There are so many enhancements around the networking policy infrastructure in z/OS V1R11 that we decided an update on how to manage that infrastructure was needed.

This session will discuss not just Policy Agent, but the full policy infrastructure and highlight the V1R11 enhancements as they apply to the management of that infrastructure.

Trademarks, notices, and disclaimers

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

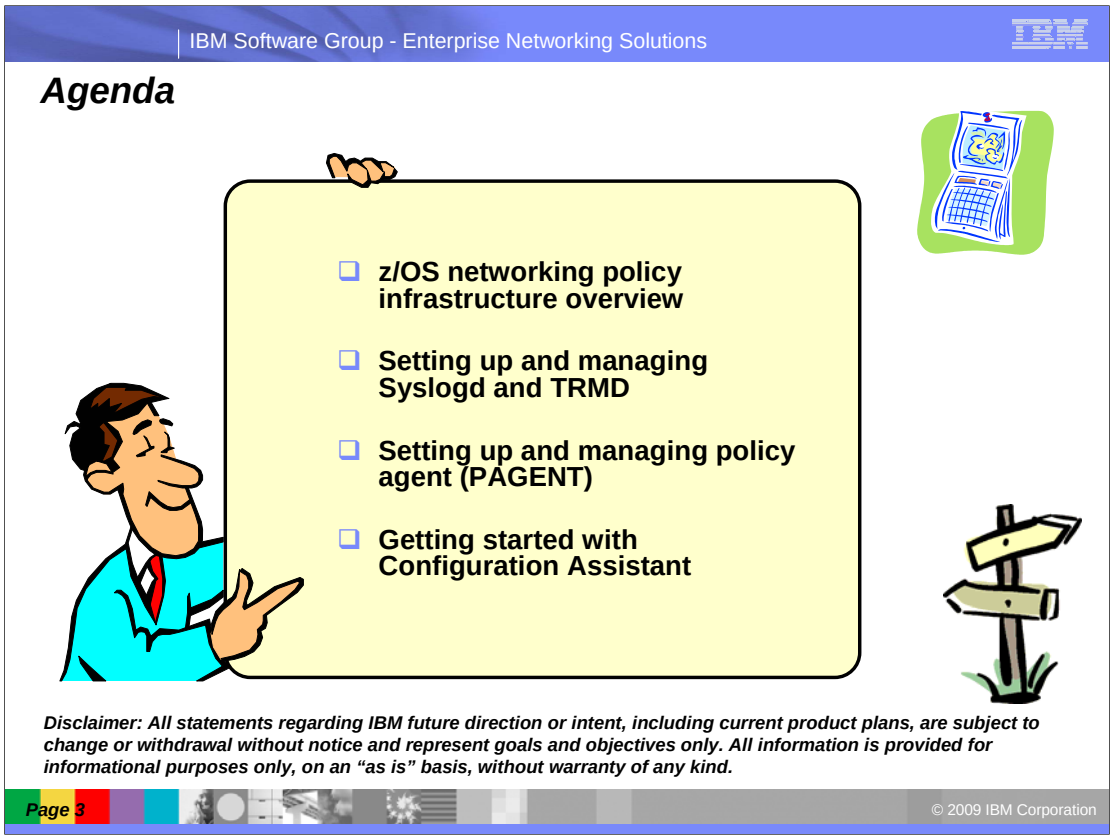
- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- 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.
- Disclaimer: All statements regarding IBM future direction or intent, including current product plans, are subject to change or withdrawal without notice and represent goals and objectives only. All information is provided for informational purposes only, on an "as is" basis, without warranty of any kind.

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.

Legal page.



The z/OS networking policy infrastructure should be somewhat well-known to most of you. Many of you are using ATTLS and some have started using IPSec and IDS.

We will start with a brief overview just to position and focus on what this session is about.

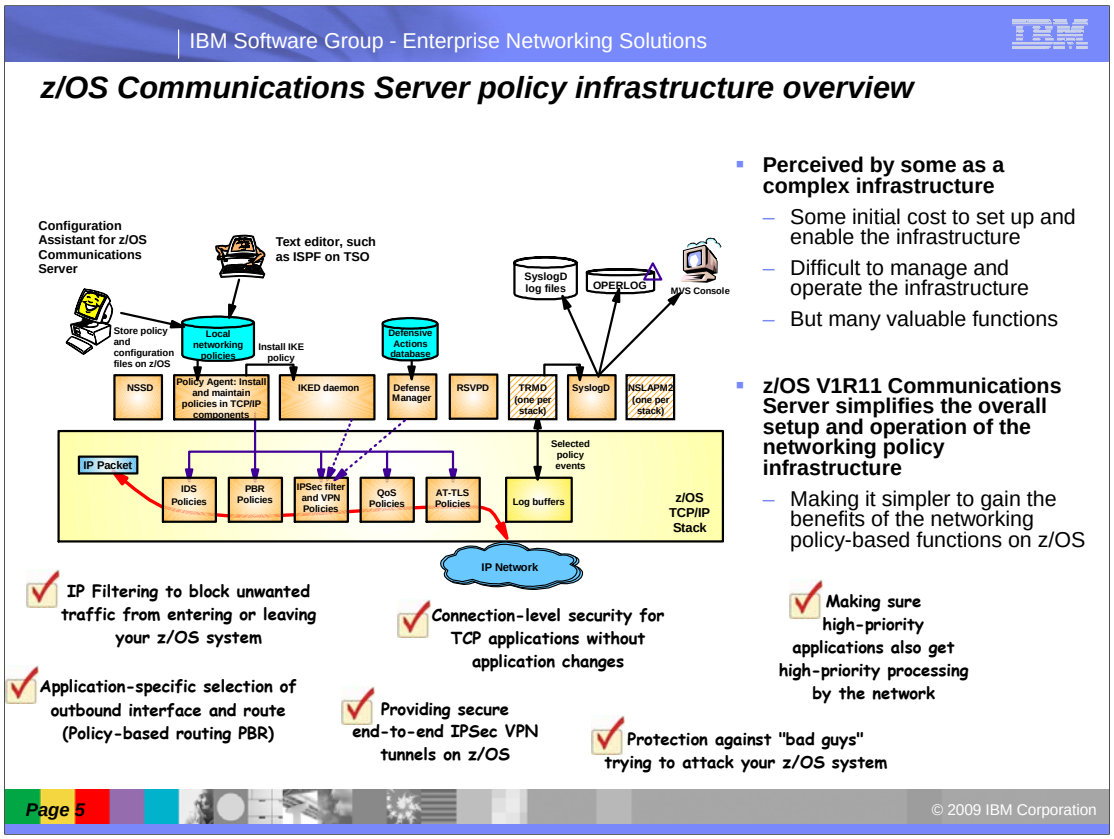
We will then look at how to ensure the auditing trail from all the policy functions: syslogd and TRMD.

The main component is obviously Policy Agent.

And finally, we will look at how the policies can be defined using the Configuration Assistant.

Configuring, operating, and monitoring Policy Agent

z/OS networking policy infrastructure overview



This slide is to refresh everyone's memory of what the z/OS Communications Server networking policy infrastructure is and what it supports.

The infrastructure consists of many components that together deliver support for a range of policy-based networking functions on z/OS. IP filtering, IP Security, Application Transparent SSL/TLS, network quality of service, Intrusion detection, and policy-based routing.

None of these functions are available unless the full or parts of the networking policy infrastructure has been customized and set up.

The full networking policy infrastructure consists of many functions implemented in a range of address spaces. Most of these are started in a single instance per LPAR and will serve one or up to 8 TCP/IP stacks on that LPA. TRMD and NSLAPM2 are stack-specific address spaces and must be started in one instance per stack on the LPAR. Of these two, only TRMD is required.

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

Policy Type	Shared by all stacks on the LPAR					Stack1		Stackn	
	PAGENT	NSSD (1)	IKED	RSVPD	SYSLOGD	TRMDA	SLAPA	TRMDn	SLAPn
QoS	Required			Optional			Optional		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								

Note 1: NSSD is really shared by all stacks in all LPARs in the NSSD domain (which could be a Sysplex or span multiple Sysplex environment)

To keep things a little simple: the basic infrastructure that is needed on any LPAR that implement networking policies is: PAGENT, TRMDx, and Syslogd. This is not fully true, but for all practical purposes, it will work.

I know of no one who uses RSVPD with QoS on z/OS – the version of RSVP that z/OS supports is old and likely not able to interoperate with many other platforms today. Some use NSLAPM2, but again – very few.

IKED is required for dynamic VPN tunnels – it is IKED that talks to an IKED on the other end point to negotiate the parameters for a security association.

NSSD is an optional element together with IKED for storing IPSec keys and certificates on a single z/OS system. NSSD can also be used with WebSphere DataPower for remote SAF access and access to centrally stored keys and certificates (new in z/OS V1R11). NSSD is really not needed on all LPARs – just a single LPAR in the Sysplex is needed for NSSD functions.

Configuration files and policy definition files - overview

Configuration and policy definitions	Manual edit (ISPF)	Configu-ration Assistant	Configu-ration Assistant in z/OS V1R11
Configuration files			
Policy Agent configuration	Yes	No	Yes
Syslogd configuration	Yes	No	(partly)
IKED configuration	Yes	Yes	Yes
NSSD configuration	Yes	Yes	Yes
RSVPD configuration	Yes	No	No
DMD configuration	Yes	No	Yes
Policy definition files			
QoS policy	Yes	Yes	Yes
IDS policy	Yes	Yes	Yes
ATTLS policy	Yes	Yes	Yes
IPSec policy	Yes	Yes	Yes
PBR policy	Yes	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 during policy activation

The policy components use both configuration files per component, and policy definitions files per policy type.

All can obviously be edited with ISPF. Most can also be created by the Configuration Assistant, which in R11 picks up support for the Policy Agent configuration and the DMD configuration. It does not create the full syslogd configuration, but will suggest snippets of syslogd configuration based on which policies are defined.

All the above files can be either z/OS UNIX files or MVS data sets, including members of PDS(E) libraries.

I (personally) prefer MVS PDS(E) members for the following reasons:

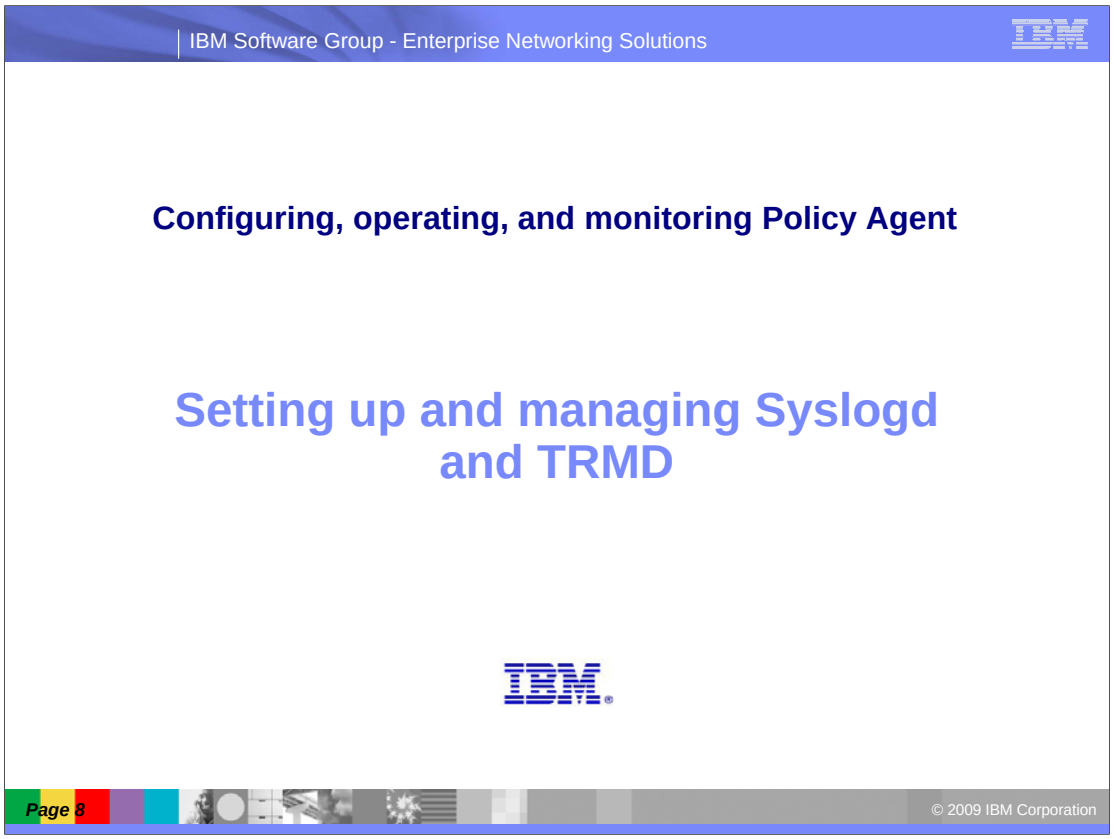
I already have an MVS PDS(E) library structure in place for LPARs and TCP/IP stack configurations, such as PROFILE, OMPROUTE, etc.

