IBM Software Group
Enterprise Networking Solutions (ENS) and Transaction Processing Facility (TPF)

Configuring, Operating, and Monitoring Policy Agent

Alfred B Christensen - alfredch@us.ibm.com
IBM Software Group, Enterprise Networking Solutions, Raleigh

IBM Systems

Trademarks and notices

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

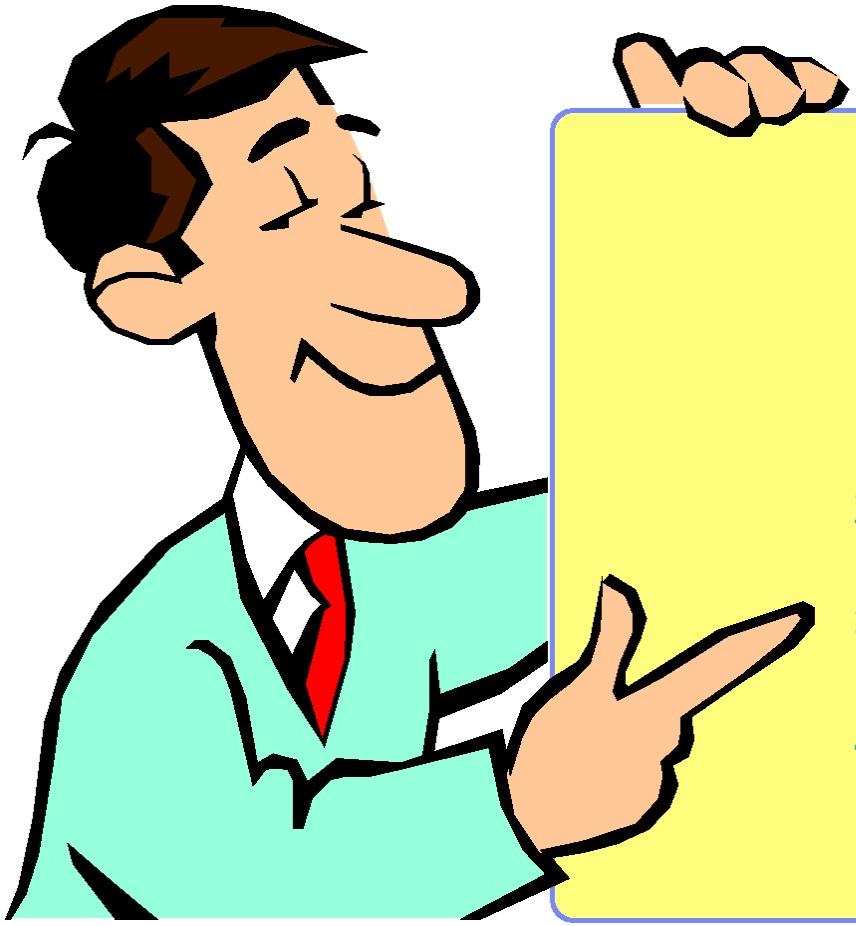
- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM (logo)®
- IBM®
- IMS
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x
- System z
- System z9
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Refer to www.ibm.com/legal/us for further legal information.

Agenda



1. **z/OS CS Networking Policy Infrastructure Overview**
2. **SyslogD and TRMD**
3. **Policy Agent (PAGENT)**
4. **Configuration Assistant**

z/OS CS Networking Policy Infrastructure Overview

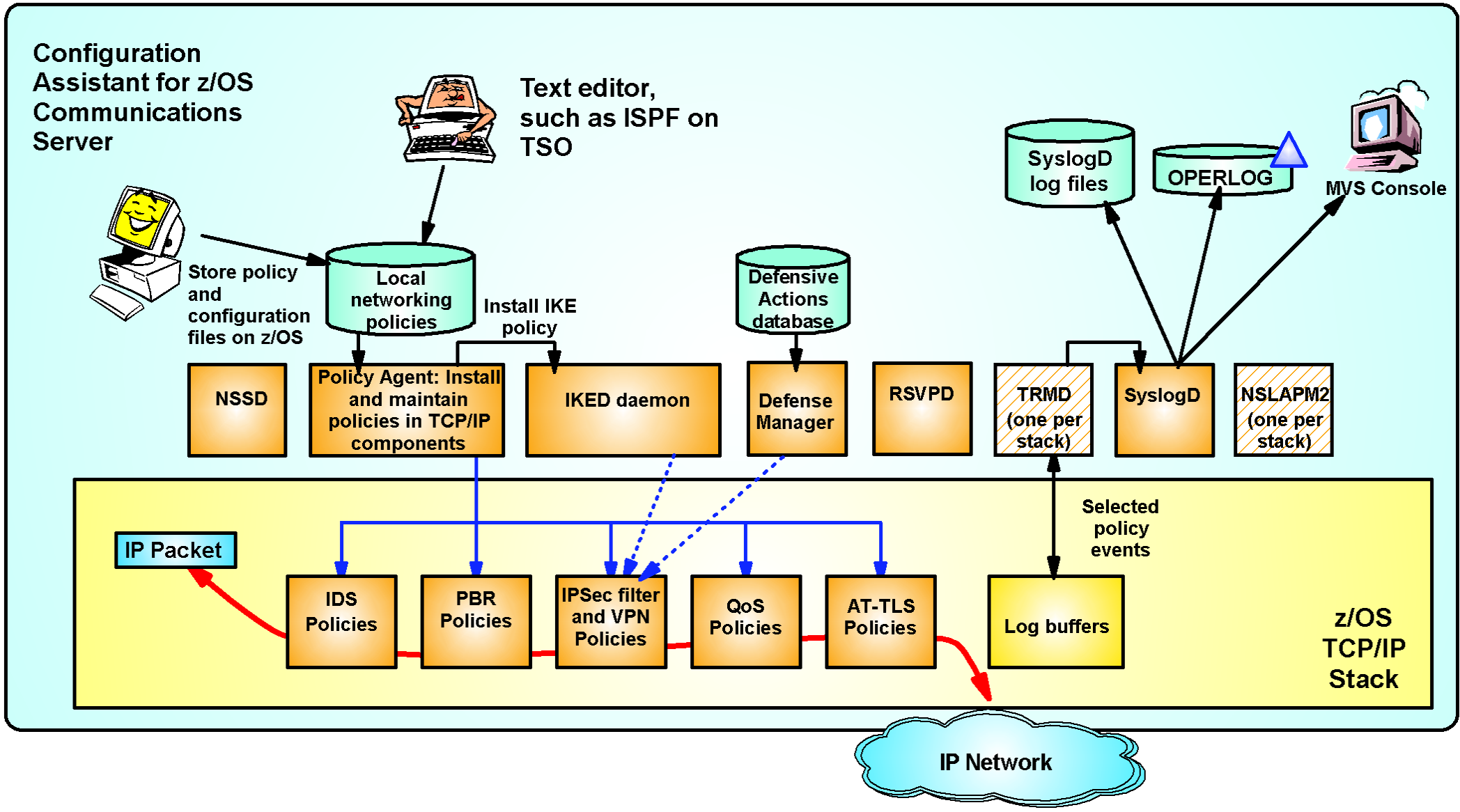
z/OS Communications Server networking policy overview

- **In z/OS V1R9, the general z/OS Communications Server policy infrastructure supports:**
  - **QoS** - Networking Quality of Service policies (TOS, Differentiated Services, VLAN priority, QDIO priority queues, etc.)
  - **IDS** - Intrusion Detection/Defense Services policies - scan, attack, flooding
  - **AT-TLS** - Application Transparent Transport Layer Security (SSL/TLS) policies
  - **IPSec IP filters** - IP filter policies - deny, permit
  - **IPSec VPNs** - Virtual Private Network policies - manual and dynamic VPN tunnel policies
  - **PBR** - Policy-based Routing policies - select network interface, first-hop router, MTU size

z/OS CS networking policies dynamically alter the way selected types of IP traffic is treated by TCP/IP on z/OS and in some cases how traffic is treated by equipment in the network

- **The general z/OS Communications Server policy infrastructure consists of:**
  - **Policy Agent** - installs and maintains policies in TCP/IP stacks (required for all policy types)
  - **TRMD** - formats and sends messages from the TCP/IP stack to SyslogD (required for all policy types)
  - **SyslogD** - UNIX System Services logging focal point (required for all policy types)
  - **IKED** - Internet Key Exchange Daemon, used for dynamic VPNs (required for IPSec dynamic VPNs)
  - **NSSD** - Network Security Server, centralized network security server (optional for IPSec VPNs)
  - **RSVPD** - Resource Reservation Protocol Daemon, QOS integrated services (optional for QoS)
  - **NSLAPM2** - QoS SNMP subagent (optional for QoS)
  - **DMD** - Defense Manager Daemon (dynamic defensive IP filters - added by z/OS V1R10 CS)

z/OS Communications Server policy infrastructure as of z/OS V1R10 CS



Which address spaces are needed for what?

- **Sample LPAR configuration with common INET and two TCP/IP stacks (Stack1 and Stack2) that both need networking policy support**

	Shared by all stacks on the LPAR					Stack1		Stack2	
	PAGENT	NSSD (note 1)	IKED	RSVPD	SYSLOGD	TRMDA	NSLAPM2A	TRMDB	NSLAPM2B
QoS	Required			Optional	Required	(Required)	Optional	(Required)	Optional
IDS	Required				Required	Required		Required	
AT-TLS	Required				Required				
IPSec filters	Required				Required	Required		Required	
IPSec static VPNs	Required				Required	Required		Required	
IPSec dynamic VPNs	Required	Optional	Required		Required	Required		Required	
PBR	Required				Required				

- **Note 1:** NSSD is really shared by all stacks in all LPARs in the NSSD domain (which could be a Sysplex or spa multiple Sysplex environment)
- **The basic infrastructure that is needed for most of the policy types consists of**
  - Per LPAR: Policy Agent, SyslogD
  - Per stack: TRMD

Configuration files and policy definition files - overview

Configuration and Policy definitions	Manuel edit (ISPF)	Configuration Assistant
Configuration files		
Policy Agent configuration file	Yes	No
SyslogD configuration file	Yes	No
IKED configuration file	Yes	Yes
NSSD configuration file	Yes	Yes
RSVPD configuration file	Yes	No
Policy definition files		
QoS Policy file	Yes	Yes
IDS Policy file	Yes	Yes
ATTLS Policy file	Yes	Yes
IPSec Policy file	Yes	Yes
PBR Policy file	Yes	Yes

