

# **z/TPF CORO/CORO64 Driver User's Guide**

© Copyright IBM Corp. 2006

*This page intentionally left blank.*

## ZTEST CORO/C064

The CORO test driver uses and verifies the commit rollback support on the TPF system. The commit rollback support allows an application to bundle file requests into an all or nothing operation. This means that the application can request any number of file requests be 'held' until the application either decides to commit these changes or rollback these changes (which throw all the changes away). The commit rollback support will insure that even if TPF is IPLed during the commit processing, either all of the changes are completed or none of the changes are completed.

The CORO driver tests the following TPF C APIs

1. tx\_begin()
2. tx\_commit()
3. tx\_rollback()
4. tx\_suspend\_tpf()
5. tx\_resume\_tpf()

The CORO64 test driver is the 64-bit version of the CORO driver. It exhibits the same testing as the CORO driver, but the assembly programs are written and run in 64-bit mode.

### Requirements and restrictions

None.

### Format

```

>>---ZTEST---+-----+---CORO---+---START---+-----+-----><
      +- -i-+  '- CO64 -' |           | .- -Ecb-1---. | |
      '- *--'            |           | +-----+-----+
                               |           | '- -Ecb-n---' | |
                               |           | .- -Nest-3---. | |
                               |           | +-----+-----+
                               |           | '- -Nest-n---' | |
                               |           | .- -Poolid-LDP--. | |
                               |           | +-----+-----+
                               |           | '- -Poolid-pt---' | |
                               |           | .- -Fa-1-----. | |
                               |           | +-----+-----+
                               |           | '- -Fa-n-----' | |
                               |           | .- -Cretc-1--. | |
                               |           | +-----+-----+
                               |           | '- -Cretc-n--' | |
                               |           |
                               |           | +-----+-----+
                               |           | .- -Ecb-1---. | |
                               |           | +-----+-----+
                               |           | '- -Ecb-n---' | |
                               |           | .- -Nest-3---. | |
                               |           | +-----+-----+
                               |           | '- -Nest-n---' | |
                               |           | .- -Poolid-LDP--. | |
                               |           | +-----+-----+
                               |           | '- -Poolid-pt --' | |
                               |           | .- -Fa-1-----. | |
                               |           | +-----+-----+
                               |           | '- -Fa-n-----' | |
                               |           |
                               |           | +-----+-----+
                               |           | .- -Ecb-1---. | |
                               |           | +-----+-----+
                               |           | '- -Ecb-n---' | |

```

```

|               | .- -Nest-3---. | |
|               +-----+-----+ | |
|               | '- -Nest-n---' | |
|               | .- -Poolid-LDP--. | |
|               +-----+-----+ | |
|               | '- -Poolid-pt---' | |
|               | .- -PRt-N----. | |
|               +-----+-----+ | |
|               | '- -PRt-Y----' | |
+-----+-----+
+-STOp-----+
+-Reset-----+
+-Verify-----+
+-STATus-----+
+-HELP-----+
+'-?'-----+'

```

*i*

indicates the specific I-stream in which the driver will be run. If *i* is not specified, the test case(s) will be executed on the I-stream on which the command is entered.

\*

specifies the driver will be invoked on all currently defined and available I-streams.

**START**

specifies that the CORO/CORO64 driver is to be run continuously. **This command can only be issued in NORM state.**

**Alter**

changes the activity levels after the driver is started.

**Once**

specifies that this driver run only once using the options specified.

**Ecb**

sets the number of ECBs to be active at the same time. The larger the number the more commit scopes are active at the same time. The number of Ecb also equal to number of ordinals that will be used with record type QL0QL (except ordinal 0).

**Nest**

sets the number of commit scope nesting levels for each ECB to perform. This tests a commit scope inside another commit scope.

**Poolid**

sets the pool type that the ECBs use to build the database for this driver. The supported options are: SDP, LDP, 4DP, SST, LST, and 4ST.

**Fa**

sets the number of pool records to be filed in each commit scope. The larger the number the bigger each commit scope is created.

**Cretc**

sets the number of seconds how quickly the driver will redrive the commit activity. The bigger the number the slower the driver runs.