There typically already is a backup/restore mechanism in place for these configuration data sets

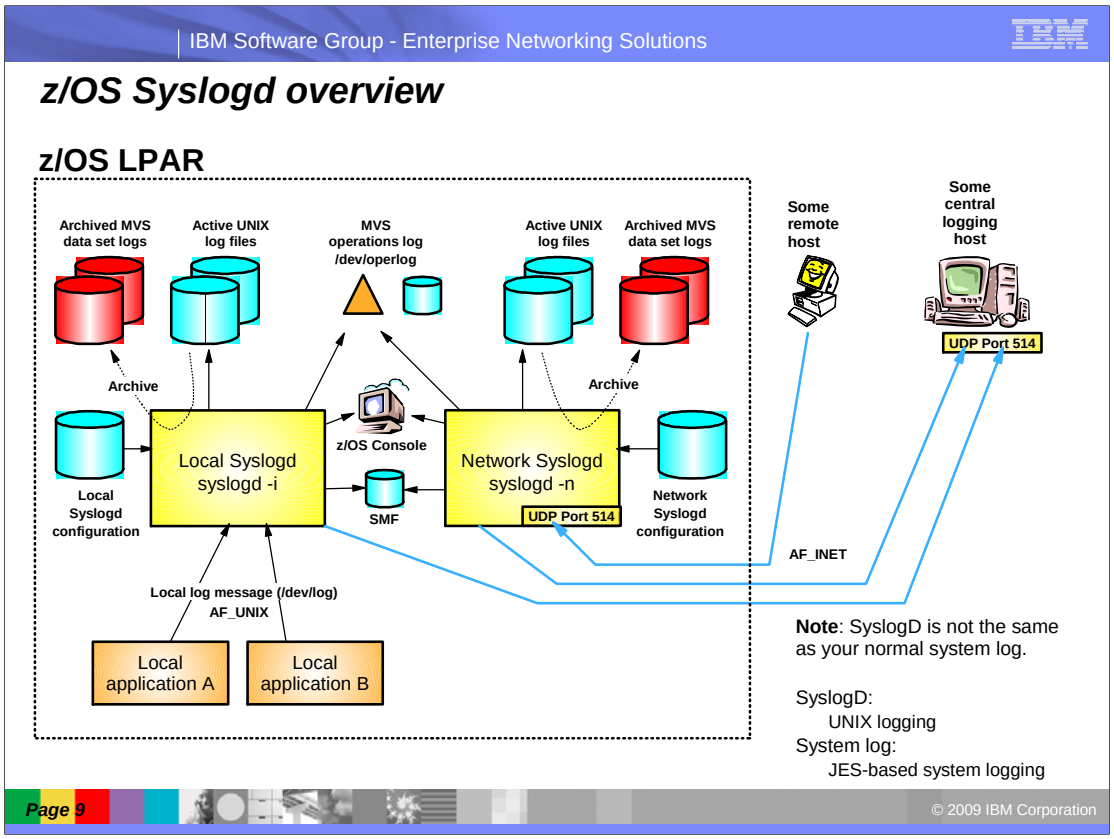
Existing change management procedures are often based on PDS(E) library structures with staging, production, and backout libraries

Standard RACF profiles control who can access them in what way

But, z/OS UNIX files will work as well.



These are the main components that are needed in support of an audit trail. Remember, many of the policies are related to various forms of security and auditors may have requirements to what, how, and for long you capture log messages related to such functions.

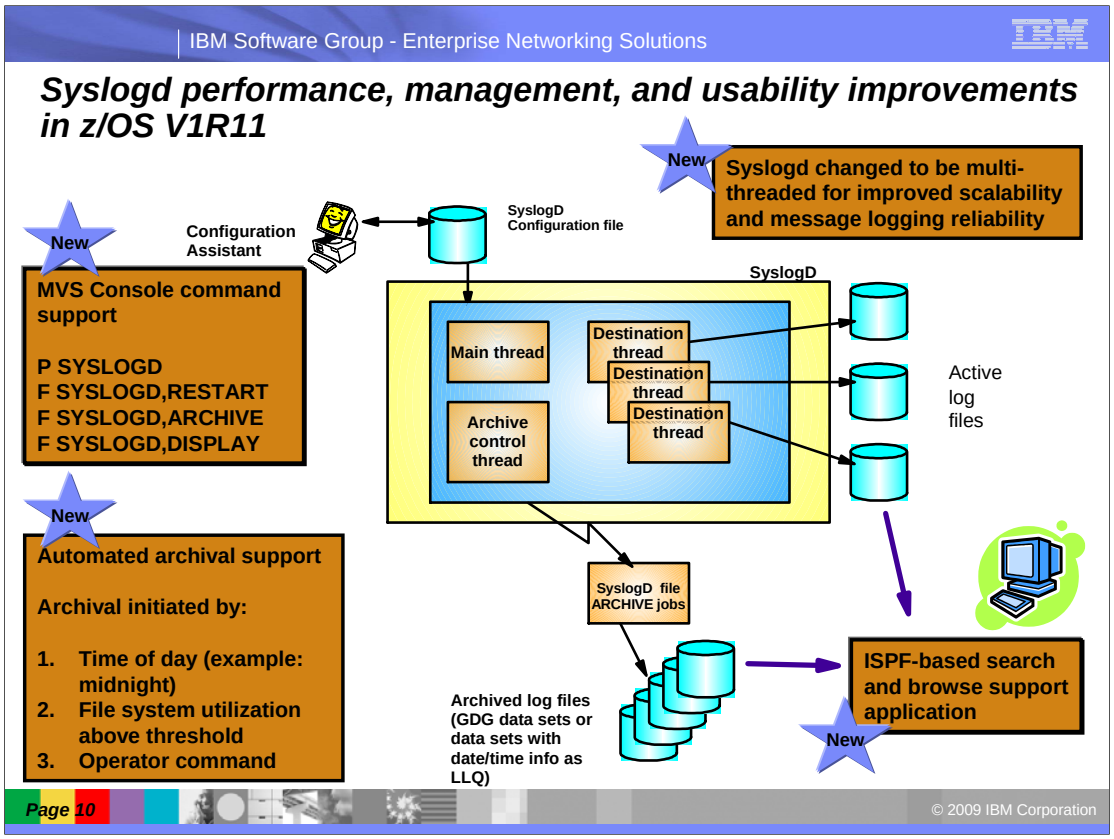


z/OS supports running two Syslogd instances:

One that is used by local applications on the z/OS system where the Syslogd instance is running. You start Syslogd with a –I flag to indicate it is a local instance. Such a Syslogd instance will not open UDP port 514 and will not be subject to remote attacks via that UDP port. Such a local Syslogd instance can be set up to send log messages to other Syslogd instances, but it cannot receive any such messages.

Another that is used by remote Syslogd instances that are configured to send their log messages to a focal consolidated Syslogd on z/OS. You start Syslogd with a –n flag to indicate it is a network Syslogd instance. Such a Syslogd instance will open UDP port 514 and received messages from remote Syslogd instances. It will not receive messages from any local applications.

Also remember that JES SYSLOG is not the same as UNIX SYSLOGD !!!!



This slide shows a high-level overview of the new and improved Syslogd components.

Syslogd is now a multi-threaded implementation allowing for more parallel processing in peak periods. Syslogd continues to write log messages to z/OS UNIX files. A new archive function will archive the content of a z/OS UNIX log file to an MVS data set. The MVS data set can either be a sequential data set (low level qualifiers specify date and time) or a new generation of a generation data group (GDG). The archive operation can be initiated by an operator. At a specific point in time (for example, shortly after midnight). Or when the utilization of one of the file systems the z/OS UNIX log files are written to exceeds a configurable threshold.

Command support includes the ability to shut Syslogd down using a P command. Syslogd will in R11 not change address space name after it has started. If you start a procedure by the name of SYSLOGD – the resulting address space name remains SYSLOGD.

If you start Syslogd via UNIX shell commands (such as from /etc/rc) then you must add an ampersand to the end: Syslogd &

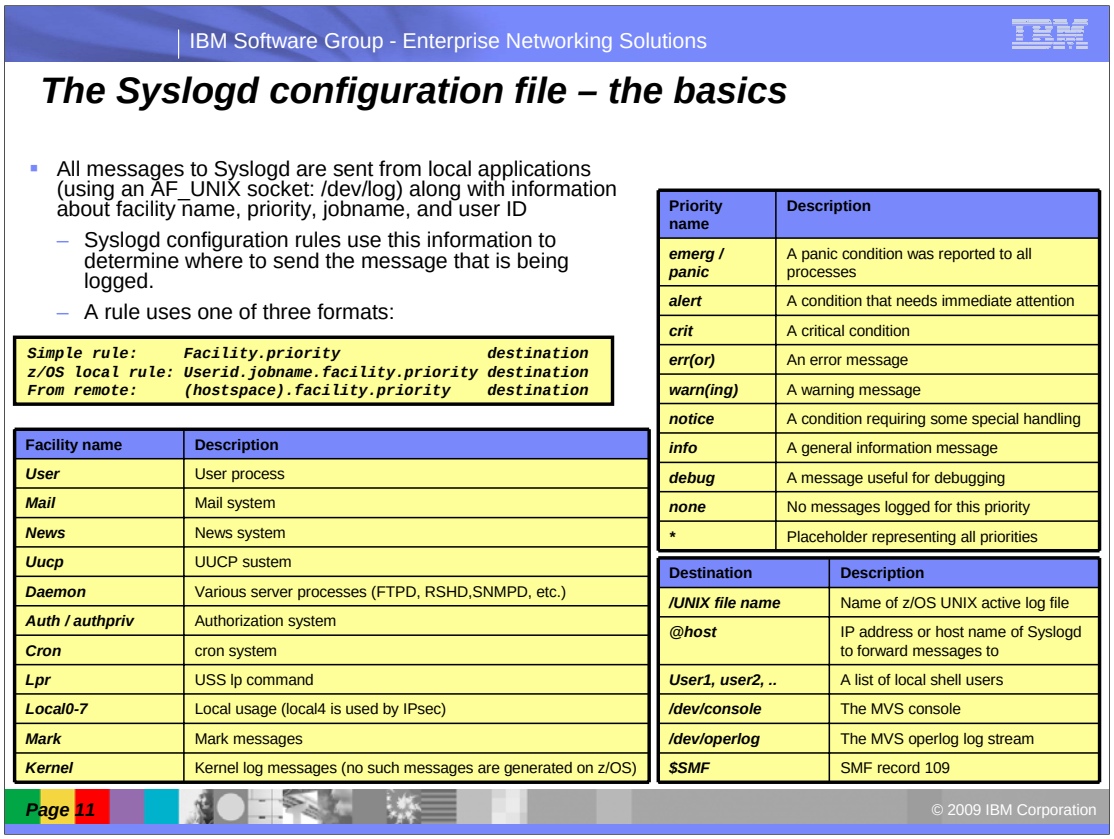
The ISPF browser starts by reading the Syslogd configuration file, locates the active z/OS UNIX files, and all available MVS archives. It supports browsing individual files or data sets, in addition to performing extensive searches in one or a series of files or data sets.

The solution adds a fully automatic archival mechanism to Syslogd, that also supports on demand archival if needed. You can archive once per day using a configurable time of day, or archive when any of the UNIX file systems reaches a configurable percentage full. You can also archive using an operator command. Syslogd archives UNIX files to either sequential or generation data group (GDG) data sets, and you can include system symbols in parts of the target data set names. You do not need to determine the space requirements of the target data sets - Syslogd takes care of that. Syslogd automatically retries previously failed archives at the next archive event. You can monitor a console message for failed archives to correct any problems, and Syslogd will eventually successfully archive all previously failed files. You can also use a new operator command to display the utilization of the Syslogd UNIX file systems.

The default for all UNIX files is not to perform an automatic archival. If you want to use this new function you must explicitly configure it.

