Troubleshooting and support

Version 11 Release 0

Contents

Troubleshooting and support

Learn how to troubleshoot the product.

Finding known problems

Known problems are documented in the form of individual technotes in the Support knowledge base.

As problems are discovered and resolved, the IBM® Support team updates the knowledge base at the IBM Support Portal. By searching the knowledge base, you can quickly find workarounds or solutions to problems. The following link runs a customized query of the live Support knowledge base: View all known problems for IBM z Systems™ Development and Test Environment.

For a complete listing of the currently available technotes, see IBM z Systems Development and Test Environment Tech Notes®.

IBM Software Support

Learn how to contact IBM Software Support for help with problems, fixes, and other support information.

For contact information and guidelines or reference materials, read the *IBM Software Support Handbook*.

For FAQs, lists of known problems and fixes, and other support information, see the product support page for IBM z Systems Development and Test Environment.

For product news, events, and other information, see the Multiplatform development home page.

Before you contact IBM Software Support, gather the background information that you need to describe your problem. When you describe a problem to an IBM software support specialist, be as specific as possible and include all relevant information so that the specialist can help you solve the problem. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, or messages that are related to the problem?
- Can you reproduce the problem? If so, what steps do you take to reproduce it?
- Is there a workaround for the problem? If so, be prepared to describe the workaround.

For answers to "how to" questions, see the z Systems Development and Test Environment Forum.

Obtaining the latest editions of Redbooks

IBM Redbooks® provide information about major components of IBM z Systems Development and Test Environment.

IBM z Systems Development and Test Environment

To obtain the latest edition of the *zPDT Guide and Reference (SG24-8205)* for use with IBM z Systems Development and Test Environment, go to the IBM Redbooks website and search for *zPDT Guide and Reference (SG24-8205)*.

Troubleshooting tips

Table 1. Troubleshooting tips

Symptom	Problem Determination
For any problem after installation of z Systems Development and Test Environment (creating a devmap, installing z/OS®, starting or stopping z Systems Development and Test Environment, z/OS networking issues, performance issues)	Validate the environment and the installation by running the `z1090instcheck` command and fixing errors. Do not worry about errors that are related to `kernel.core_pattern` unless the emulator is terminating abnormally.
AWScccnnns message that is issued from the Linux terminal that is running z Systems Development and Test Environment.	Any message beginning with AWS is a zPDT® message. Enter the following zPDT command `msgInfo` *`message-number`* Where message-number is the 10 character message. This command displays more information about the reason for the message. See 4.1.36, “The `msgInfo` command” in the zPDT Guide and Reference.

Installing z Systems Development and Test Environment

Table 2. Troubleshooting the Installation of the z Systems Development and Test Environment

Symptom	Problem Determination
Message CRIMA1076E in the install log when you are uninstalling a version of z Systems Development and Test Environment	For the step-by-step process to perform if an uninstall fails., see Uninstalling a previous version of z Systems Development and Test Environment.
When you are attempting to run launchpad.sh or launchpad-console.sh, you get the following error: `./InstallerImage_linux.gtk.x86_64/ tools/imcl: Permission denied`	This issue can happen for several reasons: • You are attempting the launchpad command without running as root. Ensure that you are executing this command as root • You copied the installation disc to a location, and then attempted executing the launchpad commands from that new location. Sometimes when you copy files in Linux, the execution permission bits are not retained during the copy. Ensure that the particular launch command is marked as executable.

Table 2. Troubleshooting the Installation of the z Systems Development and Test Environment (continued)

When you are attempting to perform an Installation Manager update for z Systems Development and Test Environment, or to run `launchpad.sh` or `launchpad-console.sh`, you get one of the following errors:
`sntl-sud-xxxx is required by z1091-1-xxxx but is already installed`
Or
`zpdt-shk-server-xxxx is required by z1091-1-xxxx but is already installed`
	• If these messages are from an attempt to migrate to a new version of z Systems Development and Test Environment by performing an Update in Installation Manager, perform an uninstall of the previous version instead, followed by a new installation. For the step-by-step process to perform an uninstall, see Uninstalling a previous version of z Systems Development and Test Environment.
	• If these messages are from running `launchpad.sh` or `launchpad-console.sh`, perform an uninstall of the previous version before you attempt a new installation. For the step-by-step process to perform an uninstall, see Uninstalling a previous version of z Systems Development and Test Environment.

