Reference

Version 11 Release 0

Contents

Reference 5
Setting up Rational Tokens 5
Installing, configuring, and administering a Rational License Key Server 5
Returning any existing license key for the USB hardware device or license manager 5
Obtaining and activating a new license key for Rational Token licensing 6
Updating the device map to point to a Rational License Key Server 6
Troubleshooting connections with Rational License Key Servers. 7
zPDT license servers. 8
Concepts 10
Client installation and configuration 14
Server installation and configuration 16
Notes 18
Scenarios 20
Renewing licenses 25
Searching servers 26
Allocating multiple licenses 27
Switch from Product License Server to License Manager 27
Accessibility features 28
z Systems Development and Test Environment 29
Keyboard shortcuts for the help system in the product. 29

Index 30

Reference

The reference material in this section of the IBM Knowledge Center provides supporting information for the instructions for using IBM z Systems Development and Test Environment.

Setting up Rational Tokens

Rational® Token licensing is an entitlement that you can purchase and use to run z Systems Development and Test Environment. With Rational Tokens, z Systems Development and Test Environment maintains a connection to a Rational License Key Server and starts and continues to run only when sufficient Rational Tokens are available.

Use of Rational Tokens does not replace the requirement for a license key for z Systems Development and Test Environment. Either a software-based license key file or a USB hardware device with a valid update file is still required. In either case, the license key file indicates that Rational Tokens are required.

After you purchase entitlement to Rational Token licensing, do these steps to use Rational Token licensing to run z Systems Development and Test Environment. These steps assume that you already have a USB hardware device.

Installing, configuring, and administering a Rational License Key Server

Setting up the Rational License Key Server is beyond the scope of this document, but additional information can be found in the Rational License Key Server documentation that is provided in the Rational License Key Server media.

If you need to use configuration files to specify extra Rational licensing configurations, you must specify the variable `RDTCONF` to point to a directory that contains the Rational configuration files. For example:

```
export RDTCONF=/etc/yourconffiles/
```

Returning any existing license key for the USB hardware device or license manager

If you want to allow Rational Token licensing for a USB hardware device or license manager you are already using, you must first return the license key that was previously created for that device. The Rational License Key Center does not generate new license keys unless any previously created license keys for that device are returned.

To return an existing license key for a license manager, see Returning a software-based license key less than 31 days before expiration.

To return the existing update file for a USB hardware device, log in to the Rational License Key Center, select your account and use the **Return Keys** link to return the old update file. For instructions for returning your existing update file, see Getting the replacement file.

Obtaining and activating a new license key for Rational Token licensing

Links to information about how to obtain and install a new license key for Rational Token licensing for license managers and for USB hardware devices.

To obtain and install a license key file for a license manager, see Activating a license manager.

To obtain a license key file for a USB hardware device, use the same procedures that are documented in Obtaining update files for Rational Tokens to obtain an update file for Rational Token Licensing. To activate the update file, use the steps that are described in Activating the USB hardware device.

Important: In a product license server configuration, do not mix USB hardware devices that require Rational Tokens with USB hardware devices that do not require Rational Tokens. This approach is not supported and can result in unpredictable behavior. For more information about Rational Tokens and activating the USB hardware device, see Product enablement checklist.

Activating the USB hardware device

Applying the update file that you obtained from the Rational License Key Center to your USB hardware device activates the device. Each time that you apply an update file, it overwrites the previous activation on the USB hardware device. To properly activate, the update file must be generated with the same serial number as the USB hardware device that it applies to. To change the activation of a USB hardware device, you must obtain and apply a new update file that activates the total number of license entitlements you intend to use on that device. Changing the activation of a USB hardware device involves returning and regenerating licenses in the Rational License Key Center.

The process for applying update files to USB hardware devices recently changed. For instructions on applying update files to USB hardware devices for z Systems Development and Test Environment version 9.1 or later, see Activating a USB hardware device

For instructions on applying update files to USB hardware devices for Rational Development and Test Environment for z Systems version 9.0 or earlier, see Activating a USB hardware device for version 9.0 or earlier

Important: In a product license server configuration, do not mix USB hardware devices that require Rational Tokens with USB hardware devices that do not require Rational Tokens. This approach is not supported and can result in unpredictable behavior.

Updating the device map to point to a Rational License Key Server

The Rational License Key Server that is used to distribute Rational Tokens must be available through TCP/IP. It is likely that your installation already has such a server configured. Installation materials and documentation for the server are available on separate e-images that are included with the z Systems Development and Test Environment offering.

When you are using Rational Tokens, an instance of z Systems Development and Test Environment must be able to locate the specific Rational License Key Server

you intend to use with that instance. The port and location of the server is specified either in the Linux environment variable that is named RDTSERVER or in the device map. If both are specified, the device map setting is used. To enable Rational Tokens in the device map, add the rdtserver statement to the [system] stanza. For example, to have z Systems Development and Test Environment retrieve Rational Tokens from port 27000 on a server at address sampsrvr.yournetwork.com, add the following line to your [system] stanza in the device map:

```
rdtserver 27000@sampsrvr.yournetwork.com
```

To have z Systems Development and Test Environment retrieve Rational Tokens from port 27000 on a server at address sampsrvr.yournetwork.com, you can export the RDTSERVER environment variable in your .bashrc script. For example, if you run z Systems Development and Test Environment with ID ibmsys1, you would edit /home/ibmsys1/.bashrc, and add the line:

```
export RDTSERVER=27000@sampsrvr.yournetwork.com
```

Troubleshooting connections 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 console 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 console.

Do not set RDTLOG=TTY in everyday use because it sends frequent unsolicited messages to your Linux console. 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.

zPDT license servers

Alternative zPDT® license and serial number servers that provide enterprise-wide management are available for IBM z Systems Development and Test Environment systems.

Note: For definitions of some of the terms used in these topics, see the Chapter 4, "z Systems Development and Test Environment Glossary," on page 31.

A zPDT system must have a license supplied by a 1090 or 1091 token or by a *software license server*. The tokens identified as 1091 tokens are for z Systems Development and Test Environment customers. The material in this section applies to both 1090 and 1091 tokens, and to software-only (LDK) license users. For several reasons, simple local token usage is not always appropriate:

- Due to security concerns, some PCs no longer have usable USB ports. The physical distribution of tokens might present a problem.
- Rack-mounted blade PCs might not have normal, dedicated USB ports. A token in a work location can be easily carried away.

In virtual environments the dedicated use of a USB port might be a problem.

- If multiple tokens are used, or are changed, the CP serial numbers become unpredictable. The consistency of the z Systems serial numbers might be important for some software licenses (for z Systems software) and might be important for some z Systems operating systems.
- In some cases, especially related to cloud usage, a hardware token at any location is difficult to manage.

Recognizing these concerns, alternative zPDT license and serial number servers that provide enterprise-wide management are available for z Systems Development and Test Environment systems. Figure 1 on page 9 shows the available options: a simple local configuration, a remote LDK-SL license server, and a remote SHK license server.

Important: The SHK and LDK-SL terminology associated with these servers, as shown in this figure, is used throughout this section. SHK servers have physical 1090 (or 1091) tokens and LDK-SL servers do not have physical tokens. The terms *license manager* and *license server* are used interchangeably. zPDT refers to both the ISV zPDT product (1090) and the z Systems Development and Test Environment (1091) product except where distinctions are noted.

