
   [RBA | RRN]
```

**Service:**

File Control

**Setup:**

```
file.<filename>=jdbc:< xml_file_descriptor.xml>[:<sql_connection_name>]
```

The xml_file_descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XF D FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is ‘file’ by default.

```
sql.<sql_connection_name>.datasource=<jndi_datasource_name>
```

The file service uses the SQL connection named ‘file’ by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

**Options:**

**DATASET(data-value)**

Specifies the filename the same as FILE, but FILE is preferred.

**EQUAL**

Specifies that the record's key must be equal to the given record ID field.

**FILE(data-value)**

Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE('abc') would refer to setup entry 'file.abc=jdbc:abc.xml'.

**GENERIC**

Specifies that the record ID field is generic, matching any with the given contents equal up the length of keylength.

**GTEQ**
Specifies that the record's key must be greater than or equal to the given record ID field.

**KEYLENGTH(data-value)**

Specifies the length of the record ID key field.

**RBA**

Specifies that the key is by relative byte address.

**REQID(data-value)**

Specifies a request ID to be used in browsing operations.

**RIDFLD(data-area)**

Specifies the record ID field, the key used to search for the record. This data item is mapped to the column marked with an ridfld attribute in the XML file descriptor.

**RRN**

Specifies that the key is by relative record number.

**SYSID(data-value)**

Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILENOTFOUND</td>
<td>The file itself could not be found.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>A logical error occurred.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error occurred, such as an invalid SQL query.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The record was not found, such as an invalid record ID field.</td>
</tr>
</tbody>
</table>

**RETRIEVE**

Retrieve data passed to the task by the START command.

If there are multiple retrieve records passed, each retrieve will fetch the next until ENDDATA is reached.

**Syntax:**

```
RETRIEVE
  {INTO(data-area) | SET(pointer-ref)}
```
Service:
Interval Control

Setup:
None

Options:

**INTO(data-area)**
Retrieves the data passed by START FROM.

**SET(pointer-ref)**
Retrieves the data passed by START FROM as a pointer.

**LENGTH(data-area)**
Specifies the length of the INTO data-area.
Retrieves the length of the START FROM buffer actually passed.

**RTRANSID(data-area)**
Retrieves the 4-character RTRANSID passed by the START command.

**RTERMID(data-area)**
Retrieves the 4-character RTERMID passed by the START command.

**QUEUE(data-area)**
Retrieves the 8-character name of the temporary storage queue passed by START.

**WAIT**

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDDATA</td>
<td>ENDDATA is returned when the end of data passed is reached.</td>
</tr>
<tr>
<td>ENVDEFERR</td>
<td>Retrieve attempted to access a parameter that was not passed.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The specified length is insufficient to hold the</td>
</tr>
</tbody>
</table>
RETURN

Return program control the next highest level, possibly setting the next transaction to process.

The ability to set the transaction is what allows pseudo-conversations to function. When a program has processed its input and generated its output, it marks the next transaction to receive the input from the current output.

Syntax:

```
RETURN

[TRANSID(name) [COMMAREA(data-area) [LENGTH(data-value)]]]
[INPUTMSG(data-area) [INPUTMSGLEN(data-value)]]
[ENDACTIVITY]
```

Service:

Program Control

Setup:

```
pct.name=program_uri
```

program_name must be a valid program reference, either a classname or URI.

```
xlt.name=program_name
```

program_name must be a valid name in the program table.

Options:

```
TRANSID(name)
```

Returns the transaction name to use on the next leg of a pseudo-conversation, or to use immediately if IMMEDIATE is passed.

```
COMMAREA(data-area)
```

The COMMAREA to pass to the linked program.

```
LENGTH(data-area)
```

The length of the COMMAREA.

```
INPUTMSG(data-area)
```

The returned data.

NOTFND
Specifies data to be received by the other program’s RECEIVE.

**INPUTMSGLEN(data-value)**

Specifies the length of the INPUTMSG.

**IMMEDIATE**

Specifies that the transaction is tied to the facility immediately.

**ENDACTIVITY**

This is used only for BTS (Business Transaction Services). This is ignored since BTS is not supported.

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cannot return COMMAREA when no facility present.</td>
</tr>
<tr>
<td>2</td>
<td>IMMEDIATE or COMMAREA was passed at other than the top program level.</td>
</tr>
<tr>
<td>8</td>
<td>Cannot return INPUTMSG when no facility present.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LENGERR</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>COMMAREA length not between 0 and 32767.</td>
</tr>
<tr>
<td>27</td>
<td>INPUTMSGLEN not between 0 and 32767.</td>
</tr>
</tbody>
</table>

**REWIND COUNTER**

Rewind the named counter that has reached its limit.

**Syntax:**

```
REWIND { COUNTER(name) | DCOUNTER(name) } [POOL(name)]
INCREMENT(data-value])
```

**Service:**

Counter

**Setup:**

```
pool.name=redirected_pool_name
```

If not specified, the pool name is the given name.

```
counter.jndi=counter_jndi_name
```
Specify the JNDI name of the Counter service. The default is 'CounterService'.

Options:

COUNTER(name)
Specifies the name of the 32-bit counter.

DCOUNTER(name)
Specifies the name of the 64-bit counter.

POOL(name)
Specifies the name of the pool of counters. Each Counter service maintains its own pools of counters.

INCREMENT(data-value)
The counter is valid to reset only if already at its limit, or if the given increment would reach its limit.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Named counter invalid.</td>
</tr>
<tr>
<td>301</td>
<td>Server error.</td>
</tr>
<tr>
<td>303</td>
<td>Unexpected error, such as connectivity loss.</td>
</tr>
<tr>
<td>304</td>
<td>Invalid pool.</td>
</tr>
<tr>
<td>305</td>
<td>Cannot connect to server.</td>
</tr>
<tr>
<td>306</td>
<td>Server abend.</td>
</tr>
<tr>
<td>308</td>
<td>This condition cannot occur (options table not loadable).</td>
</tr>
<tr>
<td>309</td>
<td>This condition cannot occur (options table error).</td>
</tr>
<tr>
<td>310</td>
<td>This condition cannot occur (options user exit).</td>
</tr>
<tr>
<td>403</td>
<td>The pool name contains invalid characters.</td>
</tr>
<tr>
<td>404</td>
<td>The counter name contains invalid characters.</td>
</tr>
<tr>
<td>406</td>
<td>The increment is invalid; it cannot be larger than the total range of the counter.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUPPRESSED</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>The counter has not reached its limit, including increment if specified.</td>
</tr>
</tbody>
</table>

REWRITE

Rewrite a record previously read using the update option.
Syntax:

REWRITE

\{FILE(data-value) | DATASET(data-value)\}

\[TOKEN(data-area)\]

FROM(data-area)

\[SYSID(data-value) LENGTH(data-area) | LENGTH(data-area)\]

\[NOSUSPEND\]

Service:

File Control

Setup:

file.<filename>=jdbc:< xml_file_descriptor.xml>[:<sql_connection_name>]

The xml_file_descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XFD FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is 'file' by default.

sql.<sql_connection_name>.datasource=<jndi_datasource_name>

The file service uses the SQL connection named 'file' by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

Options:

DATASET(data-value)

Specifies the filename the same as FILE, but FILE is preferred.

FILE(data-value)

Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE('abc') would refer to setup entry 'file.abc=jdbc:abc.xml'.

LENGTH(data-area)

Specifies the length of the from field.

NOSUSPEND

The read does not suspend on locks.

SYSID(data-value)

Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.
TOKEN(data-area)
Specifies a token relating this REWRITE back to a READ.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILENOTFOUND</td>
<td>The file itself could not be found.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>A logical error occurred.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error occurred, such as an invalid SQL query.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length of the record read exceeds the given INTO buffer's LENGTH.</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The data record to rewrite was not found, such as having been just deleted.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>RECORDBUSY</td>
<td>The record was locked.</td>
</tr>
</tbody>
</table>

ROUTE

(This command is unsupported)

Syntax:

ROUTE

ROUTE

[INTERVAL(hhmms)]
[AFTER
[HOURS(data-value)]
[MINUTES(data-value)]
[SECONDS(data-value)]
]
[TIME(hhmms)]
[AT

[HOURS(data-value)]
[MINUTES(data-value)]
[SECONDS(data-value)]
]

[ERRTERM | ERRTERM(data-value)]
[TITLE(data-area)]
[List(data-area)]
Service:

Basic Mapping Support

Setup:

None

Options:

`INTERVAL(hhmmss)`
Specifies an interval of time duration data-value. The format is a single number with two-digits each hour, minute, second.

`TIME(hhmmss)`
Specifies a final time data-value. The format is a single number with two-digits each hour, minute, second.

`AFTER`
AFTER specifies that HOURS, MINUTES and SECONDS refers to an interval duration.

`AT`
AT specifies that HOURS, MINUTES and SECONDS refers to a final time.

`HOURS(data-value)`
Specifies the number of hours, and it must be within 0..99.

`MINUTES(data-value)`
Specifies the number of minutes, and it must be within 0..59 if other HOURS or SECONDS is specified.

`SECONDS(data-value)`
Specifies the number of seconds, and it must be within 0..59 if other HOURS or MINUTES is specified.

`FROM(data-area)`
Specifies the user data area to be passed to the new task.

`ERRTERM(data-value)`
Specifies a terminal to be notified of any error in routing the message.

`LDC(data-value)`
Specifies 2-character logical device code.

LIST(data-area)
Specifies list of terminals and operators.

NLEOM
New line end of message.

OPCLASS(data-area)
Specifies list of operator classes.

REQID(data-value)
Specifies 2-character prefix of temporary storage queue.

TITLE(data-area)
Specifies the title of the route message.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IREQID</td>
</tr>
<tr>
<td>INVERRTERM</td>
</tr>
<tr>
<td>INVLDC</td>
</tr>
<tr>
<td>INVREQ</td>
</tr>
<tr>
<td>RTEFAIL</td>
</tr>
<tr>
<td>RTESAME</td>
</tr>
</tbody>
</table>

SEND CONTROL

Syntax:

SEND CONTROL
   [CURSOR | CURSOR(data-value)]
   [FORMFEED]
   [ERASE [DEFAULT | ALTERNATE] | ERASEAUP]
   [PRINT]
   [FREEKB]
   [ALARM]
   [FRSET]

Service:

Basic Mapping Support
Setup:

None

Options:

CURSOR(data-value)
FORMFEED
ERASE
DEFAULT
ALTERNATE
ERASEAUP
PRINT
FREEKB
ALARM
FRSET

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGREQCD</td>
</tr>
<tr>
<td>IGREQID</td>
</tr>
<tr>
<td>INVLDIC</td>
</tr>
<tr>
<td>INVPARTN</td>
</tr>
<tr>
<td>INVREQ</td>
</tr>
<tr>
<td>RETPAGE</td>
</tr>
<tr>
<td>TSIOERR</td>
</tr>
<tr>
<td>WRBRK</td>
</tr>
</tbody>
</table>

**SEND MAP**

**Syntax:**

SEND MAP(name)

MAPSET(name)

[FROM(data-area) | MAPONLY]

[DATAONLY]

[LENGTH(data-value)]

[CURSOR | CURSOR(data-value)]

[FORMFEED]
Service:

Basic Mapping Support

Setup:

None

Options:

MAP(name)
The name is a literal data-value.

MAPSET(name)
The name is a literal data-value.

FROM(data-area)

MAPONLY
DATAONLY

LENGTH(data-value)
CURSOR(data-value)
FORMFEED
ERASE
DEFAULT
ALTERNATE
ERASEAUP
PRINT
FREEKB
ALARM
FRSET
SEND PAGE

(THIS COMMAND IS UNSUPPORTED)

Syntax:

SEND PAGE

[RELEASE [TRANSID(name)] | RETAIN]
[TRAILER(data-area)]
[SET(pointer-ref)]
[AUTOPAGE [CURRENT | ALL] | NOAUTOPAGE]
[OPERPURGE]
[FMHPARM(name)]
[LAST]

Service:

Basic Mapping Support

Setup:

None

Options:

ALL
AUTOPAGE
CURRENT
FMHPARM(name)
LAST
NOAUTOPAGE
OPERPURGE
RELEASE
RETAIN
SET(pointer-ref)
TRAILER(data-area)
TRANSID(name)
Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGREQCD</td>
</tr>
<tr>
<td>INVREQ</td>
</tr>
<tr>
<td>RETPAGE</td>
</tr>
<tr>
<td>TSIOERR</td>
</tr>
<tr>
<td>WRBRK</td>
</tr>
</tbody>
</table>

SEND PARTNSET

(THIS COMMAND IS UNSUPPORTED)

Syntax:

SEND {PARTNSET | PARTNET(name)}

Service:

Basic Mapping Support

Setup:

None

Options:

PARTNSET(name)

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVPARTNSET</td>
</tr>
<tr>
<td>INVREQ</td>
</tr>
</tbody>
</table>

SEND TEXT

Syntax:

SEND TEXT
  FROM(data-area)
  [LENGTH(data-value)]
  [CURSOR(data-value)]
  [FORMFEED]
[ERASE [DEFAULT|ALTERNATE]]
[PRINT]
[FREEKB]
[ALARM]
[NLEOM]
[FMHPARM(name)]
[LDC(name) | [OUTPARTN(name)] [ACTPARTN(name)]]
[MSR(data-value)]

**Service:**

Basic Mapping Support

**Setup:**

None

**Options:**

FROM(data-area)
LENGTH(data-value)
CURSOR(data-value)
FORMFEED
ERASE
DEFAULT
ALTERNATE
PRINT
FREEKB
ALARM
NLEOM
FMHPARM(name)
LDC(name)
OUTPARTN(name)
ACTPARTN(name)
MSR(data-value)
Conditions:

<table>
<thead>
<tr>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGREQCD</td>
</tr>
<tr>
<td>IGREQID</td>
</tr>
<tr>
<td>INVLD</td>
</tr>
<tr>
<td>INVPARTN</td>
</tr>
<tr>
<td>INVREQ</td>
</tr>
<tr>
<td>LENGERR</td>
</tr>
<tr>
<td>RETPAGE</td>
</tr>
<tr>
<td>TSIOERR</td>
</tr>
<tr>
<td>WRBRK</td>
</tr>
</tbody>
</table>

**SIGNOFF**

This command is dependent upon the capabilities of the External Security Manager.

**Syntax:**

```plaintext
SIGNOFF
```

**Service:**

Security

**Setup:**

None

**Options:**

None

**Conditions:**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
</tbody>
</table>

**SIGNON**

This command is dependent upon the capabilities of the External Security Manager.
Syntax:

SIGNON
  USERID(data-value)
  [ESMREASON(data-area)]
  [ESMRESP(data-area)]
  [GROUPID(data-value)]
  [LANGUAGECODE(data-value) | NATLANG(data-value)]
  [LANGINUSE(data-area)]
  [NATLANGINUSE(data-area)]
  [PASSWORD(data-value)]
  [NEWPASSWORD(data-value)]
  [OIDCARD(data-value)]

Service:

Security

Setup:

None

Options:

ESMREASON(data-area)
Retrieves the external security manager reason code.

ESMRESP(data-area)
Retrieves the external security manager response code.

GROUPID(data-value)
Specifies the group ID for a user.

LANGUAGECODE(data-value)
Specifies the 3-character language code.

LANGINUSE(data-area)
Retrieves the 3-character language code assigned by login.

NATLANG(data-value)
Specifies the 1-character language code.

NATLANGINUSE(data-area)
Retrieves the 1-character language code assigned by login.
NEWPASSWORD(data-value)
Specifies the new user password.

OIDCARD(data-value)
Specifies a 65-byte magnetic strip value.

PASSWORD(data-value)
Specifies the current user password.

USERID(data-value)
Specifies the user ID.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>USERIDERR</td>
<td>The user ID is invalid.</td>
</tr>
</tbody>
</table>

SPOOLCLOSE

Close the spool file.

Syntax:

SPOOLCLOSE TOKEN(data-area)
[ KEEP | DELETE ]

Service:
Spool

Setup:

A spool queue for output is defined as:
spool[.userid.[class].[node]]=uri

and a spool queue for input as:
spool[.userid.[class]]=uri

Options:

TOKEN(data-area)
The 8-character token contents determine the spool file to be closed. It must be one opened previously by SPOOLOPEN.
KEEP
Specifies disposition of the closing spool file.
For input, the same spool file will be read at the next open input.
For output, the spool file is to be passed to its destination node.

DELETE
Specifies disposition of the closing spool file.
For input, the next open is the next spool file will be opened at the next open input.
For output, the spool file is to be purged.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSPOOL</td>
<td>4</td>
<td>The specified spool file could not be found. The token may be invalid.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>8</td>
<td>The spool file is not open, and so could not be closed.</td>
</tr>
</tbody>
</table>

SPOOLOPEN INPUT

Open a spool file for input.

Syntax:

```
SPOOLOPEN INPUT TOKEN(data-area)
   USERID(data-value)
   [CLASS(data-value)]
```

Service:

Spool

Setup:

```
spool[.userid[.class]]=uri
```

Options:

```
TOKEN(data-area)
```
The 8-character token contents determine the spool file to be closed. It must be one opened previously by SPOOLOPEN.
USERID(data-value)
Specifies the user id to be associated with the spool file.

CLASS(data-value)
Specifies a one (1) character class designation.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSPOOL</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILLOGIC</td>
<td>3</td>
</tr>
</tbody>
</table>

SPOOLOPEN OUTPUT

Open a spool file for output.

Syntax:

SPOOLOPEN OUTPUT TOKEN(data-area)
  USERID(data-value)
  NODE(data-value)
  [CLASS(data-value)]
  OUTDESCR(pointer-ref)
  [NOCC | ASA | MCC]
  [PRINT | PRINT [RECORDLENGTH(data-value)] | PUNCH]

Service:

Spool

Setup:

spool[.userid[.class[.node]]]=uri
Options:

TOKEN(data-area)
The 8-character token contents determine the spool file to be closed. It must be one opened previously by SPOOLOPEN.

USERID(data-value)
Specifies the user id to be associated with the spool file.

CLASS(data-value)
Specifies a one (1) character class designation.

NODE(data-value)
Specifies the eight (8) character destination node.

OUTDESCR(pointer-ref)
The OUTDESCR option is not supported, and will raised an UNSUPPORTED_OPTION condition if used

NOCC
Requests that the records in the spool file do not include internal carriage controls.

ASA
Requests that records in the spool file begin with an ASA carriage-control character.

MCC
Requests that records in the spool file begin with an IBM Machine Command Code character.

PRINT
The default, specifies that large records (up to 32760 bytes) may be sent to the spool.

RECORDLENGTH(data-value)
Specifies the maximum length of a record in the spool file. The default is 32760.

PUNCH
Specifies that the record length is 80.

Conditions:

<table>
<thead>
<tr>
<th>NOSPOOL</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The specified spool file could not be found. The token may be invalid.</td>
</tr>
</tbody>
</table>

INVREQ
SPOOLREAD

Read data from a spool file opened for input.

Syntax:

SPOOLREAD TOKEN(data-area)
    INTO(data-area)
    [MAXFLENGTH(data-value)]
    [TOFLENGTH(data-area)]

Service:

Spool

Setup:

spool[.userid[.class]]=uri

Options:

TOKEN(data-area)
The 8-character token contents determine the spool file to be closed. It
must be one opened previously by SPOOLOPEN.

INTO(data-area)
Specifies the data area to which the read record is copied.

MAXFLENGTH(data-value)
Specifies the maximum length of the input record.
TOFLENGTH(data-area)

Returns the length of the data read.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDFILE</td>
<td>The spool file has been completely read.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>The CLASS is invalid.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>The INTO is not specified</td>
</tr>
<tr>
<td>NOSPOOL</td>
<td>The specified spool file could not be found. The token may be invalid.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The spool file is not open.</td>
</tr>
<tr>
<td></td>
<td>The spool file was opened for output.</td>
</tr>
</tbody>
</table>

SPOOLWRITE

Write data to a spool file opened for output.

Syntax:

SPOOLWRITE TOKEN(data-area)
FROM(data-area)
[MAXFLENGTH(data-value)]
[FLENGTH(data-value)]
[LINEN | PAGE]

Service:

Spool

Setup:

spool[.userid[.class[.node]]]=uri
Options:

TOKEN(data-area)
The 8-character token contents determine the spool file to be closed. It must be one opened previously by SPOOLOPEN.

FROM(data-value)
Specifies the data record to write to the spool file.

FLENGTH(data-area)
Specifies the length of the data record to write to the spool file. The length must be between 1 and the record length specified on the open.

LINE
Specifies LINE format.

PAGE
Specifies PAGE format.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDFILE</td>
<td>The spool file has been completely read.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>The CLASS is invalid.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>The FROM is not specified.</td>
</tr>
<tr>
<td>NOSPOOL</td>
<td>The specified spool file could not be found. The token may be invalid.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The spool file is not open.</td>
</tr>
<tr>
<td></td>
<td>The spool file was opened for input.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length is not between 1 and the opened record length.</td>
</tr>
</tbody>
</table>

START

Start a separate task at a specific time.
The separate task is executed as a JEE message bean. It has no terminal and executed in the background.
INTERVAL or AFTER specifies a period of time, whereas TIME or AT specifies the final time itself.

Syntax:

START

[TRANSID(data-value)]
[REQID(data-value)]

[INTERVAL(hhmmss)]
[AFTER

[HOURS(data-value)]
[MINUTES(data-value)]
[SECONDS(data-value)]
]
[TIME(hhmmss)]
[AT

[HOURS(data-value)]
[MINUTES(data-value)]
[SECONDS(data-value)]
]

[FROM(data-area)] [LENGTH(data-value) [FMH]]
[USERID(data-value) | TERMID(data-value)]
[SYSID(data-value)]
[RTRANSID(data-value)]
[RTERMID(data-value)]
[QUEUE(data-value)]
[NOCHECK]
[PROTECT]

Service:

Interval Control

Setup:

queue[.<sysid>].etpsysq=jms:<message_queue_indi_name>
The default Elastic Transaction Platform System Queue is etpsysq_<sysid>. Each deployment has its own system message queue.

<message_queue_indi_name>.factory=<jms_factory_indi_name>
If not present, then the jms.default.factory setting is used.

```plaintext
<message_queue_jndi_name>.name=<jms_login_user_name>
<message_queue_jndi_name>.password=<jms_login_password>
```

The user and password are only required if JMS is setup to require them. Additionally, the transaction must be setup the same as for link wherever the start message will be received. The transaction property restart may be set to allow the transaction to be automatically restarted upon failure.

**Options:**

**TRANSID(data-value)**

Specifies the transaction ID to execute.

**REQID(data-value)**

Specifies a request ID through which this command may be cancelled using the CANCEL command. The request ID is valid only within the same session.

**INTERVAL(hhmmss)**

Specifies an interval of time duration data-value. The format is a single number with two-digits each hour, minute, second.

**TIME(hhmmss)**

Specifies a final time data-value. The format is a single number with two-digits each hour, minute, second.

**AFTER**

AFTER specifies that HOURS, MINUTES and SECONDS refers to an interval duration.

**AT**

AT specifies that HOURS, MINUTES and SECONDS refers to a final time.

**HOURS(data-value)**

Specifies the number of hours, and it must be within 0..99.

**MINUTES(data-value)**

Specifies the number of minutes, and it must be within 0..59 if other HOURS or SECONDS is specified.

**SECONDS(data-value)**

Specifies the number of seconds, and it must be within 0..59 if other HOURS or MINUTES is specified.

**FROM(data-area)**

Specifies the user data area to be passed to the new task.

**LENGTH(data-value)**
Specifies the length of the user data area to be passed to the new task.

**FMH**
Specifies that the user data contains function management headers.

**USERID(data-value)**
Specifies the user ID to be used for the new task.

**TERMID(data-value)**
Specifies the terminal ID for the new task. As all terminals are virtual bridges, and the new task may not have a terminal, this option is not supported. Any virtual terminal must activate its own task. The way to associate a terminal with a timed task is to have the client virtual terminal poll a transaction at the specified time, with all such client-driven interaction controlled by the client.

**SYSID(data-value)**
Specifies the system ID for the new task. This is used in finding the correct system message queue.

**RTRANSID(data-value)**
Specifies a 4-character name passed to the new task, traditionally a transaction ID.

**RTERMID(data-value)**
Specifies a 4-character name passed to the new task, traditionally a terminal ID.

**QUEUE(data-value)**
Specifies an 8-character name passed to the new task, traditionally a temporary storage queue name.

**NOCHECK**
Specifies that less error checking will occur for possibly improved performance. This option is ignored.

**PROTECT**
Specifies that a syncpoint must occur before the START may take place.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSIDERR</td>
<td>The transaction ID is invalid.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length is less than zero (0).</td>
</tr>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
<tr>
<td></td>
<td>4 Hours are invalid.</td>
</tr>
<tr>
<td></td>
<td>5 Minutes are invalid.</td>
</tr>
<tr>
<td></td>
<td>6 Seconds are invalid.</td>
</tr>
<tr>
<td>QIDERR</td>
<td>The system message queue used to pass start messages has failed.</td>
</tr>
</tbody>
</table>
IOERR | Communication with the system message queue used to pass start messages has failed.

**SUSPEND**

Suspend a task, allowing other tasks to proceed first where the task scheduler permits. This is otherwise known as a yield.

**Syntax:**

SUSPEND

**Service:**

Task Control

**Options:**

None

**START ATTACH**

Start a separate task immediately.

The separate task is executed as a JEE message bean. It has no terminal and executed in the background.

INTERVAL or AFTER specifies a period of time, whereas TIME or AT specifies the final time itself.

**Syntax:**

START ATTACH

[TRANSID(data-value)]

[FROM(data-area) [LENGTH(data-value)]]

**Service:**

Interval Control

**Setup:**
queue[.<sysid>].etpsysq=jms:<message_queue_jndi_name>
The default Elastic Transaction Platform System Queue is etpsysq_<sysid>. Each deployment has its own system message queue.

<message_queue_jndi_name>.factory=<jms_factory_jndi_name>
If not present, then the jms.default.factory setting is used.

<message_queue_jndi_name>.name=<jms_login_user_name>
<message_queue_jndi_name>.password=<jms_login_password>
The user and password are only required if JMS is setup to require them.

Additionally, the transaction must be setup the same as for link wherever the start message will be received. The transaction property restart may be set to allow the transaction to be automatically restarted upon failure.

**Options:**

**TRANSID(data-value)**
Specifies the transaction ID to execute.

**REQID(data-value)**
Specifies a request ID through which this command may be cancelled using the CANCEL command. The request ID is valid only within the same session.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPIRED</td>
<td>The interval has already expired.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
<tr>
<td></td>
<td>4 Hours are invalid.</td>
</tr>
<tr>
<td></td>
<td>5 Minutes are invalid.</td>
</tr>
<tr>
<td></td>
<td>6 Seconds are invalid.</td>
</tr>
</tbody>
</table>

**START BREXIT**

*(THIS COMMAND IS UNSUPPORTED)*

The START BREXIT command is not supported. The Elastic Transaction Platform always operates under an externally defined bridge.

**Syntax:**

```
START
  {BREXIT | BREXIT(data-value)}
  TRANSID(data-value)
  [BRDATA(data-area) [BRDATALength(data-value)]]
  [USERID(data-value)]
```
Service:

Interval Control

Options:

BREXIT(data-value)
TRANSID(data-value)
BRDATA(data-area)
BRDATALENGTH(data-value)
USERID(data-value)

STARTBR

Start browsing a file, preparing it for use with READNEXT or READPREV. The browse must be terminated using ENDBR when finished.

Currently, the only records available during a browse operation are those which are in the scope of the STARTBR operation. A READPREV cannot go further back than STARTBR's browse implies. A READNEXT cannot go further forward than STARTBR's browse implies.

Syntax:

STARTBR
   {FILE(data-value) | DATASET(data-value)}
   RIDFLD(data-area) [KEYLENGTH(data-value) [GENERIC]]
   [REQID(data-value)]
   [SYSID(data-value)]
   [RBA | RRN | DEBKEY | DEBREC]
   [GTEQ | EQUAL]
The file service uses the SQL connection named 'file' by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

**Options:**

**DATASET(data-value)**
Specifies the filename the same as FILE, but FILE is preferred.

**DEBKEY**
Deblocking by key in BDAM, this option is not used.

**DEBREC**
Deblocking by record in BDAM, this option is not used.

**EQUAL**
Specifies that the record's key must be equal to the given record ID field.

**FILE(data-value)**
Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE('abc') would refer to setup entry 'file.abc=jdbc:abc.xml'.

**GENERIC**
Specifies that the record ID field is generic, matching any with the given contents equal up the length of keylength.

**GTEQ**
Specifies that the record's key must be greater than or equal to the given record ID field.

**KEYLENGTH(data-value)**
Specifies the length of the record ID key field.

**RBA**
Specifies that the key is by relative byte address.

**REQID(data-value)**
Specifies a request ID to be used in browsing operations.

**RIDFLD(data-area)**
Specifies the record ID field, the key used to search for the record. This data item is mapped to the column marked with an ridfld attribute in the XML file descriptor.

**RRN**
Specifies that the key is by relative record number.

**SYSID(data-value)**
Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILENOTFOUND</td>
<td>The file itself could not be found.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>A logical error occurred.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error occurred, such as an invalid SQL query.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>NOTFND</td>
<td>The record was not found, such as an invalid record ID field.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The file is not open.</td>
</tr>
</tbody>
</table>

SUSPEND

Suspend a task, allowing other tasks to proceed first where the task scheduler permits. This is otherwise known as a yield.

Syntax:

SUSPEND

Service:

Task Control

Setup:

None

Options:

None

Conditions:

None
SYNCPOINT

Commit a unit of work within a task. A commit occurs automatically when a task ends normally.

**Syntax:**

```
SYNCPOINT
```

**Service:**

Recovery

**Options:**

None

**Conditions:**

| ROLLBACK  | Occurs when the task cannot commit, but rather is forced to rollback. Generally, this will occur when a remotely linked program must rollback. |

SYNCPOINT ROLLBACK

Rollback a unit of work within a task. A rollback occurs automatically when a task ends abnormally.

EIBSYNRB, EIBERR and EIBERRCD may be set.

**Syntax:**

```
SYNCPOINT ROLLBACK
```

**Service:**

Recovery

**Options:**

None
Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>SYNCPOINT ROLLBACK failed.</td>
</tr>
</tbody>
</table>

**TRACE**

Enable or disable tracing facilities. The trace command affects only the environment in which it is executed, not the global environment.

**Syntax:**

```
TRACE {ON | OFF}
  [SYSTEM]
  [EI]
  [USER]
  [SINGLE]
```

**Setup:**

- `trace.master={true|false}`
  The trace.master switch enables or disables all physical tracing. If `trace.master=false`, then the tracing commands function but no actual output is created.

- `trace=journal_name`
  The tracing destination is a journal name, which in turn is specified through `journal.name=journal_protocol`.

**Service:**

TraceControl

**Options:**

- **ON**
  TRACE settings are enabled.

- **OFF**
  TRACE settings are disabled.

- **SYSTEM**
  TRACE setting applies to all tracing system activity.

- **EI**
TRACE settings applies to EXEC interface activity.

**USER**
TRACE setting applies to user entries.

**SINGLE**
TRACE setting applies only to user trace entries within the current task. All USER entries are only within the current task, not global, so this is no different than USER.

**UNLOCK**

Release file locks. As SQL uses a different mechanism for concurrency control, this command does nothing.

**Syntax:**

```
UNLOCK
    {FILE(data-value) | DATASET(data-value)}
    [TOKEN(data-area)]
    [SYSID(data-value)]
```

**Service:**

File Control

**Setup:**

```
file.<filename>=jdbc:< xml_file_descriptor.xml>[;<sql_connection_name>]
```

The xml_file_descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XFD FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is ‘file’ by default.

```
sql.<sql_connection_name>.datasource=<jndi_datasource_name>
```

The file service uses the SQL connection named ‘file’ by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

**Options:**

**DATASET(data-value)**
Specifies the filename the same as FILE, but FILE is preferred.

**FILE(data-value)**
Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE(abc) would refer to setup entry ‘file.abc=jdbc:abc.xml’.

**SYSID(data-value)**

Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

**TOKEN(data-area)**

Specifies a token relating this REWRITE back to a READ.

**Conditions:**

None

---

**UPDATE COUNTER**

Update the current value of the named counter.

**Syntax:**

```
UPDATE { COUNTER(name) | DCOUNTER(name) }  
[POOL(name)]
VALUE(data-value)
[COMPAREMIN(data-value)]
[COMPAREMAX(data-value)]
```

**Service:**

Counter

**Setup:**

```
pool.name=redirected_pool_name
If not specified, the pool name is the given name.

counter.jndi=counter_jndi_name
Specify the JNDI name of the Counter service. The default is 'CounterService'.
```

**Options:**

**COUNTER(name)**

Specifies the name of the 32-bit counter.
**DCOUNTER***(name)***
Specifies the name of the 64-bit counter.

**POOL***(name)***
Specifies the name of the pool of counters. Each Counter service maintains its own pools of counters.

**VALUE**(data-value)
Specifies the value of the counter.

**COMPAREMIN**(data-value)
Specifies a minimum value against which to compare the value. If the value is greater than or equal, then no action is taken; if less than the minimum, a condition is raised.

If COMPAREMIN>COMPAREMAX, then either condition must be true rather than both.

Success of the command is conditional upon the comparison.

**COMPAREMAX**(data-value)
Specifies a maximum value against which to compare the value. If the value is less than or equal, then no action is taken; if less than the minimum, a condition is raised.

If COMPAREMIN>COMPAREMAX, then either condition must be true rather than both.

Success of the command is conditional upon the comparison.

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Named counter invalid.</td>
</tr>
<tr>
<td>301</td>
<td>Server error.</td>
</tr>
<tr>
<td>303</td>
<td>Unexpected error, such as connectivity loss.</td>
</tr>
<tr>
<td>304</td>
<td>Invalid pool.</td>
</tr>
<tr>
<td>305</td>
<td>Cannot connect to server.</td>
</tr>
<tr>
<td>306</td>
<td>Server abend.</td>
</tr>
<tr>
<td>308</td>
<td>This condition cannot occur (options table not loadable).</td>
</tr>
<tr>
<td>309</td>
<td>This condition cannot occur (options table error).</td>
</tr>
<tr>
<td>310</td>
<td>This condition cannot occur (options user exit).</td>
</tr>
<tr>
<td>403</td>
<td>The pool name contains invalid characters.</td>
</tr>
<tr>
<td>404</td>
<td>The counter name contains invalid characters.</td>
</tr>
<tr>
<td>406</td>
<td>The increment is invalid; it cannot be larger than the total range of the counter.</td>
</tr>
</tbody>
</table>

**SUPPRESSED**

<table>
<thead>
<tr>
<th>RESP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>103 The value is not within COMPAREMIN and</td>
</tr>
</tbody>
</table>
UNLOCK

Release file locks. As SQL uses a different mechanism for concurrency control, this command does nothing.

Syntax:

UNLOCK

{FILE(data-value) | DATASET(data-value)}

[TOKEN(data-area)]

[SYSID(data-value)]

Service:

File Control

Setup:

file.<filename>=jdbc:< xml_file Descriptor.xml>[;<sql_connection_name>]

The xml_file Descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XFD FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is 'file' by default.

sql.<sql_connection_name>.datasource=<jndi_datasource_name>

The file service uses the SQL connection named 'file' by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

Options:

DATASET(data-value)

Specifies the filename the same as FILE, but FILE is preferred.

FILE(data-value)

Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE('abc') would refer to setup entry 'file.abc=jdbc:abc.xml'.

SYSID(data-value)
Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

**TOKEN(data-area)**
Specifies a token relating this REWRITE back to a READ.

**Conditions:**
None

**VERIFY PASSWORD**
This command is dependent upon the capabilities of the External Security Manager.

**Syntax:**

```
VERIFY PASSWORD(data-value)
    USERID(data-value)
    [CHANGETIME(data-area)]
    [DAYSLEFT(data-area)]
    [ESMREASON(data-area)]
    [ESMRESP(data-area)]
    [EXPIRYTIME(data-area)]
    [INVALIDCOUNT(data-area)]
    [LASTUSETIME(data-area)]
```

**Service:**
Security

**Setup:**
None

**Options:**