You receive any error message when you are uninstalling a previous version of z Systems Development and Test Environment using the steps outlined in Uninstalling a previous version of z Systems Development and Test Environment.	1. If you are uninstalling Version 9.0 or 9.1, run the `z1091_removeall` command • Start a Linux console as user root. • Change Directory (`cd`) to `/usr/z1090/bin`, the folder that contains the `z1091_removeall` command. • Run the command as follows: `./z1091_removeall` • Reply **Yes** to any prompts that ask if you are sure that you want to uninstall 2. If you still get some form of error messages, such as failure to uninstall the Sentinel Keys Server (`zpdt-shk-server`) or Sentinel System Driver (`sntl-sud`), attempt these steps in the following order: • Enter `# rpm -e --allmatches z1090` • Enter `# rpm -e --allmatches z1091` • Enter `# rpm -e --allmatches zpdt-shk-server` • Enter `# rpm -e --allmatches sntl-sud`

Table 2. Troubleshooting the Installation of the z Systems Development and Test Environment (continued)

When an installation fails for the z Systems Development and Test Environment License Manager, and the log displays the following entries:
`error: Failed dependencies:`
`libc.so.6 is needed by aksusbd-7.40-1.i386`
`libc.so.6(GLIBC_2.0) is needed by aksusbd-7.40-1.i386`
`libc.so.6(GLIBC_2.1) is needed by aksusbd-7.40-1.i386`
`libc.so.6(GLIBC_2.2) is needed by aksusbd-7.40-1.i386`
`libc.so.6(GLIBC_2.3.4) is needed by aksusbd-7.40-1.i386`
`libc.so.6(GLIBC_2.4) is needed by aksusbd-7.40-1.i386`
`libpthread.so.0 is needed by aksusbd-7.40-1.i386`
`libpthread.so.0(GLIBC_2.0) is needed by aksusbd-7.40-1.i386`
`libpthread.so.0(GLIBC_2.1) is needed by aksusbd-7.40-1.i386`
`libpthread.so.0(GLIBC_2.3.2) is needed by aksusbd-7.40-1.i386`
	32-bit versions of the Linux glibc libraries are not installed. Install the appropriate 32 bit glibc library for your Linux system and try again.

USB Hardware device activation and licensing

Troubleshooting the USB Hardware device activation and licensing

Table 3. USB Hardware device activation and licensing. Troubleshooting USB Hardware device activation and licensing

Symptom	`SecureUpdateUtility` or `Z1091_token_update` does not apply the update file to the USB hardware device.
Problem Determination	• Ensure that the last 5 hexadecimal digits of the serial number that is etched on the USB hardware device match the serial number in the file name of the update file you are applying. Sometimes the etched number is hard to read. If you are unable to read the etched serial number, IBM support can assist you with an alternative method of obtaining the serial number. • Ensure that the hardware device is recognized by your Linux system by plugging in the device and issuing the `lsusb` command. If the device is available, you see an entry that shows "Rainbow Technologies Inc".
Symptom	Unit not found message that is issued during the `SecureUpdateUtility -u` or the `z1091_token_update` command
Problem Determination	This issue can happen for several reasons: • You try to apply an update file with one serial number to a USB hardware device with a different serial number. • You try to update a USB hardware device from a client machine that is connected to a product license server that has no local USB Hardware device Ensure the update file that is being applied was generated for the correct USB Hardware Device. Ensure that the USB Hardware device is securely plugged in to the local device or remote product license server as is appropriate.

Table 3. USB Hardware device activation and licensing (continued). Troubleshooting USB Hardware device activation and licensing

Symptom	When you are starting z Systems Development and Test Environment, (`awsstart`) you get any of the following messages at the Linux terminal that indicates a failure to get a license:
	•
	`Error: Failed to get license.`
	`Return code: 312`
	• `Error: Unable to get expected`
	`license: xxxx`
	
	or
	`Error: Unable to locate expected`
	`license: xxxx`
	•
	`AWSEMI315E zPDTA License Unavailable`
	`for CPU 0`
	`AWSEMI005I Waiting for 1090 license`
	•
	`CPU Address Out Of Range`
	•
	`SFNTGetLicense failed.`
	`Return code: 312`

Table 3. USB Hardware device activation and licensing (continued). Troubleshooting USB Hardware device activation and licensing