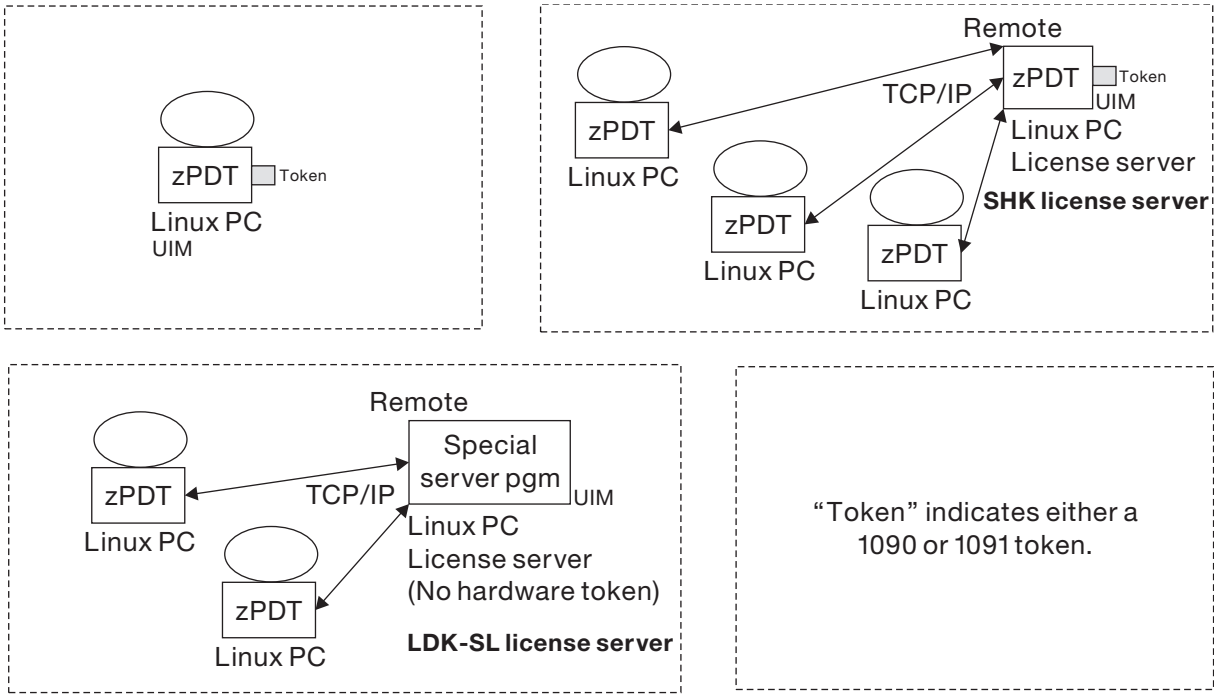

Figure 1. Options for obtaining zPDT licenses

In a simple configuration, a *local token* is installed in a USB port on the base machine running zPDT. In this case (one token installed in a local USB port), the token supplies both the zPDT license and the serial number used for the z Systems CPs, assuming that the local zPDT system has never been connected to a remote license manager or server, and has never used multiple local tokens. This configuration is used by the majority of zPDT users.

The SHK server uses a hardware token, while the LDK-SL server uses a software-only license with no physical token. An SHK server can be shifted to another physical PC by moving the token(s) and reinstalling zPDT software. An LDK-SL license server cannot be moved to a different PC. To move the server to a different PC, you must obtain new LDK-SL server software. Also, additional license charges may be associated with the use of LDK servers; consult your zPDT provider for more details.

Restriction: LDK-SL server function is intended primarily for systems accessed in the cloud.

Figure 1 also indicates UIM components. UIM means Unique Identification Manager; this is a function that provides a consistent z System serial number to zPDT. The UIM function can be used with remote UIM servers. In principle, these are separate servers from the license servers and might be on different Linux PCs. In practice, the remote UIM servers are almost always installed on the same Linux PC having the remote license server. These topics assume that a UIM server is installed concurrently with an LDK-SL or SHK license server. There is also a local UIM component with operational zPDT systems (clients) not indicated in the figure.

A *license server* is accessed (via TCP/IP) by a *client* PC running zPDT and the zPDT operational license is supplied this way. The licenses needed to decrypt z/OS IPL volumes are also provided by the server. The client machine does not have a token

and does not need a USB port. A client machine must have access to the license server as long as zPDT is operational on the client. Likewise, the client machine has access to a UIM server that supplies consistent serial numbers for the z Systems CPs.

All zPDT systems have remote client functionality but, by default, it is not configured for remote operation. If a token is installed zPDT operates normally (with a local token). If a remote client function is configured, then zPDT attempts to connect to remote servers to obtain a zPDT license and serial number.

The owner of the client machine must do some minor configuration work to enable clients to use remote license servers and UIM servers; the enabling this interface differs for SHK and LDK-SL servers. Before enabling client access to a remote server the server networking environment (IP address, domain name, firewall controls, appropriate tokens for the server) must be arranged.

The remote license and UIM servers are normally on a single remote system. However, the two servers could be on separate machines. A UIM server and/or an SHK server could be on the same machine as the client, but would still be considered remote servers in the context described here. All the following text assumes that the license server and the UIM server are on the same machine. An LDK-SL server cannot be present on the same PC running zPDT.

Tip: The LDK and SHK terminology represents different generations of license management functions from Safenet, with LDK being the newer technology. (The company is now owned by Gemalto, but these help topics continue to refer to the Safenet "token" products.) The LDK technology can use both "software license" (denoted by LDK -SL) or new hardware tokens (denoted by LDK-HL). At the time of writing, zPDT does not use the newer hardware tokens (LDK-HL).

Concepts

z Systems CECs have unique serial numbers, allowing software to identify the machine and LPAR. Some operating systems verify that the "IPLed" machine has the same serial number as the machine that last used that copy of the operating system and may react differently if there is a mismatch. Some software products are licensed by machine serial number.

A simple zPDT system has a simple unique serial number design: the serial number of the zPDT token becomes the serial number of the z Systems created by zPDT. Figure 2 on page 11 illustrates the conceptual operation.