You have three choices for each rule that contains a UNIX file log destination. You can archive the file by using the new -N parameter. You can reinitialize the file (delete its contents) when an archive occurs. Use this option with care, because the contents of the file are lost. Or you can do nothing with the file by not using either the -N or -X parameters.

All eligible files are archived for the time of day and operator command triggers. But for the file system threshold trigger, Syslogd attempts to reduce the space used by archiving files until ½ of the configured threshold is reached. For example, if you configure 80% as the threshold, Syslogd archives files until the file system reaches 40% utilization. A console message is issued if all eligible files are archived but the file system ulilization was not able to be reduced to ½ the configuredthreshold. This can happen if the file system contains files that are not managed by Syslogd.



Most of you should be aware of this by now.

Every message that is logged by syslogd on z/OS is associated with a userID, a jobname, a facility name, and a priority code. The last two are assigned by the application that sends the log message to syslogd. The first two are added transparently by the logging API function.

The syslogd configuration may consists of rules that are made up of a condition and a destination:

Simple rules - what is used on all UNIX platforms. Only uses the facility and priority to select the message

z/OS local rule – rule that can be used for locally logged z/OS UNIX messages. Takes job name, user ID, facility, and priority into consideration. Generic job names and user ID syntax is supported.

Remote rule – a rule that can be used to log messages received from remote systems over syslogd's UDP socket. Takes IP address or host name of the remote system into consideration.

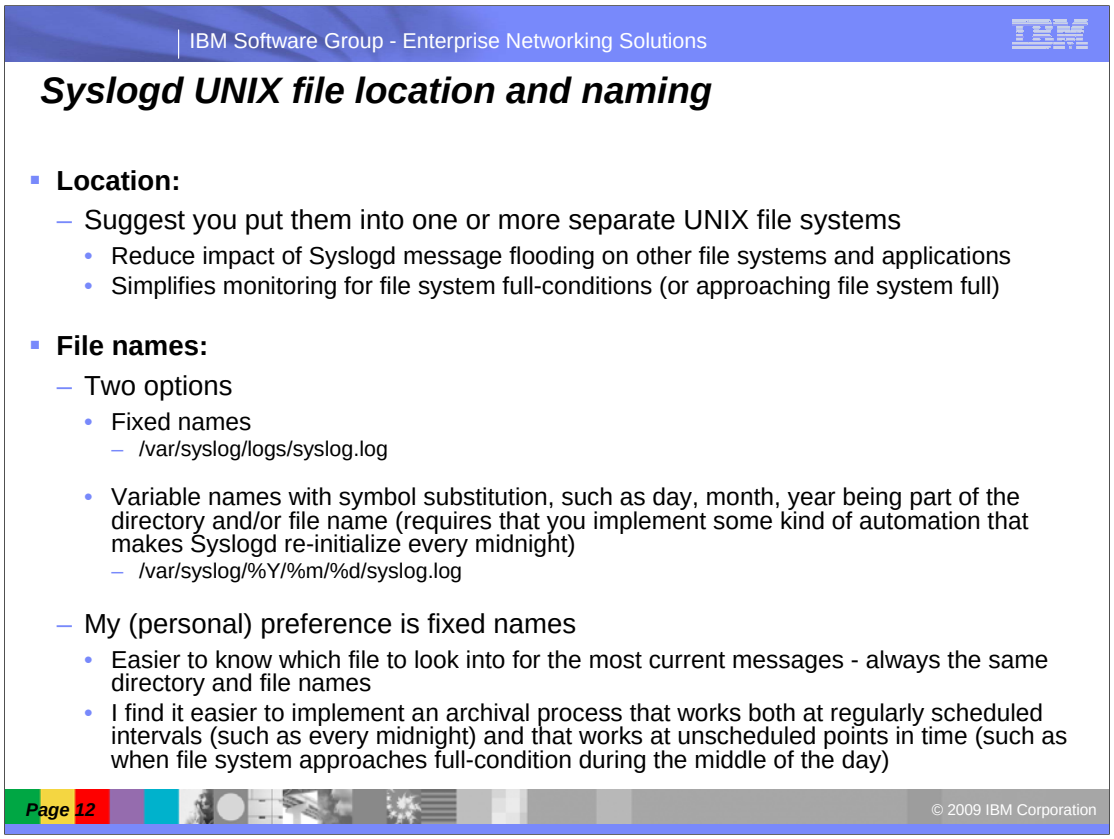
Facility names and priority codes are 'industry standards' – common on all UNIX-flavored platforms.

A rule that refers to a priority will match for messages that have that priority or a higher priority. So crit will include alert and emerg or panic also.

Log messages are most often sent to a UNIX file destination, but syslogd on z/OS does support alternate destinations, such as the MVS console, or the log stream known as operlog. Operlog may be local or Sysplex-wide.

Use of SMF should be considered carefully – syslogd could generate large volumes of SMF records.

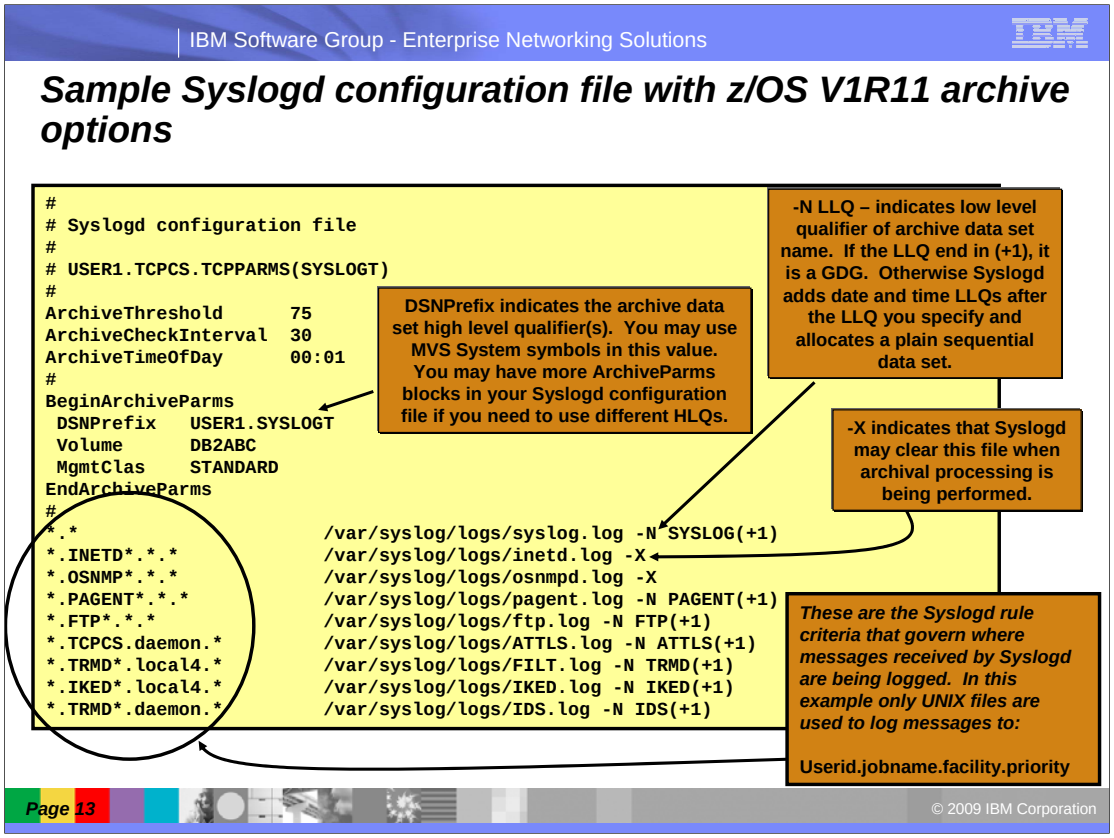
A message may meet multiple conditions and if it does, it will be logged to multiple destinations.



There are two considerations for the UNIX files, syslogd writes to.

They should reside in a file system that isn't used for other purposes. If syslogd gets into a message flooding event, it can fill up the file system. If that file system is also used for other use, that other use could be impacted by such an event. It is also much simpler to automate offload of these UNIX files when they are in a separate file system.

The names of the files. Fixed names work very well with the archival mechanism in z/OS V1R11. Some may have implemented CRON-based archival and signaling already, and they may prefer file names that change every midnight – by using the so-called %-sign syntax, where the %-symbols are substituted by syslogd during file creation with current values. Hence the need for CRON to send syslogd a sighup signal just after midnight.



The ArchiveTimeOfDay statement configures the time of day for an automatic archival, using hours and minutes in a 24 hour clock format. For example, specify 00:01 to mean one minute past midnight. If you do not want to archive at a specific time of day, then do not configure this statement.

The ArchiveCheckInterval statement configures the interval in minutes for checking the utilization of UNIX file systems. The default is 10 minutes. This statement is only used if you configure a non-zero percentage on the ArchiveThreshold statement.

The ArchiveThreshold statement configures the percentage of UNIX file system utilization that triggers an archive. The utilization is checked at the interval specified with the ArchiveCheckInterval statement . You can specify any value between 0 and 99, but you should avoid very low or very high values. A value of 0 means that Syslogd should not perform threshold based archival. The default value is 70.

The BeginArchiveParms statement configures the data set name prefix for the target data set. You must configure a data set name prefix before using the -N parameter on any Syslogd rules. You can repeat the BeginArchiveParms statement multiple times for different groups of Syslogd rules, or you can use a single instance of the statement to apply to all rules.

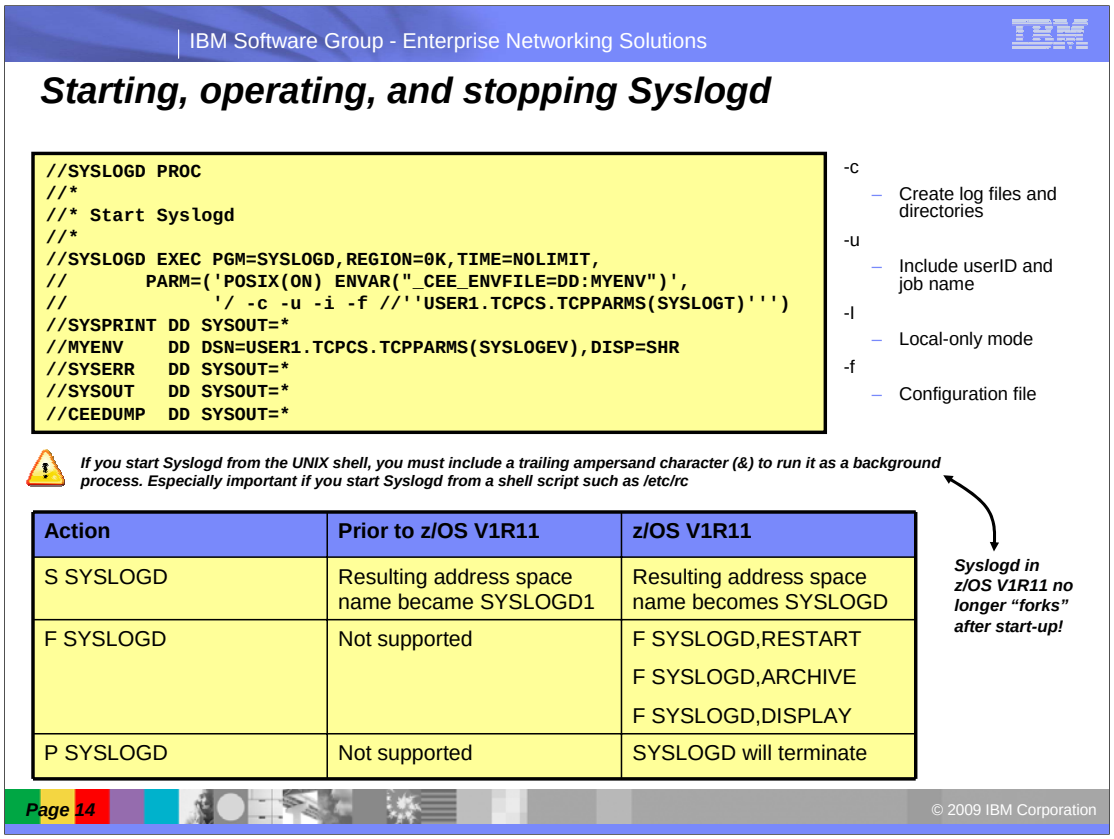
Besides the data set name prefix, you can configure several allocation parameters for the archive data set. These parameters have the same names, syntax, and meaning as the corresponding parameters on the DD JCL statement.

