z/TPF KPTS Driver

User's Guide

This page intentionally left blank.

ZTEST KPTS

Use this command to execute the KPTS driver and test >4K keypoint support interfaces. The KPTS driver executes test cases for the group of C APIs and assembler macros that provide support for the larger keypoint support.

Requirements and restrictions

None.

Format

```
                    .- -HELP---------------------------------.
>>--ZTEST-- --KPTS--+----------------------------------------+------><
                    |                                        |
                    +-+- -KPT-x---+--+--------------------+--+
                    | '- -KPT-ALL-'  '-| KPTS Parameters |-' |
                    |                                        |
                    '- -?------------------------------------'

KPTS Parameters:

   .- -NOSnap-.  .- -NOTOut-.  .- -NOVERBose-.
|--+----------+--+----------+--+-------------+--+-------------+-------->
   '- -Snap---'  '- -TOut---'  '- -VERBose---'  | .- -Cso---. |
                                                +-+---------+-+
                                                | '- -NOCso-' |
                                                |             |
                                                | .- -Bso---. |
                                                '-+---------+-'
                                                  '- -NOBso-'

>--+-------------+--+---------------+--+-----------------+------------->
   '- -PROC-proc-'  '- -IMage-image-'  | .- -NOKSA-.     |
                                       +-+---------+-----+
                                       | '- -KSA---'     |
                                       |                 |
                                       | .- -NOKBA-.     |
                                       +-+---------+-----+
                                       | '- -KBA---'     |
                                       |                 |
                                       | .- -NOKLGFBA-.  |
                                       +-+------------+--+
                                       | '- -KLGFBA---'  |
                                       |                 |
                                       | .- -NOREFRESH-. |
                                       '-+-------------+-'
                                         '- -REFRESH---'


   .- -NOVERIFY-.  .- -NOHDRS-.  .- -NOERRorcase---.
>--+------------+--+----------+--+-----------------+-------------------|
   '- -VERIFY---'  '- -HDRS---'  '- -ERRorcase-d---'
```

Help | ?
HELP, ?, or entering an incorrect format will cause a display describing the correct syntax of the ZTEST KPTS command. A list of supported keypoints is included in the help display.

KPT-*x*
specifies the keypoint to be accessed. Valid values are 0-6, 9, A-E, I, M, V. Specifying Z for the keypoint will test using an invalid keypoint.

KPT-ALL
specifies that all tests will be executed for all keypoints supported by this driver.

Snap
displays the first 100 bytes of the keypoint.

NOSnap
does not display any bytes of the keypoint.

TOut
exercises the timeout feature of keypoint support.

NOTOut
does not exercise the timeout feature of keypoint support.

VERBose
displays all informational and error messages.

NOVERBose
displays only error messages.

Cso
exercises the CSO function call API support: tpf_getkc(), tpf_updkc(), and tpf_relkc().

NOCso
does not exercise the CSO function API support.

Bso
exercises the BSO MACRO API support (GETKC, UPDKC, RELKC).

NOBso
does not exercise the BSO MACRO API support.

PROC-*proc*
specifies which processor should be used to exercise the APIs. If not specified, PROC defaults to the current processor.

IMage-*image*
specifies which image should be used to exercise the APIs. Valid values are 1-8.

NOKSA
does not exercise the keypoint staging area (KSA) API support.

KSA
exercises the KSA API support.

NOKBA
does not exercise the keypoint backup area (KBA) API support.

KBA
exercises the KBA API support.

NOKLGFBA
does not exercise the loader general file keypoint backup area (KLGFBA) API support.

KLGFBA
exercise the KLGFBA API support.

NOREFRESH
does not exercise the REFRESH parameter of the API support.

REFRESH
exercises the REFRESH parameter of the API support.

VERIFY
enables the automated tagging of a keypoint and verifies that the correct keypoint was retrieved. This option can produce expected "TAG "errors messages when the system is busy due to other activities updating keypoints.

NOVERIFY
disables the automated tagging of a keypoint and verifying that the correct keypoint was retrieved. This is the default.

HDRS
exercises the HDRS parameter of the API support.

NOHDRS
exercises the NOHDRS parameter of the API support.

ERRorcase-*d*
specifies which error test case to execute. Valid values are 1-9. See *Additional information* for an explanation of all the error cases.

NOERRorcase
specifies that the normal test cases are executed.

Source code information

The KPTS driver consists of the following program segments:

Header Files

Header File	Description
c_kpts.h	This header file was specifically created to hold common variables, definitions and struct used in qkp000.cpp and qkp010.cpp.

BSOs

Module	Makefile	Segment	Description
QKP2	N/A	qkp2.asm	Contains the logic to test the BSO timeout feature. This program is activated from qkp050 via CREMC and attempts to hold the same keypoint that is already being held by qkp050.
QKP3	qkp3.mak	qkp3.asm	Modifies the LAST_FILED_BY field of the requested keypoint on the requested remote processor. QKP3 is entered from QKP0.