Problem Determination	This issue can happen for several reasons:
	• The local USB Hardware device is not plugged in properly. Attempt to "reseat" the USB Hardware device. Unplug the device for 15 seconds, then reinsert it.
	• The correct update file is not applied to the matching USB Hardware device, or it did not apply successfully.
	– Ensure the update file that is requested and applied has the same serial number within the file name as the serial number of the USB Hardware device. Serial numbers can be hard to read. Look carefully.
	– Ensure the `SecureUpdateUtility -u` or `Z1091_token_update -u` command responded with a "`Success`" message.
	• The requested license type is not correct.
	– Validate that if your device map includes the `cpuopt zvm_couplingfacility` directive, your USB hardware key includes sysplex enabling licenses. If no sysplex license is available, your device map cannot enable sysplex operation.
	– Validate that if your device map includes the `RDTSERVER= directive` or you have an `RDTSERVER` environment variable set to enable participation in a Rational® Token licensing system, your USB hardware key includes token enabling licenses. If token enabling licenses are not available on your USB hardware key, your z Systems Development and Test Environment instance cannot participate in a Rational Token licensing system.
	• The system clock is incorrect on the system with the USB hardware device plugged in to it. Ensure that the system clock is correct. Never set the system clock to a future date or time when the USB hardware key is plugged in or you will damage the USB hardware key.
	• You are trying to use a local USB Hardware device, but your client configuration points to a product license server
	– From a Linux terminal that is running as root, enter the `clientconfig` command that is found in `/usr/z1090/bin`. Ensure the `License ContactServer` is `localhost` if you are attempting to use a local USB. No other fields need be completed.
	• The `/usr/z1090/bin/sntlconfig.xml` file is not readable by the programs that need it. Verify that `/usr/z1090/bin/sntlconfig.xml` is "world readable". If it is not, use `chmod` to change the permission bits to `644`.
	• You do not have the correct type of USB Hardware Device.
	– Perform the following command from a Linux terminal that is running as root:
	`rpm -qa | grep z109`
	
	If Rational Development and Test Environment for z System is properly installed, you should see an rpm of the form
	`z1091-xxx and not z1090-xxx`
	– Ensure the USB Hardware device that you are using is a Rational Development and Test Environment for z System Hardware device, and not a z1090 hardware device. A z1090 hardware device has a metal cable with a plastic tag attached to the end of the device. A z1091 hardware device does not.

Problem Determination (continued)	• You do not have connectivity from the instance that is performing the `awsstart` to the product license server.
	– Enter a `UIMCHECK` on the client instance to verify whether there is proper connectivity. You see the message
	`The remote QUERY command executed successfully`
	– If this procedure does not complete successfully, try the following procedure.
	- From a Linux terminal that is running as root, enter the `clientconfig` command that is found in `/usr/z1090/bin`. Ensure the `License ContactServer` is the IP address of the product license server, that the `License PortNumber` is `9450`, and that `License ipv6` is `yes` or `no` as is appropriate for your installation.
	- From a Linux terminal on the instance that is running z Systems Development and Test Environment, enter a ping to the IP address configured as the license server. If the ping command fails, you have an IP connectivity problem or the server is not started.
	• Ensure that the server was started once by a non-root ID with the `uimserverstart` command.
	- If you believe that there is connectivity to the network, from a Linux terminal that is running as root, check the firewall configuration by entering the following command
	`iptables -L -n`
	Ensure ports `9450` and `9451` are allowed. For information on how to configure the firewall to allow the appropriate ports, see Firewall considerations in Configuring the base Linux system.
	- If this installation is a newly installed product license server, check the firewall configuration on the product license server. If security is being used on the product license server, ensure that the sentinel configuration file on the server allows for the IP address of the client. This file is in
	- `/opt/safenet_sentinel/common_files/sentinel_keys_server/sntlconfigsrve.xml`
	For more information, see paragraph 8.4.2, “Security” in chapter 8 of the zPDT Guide and Reference.
	In some cases, it might be helpful to restart the license daemon where the USB key is plugged in. Restarting the license daemon is done as root with the command:
	`/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart`