Use the -N parameter on a Syslogd rule to indicate that the rule is eligible for automatic archival. Specify a data set name qualifier with the -N parameter. You must precede the rule with a valid BeginArchiveParms statement that specifies the data set name prefix.

Use the -X parameter on a Syslogd rule to indicate that the contents of the file should be deleted when an archive event occurs. You should only use this parameter if you do not need to keep the contents of the file.

If you do not use the -N or -X parameter on a Syslogd rule that specifies a UNIX file destination, then the file does not participate in automatic archival processing.

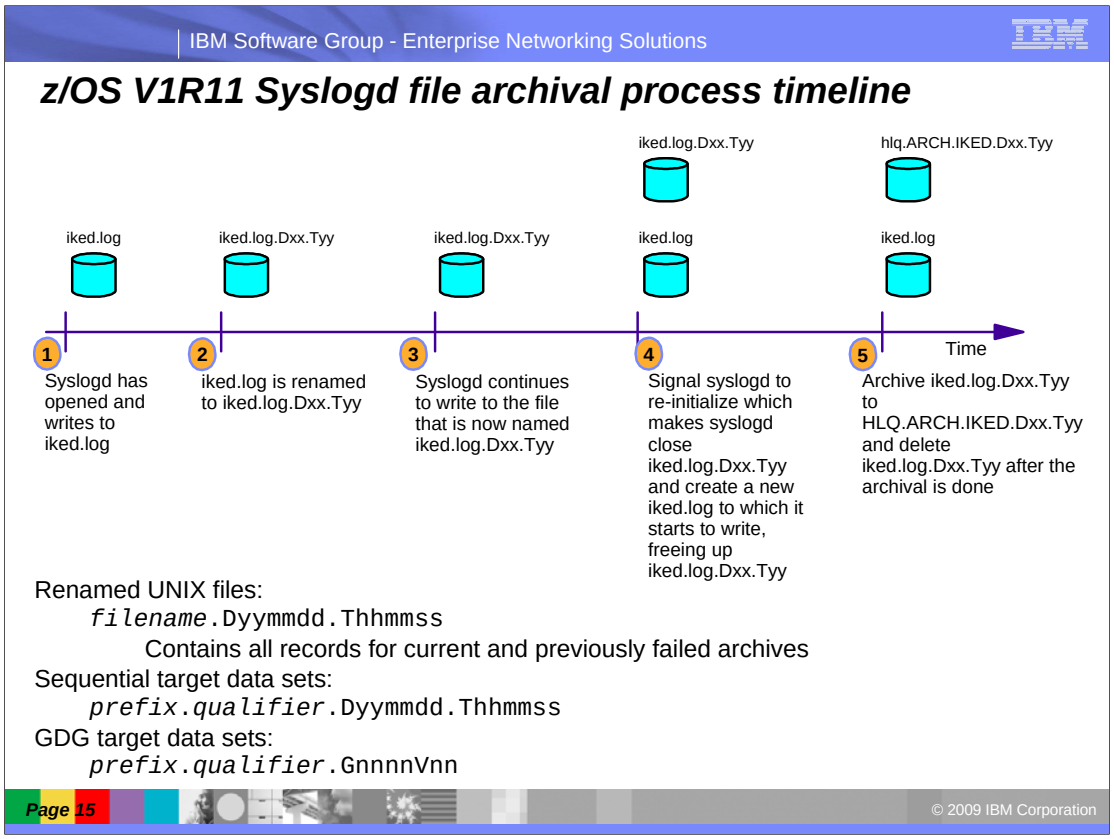
You cannot use the -N or -X parameter with the existing -D or -F parameters on a Syslogd rule. You also cannot use -N and -X together on a Syslogd rule.



The sample SYSLOGD procedure can be used to start syslogd manually. If the new policy agent monitor function is used, syslogd can be started using the same JCL proc as all the other components.

If you (continue to) start syslogd from a UNIX shell script, such as /etc/rc – you MUST make sure the start command is followed by an ampersand (&) to force syslogd to run in the background. Otherwise that script will never end. Syslogd does not re-fork itself after start in R11. This is a change and can be a migration concern if you start syslogd from a shell script.

If you start syslogd using a JCL proc, it will now retain the name you start, it will support various modify and stop commands.



This diagram shows the timeline of an archive event for a single file. The log file is named iked.log for this example.

At step one, the iked.log file is open and Syslogd is writing records to it.

At step two, Syslogd renames the open log file with a unique date and time suffix.

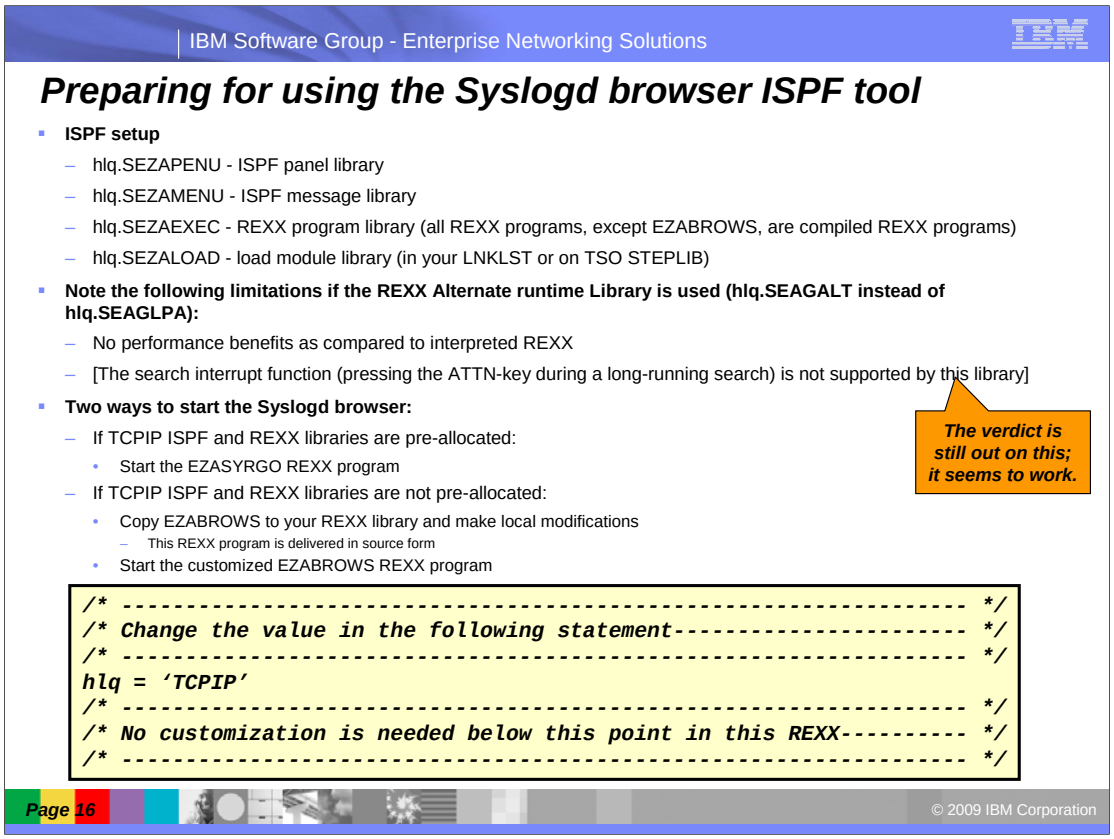
At step three, the file is still open so Syslogd can continue to write records to it.

At step four, the renamed file is closed and the original iked.log file is recreated. Syslogd now writes to the open iked.log file.

At step five, the renamed file has been archived into a target data set, and the renamed file has been deleted. The archive process consists of this sequence of events. Syslogd allocates the target data set, opens the source file and target data set, copies the file, closes the file and data set, unallocates the data set, and then deletes the renamed source UNIX file.

When Syslogd renames UNIX files to prepare them for archival, it adds a unique suffix that identifies the current date and time. You might see these files in your file system if an automatic archival fails. If successive archive failures occur, this file contains the file contents from all previous failures, and is named with the most recent date and time stamp.

The format of the target data set name depends on whether you are using sequential data sets or GDG data sets. For sequential data sets, Syslogd appends unique date and time values to the configured prefix and qualifier values. For GDG data sets, the system creates a unique suffix value according to how the GDG base data set was defined.



All components of the Syslogd browser have member names that start with EZASYxxx.

z/OS CS delivers ISPF components for panels, messages, and REXX programs. ISPF panels are in hlq.SEZAPENU. ISPF messages are in hlq.SEZAMENU. REXX programs for TSO are in hlq.SEZAEXEC.

You can pre-allocate ISPF and REXX libraries using DD names in your TSO LOGON procedure or TSO LOGON CLIST. hlq.SEZAEXEC is a new z/OS CS system library in z/OS V1R11. It is an FB, 80 library.

If you use the EZABROWS REXX to start the browser, you can copy EZABROWS to a REXX library that is pre-allocated to your TSO environment. You must customize the copied EZABROWS to identify high level qualifier of your z/OS CS ISPF libraries.

EZABROWS uses ISPF LIBDEF commands to add the z/OS CS ISPF Libraries to ISPF (ISPPLIB and ISPMLIB). It uses the TSO ALTLIB command to add the hlq.SEZAEXEC library to TSO. EZABROWS finally starts the Syslogd browser (EZASYRGO) using an ISPF SELECT with NEWPOOL, PASSLIB, and NEWAPPL(EZAS).

“Real” runtime library is REXX.SEAGLPA and normally resides in the LPA.

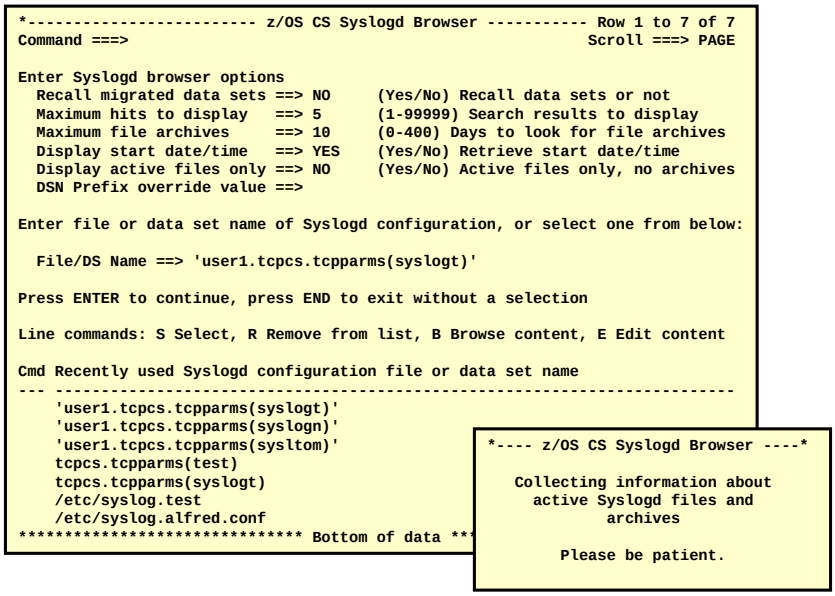
“Alternate” runtime library is REXX.SEAGALT

Syslogd browser entry panel

In z/OS V1R11, a TSO/ISPF interface to browse and search messages captured by Syslogd is also introduced.

The Syslogd browser works with active UNIX files and archived MVS data sets.

The panel shown here is the initial panel when you start the Syslogd browser. This panel is used to set general options and to select the Syslogd configuration file representing the syslog daemon you want to work with.



Syslogd browser options:

Do you want the browser to access MVS data sets that have been migrated? If you specify NO, you are not able to browse migrated archive data sets.

The maximum number of hits you want displayed as the result of a search operation. Can also be set on the search panel.

If you use z/OS UNIX file archives based on a file naming convention that uses %-symbols (for day, month, and year), the Syslogd browser will look for archives in the same directory as where the active z/OS UNIX file resides. The browser will look for such archives day by day. You use this option to specify the maximum number of days you want to look for such archives.

The display start date/time option is used to control the display of start date and time for each active file and each archive. Set this to NO if you don't need it and want to improve the performance of the Syslogd browser initialization.

The Display active files only option controls if the Syslogd browser is to be used for browsing the currently active Syslogd files only, or if it is to be used for browsing both active Syslogd files and archives. Set this to NO if you know you're only going to browse the active Syslogd files. It will improve the performance of the Syslogd browser initialization.

The DSN Prefix override value overrides the DSNPREFIX keyword in your Syslogd configuration file. This option is especially useful if you use system symbols in your DSNPREFIX and want to browse the Syslogd files of another LPAR than the one you are logged into.

