

The default JCL procedure name is PAGENT so the Policy Agent is often referred to as PAGENT.

# Trademarks

- **The following are Registered Trademarks of the International Business Machines Corporation in the United States and/or other countries.**
  - IBM
  - z/OS
- **The following are trademarks or registered trademarks of other companies.**
  - Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.
- All other products may be trademarks or registered trademarks of their respective companies.
- Refer to [www.ibm.com/legal](http://www.ibm.com/legal) for further legal information.
- OSA-Express Features
- There are many different types of OSA-Express features. In this document where OSA is mentioned it refers to an OSA port on any of the OSA-Express features unless a specific OSA-Express feature is mentioned.

## Agenda

- Structure of Policy Agent
- PAGENT Configuration Files
- Modify PAGENT and Monitor Applications
- Policy Views
- Configuration Assistant Tool
- Policy Server

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 3

Policy Agent is the component of z/OS Communications Server which is responsible for reading the policy definitions from a configuration file (pagent.conf) and/or an LDAP server and installing those policies into one or more TCP/IP stacks.

Each z/OS image has a single Policy Agent which installs the policy into one or more TCP/IP stacks running on that z/OS image.

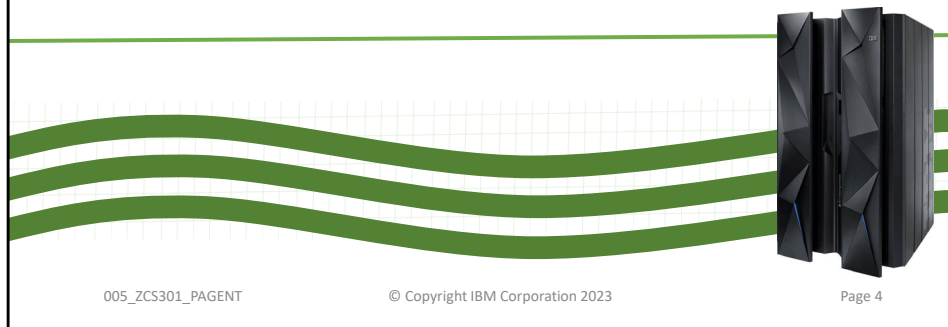
In this lecture we'll talk about the types of policies that can be defined.

We'll also look at how the Policy Agent is structured.

Then we'll see how to define and manage the network policies using the Configuration Assistant tool.

The note is to remind you that Policy Agent may be configured in a Central Policy Server structure which provides a centralized repository for all policies and simplifies management of policies. We'll talk about this later in the presentation and you can read all about it in the IP Configuration Guide and the TCP/IP Implementation red books.

# Structure of Policy Agent

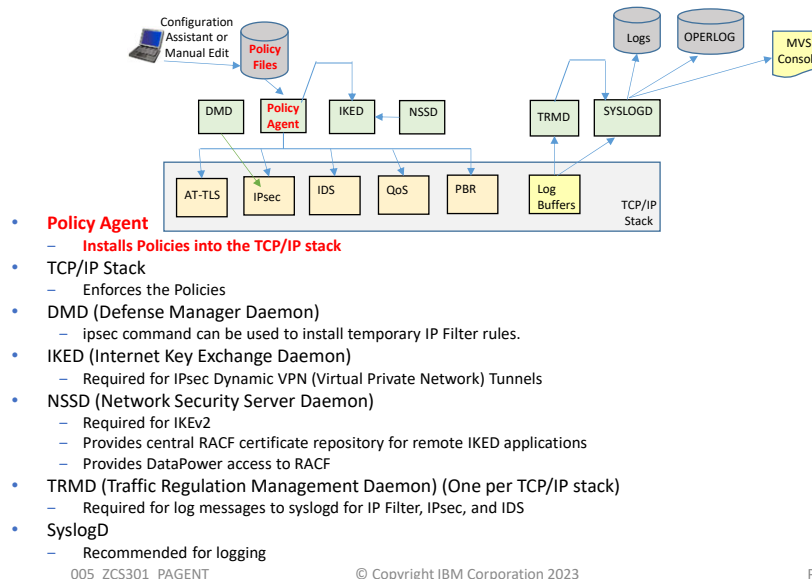


005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 4

## Lots of Different Policy Types and Started Tasks



Configuration Assistant can really help getting the whole environment setup.

Several different components or applications make up the policy infrastructure. Depending on the types of policies you use, some or all of these components might be needed.

The Configuration Assistant provides a user-friendly means to create policy definition and configuration files for most of the policy infrastructure components. You can also define these files manually.

The Policy Agent (pagent) reads and parses policy definitions, installs these policies in the TCP/IP stacks, and provides IPsec VPN-related policies to the IKE daemon.

The TCP/IP stack enforces the policies as needed to inbound and outbound traffic.

The Defense Manager daemon allows an external security monitor function to install temporary IP filters to block a specific attack or pattern of attacks.

The IKE daemon (IKED) provides support for dynamic and manual VPNs. It uses certain IPsec policies to communicate with its peers on other systems to establish secure tunnels to protect data being sent or received.

The Network Security Services server daemon (NSSD) provides certificate and network management services for IPsec, in addition to other services for XML appliances.

The Traffic Regulation Management daemon (TRMD) writes a variety of security related messages to syslogd. These records can be formatted by the trmdstat command to produce a variety of reports.

The syslog daemon (syslogd) provides a system-wide logging mechanism, and as such is not strictly related to policy. But several of the other policy components use syslogd as a logging mechanism.

The Network SLAPM2 subagent (nslapm2) helps to monitor the effectiveness of QoS policies.

## Policy Agent Prerequisites

- Syslogd for logging is recommended.
- Traffic Regulation Manager Daemon (TRMD) to collect information from the stack and send it to Syslogd or the console for policy types IP Filter, IPSec, and IDS
- RACF authorizations
  - PAGENT started Task OMVS segment with userid required
  - Commands that interoperate with PAGENT
    - psearch
    - ipsec
  - Some policies have additional access controls and definitions
    - AT-TLS
      - EZB.INITSTACK Servauth Class
    - IPsec
      - See details in the IPsec presentation
- How to configure the daemons?
  - z/OS Communications Server IP Configuration Guide SC31-8775
  - Communications Server for z/OS TCP/IP Implementation Volume 4: Security and Policy-based Networking SG24-7699, SG24-7801, SG24-7899

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 6

It is recommended that you have access to the two manuals listed on this page when setting up the Policy Agent security environment.

Policy Agent requires other daemons depending on which policies you have defined. For example, you will always need Syslogd for logging for all policy types. Syslogd is not part of Policy Agent but is very important to it in order to capture the logging and error messages that will aid in auditing and in performing problem determination.

Policy Agent requires various types of RACF authorization. Some of its policy types demand more authorization profiles than others.

A SERVAUTH class called EZB.INITSTACK is a useful definition as soon as AT-TLS policies are introduced. If you are using Application Transparent Transport Layer Security (AT-TLS), z/OS will not allow any socket-based applications to start before PAGENT is up and running so as to make sure that all the security policies are enforced. But some essential applications need to start before PAGENT. The EZB.INITSTACK authorization permits such applications to start even prior to PAGENT startup.

IPSec policies will require the IKE daemon if dynamic VPNs are involved. The IKE daemon manages the secure key exchange required to establish a dynamic VPN. You may also need NSSD if you are using a central key and certificate server.

For IP filtering you may also want to use a Defense Manager Daemon which will allow you to introduce filters immediately without having to build an IP Filtering policy..

TRMD is required for IDS, IP filtering and IPSec policy types.

## Policy Types

- IDConfig
  - Intrusion Detection Services (IDS) policies
    - Scan policies
    - Attack policies
    - Traffic Regulation policies
- IPSecConfig
  - IP Filtering policies
  - IPSec policies
    - Key exchange policies
    - Local dynamic VPN policies
    - Local manual VPN policies
- RoutingConfig
  - Policy-based Routing policies
- TTLSConfig
  - Application Transparent - Transport Layer Security (AT-TLS) policies
- QOSConfig
  - Quality of Service (QoS) policies
    - Differentiated Services (DS) policies or Data Traffic policies
    - Integrated Services policies or Resource Reservation Protocol (RSVP) policies
    - Sysplex distributor (SD) policies
- ZERTConfig
  - zERT Enforcement

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 7

Policy Agent supports several types of policies which are also called disciplines.