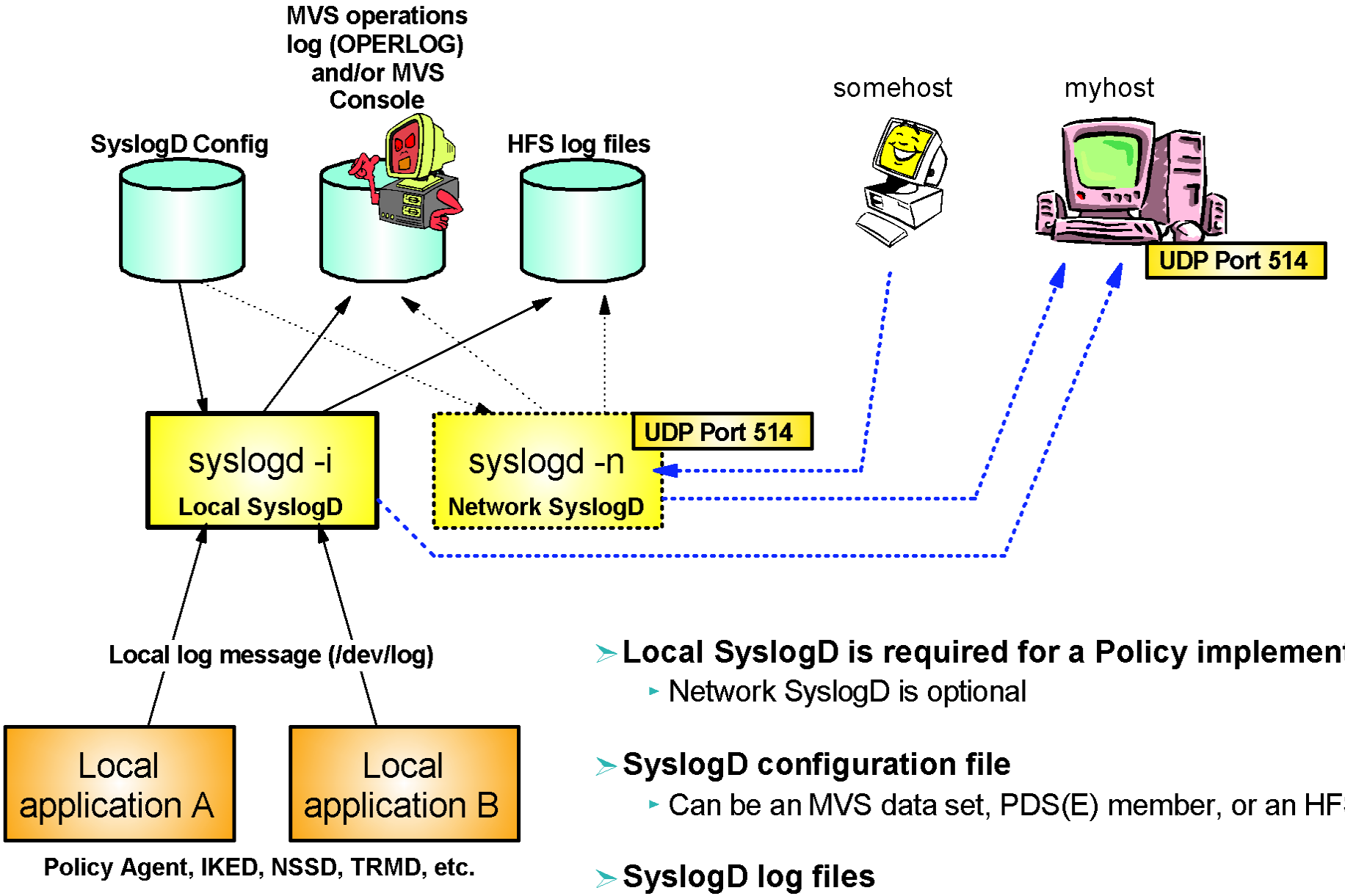
Most of the policy infrastructure components (address spaces you start) use a combination of configuration files, environment variables, and start options to control their start up processing

Per stack and policy type that you want to use, you must define a policy definition and store that in a file, which Policy Agent reads durin policy activation

- **All the above files can be either HFS files or MVS data sets, including members of PDS(E) libraries.**
  - I (personally) prefer MVS PDS(E) members for the following reasons:
    - I already have an MVS PDS(E) library structure in place for LPARs and TCP/IP stack configurations, such as PROFILE, OMPROUTE, etc.
    - There typically already is a backup/restore mechanism in place for these configuration data sets
    - Existing change management procedures are often based on PDS(E) library structures with staging, production, and backout libraries
    - Standard RACF profiles control who can access them in what way

SyslogD and TRMD

SyslogD overview



- **Local SyslogD is required for a Policy implementation**
  - Network SyslogD is optional
- **SyslogD configuration file**
  - Can be an MVS data set, PDS(E) member, or an HFS file
- **SyslogD log files**
  - Must be HFS files

SyslogD configuration file - basics

- **All messages to SyslogD are sent from local applications (using an AF_UNIX socket: /dev/log) along with information about facility name, priority, jobname, and user ID**
- **SyslogD configuration rules use this information to determine where to send the message that is being logged**

```
facility.priority                    destination
userid.jobname.facility.priority     destination
(hostspec).facility.priority         destination
```

Facility names

user	Message generated by a process (user).
mail	Message generated by mail system.
news	Message generated by news system.
uucp	Message generated by UUCP system.
daemon	This facility name is generally used by server processes. ▸ The FTPD server, the RSHD server, the REXECD server, the SNMP agent, and the SNMP subagent use this facility name to log trace messages.
auth/authpriv	Message generated by authorization daemon.
cron	Message generated by the clock daemon.
lpr	Message generated by the (USS lp command) print client.
local0-7	Names for local use. ▸ The z/OS UNIX Telnet server uses the local1 facility name for its log messages.
mark	Used for logging MARK messages.
kernel	Kernel log messages. ▸ z/OS does not generate any log messages with the kernel facility, and it does not accept log messages from local applications with the kernel facility. However, syslogd on z/OS is capable of receiving log messages over the network from other syslog daemons using the kernel facility. The kernel facility can be used in rules to direct these log messages to specific destinations.

Priority values

emerg/panic	A panic condition was reported to all processes.
alert	A condition that should be corrected immediately.
crit	A critical condition.
err(or)	An error message.
warn(ing)	A warning message.
notice	A condition requiring special handling.
info	A general information message.
debug	A message useful for debugging programs.
none	Do not log any messages for the facility.
*	Place holder used to represent all priorities.

Destinations

/hfs-file-name	HFS file name
@host	A syslog daemon on another host
user1,user2,...	A list of users.
/dev/console	The MVS console.
/dev/operlog	The MVS operlog log stream.
$SMF	SMF record 109

SyslogD HFS file location and naming

- **Location:**
  - Suggest you put them into a separate file system
    - Reduce impact of SyslogD message flooding on other file systems and applications
    - Simplifies monitoring for file system full-conditions (or approaching file system full)

- **File names:**
  - Two options
    - Fixed names
      - /var/syslog/logs/syslog.log
    - Variable names with symbol substitution, such as day, month, year being part of the directory and/or file name (requires that you implement some kind of automation that makes SyslogD re-initialize every midnight)
      - /var/syslog/%Y/%m/%d/syslog.log
  - My (personal) preference is fixed names
    - Easier to know which file to look into - always the same directory and file names
    - I find it easier to implement an archival process that works both at regularly scheduled intervals (such as every midnight) and that works at unscheduled points in time (such as when file system approaches full-condition during the middle of the day)

```
# USER1.TCPCS.TCPPARMS(SYSLOGDL)
#
*.*                          /var/syslog/logs/syslog.log
*.TRMD1.*.*                  /var/syslog/logs/trmd.log
*.TCPCS*.*.*                 /var/syslog/logs/tcpcs.log
*.INETD*.*.*                 /var/syslog/logs/inetd.log
*.OSNMP*.*.*                 /var/syslog/logs/osnmpd.log
*.PAGENT*.*.*                /var/syslog/logs/pagent.log
*.FTP*.*.*                   /var/syslog/logs/ftp.log
*.NAMED*.*.*                 /var/syslog/logs/named.log
```

If a message meets the criteria of multiple rules, it will be written to multiple destinations (in this example, all destinations are files).

Starting SyslogD

```
//SYSLOGDL PROC
//*
//* Start SyslogD
//*
//SYSLOGD EXEC PGM=SYSLOGD,REGION=0K,TIME=NOLIMIT,
//        PARM=('POSIX(ON) ENVAR("_CEE_ENVFILE=DD:MYENV")',
//             '/ -c -u -i -f //''USER1.TCPCS.TCPPARMS(SYSLOGDL)''')
//SYSPRINT DD SYSOUT=*
//MYENV    DD DSN=USER1.TCPCS.TCPPARMS(SYSLOGEV),DISP=SHR
//SYSIN    DD DUMMY
//SYSERR   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
```

-c
- Create log files and directories

-u
- Include userID and jobname

-i
- Local-only mode

-f
- Here is your configuration file

SyslogD is a UNIX process that will start and then 'fork' into another process

- If you use a procedure name that is less than 8 characters in length, the resulting address space name will be your procedure name suffixed with one - S SYSLOGD - the resulting address space will be SYSLOGD1
- If you use a procedure name of 8 characters, the resulting address space name will have the same name as your procedure name - S SYSLOGDL - the resulting address space will also be named SYSLOGDL

SyslogD does not accept an MVS console STOP command

- You can stop it using UNIX services if you know what the Process ID (PID) is
  - SyslogD stores its PID in /etc/syslog.pid
  - A network-only SyslogD stores its PID in /etc/syslog_net.pid
- You can stop it by cancelling the address space (this is where you need to know the resulting address space name)
  - C SYSLOGD1 - or - C SYSLOGDL

Archiving SyslogD HFS files without losing any messages (!)

1. SyslogD has opened and writes to trmd.log
2. trmd.log is renamed to trmd.log.renamed
3. SyslogD continues to write to the file that is now named trmd.log.renamed
4. Signal SyslogD to re-initialize which makes SyslogD close trmd.log.renamed and create a new trmd.log to which it starts to write, freeing up trmd.log.renamed
5. Archive trmd.log.renamed and delete trmd.log.renamed after the archival is done

- **You can run this process as many times as you want/need to during the day**

- **Sample REXX USS services to rename and signal SyslogD:**
  - address syscall 'rename' oldname newname
  - address syscall 'kill' NetworkSyslogDPID SIGHUP

- **Sample REXX code to determine if the SyslogD file system is approaching full condition**

```
used = 0
address Syscall 'open /var/syslog/logs/trmd.log' o_rdonly
if retval <> -1 then do
   fd = retval
   address Syscall 'fstatvfs' fd st.
   used = st.stfs_inuse*100/st.stfs_total
   address Syscall 'close' fd
end
```

Sample archival job to GDG data set

```
//ARCHSYSL JOB 1,ALFRED,CLASS=A,MSGCLASS=X,NOTIFY=USER1
//*
//* Archive a SYSLOGD file to a GDG data set
//*
//ARCHIVE  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//LOGFILE  DD PATH='/var/syslog/logs/trmd.log.renamed',
//            PATHOPTS=(ORDONLY)
//SYSPRINT DD SYSOUT=*
//GDGOUT   DD DSN=USER1.ALFRED.TRMD.LOG(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=3390,VOL=SER=DB2ABC,
//            SPACE=(CYL,(1,1)),
//            DCB=(RECFM=VB,LRECL=1024,BLKSIZE=0)
//SYSTSIN  DD *
 OCOPY INDD(LOGFILE) OUTDD(GDGOUT)
//*
//* Delete the archived SYSLOGD file
//*
//DELETE   EXEC PGM=BPXBATCH,COND=(0,LT),
//            PARM='SH rm /var/syslog/logs/trmd.log.renamed'
//STDOUT   DD PATH='/tmp/archsysl.stdout',
//            PATHMODE=SIRWXU,PATHOPTS=(OWRONLY,OCREAT)
//STDERR   DD PATH='/tmp/archsysl.stderr',
//            PATHMODE=SIRWXU,PATHOPTS=(OWRONLY,OCREAT)
```

Copy the HFS file to GDG(+1) MVS data set

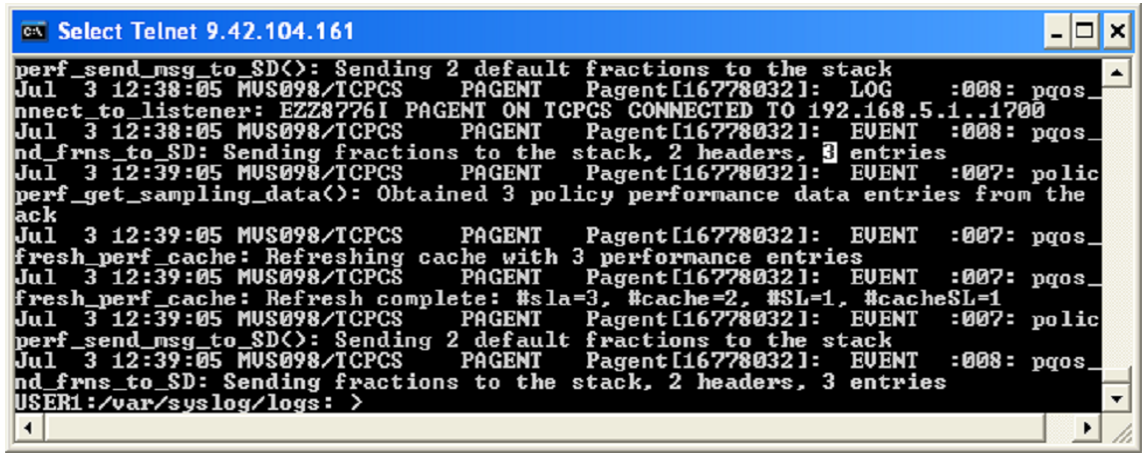
Delete the HFS file when archival done successfully