CSOs

Module	Makefile	Segment	Description
QKP0	qkp0.mak	qkp000.cpp	The main entry point for the driver containing the message parser logic and the logic to test each keypoint individually or loop to test all the keypoints.
		qkp010.cpp	Contains all the CSO function tests.
		qkp050.asm	Contains all the BSO macro tests.
QKP1	qkp1.mak	qkp100.cpp	Contains the logic to test the CSO timeout feature. This program is activated from qkp010 via cremc() and attempts to hold the same keypoint that is already being held by QKP1.

Additional information

Below is the detailed information about the BSO and CSO error test cases:

BSO Error Test Cases

The following error test cases are invoked by entering:

ZTEST KPTS KPT-*x* ERR-*d* BSO VERBOSE

- Case-1: Test GETKC when no maloc storage is available.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* Passed no-storage test for GETKC
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed no-storage test for GETKC

- Case-2: Test UPDKC when no maloc storage is available.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* Passed no-storage test for UPDKC
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed no-storage test for UPDKC

- Case-3: Test EXITC after GETKC with HOLD.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* Exit while holding. CTL-8 Expected
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed error case 3

- Case-4: Test SERRC exit after GETKC with HOLD
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* DUMP Exit holding. F00F00 Expected
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed error case 4

- Case-5: Test UPDKC UNHOLD when not holding keypoint.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* UNHOLD-not holding. CTL-24 Expected
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed UNHOLD error test for UPDKC

- Case-6: Test UPDKC with invalid IMAGE value.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* INVALID IMAGE. OPR-0640F2 Expected
  QKP00001E BSO ERROR TEST kpt: *x* Passed bad IMAGE error test - UPDKC
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed bad IMAGE error test - UPDKC

- Case-7: Test UPDKC with corrupted header.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* INVALID HDRS. OPR-0640F2 Expected
  QKP00001E BSO ERROR TEST kpt: *x* Passed bad HDRS error test - UPDKC
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed bad HDRS error test - UPDKC

- Case-8: Test RELKC when keypoint is not held.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* RELKC-not holding. CTL-24 Expected
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed RELKC error test.

- Case-9: Test GETKC with invalid PROC value.
  Normal Response:
  QKP00001E BSO ERROR TEST kpt: *x* INVALID PROC. OPR-0640FD Expected
  Error Response:
  QKP00001E BSO ERROR TEST kpt: *x* Failed INVALID PROC test.

CSO Error Test Cases

The following error test cases are invoked by entering:

ZTEST KPTS KPT-*x* ERR-*d* CSO VERBOSE

- Case-1: Test GETKC when no maloc storage is available.
  Normal Response:
  kpts: kpt-*x* testing error case 1.
  kpts: successful malloc error test.
  Error Response:
  kpts: Failed malloc error test.
  kpts: Malloc unplanned error.

- Case-2: Test UPDKC when no maloc storage is available.
  Normal Response:
  kpts: kpt-*x* testing error case 2.
  kpts: successful malloc error test.
  Error Response:
  kpts: Failed malloc error test.
  kpts: Malloc unplanned error.

- Case-3: Test EXITC after GETKC with HOLD.
  Normal Response:
  kpts: kpt-*x* testing error case 3.
  kpts: kpt-*x* Exit while holding. Expecting CTL-8

- Case-4: Test SERRC exit after GETKC with HOLD
  Normal Response:
  kpts: kpt-*x* testing error case 4.
  kpts: kpt-*x* Dump exit hold while holding. Expecting OPR-F00F00

- Case-5: Test UPDKC UNHOLD when not holding keypoint.
  Normal Response:
  kpts: kpt-*x* testing error case 5.
  kpts: kpt-*x* UNHOLD-not holding. CTL-24

- Case-6: Test UPDKC with invalid IMAGE value.
  Normal Response:
  kpts: kpt-*x* Error case 6-Invalid Image. Expecting OPR-0640F2
  kpts: kpt-*x* Expecting OPR-0640F2

- Case-7: Test UPDKC with corrupted header.
  Normal Response:
  kpts: kpt-*x* Error case 7-Corrupt HDRS.
  kpts: kpt-*x* Expecting OPR-0640F2

- Case-8: Test RELKC when keypoint is not held.
  Normal Response:
  kpts: kpt-*x* Error case 8-RELKC-not holding.
  kpts: kpt-*x* Expecting CTL-24

- Case-9: Test GETKC with invalid PROC value.

Normal Response:
kpts: kpt-*x* Error case 9-Invalid PROC.
kpts: kpt-*x* Expecting OPR-0640FD

Examples

The following example tests all of the keypoints exercising both BSO macros and CSO function calls and displaying a SNAP of the first 100 bytes of the keypoint retrieved:

```
ZTEST KPTS KPT-ALL SNAP
```

The following example tests the timeout feature with keypoint A:

```
ZTEST KPTS KPT-A TOUT
```

Messages

The following messages are those displayed when both CSO and BSO support is exercised for keypoint A with a snap of the retrieved keypoint: ZTEST KPTS KPT-A SNAP VERBOSE