QOS policies: To establish "precedence bits" in the Type of Service field of the IP Header.

- High priority, medium priority, etc.
- Can influence which queues of the OSA adapter are used and which routing path a router may select for a packet.

IDS policies:

- To detect and protect against security intrusions.
- To report on security intrusions.

AT-TLS (Application Transparent- Transport Layer Security)

- For securing traffic with authentication, data integrity checking, and encryption.
- Unlike application specific SSL/TLS, security actions are performed by the Transport Layer and do not require applications to be rewritten to invoke security APIs.
- Invoked from client to server for every TCP connection.

IP filtering

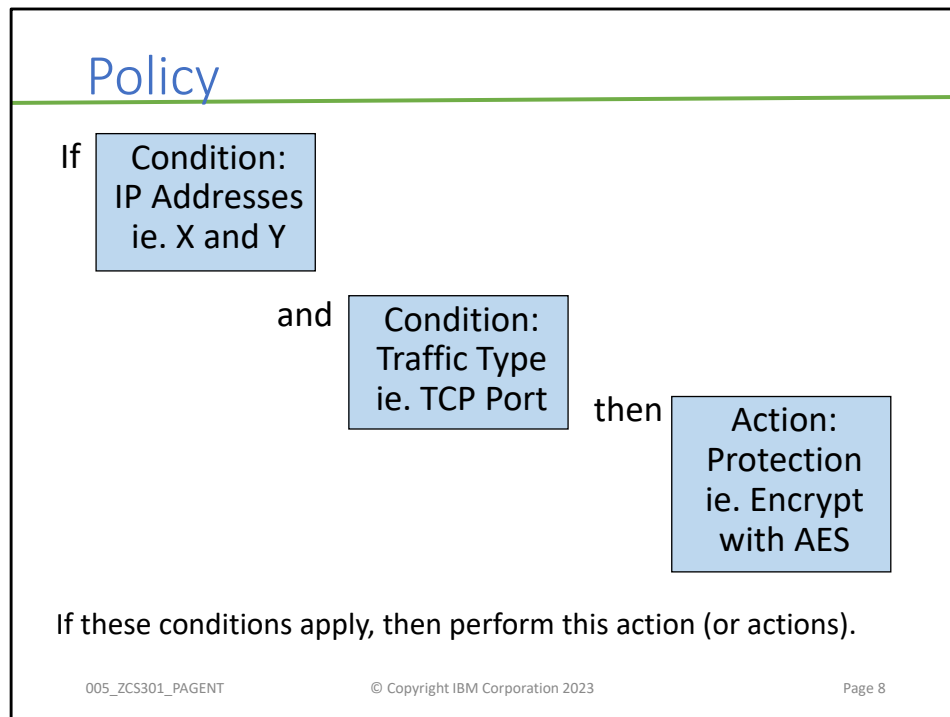
- To block or permit certain types of traffic based upon IP address, port # and other characteristics and conditions; like a firewall for z/OS

IPSec

- To implement security from IPSec endpoint to IPSec endpoint with authentication, data integrity checking, and encryption.
- Invoked at IPSec peers to build a Virtual Private Network (VPN) that can carry multiple connections and session types (e.g., TCP, UDP).

Policy-based Routing

- To influence a routing table based upon IP address and Ports of the application to which the traffic must be routed.



Generally speaking, a Rule is comprised of a set of Conditions, which, when true, cause one or more actions to be performed. Conditions can be varied and they depend on the type of security policy you are defining. That is, there are some conditions that are applicable to IPSec and others that are applicable to AT-TLS and IDS.

- IP addr
- Port #
- Protocol
- Time of Day
- Day of Week
- etc.

Actions can also be varied and will depend on the type of security policy you are defining.

- Permit
- Deny
- Encrypt
- Authenticate
- Report
- Notify
- etc.

# PAGENT Configuration Files



005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 9

## Main Configuration File

- Default is /etc/pagent.conf

```
LogLevel 31
TcplImage NM2ATCP /etc/nm2a.image FLUSH NOPURGE 600
#
PolicyPerfMonforSDR ...
#
SetSubnetPrioTosMask
#
PolicyRule....
#
PolicyAction...
#
```

Image file with pointers to policy files

Some imbedded rules for QoS

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 10

The policy agent can be configured to read its policies from an LDAP server (IDS policies only), a flat configuration file, or both. If reading from an LDAP server, protocol versions LDAPv3 is used.

However, do not use LDAP! The LDAP support is very old and outdated.

Specify LOGLEVEL to capture messages for debugging purposes. The default logging level is 31 which includes Event Logging. statement format: LogLevel i , where i (R): The sum of the following values that represent log levels:

```
LOGL_SYSERR 1; LOGL_OBJERR 2; LOGL_PROTERR 4;
LOGL_WARNING 8; LOGL_EVENT 16; LOGL_ACTION 32;
LOGL_INFO 64; LOGL_ACNTING 128; LOGL_TRACE 256
```

The LogLevel statement is used to define the amount of information to be logged by the Policy Agent. The default is to log only event, error, console, and warning messages. This might be appropriate for a stable policy configuration, but more information might be required to understand policy processing or to debug problems when first setting up policies or when making significant changes. Specify the LogLevel statement with the appropriate logging level in the main configuration file.

FLUSH indicates to clear the IP stack of all policy definitions at Policy Agent Startup; the 600 indicates a 10-minute refresh period in which the MVS file changes (if the flat file is coded in MVS datasets) will be reread for any changes. (Default is 1800 seconds, or 30 minutes.) Policies defined in unix files are updated whenever the unix file is changed if PAGENT is initialized with the "-i" option. There are two more refresh methods documented later in this presentation: SIGHUP and MODIFY command. NOPURGE indicates that the TCP/IP Stack should retain all policies when Policy Agent is terminated; PURGE indicates that the Stack should erase all policies at PAGENT termination.

statement format: TcplImage s1 s2 p n i

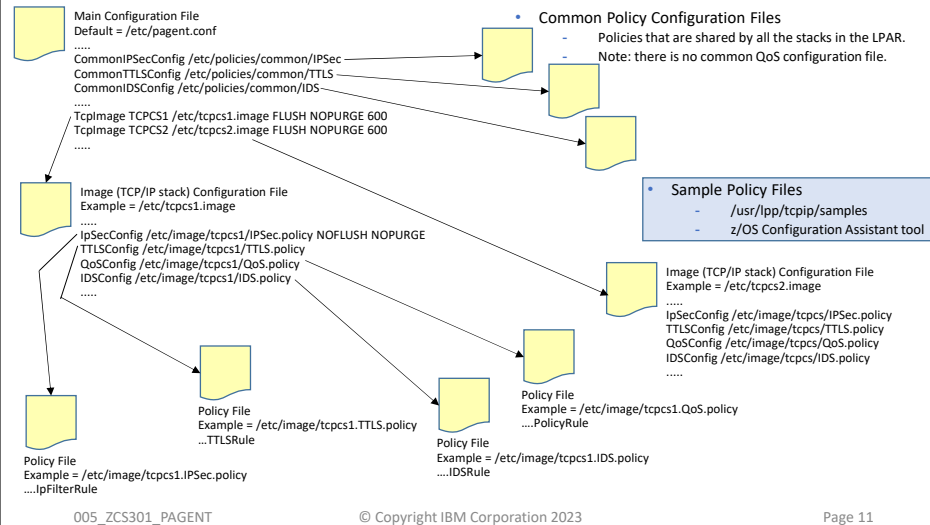
where:

- s1 (R): (8 characters) is the name of the MVS TCP/IP image
- s2 (O): is the path of the policy control file, if not specified this file is used
- p (O): FLUSH | NOFLUSH, default is NOFLUSH
- n (O): PURGE | NOPURGE, default is NOPURGE
- i (O): file/LDAP modification check interval in seconds, default is 1800 (30 minutes).

## How does Policy Agent find Policy Files? Main Policy file Option 1 – Image Files

- There is one Policy Agent per LPAR.

- This one Policy Agent supports all stacks that run in that LPAR.



The PAGENT procedure (or the Pagent Environment File) points to the MAIN Policy Configuration File, which you see represented here.

The contents of the MAIN Policy File points to the individual TCP stacks for which this file is responsible. Note the TCPIMAGE statement is also called the PEPINSTANCE.