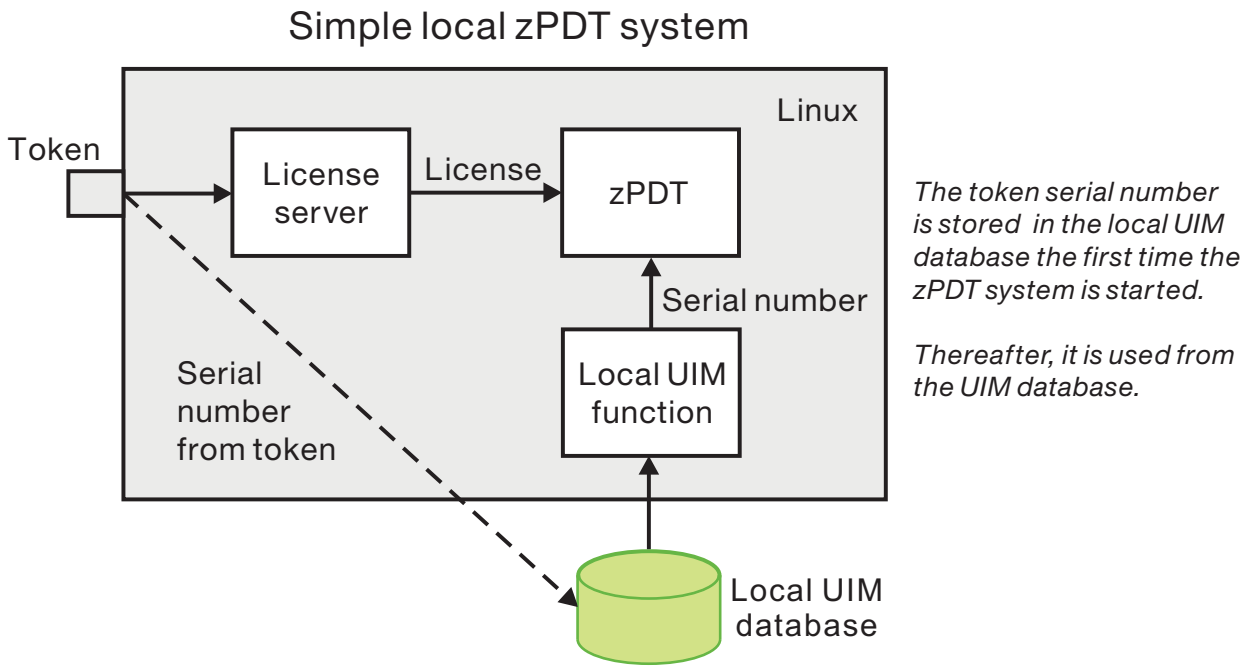

Figure 2. Simple local operation

When a remote license server is used (or if multiple local tokens are used) there needs to be a method of assigning unique serial numbers that do not change after they are assigned. Figure 3 on page 12 illustrates the general concept.

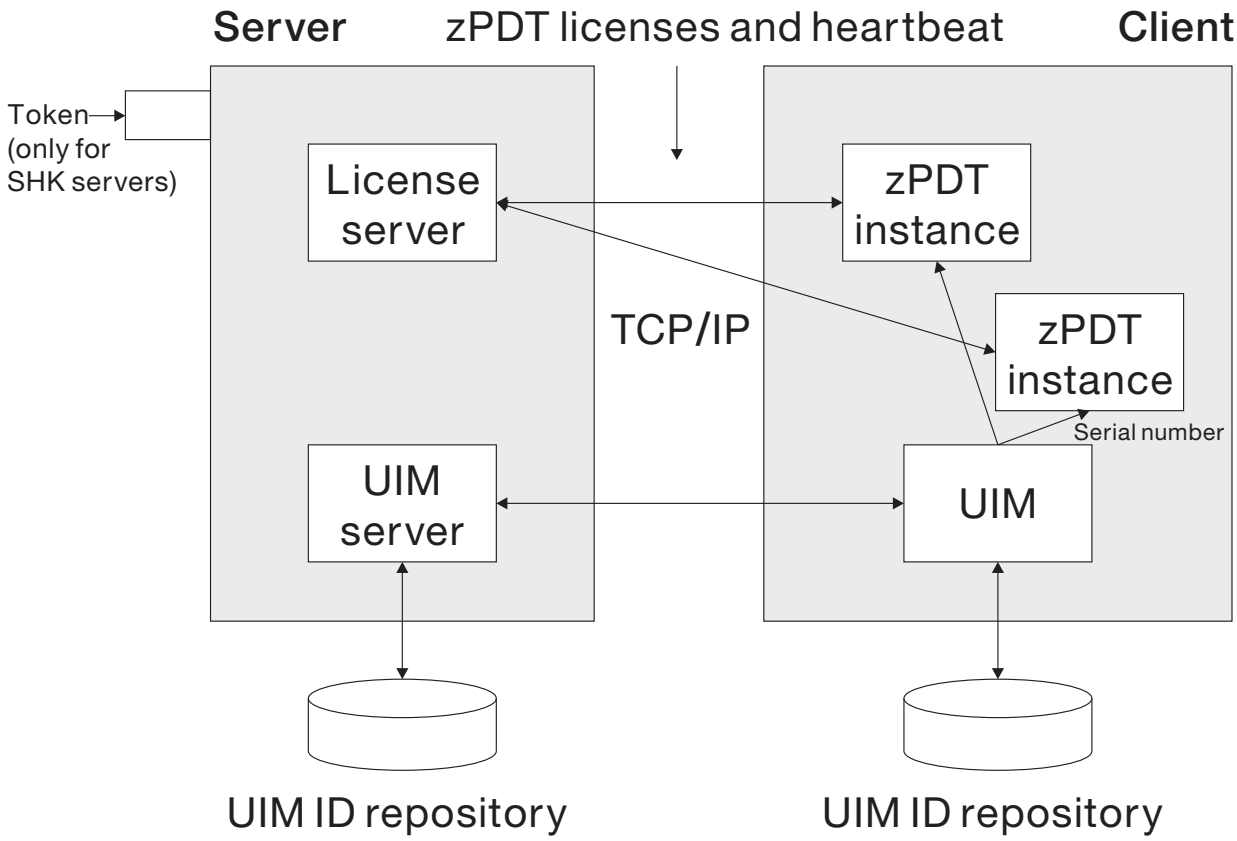

Figure 3. License and UIM servers

There are two modes of operation, *local* and *remote*. In the simple local mode both the license function (by a local token) and the UIM function run in the same machine as the client, as shown in Figure 2 on page 11, but are generally invisible to the user. An LDK-SL server cannot be used in local mode. In remote mode, the license server and UIM server program are in a remote machine which can serve licenses and serial numbers to a multitude of clients via TCP/IP.

Each Linux zPDT instance is assigned a unique serial number, either from a local token or by a UIM server. Every zPDT instance (running under a Linux user ID) has an LPAR ID assigned to it. An *instance* refers to multiple zPDT copies used on a base Linux system. The *LPAR ID* is not the same as the LPAR name. The *LPAR name* is the same as the Linux user ID that started the zPDT instance. zPDT instances have some of the characteristics of an LPAR, but full LPAR functionality is not provided by zPDT. The combination of serial number and LPAR ID becomes part of the CPUID. The CPUID is the information provided by the z Systems instruction Store CPU ID (STIDP).

Once assigned a serial number, the number is not changed even if the corresponding token (or software license) numbers are changed. The user must use the **uimreset** command to allow a serial number change. A user cannot assign an arbitrary serial number; the serial numbers are generated by UIM or taken from a token.

There are several notes relevant to Figure 3:

- The default port number for the SHK remote license server is 9450 and the default port number for the UIM server is one greater than the license server port number (and is 9451 by default) . The port number for the LDK-SL remote license server is 1947. The SHK server and UIM server port numbers are configurable; the LDK-SL port number is not configurable.
- After a zPDT instance is started (on a client) access to the UIM server is no longer needed.
- After a zPDT instance is started (on a client) the license access must be maintained for the life of the zPDT instance. If the access is dropped, the zPDT instance stops. (If the access is recovered, zPDT starts again.)
- The servers must be identified by resolvable domain names or by IP addresses. This is easy if they have direct, fixed IP address or domain names. It is not easy if DHCP-assigned addresses or NAT functions or VLAN networks are involved. Skilled network planning is required for any but the simplest environments.
- As a general statement, any PC system that can access the IP subnet of a license server can obtain a zPDT license there. Network security and license server security configurations may be important. This aspect is further described in "Security" on page 23.

Firewalls between the servers and clients must allow the required IP and port access.