The browser will save the last 10 Syslogd configuration files the user has used. For each of those, the user can edit, browse, remove from the list, or select the configuration file for use by the browser.

If you have many Syslogd UNIX files and archived MVS data sets, it will take a little while for the browser to collect information about all those files and data sets. You can speed the initialization up by either answerting NO to display start date and time, or answering YES to display active files only. If you know you are going to look into the active UNIX files only, then there is no need to collect information about archives.

Syslogd destination view

This panel lists all the rules in the specified Syslogd configuration file that writes to UNIX files.

Both primary and line commands are available on this panel to browse, search, etc.

```
*------------------------- z/OS CS Syslogd Browser ---------- Row 1 to 7 of 12
OPTION ===>                                                   Scroll ===> PAGE

  1 Change current Syslogd configuration file and/or options
  2 Guide me to a possible Syslogd destination
  3 Clear guide-me hits (indicated by ==> in the Cmd column)
  4 Search across all active Syslogd files

Current config file ==> 'user1.tcpcs.tcpparms(syslogt)'

Press ENTER to select an entry, press END to exit the Syslogd browser

Line commands: B Browse, A List archives, S Search active file and archives,
               SF Search active file, SA Search archives, I File/DSN info
                                                                  Archive
Cmd Rule/Active UNIX file name                  Start Time        Type Avail.
--- -------------------------------------------- ----------------- ---- ------
    *.*                                         09 Dec 2008 00:00 GDG  3
    /var/syslog/logs/syslog.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.TCPCS*.*.*                                09 Dec 2008 13:47 SEQ  9
    /var/syslog/logs/tcpcs.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.INETD*.*.*                                Empty       N/A   None 0
    /var/syslog/logs/inetd.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.OSNMP*.*.*                                09 Dec 2008 13:47 CLR  0
    /var/syslog/logs/osnmpd.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.PAGENT*.*.*                               09 Dec 2008 00:01 SEQ  13
    /var/syslog/logs/pagent.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.FTP*.*.*                                  08 Dec 2008 15:22 FILE 2
    /var/syslog/logs/ftp.08.12.08.log
    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    *.FTP*.*.*                                  08 Dec 2008 15:22 FILE 2
    /var/syslog/logs/ftp.08.12.2008.log
```


After having parsed a Syslogd configuration file, all z/OS UNIX file destinations are selected and all associated available archives are located. This Syslogd destination view is the main panel of the Syslogd browser interface from which other functions are selected.

Main options:

to change which Syslogd configuration you're using. Can also be used to re-initialize with the current Syslogd configuration file. This can be useful if an archive occurs while you are using the browser. The new archive file or data sets is not accessible until you re-Initialize.

invoke the guide-me function (help me to find which destination my log messages go to)

clear the indicators that were returned from an invocation of the guide-me function

start a search operation across all the active Syslogd destination files (such a search can take some time if the active files are large)

Scrollable section:

The display includes one entry per z/OS UNIX file destination for which the active file can be found. Each entry includes rule, active file name, date and time of the first logged message in the active file, archive type, and number of available archives. For MVS archives that means archives that were found online in the z/OS UNIX file system or in a z/OS catalog. For each entry, several line commands are available to browse the active file, search at various levels, and so on.

Archive types:

None - No archive processing for this file

GDG - Archive done to an MVS generation data set group

SEQ - Archive done to a sequential MVS data set

CLR - No archive. The z/OS UNIX file is cleared during archive processing (the -X option used in the Syslogd configuration file)

FILE - z/OS UNIX files based on use of %-symbols

Browse an active Syslogd file

```
 BROWSE    /var/syslog/logs/pagent.log                Line 00000000 Col 001 080
 Command ===>                                           Scroll ===> PAGE
******************************** Top of Data ********************************
00000001 Dec  9 00:01:10 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :006:
         policy_perf_get_sampling_data(): Obtained 2 policy performance data
         entries from the stack
00000002 Dec  9 00:01:10 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :006:
         pqos_refresh_perf_cache: Refreshing cache with 2 performance entries
00000003 Dec  9 00:01:10 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :006:
         pqos_refresh_perf_cache: Refresh complete: #sla=2, #cache=1, #SL=1,
         #cacheSL=1
00000004 Dec  9 00:01:10 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :006:
         policy_perf_send_msg_to_SD(): Sending 1 default fractions to the stack
00000005 Dec  9 00:01:10 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :008:
         pqos_send_frns_to_SD: Sending fractions to the stack, 1 headers, 1
         entries
00000006 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :001:
         check_main_config_file: Main configuration file updated
00000007 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :001:
         check_main_config_file: pagentRefresh = NO
00000008 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :005:
         check_config_files: Thread cleanup completed
00000009 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :007:
         qosListener: Thread cleanup completed
00000010 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  SYSERR :008:
         pqos_recv_msg_from_listener: recv with peek failed, errno EDC8121I
         Connection reset., errno2 76650446
00000011 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  OBJERR :008:
         pqos_get_info_from_listeners: pqos_recv_msg_from_listener failed
00000012 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  LOG    :008:
         pqos_get_info_from_listeners: EZZ8775I PAGENT ON TCPCS CONNECTION NO
         LONGER ACTIVE TO 192.168.5.1..1700
00000013 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :008:
         pqos_get_info_from_listeners: Thread cleanup completed
00000014 Dec  9 00:02:09 MVS098/TCPCS    PAGENT   Pagent[13]:  EVENT  :006:
         policy_perf_monitor: Thread cleanup completed
```

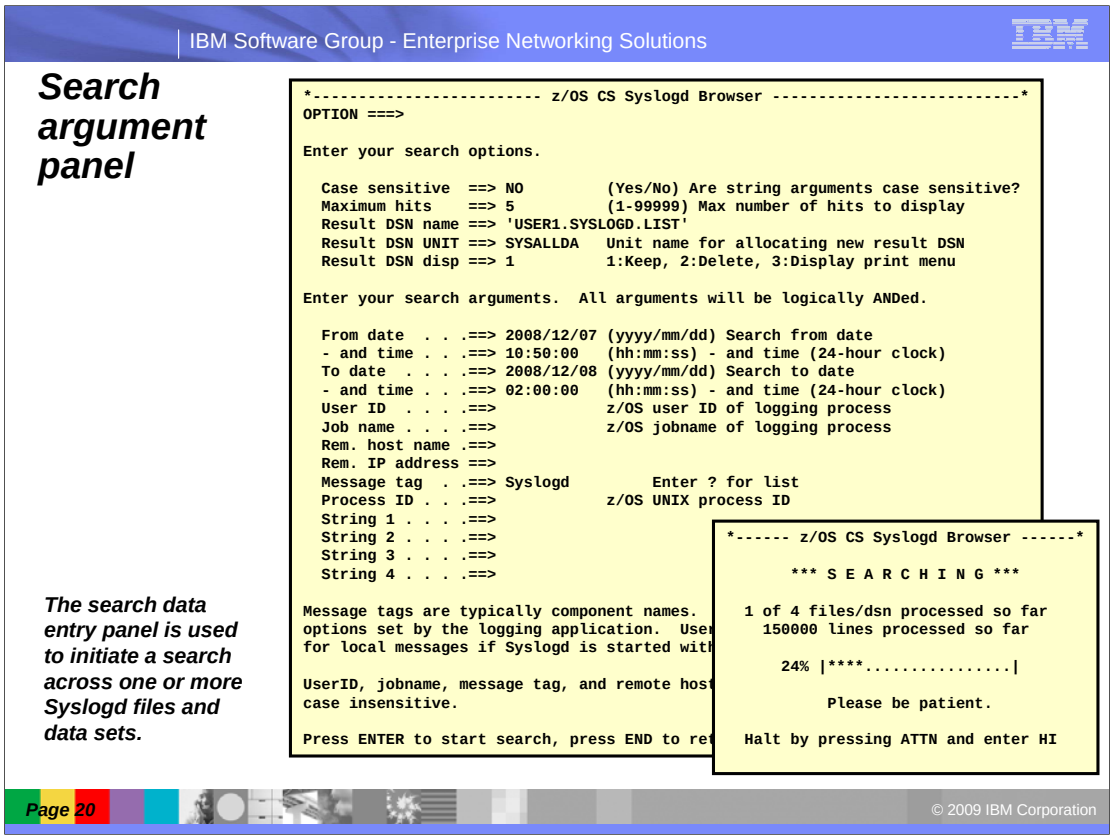
A normal ISPF browser interface.

By entering a ‘B’ for an entry at the destination view, you will see a display of the active UNIX file.

The actual browse window is built using the ISPF BRIF interface, which allows the browser to read only portions of a file or data set into storage at a time.

Long messages are folded into lines that fit the current ISPF screen width.

Normal ISPF FIND command support is available and can be used for simple searches in the file that is being browsed.



By entering one of the S-commands for an entry in the destination view, you will get to the search interface.

Search options:

These options governs the search operation. The result data set name can be an existing data set. If it does not exist, it is allocated using the specified UNIT name (which is initialized to your corresponding ISPF allocation unit). After the search, you can keep the result data set, delete it, or have a standard ISPF print dialog displayed.

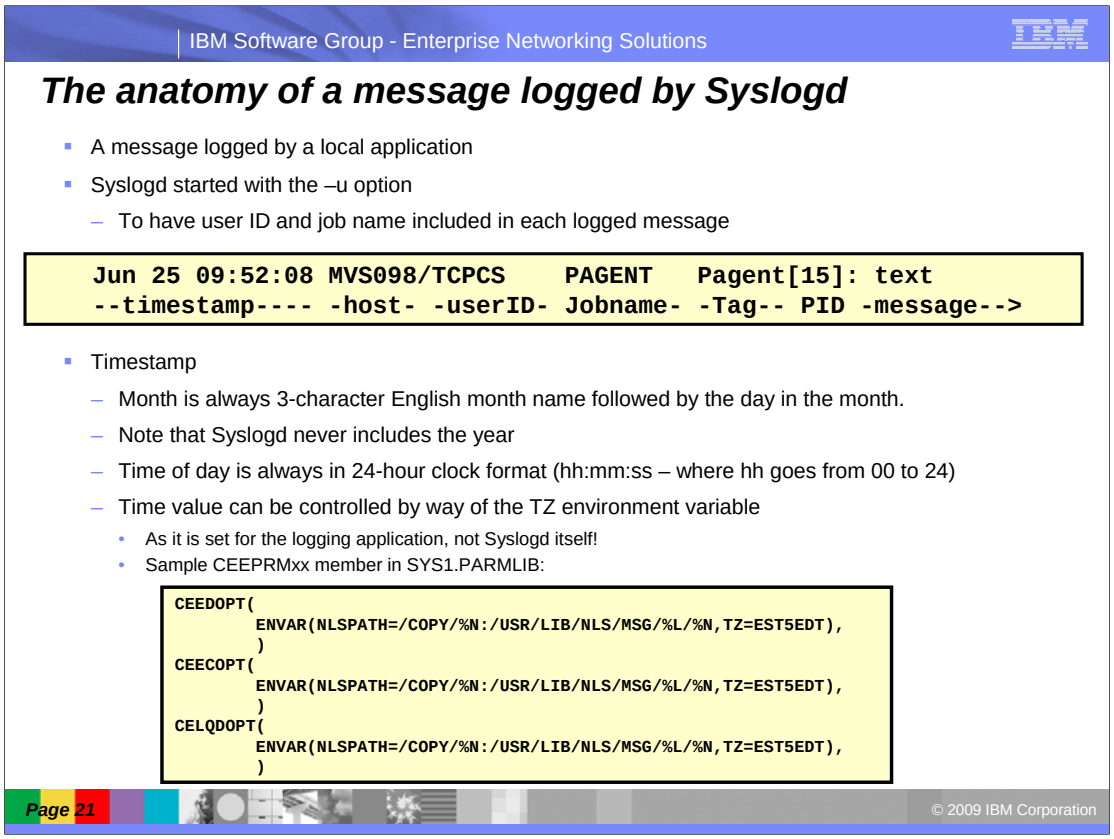
Search arguments:

All search arguments are optional. A time value must be accompanied by the corresponding date. A date can be entered without a time (default from time is 00:00:00 and default to time is 24:00:00).

All specified search arguments are logically ANDed together.

If you specify a specific search criteria and a messages has no value for that criteria, the message is considered a non-hit. Example: if you specify a user ID, but a message has no user ID such a message is not considered a hit. This can be the case if Syslogd has not been started with the –u option,

Sometimes, message text case does matter. If you say NO to ‘Case sensitivity’, search for a string of ‘abc’, messages with ‘ABC’, ‘abc’, ‘Abc’, and so on. will be considered hits. If you specify YES to ‘Case sensitivity’ search for a string of ‘abc’, then only messages with the exact matching case ‘abc’ will be considered hits. Note the case sensitivity option only applies to the four free-form string fields.