This main file may also contain policies.

Currently all configuration statements (TCPIMAGE, TLSConfig, IDSConfig, etc.) must be coded on a SINGLE LINE.

All the policies referenced for a particular stack are then loaded into that stack.

Individual policy configuration files may override the general FLUSH PURGE rules specified in the MAIN Configuration File.

Samples are available in the /usr/lpp/tcpip/samples directory.

Policy rule and action names are limited to 32 characters. If QoS and IDS LDAP statement names longer than 32 characters are specified they are silently truncated. All other statements longer than 32 characters cause an error message to be written to the log.

Policies can be stored in MVS data sets and PDS(E) members, but Policy Agent can only detect dynamic changes if unix files are used.

There is one Policy Agent per MVS image. It can service multiple TCP/IP images if you are running CINET.

If you run CINET, you may wish to point to Common Policy Configuration files that could then be installed in one or multiple TCP/IP stacks in the same MVS image.

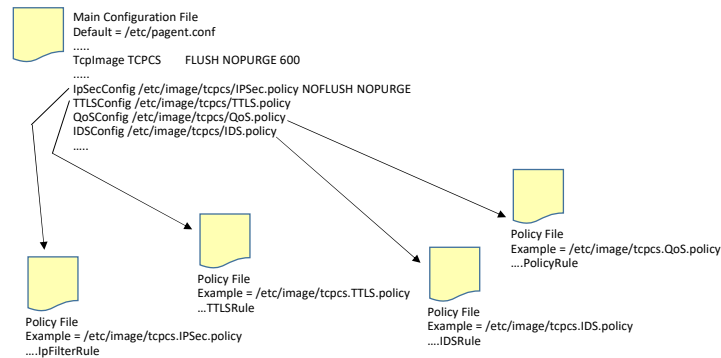
Common policy files are not as useful as you might always like if you intend to export policies from a flat file back into the Configuration Assistant tool.

The main configuration file can contain policies itself. These policies might be a sample set of QoS policies or you could point to either Common Policy Configuration Files or to other types of individual policy files.

Or ... it can point to TcplImage files for each TCP/IP stack supported.

These TcplImage files can point to individual policy configuration files.

## How does Policy Agent find Policy Files? Main Policy file **Option 2 – No Image Files**



- Sample Policy Files
  - /usr/lpp/tcpip/samples
  - z/OS Configuration Assistant tool

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 12

The PAGENT procedure (or the Pagent Environment File) points to the MAIN Policy Configuration File, which you see represented here.

The contents of the MAIN Policy File in this second strategy points directly to the various policy files. Note how the TcpImage statement does not point to any Image File; this means that the Main Policy File is ALSO the IMAGE file.

Currently all configuration statements (TCPIMAGE, TLSConfig, IDSCConfig, etc.) must be coded on a SINGLE LINE.

All the policies referenced for a particular stack are then loaded into that stack.

Individual policy configuration files may override the general FLUSH PURGE rules specified in the MAIN Configuration File.

Samples are available in the /usr/lpp/tcpip/samples directory.

Policy rule and action names are limited to 32 characters. If QoS and IDS LDAP statement names longer than 32 characters are specified they are silently truncated. All other statements longer than 32 characters cause an error message to be written to the log.

## Policy Agent JCL Procedure

Configuration file may be a unix file or an MVS data set.

```
//PAGENT PROC
//*
//* Status = CSV1R9
//*
//PAGENT EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//    PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")'/-c/
//    etc/pagent2.conf -i -l SYSLOGD -d4 -t1'
//* Provide environment variables to run with the desired
//* configuration. As an example, the data set or file specified by
//* STDENV could contain:
//*
//* PAGENT_CONFIG_FILE=/etc/pagent2.conf
//* PAGENT_LOG_FILE=/tmp/pagent2.log
//*
//STDENV DD PATH='/etc/pagent2.env',PATHOPTS=(ORDONLY)
//*
//*STDENV DD DSN=TCPIP.PAGENT.ENV(PAGENT),DISP=SHR
//*
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//*
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

```
TZ=EST5EDT
PAGENT_CONFIG_FILE=/etc/pagent2.conf
PAGENT_LOG_FILE=/tmp/pagent2.log
PAGENT_LOG_FILE_CONTROL=300,3
```

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 13

Policy Agent requires superuser authority to start it.

There is a parameter, "-L", to indicate a logging location that would override what has been coded in the Environment file. Note that we are logging to SYSLOGD.

-c/C: The -c/C option allows a policy configuration file name to be specified. If it is not specified, the configuration file is located using the search order. This can be a unix file or MVS file.

Note: Search order for pagent.conf file:

- File or dataset specified on the -c parameter at policy agent invocation
- File or dataset specified by the PAGENT\_CONFIG\_FILE environment variable
- /etc/pagent.conf
- hlq.PAGENT.CONF

-i/I: When specified, the policy agent monitors its local files (all configuration files) in real time for changes. The time interval configured on the TcplImage statement is also used to monitor configuration files for updates. Use of the -i/I option provides the following benefits:

- More timely updating of policy statements when a unix configuration file is changed

-l/L logfile: The -l/L option can be used to specify the destination of the log output file. Either a unix file or SYSLOGD can be specified. The environment variable PAGENT\_LOG\_FILE also specifies the destination of the log file, using the same format as this option. The -l/L option overrides the PAGENT\_LOG\_FILE environment variable.

Another environment variable, PAGENT\_LOG\_FILE\_CONTROL, specifies the number and size of log files (if SYSLOGD is not specified). The format is: PAGENT\_LOG\_FILE\_CONTROL=x,y where x is the log file size (kilobytes). A maximum value of 1000000 can be specified. y is the number of log files. The default is 3 log files, each 300 kilobytes in size.

- The default is /tmp/pagent.log.

-d/D n: When -d is specified, all trace messages are logged in the policy agent log file. If -d is not used, log messages are written to the policy agent log file as specified by the LogLevel configuration statement. The log file should be the first place checked for error messages.

n can be one of the following values:

- 0 No debug messages are logged. This is the default.
- 1 The Policy Agent logs internal debug information. Overrides LogLevel, forcing LogLevel to 511.
- 2 The Policy Agent logs information as for level 1, plus logs additional information about each LDAP object attribute that is processed.
- 4 Sysplex Distributor summary information
- 8 Sysplex Distributor detail information
- -t Specifies that LDAP client library tracing be enabled

Debug levels 2 through 8 do not affect the current LogLevel.

Debug levels can be combined by adding them together.

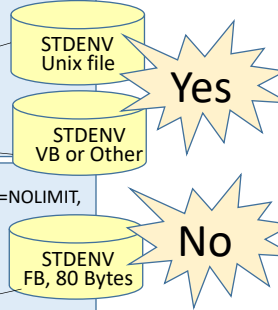
- -d 9 (debug level 1 plus debug level 8)

## Fixed Block File Problem

- Standard Environment File (STDENV)

```
//PAGENT PROC
//PAGENT EXEC PGM=PAGENT,REGION=4096K,TIME=NOLIMIT,
//  PARM=('POSIX(ON) ALL31(ON)',
//  'ENVAR("_CEE_ENVFILE=DD:STDENV")',
//  '/ -c /etc/pagent2.conf -i -L SYSLOGD -d4 -t1')
//*
//STDENV DD PATH='/etc/pagent2.env',
//  PATHOPTS=(ORDONLY)
//*STDENV DD DSN=SYS1.TCPIP.STDENV,DISP=SHR
...

//PAGENT PROC
//PAGENT EXEC PGM=PAGENT,REGION=4096K,TIME=NOLIMIT,
//  PARM=('POSIX(ON) ALL31(ON)',
//  'ENVAR("_CEE_ENVFILE=DD:STDENV")',
//  '/ -c /etc/pagent2.conf -i -L SYSLOGD -d4 -t1')
//*
//STDENV DD DSN=SYS1.TCPIP.TCPPARMS(PAGENTV),DISP=SHR
...
```



- Make sure your STDENV file is not in a fixed block dataset.