If delete produced errors, print them out to a SYSOUT file

```
//*
//* Copy stdout and stderr from the delete
//* step to a sysout file
//*
//COPYPRT  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//STDOUT   DD PATH='/tmp/archsysl.stdout',
//            PATHOPTS=(ORDONLY)
//STDERR   DD PATH='/tmp/archsysl.stderr',
//            PATHOPTS=(ORDONLY)
//SYSPRINT DD SYSOUT=*
//HFSOUT   DD SYSOUT=*,DCB=(RECFM=V,LRECL=256)
//SYSTSIN  DD *
 OCOPY INDD(STDOUT) OUTDD(HFSOUT)
 OCOPY INDD(STDERR) OUTDD(HFSOUT)
/*
//
```

Browsing SyslogD HFS files

- **The lines are very long (up to 1024 characters)**
  - If you log in to the z/OS UNIX Telnet server, the telnet client will fold the lines onto as many lines as needed



  - With ISPF browse, you need to scroll right/left many times, which is confusing

```
Jul  3 12:40:05 MVS098/TCPCS     PAGENT    Pagent[16778032]:   EVENT   :007: pqos_re
Jul  3 12:40:05 MVS098/TCPCS     PAGENT    Pagent[16778032]:   EVENT   :007: pqos_re
Jul  3 12:40:05 MVS098/TCPCS     PAGENT    Pagent[16778032]:   EVENT   :007: policy_
Jul  3 12:40:05 MVS098/TCPCS     PAGENT    Pagent[16778032]:   EVENT   :008: pqos_se
******************************** Bottom of Data ********************************
```

scroll right to see the remaining part(s) of the message

  - You may want to create some customized REXX logic to browse the SyslogD files

```
Jul 3 12:43:05  EVENT  :007: pqos_refresh_perf_cache: Refreshing cache with 3
  performance entries
Jul 3 12:43:05  EVENT  :007: pqos_refresh_perf_cache: Refresh complete: #sla=3,
  #cache=2, #SL=1, #cacheSL=1
******************************** BOTTOM OF DATA ********************************
```

TRMD

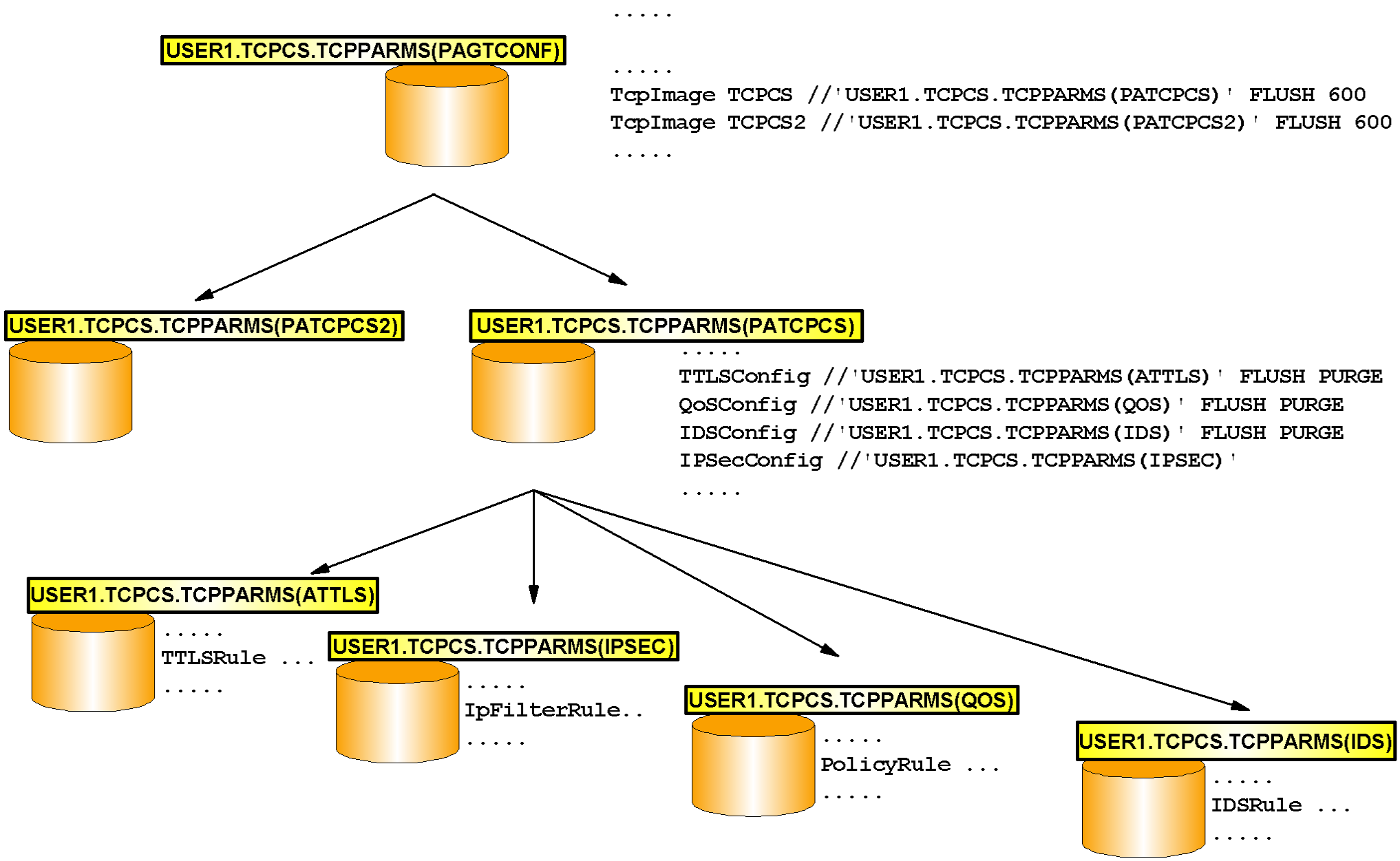
- **TRMD is stack-specific**
  - It determines which stack to use based on the TCPIPJOBNAME in its resolver configuration file
- **TRMD procedure**

```
//TRMDA     PROC
//*
//TRMD   EXEC PGM=EZATRMD,REGION=4096K,TIME=NOLIMIT,
//  PARM=('POSIX(ON) ALL31(ON)',
//  'ENVAR("_CEE_ENVFILE=DD:MYENV")/')
//MYENV    DD DSN=USER1.TCPCS.TCPPARMS(TRMDENV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
```

- **TRMD environment variables in my USER1.TCPCS.TCPPARMS(TRMDENV) member:**
  - RESOLVER_CONFIG=//'USER1.TCPCS.TCPPARMS(TCPDATA)'
- **TRMD forks, so the resulting address space becomes TRMDA1 in this example**
  - TRMD does support a STOP command
    - P TRMDA1
  - TRMD can also be stopped via a UNIX kill command, but it doesn't store its PID in any specific file (you can still determine it by using a "ps -ef | grep TRMD" command)

Policy Agent (PAGENT)

One way of structuring the Policy Agent configuration and policy files



Policy Agent start procedure and envionment variables

Policy Agent start procedure

```
//PAGENT   PROC P='-d 0'
//*
//* Policy Agent start procedure
//*
//PAGENT   EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//        PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:MYENV")/&P'
//*
//MYENV    DD DSN=USER1.TCPCS.TCPPARMS(PAGTENV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

Policy Agent environment variables

```
PAGENT_CONFIG_FILE=//'USER1.TCPCS.TCPPARMS(PAGTCONF)'
PAGENT_LOG_FILE=SYSLOGD
RESOLVER_CONFIG=//'USER1.TCPCS.TCPPARMS(TCPDATA)'
```

Policy Agent root and stack-specific configuration files

Policy Agent configuration file

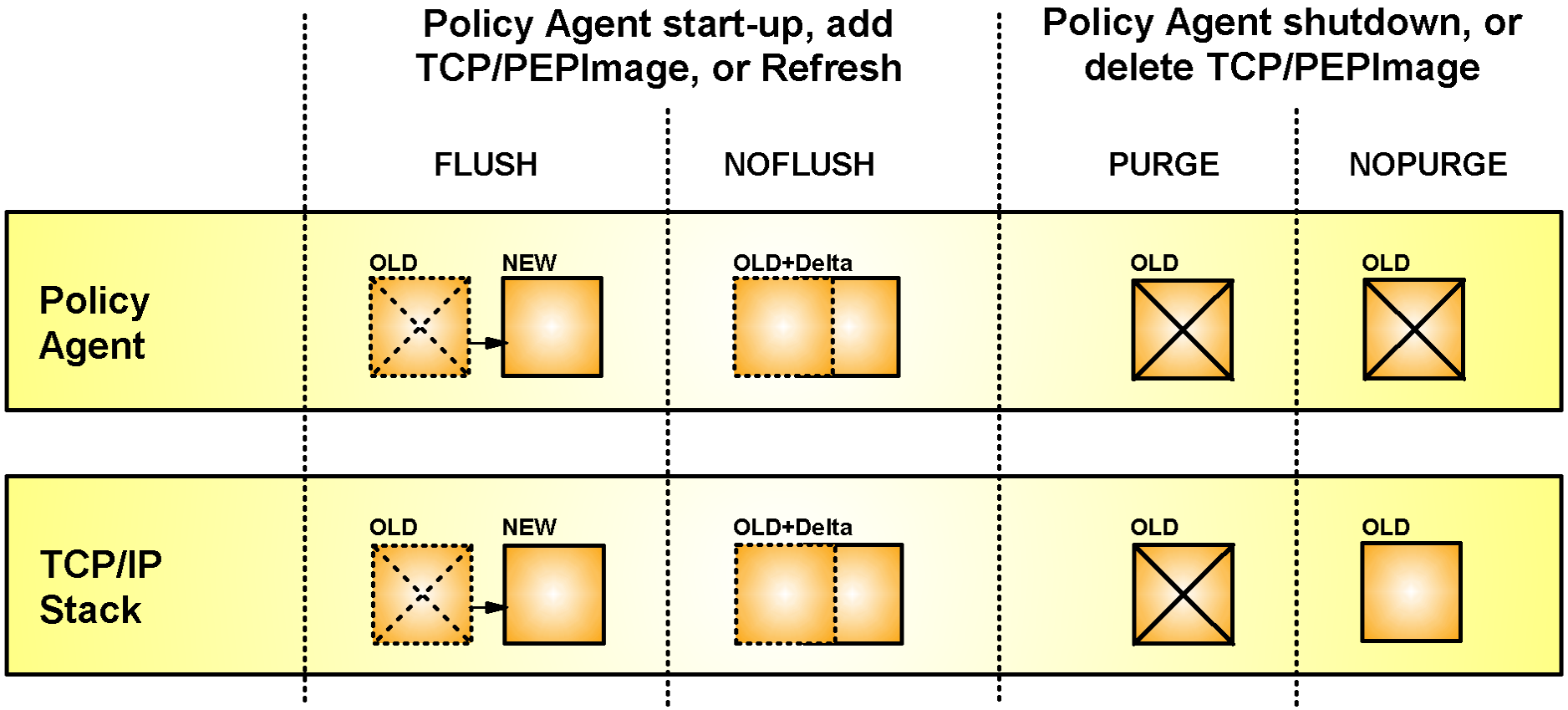
```
##
## USER1.TCPCS.TCPPARMS(PAGTCONF)
## Policy Agent root configuration file that is pointed to be
## the PAGENT_CONFIG_FILE=//'USER1.TCPCS.TCPPARMS(PAGTCONF)'
## environment variable when starting Policy Agent
## Image: mvs098  - Stack: TCPCS
##
TcpImage TCPCS //'USER1.TCPCS.TCPPARMS(PATCPCS)' FLUSH 600
##
## Image: mvs098  - Stack: TCPCS2
##
TcpImage TCPCS2 //'USER1.TCPCS.TCPPARMS(PATCPCS2)' FLUSH 600
```

TCPCS stack-specific Policy Agent configuration file