To have all messages logged with your local time, it is recommended you set the TZ environment variable in the CEEPRMxx PARMLIB member – you need to define the TZ environment variable for all three LE option sets (CEEDOPT, CEECOPT, and CELQDOPT).

To support search across new year, the browser applies this logic to all time stamps. If message month.date is later than the current month.date – the year is assumed to be the previous year, otherwise the year is assumed to be the current year. Example: if today is Jan 4 2009, and a message with a date of Dec 30 is processed, the year of the message is set to 2008. Another message with the date of Jan 1 is processed, the year of the message is set to 2009. This logic allows browsing a year back in time across new year, but not more than one year.

For local messages, host name is the host name that is configured in TCPIP.DATA.

For remote messages, host name is the DNS name of the remote host or the IP address of the remote host where the IP address is included in parenthesis: (10.1.2.3). Syslogd will resolve remote IP addresses to host names only when you start Syslogd with the –x option.

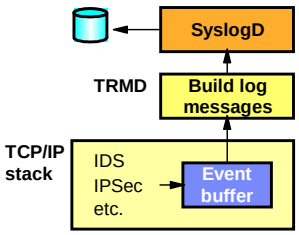
User ID and job name are available for local messages when Syslogd has been started with the –u flag. The message tag is an optional character string that can be passed by the logging application and generally identified the application or component that created this log message.

The process ID is included if the logging application specifies the LOG_PID option on its open_log call. The PID is always enclosed in square brackets and those square brackets are always encoded according to IBM-1047 (the square brackets in the logged messages are not subject to any locale configured by the installation).