```
//PAGENT EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//  PARM=('POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE_S=DD:STDENV")/-
//  c/etc/pagent2.conf -i -L SYSLOGD -d4 -t1'
```

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 14

PAGENT is a UNIX application.

UNIX pads fixed-block file lines with blanks to the fixed-block size.

In UNIX, blanks are valid filename characters.

You are always safer allocating your STDENV file as either a unix file or as a non-Fixed Block file, unless the files you reference inside the Environment file (STDENV) are MVS datasets. The next page explains more.

## Fixed Block File Padded Blanks

STDENV  
FB, 80 Bytes

0	10	20	30	40	50	60	70	80
+.....+.....+.....+.....+.....+.....+.....+.....+.....+								
PAGENT_CONFIG_FILE=/etc/pagent2.conf								
TZ=EST5EDT								

- EZZ7822 Could not find configuration file
- - or -
- Trailing blanks in directory names or filenames are not supported by edit or browse
- - or -
- EDC5129I No such file or directory.

- TRMD and PAGENT are UNIX applications.
- UNIX pads fixed-block file lines with blanks to the fixed-block size
- In UNIX, blanks are valid filename characters.
- So in this example PAGENT would be looking for files named:

"/etc/pagent2.conf
"

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 15

If you use a Language Environment (LE) STDENV file and get this error on any UNIX process startup:

- EZZ7822 Could not find configuration file

This means that your configuration file is in a unix file. UNIX pads fixed-block file lines with blanks to the fixed-block size and thus converts your configuration file name into a name that is padded with blanks. Since the actual configuration file name is not padded with blanks, the file cannot be found!

**BEST SOLUTION:** Make sure your STDENV file is not in a fixed-block dataset: either a Variable-Blocked or Sequential Dataset or a unix file.

**ALTERNATE SOLUTION #1:** If STDENV must remain FB, put your configuration file into an MVS Dataset.

If you use a STDENV file and get any of the following errors when you try to read or browse any dataset or file that is referenced inside a STDENV file, immediately suspect that the STDENV has been incorrectly allocated in a fixed block file.

- Trailing blanks in directory names or filenames are not supported by edit or browse
- cat: <file> EDC5129I No such file or directory.
- Errno=81x No such file or directory exists; Reason=05620062x
  - This means that your file has been coded inside a Standard Environment File that resides in a Fixed Block Dataset in MVS and it is pointing to a unix file. UNIX dynamically creates the file. As explained previously, UNIX pads fixed-block file lines with blanks to the fixed-block size and thus converts your debug trace file into a name that is padded with blanks.
- **BEST SOLUTION:** Make sure your STDENV file is not in a fixed-block dataset: either a Variable-Blocked or Sequential Dataset or a unix file.
- **ALTERNATE SOLUTION #1:** If STDENV must remain FB, copy the unix file into an MVS Dataset and browse from there. (This is supported.)

# Modify PAGENT and Monitor Applications



005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 16

## FLUSH, PURGE, and Modify

- FLUSH or NOFLUSH and PURGE or NOPURGE are defined in policy configuration files:
  - Tcplmage TCP/IP /etc/tcpip\_policy.config FLUSH PURGE
  - IpSecConfig /etc/IPSec.policy NOFLUSH NOPURGE
- Original design:
  - FLUSH – When PAGENT is started, if policies exist in the TCP/IP stack, policies are removed, and then reloaded.
  - NOFLUSH – When PAGENT is started, if policies exist in the TCP/IP stack, then they are not removed and reloaded.
  - PURGE – When PAGENT is stopped, policies are removed from the TCP/IP stack.
  - NOPURGE – When PAGENT is stopped, policies are not removed from the TCP/IP stack.
  - FLUSH effects Modify behavior as well.
- Modify Command
  - Refresh removes all policies from the TCP/IP stack and then reloads them.
    - F PAGENT,REFRESH
  - Update only changes the active policies that have been changed in the files.
    - F PAGENT,UPDATE
- Changes in configuration and policy files are installed:
  - Immediately when file is saved if -i is defined on startup – for unix files only
 