```
PASSWORD(data-value)
    Specifies the user's password.
USERID(data-value)
    Specifies the user id.
```
**CHANGETIME(data-area)**
Retrieves data and time password last changed, as in ABSTIME.

**DAYSLEFT(data-area)**
Retrieves number of days until password expires, -1 if no expiration.

**ESMREASON(data-area)**
Retrieves the external security manager reason code.

**ESMRESP(data-area)**
Retrieves the external security manager response code.

**EXPIRYTIME(data-area)**
Retrieves date and time password expires, as in ABSTIME.

**INVALIDCOUNT(data-area)**
Retrieves count of invalid password attempts by user.

**LASTUSETIME(data-area)**
Retrieves last time user id was used, as in ABSTIME.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>USERID</td>
<td>Invalid user ID.</td>
</tr>
</tbody>
</table>

**WAIT EXTERNAL**

(THIS COMMAND IS UNSUPPORTED)
This command is not supported and not intended for COBOL usage.
Synchronize events.

**Syntax:**

```
WAIT EXTERNAL
   ECBLIST(pointer-value)
   NUMEVENTS(data-value)
   {PURGEABLE | PURGEABILITY(cvda) | NOTPUREGABLE}
   [NAME(data-value)]
```

**Service:**

Task Management
Setup:
None

Options:
None

Conditions:
None

**WAIT EVENT**

Wait for a post event to occur.

**Syntax:**

```
WAIT EVENT
   ECADDR(pointer-value)
   [NAME(data-value)]
```

**Service:**

Interval Control

**Setup:**

None

**Options:**

**ECADDR(pointer-value)**
Specifies the pointer returned via the POST command. This pointer-value cannot be another source, or the post will wait indefinitely.

**NAME(data-value)**
Specifies a name for the wait. This value is not currently used.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPIRED</td>
<td>The interval has already expired.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
</tbody>
</table>
4 Hours are invalid.
5 Minutes are invalid.
6 Seconds are invalid.

**WAIT JOURNALNAME**

**Syntax:**

```cobol
WAIT JOURNALNAME(data-value)
   [REQID(data-value)]
```

**Service:**

Journal

**Setup:**

```cobol
journal.<name>=<journal_protocol_reference>
```

**Options:**

**JOURNALNAME(data-value)**

Data-value is a two-digit binary value, representing the journal name DFHJnn, where nn is the two-digit value.

**REQID(data-value)**

Specify the request ID obtained from WRITE JOURNALNAME on which to wait. Note that many journal protocols will wait on the entire journal to be synchronized rather than only a particular request.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>There have been no writes on which to wait.</td>
</tr>
<tr>
<td>IOERR</td>
<td>Input/output error.</td>
</tr>
<tr>
<td>JIDERR</td>
<td>The journal reference is unknown.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The journal itself was recognized but could not be opened.</td>
</tr>
</tbody>
</table>

**WAIT JOURNALNUM**

Wait for the journal to be completely written before continuing.
Syntax:

WAIT JOURNALNUM(data-value)
[REQID(data-value)]

Service:

Journal

Setup:

journal.DFHJ<nn>=<journal_protocol_reference>

Options:

JOURNALNUM(data-value)
Data-value is a two-digit binary value, representing the journal name
DFHJnn, where nn is the two-digit value.

REQID(data-value)
Specify the request ID obtained from WRITE JOURNALNUM on which to
wait. Note that many journal protocols will wait on the entire journal to be
synchronized rather than only a particular request.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>There have been no writes on which to wait.</td>
</tr>
<tr>
<td>IOERR</td>
<td>Input/output error.</td>
</tr>
<tr>
<td>JIDERR</td>
<td>The journal reference is unknown.</td>
</tr>
</tbody>
</table>
| NOTOPEN   | The journal itself was recognized but could not be
            | opened. |

WAITCICS

(THIS COMMAND IS UNSUPPORTED)
This command is not supported and not intended for COBOL usage.

Syntax:

WAITCICS

ECBLIST(pointer-value)
NUMEVENTS(data-value)
{PURGEABLE | PURGEABILITY(cvda) | NOTPURGEABLE}
[NAME(data-value)]

Service:
Task Management

Setup:
None

Options:
None

Conditions:
None

WAIT JOURNALNAME

Syntax:

WAIT JOURNALNAME(data-value)
[REQID(data-value)]

Service:
Journal

Options:

JOURNALNAME(data-value)
Data-value is a two-digit binary value, representing the journal name DFHJnn, where nn is the two-digit value.

REQID(data-value)
Specify the request ID obtained from WRITE JOURNALNAME on which to wait. Note that many journal protocols will wait on the entire journal to be synchronized rather than only a particular request.

Conditions:

INVREQ There have been no writes on which to wait.
WAIT JOURNALNUM

Wait for the journal to be completely written before continuing.

Syntax:

WAIT JOURNALNUM(data-value)
    [REQID(data-value)]

Service:

Journal

Options:

JOURNALNUM(data-value)
Data-value is a two-digit binary value, representing the journal name DFHJnn, where nn is the two-digit value.

REQID(data-value)
Specify the request ID obtained from WRITE JOURNALNUM on which to wait. Note that many journal protocols will wait on the entire journal to be synchronized rather than only a particular request.

Conditions:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>There have been no writes on which to wait.</td>
</tr>
<tr>
<td>IOERR</td>
<td>Input/output error.</td>
</tr>
<tr>
<td>JIDERR</td>
<td>The journal reference is unknown.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The journal itself was recognized but could not be opened.</td>
</tr>
</tbody>
</table>
WEB ENDBROWSE COOKIE

End the browsing of HTTP cookies.

Syntax:

WEB ENDBROWSE COOKIE

Service:

Web

Options:

None

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
<tr>
<td>4</td>
<td>There was no prior WEB STARTBROWSE COOKIE.</td>
</tr>
</tbody>
</table>

WEB ENDBROWSE FORMFIELD

End the browsing of HTTP form fields.

Syntax:

WEB ENDBROWSE FORMFIELD

Service:

Web

Options:

None

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
</tbody>
</table>
WEB ENDBROWSE HTTPHEADER

End the browsing of HTTP headers.

Syntax:

WEB ENDBROWSE HTTPHEADER

Service:

Web

Options:

None

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
<tr>
<td>4</td>
<td>There was no prior WEB STARTBROWSE HTTPHEADER.</td>
</tr>
</tbody>
</table>

WEB EXTRACT

Extract information about the Web HTTP request.
Note that not all fields are available from all HTTP requests.

Syntax:

WEB EXTRACT

[HTTPMETHOD(data-area) METHODLENGTH(data-area)]
[HTTPVERSION(data-area) VERSIONLEN(data-area)]
[PATH(data-area) PATHLENGTH(data-area)]
[REQUESTTYPE(cvda)]
[AUTHTYPE(data-area) AUTHLENGTH(data-area)]
Service:

Web

Options:

HTTPMETHOD(data-area)
Returns the HTTP method (e.g. "GET").

METHODLENGTH(data-area)
Pass HTTPMETHOD buffer length, returns the length of the extracted HTTPMETHOD.

HTTPVERSION(data-area)
Returns the HTTP version string (e.g. "HTTP/1.1").

VERSIONLEN(data-area)
Pass HTTPVERSION buffer length, returns the length of the extracted HTTPVERSION.
PATH(data-area)
Returns the HTTP path string.

PATHLENGTH(data-area)
Pass PATH buffer length, returns the length of the extracted PATH.

REQUESTTYPE(cvda)
Returns HTTP or NONHTTP

AUTHTYPE(data-area)

AUTHLENGTH(data-area)

CONTEXTPATH(data-area)
(e.g. "/myapp").

CONTEXTPATHLENGTH(data-area)

PATHTRANSLATED(data-area)

PATHTRANSLATEDLENGTH(data-area)

QUERYSTRING(data-area)
Returns the query string from the URL, the portion after the question mark (?). (e.g. "hl=en&ie=UTF-8&oe=UTF-8&q=Search+Query").

QUERYSTRINGLENGTH(data-area)

REMOTEUSER(data-area)
Returns the remote user if logged in (e.g. "root").

REMOTEUSERLENGTH(data-area)

REQUESTEDSESSIONID(data-area)

REQUESTEDSESSIONIDLENGTH(data-area)

REQUESTURI(data-area)
(e.g. "/myapp/myapp").

REQUESTURILength(data-area)

REQUESTURL(data-area)
Returns the entire URL for this web request. (e.g. "http://192.168.0.234:7001/myapp/myapp").

REQUESTURLLENGTH(data-area)

SERVLETPATH(data-area)
(e.g. "/myapp").

SERVLETPATHLENGTH(data-area)

CHARACTERENCODING(data-area)
Returns the character encoding.
CHARACTERENCODINGLENGTH(data-area)
Returns length of CHARACTERENCODING.

CONTENTLENGTH(data-area)
Returns the content length.

CONTENTTYPE(data-area)

CONTENTTYPELENGTH(data-area)

LOCALE(data-area)
Returns the locale in language2_COUNTRY2 format. (e.g. "en_US").

LOCALELENGTH(data-area)

PROTOCOL(data-area)
(e.g. "HTTP/1.1").

PROTOCOLLENGTH(data-area)

REMOTEADDR(data-area)
Returns the remote machine's TCP/IP address. (e.g. "192.168.0.123").

REMOTEADDRLENGTH(data-area)

REMOTEHOST(data-area)
Returns the remote machine's hostname, if possible. (e.g. "mycomputer.internal.myhost.com").

REMOTEHOSTLENGTH(data-area)

SCHEME(data-area)
(e.g. "http").

SCHEMELENGTH(data-area)

SERVERNAME(data-area)
(e.g. "192.168.0.234").

SERVERNAMELENGTH(data-area)

SERVERPORT(data-area)
Returns the server TCP/IP port.

SECURE(cvda)
Returns SSL or NOSSL.

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>The command is a not a Web application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
<tr>
<td>LENGERR</td>
<td></td>
</tr>
</tbody>
</table>
The passed LENGTH was less than zero (0).

WEB READ COOKIE

Retrieve named HTTP cookie.

Syntax:

WEB READ
   COOKIE(data-area)
   NAMELENGTH(data-value)
   VALUE(data-area)
   VALUELENGTH(data-area)
   COMMENT(data-area) COMMENTLENGTH(data-area)
   DOMAIN(data-area) DOMAINLENGTH(data-area)
   MAXAGE(data-area)
   PATH(data-area) PATHLENGTH(data-area)
   SECURE(cvda)
   VERSION(data-area)

Service:

Web

Options:

COOKIE(data-area)
Specify the HTTP cookie to retrieve.
NAMELENGTH(data-value)
Specify the length of the HTTP cookie.
VALUE(data-area)
Retrieves the value of the HTTP cookie.
VALUELENGTH(data-area)
Pass the length of the VALUE buffer, retrieves the length of the VALUE.
COMMENT(data-area)
Retrieves the cookie’s comment. (Comment is not supported by early versions of JEE.)
COMMENTLENGTH(data-area)
Pass the length of the COMMENT buffer, retrieve the length of the retrieved comment.

**DOMAIN(data-area)**
Retrieves the cookie's domain.

**DOMAINLENGTH(data-area)**
Pass the length of the DOMAIN buffer, retrieve the length of the retrieved domain.

**MAXAGE(data-area)**
Retrieves the maximum age of the cookie.

**PATH(data-area)**
Retrieves the path of the cookie.

**PATHLENGTH(data-area)**
Pass the length of the PATH buffer, retrieve the length of the retrieved path.

**SECURE(data-area)**
Returns SECURE or NOSECURE, reflecting the status of cookie.

**VERSION(data-area)**
Returns the version of the cookie.

### Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td></td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td></td>
<td>There is no HTTP request.</td>
</tr>
<tr>
<td></td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LENGERR</td>
<td>The passed VALUELENGTH was less than zero (0).</td>
</tr>
<tr>
<td></td>
<td>The cookie value was truncated, buffer too small.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTFND</td>
<td>The cookie could not be found.</td>
</tr>
</tbody>
</table>

**WEB READ FORMFIELD**

Retrieve named HTTP form field.
Syntax:

WEB READ
FORMFIELD(data-area)
NAMELENGTH(data-value)
VALUE(data-area)
VALUELENGTH(data-area)

Service:

Web

Options:

FORMFIELD(data-area)
Specify the HTTP form field to retrieve.
NAMELENGTH(data-value)
Specify the length of the HTTP form field.
VALUE(data-area)
Retrieves the value of the HTTP form field.
VALUELENGTH(data-area)
Pass the length of the VALUE buffer, retrieves the length of the VALUE.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INVREQ</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LENGERR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The passed VALUELENGTH was less than zero (0).</td>
</tr>
<tr>
<td>2</td>
<td>The form field value was truncated, buffer too small.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NOTFND</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The cookie could not be found.</td>
</tr>
</tbody>
</table>
WEB READ HTTPHEADER

Retrieve named HTTP header.

Syntax:

WEB READ
  HTTPHEADER(data-area)
  NAMELENGTH(data-value)
  VALUE(data-area)
  VALUELENGTH(data-area)

Service:

Web

Options:

HTTPHEADER(data-area)
  Specify the HTTP header to retrieve.

NAMELENGTH(data-value)
  Specify the length of the HTTP header.

VALUE(data-area)
  Retrieves the value of the HTTP header.

VALUELENGTH(data-area)
  Pass the length of the VALUE buffer, retrieves the length of the VALUE.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LENGERR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The passed VALUELENGTH was less than zero (0).</td>
</tr>
<tr>
<td>2</td>
<td>The header value was truncated, buffer too small.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NOTFND</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The header could not be found.</td>
</tr>
</tbody>
</table>
WEB READNEXT COOKIE

Retrieve the next HTTP cookie.

Syntax:

WEB READNEXT
    COOKIE(data-area) NAMELENGTH(data-area)
    VALUE(data-area) VALUELENGTH(data-area)
    COMMENT(data-area) COMMENTLENGTH(data-area)
    DOMAIN(data-area) DOMAINLENGTH(data-area)
    MAXAGE(data-area)
    PATH(data-area) PATHLENGTH(data-area)
    SECURE(cvda)
    VERSION(data-area)

Service:

Web

Options:

COOKIE(data-area)
    Retrieves the field name.

NAMELENGTH(data-area)
    Pass the length of the FORMFIELD buffer, retrieve the length of the
    FORMFIELD.

VALUE(data-area)
    Retrieves the FORMFIELD's value.

VALUELENGTH(data-area)
    Pass the length of the VALUE buffer, retrieve the length of the VALUE.

COMMENT(data-area)
    Retrieves the cookie's comment. (Comment is not supported by early
    versions of JEE.)

COMMENTLENGTH(data-area)
    Pass the length of the COMMENT buffer, retrieve the length of the retrieved
    comment.

DOMAIN(data-area)
    Retrieves the cookie's domain.
DOMAINLENGTH(data-area)
Pass the length of the DOMAIN buffer, retrieve the length of the retrieved domain.

MAXAGE(data-area)
Retrieves the maximum age of the cookie.

PATH(data-area)
Retrieves the path of the cookie.

PATHLENGTH(data-area)
Pass the length of the PATH buffer, retrieve the length of the retrieved path.

SECURE(data-area)
Returns SECURE or NOSECURE, reflecting the status of cookie.

VERSION(data-area)
Returns the version of the cookie.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDFILE</td>
<td>The end of the HTTP headers has been reached.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td></td>
<td>1 The command is a not a Web application.</td>
</tr>
<tr>
<td></td>
<td>3 There is no HTTP request.</td>
</tr>
<tr>
<td></td>
<td>4 There was no prior WEB STARTBROWSE COOKIE</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The passed VALUELENGTH was less than zero (0).</td>
</tr>
<tr>
<td></td>
<td>1 The passed VALUELENGTH was less than zero (0).</td>
</tr>
<tr>
<td></td>
<td>4 The form field name was truncated, buffer too small.</td>
</tr>
<tr>
<td></td>
<td>5 The value was truncated, buffer too small.</td>
</tr>
</tbody>
</table>

WEB READNEXT FORMFIELD

Retrieve the next HTTP form field.

Syntax:

WEB READNEXT
    FORMFIELD(data-area) NAMELENGTH(data-area)
    VALUE(data-area) VALUELENGTH(data-area)

Service:

Web
Options:

**FORMFIELD(data-area)**
Retrieves the field name.

**NAMELENGTH(data-area)**
Pass the length of the FORMFIELD buffer, retrieve the length of the FORMFIELD.

**VALUE(data-area)**
Retrieves the FORMFIELD’s value.

**VALUELENGTH(data-area)**
Pass the length of the VALUE buffer, retrieve the length of the VALUE.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ENDFILE</strong></td>
<td>The end of the HTTP headers has been reached.</td>
</tr>
<tr>
<td><strong>INVREQ</strong></td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
<tr>
<td>4</td>
<td>There was no prior WEB STARTBROWSE FORMFIELD.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LENGERR</strong></td>
<td>The passed length was less than zero (0).</td>
</tr>
<tr>
<td>1</td>
<td>The passed length was less than zero (0).</td>
</tr>
<tr>
<td>4</td>
<td>The form field name was truncated, buffer too small.</td>
</tr>
<tr>
<td>5</td>
<td>The value was truncated, buffer too small.</td>
</tr>
</tbody>
</table>

**WEB READNEXT HTTPHEADER**

Retrieve the next HTTP header.

Syntax:

```cobol
WEB READNEXT
   HTTPHEADER(data-area) NAMELENGTH(data-area)
   VALUE(data-area) VALUELENGTH(data-area)
```

Service:

Web
Options:

**HTTPHEADER**(data-area)
Retrieves the HTTP header name.

**NAMELENGTH**(data-area)
Pass the length of the HTTPHEADER buffer, retrieve the length of the HTTPHEADER.

**VALUE**(data-area)
Retrieves the HTTP header's value.

**VALUELENGTH**(data-area)
Pass the length of the VALUE buffer, retrieve the length of the VALUE.

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ENDFILE</strong></td>
<td>The end of the HTTP headers has been reached.</td>
</tr>
<tr>
<td><strong>INVREQ</strong></td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
<tr>
<td>4</td>
<td>There was no prior WEB STARTBROWSE HTTPHEADER.</td>
</tr>
<tr>
<td><strong>LENGERR</strong></td>
<td>The passed length was less than zero (0).</td>
</tr>
<tr>
<td>1</td>
<td>The passed length was less than zero (0).</td>
</tr>
<tr>
<td>4</td>
<td>The header name was truncated, buffer too small.</td>
</tr>
<tr>
<td>5</td>
<td>The value was truncated, buffer too small.</td>
</tr>
</tbody>
</table>

**WEB RECEIVE**

Receive all data from the HTTP request.

Syntax:

```
WEB RECEIVE
    { INTO(data-area) [MAXLENGTH(data-value)] | SET(ptr-ref) }
    LENGTH(data-area)
    [TYPE(cvda)]
    [NOTRUNCATE]
    [CLNTCODEPAGE(name)]
    [HOSTCODEPAGE(name)]
```
Service:

Web

Options:

**INTO**(data-area)
Specifies the buffer to contain the data.

**SET**(ptr-ref)
Specifies a pointer, which will then point to the data.

**LENGTH**(data-area)
Returns the length of the data received.

**MAXLENGTH**(data-value)
Specifies the maximum length of data to receive.

**TYPE**(cvda)
Returns HTTP or NONHTTP.

**NOTRUNCATE**
If the data exceeds the maximum amount, do not truncate; rather, return it on subsequent RECEIVE commands.

**CLNTCODEPAGE**(name)
Specifies the name of the client's codepage.
HOSTCODEPAGE must also be specified.

**HOSTCODEPAGE**(name)
Specifies the name of the host's codepage.
CLNTCODEPAGE must also be specified.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>The command is a not a Web application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LENGERR</th>
<th>The passed length was less than zero (0).</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The length is positive, but exceeded the maximum.</td>
</tr>
</tbody>
</table>

**WEB RETRIEVE**

Retrieve a document.
Syntax:

WEB RETRIEVE DOCTOKEN(data-area)

Service:

Web

Options:

DOCTOKEN(data-area)

Specifies 16-byte buffer to contain DOCTOKEN of document previously sent using WEB SEND.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is not a Web application.</td>
</tr>
<tr>
<td>2</td>
<td>There was no prior WEB SEND command.</td>
</tr>
</tbody>
</table>

WEB SEND

Send documents or text to the HTTP client.

Syntax:

WEB SEND

{ }

DOCTOKEN(name)

[ CONTENTTYPE(data-area) CONTENTTYPELENGTH(data-value) ] | TEXT(data-value) | TEXTLINE(data-value) | HTML(data-value) | HTMLLINE(data-value) }

[CLNTCODEPAGE(name)]

[STATUSCODE(data-value)]

[STATUSTEXT(data-area) LENGTH(data-value)]
**Service:**

Web

**Options:**

**DOCTOKEN(name)**
Specifies the 16-byte name of the document to send to the web browser client. This comes from the document commands.

**CONTENTTYPE(data-area)**
Specifies a CONTENTTYPE header for the output document.

**CONTENTTYPELENGTH(data-value)**
Specifies the length of the CONTENTTYPE data.

**TEXT(data-value)**
Specifies text to send to the web browser client. HTML special characters are automatically escaped, e.g. `<` to `&lt;`.

**TEXTLINE(data-value)**
Specifies text line to send to the web browser client; a `<br>` is automatically appended. HTML special characters are automatically escaped, e.g. `<` to `&lt;`.

**HTML(data-value)**
Specifies HTML to send to the web browser client. The content is passed through directly to the client.

**HTMLLINE(data-value)**
Specifies HTML line to send to the web browser client; a `<br>` is automatically appended. The content is passed through directly to the client.

**CLNTCODEPAGE(name)**
Specifies the client codepage.

**STATUSCODE(data-value)**
Specify the HTTP status code; must be valid HTTP status code.

**STATUSTEXT(data-area)**
Specify text describing the status code.

**LENGTH(data-value)**
Specify length of STATUSTEXT.

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The command is a not a Web application.</td>
</tr>
</tbody>
</table>
**WEB STARTBROWSE COOKIE**

**Syntax:**

WEB STARTBROWSE COOKIE

**Service:**

Web

**Options:**

None

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

**WEB STARTBROWSE FORMFIELD**

Start browsing form fields.

**Syntax:**

WEB STARTBROWSE FORMFIELD

**Service:**

Web

**Options:**

None
Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

WEB STARTBROWSE HTTPHEADER

Start browsing HTTP headers.

Syntax:

WEB STARTBROWSE HTTPHEADER

Service:

Web

Options:

None

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is a not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

WEB WRITE COOKIE

Set an HTTP cookie.

A cookie is stored on the client machine, where it may not be stored in a secure manner. So never store secure information within a cookie; rather, use the cookie as a reference to another secure source. Also, supported cookie sizes may be very small, so do not rely on large quantities of information being stored.

Syntax:

WEB WRITE

  COOKIE(data-area) NAMELENGTH(data-value)
  VALUE(data-area) VALUELENGTH(data-value)
COMMENT(data-area) COMMENTLENGTH(data-area)
DOMAIN(data-area) DOMAINLENGTH(data-area)
MAXAGE(data-area)
PATH(data-area) PATHLENGTH(data-area)
[SECURE(cvda) | SECURE | NOSECURE]
VERSION(data-area)

Service:
Web

Options:

COOKIE(data-area)
Specifies the HTTP cookie name.
The name is an RFC 2109 name, so it may contain only ASCII alphanumeric characters, not white space, commas, or semicolons, and it may not begin with a dollar sign.

NAMELENGTH(data-value)
Specifies the length of the HTTP cookie name.

VALUE(data-area)
Specifies the value of the HTTP cookie.
With Version 0 cookies, the value should not contain white space, brackets, parentheses, equals signs, commas, double quotes, slashes, question marks, at signs, colons or semicolons; empty values may behave differently in different browsers.

VALUELENGTH(data-value)
Specifies the length of the value.

COMMENT(data-area)
Specifies the cookie's comment. The comment describes the cookie, but it is not used for programmatic purposes.

COMMENTLENGTH(data-area)
Specifies the length of the COMMENT buffer.

DOMAIN(data-area)
Specifies the cookie's domain.

DOMAINLENGTH(data-area)
Specifies the length of the DOMAIN buffer.

MAXAGE(data-area)
Specifies the maximum age of the cookie in seconds. -1 indicates until the browser is shutdown.

**PATH(data-area)**

Specifies the path of the cookie.

**PATHLENGTH(data-area)**

Specifies the length of the PATH buffer, retrieve the length of the retrieved path.

**SECURE(data-area)**

Specifies SECURE or NOSECURE, reflecting whether the cookie may be sent over an insecure channel.

**VERSION(data-area)**

Specifies the version of the cookie. Version 0 is the original Netscape specification, Version 1 complies with RFC 2109. RFC 2109 is still considered experimental, so use it only for experimentation, not for production sites.

**Conditions:**

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>The command is a not a Web application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

**WEB WRITE HTTPHEADER**

Set an HTTP header.

**Syntax:**

WEB WRITE

HTTPHEADER(data-area) NAMELENGTH(data-value)

VALUE(data-area) VALUELENGTH(data-value)

ACCUMULATE

**Service:**

Web

**Options:**

HTTPHEADER(data-area)

Specifies the HTTP header name.
NAMELENGTH(data-value)
Specifies the length of the HTTP header name.

VALUE(data-area)
Specifies the value of the HTTP header.

VALUELENGTH(data-value)
Specifies the length of the value.

ACCUMULATE
Add the header rather than replacing it if already present.

Conditions:

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The command is not a Web application.</td>
</tr>
<tr>
<td>3</td>
<td>There is no HTTP request.</td>
</tr>
</tbody>
</table>

WRITE

Write a record to a file.

Syntax:

WRITE
{FILE(data-value) | DATASET(data-value)}
[MASSINSERT]
FROM(data-area)
RIDFLD(data-area)
[KEYLENGTH(data-value)]
[SYSID(data-value) LENGTH(data-area) | LENGTH(data-area)]
[RBA | RRN]
[NOSUSPEND]

Service:

File Control

Setup:

file.<filename>=jdbc:< xml_file_descriptor.xml>[;<sql_connection_name>]
The xml_file_descriptor is an XML file described in the file setup, containing information relating the database columns to the record positions. It must be placed in the resources folder. Placing a $XFD FILE="filename.xml" command before a record in the data division will generate a starting template in the listing folder. The sql_connection_name is 'file' by default.

\[\text{sql.<sql_connection_name>.datasource=<jndi_datasource_name>}\]

The file service uses the SQL connection named 'file' by default, but it may be overridden in the file setup. It must be setup in the same manner as all SQL connections.

**Options:**

**DATASET(data-value)**
Specifies the filename the same as FILE, but FILE is preferred.

**FILE(data-value)**
Specifies the filename. This filename is referred to in the setup via file.<filename>. For example, FILE('abc') would refer to setup entry 'file.abc=jdbc:abc.xml'.

**FROM(data-area)**
Specifies the data record to be written.

**KEYLENGTH(data-value)**
Specifies the length of the record ID key field.

**LENGTH(data-area)**
Specifies the length of the from field.

**MASSINSERT**
Specifies that this write is part of a mass insert of data, where each WRITE contains MASSINSERT.

**NOSUSPEND**
The read does not suspend on locks.

**RBA**
Specifies that the key is by relative byte address.

**RIDFLD(data-area)**
Specifies the record ID field, the key used to search for the record. This data item is mapped to the column marked with an ridfld attribute in the XML file descriptor.

**RRN**
Specifies that the key is by relative record number.

**SYSID(data-value)**
Specifies the system ID for the file. This should be done instead by setting the appropriate SQL file connection.

**Conditions:**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILENOTFOUND</td>
<td>The file itself could not be found.</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>A logical error occurred.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error occurred, such as an invalid SQL query.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length of the record read exceeds the given INTO buffer's LENGTH.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The file is not open.</td>
</tr>
<tr>
<td>RECORDBUSY</td>
<td>The record was locked.</td>
</tr>
</tbody>
</table>

**WRITE JOURNALNAME**

Create a journal record and write it to a journal destination.

**Syntax:**

```
WRITE JOURNALNAME(data-value)
   JTYPEID(data-value)
   FROM(data-area) [FLENGTH(data-value)]
   REQID(data-area)
   PREFIX(data-area) PFXLENG(data-value)
   WAIT
   NOSUSPEND
```

**Service:**

Journal

**Setup:**

```
journal.<name>=<journal_protocol_reference>
```

**Options:**

```
JOURNALNAME
JTYPEID
```
FROM
LENGTH
REQID
PREFIX
PFXLENG
WAIT
NOSUSPEND

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>The command is not valid for the environment.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error has occurred in writing the journal record. This occurs when writing, flushing or syncing a file.</td>
</tr>
<tr>
<td>JIDERR</td>
<td>The named journal is not available. This condition occurs when there is no journal setup for the named journal, or the named journal's protocol is invalid.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The journal record size is too large.</td>
</tr>
<tr>
<td>NOJBUFSP</td>
<td>The journal has no more space. This may also occur if a wait is interrupted.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized to write to this journal. This may occur when opening a file where not allowed.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The journal is not open or cannot be opened.</td>
</tr>
</tbody>
</table>

WRITE JOURNALNUM

This command has been deprecated in favor of WRITE JOURNAL.

Syntax:

```
JOURNAL
  {JFILEID(data-value) | JOURNALNUM(data-value)}
  JTYPEID(data-value)
  FROM(data-area) LENGTH(data-value)
  PREFIX(data-value) PFXLENG(data-value)
  REQID(data-area)
  WAIT
  STARTIO
  NOSUSPEND
```
Service:

Journal

Setup:

`journal.DFHJnn=journal_protocol_reference`

Options:

**JFILEID or JOURNALNUM**

Binary halfword indicating journal from 1 through 99. The corresponding journal name is DFHJnn, where nn is the two-digit journal number.

**JTYPEID**

Two-character code.

**FROM**

Journal record data.

**LENGTH**

Length of journal record data.

**PREFIX**

User prefix data.

**PFXLENG**

User prefix length.

**REQID**

The transaction system returns a value in this field unique to the task and journal record.

**WAIT**

Journal record is written synchronously.

**STARTIO**

Journal record should be written immediately.

**NOSUSPEND**

If insufficient journal space, return immediately.

Conditions:

<table>
<thead>
<tr>
<th>IOERR</th>
<th>An input/output error has occurred in writing the journal record. This occurs when writing, flushing or syncing a file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIDERR</td>
<td>The named journal is not available. This condition</td>
</tr>
</tbody>
</table>
### Write Journalname

Create a journal record and write it to a journal destination.

**Syntax:**

```cobol
WRITE JOURNALNAME(data-value)
  JTYPEID(data-value)
  FROM(data-area) [FLENGTH(data-value)]
  REQID(data-area)
  PREFIX(data-area) PFXLENG(data-value)
  WAIT
  NOSUSPEND
```

**Service:**

- Journal

**Setup:**

```cobol
journal.name=journal_protocol_reference
```

**Options:**

- **JOURNALNAME**
- **JTYPEID**
- **FROM**
- **FLENGTH**
- **REQID**
- **PREFIX**
- **PFXLENG**
- **WAIT**
NOSUSPEND

Conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVREQ</td>
<td>The command is not valid for the environment.</td>
</tr>
<tr>
<td>IOERR</td>
<td>An input/output error has occurred in writing the journal record. This occurs when writing, flushing or syncing a file.</td>
</tr>
<tr>
<td>JIDERR</td>
<td>The named journal is not available. This condition occurs when there is no journal setup for the named journal, or the named journal's protocol is invalid.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The journal record size is too large.</td>
</tr>
<tr>
<td>NOJBUFSP</td>
<td>The journal has no more space. This may also occur if a wait is interrupted.</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>Not authorized to write to this journal. This may occur when opening a file where not allowed.</td>
</tr>
<tr>
<td>NOTOPEN</td>
<td>The journal is not open or cannot be opened.</td>
</tr>
</tbody>
</table>

WRITE OPERATOR

Write a message to the system operator. The system operator is represented by a journal.

Syntax:

```
WRITE OPERATOR
   TEXT(data-value) [TEXTLENGTH(data-value)]
   [ROUTECODES(data-value) NUMROUTES(data-value)]
   [EVENTUAL | ACTION(cvda) | CRITICAL | IMMEDIATE | operator-reply]
```

where operator-reply is:
```
   REPLY(data-area) MAXLENGTH(data-value)
   [REPLYLENGTH(data-area)]
   [TIMEOUT(data-value)]
```
Service:

Operator

Setup:

**operator.route=journal_name**

The operator destination is a journal name, which in turn is specified through journal.name=journal_protocol. The referent of the journal_protocol_reference is operator.route, by default, which may be modified to operator_route in some protocols. So, if operator.2 points to a WebLogic journal, then the WebLogic logging catalog will be 'operator_2'.

Options:

**ACTION(cvda)**

The cvda must be one of the following values, or the name of the cvda may be provided directly.

**EVENTUAL**

The EVENTUAL message is intended for low-priority operator messages.

The journal entry is written using JTYPEID('!E'), PREFIX( "Operator, Eventual Action Required"). If selected by default rather than explicitly, the PREFIX("Operator") is used instead.

**IMMEDIATE**

The IMMEDIATE message is intended for medium-priority operator messages.

The journal entry is written using JTYPEID('!!'), PREFIX("Operator, Immediate Action Required").

**CRITICAL**

The CRITICAL message is intended for high-priority operator messages.

The journal entry is written using JTYPEID('!C'), PREFIX("Operator, Critical Action Required").

**MAXLENGTH(data-value)**

Specifies the maximum length of the reply area.

**NUMROUTES(data-value)**

Specifies the length of the routing table.

**REPLY(data-area)**

This data-area is filled with the operator reply. Note that many operator protocols do not support a REPLY and will issue an EXPIRED condition under all circumstances.
The journal entry is written using JTYPEID('!R'), PREFIX("Operator, Reply Required").

**REPLYLENGTH(data-area)**

Returns the actual length of the operator reply.

**ROUTECODES(data-value)**

Contains NUMROUTES bytes, each of which is a binary value 0..255. The default route if unspecified is 2. For compatibility with other implementations, restrict the route to a binary value 1..28.

**TEXT(data-value)**

Contains the text sent to the operator destination.

**TEXTLENGTH(data-value)**

Specifies the length of the TEXT sent to the operator destination.

**Conditions:**

The following conditions are allocated to WRITE OPERATOR, but individual journal protocols may produce additional conditions. See the journal protocol for more information on conditions produced.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPIRED</td>
<td>RESP2</td>
</tr>
<tr>
<td>7</td>
<td>TIMEOUT has occurred</td>
</tr>
<tr>
<td>100</td>
<td>REPLY is not supported for the protocol.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>RESP2</td>
</tr>
<tr>
<td>1</td>
<td>TEXTLENGTH is invalid.</td>
</tr>
<tr>
<td>2</td>
<td>NUMROUTES is invalid.</td>
</tr>
<tr>
<td>3</td>
<td>ROUTECODES is invalid.</td>
</tr>
<tr>
<td>4</td>
<td>MAXLENGTH is invalid.</td>
</tr>
<tr>
<td>5</td>
<td>TIMEOUT is invalid.</td>
</tr>
<tr>
<td>6</td>
<td>ACTION is invalid.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>RESP2</td>
</tr>
<tr>
<td>8</td>
<td>Reply was truncated because it is longer than MAXLENGTH.</td>
</tr>
</tbody>
</table>

**WRITEQ TD**

Write data to a transient data queue.
Syntax:

WRITEQ TD QUEUE(name)
    FROM(data-area)
    [LENGTH(data-value)]
    [SYSID(name)]

Service:

Transient Data Control

Setup:

queue[.sysid_value].name=uri
    [sysid.name=sysid_value]

Options:

QUEUE(name)
    Specifies the name of the transient data queue. The queue must be defined.

FROM(data-area)
    Specifies the data area contents to send to the queue.

LENGTH(data-value)
    Specifies the length of the data area contents. The default is the length of data-area.

SYSID(name)
    Specifies the system ID. This is used to determine which queue definition is used.

Conditions:

<table>
<thead>
<tr>
<th>SYSIDERR</th>
<th>An unknown sysid is being used.</th>
</tr>
</thead>
<tbody>
<tr>
<td>QIDERR</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>There is no such queue definition.</td>
</tr>
<tr>
<td>2</td>
<td>The queue could not be obtained.</td>
</tr>
<tr>
<td>INVREQ</td>
<td>The queue producer could not be created, for reasons such as the queue being for input only.</td>
</tr>
<tr>
<td>IOERR</td>
<td>The message send to the queue failed.</td>
</tr>
</tbody>
</table>
WRITEQ TS

Write data to a temporary storage data queue.

Syntax:

```
WRITEQ TS {QUEUE(name) | QNAME(name)}
   FROM(data-area)
   [LENGTH(data-value)]
   [SYSID(name)]
   [AUXILIARY | MAIN]
   [NOSUSPEND]
   [NUMITEMS(data-area) | {ITEM(data-area) [REWRITE]}]
```

Service:

Temporary Storage Control

Setup:

```
queue[.sysid_value].name[.main | .auxiliary]=uri
[sysid.name=sysid_value]
```

Options:

- **QUEUE(name)**
  Specifies the name of the transient data queue. The queue must be defined.
- **QNAME(name)**
  See QUEUE.
- **FROM(data-area)**
  Specifies the data area contents to send to the queue.
- **LENGTH(data-value)**
  Specifies the length of the data area contents. The default is the length of data-area. The length must be between 1 and 32763.
- **SYSID(name)**
  Specifies the system ID. This is used to determine which queue definition is used.
- **NUMITEMS(data-area)**
Returns the number of items in the queue.

**ITEM(data-area)**

If REWRITE is not specified, returns the number of items in the queue. This is for compatibility reasons, NUMITEMS is preferred for this case.

If REWRITE is specified, specifies the item number to rewrite.

**REWRITE**

Rewrite a particular item number, given by ITEM, within the queue.

**NOSUSPEND**

Raise NOSPACE rather than suspending if there is no remaining room in the queue.

**MAIN**

Requests that the queue use main storage. It does so by selecting the queue reference ending in .main, if available.

**AUXILIARY**

Requests that the queue use auxiliary (disk) storage. It does so by selecting the queue reference ending in .auxiliary, if available.

**Conditions**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIDERR</td>
<td>An unknown sysid is being used.</td>
</tr>
<tr>
<td>QIDERR</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>There is no such queue definition.</td>
</tr>
<tr>
<td>2</td>
<td>The queue could not be obtained.</td>
</tr>
<tr>
<td>LENGERR</td>
<td>The length is invalid, it must be between 1 and 32763.</td>
</tr>
<tr>
<td>NOSPACE</td>
<td>Insufficient space is available to write, and NOSUSPEND was not coded.</td>
</tr>
</tbody>
</table>

**XCTL**

Transfer program control. Link to another program, after ending the current program.

**Syntax:**

XCTL PROGRAM(name)

[COMMAREA(data-area) [LENGTH(data-area)]]

[INPUTMSG(data-area) [INPUTMSGLEN(data-value)]]
**Service:**

Program Control

**Setup:**

\[\text{pct.name} = \text{program_uri}\]
program_name must be a valid program reference, either a classname or URI.

\[\text{xlt.name} = \text{program_name}\]
program_name must be a valid name in the program table.

**Options:**

\[\text{PROGRAM(name)}\]
The program name to be loaded.

\[\text{COMMAREA(data-area)}\]
The COMMAREA to pass to the linked program.

\[\text{LENGTH(data-area)}\]
The length of the COMMAREA.

\[\text{INPUTMSG(data-area)}\]
Specifies data to be received by the other program's RECEIVE.

\[\text{INPUTMSGLEN(data-value)}\]
Specifies the length of the INPUTMSG.

**Conditions:**

<table>
<thead>
<tr>
<th>LENGERR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>COMMAREA length not between 0 and 32767.</td>
</tr>
<tr>
<td>27</td>
<td>INPUTMSGLEN not between 0 and 32767</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PGMIDERR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The program name has no table entry.</td>
</tr>
<tr>
<td>3</td>
<td>The program could not be loaded.</td>
</tr>
</tbody>
</table>
Appendix DFHVALUE(name)

Access to special constants is provided in source code by coding DFHVALUE(name), where name is one of the following.

DFHVALUE constants

<table>
<thead>
<tr>
<th>Num</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IGNORE</td>
</tr>
<tr>
<td>1</td>
<td>NOTAPPLIC</td>
</tr>
<tr>
<td>2</td>
<td>BDAM</td>
</tr>
<tr>
<td>3</td>
<td>VSAM</td>
</tr>
<tr>
<td>3</td>
<td>SFS</td>
</tr>
<tr>
<td>4</td>
<td>REMOTE</td>
</tr>
<tr>
<td>5</td>
<td>ESDS</td>
</tr>
<tr>
<td>6</td>
<td>KSDS</td>
</tr>
<tr>
<td>7</td>
<td>RRDS</td>
</tr>
<tr>
<td>8</td>
<td>KEYED</td>
</tr>
<tr>
<td>9</td>
<td>NOTKEYED</td>
</tr>
<tr>
<td>10</td>
<td>BASE</td>
</tr>
<tr>
<td>11</td>
<td>PATH</td>
</tr>
<tr>
<td>12</td>
<td>FIXED</td>
</tr>
<tr>
<td>13</td>
<td>VARIABLE</td>
</tr>
<tr>
<td>14</td>
<td>UNDEFINED</td>
</tr>
<tr>
<td>15</td>
<td>NOTSUPPORTED</td>
</tr>
<tr>
<td>16</td>
<td>BLOCKED</td>
</tr>
<tr>
<td>17</td>
<td>UNBLOCKED</td>
</tr>
<tr>
<td>18</td>
<td>OPEN</td>
</tr>
<tr>
<td>19</td>
<td>CLOSED</td>
</tr>
<tr>
<td>20</td>
<td>OPENING</td>
</tr>
<tr>
<td>21</td>
<td>CLOSING</td>
</tr>
<tr>
<td>22</td>
<td>CLOSEREQUEST</td>
</tr>
<tr>
<td>23</td>
<td>ENABLED</td>
</tr>
<tr>
<td>24</td>
<td>DISABLED</td>
</tr>
<tr>
<td>25</td>
<td>DISABLING</td>
</tr>
<tr>
<td>26</td>
<td>OLD</td>
</tr>
<tr>
<td>27</td>
<td>SHARE</td>
</tr>
<tr>
<td>28</td>
<td>NEW</td>
</tr>
<tr>
<td>29</td>
<td>RECOVERABLE</td>
</tr>
<tr>
<td>30</td>
<td>NOTRECOVABLE</td>
</tr>
<tr>
<td>30</td>
<td>NOTRECOVERABLE</td>
</tr>
<tr>
<td>31</td>
<td>EMPTYREQ</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>-------------------</td>
</tr>
<tr>
<td>32</td>
<td>NOEMPTYREQ</td>
</tr>
<tr>
<td>33</td>
<td>UNENABLED</td>
</tr>
<tr>
<td>34</td>
<td>UNENABLING</td>
</tr>
<tr>
<td>35</td>
<td>READABLE</td>
</tr>
<tr>
<td>36</td>
<td>NOTREADABLE</td>
</tr>
<tr>
<td>37</td>
<td>UPDATABLE</td>
</tr>
<tr>
<td>38</td>
<td>NOTUPDATABLE</td>
</tr>
<tr>
<td>39</td>
<td>BROWSABLE</td>
</tr>
<tr>
<td>40</td>
<td>NOTBROWSABLE</td>
</tr>
<tr>
<td>41</td>
<td>ADDABLE</td>
</tr>
<tr>
<td>42</td>
<td>NOTADDABLE</td>
</tr>
<tr>
<td>43</td>
<td>DELETABLE</td>
</tr>
<tr>
<td>44</td>
<td>NOTDELETABLE</td>
</tr>
<tr>
<td>45</td>
<td>HEX</td>
</tr>
<tr>
<td>46</td>
<td>DEC</td>
</tr>
<tr>
<td>47</td>
<td>BLK</td>
</tr>
<tr>
<td>48</td>
<td>EXCTL</td>
</tr>
<tr>
<td>49</td>
<td>NOEXCTL</td>
</tr>
<tr>
<td>52</td>
<td>ALTERABLE</td>
</tr>
<tr>
<td>53</td>
<td>NOTALTERABLE</td>
</tr>
<tr>
<td>54</td>
<td>LOG</td>
</tr>
<tr>
<td>55</td>
<td>NOLOG</td>
</tr>
<tr>
<td>56</td>
<td>CTRLABLE</td>
</tr>
<tr>
<td>57</td>
<td>NOTCTRLABLE</td>
</tr>
<tr>
<td>60</td>
<td>VTAM</td>
</tr>
<tr>
<td>61</td>
<td>BSAM</td>
</tr>
<tr>
<td>62</td>
<td>BTAM</td>
</tr>
<tr>
<td>62</td>
<td>BTAM-ES</td>
</tr>
<tr>
<td>63</td>
<td>BGAM</td>
</tr>
<tr>
<td>64</td>
<td>TCAM</td>
</tr>
<tr>
<td>65</td>
<td>TCAMSNA</td>
</tr>
<tr>
<td>66</td>
<td>CONSOLE</td>
</tr>
<tr>
<td>67</td>
<td>CREATE</td>
</tr>
<tr>
<td>68</td>
<td>NOCREATE</td>
</tr>
<tr>
<td>69</td>
<td>ACQUIRED</td>
</tr>
<tr>
<td>70</td>
<td>RELEASED</td>
</tr>
<tr>
<td>71</td>
<td>ACQUIRING</td>
</tr>
<tr>
<td>72</td>
<td>COLDACQ</td>
</tr>
<tr>
<td>73</td>
<td>INSERVICE</td>
</tr>
<tr>
<td>74</td>
<td>OUTSERVICE</td>
</tr>
<tr>
<td>75</td>
<td>ATI</td>
</tr>
<tr>
<td>76</td>
<td>NOATI</td>
</tr>
<tr>
<td>77</td>
<td>TTI</td>
</tr>
<tr>
<td>78</td>
<td>NOTTI</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------</td>
</tr>
<tr>
<td>79</td>
<td>PAGEABLE</td>
</tr>
<tr>
<td>80</td>
<td>AUTOPAGEABLE</td>
</tr>
<tr>
<td>81</td>
<td>ALLOCATED</td>
</tr>
<tr>
<td>82</td>
<td>CONFFREE</td>
</tr>
<tr>
<td>83</td>
<td>CONFRECEIVE</td>
</tr>
<tr>
<td>84</td>
<td>CONFSEND</td>
</tr>
<tr>
<td>85</td>
<td>FREE</td>
</tr>
<tr>
<td>86</td>
<td>PENDFREE</td>
</tr>
<tr>
<td>87</td>
<td>PENDRECEIVE</td>
</tr>
<tr>
<td>88</td>
<td>RECEIVE</td>
</tr>
<tr>
<td>89</td>
<td>ROLLBACK</td>
</tr>
<tr>
<td>90</td>
<td>SEND</td>
</tr>
<tr>
<td>91</td>
<td>SYNCFREE</td>
</tr>
<tr>
<td>92</td>
<td>SYNCRECEIVE</td>
</tr>
<tr>
<td>93</td>
<td>SYNCSEND</td>
</tr>
<tr>
<td>94</td>
<td>FREEING</td>
</tr>
<tr>
<td>95</td>
<td>AVAILABLE</td>
</tr>
<tr>
<td>96</td>
<td>OBTAINING</td>
</tr>
<tr>
<td>100</td>
<td>NOTTABLE</td>
</tr>
<tr>
<td>101</td>
<td>CICSTABLE</td>
</tr>
<tr>
<td>102</td>
<td>USERTABLE</td>
</tr>
<tr>
<td>103</td>
<td>REMTABLE</td>
</tr>
<tr>
<td>110</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>111</td>
<td>TAKEOVER</td>
</tr>
<tr>
<td>121</td>
<td>IRC</td>
</tr>
<tr>
<td>122</td>
<td>INDIRECT</td>
</tr>
<tr>
<td>123</td>
<td>XM</td>
</tr>
<tr>
<td>124</td>
<td>APPC</td>
</tr>
<tr>
<td>125</td>
<td>LU61</td>
</tr>
<tr>
<td>126</td>
<td>PENDING</td>
</tr>
<tr>
<td>127</td>
<td>NOTPENDING</td>
</tr>
<tr>
<td>143</td>
<td>XOK</td>
</tr>
<tr>
<td>144</td>
<td>XNOTDONE</td>
</tr>
<tr>
<td>149</td>
<td>C</td>
</tr>
<tr>
<td>150</td>
<td>ASSEMBLER</td>
</tr>
<tr>
<td>151</td>
<td>COBOL</td>
</tr>
<tr>
<td>152</td>
<td>PL1</td>
</tr>
<tr>
<td>154</td>
<td>PROGRAM</td>
</tr>
<tr>
<td>155</td>
<td>MAP</td>
</tr>
<tr>
<td>155</td>
<td>MAPSET</td>
</tr>
<tr>
<td>156</td>
<td>PARTITIONSET</td>
</tr>
<tr>
<td>158</td>
<td>ANY</td>
</tr>
<tr>
<td>159</td>
<td>BELOW</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------</td>
</tr>
<tr>
<td>160</td>
<td>PURGEABLE</td>
</tr>
<tr>
<td>161</td>
<td>NOTPURGEABLE</td>
</tr>
<tr>
<td>162</td>
<td>OLDCOPY</td>
</tr>
<tr>
<td>163</td>
<td>HOLD</td>
</tr>
<tr>
<td>164</td>
<td>NOHOLD</td>
</tr>
<tr>
<td>165</td>
<td>LPA</td>
</tr>
<tr>
<td>166</td>
<td>NOTLPA</td>
</tr>
<tr>
<td>167</td>
<td>NEWCOPY</td>
</tr>
<tr>
<td>168</td>
<td>PHASEIN</td>
</tr>
<tr>
<td>169</td>
<td>ALLCONN</td>
</tr>
<tr>
<td>170</td>
<td>AUTOCONN</td>
</tr>
<tr>
<td>171</td>
<td>NONAUTOCONN</td>
</tr>
<tr>
<td>172</td>
<td>GOINGOUT</td>
</tr>
<tr>
<td>173</td>
<td>SHARED</td>
</tr>
<tr>
<td>174</td>
<td>PRIVATE</td>
</tr>
<tr>
<td>175</td>
<td>SPRSTRACE</td>
</tr>
<tr>
<td>175</td>
<td>SPRSPACE</td>
</tr>
<tr>
<td>176</td>
<td>STANTRACE</td>
</tr>
<tr>
<td>177</td>
<td>SPECTRACE</td>
</tr>
<tr>
<td>178</td>
<td>DYNAMIC</td>
</tr>
<tr>
<td>179</td>
<td>STATIC</td>
</tr>
<tr>
<td>180</td>
<td>STARTUP</td>
</tr>
<tr>
<td>181</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>182</td>
<td>FIRSTQUIESCE</td>
</tr>
<tr>
<td>183</td>
<td>FINALQUIESCE</td>
</tr>
<tr>
<td>184</td>
<td>SYSDUMP</td>
</tr>
<tr>
<td>185</td>
<td>NOSYSDUMP</td>
</tr>
<tr>
<td>186</td>
<td>TRANDUMP</td>
</tr>
<tr>
<td>187</td>
<td>NOTRANDUMP</td>
</tr>
<tr>
<td>188</td>
<td>SWITCH</td>
</tr>
<tr>
<td>192</td>
<td>BACKOUT</td>
</tr>
<tr>
<td>194</td>
<td>EXTSECURITY</td>
</tr>
<tr>
<td>194</td>
<td>CICSSECURITY</td>
</tr>
<tr>
<td>194</td>
<td>SECURE</td>
</tr>
<tr>
<td>195</td>
<td>TRANSACTIONSECURITY</td>
</tr>
<tr>
<td>196</td>
<td>NOSECURITY</td>
</tr>
<tr>
<td>196</td>
<td>NOSECURE</td>
</tr>
<tr>
<td>197</td>
<td>ALTERNATE</td>
</tr>
<tr>
<td>198</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>200</td>
<td>OFF</td>
</tr>
<tr>
<td>201</td>
<td>ON</td>
</tr>
<tr>
<td>202</td>
<td>RESSECNO</td>
</tr>
<tr>
<td>203</td>
<td>RESSECINT</td>
</tr>
<tr>
<td>204</td>
<td>RESSECYES</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
</tr>
<tr>
<td>204</td>
<td>RESSECEXT</td>
</tr>
<tr>
<td>205</td>
<td>CMDSECNO</td>
</tr>
<tr>
<td>207</td>
<td>CMDSECEXT</td>
</tr>
<tr>
<td>208</td>
<td>COMMIT</td>
</tr>
<tr>
<td>209</td>
<td>READBACK</td>
</tr>
<tr>
<td>210</td>
<td>EMPTY</td>
</tr>
<tr>
<td>211</td>
<td>NOTEMPTY</td>
</tr>
<tr>
<td>212</td>
<td>FULL</td>
</tr>
<tr>
<td>213</td>
<td>TERMINAL</td>
</tr>
<tr>
<td>214</td>
<td>NOTERMINAL</td>
</tr>
<tr>
<td>215</td>
<td>PHYSICAL</td>
</tr>
<tr>
<td>216</td>
<td>LOGICAL</td>
</tr>
<tr>
<td>221</td>
<td>EXTRA</td>
</tr>
<tr>
<td>222</td>
<td>INTRA</td>
</tr>
<tr>
<td>223</td>
<td>NOCTL</td>
</tr>
<tr>
<td>224</td>
<td>ASACTL</td>
</tr>
<tr>
<td>225</td>
<td>SWITCHING</td>
</tr>
<tr>
<td>226</td>
<td>INPUT</td>
</tr>
<tr>
<td>227</td>
<td>OUTPUT</td>
</tr>
<tr>
<td>228</td>
<td>DISPATCHABLE</td>
</tr>
<tr>
<td>229</td>
<td>RUNNING</td>
</tr>
<tr>
<td>231</td>
<td>SUSPENDED</td>
</tr>
<tr>
<td>233</td>
<td>TASK</td>
</tr>
<tr>
<td>234</td>
<td>TERM</td>
</tr>
<tr>
<td>235</td>
<td>DEST</td>
</tr>
<tr>
<td>236</td>
<td>PURGE</td>
</tr>
<tr>
<td>237</td>
<td>FORCEPURGE</td>
</tr>
<tr>
<td>241</td>
<td>MCHCTL</td>
</tr>
<tr>
<td>242</td>
<td>PRESETSEC</td>
</tr>
<tr>
<td>243</td>
<td>NOPRESETSEC</td>
</tr>
<tr>
<td>244</td>
<td>SIGNEDON</td>
</tr>
<tr>
<td>245</td>
<td>SINGEDOFF</td>
</tr>
<tr>
<td>246</td>
<td>UOW</td>
</tr>
<tr>
<td>247</td>
<td>AUXILIARY</td>
</tr>
<tr>
<td>248</td>
<td>MAIN</td>
</tr>
<tr>
<td>250</td>
<td>TAPE1</td>
</tr>
<tr>
<td>251</td>
<td>TAPE2</td>
</tr>
<tr>
<td>252</td>
<td>DISK1</td>
</tr>
<tr>
<td>253</td>
<td>DISK2</td>
</tr>
<tr>
<td>254</td>
<td>DISK2PAUSE</td>
</tr>
<tr>
<td>255</td>
<td>SMF</td>
</tr>
<tr>
<td>256</td>
<td>OPENINPUT</td>
</tr>
<tr>
<td>257</td>
<td>OPENOUTPUT</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------</td>
</tr>
<tr>
<td>258</td>
<td>READY</td>
</tr>
<tr>
<td>259</td>
<td>NOTREADY</td>
</tr>
<tr>
<td>260</td>
<td>CURRENT</td>
</tr>
<tr>
<td>261</td>
<td>CLOSELEAVE</td>
</tr>
<tr>
<td>262</td>
<td>AUTOARCH</td>
</tr>
<tr>
<td>263</td>
<td>NOAUTOARCH</td>
</tr>
<tr>
<td>264</td>
<td>REVERTED</td>
</tr>
<tr>
<td>265</td>
<td>ADVANCE</td>
</tr>
<tr>
<td>266</td>
<td>COLDSTART</td>
</tr>
<tr>
<td>267</td>
<td>WARMSTART</td>
</tr>
<tr>
<td>268</td>
<td>EMERGENCY</td>
</tr>
<tr>
<td>269</td>
<td>LOGTERM</td>
</tr>
<tr>
<td>270</td>
<td>FINPUT</td>
</tr>
<tr>
<td>271</td>
<td>FOUTPUT</td>
</tr>
<tr>
<td>272</td>
<td>FOPEN</td>
</tr>
<tr>
<td>273</td>
<td>FCLOSE</td>
</tr>
<tr>
<td>274</td>
<td>OK</td>
</tr>
<tr>
<td>275</td>
<td>NOWRITE</td>
</tr>
<tr>
<td>275</td>
<td>READONLY</td>
</tr>
<tr>
<td>276</td>
<td>REMOVE</td>
</tr>
<tr>
<td>277</td>
<td>RECOVERED</td>
</tr>
<tr>
<td>285</td>
<td>NOSWITCH</td>
</tr>
<tr>
<td>286</td>
<td>SWITCHNEXT</td>
</tr>
<tr>
<td>287</td>
<td>SWITCHALL</td>
</tr>
<tr>
<td>288</td>
<td>SHUTDOWN</td>
</tr>
<tr>
<td>289</td>
<td>NOSHUTDOWN</td>
</tr>
<tr>
<td>290</td>
<td>RESET</td>
</tr>
<tr>
<td>291</td>
<td>ADD</td>
</tr>
<tr>