```
## USER1.TCPCS.TCPPARMS(PATCPCS)
## This configuration is pointed to from the Policy agent
## root configuration file for stack TCPCS on mvs098
##
TTLSConfig //'USER1.TCPCS.TCPPARMS(ATTLS)' FLUSH NOPURGE
QoSConfig //'USER1.TCPCS.TCPPARMS(QOS)' FLUSH NOPURGE
IDSConfig //'USER1.TCPCS.TCPPARMS(IDS)' FLUSH NOPURGE
IPSecConfig //'USER1.TCPCS.TCPPARMS(IPSEC)'
##
```

What does FLUSH/NOFLUSH and PURGE/NOPURGE mean?

- **Policy definitions are stored both inside Policy Agent and the TCP/IP stacks**
  - There may be differences - time-based policies exist in Policy Agent, but only in the stack when the time interval is current
  - You may have shut Policy Agent down, but the stacks may retain their policies



FLUSH/NOFLUSH and PURGE/NOPURGE - more details

Parameter	When used	Results
FLUSH and NOFLUSH	Used after policies are read without errors, when triggered by the following events: ▸ Policy Agent starts up ▸ TCPImage or PEPInstance statements are added ▸ Modify REFRESH command	For FLUSH: ▸ All policies are deleted from Policy Agent and the TCP/IP stack ▸ QoS policy statistics are reset to 0 For NOFLUSH ▸ No policies are deleted from Policy Agent or the TCP/IP stack ▸ Policies removed from the configuration are not deleted from the Policy Agent or the TCP/IP stack
PURGE and NOPURGE	Used at ▸ Policy Agent shutdown ▸ TCPImage or PEPImage statements are deleted	For PURGE: ▸ All policies are deleted from Policy Agent and the TCP/IP stack For NOPURGE: ▸ All policies are deleted from Policy Agent ▸ No policies are deleted from the TCP/IP stack

PBR - Cannot be configured; will always do FLUSH and NOPURGE
IDS - Configurable (TCPImage/PEPImage, IDSConfig)
IPSec - Cannot be configured; will always do FLUSH and NOPURGE
QoS - Configurable (TCPIMage/PEPImage)
AT-TLS - Configurable (TCPImage/PEPImage, TTLSConfig)

Sample policy definition file (snippet of ATTLS policies)

- **Policy Agent ATTLS policy file for stack TCPCS on LPAR mvs098**

```
##
## AT-TLS Policy Agent Configuration file for:
##    Image: MVS098
##    Stack: TCPCS
##
## Created by the IBM Configuration Assistant for z/OS Communications Server
## Version 1 Release 9
## Backing Store = 'USER1.ALFRED.POLICY.BACKSTOR(DEFAULT)'
## FTP History:
## 2007-06-05 10:02:27  user1 to mvs098o.tcp.raleigh.ibm.com
##   Connections to QDIO4: just TLS, DVIPA1: TLS w. client auth
## 2007-06-05 09:39:25  user1 to mvs098o.tcp.raleigh.ibm.com
##   Added application control for TN3270 server port 2025
## 2007-06-01 01:36:13  user1 to mvs098o.tcp.raleigh.ibm.com
##   Added FTP port 4021 to ATTLS configuration
## 2007-06-01 12:54:27  user1 to mvs098o.tcp.raleigh.ibm.com
##   Support for TN3270 server port 2025 (only ATTLS port in this config
##
TTLSRule                              Conn_to_QDIO4~1
{
  LocalAddrRef                        addr1
  RemoteAddrSetRef                    addr2
  LocalPortRangeRef                   portR1
++++++ (Many more statements) +++++++++++++++
```

Controling Policy Agent

- **Policy Agent does not fork after it has been started, so the resulting address space name is the same name as you started (typically PAGENT)**
- **Policy Agent supports MVS console modify commands**
- **An F PAGENT,REFRESH command can be used to ask PAGENT to reread all its configuration and policy flat files and re-install those**
  - Useful after you have made an update to your policy flat files
  - A REFRESH command will refresh all policies

```
12.33.57  f pagent,refresh
12.33.57  EZZ8443I PAGENT MODIFY COMMAND ACCEPTED
12.33.58  EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR TCPCS : IDS
12.33.58  EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR TCPCS : QOS
12.33.58  EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR TCPCS : TTLS
```

- **Policy Agent also supports an F PAGENT,UPDATE command that can be used to have PAGENT only update those policies that have been changed as opposed to doing a total refresh**

```
12.35.37  f pagent,update
12.35.37  EZZ8443I PAGENT MODIFY COMMAND ACCEPTED
12.35.37  EZZ8771I PAGENT CONFIG POLICY PROCESSING COMPLETE FOR TCPCS : NONE
```

- **Policy Agent can be stopped using a P command**

```
p pagent
```

Monitoring which policies Policy Agent currently manages

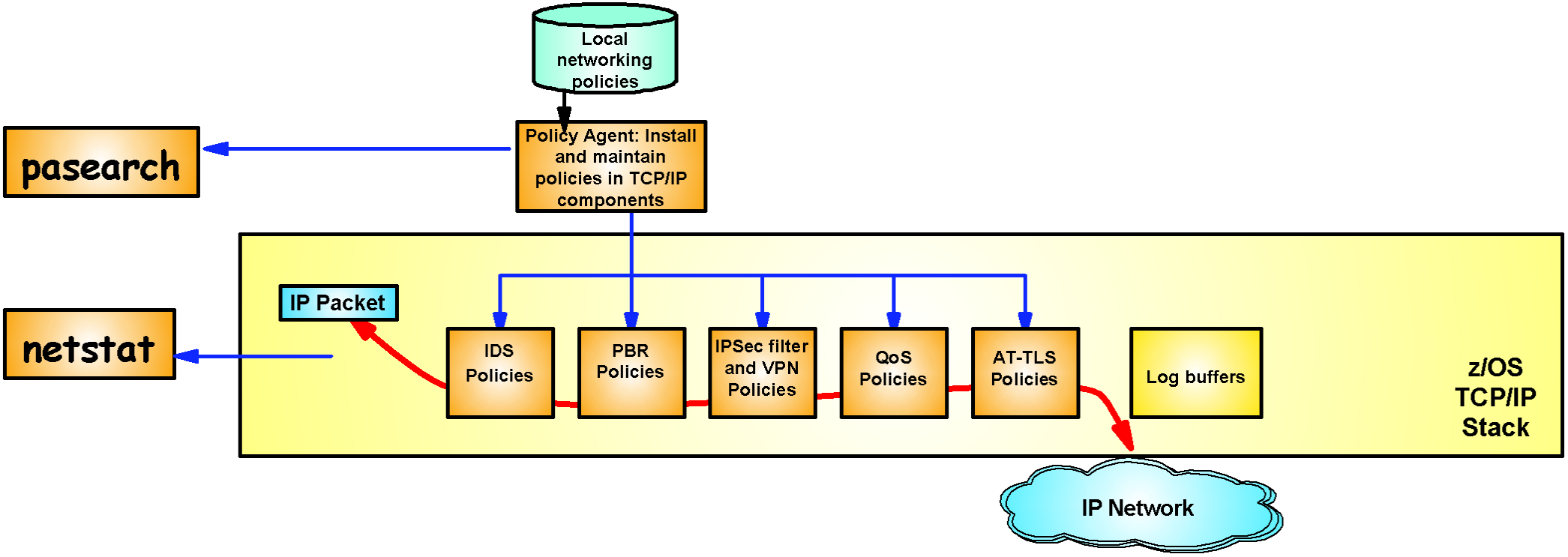
- **The pasearch command is a UNIX shell command that uses a Policy Agent programming interface to retrieve details about the policies Policy Agent currently knows about**

```
USER1:/u/user1: >pasearch -A -t
TCP/IP pasearch CS V1R9                    Image Name: TCPCS
  Date:                  07/05/2007        Time:  13:00:02
  TTLS Instance Id:      1183653238

policyRule:              Conn_to_QDIO4~1
  Rule Type:             TTLS
  Version:               3                 Status:             Active
  Weight:                255               ForLoadDist:        False
  Priority:              255               Sequence Actions:   Don't Care
  No. Policy Action:     3
  policyAction:          gAct1
   ActionType:           TTLS Group
   Action Sequence:      0
  policyAction:          eAct1~ABC_FTP_4021
   ActionType:           TTLS Environment
   Action Sequence:      0
  policyAction:          cAct1~ABC_FTP_4021
   ActionType:           TTLS Connection
   Action Sequence:      0
  Time Periods:
   Day of Month Mask:
   First to Last:        11111111111111111111111111111111
 +++++ (much more output) +++++
```

What's the difference between pasearch and netstat?

- **pasearch**
  - Tells you what Policy Agent knows about, which is the full content of the current policy definition files. This include time-based policies that may not currently be active.

- **netstst**
  - Tells you what the stack knows about, which is the currently active policy definitions that Policy Agent has installed into the stack.
  - There are not netstat reports for all policy types
    - SLAP - QoS policies
    - IDS - IDS policies
    - TTLS - AT-TLS connections



Pasearch is a UNIX shell command only - what about TSO and NetView environments?

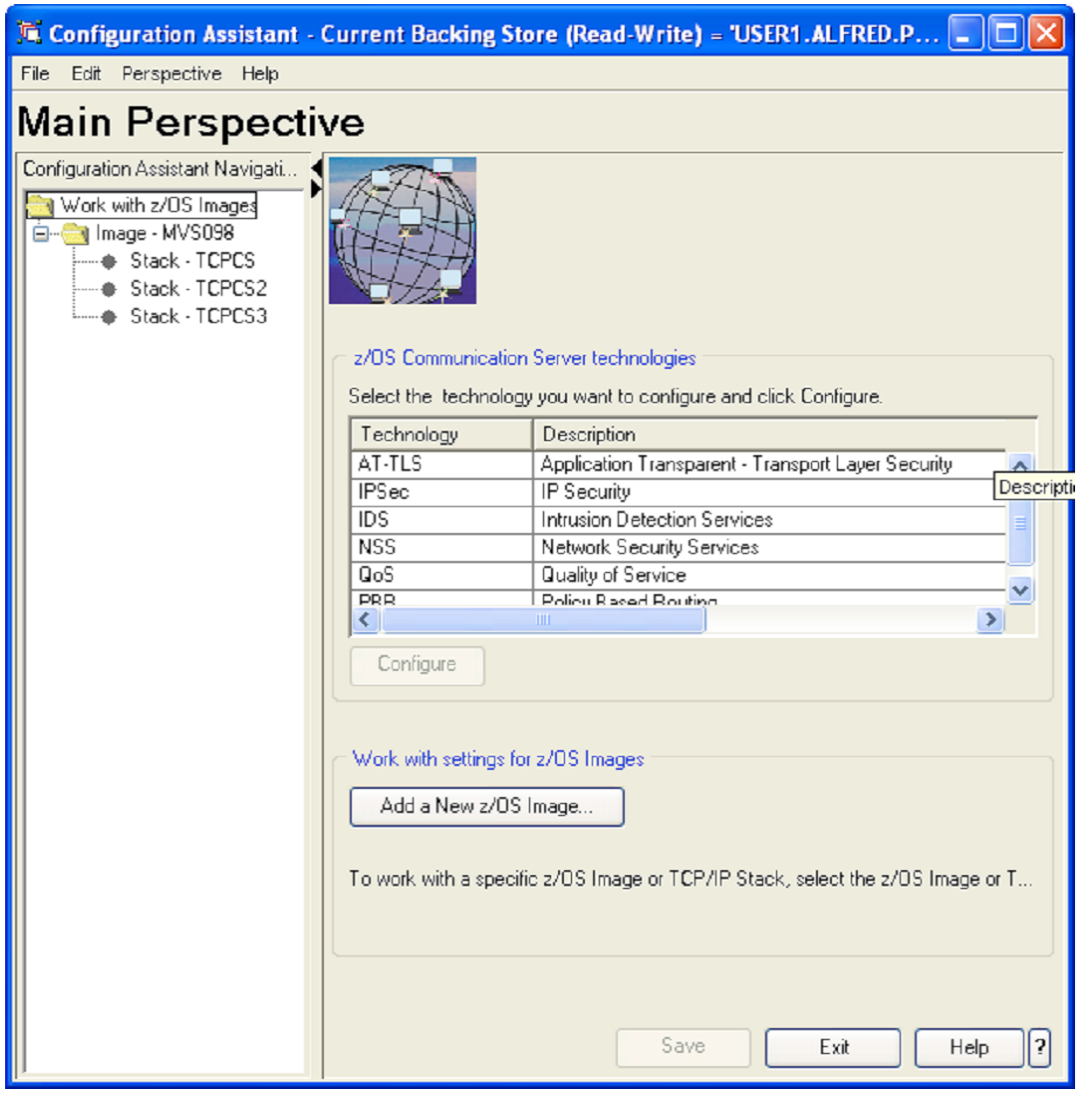
```
/* rexx */
parse arg input

parse var input number mycmd
if datatype(number,'N') then
   maxlines = number
else do
   maxlines = 9999
   mycmd = input
end

if syscalls('ON')>3 then do
  say 'Unable to establish the SYSCALL environment'
  return
end
environment.0 = 2;
environment.1 = "PATH=/bin:/usr/sbin"
environment.2 = "LIBPATH=/lib:/usr/lib"
call bpxwunix mycmd,,stdout.,stderr.,environment.