```
//PAGENT EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//  PARM='POSIX(ON) ALL31(ON) ENVAR("CEE_ENVFILE=DD:STDENV")' /-c -i
```
  - At time interval if -i is defined on Tcplmage statement (default 1800 seconds) – for all file types
    - Tcplmage TCP/IP /etc/tcp\_policy.conf NOFLUSH NOPURGE -i
  - When Modify command is issued - for all file types
    - F PAGENT,REFRESH
    - F PAGENT,UPDATE
  - When SIGHUP is issued - for all file types
    - kill -1 3425

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 17

What happens if you need to change some information in your policy?

There are several ways to delete, reload or refresh the policy information.

You can specify FLUSH/NOFLUSH and PURGE/NOPURGE on the Tcplmage (or PEPInstance) configuration or on individual policy statements. FLUSH/NOFLUSH controls whether Policy Agent, when initialized, will delete all the policies already installed in the stack. PURGE/NOPURGE controls whether the policies should be deleted from the stack at PAGENT termination. NOFLUSH and NOPURGE are the defaults.

The policies managed by policy agent become active when the TCP/IP stack has completed initialization. Policies stored in a unix file or in LDAP are refreshed dynamically if "-i" is specified at startup of the Policy Agent \*AND\* if the policy agent configuration file resides in a unix file and has been modified. (Policies stored in MVS members or datasets are not dynamically refreshed when they are altered; you must either issue a MODIFY command or you must wait for the refresh interval specified on the IMAGE file or individual policy file to expire.)

An interval specified on the TCPIIMAGE statement in the configuration file also triggers a refresh of the policies at regular intervals (default is 1800 seconds/30 minutes); this allows policies that have been stored in MVS files to be modified and installed dynamically into the stack. This value can never be disabled but you can set it to the maximum value of 2 147 483 647 seconds.

If a value is not specified, the default is 1800 seconds (30 minutes).

If a value of 0 is specified, the default value of 1800 (30 minutes) is used.

Any value in the range 1 - 59 is rounded up to 60 seconds (1 minute).

A Modify command or a SIGHUP can also refresh the policies for policies stored in a unix file, MVS file, and LDAP.

A MODIFY,procname,UPDATE or REFRESH command is available.

F REFRESH will cause all policies to be flushed and then reloaded.

If the FLUSH parameter was specified on the Tcplmage or discipline configuration statement, the REFRESH command triggers FLUSH processing. One consequence of this is that policy statistics being collected in the TCPIP stack are reset, because FLUSH deletes and reinstalls all policies.

A SIGHUP process signal (eg.: "kill -1 <pagent\_pid>") will also cause a refresh of the policy from unix, MVS, or LDAP just as does the MODIFY REFRESH command.

F UPDATE will cause only the changed policies to be reloaded.

This command is different from the REFRESH command because Pagent only installs or removes from the stack as appropriate any new, changed, or deleted policies.

## Behavior is Different for Policy Types

PAGENT start	FLUSH defined	IPsec and zERT Policies – All policies are replaced in the TCP/IP stack.
		PBR and All Other Policies – All policies are deleted, and then all policies are reloaded into the TCP/IP stack.
	NOFLUSH defined	IPsec and zERT Policies – All policies are replaced in the TCP/IP stack.
		PBR Policies – All policies are deleted, and then all policies are reloaded into the TCP/IP stack.
PAGENT Termination	PURGE defined	All Other Policies – All changed policies are updated in the TCP/IP stack. No deleted policies are removed from the TCP/IP stack.
		IPsec, zERT and PBR Policies – TCP/IP stack policies are unchanged.
	NOPURGE defined	All Other Policies – All policies are removed from the TCP/IP stack.
		IPsec, zERT, PBR, and All Other Policies – TCP/IP stack policies are unchanged.
PAGENT REFRESH	FLUSH defined	IPsec and zERT Policies – If there are any changed or deleted policies, then all policies are replaced in the TCP/IP stack.
		PBR and All Other Policies – If there are any changed or deleted policies, then all policies are deleted, and then all policies are reloaded into the TCP/IP stack.
	NOFLUSH defined	IPsec and zERT Policies – If there are any changed or deleted policies, then all policies are replaced in the TCP/IP stack.
		PBR Policies – If there are any changed or deleted policies, then all policies are deleted, and then all policies are reloaded into the TCP/IP stack.
PAGENT UPDATE	FLUSH defined	All Other Policies – If there are any changed policies, then they are replaced in the TCP/IP stack. No deleted policies are removed from the TCP/IP stack.
		IPsec and zERT Policies – If there are any changed or deleted policies, then all policies are replaced in the TCP/IP stack.
	NOFLUSH defined	PBR and All Other Policies – If there are any changed policies, then they are replaced in the TCP/IP stack, and then all deleted policies are removed from the TCP/IP stack.
		IPsec and zERT Policies – If there are any changed or deleted policies, then all policies are replaced in the TCP/IP stack.
	FLUSH defined	PBR Policies – If there are any changed policies, then they are replaced in the TCP/IP stack, and then all deleted policies are removed from the TCP/IP stack.
		All Other Policies – If there are any changed policies, then they are replaced in the TCP/IP stack. No deleted policies are removed from the TCP/IP stack.
	NOFLUSH defined	IPsec and zERT Policies – If there are any changed or deleted policies, then all policies are replaced in the TCP/IP stack.
		PBR Policies – If there are any changed policies, then they are replaced in the TCP/IP stack, and then all deleted policies are removed from the TCP/IP stack.

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 18

IPsec policies are too important to remove, so those policies are treated differently.  
Policy Based Routing (PBR) is more recently added support. When it came along it was thought best to change the behavior.

# Logging

Log and Debug levels are not changed by updating the PAGENT configuration file and issuing Modify Update or Refresh!

- Log Level, Debug, and Trace
  - Log Level parameter is defined in the image Policy File.
  - Debug and Trace are specified on the Policy Agent start.
- Log Level
  - 1 - SYSERR - System error messages
  - 2 - OBJERR - Object error messages
  - 4 - PROTERR - Protocol error messages
  - 8 - WARNING - Warning messages
  - 16 - EVENT - Event messages
  - 32 - ACTION - Action messages
  - 64 - INFO - Informational messages
  - 128 - ACNTING - Accounting messages
  - 256 - TRACE - Trace messages
  - Log Level values are added together for a maximum Log Level value of 511.
  - 31 is the default
- Debug
  - 0 None. No debug messages are logged. This is the default.
  - 1 Base. The Policy Agent logs internal debug information.
    - When this level is selected, the Policy Agent also uses the maximum LogLevel value, regardless of what is configured.
  - 2 LDAP. The Policy Agent logs information about each LDAP object attribute that is processed.
  - 4 Sysplex summary. The Policy Agent logs summary information about performance monitor QoS fraction calculations at target stacks.
  - 8 Sysplex detail. The Policy Agent logs detailed information about performance monitor QoS fraction calculations at target stacks, and additional sysplex distributor information.
  - 16 Memory trace. The Policy Agent logs inline details of all memory allocation and free requests. This debug level is independent of the -m startup option.
  - 32 Policy install trace. The Policy Agent logs details of all policies as the policies are installed in the TCP/IP stack.
  - 64 Lock trace. The Policy Agent logs information about locks.
  - 128 Remote connection trace. The Policy Agent logs details about remote PAPI connections on the policy server and about connections to the policy server on the policy client.
  - 256 Discovery connection trace. The Policy Agent logs details about requests to discover TCP/IP profile information from import requestors.
  - Debug values are added together for a maximum Debug Level of 511.
- Trace
  - 0 No LDAP client debugging. This is the default.
  - 1 This level turns on LDAP client debugging.

- Set your loglevel and debug parameters with commands if you need more than the PAGENT defaults. Then disable the additional logging once no longer needed.
  - F PAGENT,LOGLEVEL,LEVEL=
  - F PAGENT,DEBUG,LEVEL=
  - F PAGENT,QUERY

005\_ZCS301\_PAGENT

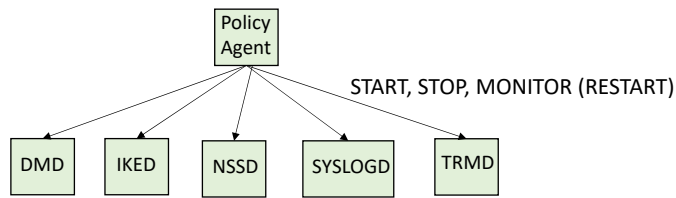
© Copyright IBM Corporation 2023

Page 19

Log Level and Debug Level can only be dynamically modified via the modify commands listed on this page. Updating the Policy file and issuing F PAGENT,UPDATE or F PAGENT,REFRESH does not update the Log Level and Debug Level.

## Monitor Applications

- Policy Agent may be configured to automatically Start, Stop, and Monitor applications.
  - Configure with AutoMonitorApps statement.

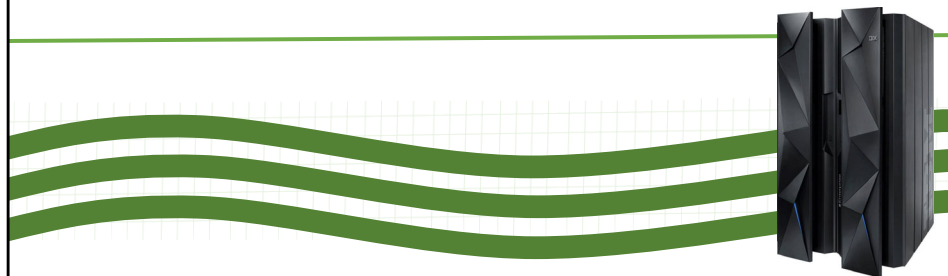


The idea is that it should be possible to only manually start Policy Agent and your TCP/IP stacks. Policy Agent will then be able to start and monitor all other policy-related components.

The sequence of events is to start Policy Agent first, and then the stack or stacks.

Definitions in the Policy Agent configuration file instructs Policy Agent what to start/monitor/stop.

# Policy Views



005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

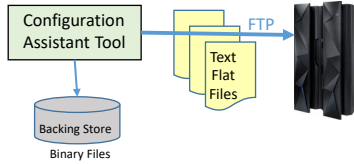
Page 21

## Example of Policy in Configuration Assistant Tool

```
## Created by the IBM Configuration Assistant for z/OS Communications Server
## Version 1 Release 10
## Backing Store = C:\...\\IBM\\zCSConfigAssist\\V1R10\\files\\ATLSTEAM22_301
## FTP History:
##
## End of Configuration Assistant information
#####
# PolicyRule statements
#####

policyRule 0~1
{
  PolicyRulePriority    65000
  DestinationAddressRange 10.1.1.0-10.1.1.255
  SourcePortRange      1024-65535
  DestinationPortRange 21
  ProtocolNumberRange  6
  PolicyActionReference action~1
}
#####
# PolicyAction Statements
#####

PolicyAction action~1
{
  PolicyScope      DataTraffic
  OutgoingTOS      01000000
  DiffServInProfileRate 256
  DiffServInProfileTokenBucket 512
  DiffServExcessTrafficTreatment Drop
}
```



005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 22

When you create a similar QoS (or other) policy with the Configuration Assistant, the tool produces a binary file called the "Backing Store file".

The administrator FTPs text files to the target host where PAGENT is to install them.

When it arrives at the target node, it is in text file format.

The transfer of the policy files may be from the tool to the z/OS where Policy Agent will use then, or the policies may be transferred from the Policy Agent to the tool.

## Example of Policy on z/OS

- Stored as "flat file" in MVS dataset or in Unix file

```

PolicyRule          DiffServ_Rule1
{
  DestinationAddressRange  211.40.100.0-211.40.100.255
  SourcePortRange          20-21
  PolicyActionReference     DiffServ_Action1
  DayOfWeekMask            0111110
}
#
PolicyAction         DiffServ_Action1
{
  PolicyScope              DataTraffic
  OutgoingTOS               01000000
  DiffServInProfileRate     512    # 512 Kbps
  DiffServInProfileTokenBucket 64    # 64 Kbits
  DiffServInProfilePeakRate 1500    # 1.5 Mbps
  DiffServInProfileMaxPacketSize 120  # 120 Kbits
  DiffServOutProfileTransmittedTOSByte 00000000
  DiffServExcessTrafficTreatment BestEffort
}
    
```

QoS

- Beware of Duplicate Object Names
  - Sometimes a warning message is issued; sometimes it is not.
  - Sometimes the first entry is ignored; sometimes the last entry is ignored.

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 23

Syntax rules for policies...

Policy Rule and Action Names:

- Policy rule and action names are limited to 32 characters. If QoS and IDS LDAP statement names longer than 32 characters are specified they are silently truncated. All other statements longer than 32 characters cause an error message to be written to the log.

If Duplicate Statement or Object Names are Present, Policy Agent keeps the first or the last statement or object, as follows.

- For IDS (LDAP) and QoS, Policy Agent keeps the first entry.
- For IDS (configuration file), IPSec, Routing, and AT-TLS, Policy Agent keeps the last entry.
- If a QoS or IDS statement or object is defined with the same name in both a configuration file and LDAP, Policy Agent keeps the first such statement or object that it reads.

## pasearch Example