Symptom	When you are starting Rational Development and Test Environment for z Systems, (`awsstart`) you get the following messages at the Linux terminal that indicate an expired license
	`CPU 0, No Sentinel License Available 65535`
	`Unknown Error Code`
	`CPU 1, No Sentinel License Available 65535`
	`Unknown Error Code`
	`CPU 2, No Sentinel License Available 65535`
	`Unknown Error Code`
	`Expired License. EXP=3/1/2015`
	`AWSEMI315E zPDTA License Unavailable for CPU x`

Problem Determination	A replacement update file must be obtained from the Rational License Key Center when a license expires, and the update file must be applied to the USB hardware device being used. For more information, see "Steps to get the replacement file" in the *z Systems Development and Test Environment Activation Guide (SC27-6630)*.
Symptom	Time cheat errors
Problem Determination	• Time cheat errors occur when the time stored on the USB hardware device is in the future when compared to the system clock of the Linux system it is plugged into. The USB device stores the latest date of the system clock it sees and will not provide licenses until the Linux system clock matches. Verify that the Linux system clock is correct. Have your system clock set to UTC to avoid errors that are caused by semi-annual time changes. There is no way to inspect the time on the USB key. However, you might wait a short time to see whether the Linux clock catches up to the time stored on the USB key. Do NOT set the Linux clock ahead of the actual time. If you suspect that the time stored on the USB key is in the future, contact IBM support.

Troubleshooting installing, operating, and uninstalling the License Manager

Learn how to troubleshoot problems installing, operating, and uninstalling the License Manager.

Symptom

Uninstalling the License Manager fails.

Problem Determination

Always stop the aksusbd daemon and perform a **uimserverstop** before attempting to uninstall. If the **uimserverstop** is not performed and the uninstalling operation fails, perform the following commands:

```
rpm -e --allmatches UIM
rpm -e --allmatches aksusbd
```

Installing z/OS volume images

Table 4. Trouble shooting the Installation of z/OS volume images

Symptom	Problem Determination

You get the following errors while performing a z1091_ADCD_install command:
`Error: Failed to get license.`
`Return code: 312.`
`Error: Unable to locate expected license: 0D98`
	For information on possible causes, see the same symptom in ""USB Hardware device activation and licensing" on page 4".
	
	Also this can occur if the USB Hardware device, either local or in a product license server, has been updated with a `.upw` update file instead of a `.zip` update file. If you are migrating a z Systems Development and Test Environment instance from version 9 to version 9.1, you must request a new update file from the Rational License Key Center to install the ADCD `S1RES1` and `SARES1` volumes.

Table 4. Trouble shooting the Installation of z/OS volume images (continued)

Defining the z Systems Development and Test Environment machine characteristics (DEVMAP)

Table 5. Defining the z Systems Development and Test Environment machine characteristics (DEVMAP). Troubleshooting defining the z Systems Development and Test Environment machine characteristics (DEVMAP)

Symptom	Problem Determination
Unable to run `find_io` command at `create_devmap.pl`	The path to the `find_io` command is not available. Ensure that you run `aws_bashrc` from the user ID that runs `create_devmap.pl` and z Systems Development and Test Environment. This updates the path to the needed files.

Starting and stopping z Systems Development and Test Environment

Table 6. Starting and stopping z Systems Development and Test Environment. Troubleshooting the starting and stopping of z Systems Development and Test Environment

Symptom	Problem Determination
`AWSSTA140E – Initialization failure after license obtained`	Ensure that you are starting z Systems Development and Test Environment from the home directory of the user who is created for running z Systems Development and Test Environment. Ensure that you have run the `aws_bashrc` command while logged on as that user, and from that user's home directory.
The following messages occur during startup: `AWSECH004S Unable to define RAS/FEDC memory, RC=-3.` `AWSECH004S Unable to define RAS/FEDC memory, RC=-3.`	This message is seen with many `awsckd` definitions. The Linux kernel setting `SHMMNI`, which sets the system-wide maximum number of shared memory segments, is probably too low. The default is `4096`. Try increasing this setting by adding `kernel.shmmni = 8192` to `/etc/sysctl.conf` and activate it using the command `sysctl -p`.
The following message occurs when ipling z/VM® 6.2 "CP requires hardware features not available on this processor"	Older z/VM V6R2 maintenance levels may not work with zPDT Version 1 Release 6. Either migrate to z/VM V6R3 or ensure PTF VM65007 is applied to your z/VM V6R2 system.

z/OS networking issues

Table 7. z/OS networking issues. Troubleshooting z/OS networking issues

Symptom	General z/OS networking configuration errors
Problem Determination	• Check syslog (z/OS console messages) for errors during the start of TCP/IP or VTAM®. • Validate configuration files match for device numbers and names in the device map, VTAM definitions, and TCP/IP configuration profile. Specifically, follow the definitions to ensure that the tunnel and external addresses in the device map are defined through VTAM and TCP/IP to be the correct device names and addresses in z/OS. These configuration errors can be tricky because device names in the TCP/IP profile are arbitrary.
Symptom	Unable to establish a network tunnel on a `10.x.x.x` network.