outlines = 0
if stdout.0 > 0 then do i=1 to stdout.0
   if i > maxlines then do
      x=maxlines
      parm='MAX output lines ('||x||') reached'
      parm=parm||' - report truncated'
      say parm
      leave
   end
   say stdout.i
   outlines = outlines + 1
end
```

```
if stderr.0 > 0 then do i=1 to stderr.0
   if i > maxlines-outlines then do
      x=maxlines
      parm='MAX output lines ('||x||') reached'
      parm=parm||' - report truncated'
      say parm
      leave
   end
   say stderr.i
end
exit
```

- **From TSO:**
  - abccmdts '30 pasearch -A -t'
    - or without limiting output lines:
  - abccmdts 'pasearch -A -t'
- **From NetView:**
  - netvasis abccmdts 'pasearch -A -t'
- **This sample REXX can be used with other UNIX Shell commands, such as**
  - trmdstat
  - ipsec
  - nssctl

Configuration Assistant

Configuration Assistant for z/OS CS networking policies

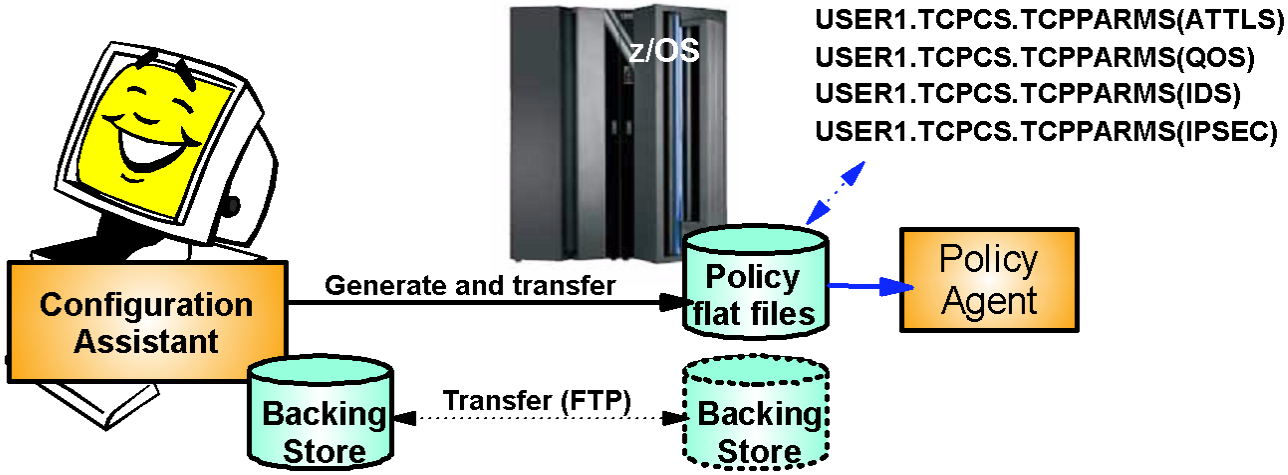


- **Download of the Configuration Assistant for z/OS CS V1R7, V1R8, and V1R9:**

http://www.ibm.com/support/docview.wss?rs=852&context=SSSN3L&dc=D400&uid=swg24013160&loc=en_US&cs=UTF-8&lang=en&rss=ct852other

Configuration Assistant files

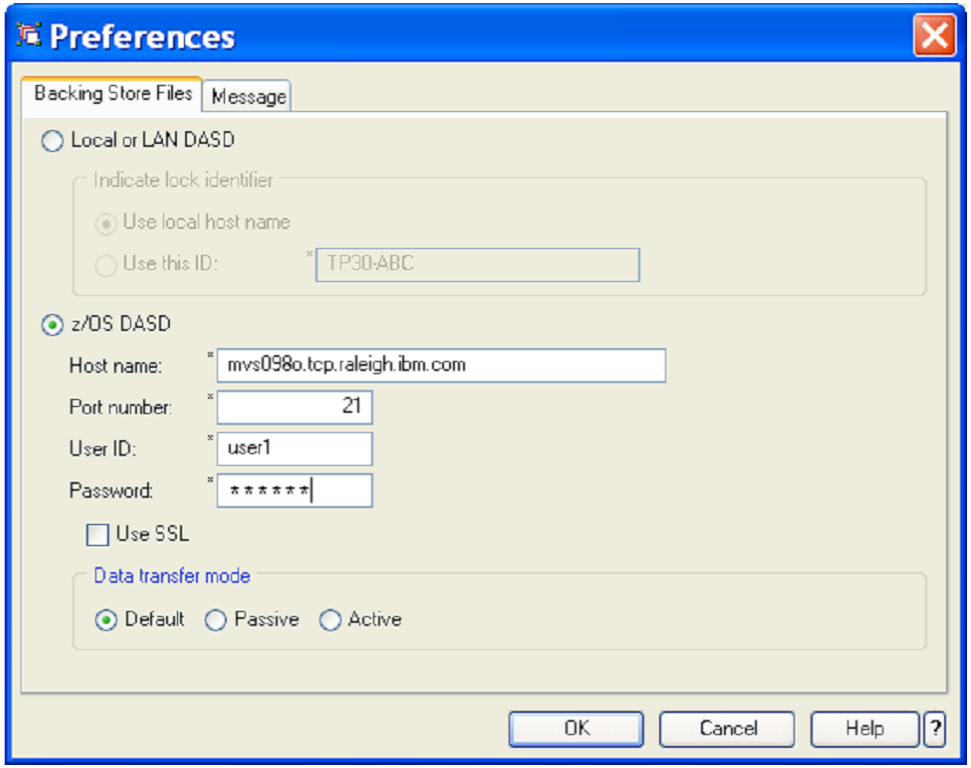
- **The configuration assistant tool reads and stores all policy-related information in binary form in the backing store file**
  - Think of it as a binary version of all your z/OS CS networking policy definitions
  - You can maintain policies for many LPARs, stacks, and policy types in a single backing store file
  - If all policies are maintained by the same people, then I use a single backing store file per sysplex
    - Allows me to reuse some of the defintions, such as traffic descriptors across stacks
- **The backing store file may reside on your Windows workstation, on a LAN server (SMB server), or on z/OS in an HFS file or MVS data set**
  - If on z/OS, open/save of the backing store file results in an FTP transfer to/from z/OS
  - The backing store file is protected against updates by more than one user at a time
    - Locking technology allows one user to update, others to access in read-only mode
- **When a policy has been created, the configuration assistant can generate the policy flat file that can b read by Policy Agent - and transfer it to z/OS using FTP**
- **The policy flat file is a text file that is read by the Policy Agent**
  - While the flat file can be altered with a text editor, such as ISPF, any such changes are not reflected back into the backing store file. As such, we recommend that you make any required changes through the Configuration Assistan and then re-upload the new flat files.

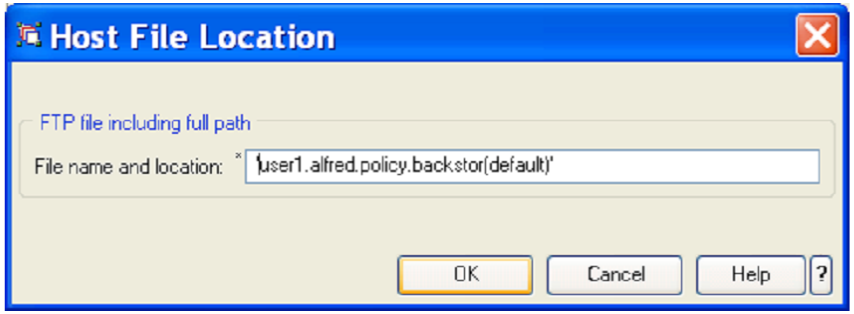


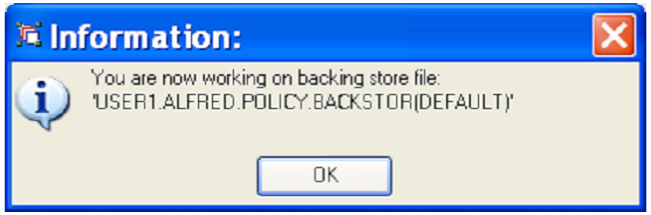
Backing store file can in z/OS V1R9 be saved on z/OS

- **Allow backing store files to be stored on z/OS DASD via FTP directly from the Configuration Assistant**

- **The Configuration Assistant saves the user's configuration data in a file called the backing store file.**
  - Users can manage different sets of configuration information by keeping them in different backing store files.
  - In V1R9, backing store files can now be stored on z/OS DASD using FTP as well as on the local file system. This allows for easier sharing of backing store files between multiple users.

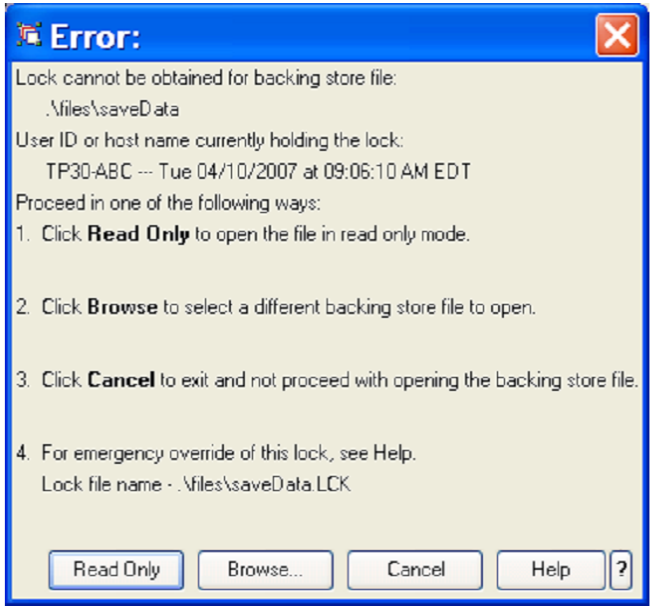


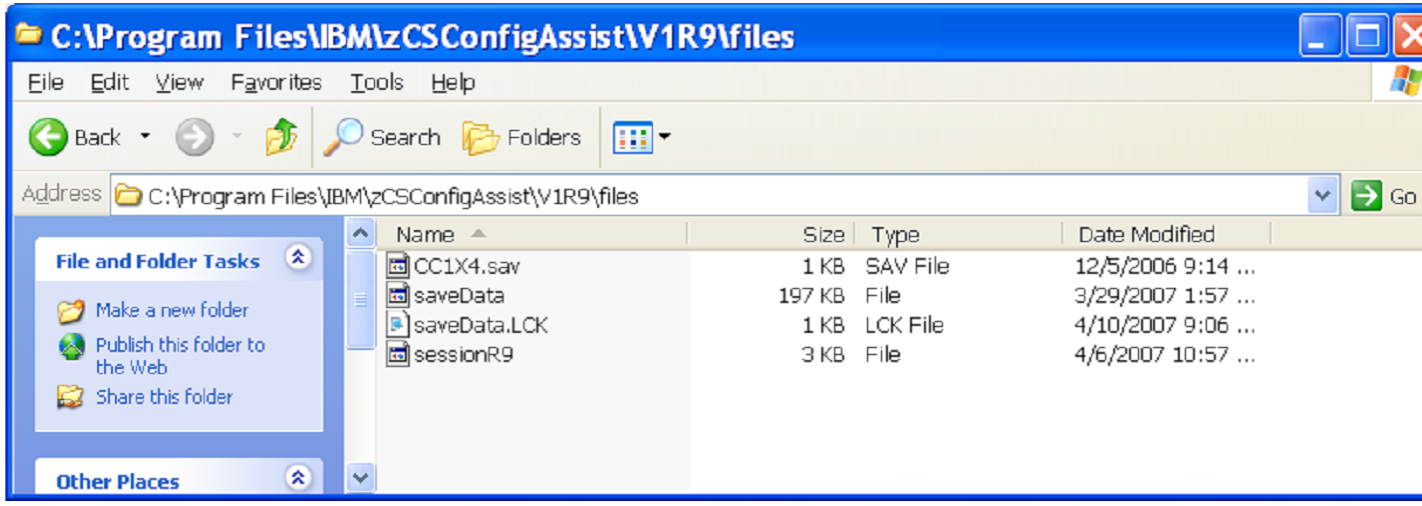




Protecting against inadvertant overlapping updates - backing store file locking mechanism is now in place

- **When a backing store file is opened in the Configuration Assistant, a lock file is created which contains a lock ID value, and the date and time at which the lock file was created.**
  - When you specify "Local or LAN DASD" as your preference, you may create a lock ID value of your own or use the host name of the workstation which is running the Configuration Assistant.
  - When you specify "z/OS DASD" as your preference, the lock ID value is the user id that is used to establish the FTP connection.
  - If needed, the locking mechanism can be circumvented by manually deleting the lock file.
- **The backing store locking solution was developed as part of z/OS V1R9, but has been rolled back to the z/OS V1R7 and V1R8 configuration assistant**

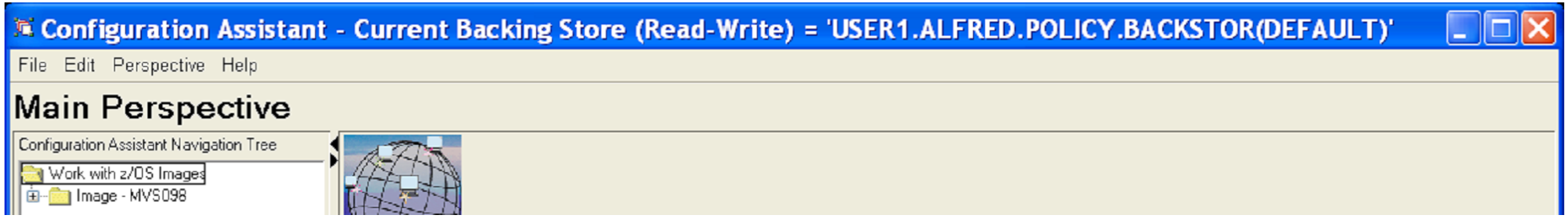




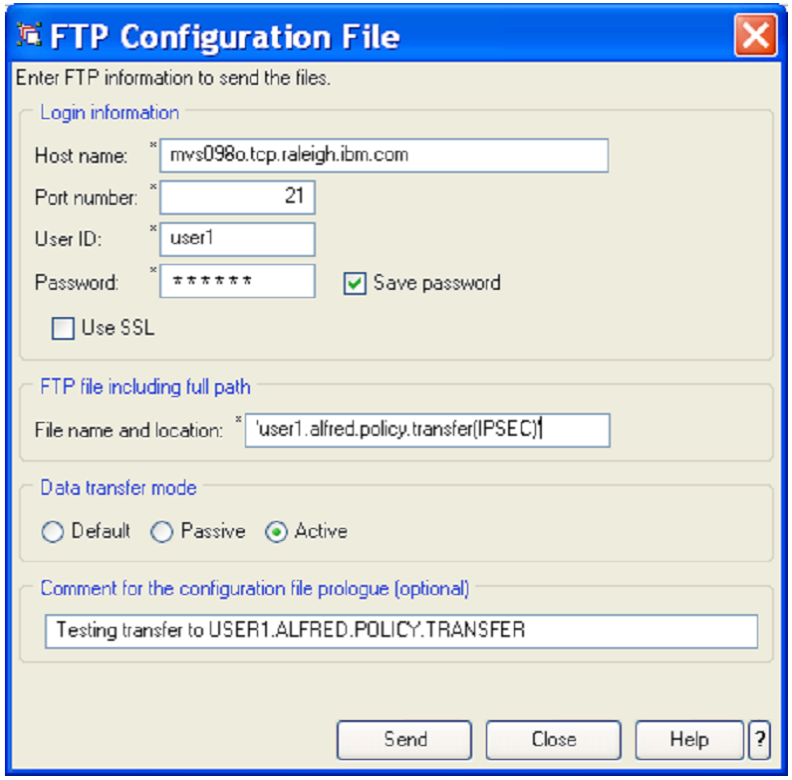
Working with backing store files on z/OS

- **In this sample, I use a PDS library - USER1.ALFRED.POLICY.BACKSTOR**
  - When I save my backing store as member DEFAULT in that library, a separate sequential MVS data set is allocated as the lock file USER1.ALFRED.POLICY.BACKSTOR.DEFAULT.LCK
  - The lock file only exists while someone has the backing store file opened in read-write mode