```
policyRule:      VIPAs2VIPAs~1
Rule Type:      TLS
Version:        3      Status:      Active
Weight:         255    ForLoadDist: False
Priority:        255    Sequence Actions: Don't Care
No. Policy Action: 3
policyAction:    gAct1
ActionType:      TLS Group
Action Sequence: 0
policyAction:    eAct1~AllSecureFTPUsers
ActionType:      TLS Environment
Action Sequence: 0
policyAction:    cAct1~AllSecureFTPUsers
ActionType:      TLS Connection
Action Sequence: 0
Time Periods:
Day of Month Mask:
First to Last:   11111111111111111111111111111111
Last to First:   11111111111111111111111111111111
Month of Yr Mask: 111111111111
Day of Week Mask: 1111111 (Sunday - Saturday)
Start Date Time: None
End Date Time:   None
Fr TimeOfDay:    00:00    To TimeOfDay: 24:00
Fr TimeOfDay UTC: 05:00    To TimeOfDay UTC: 05:00
TimeZone:        Local
TLS Condition Summary: NegativeIndicator: Off
```

```
Local Address:
FromAddr:      192.168.20.101
ToAddr:        192.168.20.105
Remote Address:
FromAddr:      192.168.20.101
ToAddr:        192.168.20.105
LocalPortFrom: 1024    LocalPortTo: 65535
RemotePortFrom: 21    RemotePortTo: 21
JobName:        Userid: USER*
ServiceDirection: Outbound
Policy created: Mon Nov 23 17:39:59 2009
Policy updated: Mon Nov 23 17:39:59 2009

TLS Action:      gAct1
Version:         3
Status:          Active
Scope:           Group
TLSEnabled:      On
CtraceClearText: Off
Trace:           2
TLSGroupAdvancedParms:
SecondaryMap:    Off
SyslogFacility: Daemon
Policy created: Mon Nov 23 17:39:59 2009
Policy updated: Mon Nov 23 17:39:59 2009
...
```

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 24

This is the output display from a pasearch command.

The pasearch command is subject to RACF authorizations. The IP Configuration Guide tells you how to set up these authorizations. Note how all the separate pieces of this TLS policy rule are pulled together for display. If you were to examine the Policy Flat file, you would see many separate sections that are separated in different parts of the flat file. PASEARCH pulls all of this together for a consistent view of the entire Rule, its conditions and its actions.

Note that this Rule contains many conditions and three actions that should be applied.

You see the address conditions, the port conditions, the timeframe conditions and even the userid conditions.

Only one of the three actions is listed.

## RACF Protection for pasearch

- **pasearch** command causes Policy Agent to display active policies.
  - **pasearch -i** IDS policies
  - **pasearch -q** QoS policies
  - **pasearch -R** PBR policies
  - **pasearch -t** AT-TLS policies
  - **pasearch -v** IPsec policies
- **SERVAUTH** class profile EZB.PAGENT.sysname.tcpimage.ptype
  - Where sysname is the z/OS system name,
  - tcpimage is the TCP/IP stack name,
  - ptype is either QOS or IDS.

```

EZB.PAGENT.SYSTEM1.TCPCS.QOS
EZB.PAGENT.SYSTEM1.*.IDS
EZB.PAGENT.SYSTEM1.*.*
      
```
- **EZACMD** command
  - REXX job that makes previous unix only commands available to TSO, NetView, and the z/OS console.
    - > **pasearch**
    - > **ipsec**
    - > **trmdstat**
    - > **nssctl**
    - > **ping**

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 25

Additional security checking for the pasearch command:

- Existing superuser authority OR
- Security product authorization via SERVAUTH class profile

The security product profiles in place affect the results of pasearch - pasearch only returns policies that the user is allowed to see.

The first sample profile only allows QoS policies for the TCPCS stack on SYSTEM1 to be seen.

The second sample profile allows IDS policies for all stacks on SYSTEM1 to be seen.

The third sample profile allows all policy types for all stacks on SYSTEM1 to be seen.

RACF Commands to establish these definitions:

- Activate the SERVAUTH class
  - SETROPTS CLASSACT(SERVAUTH)
  - SETROPTS RACLIST(SERVAUTH)
- Define EZB.PAGENT.sysname.tcpname.QOS|IDS|\*
  - RDEFINE SERVAUTH EZB.PAGENT.sysname.tcpprocname.QOS | IDS | \*
- Permit access to the EZB.PAGENT.sysname.tcpname.<policy type> resource to users
  - PERMIT EZB.PAGENT.sysname.tcpprocname.<policy type> CL(SERVAUTH) ID(userid) ACCESS(READ)
  - READ access allows the user to access the Policy Agent
- SETROPTS RACLIST(SERVAUTH) REFRESH

The samples for PASEARCH authorization are in SYS1.TCPIP.SEZAINST(EZARACF):

```

/*PAGNACC EXEC PGM=IKJEFT01
/*SYSTSPRT DD SYSOUT=*
/*SYSTSIN DD *
/* SETROPTS CLASSACT(SERVAUTH)
/* SETROPTS RACLIST (SERVAUTH)
/* SETROPTS GENERIC (SERVAUTH)
/* RDEFINE SERVAUTH EZB.PAGENT.sysname.imagename.* -
/* UACC(NONE)
/* PERMIT EZB.PAGENT.sysname.imagename.* -
/* CLASS(SERVAUTH) ID(userid) ACCESS(READ)
/* SETROPTS GENERIC(SERVAUTH) REFRESH
/* SETROPTS RACLIST(SERVAUTH) REFRESH
      