Problem Determination	On a `10.x.x.x` network, you can establish the tunnel by configuring the tunnel to be on a `192.168.1.x` address. For example, configure the tunnel device in the devmap as:
	`name awsosa 0009 --path=A0`
	`--pathtype=OSD`
	`--tunnel_intf=y`
	`--tunnel_ip=192.168.1.1`
	
	Configure z/OS TCP IP profile as a tunnel IP of
	`192.168.1.2 netmask 255.255.255.0`

Symptom	Cannot connect to z Systems Development and Test Environment z/OS using a remote 3270 terminal emulator
Problem Determination	• Ensure that you have network connectivity to the Linux adapter by pinging the Linux adapter IP address from the machine that is running your remote 3270 terminal emulator • Ensure that you have a tunnel OSA defined with an internal `10.x.x.x` address. Ensure that there is a matching route that sends all `10.x.x.x` traffic through the tunnel OSA • Ensure that the remote 3270 terminal emulator configuration is attempting to connect to the Linux IP address or host name, and port 3270 (or whatever port is defined on the 3270port statement) • Ensure that your firewall configuration allows traffic to the 3270port. See Firewall considerations in “Configuring the base Linux Server” above for how to configure the firewall to allow the appropriate ports.
Symptom	Cannot ping to a z/OS defined IP address on the z Systems Development and Test Environment instance

Table 7. z/OS networking issues (continued). Troubleshooting z/OS networking issues

Problem Determination	• Ensure that you have a tunnel OSA defined with an internal `10.x.x.x` address. Ensure that there is a matching route that sends all `10.x.x.x` traffic through the tunnel OSA • You can only ping to the z/OS IP address when an IP address is assigned that is advertised to the network attached to the Linux ethernet adapter. The customizations in this book define one way to do this, described as scenario 4 in the zPDT Guide and Reference. If you are using that scenario, ensure that you have completed all of these steps, which are defined in Setting up TCP/IP: – Defined a 2nd OSA, with an IP address in the same subnet as the IP address of the Linux ethernet adapter – Defined a default route through that OSA – Updated your TCPIP.DATA files to point to your domain name server and the host name for your z/OS – Updated your procedures to point to the new TCPIP.DATA files – Updated your IPNODES definitions to include the z/OS host name and IP address – Updated your resolver parms to point to the new IPNODES definitions – Updated your VTAM definitions to add the tunnel OSA and any other OSAs defined • If you are using a configuration similar to scenario 5 in the zPDT Guide and Reference, the z/OS IP address is advertised through the network using Linux Network Address Translation (NAT) functions. – Check your firewall configuration with the command `iptables -L -n` - Ensure the correct `chpid` address, z/OS IP address, broadcast address, netmask for the broadcast address, and the tunnel IP address are all defined in the table – Check your routing definitions in `BEGINROUTES` or `OMPROUTE`. Ensure the `firsthop` address is the Linux tunnel address if all traffic is being routed through the Linux. – For more information, see Chapter 7 “LANs” in the zPDT Guide and Reference.
Symptom	Cannot FTP or Telnet to a z/OS defined IP address on the z Systems Development and Test Environment instance
Problem Determination	• Verify that you can ping to the address from the remote location. If you cannot, follow the steps defined here. • If you can ping, check your firewall configuration with the command `iptables -L -n` – Ensure that the appropriate ports are allowed; typically ports 21 for FTP and 23 for Telnet

Performance

Table 8. Performance. Troubleshooting performance issues

Symptom	High volume traffic through an emulated OSA-Express® ethernet adapter on z Systems Development and Test Environment suffers poor throughput.