- A client machine may be changed to a stand-alone machine (with token) by changing a configuration file, and vice versa.
- In normal operation, a client machine always has the same z Systems serial number. This number, once assigned via a local or remote function, might not be related to any physical token number.

Any license or UIM configuration changes should be made when zPDT is not operational.

The rules for obtaining a zPDT license are straight-forward. Either a local token is used or a remote license server. The indicated source must have an appropriate token or software license pool to provide a zPDT license.

The rules for zPDT serial numbers are more complex. The goal is to always have the same unique serial number for a given zPDT instance. The following general rules are used to determine the z Systems serial number for a zPDT instance. The term *UIM serial number* means a serial number generated and assigned by a UIM server. The term *random serial number* is also used for serial numbers created by a UIM server. After a random serial number is generated and assigned to a client, it is used consistently. The term *random* applies only to the initial generation of a serial number by a UIM server and indicates that the serial is not related to a specific token serial number. You cannot create the random number yourself.

If a single local token is used (and no previous serial has been assigned):

- The first zPDT startup will take the z Systems serial number from the token. This serial number is then written in the local UIM database.
- Subsequent zPDT startups must use the same token.
- If a different token is used, the uimreset -l command must be issued first (before zPDT is started). This erases the existing serial number in the local UIM database, allowing a new token (with a different serial number) to be used.

- Or, the RANDOM parameter may be specified in the XML configuration file. This allows any token to be used with an existing serial number in the local UIM database. (The **clientconfig** command is used to change this parameter.)

If a single local token is used and if a UIM serial number is present in the local UIM database (due to a previous connection to a UIM server) then the UIM serial number is used and the local token serial number is ignored. (The local token still supplies the zPDT license unless a remote license server is configured.)

If multiple local tokens are present (and no previous serial number exists in the local UIM database) the serial number of one of the tokens is accepted and stored in the client UIM database. This stored serial number is used subsequently, without further reference to the serial numbers of the tokens. In this case the RANDOM option must be specified by using the **clientconfig** command.

If the client is configured for a remote UIM server the following information applies:

- If no serial number is known for the client system, the UIM server generates a random serial number, that is, one that is not related to a token serial number, and sends it to the client UIM database.
- If the local client UIM database already contains a valid serial number that does not conflict with another client's serial number (as stored in the UIM server database) that serial number is used.

  If the client serial number (in the client UIM database) conflicts with a serial number in the UIM server database, the client operation fails. In this case, the client system may use the uimreset -l command to remove the serial number in the local UIM database.

If the client changes to a local configuration after previously using a remote configuration the previously assigned serial number (from the remote server and stored in the local UIM database) is used. The local token serial number is ignored.

Client installation and configuration

All client functions (for both licenses and UIM functions) are included and installed by the zPDT installation package. Whether the remote functions are used depends on configuration file options. for more information about the basic zPDT client installation process see Chapter 5 of the zPDT Guide and Reference.

SHK client configuration

After a nomal zPDT installation, SHK client operation is configured by settings in file `/usr/z1090/bin/sntlconfig.xml`. This file is referred to as the XML file.

This example shows the general syntax of this file. The actual XML file might have different spacing and more comments than shown here.

```
<SentinelConfiguration>
     <SentinelKeys>
          <ContactServer>localhost</ContactServer>
          <ServerPort>9540</ServerPort> <Protocol>SP_TCP_PROTOCOL</Protocol>
     </SentinelKeys>
     <UniqueIdentificationManager>
          <UIMContactServer></UIMContactServer>
          <UIMServerPort></UIMServerPort>
          <UIMProtocol></UIMProtocol>
          <UIMLocalSerialMethod></UIMLocalSerialMethod>
     </UniqueIdentificationManager>
</SentinelConfiguration>
```

Do not modify this file directly. Direct editing of an XML file is error prone and can be difficult to debug. Instead, use the **clientconfig** command to make changes. You must operate as root to modify the file or to use the **clientconfig** command. This command produces a display similar to this example:

```
Gen2 ContactServer....... _________________ (default is blank)
Gen2 BackupServer........ _________________ (default is blank)
Gen1 ContactServer....... _________________ (default is blank)
Gen1 BackupServer........ _________________ (default is blank)
UIM ContactServer........ _________________ (default is blank)
UIM Local Serial Random.. _ (y or blank)
Factory Reset............ _ (Enter "y" to reset file)
```

To change parameters values, overtype them.

- Leave Gen2 Contact Server empty. This field is used for software-based license manager.
- Leave Gen2 BackupServer empty. This field is used for software-based license manager.
- Set Gen1 Contact Server to point to IP address or `hostname` of product license server.
- Set Gen1 BackupServer if you want to set up the failover product license server. Set this field to point to the IP address or `hostname` of failover product license server.
- Set UIM ContactServer if you want to use different UIM server than product license server. Otherwise, leave this field empty. The field is empty.
- Set UIM Local Serial Random to "y", or leave this field empty. This field is optional. The UIM Local Serial Random specification is needed if multiple tokens are used on a local client or if different tokens are used at different times.
- If the Factory Reset option is set to "y", all other parameters are ignored, and the XML file is restored to the original values shipped with zPDT.
- Press Enter twice to save these values.

Changes to the configuration file are not dynamic. They take effect only when zPDT is started.

By default, the clientconfig command operates on the sntlconfig.xml file located in directory /usr/z1090/bin. The file name `sntlconfig.xml` is constant, but you may specify an alternate directory location as an operand:

```
# clientconfig /my/special/directory/
```

LDK client configuration

After the normal zPDT package is installed the LDK client can be activated. The LDK client (and server) requires a 32-bit version of the Linux glibc library and the client installation process automatically accesses several Internet sites to obtain the latest version of this library.

Be certain you have a working Internet connection before starting this process. Your base Linux might already have glibc-32bit installed. If glibc-32bit is not already installed on your base Linux and if you cannot connect to the Internet (perhaps due to firewalls) then you must obtain and install glibc-32bit in some other way. The LDK functions (client and server) will not operate without this rpm. After checking your Internet connectivity, and working as root, issue this command:

```
# /usr/z1090/bin/gen2_init
```

The resulting display depends on your Linux distribution, but might look like the following:

```
[root@zdt-dev3 ~]# /usr/z1090/bin/gen2_init

 Script for installing 32-bit compatibility packages for 64-bit Linux.
 Copyright (C) 2013, SafeNet, Inc. All rights reserved.

   Detected glibc 32bit support already installed

 Installing LDK client side license daemon  ....
Preparing...                          ################################# [100%]
Updating / installing...
   1:aksusbd-7.40-1                   ################################# [100%]
Starting aksusbd (via systemctl):                          [  OK  ]

..Done.
```

This setup is done only once. Thereafter the LDK client is started automatically when the client Linux system is booted.

Issue `./query_license` in the path `/opt/IBM/LDK` to see if the client is correctly configured and able to get license during IPL⌂

Client UIM configuration

The client UIM information is held in `/etc/z1090/uim/uimclient.db`. In unusual error situations you might be advised to delete this file. Deleting this file causes the UIM function to obtain or create a new serial number (working with your local token or with a remote UIM server) when zPDT is next started.

To configure the client UIM:

- For a local token client, the UIM function is normally transparent; no action is needed.
- For an SHK license server, the license server configuration (with the **clientconfig** command) also configures access to the UIM server. By default, the UIM server is assumed to be at the same IP address as the SHK or LDK server..
- To set up the different UIM server than the SHK/LDK server, set the UIM Contact Server field.