```

# Network Configuration Assistant Tool

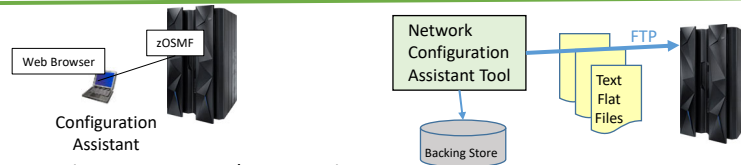


005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 26

## IBM Network Configuration Assistant for z/OS Communications Server



- Runs on zOSMF (available since z/OS V1R11)
- Network Configuration Assistant configurations are stored in binary files
  - Named "Backing Store" files (also referred to as Persistent Data Store)
  - Only Network Configuration Assistant for z/OS Communications Server can use Backing Store files!
  - zOSMF Tool saves Backing Store files on z/OS
    - Auto-backup to protect against loss of changes due to web browser session interruptions
- To use Network Configuration Assistant configurations the tool is used to send text files to z/OS
  - Network Configuration Assistant uses FTP to send the text files (FTP Server is required on z/OS)
  - Different text files can be generated by the Configuration Assistant
    - Separate policy file for each policy type (AT-TLS, IPsec, IDS, QoS, PBR)
    - Application setup files (IKED, NSSD, etc.)
- Older versions of Network Configuration Assistant Backing Store files may be upgraded to a later version.

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 27

Allows policy definition to be performed at higher level of abstraction than policy file statements.

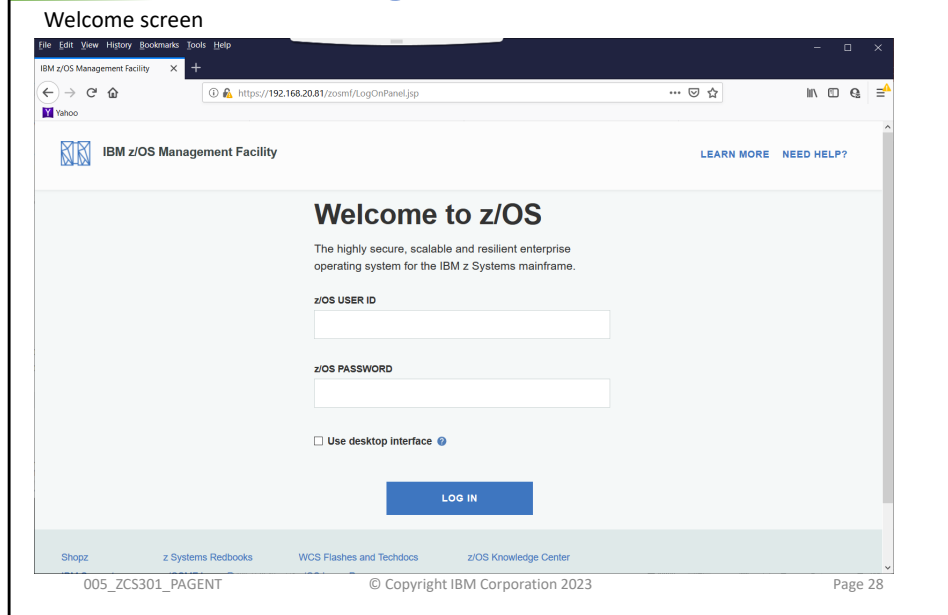
Locking support at the persistent data store to prevent inadvertent loss of data.

- When file is being edited and another person tries to open the file they will get a message that it is already in use.

If z/OS policies are imported into the Configuration Assistant, it is recommended to do the import only once, or as few times as possible.

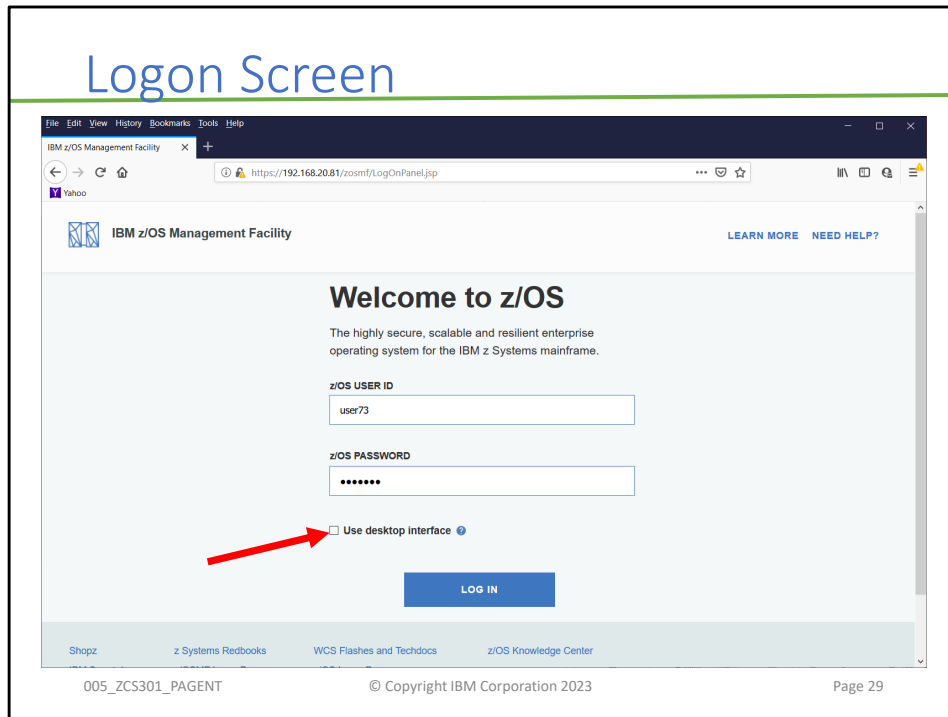
- Internal names are created by the tool so the text files do not match the original policy files on z/OS, but the policy execution is the same.

# Network Configuration Assistant Tool

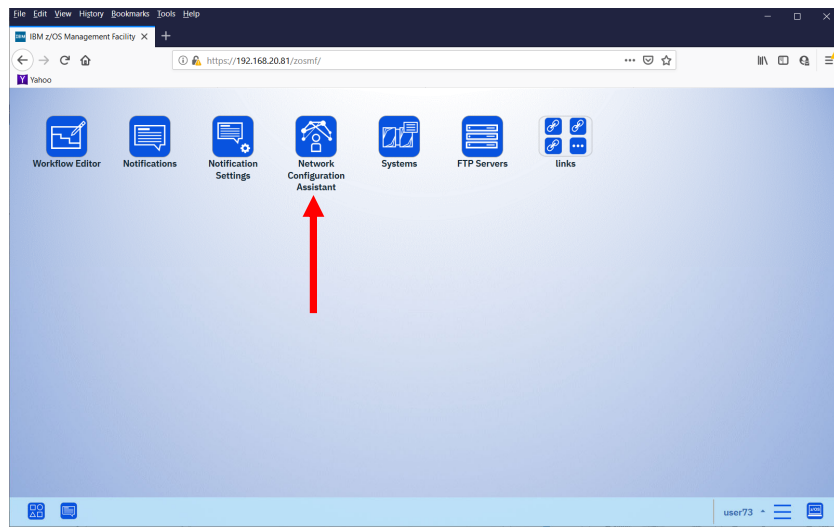


Another way to download the GUI executable is from the following web page:  
<http://www.ibm.com/software/network/commserver/zos/support/>  
then select "Download" from the bottom of the page and search for Configuration Assistant.

# Logon Screen



# Desktop Interface

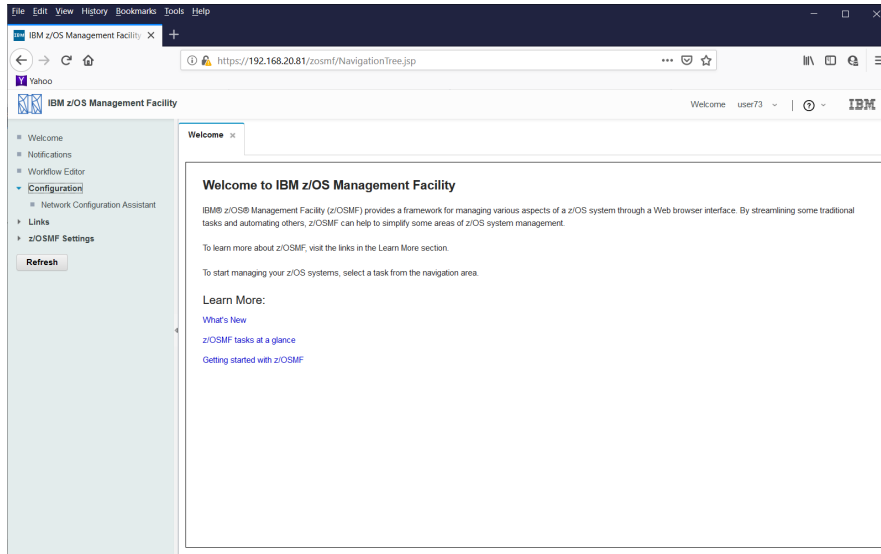


005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 30

# Network Configuration Assistant



005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 31

## Network Configuration Assistant Tool Usage

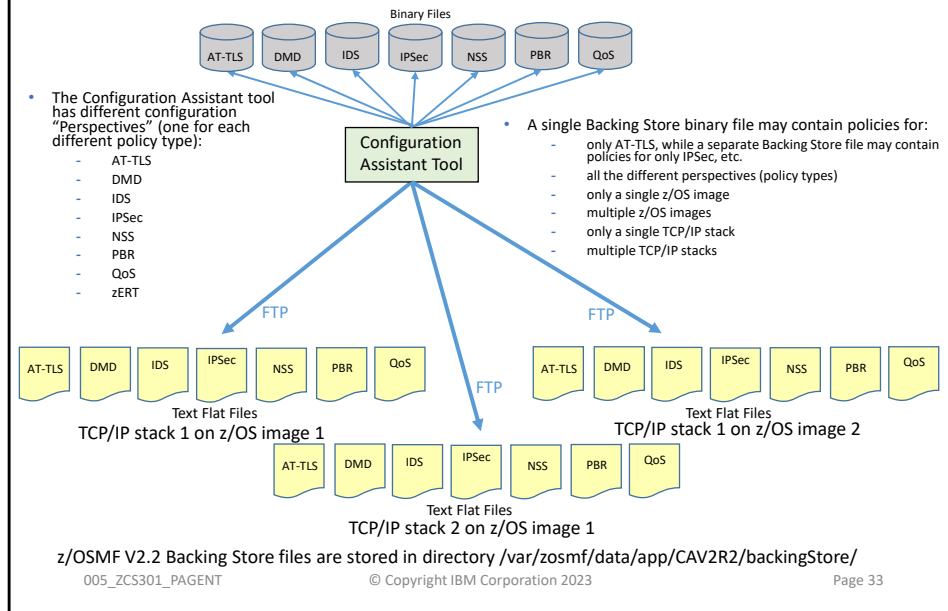
- As you can see on the previous foil, the Network Configuration Assistant (CA) tool may be used to configure:
  - z/OS Cloud – not covered in this class (see CA tutorial for more info)
  - AT-TLS
  - DMD
  - IDS
  - IPSec
  - NSS
  - PBR - not covered in this class
  - QoS - not covered in this class
  - zERT – not covered in this class
  - TCP/IP Profile – not covered in this class (may be used to customize a profile file for a TCP/IP stack)

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

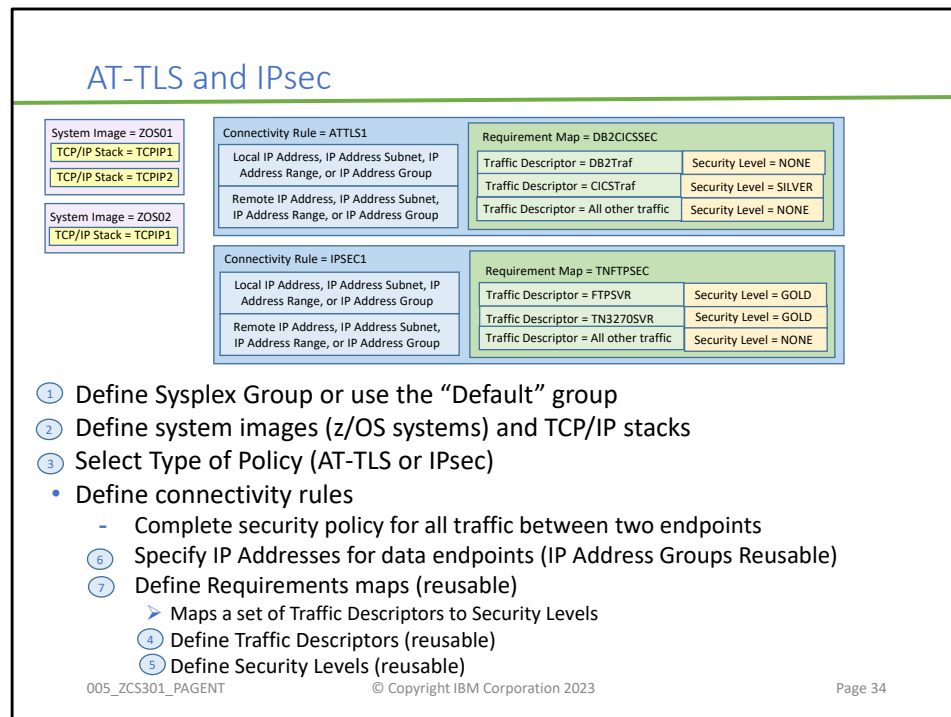
Page 32

# Backing Store Files and Flat Files



There can be a single Backing Store file or multiple. A Backing Store file may contain a single policy type or multiple. A Backing Store file may contain policies for a single TCP/IP stack or multiple. A Backing Store file may contain policies for a single z/OS image or multiple.

Each Policy text flat file that gets sent to a z/OS image for usage by Policy Agent only contain a single policy type. There is a separate policy text file sent for each policy type defined.



There are Wizards that start for each section with a set of panels that must be completed to create the selected item. Wizards and dialogs guide you through a top-down approach to the configuration. The order depicted on the chart above. Navigational tree supports a bottom-up approach to allow an experienced user to bypass wizard screens. Navigational tree appears on the left hand side of the Window.