Table 8. Performance (continued). Troubleshooting performance issues

Problem Determination	If newer Linux kernels are installed, you might experience a drastic slowdown of OSA performance, This slowdown is immediately obvious and is due to Linux attempting to offload various functions into the adapter, which is not acceptable to the current awsOSA implementation. One or more of the following commands, intended to disable the Linux offloading of IP functions, might improve the situation:
	`# ethtool -K eth0 rx off (disable RX checksumming offload)`
	`# ethtool -K eth0 tso off (disable TCP segmentation offload)`
	`# ethtool -K eth0 gso off (disable generic segmentation offload)`
	`# ethtool -K etho gro off (disable generic RX offload)`
	`# ethtool -K eth0 lro off (disable large RX offload)`
	`# ethtool -K eth0 rxvlan off (if you are using VLANs)`
	`# ethtool -k eth0 (display status of NIC)`
	`# ethtool -S eth0 (display statistics)`
	`# ethtool -K em1 rx off (newer style of NIC naming)`
	`# ethtool -K enp0s25 rx off (newer style of NIC naming)`
	
	You might need to experiment with these commands.
	
	These commands must be entered after each Linux boot. If a script is used to start z Systems Development and Test Environment, such as the `runzpdt` script shipped with the product, that script can be modified to enter one or more of these commands each time z Systems Development and Test Environment is started.
	
	Effective combinations of these options differ with various Linux levels and with various NIC adapters. This problem was first noticed with Linux kernel level 2.6.36.2
	
	If frames larger than expected are used, an excessive number of frames might be dropped (causing a re-transmission). This may not be noticed unless careful measurements or comparisons are made. This problem might be resolved by including the `sysctl` parameter that is now recommended:
	`net.core.rmem_max=1048576`
	
	For more information on zPDT performance and performance problems, see paragraph 2.3.4 “Performance” and 7.5.9 “Performance problems” in the zPDT Guide and Reference.

Symptom	z Systems Development and Test Environment performs slowly. CPUs reach 100% utilization.
Problem Determination	Verify that you are meeting the minimum requirements for both the hardware and the software. See “z Systems Development and Test Environment prerequisites” and paragraph 2.3.2 “zPDT instances” in the zPDT Guide and Reference.

Connecting with Rational License Key Servers

Activity that is associated with the use of Rational Tokens is logged for diagnostic purposes in the log directory `$HOME/z1090/logs` in files that start with the name `feutlicm`. Messages from a Rational License Key Server and diagnostic information can be found in these logs. These logs are intended for use by IBM service but might provide useful information for quick diagnosis of problems when Rational Tokens cannot be obtained. In some cases when Rational Tokens cannot be obtained, the messages that are issued by the Rational License Key Server are also

written to the Linux console on which the `awsstart` command was entered. The `feutlicm` log can be viewed with the `less` command while z Systems Development and Test Environment is running.

Two environment variables can be used to help in troubleshooting connections to Rational License Key Servers and problems that occur when you are obtaining Rational Tokens.

Variable `RDTLOG=TTY`, if set before you start z Systems Development and Test Environment, routes all Rational Token-related logging to the Linux terminal in addition to the log.

Variable `RDTDEBUG=DEBUG`, if set before you start z Systems Development and Test Environment, adds more information to the logs. If `RDTLOG=TTY` is also set, these additional messages are also written to the Linux terminal.

Do not set `RDTLOG=TTY` in everyday use, because it sends frequent unsolicited messages to your Linux terminal. Setting `RDTDEBUG=DEBUG` in regular use has no negative side effects, other than slightly larger logs.

Rational Tokens are checked out and checked back in so that they will become available automatically after 30 minutes unless z Systems Development and Test Environment renews them before that time. z Systems Development and Test Environment renews tokens approximately every half hour.

When z Systems Development and Test Environment ends, tokens are returned immediately. However, it can take up to 2 minutes for those tokens to become available for use again. If network connectivity is lost to the Rational License Key Server, or if anything prevents a normal return, the Rational Tokens become available within 30 minutes.

To limit unnecessary log file growth, logging of successful interactions with the Rational License Key Server is suspended after about 30 successful token renewal cycles. Logging resumes if any errors are encountered.

Troubleshooting license manager and USB hardware device activation

Configuration and usage for z Systems Development and Test Environment is described in Configuring an instance of z Systems Development and Test Environment and the zPDT Guide and Reference.

Index