Managing TRMD

- **TRMD is stack-specific**
  - It determines which stack to use based on the TCPIPJOBNAME in its resolver configuration file
  - z/OS V1R11 adds a start option to specify the stack name in the EXEC PARM field
    - -p stackname

```
//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)

TRMD is used as a go-between some of the stack components that need to log messages to syslogd and syslogd itself. The IP filtering and VPN functions in the stack are at critical paths in the process and to avoid delaying them by formatting messages, they record a small structure in a storage buffer, and TRMD then a little later picks that entry up, formats a message, and sends it to syslogd.

This also means that if the function that recorded the event put a time stamp into the buffer entry, the message when seen in the syslogd log file may have two timestamps:

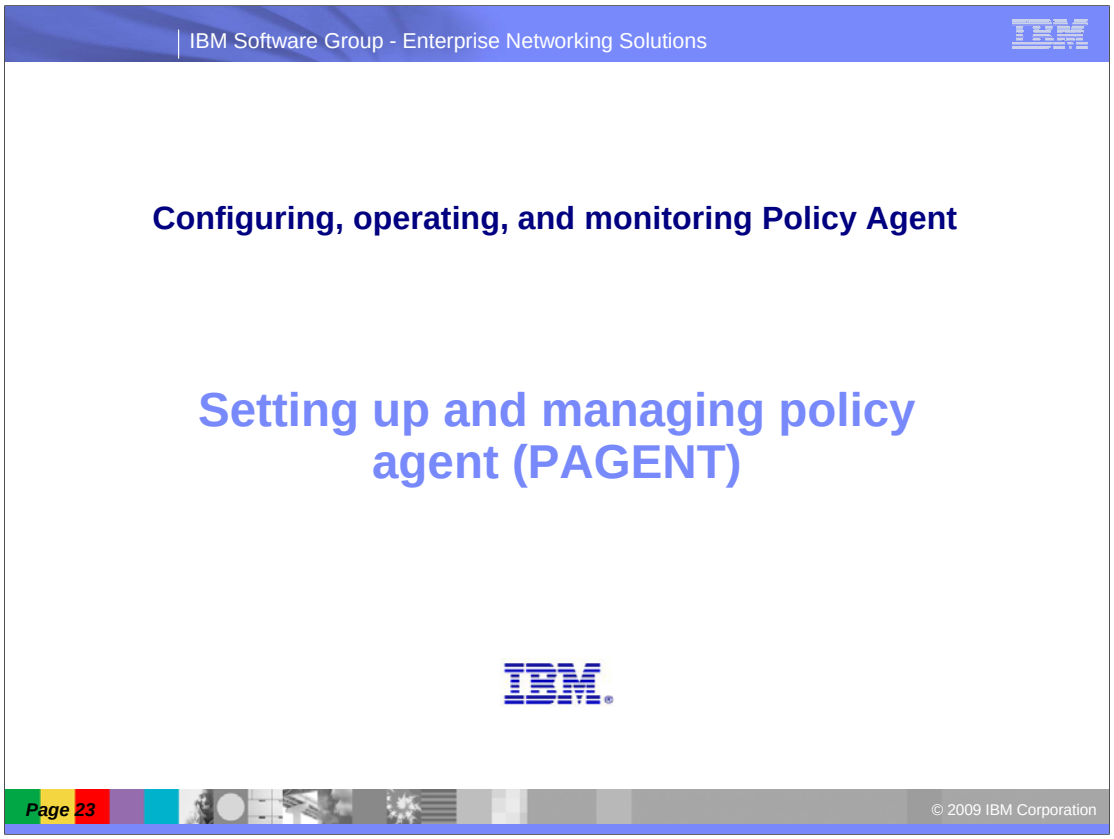
One when logged by syslogd

One when the event occurred

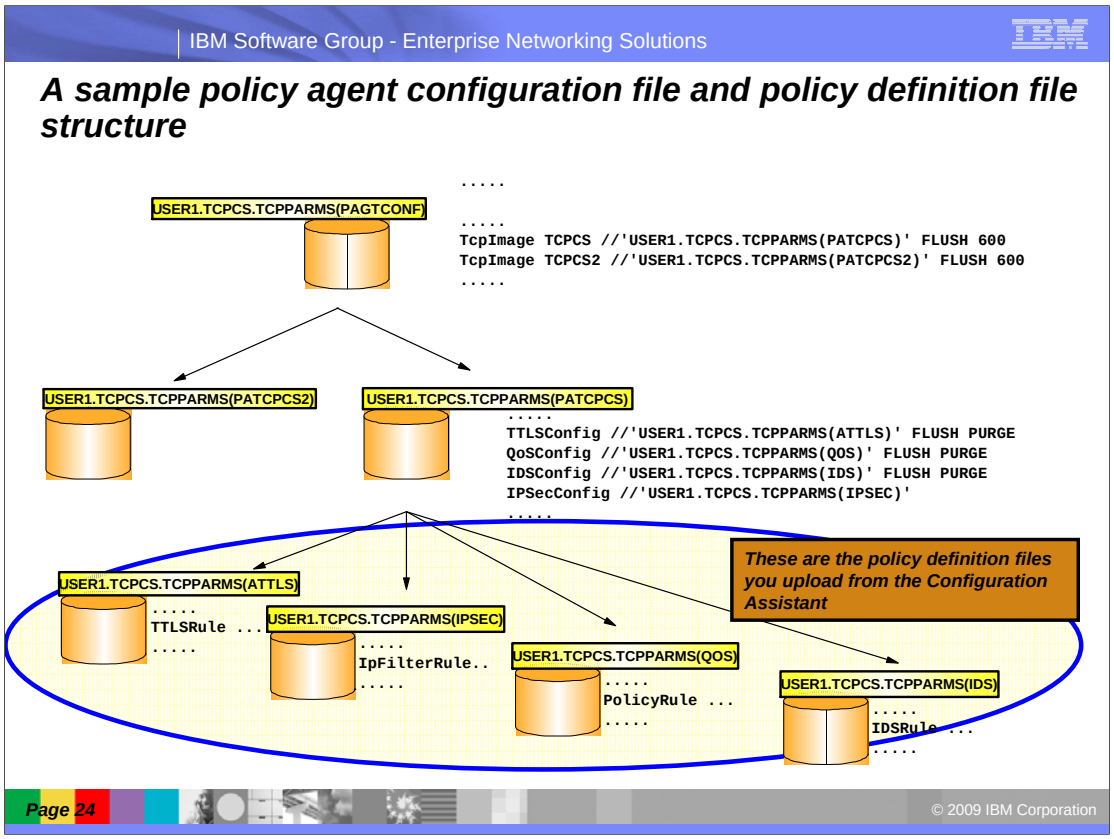
These two may be several seconds apart. This is not an error.

Up until R11, TRMD was told which stack to work with indirectly via the TCPIPJOBNAME in the resolver file it was instructed to use. R11 adds a command line option to specify the stack name to make it work more like most of the other components.

TRMD still forks, so when starting TRDMA – it becomes TRMDA1. You can stop TRMDA1 – or you can use the new Policy Agent monitoring function to control the start and stop of TRMD.

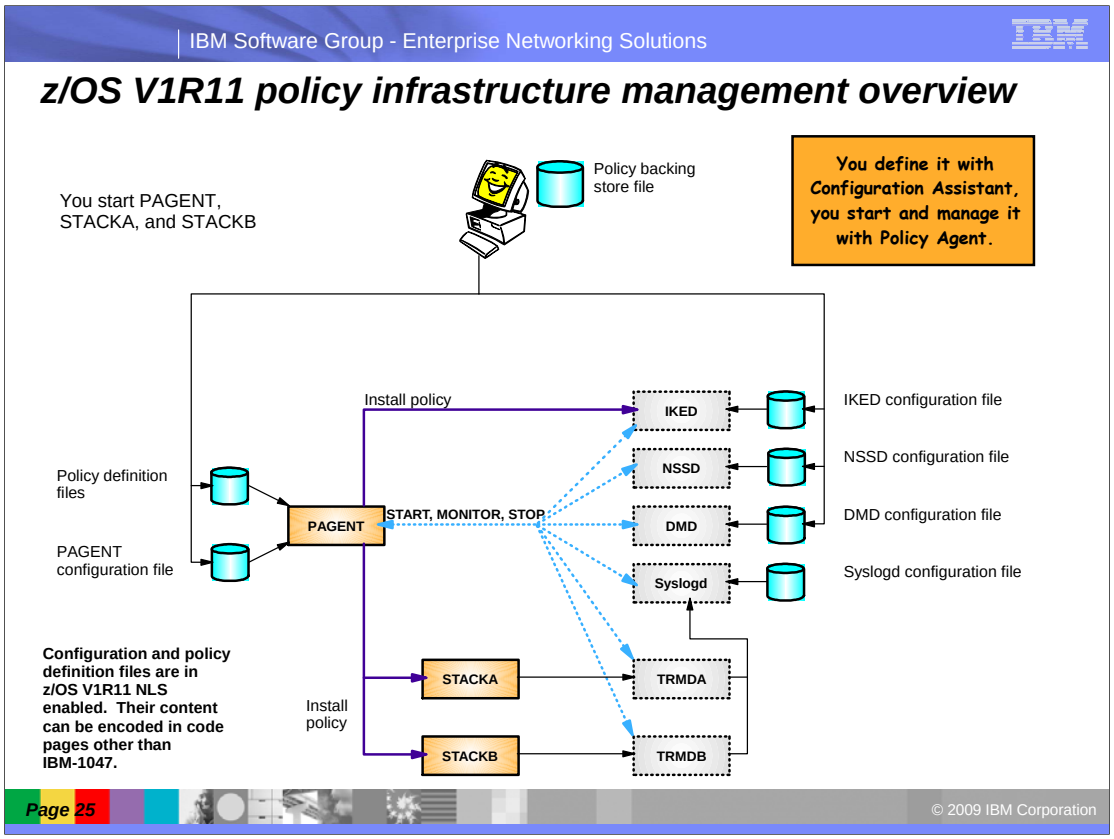


Next step is to get Policy Agent up-and-running.

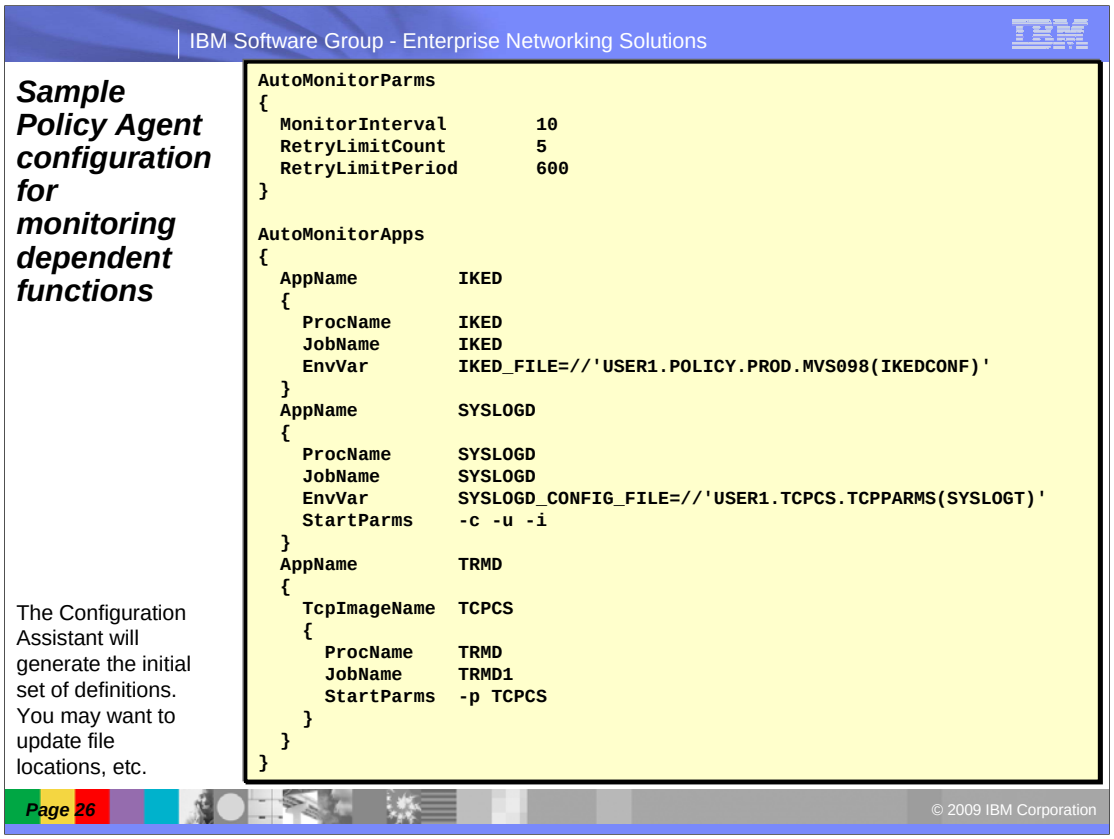


You want to keep your policy definition files in separate locations per stack and policy type. Having such a structure makes it easier to use the Configuration Assistant to maintain these files – it generates them per stack and per policy type.

Policy Agent supports such a structure as represented in the slide. The new functions in the Configuration Assistant to create the Policy Agent configuration uses this approach.



With this new support, one method of starting TCPIP in an LPAR is to start by starting Policy Agent, which then will start and monitor the policy-related functions. Then start each of the stacks on that LPAR (and let AUTOLOG start and monitor servers). When a stack is started, it is registered by Policy Agent, and it will the stack-specific TRMD address space and load the needed policies into the stack.



The AutoMonitorParms statement in the Policy Agent main configuration file configures global application monitoring parameters.

The MonitorInterval parameter specifies the number of seconds in the monitor interval, which is how often Policy Agent checks to see if the monitored applications are active. The default is 10 seconds.

The RetryLimitCount and RetryLimitPeriod parameters work together. They indicate how many times within a given time period applications should be restarted. The default is five times within 600 seconds (10 minutes). If these limits are exceeded, Policy Agent stops monitoring the application, and does not try to restart it any more. After you've resolved the application problem, you can restart the application and resume monitoring using an operator command. The commands are described later in this presentation.

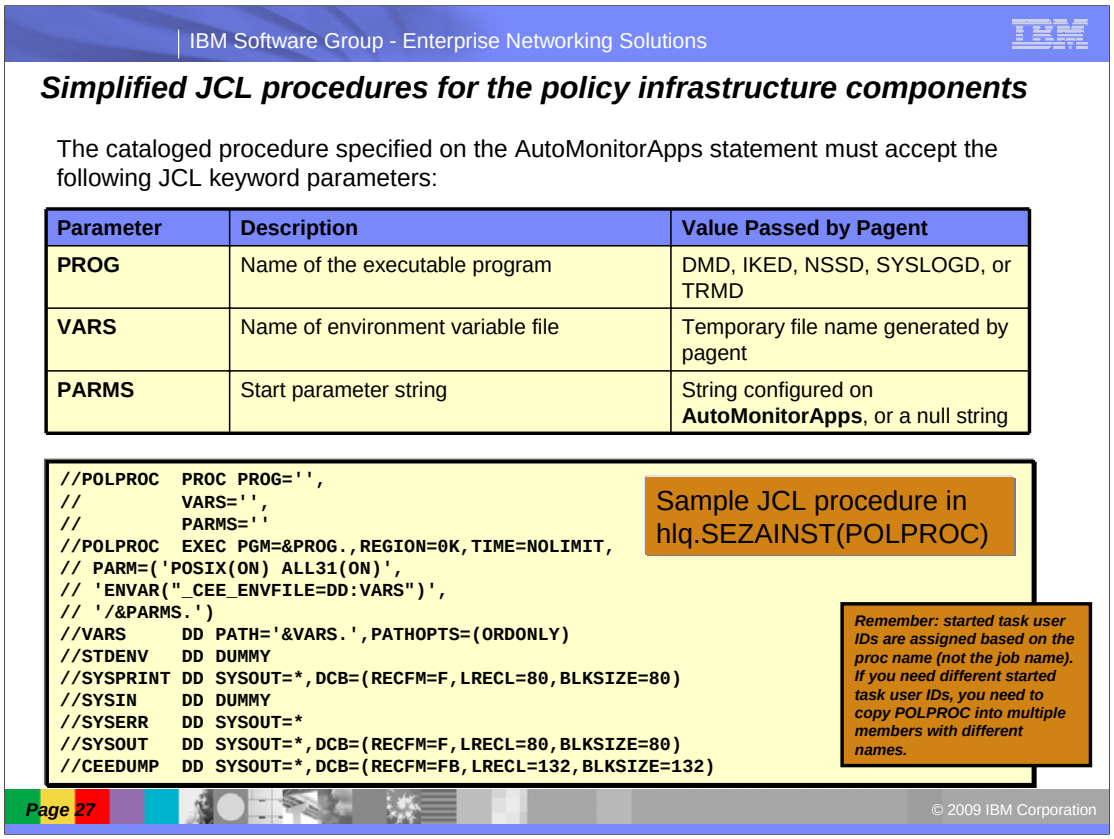
The AutoMonitorApps statement in the Policy Agent main configuration file configures which applications are to be monitored and parameters specific to those applications. You use the AppName parameter to specify each application, and repeat this parameter for the remaining applications you want to monitor. For those applications that run a unique instance on each TCP/IP stack, specify the TcpImageName parameter for the stack name, and repeat this parameter for each stack. You must also configure the TcpImage statement in the main configuration file for each stack configured on the AutoMonitorApps statement.

The ProcName parameter is required, and specifies the name of the cataloged procedure that starts the application. The specified procedure must accept parameters that are passed to it by Policy Agent. The parameters are detailed on the next slide.

The Jobname parameter specifies the job name that is used when the application runs. It defaults to the AppName parameter. You should specify a unique job name for each instance of TRMD.

The StartParms parameter specifies the start options for the application. Specify parameters like you do on the PARM parameter on the JCL EXEC statement, or when starting the application from the UNIX shell. The maximum length of this parameter is 45 characters, so you should specify long values using environment variables where possible.

The EnvVar parameter specifies an environment variable. Specify environment variables like you do in an environment variable file or on the UNIX shell export command. For example: EnvVar TZ=EST5EDT. Repeat this parameter for each environment variable you want to specify.



Three parameters must be accepted by the cataloged procedure that Policy Agent uses to start or restart monitored applications.

The PROG parameter specifies the name of the executable program, and one of the supported application names is always passed by Policy Agent.

The VARS parameter is the name of a file containing environment variables. Policy Agent creates a temporary file and populates it with the configured environment variables. It then passes the name of the temporary file to the procedure.

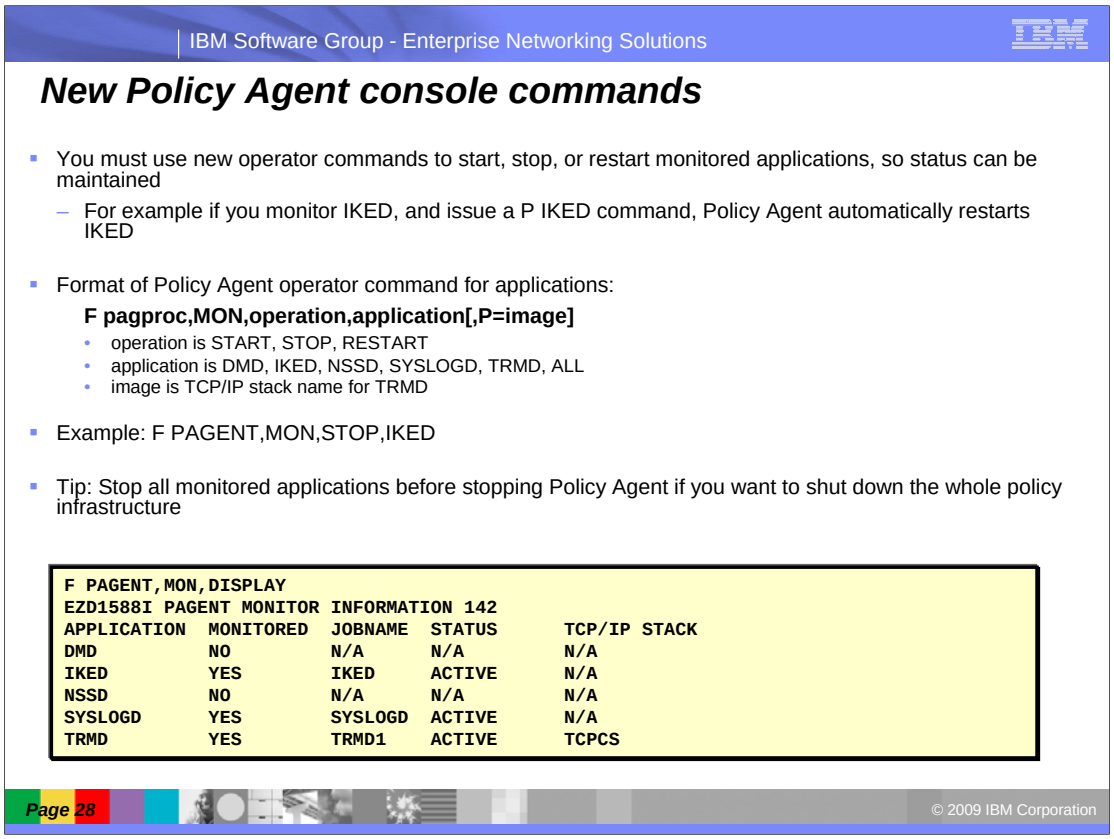
The PARMS parameter specifies the start parameter string. Policy Agen passes the configured string, or a null string if no start parameters are configured.

You can use the sample procedure shipped in SEZAINST(POLPROC) as a template to develop your own procedures.

When Policy Agent starts or restarts an application it waits up to one minute for the application to become active. If the application isn't active after one minute, Policy Agent restarts it. You should allow for this one minute start wait when configuring the RetryLimit and RetryPeriod parameters. For example, if you configure five retries within a three minute period, Policy Agent will never stop trying to restart an application. This is because it takes three minutes to restart the application three times, so you never reach five retries within that period.