<td>292</td>
<td>DELETE</td>
</tr>
<tr>
<td>310</td>
<td>INTSTART</td>
</tr>
<tr>
<td>311</td>
<td>INTSTOP</td>
</tr>
<tr>
<td>312</td>
<td>AUXSTART</td>
</tr>
<tr>
<td>313</td>
<td>AUXPAUSE</td>
</tr>
<tr>
<td>314</td>
<td>AUXSTOP</td>
</tr>
<tr>
<td>317</td>
<td>GTFSTART</td>
</tr>
<tr>
<td>318</td>
<td>GTFSTOP</td>
</tr>
<tr>
<td>319</td>
<td>SYSTEMON</td>
</tr>
<tr>
<td>320</td>
<td>SYSTEMOFF</td>
</tr>
<tr>
<td>321</td>
<td>USERON</td>
</tr>
<tr>
<td>322</td>
<td>USEROFF</td>
</tr>
<tr>
<td>323</td>
<td>SINGLEON</td>
</tr>
<tr>
<td>324</td>
<td>SINGLEOFF</td>
</tr>
<tr>
<td>330</td>
<td>PERF</td>
</tr>
<tr>
<td>331</td>
<td>NOPERF</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
</tr>
<tr>
<td>332</td>
<td>EXCEPT</td>
</tr>
<tr>
<td>333</td>
<td>NOEXCEPT</td>
</tr>
<tr>
<td>334</td>
<td>EVENT</td>
</tr>
<tr>
<td>335</td>
<td>NOEVENT</td>
</tr>
<tr>
<td>340</td>
<td>WAIT</td>
</tr>
<tr>
<td>341</td>
<td>NOWAIT</td>
</tr>
<tr>
<td>342</td>
<td>FORCE</td>
</tr>
<tr>
<td>349</td>
<td>CLOSEFAILED</td>
</tr>
<tr>
<td>350</td>
<td>IMMCLOSE</td>
</tr>
<tr>
<td>351</td>
<td>FORCECLOSE</td>
</tr>
<tr>
<td>352</td>
<td>IMMCLOSING</td>
</tr>
<tr>
<td>353</td>
<td>FORCECLOSING</td>
</tr>
<tr>
<td>354</td>
<td>FWDRECOVABLE</td>
</tr>
<tr>
<td>355</td>
<td>UNDETERMINED</td>
</tr>
<tr>
<td>356</td>
<td>NORMALBKOUT</td>
</tr>
<tr>
<td>357</td>
<td>FAILEDBKOUT</td>
</tr>
<tr>
<td>358</td>
<td>FAILINGBKOUT</td>
</tr>
<tr>
<td>359</td>
<td>INVALID</td>
</tr>
<tr>
<td>360</td>
<td>VALID</td>
</tr>
<tr>
<td>361</td>
<td>NOTFWDRCVBLE</td>
</tr>
<tr>
<td>362</td>
<td>EXITTRACE</td>
</tr>
<tr>
<td>363</td>
<td>NOEXITTRACE</td>
</tr>
<tr>
<td>364</td>
<td>ZCPTRACE</td>
</tr>
<tr>
<td>365</td>
<td>NOZCPTRACE</td>
</tr>
<tr>
<td>366</td>
<td>TCEXITALL</td>
</tr>
<tr>
<td>367</td>
<td>TCEXITSYSTEM</td>
</tr>
<tr>
<td>368</td>
<td>TCEXITNONE</td>
</tr>
<tr>
<td>369</td>
<td>TCEXITALLOFF</td>
</tr>
<tr>
<td>370</td>
<td>CEDF</td>
</tr>
<tr>
<td>370</td>
<td>MODEL</td>
</tr>
<tr>
<td>371</td>
<td>NOCEDF</td>
</tr>
<tr>
<td>371</td>
<td>SURROGATE</td>
</tr>
<tr>
<td>372</td>
<td>SESSION</td>
</tr>
<tr>
<td>373</td>
<td>APPCSINGLE</td>
</tr>
<tr>
<td>374</td>
<td>APPCPARALLEL</td>
</tr>
<tr>
<td>375</td>
<td>COBOLII</td>
</tr>
<tr>
<td>376</td>
<td>NOTINIT</td>
</tr>
<tr>
<td>377</td>
<td>LE370</td>
</tr>
<tr>
<td>378</td>
<td>INACTIVE</td>
</tr>
<tr>
<td>379</td>
<td>CICSDATAKEY</td>
</tr>
<tr>
<td>380</td>
<td>USERDATAKEY</td>
</tr>
<tr>
<td>381</td>
<td>CICSEXECKEY</td>
</tr>
<tr>
<td>382</td>
<td>USEREXECKEY</td>
</tr>
<tr>
<td>383</td>
<td>DPLSUBSET</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------</td>
</tr>
<tr>
<td>384</td>
<td>FULLAPI</td>
</tr>
<tr>
<td>385</td>
<td>FMHPARM</td>
</tr>
<tr>
<td>386</td>
<td>NOFMHPARM</td>
</tr>
<tr>
<td>387</td>
<td>OBOPERID</td>
</tr>
<tr>
<td>388</td>
<td>NOOBOPERID</td>
</tr>
<tr>
<td>391</td>
<td>APLKYBD</td>
</tr>
<tr>
<td>392</td>
<td>NOAPLKYBD</td>
</tr>
<tr>
<td>393</td>
<td>APLTEXT</td>
</tr>
<tr>
<td>394</td>
<td>NOAPLTEXT</td>
</tr>
<tr>
<td>395</td>
<td>AUDALARM</td>
</tr>
<tr>
<td>396</td>
<td>NOAUDALARM</td>
</tr>
<tr>
<td>397</td>
<td>BACKTRANS</td>
</tr>
<tr>
<td>398</td>
<td>NOBACKTRANS</td>
</tr>
<tr>
<td>399</td>
<td>COLOR</td>
</tr>
<tr>
<td>400</td>
<td>NOCOLOR</td>
</tr>
<tr>
<td>401</td>
<td>COPY</td>
</tr>
<tr>
<td>402</td>
<td>NOCOPY</td>
</tr>
<tr>
<td>403</td>
<td>DUALCASE</td>
</tr>
<tr>
<td>404</td>
<td>NODUALCASE</td>
</tr>
<tr>
<td>405</td>
<td>EXTENDED DDS</td>
</tr>
<tr>
<td>406</td>
<td>NOEXTENDED DDS</td>
</tr>
<tr>
<td>407</td>
<td>FORMFEED</td>
</tr>
<tr>
<td>408</td>
<td>NOFORMFEED</td>
</tr>
<tr>
<td>409</td>
<td>HFORM</td>
</tr>
<tr>
<td>410</td>
<td>NOHFORM</td>
</tr>
<tr>
<td>411</td>
<td>VFORM</td>
</tr>
<tr>
<td>412</td>
<td>NOVFORM</td>
</tr>
<tr>
<td>413</td>
<td>HILIGHT</td>
</tr>
<tr>
<td>414</td>
<td>NOHILIGHT</td>
</tr>
<tr>
<td>415</td>
<td>KATAKANA</td>
</tr>
<tr>
<td>416</td>
<td>NOKATAKANA</td>
</tr>
<tr>
<td>417</td>
<td>LIGHTPEN</td>
</tr>
<tr>
<td>418</td>
<td>NOLIGHTPEN</td>
</tr>
<tr>
<td>419</td>
<td>MSRCCONTROL</td>
</tr>
<tr>
<td>420</td>
<td>NOMSRCCONTROL</td>
</tr>
<tr>
<td>421</td>
<td>OBFORMAT</td>
</tr>
<tr>
<td>422</td>
<td>NOOBFORMAT</td>
</tr>
<tr>
<td>423</td>
<td>OUTLINE</td>
</tr>
<tr>
<td>424</td>
<td>NOOUTLINE</td>
</tr>
<tr>
<td>425</td>
<td>PARTITIONS</td>
</tr>
<tr>
<td>426</td>
<td>NOPARTITIONS</td>
</tr>
<tr>
<td>427</td>
<td>PRINTADAPT</td>
</tr>
<tr>
<td>428</td>
<td>NOPRINTADAPT</td>
</tr>
<tr>
<td>429</td>
<td>PROGSYMBOL</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------</td>
</tr>
<tr>
<td>430</td>
<td>NOPROGSYMBOL</td>
</tr>
<tr>
<td>431</td>
<td>ALLQUERY</td>
</tr>
<tr>
<td>432</td>
<td>NOQUERY</td>
</tr>
<tr>
<td>433</td>
<td>COLDQUERY</td>
</tr>
<tr>
<td>434</td>
<td>SOSI</td>
</tr>
<tr>
<td>435</td>
<td>NOSOSI</td>
</tr>
<tr>
<td>436</td>
<td>TEXTKYBD</td>
</tr>
<tr>
<td>437</td>
<td>NOTEXTKYBD</td>
</tr>
<tr>
<td>438</td>
<td>TEXTPRINT</td>
</tr>
<tr>
<td>439</td>
<td>NOTEXTPRINT</td>
</tr>
<tr>
<td>440</td>
<td>VALIDATION</td>
</tr>
<tr>
<td>441</td>
<td>NOVALIDATION</td>
</tr>
<tr>
<td>442</td>
<td>RELREQ</td>
</tr>
<tr>
<td>443</td>
<td>NORELREQ</td>
</tr>
<tr>
<td>444</td>
<td>DISCREQ</td>
</tr>
<tr>
<td>445</td>
<td>NODISCREQ</td>
</tr>
<tr>
<td>446</td>
<td>ALTPRTCOPY</td>
</tr>
<tr>
<td>447</td>
<td>NOALTPRTCOPY</td>
</tr>
<tr>
<td>448</td>
<td>PRTCOPY</td>
</tr>
<tr>
<td>449</td>
<td>NOPRTCOPY</td>
</tr>
<tr>
<td>450</td>
<td>UCTRAN</td>
</tr>
<tr>
<td>451</td>
<td>NOUCTRAN</td>
</tr>
<tr>
<td>452</td>
<td>TRANIDONLY</td>
</tr>
<tr>
<td>485</td>
<td>NEWSESSION</td>
</tr>
<tr>
<td>486</td>
<td>OLDSESSION</td>
</tr>
<tr>
<td>487</td>
<td>NOSTSN</td>
</tr>
<tr>
<td>488</td>
<td>STSNSET</td>
</tr>
<tr>
<td>489</td>
<td>STSNTEST</td>
</tr>
<tr>
<td>490</td>
<td>EB</td>
</tr>
<tr>
<td>491</td>
<td>CD</td>
</tr>
<tr>
<td>492</td>
<td>MORE</td>
</tr>
<tr>
<td>493</td>
<td>LIC</td>
</tr>
<tr>
<td>494</td>
<td>RU</td>
</tr>
<tr>
<td>495</td>
<td>NOTCDEB</td>
</tr>
<tr>
<td>496</td>
<td>NONE</td>
</tr>
<tr>
<td>497</td>
<td>DEFRESP1</td>
</tr>
<tr>
<td>498</td>
<td>DEFRESP2</td>
</tr>
<tr>
<td>499</td>
<td>DEFRESP3</td>
</tr>
<tr>
<td>500</td>
<td>NOALARM</td>
</tr>
<tr>
<td>501</td>
<td>ALARM</td>
</tr>
<tr>
<td>502</td>
<td>FMH</td>
</tr>
<tr>
<td>503</td>
<td>NOFMH</td>
</tr>
<tr>
<td>504</td>
<td>PROTECTED</td>
</tr>
<tr>
<td>505</td>
<td>UNPROTECTED</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
</tr>
<tr>
<td>506</td>
<td>MDT</td>
</tr>
<tr>
<td>507</td>
<td>NOMDT</td>
</tr>
<tr>
<td>508</td>
<td>DATA</td>
</tr>
<tr>
<td>509</td>
<td>STSN</td>
</tr>
<tr>
<td>510</td>
<td>BEGINSESSION</td>
</tr>
<tr>
<td>511</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td>512</td>
<td>INSTALLFAIL</td>
</tr>
<tr>
<td>513</td>
<td>DISCARDFAIL</td>
</tr>
<tr>
<td>514</td>
<td>SETFAIL</td>
</tr>
<tr>
<td>515</td>
<td>ACQFAIL</td>
</tr>
<tr>
<td>516</td>
<td>SESSIONLOST</td>
</tr>
<tr>
<td>517</td>
<td>SESSIONFAIL</td>
</tr>
<tr>
<td>518</td>
<td>CONVIDLE</td>
</tr>
<tr>
<td>519</td>
<td>ADDFAIL</td>
</tr>
<tr>
<td>520</td>
<td>DELETEFAIL</td>
</tr>
<tr>
<td>521</td>
<td>UNSOLDATA</td>
</tr>
<tr>
<td>522</td>
<td>NORMALRESP</td>
</tr>
<tr>
<td>523</td>
<td>EXCEPTRESP</td>
</tr>
<tr>
<td>524</td>
<td>ATTENTION</td>
</tr>
<tr>
<td>525</td>
<td>LUSTAT</td>
</tr>
<tr>
<td>526</td>
<td>CANCEL</td>
</tr>
<tr>
<td>527</td>
<td>RTR</td>
</tr>
<tr>
<td>528</td>
<td>DEFRESP1OR2</td>
</tr>
<tr>
<td>529</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>530</td>
<td>NEGATIVE</td>
</tr>
<tr>
<td>531</td>
<td>NOMSGJRNRL</td>
</tr>
<tr>
<td>532</td>
<td>INOUT</td>
</tr>
<tr>
<td>533</td>
<td>T3278M2</td>
</tr>
<tr>
<td>534</td>
<td>T3278M3</td>
</tr>
<tr>
<td>535</td>
<td>T3278M4</td>
</tr>
<tr>
<td>536</td>
<td>T3278M5</td>
</tr>
<tr>
<td>537</td>
<td>T3279M2</td>
</tr>
<tr>
<td>538</td>
<td>T3279M3</td>
</tr>
<tr>
<td>539</td>
<td>T3279M4</td>
</tr>
<tr>
<td>540</td>
<td>T3279M5</td>
</tr>
<tr>
<td>541</td>
<td>LUP</td>
</tr>
<tr>
<td>542</td>
<td>FORMATTED</td>
</tr>
<tr>
<td>543</td>
<td>DATASTREAM</td>
</tr>
<tr>
<td>544</td>
<td>LOSE</td>
</tr>
<tr>
<td>545</td>
<td>WIN</td>
</tr>
<tr>
<td>546</td>
<td>NOTINBOUND</td>
</tr>
<tr>
<td>547</td>
<td>INBOUND</td>
</tr>
<tr>
<td>549</td>
<td>RELEASING</td>
</tr>
<tr>
<td>550</td>
<td>INSTALLED</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
</tr>
<tr>
<td>551</td>
<td>NOTINSTALLED</td>
</tr>
<tr>
<td>552</td>
<td>TPS55M2</td>
</tr>
<tr>
<td>553</td>
<td>TPS55M3</td>
</tr>
<tr>
<td>554</td>
<td>TPS55M4</td>
</tr>
<tr>
<td>555</td>
<td>TPS55M5</td>
</tr>
<tr>
<td>556</td>
<td>NOCONV</td>
</tr>
<tr>
<td>557</td>
<td>PENDSTSN</td>
</tr>
<tr>
<td>558</td>
<td>PENDBEGIN</td>
</tr>
<tr>
<td>559</td>
<td>APPLICATION</td>
</tr>
<tr>
<td>560</td>
<td>PENDDATA</td>
</tr>
<tr>
<td>561</td>
<td>PENDSTART</td>
</tr>
<tr>
<td>562</td>
<td>PENDRELEASE</td>
</tr>
<tr>
<td>563</td>
<td>RELEASE</td>
</tr>
<tr>
<td>564</td>
<td>PENDUNSOL</td>
</tr>
<tr>
<td>565</td>
<td>PENDPASS</td>
</tr>
<tr>
<td>600</td>
<td>CONVERSE</td>
</tr>
<tr>
<td>601</td>
<td>NOCONVERSE</td>
</tr>
<tr>
<td>602</td>
<td>SYNCPOINT</td>
</tr>
<tr>
<td>603</td>
<td>NOSYNCPOINT</td>
</tr>
<tr>
<td>604</td>
<td>GMT</td>
</tr>
<tr>
<td>605</td>
<td>LOCAL</td>
</tr>
<tr>
<td>606</td>
<td>FORMATEDF</td>
</tr>
<tr>
<td>607</td>
<td>NOFORMATEDF</td>
</tr>
<tr>
<td>608</td>
<td>NOTASKSTART</td>
</tr>
<tr>
<td>609</td>
<td>STARTED</td>
</tr>
<tr>
<td>610</td>
<td>STOPPED</td>
</tr>
<tr>
<td>611</td>
<td>TASKSTART</td>
</tr>
<tr>
<td>612</td>
<td>BUSY</td>
</tr>
<tr>
<td>613</td>
<td>NOTBUSY</td>
</tr>
<tr>
<td>614</td>
<td>SCS</td>
</tr>
<tr>
<td>615</td>
<td>DS3270</td>
</tr>
<tr>
<td>616</td>
<td>ASCII7</td>
</tr>
<tr>
<td>617</td>
<td>ASCII8</td>
</tr>
<tr>
<td>618</td>
<td>AUTOSTART</td>
</tr>
<tr>
<td>620</td>
<td>INDOUBT</td>
</tr>
<tr>
<td>621</td>
<td>INFLIGHT</td>
</tr>
<tr>
<td>622</td>
<td>WAITFORGET</td>
</tr>
<tr>
<td>623</td>
<td>CONTROLSHUT</td>
</tr>
<tr>
<td>624</td>
<td>CANCELLED</td>
</tr>
<tr>
<td>625</td>
<td>FIRSTINIT</td>
</tr>
<tr>
<td>626</td>
<td>SECONDDINIT</td>
</tr>
<tr>
<td>627</td>
<td>THIRDDINIT</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>-----------------</td>
</tr>
<tr>
<td>628</td>
<td>INITCOMPLETE</td>
</tr>
<tr>
<td>629</td>
<td>STANDBY</td>
</tr>
<tr>
<td>630</td>
<td>AUTOACTIVE</td>
</tr>
<tr>
<td>631</td>
<td>AUTOINACTIVE</td>
</tr>
<tr>
<td>632</td>
<td>CTLGALL</td>
</tr>
<tr>
<td>633</td>
<td>CTLGMEMODIFY</td>
</tr>
<tr>
<td>634</td>
<td>CTLGIGNORE</td>
</tr>
<tr>
<td>635</td>
<td>START</td>
</tr>
<tr>
<td>636</td>
<td>POST</td>
</tr>
<tr>
<td>637</td>
<td>DELAY</td>
</tr>
<tr>
<td>638</td>
<td>ROUTE</td>
</tr>
<tr>
<td>640</td>
<td>CLEAR</td>
</tr>
<tr>
<td>641</td>
<td>NOCLEAR</td>
</tr>
<tr>
<td>642</td>
<td>USER</td>
</tr>
<tr>
<td>643</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>644</td>
<td>SHUTENABLED</td>
</tr>
<tr>
<td>645</td>
<td>SHUTDISABLED</td>
</tr>
<tr>
<td>650</td>
<td>EXCI</td>
</tr>
<tr>
<td>651</td>
<td>GENERIC</td>
</tr>
<tr>
<td>652</td>
<td>SPECIFIC</td>
</tr>
<tr>
<td>653</td>
<td>SYSCONNECT</td>
</tr>
<tr>
<td>654</td>
<td>NOSYSCONNECT</td>
</tr>
<tr>
<td>655</td>
<td>FORCECANCEL</td>
</tr>
<tr>
<td>657</td>
<td>NOISOLATE</td>
</tr>
<tr>
<td>658</td>
<td>ISOLATE</td>
</tr>
<tr>
<td>659</td>
<td>NOTDEFINED</td>
</tr>
<tr>
<td>660</td>
<td>CICS</td>
</tr>
<tr>
<td>661</td>
<td>NONCICS</td>
</tr>
<tr>
<td>663</td>
<td>SUBSPACE</td>
</tr>
<tr>
<td>664</td>
<td>BASESPACE</td>
</tr>
<tr>
<td>665</td>
<td>XCF</td>
</tr>
<tr>
<td>666</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>667</td>
<td>NOTREQUIRED</td>
</tr>
<tr>
<td>668</td>
<td>SOS</td>
</tr>
<tr>
<td>669</td>
<td>NOTSOS</td>
</tr>
<tr>
<td>670</td>
<td>REGISTERED</td>
</tr>
<tr>
<td>671</td>
<td>UNREGISTERED</td>
</tr>
<tr>
<td>672</td>
<td>UNAVAILABLE</td>
</tr>
<tr>
<td>673</td>
<td>CMDPROTECT</td>
</tr>
<tr>
<td>674</td>
<td>NOCMDPROTECT</td>
</tr>
<tr>
<td>675</td>
<td>RELATED</td>
</tr>
<tr>
<td>677</td>
<td>REGERROR</td>
</tr>
<tr>
<td>678</td>
<td>Deregistered</td>
</tr>
<tr>
<td>679</td>
<td>Deregerror</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------</td>
</tr>
<tr>
<td>680</td>
<td>REENTPROT</td>
</tr>
<tr>
<td>681</td>
<td>NOREENTPROT</td>
</tr>
<tr>
<td>682</td>
<td>SOSBELOW</td>
</tr>
<tr>
<td>683</td>
<td>SOSABOVE</td>
</tr>
<tr>
<td>684</td>
<td>DAE</td>
</tr>
<tr>
<td>685</td>
<td>NODAE</td>
</tr>
<tr>
<td>690</td>
<td>CONNECTED</td>
</tr>
<tr>
<td>691</td>
<td>NOTCONNECTED</td>
</tr>
<tr>
<td>693</td>
<td>SPI</td>
</tr>
<tr>
<td>694</td>
<td>NOSPI</td>
</tr>
<tr>
<td>700</td>
<td>NORECOVDATA</td>
</tr>
<tr>
<td>701</td>
<td>RECOVDATA</td>
</tr>
<tr>
<td>702</td>
<td>RESYNC</td>
</tr>
<tr>
<td>703</td>
<td>UNCONNECTED</td>
</tr>
<tr>
<td>706</td>
<td>IMMQUIESCED</td>
</tr>
<tr>
<td>707</td>
<td>QUIESCED</td>
</tr>
<tr>
<td>708</td>
<td>QUIESCING</td>
</tr>
<tr>
<td>709</td>
<td>UNQUIESCED</td>
</tr>
<tr>
<td>710</td>
<td>NOLOSTLOCKS</td>
</tr>
<tr>
<td>711</td>
<td>NORETAILED</td>
</tr>
<tr>
<td>712</td>
<td>RECOVERLOCKS</td>
</tr>
<tr>
<td>713</td>
<td>REMLOSTLOCKS</td>
</tr>
<tr>
<td>714</td>
<td>RESETLOCKS</td>
</tr>
<tr>
<td>715</td>
<td>RETAINED</td>
</tr>
<tr>
<td>716</td>
<td>RETRY</td>
</tr>
<tr>
<td>720</td>
<td>RLS</td>
</tr>
<tr>
<td>721</td>
<td>NOTRLS</td>
</tr>
<tr>
<td>722</td>
<td>UNCOMMITTED</td>
</tr>
<tr>
<td>723</td>
<td>CONSISTENT</td>
</tr>
<tr>
<td>724</td>
<td>REPEATABLE</td>
</tr>
<tr>
<td>730</td>
<td>RLSACTIVE</td>
</tr>
<tr>
<td>731</td>
<td>RLsinactive</td>
</tr>
<tr>
<td>732</td>
<td>VRRDS</td>
</tr>
<tr>
<td>740</td>
<td>REMSESSION</td>