  For more information about license server configuration, see “SHK client configuration” on page 14 and “LDK client configuration” on page 15

Server installation and configuration

Both the SHK license server and UIM server are included in the standard zPDT package. The license server runs as a daemon and is automatically started when Linux is booted. (This is true even for local token use.) Both servers are TCP/IP users and your network configuration (including firewalls) must allow connectivity to the servers. The default port numbers are 9450 (license server) and 9451 (UIM server).

The LDK-SL license server and UIM server are not part of the standard zPDT package. A separate package with these two components is available as a separate offering. The LDK-SL offering is only for z Systems Development and Test Environment customers.

UIM server

The UIM server is automatically installed when installing either the SHK or the LDK-SL license server packages provided for zPDT.

Once installed, the remote UIM server must initially be started manually; thereafter it is automatically managed by cron. It must not run as root. It runs under a normal Linux userid and places its database in the home directory of that userid. It also places small log files in the home directory. For this reason, the same Linux userid (not root) should always be used to run the UIM server.

Two commands are associated with running the UIM server:

```
$ uimserverstart
    Start the UIM server.
$ uimserverstop
    Stop the UIM server.
```

The **uimserverstart** command, in addition to starting the server, places entries in the Linux cron files such that the UIM server is restarted automatically (after 10 minutes) if it fails. It is also started automatically during a Linux reboot. The **uimserverstop** command stops the server and removes these cron entries.

No other configuration is needed for the UIM server. You must not edit the UIM database file that is created in a subdirectory of the home directory of the userid running the UIM server.

SHK license server

The SHK license server is part of the standard zPDT package and is installed as if you were installing a zPDT client. It is activated by the actions of the two token "driver" components that are part of zPDT installation.

One or more 1090 or 1091 tokens must be installed in the license server machine before it can be used. The license server configuration file is located in: `/opt/safenet-sentinel/common_files/sentinel_key_server/sntlconfigsrvr.xml`

This file typically does not require any additional configuration. If you want to change the license server port number, you can edit and change this file. You would then need to restart the server by using these commands:

```
# cd /opt/safenet_sentinel/common_files/sentinel_keys_server

# ./loadserv restart
```

Several security functions may be specified in the `sntlconfigsrve.xml` file.

LDK-SL License server

Several steps are involved in preparing an LDK- SL license server. The license server (and the associated UIM server) are supplied in a file with a name similar to that shown in the following command.

Place this file in a convenient directory and, working as root, execute this file:

```
#          ./zPDT_LS-1.6.49.20L-x86_64
```

This file must be executable. This might require a **chmod u+x** operation. Also, the exact file name may change slightly to match newer levels of zPDT.

The installation process causes an Internet search for the latest version of the 32-bit glibc library, as described in "LDK client configuration" on page 15. Both the LDK-SL license server and a UIM server are installed. The LDK server is installed in `/opt/IBM` instead of the traditional `/user/z1090/bin` that was used for other zPDT modules. The next step is to obtain licenses that can be "served" by the license server. Working as root, issue this command:

```
#
          /opt/IBM/LDK/request_license
```

This command creates a file named *hostname_xxxxxx*.zip in root's home directory, where *hostname* is your Linux system's name and *xxxxxx* is a timestamp. This file contains a fingerprint of the license server. You must send this file to the appropriate IBM licensing facility (as identified by your zPDT contract). In return you will receive a "v2c" file containing the number and type of licenses your server can supply to clients. Receive this file into a convenient directory and install it by using this command:

```
# /opt/IBM/LDK/update_license hostname_xxxxxx_update.v2c
```

Then restart the license server daemon by using one of these commands:

#systemctl restart aksusbd.service
: Used with newer Linux distributions.

 # service aksusbd restart
: Used with older Linux distributions.

This completes the LDK-SL license server installation. You might need to start the UIM server on your server system.

The v2c file that conveys licenses to the server also contains ADCD decryption licenses that become available to the client systems.

Notes

Learn about UIM function commands, how to manage firewalls involved with remote servers, how changing the Linux disk (HDD) might change the identifier that is part of the identification used by UIM, cloning a zPDT system, and removing zPDT functions.

UIM function commands

Several commands are associated with the UIM function.

 # uimreset [-l] [-r]
: This command clears the serial number in the local UIM database [-l] or in both the remote and local UIM database [-r]. This command must be run by root.

  If you decide to "start all over" and reinstall your zPDT system, there might be a problem with serial numbers. If you use the same single local token that was used previously, zPDT will obtain the same serial number from it. If you use a remote license server and used the **uimreset - r** command to delete any previous references or if you have multiple local tokens your new zPDT installation might not have the same serial number as the previous setup. If you do not care about z System serial numbers then this is not a problem. If you do care about z System serial numbers because of software contracts or software sensitivity this can be a problem. The only certain way to obtain the same z System serial number is to use the same single local token.

$uimserverstart and $uimserverstop
: These commands start and stop a UIM server. A UIM server can run under any user ID (except root) on the server machine, but it should always be the same user ID. These commands are not normally used in a purely local client environment. These commands cannot be run by root.

$ uimcheck

The **uimcheck** command should be used if there is any question about the state of the serial number on a zPDT machine. Any user may issue this command.

SecureUpdateUtility, Z1090_token_update, and Z1091_token_update

Do not run **SecureUpdateUtility** or **Z1090_token_update** from a client zPDT machine when using a remote license server. This utility cannot affect tokens or licenses in the remote license server, but will attempt to access a token in the local PC. You may run **SecureUpdateUtility** or **Z1090_token_update** in the SHK license server, to update the tokens in the server. Normal guidelines for **SercureUpdateUtility** or **Z1090_token_update** and **Z1091_token_update** apply. For example, only one token should be connected to the PC when you use these commands.

The administrator of a license server is responsible for ensuring the license keys do not expire while in use. The situation in which multiple tokens are installed (in an SHK license server) and the licenses in one token expire can be complex. Clients see license expiration warning messages starting a month before the license expires. However, if multiple tokens are present it is not predictable which token will furnish the license (or licenses) for a zPDT startup.

$ token

The license expiration date displayed by the **token** command (in a client machine) may not reflect the effective expiration date of all the active tokens in a license server. The **token** command (when zPDT is running) produces additional information:

```
CPU 0, zPDTA (1090) available and working. Serial 6186(0x182A)
Lic=88570(0x159FA) EXP=4/15/2017
```

In this example, the zPDT license was obtained from token 0x159FA (decimal 88570) and the CP serial number used by zPDT is 0x182A. There is no indication of whether a license server and UIM server are being used. Because the serial number and license number are different, we know that at some point the serial number was obtained from a license server. However, it is possible that the token is in the local client but that the serial number previously obtained from a UIM server is being used. This fulfills the goal of using a consistent serial number once it is assigned.

Firewalls

You or your network administrators must manage any firewalls involved with remote servers. If you are initially installing in a test environment, disable all firewalls until you are satisfied with your zPDT license operations. Disabling firewalls helps distinguish network-related problems from license manager issues. If you operate through firewalls you must ensure that the relevant port numbers can pass through the firewalls.