**PRt**

specifies whether the CORO/CORO64 driver displays diagnostic messages when ZTEST CORO/CO64 ONCE is specified.

**STOp**

stops continuous mode.

**Reset**

resets the this driver records. **This parameter must be issued once before using the driver or if the driver is hung. This command can be issued in 1052 state.**

**Verify**

validates that the database is consistent.

**STATUS**

displays the current status of the driver.

**HELP | ?**

displays the correct syntax of the command.

**Source code information**

The CORO driver consists of the following program segments:

**Macro File**

Header File	Description
sd0rv.mac	This macro file contains the DSECT for non-keypointable I-stream shared global (@ISSDRV) used by various drivers.

**Header Files**

Header File	Description
c_coro.h	This header file was specifically created to hold the definition of a structure used by the CORO/CORO64 driver to build the driver database.
c_sd0rv.h	This header file maps the non-keypointable I-stream shared global (@ISSDRV) used by various drivers.

**BSOs**

Module	Makefile	Segment	Description
QFN1	app_drvs.mak	qfn1.asm	Monitor segment of the CORO driver. It controls starting and stopping the driver, displays status, and maintains activity levels of the driver.
QFN2	app_drvs.mak	qfn2.asm	This is the command parser

**CSOs**

Module	Makefile	Segment	Description
QFN3	qfn3.mak	qfn3.cpp	<p>For start command:</p> <ul style="list-style-type: none"> <li>- lock root record on DE</li> <li>- add file pool address to the chain if maximum is not reached yet.</li> <li>- update the file chain, file count and checksum.</li> <li>- verify the data content, file chain and checksum.</li> <li>- unhold the root</li> </ul> <p>For verify command, verify all 200 fixed file records:</p> <ul style="list-style-type: none"> <li>- look root in core</li> <li>- verify the file count and checksum</li> <li>- release file chain and initialize the root block</li> <li>- unhold the root</li> </ul>

The CORO64 driver consists of the following program segments:

**Macro File**

Header File	Description
sd0rv.mac	This macro file contains the DSECT for non-keypointable I-stream shared global (@ISUDRV) used by various drivers.

**Header Files**

Header File	Description
c_coro.h	This header file was specifically created to hold the definition of a structure used by the CORO/CORO64 driver to build the driver database.
c_sd0rv.h	This header file maps the non-keypointable I-stream shared global (@ISUDRV) used by various drivers.

**BSOs**

Module	Makefile	Segment	Description
QMS0	app_drvs.mak	qms0.asm	Monitor segment of the CORO64 driver. It controls starting and stopping the driver, displays status, and maintains activity levels of the driver.
QMS1	app_drvs.mak	qms1.asm	This is the command parser

**CSOs**

Module	Makefile	Segment	Description
QMS2	qms2.mak	qms2.cpp	<p>For start command:</p> <ul style="list-style-type: none"> <li>- lock root record on DE</li> <li>- add file pool address to the chain if maximum is not reached yet.</li> <li>- update the file chain, file count and checksum.</li> <li>- verify the data content, file chain and checksum.</li> <li>- unhold the root</li> </ul> <p>For verify command, verify all 200 fixed file records:</p> <ul style="list-style-type: none"> <li>- look root in core</li> <li>- verify the file count and checksum</li> <li>- release file chain and initialize the root block</li> <li>- unhold the root</li> </ul>

**Linkage information**

For CORO driver:

CVZZ --> QFN2 --> QFN1 --> QFN3

For CORO64 driver:

CVZZ --> QMS1 --> QMS0 --> QMS2

**Additional information**

None.