BSO exercises of GETKC without hold

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
QKP00001I 08.26.14 BSO test kpt: A  retrieving.     +
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
QKP00001I 08.26.14 BSO test kpt: A  retrieved.     +
```

BSO exercises of GETKC with hold

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
QKP00001I 08.26.14 BSO test kpt: A  retrieving HOLD.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
QKP00001I 08.26.14 BSO test kpt: A  retrieved HOLD.+
```

SNAP of retrieved data

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
SNAP0003I 08.26.14 CPU-B SS-BSS SSU-HPN IS-1
PSW-07150001 80000000 00000003 91165080
PGM-QKP0 CODE-I000FFFFF TERM-010000B
KPTA     AT 000000000CE00000-C3D20100 D8D2D7F0 00000000 CK..QKP0.... _
KPTA     AT 000000000CE0000C-00000000 00000000          ............
END OF SNAPC CONSOLE DISPLAY+
```

BSO exercises of UPDKC

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
QKP00001I 08.26.14 BSO test kpt: A  UPDATE        +
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
QKP00001I 08.26.14 BSO test kpt: A  UPDATED       +
```

BSO exercises of RELKC

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
QKP00001I 08.26.14 BSO test kpt: A  RELEASE       +
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
```

```
QKP00001I 08.26.14 BSO test kpt: A  RELEASED       +
```

SNAP of retrieved data

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
SNAP0003I 08.26.14 CPU-B SS-BSS SSU-HPN IS-1
PSW-07150001 80000000 00000003 91165080
PGM-QKP0 CODE-I000FFFFF TERM-010000B
KPTA     AT 000000000CE00000-C3D20100 D8D2D7F0 00000000 CK..QKP0.... _
KPTA     AT 000000000CE0000C-00000000 00000000          ............
END OF SNAPC CONSOLE DISPLAY+
```

Begin CSO exercises

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
kpts: testing kpt A+
```

CSO exercises of tpf_getkc()

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
getkc: entering getkc_func.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
getkc: retrieving kpt-A.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_getkc processing request.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_getkc: kpt-A retrieved +
```

SNAP of retrieved data (delayed console display from the previous nohold test)

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
SNAP0003I 08.26.14 CPU-B SS-BSS SSU-HPN IS-1
PSW-07151001 80000000 00000000 0CD0F1CC
PGM-QKP0 CODE-A00009111 TERM-010000B
KPT-DATA AT 000000000CE0C008-C3D20100 D8D2D7F0 00000000 CK..QKP0.... _
KPT-DATA AT 000000000CE0C014-00000000 00000000 00000000 ............
KPT-DATA AT 000000000CE0C020-00000000 00000000 0001C100 ..........A.
KPT-DATA AT 000000000CE0C02C-00000000 06400000 00000000 ..... ......
KPT-DATA AT 000000000CE0C038-50A0008F 42FC01E0 B0009800 ..... ....q. _
KPT-DATA AT 000000000CE0C044-00010000 000F0000 000001F4 ...........4
KPT-DATA AT 000000000CE0C050-00000005 00780000 00000000 ............
KPT-DATA AT 000000000CE0C05C-00000000 00000000 00000000 ............
KPT-DATA AT 000000000CE0C068-00000000                   ............
END OF SNAPC CONSOLE DISPLAY+
```

CSO exercises of tpf_getkc() with HOLD

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
getkc: retrieving kpt-A HOLD.+
```

Retrieving the keypoint

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_getkc processing request.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_getkc: kpt-A retrieved
```

SNAP of retrieved data

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
SNAP0003I 08.26.14 CPU-B SS-BSS SSU-HPN IS-1
```

```
PSW-07151001 80000000 00000000 0CD0F1CC
PGM-QKP0 CODE-A00009111 TERM-010000B
KPT-DATA AT 000000000CE0C008-C3D20100 D8D2D7F0 00000000 CK..QKP0.... _
KPT-DATA AT 000000000CE0C014-00000000 00000000 00000000 ............
KPT-DATA AT 000000000CE0C020-00000000 00000000 0001C100 ..........A.
KPT-DATA AT 000000000CE0C02C-00000000 06400000 00000000 ..... ......
KPT-DATA AT 000000000CE0C038-50A0008F 42FC01E0 B0009800 ..... ....q. _
KPT-DATA AT 000000000CE0C044-00010000 000F0000 000001F4 ...........4
KPT-DATA AT 000000000CE0C050-00000005 00780000 00000000 ............
KPT-DATA AT 000000000CE0C05C-00000000 00000000 00000000 ............
KPT-DATA AT 000000000CE0C068-00000000                   ............
END OF SNAPC CONSOLE DISPLAY+
```

CSO exercise of tpf_updkc()

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_updkc NOWAIT processing request.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_updkc WAIT processing request.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_getkc after update processing request.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_updkc: kpt-A updated+
```

CSO exercise of tpf_relkc()

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_relkc processing request.+
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
tpf_relkc: kpt-A released.+
```

Completed exercises for keypoint A

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
kpts: completed test kpt A.+
```

All exercises completed. Driver exiting

```
CSMP0097I 08.26.14 CPU-B SS-BSS  SSU-HPN  IS-01
KPTS driver exited +
```

References

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