## Network Configuration Assistant Data Model

- Define system images (z/OS systems) and TCP/IP stacks
- Select Type of Policy or Configuration Option
  - DMD
    - IP Address subnets (for exclusion)
  - IP Filtering
    - Local IP Address, Subnet, Range, or Group
    - Remote IP Address, Subnet, Range, or Group
    - Traffic Descriptor
    - Permit or Deny
  - IDS
    - Attack Types
    - Scans – Traffic Descriptors and Sensitivity
    - Traffic Regulation – Traffic Descriptors and Actions (Limit, Report, or both)
  - NSS
    - Server or Client
  - QoS
    - Local IP Address, Subnet, or Range
    - Remote IP Address, Subnet, or Range
    - Traffic Descriptors
    - Priority
    - Traffic Shaping Level
  - PBR
    - Traffic Descriptors
    - Local IP Address, Subnet, Range, or Group
    - Remote IP Address, Subnet, Range, or Group
    - Routing Tables
    - Use Main Route Table?
  - Type of Policy or Configuration Option
    - Application Transparent – Transport Layer Security (AT-TLS)
    - Defense Manager Daemon (DMD)
    - IP Filtering and IPsec
    - Intrusion Detection Services (IDS)
    - Network Security Services (NSS)
    - Quality of Service (QoS)
    - Policy Based Routing (PBR)
    - TCP/IP Profile
  - zERT
    - Protocol
    - Port
    - Action

005\_ZCS301\_PAGENT

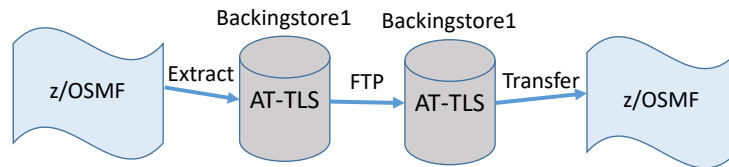
© Copyright IBM Corporation 2023

Page 35

## Samples, Samples, Samples

- Samples in the z/OS Communications Server  
TCP/IP Unix sample directory:
  - /usr/lpp/tcpip/samples
- Samples in Network Configuration Assistant for z/OS
  - Connectivity Rules
  - Requirement Maps
  - Traffic Descriptors
  - Security Levels

## Extract and Transfer



- From “Manage Backing Stores” panel
- Extract
  - Save Backing Store to a unix file.
- Transfer
  - Load Backing Store file from a unix file.

z/OSMF backing store files are located in the path `/var/zosmf/data/app/` for example the Version 2.2 path is `/var/zosmf/data/app/CAV2R2/backingStore/`

# Policy Server

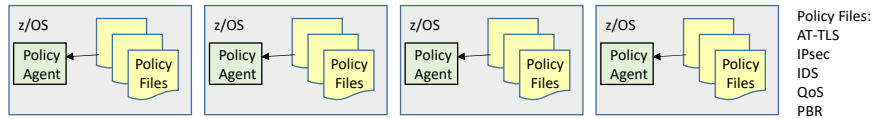


005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

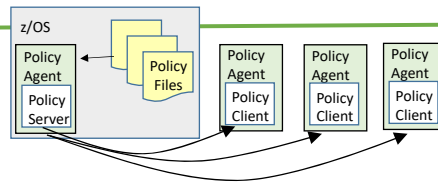
Page 38

## No Central Location for Policies



- Each z/OS system Policy Agent may have their own Policy files stored locally.
- Policy Administration may be from a single location
  - Configuration (Configuration Assistant or manual edits) and monitoring (pasearch) can be done from a single location (administrator computer).

## Policy Server



- Centralized policy storage
  - Policy Server and Policy Clients can be non-sysplex, in sysplex, or cross-sysplex
    - IP connectivity required
- Policy Server provides policies to Policy Clients
  - Policy Client requests policies (ie. when client comes up or modify command)
  - When policies are changed on the server they are sent to clients
- Sysplex Not Required
  - Policy Server and Policy Clients can be non-sysplex, in sysplex, or cross-sysplex
- Availability
  - Backup Policy Server is supported
- Local Policies still supported
  - If Policy Client has policies locally stored, they will take precedence over policies from Policy Server.
- Administration may be from a single location (same as without Policy Server)
  - Configuration (Configuration Assistant or manual edits) and monitoring (pasearch) can be done from a single location (administrator computer).

Policy Files:  
AT-TLS  
IPsec  
IDS  
QoS  
PBR

005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 40

Policies for multiple z/OS systems can be stored on a single z/OS system.

# End of Topic



005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 41

# End of Topic



005\_ZCS301\_PAGENT

© Copyright IBM Corporation 2023

Page 42