There are many management techniques for firewalls, depending on what product is being used. Many Linux systems respond to **iptables** commands:

```
# iptables -I INPUT -p tcp --dport 1947 -j ACCEPT
# iptables -I INPUT -p tcp --dport 9450 -j ACCEPT
# iptables -I INPUT -p tcp --dport 9451 -j ACCEPT
```

Disk and Linux changes

Changing the Linux disk (HDD) might change the identifier that is part of the identification used by UIM. After changing the hard disk, you might need to use

the **uimreset -l** command to reset the local serial number or the **uimreset -r** command to reset the remote serial number.

Upgrading to a new Linux kernel might change the identification used by UIM. You might need to reset the local serial number or the remote serial number. If this does not solve the problem, delete the UIM database at `/usr/zpdt/uim`.

Cloning zPDT

If you clone a zPDT system, you must delete the files in `/usr/z1090/uim` on the new system. This is because the UUID of the new system differs from that of the old system. zPDT will build new uim files when the new system is started.

Removing functions

Use these commands to remove zPDT functions.

All SHK server functions (and associated UIM) can be removed by simply removing zPDT on that server. You can use either of these methods. In the first command, notice that the **--removeall** parameter is preceded by two dashes.

```
# z1090-1-6-49.17.x86_64 --removeall
# rpm -e z1090
#       rpm -e z1091
```

The LDK client function can be removed with a command like the following (where the exact file should match whatever name was used to install the LDK client function). Notice that the **-- remove** parameter is preceded by two dashes.

```
# /usr/z1090/bin/LDKc_setup.sh --remove
```

An LDK server is a normal rpm package that can be removed with this command, which also automatically removes the UIM server that was associated with the LDK server.

```
# rpm -e aksusbd
```

Scenarios

Learn about common usage scenarios.

License search order

zPDT attempts to obtain a license from an LDK server if one is configured, then attempts to obtain a license from an SHK server if one is configured, and lastly attempts to obtain a license from a local token. There is a considerable timeout involved in trying to access the two servers and depending on this automatic search order is not reasonable for normal operation. The **--localtoken** option of the **awsstart** command simply short circuits any attempts to use remote license servers.

Switch from local to remote server

Learn how to switch from a local to a remote license server.

In this scenario, two zPDT systems, A and B, each use a different PC for zPDT. System A has a zPDT token with serial number 12345.

1. The system A owner installs token 12345 in the PC and starts zPDT. When this is done, serial 12345 is recorded in the local system A UIM database. This scenario assumes no prior conflicting information was in the local UIM database. System A can be used in this configuration indefinitely until the token license expires with no reference to remote license or UIM servers.

2. The token is taken from system A for some reason, and the system A owner now wants to use remote license and UIM servers. With zPDT not running and working as root, the owner configures a client as described in "SHK client configuration" on page 14 or "LDK client configuration" on page 15.
3. The remote UIM server with the SHK or LDK-SL server, whichever one is being used sees that system A has serial number 12345 recorded in its local UIM database. The server checks whether this serial number is assigned to any other system. If there are no conflicts, the server records serial 12345 in the server database as belonging to system A.
4. Separately, the remote license manager serves a zPDT license based on a token present in the license server machine if it is using SHK, but the serial number of that token is not relevant.

   So far, system A retains a consistent serial number, 12345, when switching from a local token to remote token or UIM servers. It retains this serial number every time this zPDT instance is used. If Multiple zPDT instances run on the same machine, they must run under different Linux user IDs. The serial number for each of the instances uses the LPAR portion of the serial number to differentiate the instances.
5. If token 12345 is transferred to the owner of system B, and the owner installs and uses it locally with no connection to the remote license or UIM servers, then both A and B have the same zPDT serial number. There is no way to avoid this.
6. If the system B owner then connects to the license or UIM servers, the UIM server sees serial 12345 in B's local UIM database and terminates the zPDT instance because 12345 has already been assigned to system A.

   This scenario is problematic because both A and B are attempting to use the same serial number, 12345, but the UIM server has it assigned to A. There are two ways to resolve this conflict:

   - The system B owner can issue **uimreset -l** to clear the serial number in the local UIM database. The owner can then connect to the remote servers and receive a new random serial number.
   - The system A owner can issue **uimreset -r** to clear the system A serial number from both the local and remote UIM databases. The next time system A zPDT starts, it requests a new random serial number from the server. System B can then use serial number 12345, which is stored in its local UIM database.

Temporarily switch from server to local

Learn how to temporarily switch a client from a remote license and UIM server to local license.

In this scenario a notebook zPDT system is normally used with remote license and UIM servers. The owner wants to take the system home overnight, but the servers cannot be accessed from home.

If a token is available, you can start zPDT with the local option:

```
$ awsstart devmap_name --localtoken
```

In this case there is no need to use the **clientconfig** command to change the configuration file. The **--localtoken** option overrides the configuration file. The user must, of course, have a token to supply a license. In this case the serial number stored in the local UIM database is used and the serial number of the temporary token is ignored.

Switch from remote server to local

Learn how to switch from a remote license and UIM server to a local license.

A system owner has been using a remote SHK license server and UIM server. To change to a local token, the owner used the **clientconfig** command to change the `LicenseContactServer` value to `localhost`. This command has the following effects:

- It effectively removes the `UIMContactServer` stanza from the XML file. The absence of this stanza indicates that no UIM server is to be used.
- In this case, zPDT looks in the local UIM database for a serial number. If one is present, it is used. If the local UIM database does not exist, or if the **uimreset -l** command was used, the serial number of the local token is placed in the local UIM database and then used by zPDT.

Using zPDT on the license and UIM server

Learn how to run zPDT on the same machine that is running the SHK license server and UIM servers.

To run zPDT on the same machine that is running the SHK license server and UIM servers, use the **clientconfig** command to specify `LicenseContactServer` as `localhost` and `UIMContactServer` as `localhost`. This command has these effects:

- The presence of the `UIMContactServer` stanza means that a UIM server must be available on the indicated system, which is `localhost` in this example. Before starting zPDT on this system the user must issue a **uimserverstart** command.
- Give some thought to the Linux user ID that issues the **uimserverstart** command. The same user IDd must always be used for this command because the UIM server database is created in the `home` directory of this Linux user ID.
- No special setup is needed for the license server. Any zPDT system, meaning the SafeNet server that is installed with zPDT, can act as a license server.
- Combined operation, as server and client, is not possible with an LDK server.

Switching tokens

Learn how to switch tokens in a local UIM database.

In this scenario, token 12345 is used with a newly installed zPDT system. When zPDT is first started, this serial number is written in the local UIM database. If a different token is used on a subsequent startup, the zPDT startup fails. Use the **uimreset -l** command to remove serial 12345 from the UIM database. After the original serial number is removed, a new token can be used.

If the serial number in the local UIM database was assigned by a UIM server, or if the **RANDOM** parameter was used with the **clientconfig** command, then any local tokens can be used; the operational serial number is taken from the local UIM database.

The important point is that zPDT recognizes the difference between a UIM server-assigned serial number, which can be used with any token, and a locally installed serial number, which is taken from a local token. A locally installed serial number must match the token being used, unless the **RANDOM** option is set.

Change from single token to multiple tokens

Learn how to switch from a single token to one of several other tokens.

You can switch from a single token one of several tokens. This procedure assumes that you are not using a remote license server.

1. From a user ID with root authority, enter a **uimreset -l** command.
2. Use the **clientconfig** command to set the UIM Local Serial Random value to Y.
3. Select the token containing the serial number you want assigned to the zPDT system. Start zPDT using this token.