```
  Menu  Options  View  Utilities  Compilers  Help
------------------------------------------------------------------------------
DSLIST - Data Sets Matching USER1.ALFRED.POLICY            Data Set - Browsed
Command ===>                                                Scroll ===> PAGE

Command - Enter "/" to select action                Message          Volume
------------------------------------------------------------------------------
        USER1.ALFRED.POLICY.BACKSTOR                Browsed          CPDLB0
        USER1.ALFRED.POLICY.BACKSTOR.DEFAULT.LCK    Browsed          CPDLB0
****************************** End of Data Set list ****************************
```



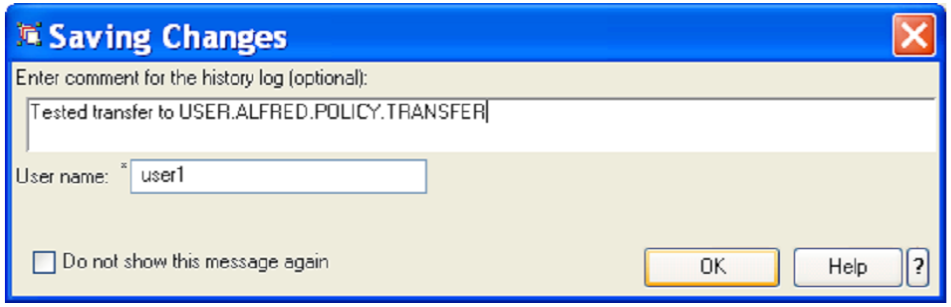
Policy flat file prolog (change log) maintained for auditing

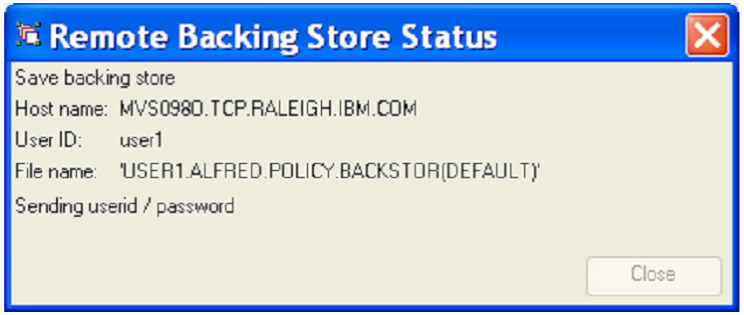


Prolog in IPSec Policy flat file that was transferred to z/OS

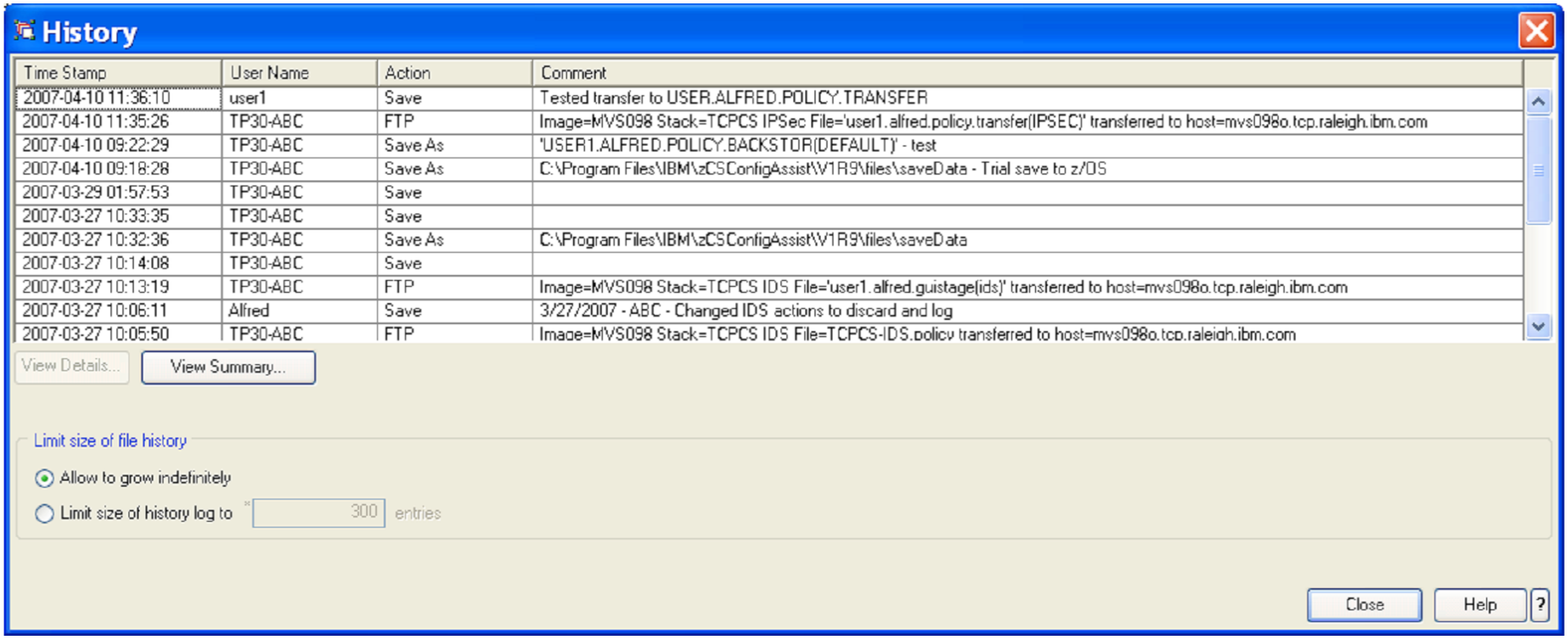
```
##
## IPSec Policy Agent Configuration file for:
##    Image: MVS098
##    Stack: TCPCS
##
## Created by the IBM Configuration Assistant for z/OS Communications Se
## Version 1 Release 9
## Backing Store = 'USER1.ALFRED.POLICY.BACKSTOR(DEFAULT)'
## FTP History:
## 2007-04-10 11:35:12  user1 to mvs098o.tcp.raleigh.ibm.com
##   Testing transfer to USER1.ALFRED.POLICY.TRANSFER
## 2007-03-27 09:54:33  user1 to mvs098o.tcp.raleigh.ibm.com
##   Added local Multicast address for OSPF
## 2007-03-27 09:29:13  user1 to mvs098o.tcp.raleigh.ibm.com
##   IPSec filter rules - initial configuration
##
```

- **Immediately following an FTP transfer, a backing store file save function will be performed**

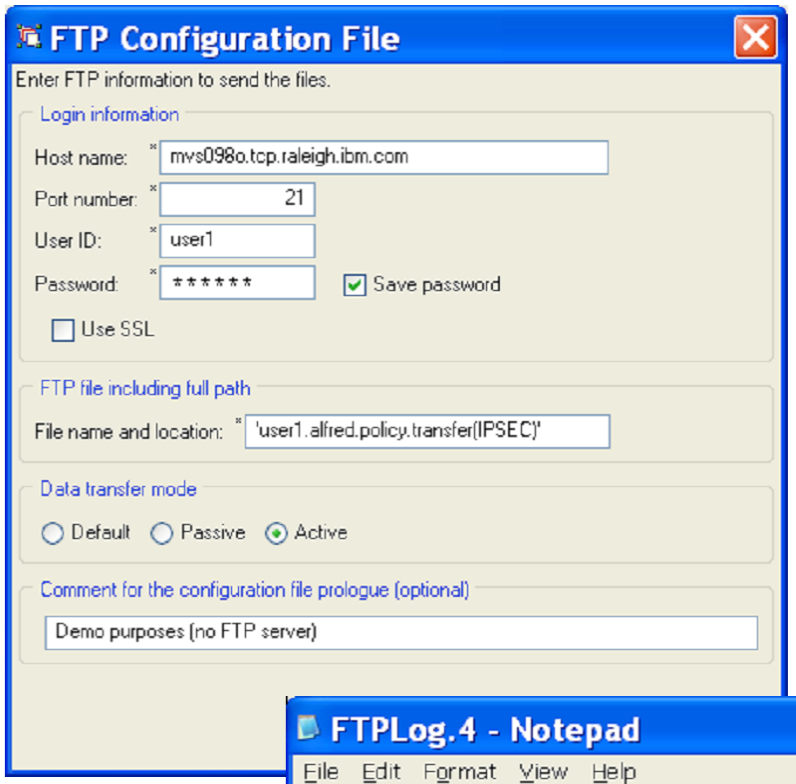


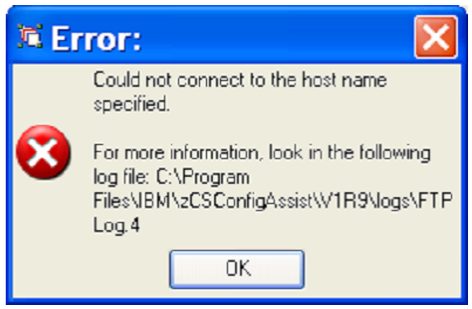


History information with user-entered comments for auditing purposes whenever the backing store file is saved

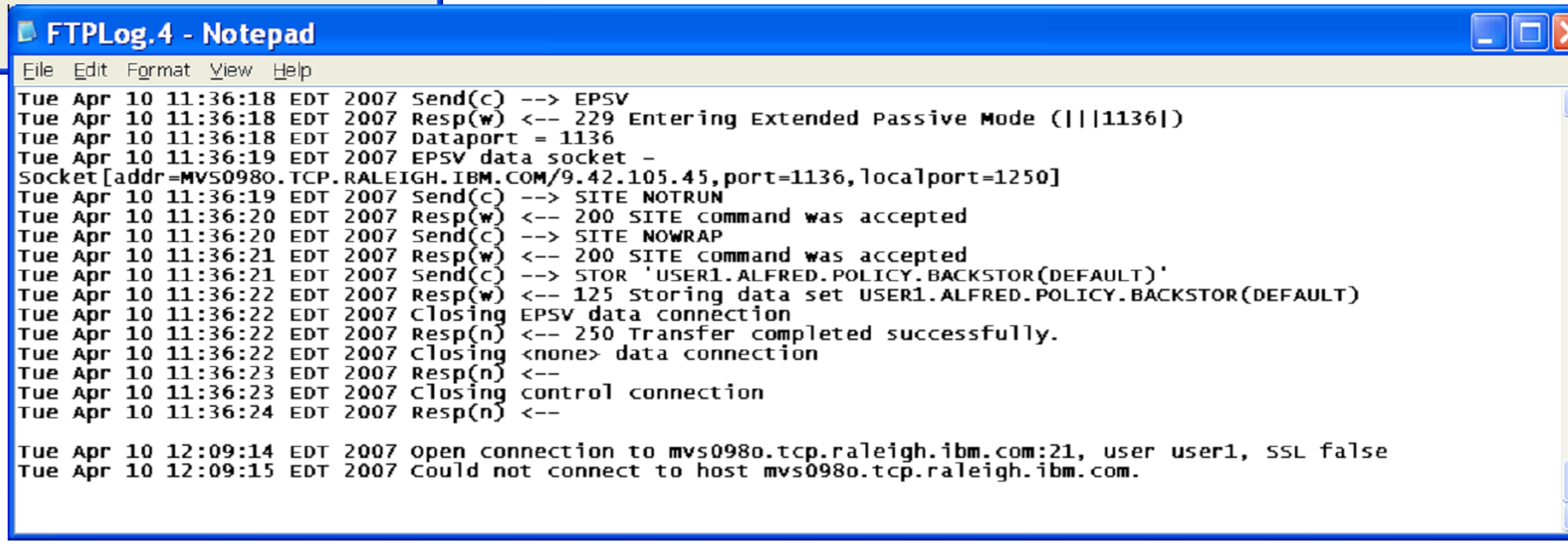


Configuration Assistant FTP functions

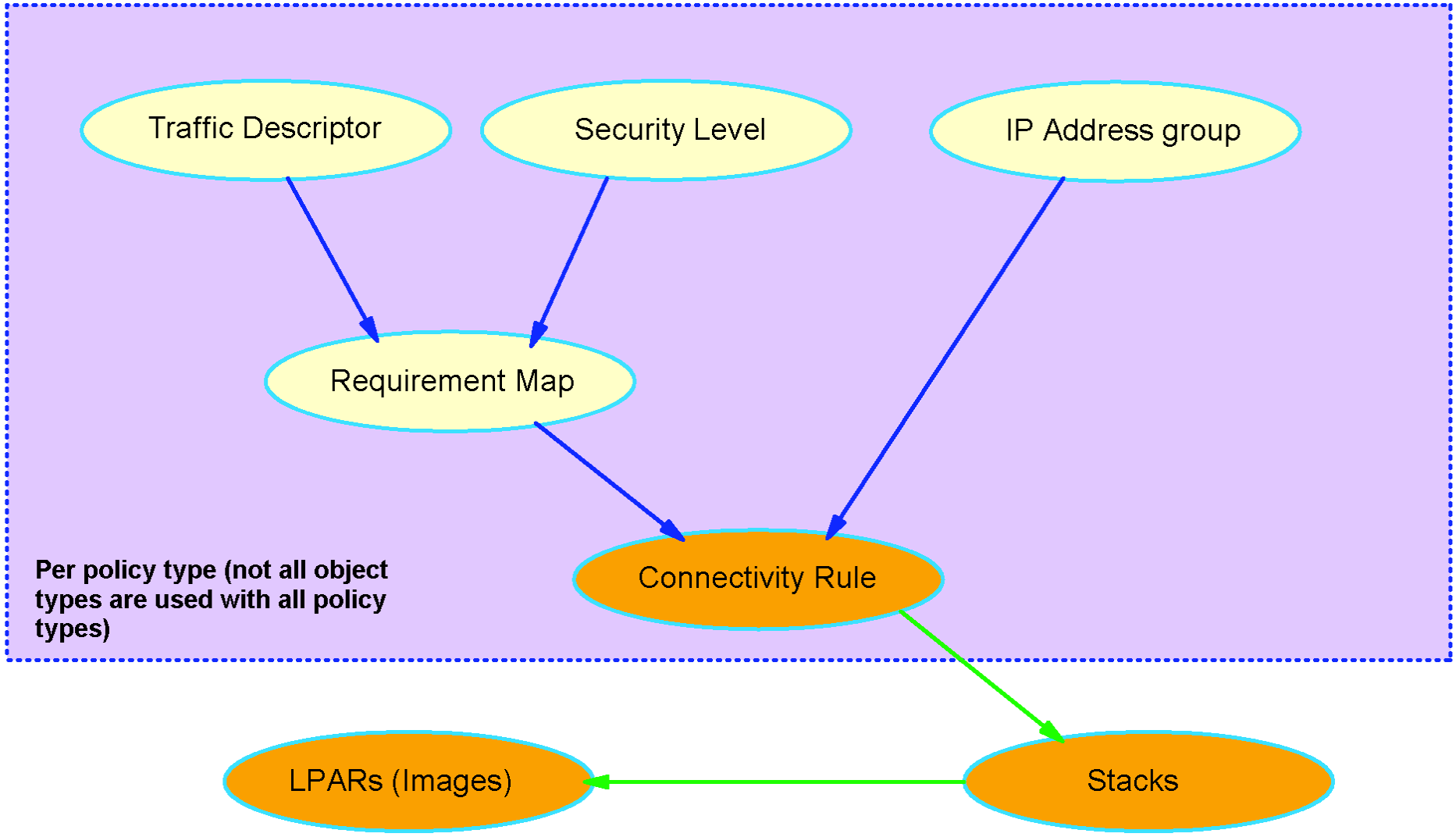




- **Both active and passive mode FTP are now supported.**
- **A text log file of all FTP commands and replies is maintained and can be consulted for details of FTP activity**



Quick Guide to working with the Configuration Assistant objects - ATTLS example



Quick Guide to working with the Configuration Assistant objects - ATTLS example

Traffic Descriptor:

- Identifies a specific type of application network traffic
- Based on protocol (TCP/UDP), local and/or remote ports, connection direction, z/OS jobname, etc.
- A traffic descriptor does not refer to IP addresses
- IBM provides a long list of traffic descriptors for different types of network traffic
- Reuseable across LPARs and stacks in the same backing store file
  - But not reuseable across policy technologies
  - Each policy technology has unique attributes per traffic descriptor

Protocol	Local Port	Remote Port	Connect Direction	Job Name	User ID	AT-TLS Configuration Index
TCP	4021	1024-65535	Inbound	---	---	0
TCP	4020	1024-65535	Outbound	---	---	1
TCP	50000-50200	1024-65535	Inbound	---	---	2

Configuration Associated with this AT-TLS Application

AT-TLS Configuration Index	Handshake Role	Key Ring	Certificate Label	Application Controlled	Secondary Map	Handshake Timeout	Unique SSL Environment	Syspl Cachi