**Examples****Examples**

The following example will run all CORO test case once on I-stream 1, with messages printed out with driver status:

```
User:    ZTEST CORO ONCE PRT-Y
System:  QFN20005I  CORO DRIVER IS RUNNING ONCE+
```

The following example will reset the CORO driver record on I-stream 2:

```
User:    ZTEST 2 CORO RESET
System:  QFN20004I  CORO DRIVER RESET BEGUN+
         QFN10010I  CORO DRIVER RESET COMPLETE+
```

The following example will verify the CORO driver database:

```
User:    ZTEST CORO VERIFY
System:  QFN20007I  CORO DRIVER IS VERIFYING:+
         QFN30001I  CORO driver database verification is completed+
```

The following example will execute the CORO64 driver in continuous mode:

```
User:    ZTEST CO64 START
System:  QMS10001I  CORO DRIVER IS STARTING+
```

The following example will display the status of the CORO64 driver:

```
User:    ZTEST CO64 STATUS
System:  QMS00006I  CORO DRIVER IS RUNNING

        REQUESTED ECBS=  1  NEST=  3  RIAT=D3D3  FA=  1  CRETC=  1
        RUNNING ECBS  =  0
        TOTAL COMMITS  =           49
        TOTAL ROLLBACKS =           49 +
```

The following example stop the CORO64 driver in continuous mode:

```
User:    ZTEST CO64 STOP
System:  QMS00003I  CORO DRIVER IS NOW STOPPING+
         QMS00007I  CORO DRIVER IS NOW STOPPED+
```

## Messages

Below is a list of the CORO driver messages:

### **QFN10001 CORO DRIVER IS ALREADY ACTIVE, START REJECTED**

**Explanation:** The operator entered ZTEST CORO START and the CORO driver is already running in continuous mode. Enter ZTEST CORO STOP to stop continuous mode.

### **QFN10002 CORO DRIVER IS NOT ACTIVE, STOP REJECTED**

**Explanation:** The operator entered ZTEST CORO STOP but the CORO driver is not running in continuous mode.

**QFN10003 CORO DRIVER IS NOW STOPPING**

**Explanation:** The operator entered ZTEST CORO SOP to stop continuous mode.

**QFN10004 CORO DRIVER IS NOT ACTIVE**

**Explanation:** The operator entered ZTEST CORO STAT and the CORO driver is not running in continuous mode.

**QFN10006 CORO DRIVER IS RUNNING**

**REQUESTED ECB= e NEST = n RIAT=D3D3 FA= f CRETC= c**  
**RUNNING ECBS = r**  
**TOTAL COMMITS = t1**  
**TOTAL ROLLBACKS = t2**

**Explanation:** The operator entered ZTEST CORO STAT and the CORO driver is running in continuous mode. The CORO driver activity level is being displayed.

**QFN10007 CORO DRIVER IS NOW STOPPED**

**Explanation:** The operator entered ZTEST CORO STOP to stop continuous mode, and the driver has successfully stopped.

**QFN10008 CORO DRIVER IS NOT ACTIVE, ALTER REJECTED**

**Explanation:** The operator entered ZTEST CORO ALTER but the CORO driver is not running in continuous mode.

**QFN10009 CORO DRIVER LEVELS HAVE BEEN ALTERED**

**Explanation:** The operator entered ZTEST CORO ALTER and the driver activity level was altered successfully.

**QFN10010 CORO DRIVER RESET COMPLETE**

**Explanation:** The operator entered ZTEST CORO RESET and the command completed successfully.

**QFN10011 CORO DRIVER FIND ERROR ON QL0QL**

**Explanation:** The operator entered ZTEST CORO RESET but the CORO driver hit an error locating #QL0QL records.

**QFN20001 CORO DRIVER IS STARTING**

**Explanation:** The operator entered ZTEST CORO START and the CORO driver started to run in continuous mode.

**QFN20004 CORO DRIVER RESET BEGUN**

**Explanation:** The operator entered ZTEST CORO RESET and the CORO driver began to reset the CORO driver records .

**QFN20005 CORO DRIVER IS RUNNING ONCE**

**Explanation:** The operator entered ZTEST CORO ONCE and the CORO driver started to run through all test cases once.

**QFN20006 INVALID POOLID VALUE GIVEN**

**Explanation:** The operator entered ZTEST CORO with the POOLID parameter specified and input pool ID is not valid.

**QFN20007 CORO DRIVER IS VERIFYING**

**Explanation:** The operator entered ZTEST CORO VERIFY and the CORO driver started to validate that the CORO database is consistent.

**QFN20008 INVALID FA VALUE. 200 MAX**

**Explanation:** The operator entered ZTEST CORO with the FA parameter specified and the FA value has exceeded the maximum (200) allowed.