</tr>
<tr>
<td>751</td>
<td>EXECENQ</td>
</tr>
<tr>
<td>752</td>
<td>EXECENQADDR</td>
</tr>
<tr>
<td>753</td>
<td>OWNER</td>
</tr>
<tr>
<td>754</td>
<td>WAITER</td>
</tr>
<tr>
<td>755</td>
<td>CONNECTION</td>
</tr>
<tr>
<td>756</td>
<td>DATASET</td>
</tr>
<tr>
<td>757</td>
<td>HEURCOMMIT</td>
</tr>
<tr>
<td>758</td>
<td>HEURBACKOUT</td>
</tr>
<tr>
<td>761</td>
<td>RLSSERVER</td>
</tr>
<tr>
<td>762</td>
<td>SHUNTED</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
</tr>
<tr>
<td>763</td>
<td>WAITCOMMIT</td>
</tr>
<tr>
<td>765</td>
<td>WAITING</td>
</tr>
<tr>
<td>766</td>
<td>WAITRMI</td>
</tr>
<tr>
<td>767</td>
<td>TDQ</td>
</tr>
<tr>
<td>768</td>
<td>TSQ</td>
</tr>
<tr>
<td>770</td>
<td>COORDINATOR</td>
</tr>
<tr>
<td>771</td>
<td>RMI</td>
</tr>
<tr>
<td>772</td>
<td>STARTING</td>
</tr>
<tr>
<td>773</td>
<td>SUBORDINATE</td>
</tr>
<tr>
<td>774</td>
<td>NRS</td>
</tr>
<tr>
<td>780</td>
<td>MVS</td>
</tr>
<tr>
<td>781</td>
<td>DUMMY</td>
</tr>
<tr>
<td>782</td>
<td>FAILED</td>
</tr>
<tr>
<td>783</td>
<td>FLUSH</td>
</tr>
<tr>
<td>784</td>
<td>NOSYSLOG</td>
</tr>
<tr>
<td>785</td>
<td>SYSLOG</td>
</tr>
<tr>
<td>788</td>
<td>COLD</td>
</tr>
<tr>
<td>789</td>
<td>INITIAL</td>
</tr>
<tr>
<td>790</td>
<td>ENDAFFINITY</td>
</tr>
<tr>
<td>791</td>
<td>CACHE</td>
</tr>
<tr>
<td>792</td>
<td>COMMITFAIL</td>
</tr>
<tr>
<td>793</td>
<td>DATASETFULL</td>
</tr>
<tr>
<td>794</td>
<td>DEADLOCK</td>
</tr>
<tr>
<td>795</td>
<td>DELEXITERROR</td>
</tr>
<tr>
<td>796</td>
<td>INDEXRECFULL</td>
</tr>
<tr>
<td>797</td>
<td>IERROR</td>
</tr>
<tr>
<td>798</td>
<td>OPENERRORE</td>
</tr>
<tr>
<td>799</td>
<td>RLSGONE</td>
</tr>
<tr>
<td>800</td>
<td>BACKUPNONBWO</td>
</tr>
<tr>
<td>810</td>
<td>SKIP</td>
</tr>
<tr>
<td>811</td>
<td>REWIND</td>
</tr>
<tr>
<td>811</td>
<td>LEAVE</td>
</tr>
<tr>
<td>812</td>
<td>REREAD</td>
</tr>
<tr>
<td>813</td>
<td>MOD</td>
</tr>
<tr>
<td>814</td>
<td>QUEUE</td>
</tr>
<tr>
<td>815</td>
<td>REJECT</td>
</tr>
<tr>
<td>816</td>
<td>IGNORERR</td>
</tr>
<tr>
<td>820</td>
<td>UNATTEMPTED</td>
</tr>
<tr>
<td>830</td>
<td>RRREJECT</td>
</tr>
<tr>
<td>831</td>
<td>RRINDOUBT</td>
</tr>
<tr>
<td>832</td>
<td>LCKSTRUCFULL</td>
</tr>
<tr>
<td>833</td>
<td>CFTABLE</td>
</tr>
<tr>
<td>834</td>
<td>LOAD</td>
</tr>
<tr>
<td>835</td>
<td>NOLOAD</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>------</td>
<td>---------------------</td>
</tr>
<tr>
<td>836</td>
<td>CONTENTION</td>
</tr>
<tr>
<td>837</td>
<td>LOCKING</td>
</tr>
<tr>
<td>900</td>
<td>ABEND</td>
</tr>
<tr>
<td>902</td>
<td>CGROUP</td>
</tr>
<tr>
<td>903</td>
<td>CONNECT</td>
</tr>
<tr>
<td>904</td>
<td>CONNECTING</td>
</tr>
<tr>
<td>905</td>
<td>CSIGN</td>
</tr>
<tr>
<td>906</td>
<td>CTERM</td>
</tr>
<tr>
<td>907</td>
<td>CTX</td>
</tr>
<tr>
<td>908</td>
<td>COPID</td>
</tr>
<tr>
<td>909</td>
<td>CUSERID</td>
</tr>
<tr>
<td>910</td>
<td>DISCONNNING</td>
</tr>
<tr>
<td>911</td>
<td>EQUAL</td>
</tr>
<tr>
<td>912</td>
<td>EXIT</td>
</tr>
<tr>
<td>913</td>
<td>GROUP</td>
</tr>
<tr>
<td>914</td>
<td>HIGH</td>
</tr>
<tr>
<td>915</td>
<td>LOW</td>
</tr>
<tr>
<td>916</td>
<td>NOCONNECT</td>
</tr>
<tr>
<td>917</td>
<td>NOEXIT</td>
</tr>
<tr>
<td>918</td>
<td>NORELEASE</td>
</tr>
<tr>
<td>919</td>
<td>NOROLLLBACK</td>
</tr>
<tr>
<td>920</td>
<td>NOTWAIT</td>
</tr>
<tr>
<td>921</td>
<td>N906D</td>
</tr>
<tr>
<td>922</td>
<td>POOL</td>
</tr>
<tr>
<td>923</td>
<td>REBUILD</td>
</tr>
<tr>
<td>924</td>
<td>RECONNECT</td>
</tr>
<tr>
<td>925</td>
<td>SIGN</td>
</tr>
<tr>
<td>926</td>
<td>SQLCODE</td>
</tr>
<tr>
<td>927</td>
<td>TWAIT</td>
</tr>
<tr>
<td>928</td>
<td>TXID</td>
</tr>
<tr>
<td>929</td>
<td>TX</td>
</tr>
<tr>
<td>930</td>
<td>USERID</td>
</tr>
<tr>
<td>931</td>
<td>N906</td>
</tr>
<tr>
<td>932</td>
<td>TPOOL</td>
</tr>
<tr>
<td>933</td>
<td>OPI D</td>
</tr>
<tr>
<td>1000</td>
<td>NOTFIRED</td>
</tr>
<tr>
<td>1001</td>
<td>FIRED</td>
</tr>
<tr>
<td>1002</td>
<td>ACTIVITY</td>
</tr>
<tr>
<td>1003</td>
<td>COMPOSITE</td>
</tr>
<tr>
<td>1004</td>
<td>TIMER</td>
</tr>
<tr>
<td>1005</td>
<td>AND</td>
</tr>
<tr>
<td>1006</td>
<td>OR</td>
</tr>
<tr>
<td>1010</td>
<td>PROCESS</td>
</tr>
<tr>
<td>1011</td>
<td>NOUSER</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1012</td>
<td>EVENTFAIL</td>
</tr>
<tr>
<td>1013</td>
<td>FORCED</td>
</tr>
<tr>
<td>1014</td>
<td>INCOMPLETE</td>
</tr>
<tr>
<td>1015</td>
<td>UNEXPECTED</td>
</tr>
<tr>
<td>1016</td>
<td>NORMAL</td>
</tr>
<tr>
<td>1017</td>
<td>EXPIRED</td>
</tr>
<tr>
<td>1018</td>
<td>UNEXPIRED</td>
</tr>
<tr>
<td>1021</td>
<td>NOTDYNAMIC</td>
</tr>
<tr>
<td>1022</td>
<td>NOTROUTABLE</td>
</tr>
<tr>
<td>1023</td>
<td>ROUTABLE</td>
</tr>
<tr>
<td>1024</td>
<td>DORMANT</td>
</tr>
<tr>
<td>1025</td>
<td>CANCELLING</td>
</tr>
<tr>
<td>1026</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>1027</td>
<td>NOTSUSPENDED</td>
</tr>
<tr>
<td>1030</td>
<td>SSL</td>
</tr>
<tr>
<td>1031</td>
<td>NOSSL</td>
</tr>
<tr>
<td>1032</td>
<td>CLIENTAUTH</td>
</tr>
<tr>
<td>1033</td>
<td>HTTPYES</td>
</tr>
<tr>
<td>1034</td>
<td>HTTPNO</td>
</tr>
<tr>
<td>1036</td>
<td>APPEND</td>
</tr>
<tr>
<td>1037</td>
<td>NOAPPEND</td>
</tr>
<tr>
<td>1038</td>
<td>BINARY</td>
</tr>
<tr>
<td>1039</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>1050</td>
<td>QUASIRENT</td>
</tr>
<tr>
<td>1051</td>
<td>THREADSAFE</td>
</tr>
<tr>
<td>1052</td>
<td>BASEAPI</td>
</tr>
<tr>
<td>1053</td>
<td>OPENAPI</td>
</tr>
<tr>
<td>1054</td>
<td>NOFORCE</td>
</tr>
<tr>
<td>1055</td>
<td>CKOPEN</td>
</tr>
<tr>
<td>1056</td>
<td>UKOPEN</td>
</tr>
<tr>
<td>1057</td>
<td>QR</td>
</tr>
<tr>
<td>1058</td>
<td>INTERNAL</td>
</tr>
<tr>
<td>1071</td>
<td>NOAUTO</td>
</tr>
<tr>
<td>1072</td>
<td>PROGAUTO</td>
</tr>
<tr>
<td>1073</td>
<td>FULLAUTO</td>
</tr>
<tr>
<td>1080</td>
<td>JVM</td>
</tr>
<tr>
<td>1081</td>
<td>NOJVM</td>
</tr>
<tr>
<td>1082</td>
<td>DEBUG</td>
</tr>
<tr>
<td>1083</td>
<td>NODEBUG</td>
</tr>
<tr>
<td>1084</td>
<td>NONLE370</td>
</tr>
<tr>
<td>2000</td>
<td>HTTP</td>
</tr>
<tr>
<td>2001</td>
<td>NONHTTP</td>
</tr>
</tbody>
</table>
## DFHVALUE(name) Device Commands

<table>
<thead>
<tr>
<th>Num</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T7770</td>
</tr>
<tr>
<td>2</td>
<td>SYSTEM7</td>
</tr>
<tr>
<td>8</td>
<td>TCONSOLE</td>
</tr>
<tr>
<td>18</td>
<td>SEQDISK</td>
</tr>
<tr>
<td>20</td>
<td>MAGTAPE</td>
</tr>
<tr>
<td>24</td>
<td>CDRDLPRT</td>
</tr>
<tr>
<td>32</td>
<td>HARDCOPY</td>
</tr>
<tr>
<td>33</td>
<td>TWX3335</td>
</tr>
<tr>
<td>34</td>
<td>TEETYPE</td>
</tr>
<tr>
<td>36</td>
<td>T1050</td>
</tr>
<tr>
<td>40</td>
<td>T2740</td>
</tr>
<tr>
<td>42</td>
<td>T2741COR</td>
</tr>
<tr>
<td>43</td>
<td>T2741BCD</td>
</tr>
<tr>
<td>64</td>
<td>VIDEOTERM</td>
</tr>
<tr>
<td>65</td>
<td>T2260L</td>
</tr>
<tr>
<td>72</td>
<td>T2260R</td>
</tr>
<tr>
<td>74</td>
<td>T1053</td>
</tr>
<tr>
<td>76</td>
<td>T2265</td>
</tr>
<tr>
<td>80</td>
<td>TTCAM</td>
</tr>
<tr>
<td>128</td>
<td>BISYNCH</td>
</tr>
<tr>
<td>130</td>
<td>T2770</td>
</tr>
<tr>
<td>132</td>
<td>T2780</td>
</tr>
<tr>
<td>133</td>
<td>T3780</td>
</tr>
<tr>
<td>134</td>
<td>T2980</td>
</tr>
<tr>
<td>136</td>
<td>T3735</td>
</tr>
<tr>
<td>137</td>
<td>T3740</td>
</tr>
<tr>
<td>138</td>
<td>T3600BI</td>
</tr>
<tr>
<td>145</td>
<td>T3277R</td>
</tr>
<tr>
<td>146</td>
<td>T3275R</td>
</tr>
<tr>
<td>147</td>
<td>T3284R</td>
</tr>
<tr>
<td>148</td>
<td>T3286R</td>
</tr>
<tr>
<td>153</td>
<td>T3277L</td>
</tr>
<tr>
<td>155</td>
<td>T3284L</td>
</tr>
<tr>
<td>156</td>
<td>T3286L</td>
</tr>
<tr>
<td>160</td>
<td>BIPROG</td>
</tr>
<tr>
<td>161</td>
<td>SYSTEM3</td>
</tr>
<tr>
<td>164</td>
<td>SYS370</td>
</tr>
<tr>
<td>166</td>
<td>SYS7BSCA</td>
</tr>
<tr>
<td>176</td>
<td>SDLC</td>
</tr>
<tr>
<td>177</td>
<td>T3601</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>--------------</td>
</tr>
<tr>
<td>178</td>
<td>T3614</td>
</tr>
<tr>
<td>180</td>
<td>T3790</td>
</tr>
<tr>
<td>181</td>
<td>T3790UP</td>
</tr>
<tr>
<td>182</td>
<td>T3790SCSP</td>
</tr>
<tr>
<td>184</td>
<td>T3650PIPE</td>
</tr>
<tr>
<td>185</td>
<td>T3653HOST</td>
</tr>
<tr>
<td>186</td>
<td>T3650ATT</td>
</tr>
<tr>
<td>187</td>
<td>T3650USER</td>
</tr>
<tr>
<td>189</td>
<td>CONTNLU</td>
</tr>
<tr>
<td>190</td>
<td>INTACTLU</td>
</tr>
<tr>
<td>191</td>
<td>BATCHLU</td>
</tr>
<tr>
<td>192</td>
<td>LUTYPE6</td>
</tr>
<tr>
<td>193</td>
<td>LUTYPE4</td>
</tr>
<tr>
<td>208</td>
<td>RESSYS</td>
</tr>
<tr>
<td>209</td>
<td>ISCMMMCNCV</td>
</tr>
<tr>
<td>210</td>
<td>LUCMODGRP</td>
</tr>
<tr>
<td>211</td>
<td>LUCSESS</td>
</tr>
</tbody>
</table>
## Appendix DFHRESP

RESP codes are available symbolically using DFHRESP(name), such as DFHRESP(NORMAL).

<table>
<thead>
<tr>
<th>Num</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NORMAL</td>
</tr>
<tr>
<td>1</td>
<td>ERROR</td>
</tr>
<tr>
<td>2</td>
<td>RDATT</td>
</tr>
<tr>
<td>3</td>
<td>WRBRK</td>
</tr>
<tr>
<td>4</td>
<td>EOF</td>
</tr>
<tr>
<td>5</td>
<td>EODS</td>
</tr>
<tr>
<td>6</td>
<td>EOC</td>
</tr>
<tr>
<td>7</td>
<td>INBFMH</td>
</tr>
<tr>
<td>8</td>
<td>ENDINPT</td>
</tr>
<tr>
<td>9</td>
<td>NONVAL</td>
</tr>
<tr>
<td>10</td>
<td>NOSTART</td>
</tr>
<tr>
<td>11</td>
<td>TERMIDERR</td>
</tr>
<tr>
<td>12</td>
<td>FILENOTFOUND</td>
</tr>
<tr>
<td>13</td>
<td>NOTFND</td>
</tr>
<tr>
<td>14</td>
<td>DUPREC</td>
</tr>
<tr>
<td>15</td>
<td>DUPKEY</td>
</tr>
<tr>
<td>16</td>
<td>INVREQ</td>
</tr>
<tr>
<td>17</td>
<td>IOERR</td>
</tr>
<tr>
<td>18</td>
<td>NOSPACE</td>
</tr>
<tr>
<td>19</td>
<td>NOTOPEN</td>
</tr>
<tr>
<td>20</td>
<td>ENDFILE</td>
</tr>
<tr>
<td>21</td>
<td>ILLOGIC</td>
</tr>
<tr>
<td>22</td>
<td>LENGERR</td>
</tr>
<tr>
<td>23</td>
<td>QZERO</td>
</tr>
<tr>
<td>24</td>
<td>SIGNAL</td>
</tr>
<tr>
<td>25</td>
<td>QBUSY</td>
</tr>
<tr>
<td>26</td>
<td>ITEMERR</td>
</tr>
<tr>
<td>27</td>
<td>PGMIDERR</td>
</tr>
<tr>
<td>28</td>
<td>TRANSIDERR</td>
</tr>
<tr>
<td>29</td>
<td>ENDDATA</td>
</tr>
<tr>
<td>30</td>
<td>INVTSREQ</td>
</tr>
<tr>
<td>31</td>
<td>EXPIRED</td>
</tr>
<tr>
<td>32</td>
<td>RETPAGE</td>
</tr>
<tr>
<td>33</td>
<td>RTEFAIL</td>
</tr>
<tr>
<td>34</td>
<td>RTESOME</td>
</tr>
<tr>
<td>35</td>
<td>TSIOERR</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------</td>
</tr>
<tr>
<td>36</td>
<td>MAPFAIL</td>
</tr>
<tr>
<td>37</td>
<td>INVERRTERM</td>
</tr>
<tr>
<td>38</td>
<td>INVMPSZ</td>
</tr>
<tr>
<td>39</td>
<td>IGREQID</td>
</tr>
<tr>
<td>40</td>
<td>OVERFLOW</td>
</tr>
<tr>
<td>41</td>
<td>INVLDC</td>
</tr>
<tr>
<td>42</td>
<td>NOSTG</td>
</tr>
<tr>
<td>43</td>
<td>JIDERR</td>
</tr>
<tr>
<td>44</td>
<td>QIDERR</td>
</tr>
<tr>
<td>45</td>
<td>NOJBUFSP</td>
</tr>
<tr>
<td>46</td>
<td>DSSTAT</td>
</tr>
<tr>
<td>47</td>
<td>SELNERR</td>
</tr>
<tr>
<td>48</td>
<td>FUNCERR</td>
</tr>
<tr>
<td>49</td>
<td>UNEXPIN</td>
</tr>
<tr>
<td>50</td>
<td>NOPASSBKRD</td>
</tr>
<tr>
<td>51</td>
<td>NOPASSBKWR</td>
</tr>
<tr>
<td>52</td>
<td>SEGIDERR</td>
</tr>
<tr>
<td>53</td>
<td>SYSIDERR</td>
</tr>
<tr>
<td>54</td>
<td>ISCINVREQ</td>
</tr>
<tr>
<td>55</td>
<td>ENQBUSY</td>
</tr>
<tr>
<td>56</td>
<td>ENVDEFERR</td>
</tr>
<tr>
<td>57</td>
<td>IGREQCD</td>
</tr>
<tr>
<td>58</td>
<td>SESSIONERR</td>
</tr>
<tr>
<td>59</td>
<td>SYSBUSY</td>
</tr>
<tr>
<td>60</td>
<td>SESSBUSY</td>
</tr>
<tr>
<td>61</td>
<td>NOTALLOC</td>
</tr>
<tr>
<td>62</td>
<td>CBIDERR</td>
</tr>
<tr>
<td>63</td>
<td>INVEXITREQ</td>
</tr>
<tr>
<td>64</td>
<td>INVPARTNSET</td>
</tr>
<tr>
<td>65</td>
<td>INVPARTN</td>
</tr>
<tr>
<td>66</td>
<td>PARTNFAIL</td>
</tr>
<tr>
<td>69</td>
<td>USERIDERR</td>
</tr>
<tr>
<td>70</td>
<td>NOTAUTH</td>
</tr>
<tr>
<td>71</td>
<td>VOLIDERR</td>
</tr>
<tr>
<td>72</td>
<td>SUPPRESSED</td>
</tr>
<tr>
<td>80</td>
<td>NOSPOOL</td>
</tr>
<tr>
<td>81</td>
<td>TERMERR</td>
</tr>
<tr>
<td>82</td>
<td>ROLLEDBACK</td>
</tr>
<tr>
<td>83</td>
<td>END</td>
</tr>
<tr>
<td>84</td>
<td>DISABLED</td>
</tr>
<tr>
<td>85</td>
<td>ALLOCERR</td>
</tr>
<tr>
<td>86</td>
<td>STRELERR</td>
</tr>
<tr>
<td>87</td>
<td>OPENERR</td>
</tr>
<tr>
<td>88</td>
<td>SPOLBUSY</td>
</tr>
<tr>
<td>Num</td>
<td>Name</td>
</tr>
<tr>
<td>-----</td>
<td>-------------------</td>
</tr>
<tr>
<td>89</td>
<td>SPOLErr</td>
</tr>
<tr>
<td>90</td>
<td>NODEIDERR</td>
</tr>
<tr>
<td>91</td>
<td>TASKIDERR</td>
</tr>
<tr>
<td>92</td>
<td>TCIDERR</td>
</tr>
<tr>
<td>93</td>
<td>DSNNOTFOUND</td>
</tr>
<tr>
<td>94</td>
<td>LOADING</td>
</tr>
<tr>
<td>95</td>
<td>MODELIDERR</td>
</tr>
<tr>
<td>96</td>
<td>OUTDESCRERR</td>
</tr>
<tr>
<td>97</td>
<td>PARTNERIDERR</td>
</tr>
<tr>
<td>98</td>
<td>PROFILEIDERR</td>
</tr>
<tr>
<td>99</td>
<td>NETNAMERR</td>
</tr>
<tr>
<td>100</td>
<td>LOCKED</td>
</tr>
<tr>
<td>101</td>
<td>RECORDBUSY</td>
</tr>
<tr>
<td>102</td>
<td>UOWNNOTFOUND</td>
</tr>
<tr>
<td>103</td>
<td>UOWLTRNOTFOUND</td>
</tr>
<tr>
<td>104</td>
<td>LINKABEND</td>
</tr>
<tr>
<td>105</td>
<td>CHANGED</td>
</tr>
<tr>
<td>106</td>
<td>PROCESSBUSY</td>
</tr>
<tr>
<td>107</td>
<td>ACTIVITYBUSY</td>
</tr>
<tr>
<td>108</td>
<td>PROCESSERR</td>
</tr>
<tr>
<td>109</td>
<td>ACTIVITYERR</td>
</tr>
<tr>
<td>110</td>
<td>CONTAINERERR</td>
</tr>
<tr>
<td>111</td>
<td>EVENTERR</td>
</tr>
<tr>
<td>112</td>
<td>TOKENERR</td>
</tr>
<tr>
<td>113</td>
<td>NOTFINISHED</td>
</tr>
<tr>
<td>114</td>
<td>POOLERR</td>
</tr>
<tr>
<td>115</td>
<td>TIMERERR</td>
</tr>
<tr>
<td>116</td>
<td>SYMBOLERR</td>
</tr>
<tr>
<td>117</td>
<td>TEMPLATERR</td>
</tr>
</tbody>
</table>
Appendix EJB Setup Files

To setup remote EJB usage of the transaction, several .xml files may be required, including ejb-jar.xml and possibly one or more required for a particular application server. Samples of these are included.

These files are automatically generated by the IDE during deployment.

ejb-jar.xml