0	Server	Use default	---	On	On	10 Seconds	No	On
1	Server	Use default	---	Off	Off	10 Seconds	No	On
2	Server	Use default	---	Off	Off	10 Seconds	No	On

Security Level:

- Identifies the SSL/TLS security requirements, such as ciphersuites, allowed protocol versions (SSLv2, SSLv3, TLSv1), etc.
- Reuseable across LPARs and stacks in the same backing store file
  - But not reuseable across policy technologies

Type:
AT-TLS
Encryption:
0x2F - TLS_RSA_WITH_AES_128_CBC_SHA (first choice)
Use TLS Version 1:
Yes
Use SSL Version 3:
Yes
Use SSL Version 2:
No
Client authentication:
None

Quick Guide to working with the Configuration Assistant objects - ATTLS example

Requirement Map

- Identifies what type of processing you want applied to your traffic descriptors
- Specific requirements are policy-type dependent
  - For ATT-TLS policies, you define security levels and then you use a requirement map to tie your traffic descriptors to those security levels
  - Reuseable across LPARs and stacks in the same backing store file
    - But not reuseable across policy technologies

Requirement Map: ABC_ATTLS - ATTLS for TN3270 (port 2025) and FTP (port 4021)

Traffic Descriptor	AT-TLS Security Level
ABC_FTP_4021 - FTP Server on port 4021	ABC_Gold_AES - Modified Gold w. AES-128
ABC_TN3270_2025 - TN3270 server on port 2025	ABC_Gold_AES - Modified Gold w. AES-128

IP Address Groups

- Group IP addresses that need the same treatment
  - For example all VIPA addresses, or all real network interface addresses
- Simplifies creation of connectivity rules
- Reuseable across LPARs and stacks in the same backing store file
  - But not reuseable across policy technologies

Address Group: ABC_TCPCS_LAN - LAN network interfaces on TCPCS in LPAR mvs098

Address
9.42.103.11
9.42.105.45

IP Address groups are part of the z/OS V1R10 Configuration Assistant

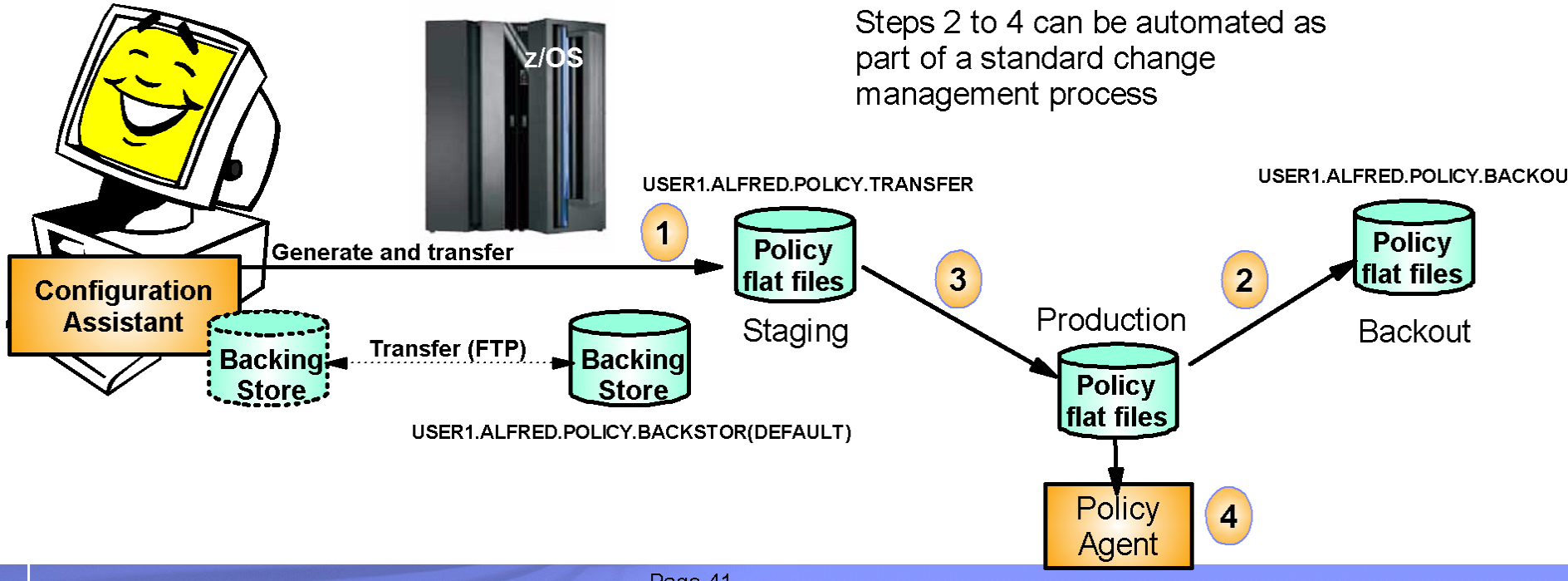
Connectivity Rule

- Here is where IP addresses come into the picture
- Connectivity rules are stack-specific and ties IP addresses to requirement maps
  - And by that, type of processing to traffic descriptors
- Either individual IP addresses or groups of IP addresses

Name	Local Data Endpoint	Remote Data Endpoint	Requirement Map	Status
Conn_to_QDIO4	ABC_TCPCS_LAN	All_IPv4_Addresses	ABC_ATTLS - ATTLS for TN3270 (port 2025) and FTP (port 4021)	Enabled

One suggested approach for updating policies

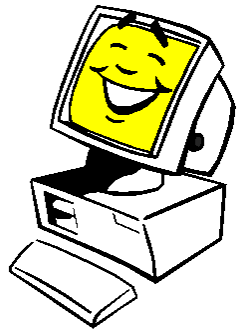
1. **Transfer policy flat file to transfer (staging) library on z/OS**
   - USER1.ALFRED.POLICY.TRANSFER(ATTLS)
2. **Backup current production policy flat file**
   - Copy USER1.TCPCS.TCPPARMS(ATTLS) to USER1.ALFRED.POLICY.BACKOUT(ATTLS)
3. **Copy new policy flat file into production**
   - Copy USER1.ALFREDS.POLICY.TRANSFER(ATTLS) to USER1.TCPCS.TCPPARMS(ATTLS)
4. **Request Policy Agent to refresh the policies**
   - F PAGENT,REFRESH



Should I use the Configuration Assistant or ISPF ???

- **All policies can be configured using ISPF**
  - Full syntax documentation in the IP Configuration Reference manual
  - About 130 pages of detailed syntax