Now you can start zPDT with any token. The serial number you selected in step 3 is used, regardless of which token you are currently using.

Display serial number assignments

To display the zPDT serial number assignments, open a browser to the remote UIM server (http://*uimserveraddress*:9451). Port 9451 is the default UIM port.

The browser displays information similar to this example:

Serial	Host	UUID	Year	Day
2099	hostname.domain.com	E6D96D01-493E-11CB-AD29-B8F42F7F8461	2016	009

Security

If the license managers are used only from a single subnet, or a well-designed VPN, then security is not a major issue. If the license servers are accessed from the general Internet then security can be a significant issue. For example, your license server could provide zPDT licenses to someone completely unassociated with your enterprise.

SHK server:

The SafeNet SHK license server can have three lists of IP addresses, domain names, or ranges of IP addresses.

- The Authorized User List determines which systems can use a web interface to manage the SafeNet license server. The default list contains only one address: 127.0.0.1, which is the local host and is always allowed whether specified or not.
- The Allowed Site Address list determines which clients can obtain zPDT licenses from the server. If the list is empty (the default) then any client can obtain a license from the server.
- The Blocked Site Address list specifies client addresses that canot obtain a license from this server. If the list is empty (the default) then no client addresses are blocked.

Each list is limited to 32 entries. These lists are in the `sntlconfigsrve.xml` file in `/opt/safenet_ sentinel/common_files/sentinel_ keys_server/` and can be edited there. They can also be managed by opening a browser on port 7002 on the machine running the SafeNet license server:

```
http://localhost:7002
```

Restriction: The browser function provided by SafeNet appears to depend on specific Java™ levels. It might not work with the default Java level on current Linux systems.

If a different machine is used to access the server web interface, then the IP address of that machine must be listed in the Authorized User List. Use the

browser method, if possible, because directly editing this XML file is prone to introducing syntax errors that might cause the license server to fail. List entries might take any of these forms:

127.0.0.1
A simple IP address.

my.local.domain.com
A domain name.

10.1.1.2-10.3.255.254
A range of domain addresses.

If you are using the browser interface, be certain to click the update button on the web page after entering updates to the lists. You must then restart the SafeNet server:

```
# cd /opt/safenet_sentinel/common_files/sentinel_keys_server

# ./loadserv restart
```

These lists provide one way to secure use of a zPDT license server. Other methods, such as restricted router interfaces or nonroutable IP addresses, might be more appropriate.

LDK-SL server:

You can use the browser interface to control access to the server.

Open a browser on URL address localhost:1947. In the Configuration page, under Access from Remote Clients, you can enter Access Restrictions. These can consist of mixtures of IP addresses in either numeric or domain name form, and user IDs. The browser pages contain useful help information for this functions. The security list is evaluated in the order of the statements. This example denies licenses to all requesters who are not on the 9.12.45.* subnet.

```
allow=9.12.45.*
deny=all
```

Firewalls:

Working with the zPDT default port numbers, a firewall on a license and UIM server must allow connections to ports 9450 and 9451. One solution is to simply disable the firewall on the license server. Another solution is to enable the firewall and open the required ports.

To enable the firewall and open the required ports, issue these commands. These commands must be entered from a root user ID after the server Linux system is booted.

Important: Network management skills are needed to properly implement the server functions.

```
# iptables -I INPUT -p tcp --dport 9450 -j ACCEPT
# iptables -I INPUT -p tcp --dport 9451 -j ACCEPT
# iptables -I INPUT -p tcp --dport 1947 -j ACCEPT
```

Resetting UIM

You can usually remove the local UIM serial numbers with the **uimreset -l** command. You can remove both the local UIM serial numbers and corresponding entries in the UIM server database with the **uimreset -r** command.

If the local UIM database is corrupted, the **uimreset** command might fail. You can delete the files in the /usr/z1090/uim directory. However, the previous UIM serial for the client is still provided by a UIM server if the client XML file is configured for connection to the server. In this case, you can use the **uimreset -r** command to remove the relevant entry from the UIM server database.

The UIM server can be reinitialized by removing everything in the UIMserver subdirectory in the home directory of the Linux user ID that runs the UIM server. This action should not be done in normal operational environments. If the UIMserver directory is cleared, some of the entries will be restored by future client connections in which the client still has previous UIM local data.

The client configuration file can be restored to its original state, which does not reference any remote servers, by using the Factory Reset option with the **clientconfig** command.

Restarting Safe Net modules

Two SafeNet functions are involved with zPDT: the SHK or LDK-SL license servers and a daemon, or token driver, that communicates with tokens in USB ports. After zPDT is installed, both these functions are started automatically when Linux is started. Changing the license server files requires restarting the license server. It should not be necessary to restart the token driver except in unusual situations.

To restart the USB token daemon, enter these commands from root:

```
$ su
#cd /opt/safenet_sentinel/common_files/sentinel_usb_daemon
#./load_daemon.sh   restart
```

To restart the SHK server, enter these commands:

```
# cd /opt/safenet_sentinel/common_files/sentinel_keys_server
# ./loadserv restart
```

To restart the LDK-SL server, enter this command on newer Linux distributions:

```
# systemctl restart aksusbd.service
```

Enter this command on older Linux distributions:

```
# service aksusbd restart
```

Renewing licenses

zPDT licenses in a token or in a software license server are usually valid for a year and must be renewed after that time. The procedure for renewing token licenses varies with different categories of zPDT users.

Users with 1090 tokens (typically ISVs or IBM internal users) normally create a request file by using the **Z1090_token_update -r** command and send this file to their zPDT provider. The provider, in turn, returns an update file that is installed with the **Z1090_token_update -u** command. This example shows one sequence of commands for requesting and installing the update file.

To request an update file, be sure that only a single token is connected to the computer. zPDT must be stopped and you must be logged in as root and in the /usr/z1090/bin directory.

```
$ awsstop
$ su
# cd /usr/z1090/bin
# Z1090_token_update -r mytoken.req
```

In this sequence of commands, the file name `mytoken.req` is an arbitrary name that you provide for the request. The `mytoken.req` file is sent to the zPDT provider. The provider returns the file `mytoken.zip`.

To process the returned file, you must be logged in as root in the `/usr/z1090/bin` directory. zPDT must be stopped .

```
# Z1090_token_update -u mytoken.zip
# exit
```

Older versions of zPDT use the **SecureUpdateUtility** command instead of **Z1090_token_update** and work with `.upwreturned` files instead of `.zip` files.

The returned `.zip` file also installs the token licenses needed to decrypt z/OS IPL volumes. After installing the new file, the token must be removed from the computer for about 15 seconds. Removing the token forces the token drivers to reread the token when it is reinstalled.

For IBM internal users, the provider is Resource Link®.

Users with 1091 tokens, who are typically z Systems Development and Test Environment users, might have a somewhat different process that does not require a request file. This process sends the customer a customized `.zip` file, which is installed by using this sequence of commands. Be sure that only a single token is connected to the computer. zPDT must be stopped and you must be logged in as root and in the `/usr/z1090/bin` directory. In this sequence of commands, *xxxxx*`.zip` is a file name assigned by the zPDT provider.

```
$ awsstop
$ su
# cd /usr/z1090/bin
# Z1091_token_update -u xxxxx.zip
# exit
```

After you install the `.zip` file, the token must be removed for about 15 seconds. Software licenses (for LDK-SL servers) are installed by installing a new v2c file as described in “LDK-SL server” on page 24.