```xml
<ejb-jar>
  <enterprise-beans>
    <session>
      <ejb-name>LinkDispatcher</ejb-name>
      <home>com.heirloomcomputing.etp.transaction.env.ejb.LinkDispatcherHome</home>
      <remote>com.heirloomcomputing.etp.transaction.env.ejb.LinkDispatcher</remote>
      <local-home>com.heirloomcomputing.etp.transaction.env.ejb.LinkDispatcherLocalHome</local-home>
      <local-home>com.heirloomcomputing.etp.transaction.env.ejb.LinkDispatcherLocal</local-home>
      <ejb-class>com.heirloomcomputing.etp.transaction.env.ejb.LinkDispatcherEJB</ejb-class>
    </session>
    <session-type>Stateful</session-type>
    <transaction-type>Bean</transaction-type>
    <env-entry>
      <env-entry-name>xlt.ejb_transid</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>bound_program_name</env-entry-value>
    </env-entry>
    <env-entry>
      <env-entry-name>pct.bound_program_name</env-entry-name>
      <env-entry-type>java.lang.String</env-entry-type>
      <env-entry-value>program_id</env-entry-value>
    </env-entry>
  </session>
</enterprise-beans>
</ejb-jar>
```

weblogic-ejb-jar.xml

```xml
<?xml version="1.0"?>
<!DOCTYPE weblogic-ejb-jar PUBLIC
"-//BEA Systems, Inc.//DTD WebLogic 6.0.0 EJB//EN"
"http://www.bea.com/servers/wls600/dtd/weblogic-ejb-jar.dtd">

<weblogic-ejb-jar>
  <weblogic-enterprise-bean>
    <ejb-name>LinkDispatcher</ejb-name>
    <jndi-name>remote_jndi_name</jndi-name>
  </weblogic-enterprise-bean>
</weblogic-ejb-jar>
```
Appendix Unsupported Commands

The Elastic Transaction Platform supports the CICS API command set with the following exceptions:

**Distributed Processing Service:**

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECEIVE</td>
<td>ISSUE ERASEAUP</td>
<td>EXTRACT PROCESS</td>
</tr>
<tr>
<td>SEND</td>
<td>ISSUE ENDFILE</td>
<td>ISSUE ABEND</td>
</tr>
<tr>
<td>CONVERSE</td>
<td>ISSUE PRINT</td>
<td>CONNECT PROCESS</td>
</tr>
<tr>
<td>ISSUE EODS</td>
<td>ISSUE SIGNAL</td>
<td>ISSUE</td>
</tr>
<tr>
<td>ISSUE COPY</td>
<td>ALLOCATE</td>
<td>CONFIRMATION</td>
</tr>
<tr>
<td>WAIT TERMINAL</td>
<td>FREE</td>
<td>ISSUE ERROR</td>
</tr>
<tr>
<td>ISSUE LOAD</td>
<td>POINT</td>
<td>ISSUE PREPARE</td>
</tr>
<tr>
<td>WAIT SIGNAL</td>
<td>BUILD ATTACH</td>
<td>ISSUE PASS</td>
</tr>
<tr>
<td>ISSUE RESET</td>
<td>EXTRACT ATTACH</td>
<td>EXTRACT LOGONMSG</td>
</tr>
<tr>
<td>ISSUE DISCONNECT</td>
<td>EXTRACT TCT</td>
<td>EXTRACT</td>
</tr>
<tr>
<td>ISSUE ENDDOUPUT</td>
<td>WAIT CONVID</td>
<td>ATTRIBUTES</td>
</tr>
</tbody>
</table>

**Interval Control:**

- START BREXIT

**Batch Data Interchange:**

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISSUE ADD</td>
<td>ISSUE QUERY</td>
<td>ISSUE WAIT</td>
</tr>
<tr>
<td>ISSUE ERASE</td>
<td>ISSUE END</td>
<td>ISSUE SEND</td>
</tr>
<tr>
<td>ISSUE REPLACE</td>
<td>ISSUE RECEIVE</td>
<td></td>
</tr>
<tr>
<td>ISSUE ABORT</td>
<td>ISSUE NOTE</td>
<td></td>
</tr>
</tbody>
</table>

**TCP/IP:**

- EXTRACT CERTIFICATE

**Trace Control:**

- MONITOR
Task Management Service:

WAITCICS

WAIT EXTERNAL