**QFN30001 CORO driver database verification is completed**

**Explanation:** The operator entered ZTEST CORO VERIFY and the CORO driver completed validating the CORO database.

Below is a list of the CORO64 driver messages:

**QMS00001 CORO64 DRIVER IS ALREADY ACTIVE, START REJECTED**

**Explanation:** The operator entered ZTEST CO64 START and the CORO64 driver is already running in continuous mode. Enter ZTEST CO64 STOP to stop continuous mode.

**QMS00002 CORO64 DRIVER IS NOT ACTIVE, STOP REJECTED**

**Explanation:** The operator entered ZTEST CO64 STOP but the CORO64 driver is not running in continuous mode.

**QMS00003 CORO64 DRIVER IS NOW STOPPING**

**Explanation:** The operator entered ZTEST CO64 STOP to stop continuous mode.

**QMS00004 CORO64 DRIVER IS NOT ACTIVE**

**Explanation:** The operator entered ZTEST CO64 STAT and the CORO64 driver is not running in continuous mode.

**QMS00005 CORO64 DRIVER IS STOPPING  
RUNNING ECB = n**

**Explanation:** The operator entered ZTEST CO64 STAT but the CORO64 driver is in the process of stopping the continuous mode.

**QMS00006 CORO64 DRIVER IS RUNNING**  
**REQUESTED ECB= e NEST = n RIAT=D3D3 FA= f CRETC= c**  
**RUNNING ECBS = r**  
**TOTAL COMMITS = t1**  
**TOTAL ROLLBACKS = t2**

**Explanation:** The operator entered ZTEST CO64 STAT and the CORO64 driver is running in continuous mode. The CORO64 driver activity level is being displayed.

**QMS00007 CORO64 DRIVER IS NOW STOPPED**

**Explanation:** The operator entered ZTEST CO64 STOP to stop continuous mode, and the driver has successfully stopped.

**QMS00008 CORO64 DRIVER IS NOT ACTIVE, ALTER REJECTED**

**Explanation:** The operator entered ZTEST CO64 ALTER but the CORO64 driver is not running in continuous mode.

**QMS00009 CORO64 DRIVER LEVELS HAVE BEEN ALTERED**

**Explanation:** The operator entered ZTEST CO64 ALTER and the driver activity level was altered successfully.

**QMS00010 CORO64 DRIVER RESET COMPLETE**

**Explanation:** The operator entered ZTEST CO64 RESET and the command completed successfully.

**QMS00011 CORO64 DRIVER FIND ERROR ON QL0QL**

**Explanation:** The operator entered ZTEST CO64 RESET but the CORO64 driver hit an error locating #QL0QL records.

**QMS10001 CORO DRIVER IS STARTING**

**Explanation:** The operator entered ZTEST CO64 START and the CORO64 driver started to

run in continuous mode.

---

**QMS10004 CORO64 DRIVER RESET BEGUN**

**Explanation:** The operator entered ZTEST CO64 RESET and the CORO64 driver began to reset the CORO64 driver records.

---

**QMS10005 CORO64 DRIVER IS RUNNING ONCE**

**Explanation:** The operator entered ZTEST CO64 ONCE and the CORO64 driver started to run through all test cases once.

---

**QMS10006 INVALID POOLID VALUE GIVEN**

**Explanation:** The operator entered ZTEST CO64 with the POOLID specified and input pool ID is not valid.

---

**QMS10007 CORO64 DRIVER IS VERIFYING**

**Explanation:** The operator entered ZTEST CO64 VERIFY and the CORO64 driver started to validate the CORO64 database is consistent.

---

**QMS10008 INVALID FA VALUE. 200 MAX**

**Explanation:** The operator entered ZTEST CO64 with the FA parameter specified and the FA value has exceeded the maximum (200) allowed.

---

**QMS20001 CORO64 driver database verification is completed**

**Explanation:** The operator entered ZTEST CO64 VERIFY and the CORO64 driver completed validating the CORO64 database.

## References

For more information about reading syntax diagrams, also referred to as railroad diagrams, see *Accessibility information* in the TPF Product Information Center.