Licenses must be renewed on the computer that runs the server. You cannot renew or update the token licenses remotely. The **Z1090_token_update** command or the older **SecureUpdateUtility** command work only with a single token installed in a local USB port.

Searching servers

More than one SHK or LDK-SL server can be specified for a client. Additional servers are simply listed by domain names or IP addresses in the respective client setup for the two license servers. The servers are searched for an appropriate license in the order listed. There is no coordination among the servers; each must have available licenses in the form of additional tokens for SHK servers or software entitlements for LDK-SL servers in order to serve them to clients. This means that the customer installation has purchased additional licenses or has split the available licenses among multiple servers in some way.

A zPDT client searches all available license sources until it finds the licenses it requires. If any LDK-SL servers are defined for the client, they are searched first, followed by SHK servers, followed by locally installed USB tokens. If remote

license servers are defined for a client but cannot be accessed by a TCP/IP connection, there will be delays while the access attempts timeout before another license server is tried.

If multiple license servers are routinely used it is possible, but unlikely, that a duplicate UIM serial number could be assigned to zPDT instances. If you use multiple license servers, reserve them for failover situations and not for routine use.

An LDK-SL license server cannot be shifted to another computer. Moving an LDK-SL license server function to a different computer involves multiple interactions with your zPDT license provider to ensure that the license entitlement information is removed from the old server and that a new license entitlement v2c file is created for the new server.

Allocating multiple licenses

Learn how the zPDT system allocates multiple licenses to clients.

This example uses a remote SHK or LDK-SL license server with five zPDT licenses to illustrate how it allocates licenses to clients. A single client could request all five licenses by coding `processors 5` in the devmap. Or five different clients could each request a single license. Or there could be a combination of clients that consume the five available licenses. When a client zPDT ends (with the **awsstop** command) the licenses used by that client are available to other clients. At any given instant no more than five zPDT client licenses, representing five CPs, can be allocated to clients.

Over time, many client zPDT systems might connect to this remote license server provided that not more than five licenses are allocated at any one time. Each of the many clients has a unique serial number provided by the remote UIM server. In this case, where five licenses are available , ten serial numbers can be associated with these five licenses. This distinction between numbers of licenses and numbers of serial numbers might be important for some ISV software license situations.

A single zPDT instance cannot have more than eight CPs, each requiring a zPDT license. IBM contract conditions might have a smaller limit. Assuming that the maximum of eight could be used, the devmap for an instance could request eight licenses from the remote server. In our example, only five licenses are available and the client would receive all five licenses (if no one else is using any licenses). Perhaps the intention of the customer is to share his five licenses among several development systems. There is no technical way to prevent a single user (that is, a single development system) from using all the licenses (up to eight, if that many are available). Management control is needed to ensure “fair” sharing of zPDT licenses in situations where a limited number of licenses are serving multiple remote clients.

Switch from Product License Server to License Manager

Learn how to migrate from a Product License Server (hardware-based license) to a License Manager (software-based license).

To migrate from hardware-based license to software-based license, follow below process step-by-step.

- “Returning any existing license key for the USB hardware device or license manager” on page 5.

You can return a license entitlement in the Rational License Key Center in several ways. The easiest method is to use the View Keys by Host link. Also, you can use the Return Keys link.

For z Systems Development and Test Environment, the term host in the Rational License Key Center refers to the USB hardware device that is uniquely identified by its serial number.

1. Log in to Rational License Key Center at https://licensing.subscribenet.com/control/ibmr/login, and select your account.
2. On the left side of the screen, select **View Keys by Host**.
3. Select the serial number of the USB hardware device you want to work with. This serial number is in the Host column.
4. A table is displayed with data for the USB hardware device selected. At the far right of the table, click the Change link.
5. A list of devices with license entitlements that are assigned to them from the same Order Line is displayed. Locate the serial number of the USB hardware device you are working with, and click Return. A message is displayed to confirm that the license entitlements were successfully returned.

- Uninstalling a previous version of z Systems Development and Test Environment.

  Before configuring the license manager, you must uninstall the product license server by following the steps in this topic.
- Installing and starting the license manager.

  To install the license manager, follow the steps in this topic.
- Issuing command to reset the UIM server.

  ```
  uimreset -l
  ```
- Activating a license manager.

  To activate the software license manager, follow the steps in this topic.
- Activating a license manager with Rational Tokens.

  If you want to use the Rational tokens, follow the steps in this topic.
- Activating and configuring a license manager client.

  To configure and activate the license manager client, follow the steps in this topic.

Accessibility features

Users who have a physical disability, such as limited vision, can review the available accessibility features to use their information technology products successfully.

Accessibility features are product dependent and might include one or more of the following aspects:

- Keyboard-only operation
- Screen reader usage
- Color and typeface preferences

Note: The accessibility features mentioned here apply to the Windows operating system. Some of these features might also work on Linux, but are not officially supported.

z Systems Development and Test Environment

You can run many optional, supporting programs on the z Systems Development and Test Environment emulator. A Voluntary Product Accessibility Template (VPAT) is available for an optional program upon request.

Keyboard shortcuts for the help system in the product

You can use shortcuts to control the help system by using the keyboard.

Table 1. Help system keyboard shortcuts

Key combination	Context	Function
F6	Anywhere in the help browser	Puts focus in the next browser frame.
F6+Shift	Anywhere in the help browser	Puts focus in the previous browser frame.
Right Arrow	Navigation tree	Expand section
Left Arrow	Navigation tree	Collapse section
Down Arrow or Tab	Navigation tree	Move to next topic node
Up Arrow or Shift+Tab	Navigation tree	Move to previous topic node
Enter	Navigation tree	Displays the selected topic in the Content frame
Tab	Content frame	Next link or toolbar icon
Home	Content frame	Move to top of frame
End	Content frame	Move to bottom of frame
Alt+Left Arrow	Content frame	Back
Alt+Right Arrow	Content frame	Forward
Ctrl+P	Content frame	Print

Example

To open a topic by using keyboard shortcuts and have the content read by a screen reader:

1. Start the screen reader application.
2. Open the help system in the browser.
3. Press F6 three times to move the focus into the Contents pane.
4. Press Tab to navigate to a container that you want to open. Example: Installing
5. Press the Right Arrow to expand the section.
6. Press Tab to navigate to a topic that you want to open.
7. Press Enter to open the topic contents.
8. Press F6 three times to move the focus into the topic pane. If you are using Mozilla Firefox, the topic contents are read by the screen reader application.
9. If you are using Internet Explorer, press the Down Arrow to make the screen reader application read the topic contents.

Index

A

accessibility
  keyboard shortcuts
    help system 29
  overview 28
Activating the USB hardware device 6

C

connections, troubleshooting Rational License Key Servers 7

D

device, activating 6

H

hardware device, activating 6
help
  keyboard shortcuts 29

K

key, activating 6
keyboard shortcuts
  help system 29

L

license key, activating 6

P

pointing to a Rational License Key Server 6

R

Rational License Key Server, pointing to 6
Rational License Key Servers, troubleshooting connections 7
Rational Tokens 5

S

shortcuts
  keyboard
    help system 29
support
  accessibility 28

T

Tokens 5
troubleshooting connections, Rational License Key Servers 7

U

USB hardware device 6
USB hardware device, activating 6