  - Some configuration capabilities currently only exist when doing manual edits
    - IP address group support as an example

- **Using the Configuration Assistant is very helpful when learning about a new policy type**
  - Guided panel and workflow sequences
  - Wizards
  - Syntax checking
  - Lots of help information
  - Health-checking to verify 'sanity' of definitions before uploading to z/OS

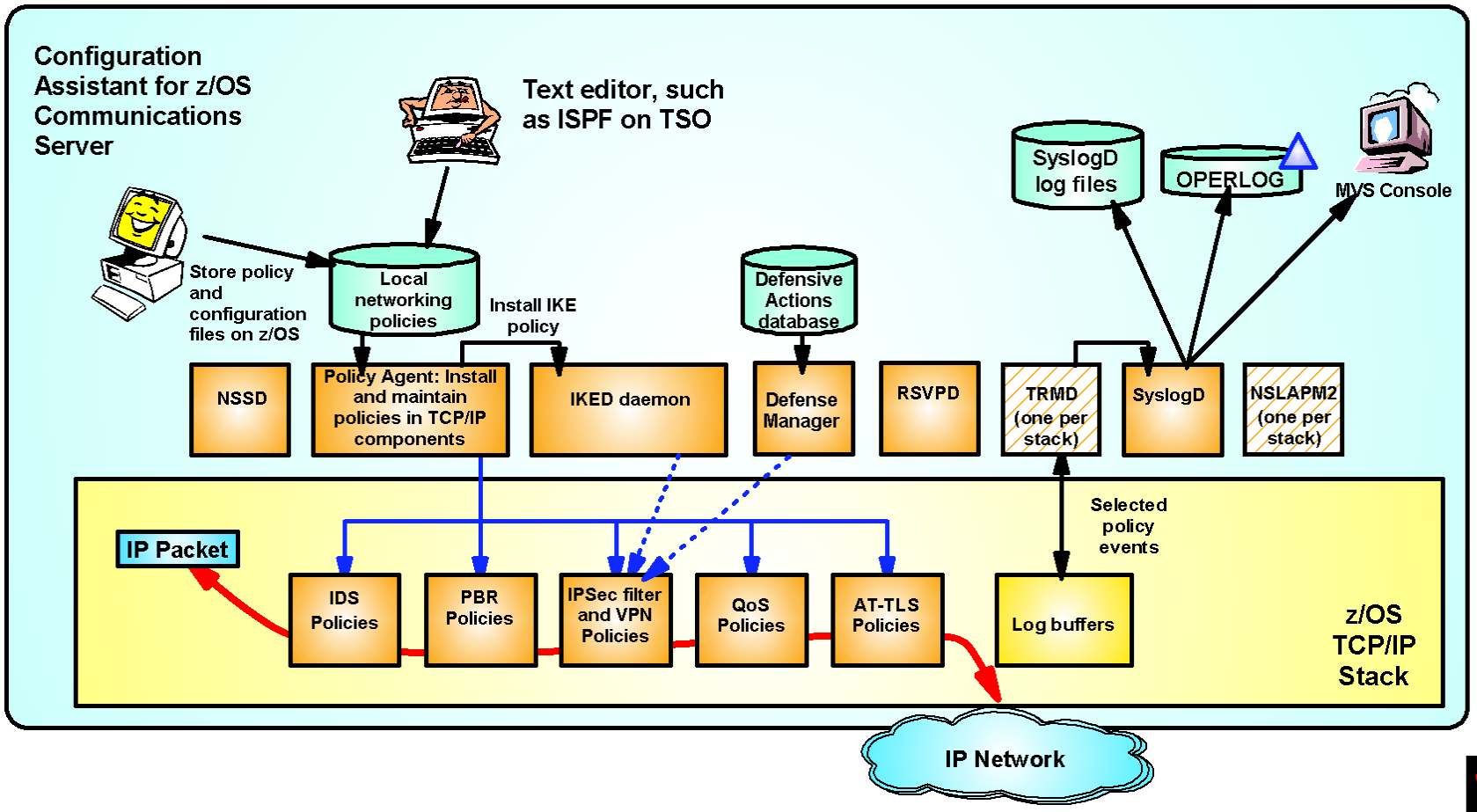


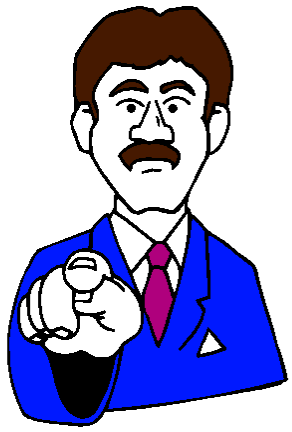
- **With z/OS V1R9 enhancements to store the backing store file on z/OS and the ability to protect against concurrent updates, some of the concerns related to the backing store file have been addressed**

- **When first having learned how to use the Configuration Assistant, it will speed up your work with z/OS CS networking policies**
  - A simple AT-TLS configuration for FTP and TN3270 was relatively easily created using the Configuration Assistant - in about 20 minutes for the first stack
  - The policy flat file that was generated by the Configuration Assistant was 182 lines long

- **Maybe a compromise**
  - Learn with the Configuration Assistant and create your initial policy definitions with it
  - From then on, if changes are merely a question of adding new connectivity rules or making minor changes to existing definitions, you may choose to use ISPF for those changes
    - But do keep in mind that your backing store file in that case will get out-of-sync with your policy flat files

z/OS CS network policy infrastructure is the basis for many valuable functions in your z/OS CS TCP/IP environment





Maybe it is time to go and set it up!!

- IP Filtering to block unwanted traffic from entering z/OS
- Connection-level security for TCP applications without application changes
- Making sure high-priority application also get high-priority processing by the network
- Application-specific selection of outbound interface and route
- Providing secure end-to-end IPSec VPN tunnels on z/OS
- Protection against "bad guys" trying to attack your z/OS system

For more information....

URL	Content
http://www.ibm.com/systems/z/	IBM Mainframe
http://www.ibm.com/systems/z/hardware/networking/index.html	IBM Mainframe Networking
http://www.ibm.com/software/network/commserver/	Communications Server product overview
http://www.ibm.com/software/network/commserver/zos/	z/OS Communications Server overview
http://www.ibm.com/software/network/commserver/z_lin/	Communications Server for Linux on system z
http://www.ibm.com/software/network/ccl/	Communication Controller for Linux on system z
http://www.ibm.com/software/network/commserver/library/	Communications Server products - white papers, product documentation, etc.
http://www.ibm.com/systems/z/os/zos/bkserv/	z/OS Internet library - PDF versions of z/OS manuals (including z/OS CS)
http://www.redbooks.ibm.com	ITSO Redbooks
http://www.ibm.com/software/network/commserver/support	Communications Server technical Support
http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs	Technical support documentation from ATS (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)
http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp	IBM education assistant