When an application stops unexpectedly, Policy Agent is immediately informed. The application is restarted after a short delay. Notice that this occurs regardless of the configured monitor interval.

You might configure an application to be monitored, but then start the application before starting Policy Agent. If you do this, and the application was already running with the configured job name, there are several consequences. First, Policy Agent will still try to start the application, and the start will fail because the application is already active. Also, if Policy Agent needs to later restart the application, it uses the configured procedure name and job name. These might not match the procedure name and job name that you used to originally start the application. It's therefore best if you do not start monitored applications before starting Policy Agent. The one exception is Syslogd, which you normally want to start very early. Notice that if you start a monitored application using a different job name than configured, Policy Agent is not able to successfully monitor the application, because it looks for active applications using the configured job name.



When you monitor applications using the Policy Agent, you need to use a set of new Policy Agent operator commands to start, stop, or restart the applications. This allows the Policy Agent to keep track of the current status of the applications. One example of why this is needed is as follows. If you were to stop an application directly, for example by issuing a P IKED command, Policy Agent does not know you intended to stop IKED and restarts it.

The format of the new commands is shown on this slide. You can perform start, stop, and restart operations against an individual application, or all applications. For TRMD, you can select which instance by using the P=image parameter on the command.

If you stop the Policy Agent, all monitored applications remain active. If you want to stop all policy infrastructure components that are being monitored, issue the MODIFY MON,STOP,ALL command before stopping Policy Agent.

You can use the MODIFY MON,DISPLAY command to display the current status of all applications. This includes whether the application is monitored, the job name, and the status. For TRMD, it also includes the associated stack name. A complete description of the various status values is shown later in this presentation.

The application status is the most important piece of information in case of problems.

STOPPED - Application has never been started, failed to start, or was manually stopped

INACTIVE - Application is temporarily inactive (for TRMD this means the stack is inactive)

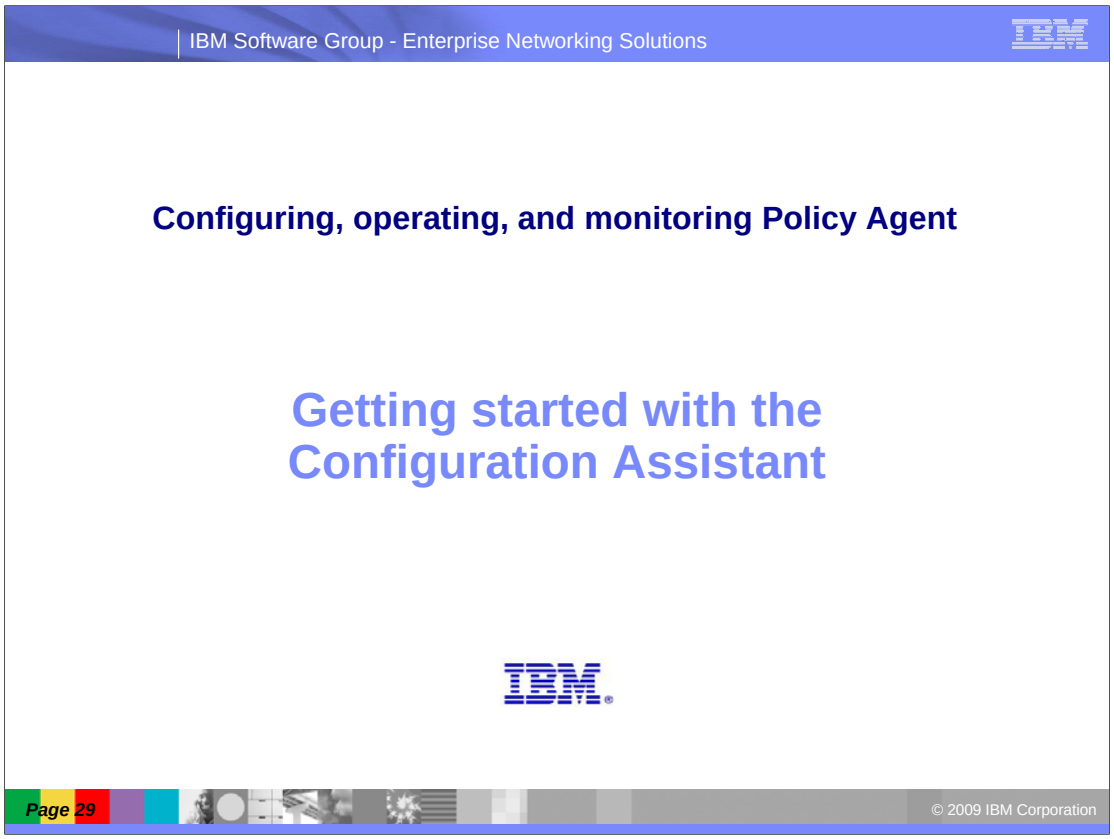
STARTING - Application has been started but is not yet active

RESTARTING - Application has been restarted but is not yet active

STOPPING - Application has been stopped but is not yet inactive

ACTIVE - Application is active

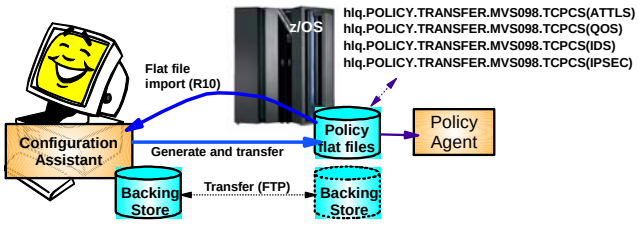
N/A - Application is not being monitored



This is not a full-blown tutorial on the how to use the Configuration Assistant, but merely a brief introduction of some basic concepts that are needed ion order to get started using the Configuration Assistant.

Configuration Assistant files - overview

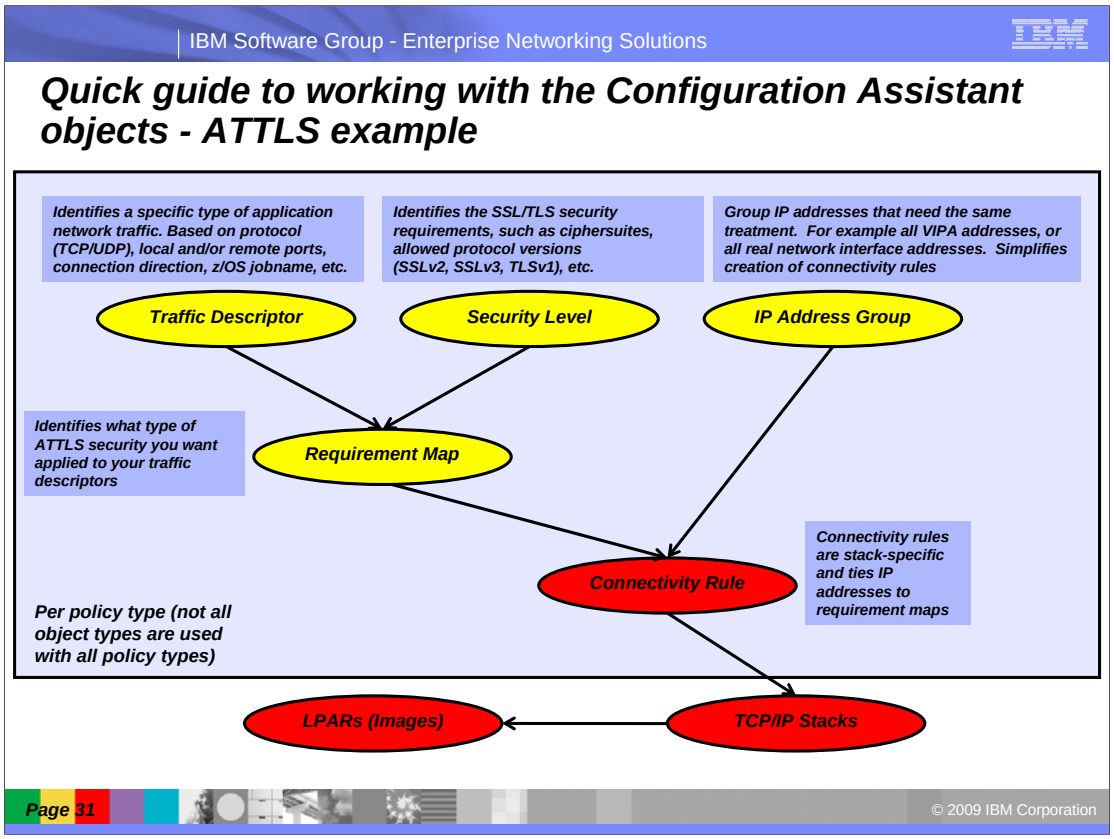
- **The configuration assistant reads and stores all information in binary form in the backing store file:**
  - Think of it as a binary version of all your z/OS CS networking policy definitions
  - You can maintain policies for many LPARs, stacks, and policy types in a single backing store file
  - 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 definitions, 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 a z/OS UNIX file or MVS data set**
  - z/OS backing store files supported from z/OS V1R9
  - 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 discipline has been updated, the configuration assistant can generate the policy flat file that can be read by Policy Agent - and transfer it to z/OS using FTP**
- **In z/OS V1R10, Configuration Assistant can read and import an existing policy flat file**
  - Start from manually created policy definition files
  - Import into the Configuration Assistant after manual edit of a policy definition file



Recommendation is to have a backing store file per Sysplex – have all LPARs, stacks, and policy types for that Sysplex in a single backing store file. Doing so provides the maximum amount of reusability.

The backing store file may be on Windows, on a LAN server, in a z/OS UNIX file, or in a z/OS MVS data set. The z/OS MF version does not support MVS data sets.

The backing store file is a binary representation of your definitions. They need to be converted to a format the z/OS policy components understand – a text-based format. So when you have completed defining your policies, you need to generate and store the actual text-based configuration and policy definitions files on z/OS. They can be stored in MVS data sets (typically PDS(E) members), or in z/OS UNIX files.

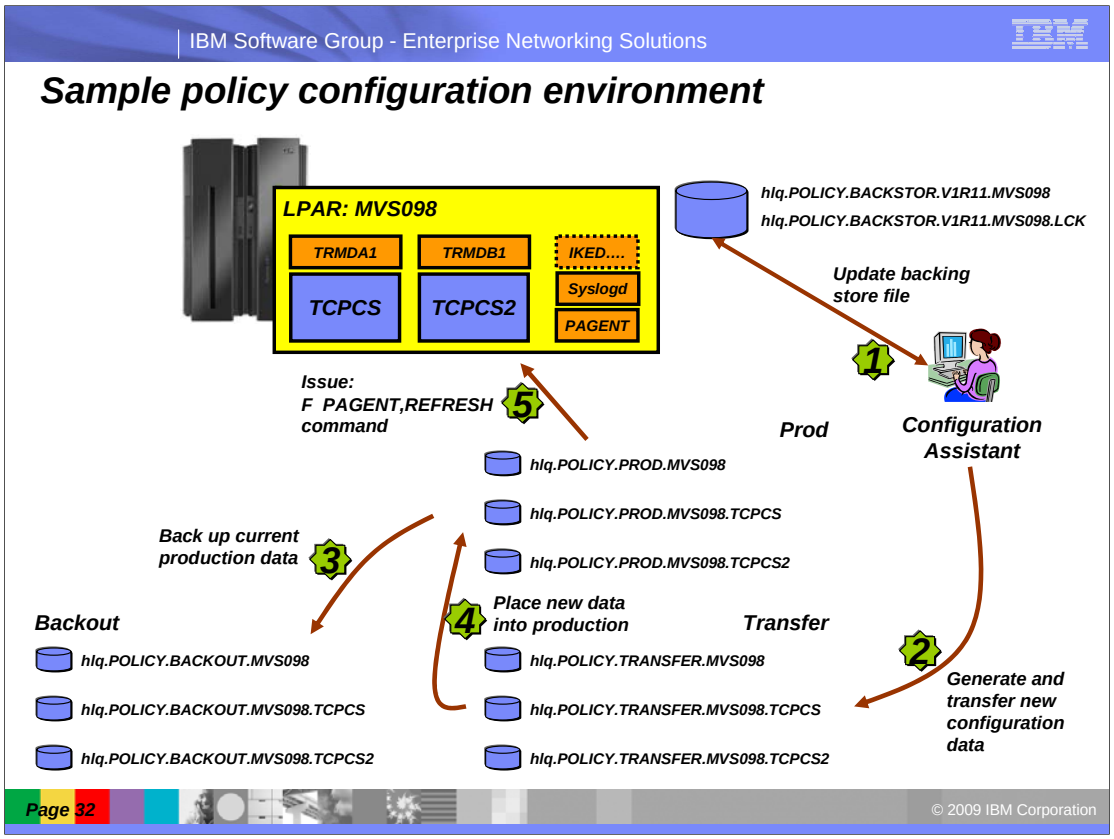


The configuration Assistant simplifies the configuration structuring the definitions into inter-related objects.

For AT-TLS, traffic descriptors describe the application (transport protocol, port numbers, etc.), and the security level describes the level of encryption needed. Combining traffic descriptors with security levels form what is referred to as a requirement map – mapping the application security requirements to supported security levels. So far, all definitions are generic – no IP address information up until now. All definitions so far can be re-used across multiple stacks and LPARs (that reside in the same backing store file).

Per stack, the requirement maps are lined to local and remote IP address information. Either individual IP addresses, or addressed defined in re-usable IP address groups. Dynamic VIPA addresses would typically be in a group – the same requirements exists no matter which stack a given DVIPA is active on at a given point in time.

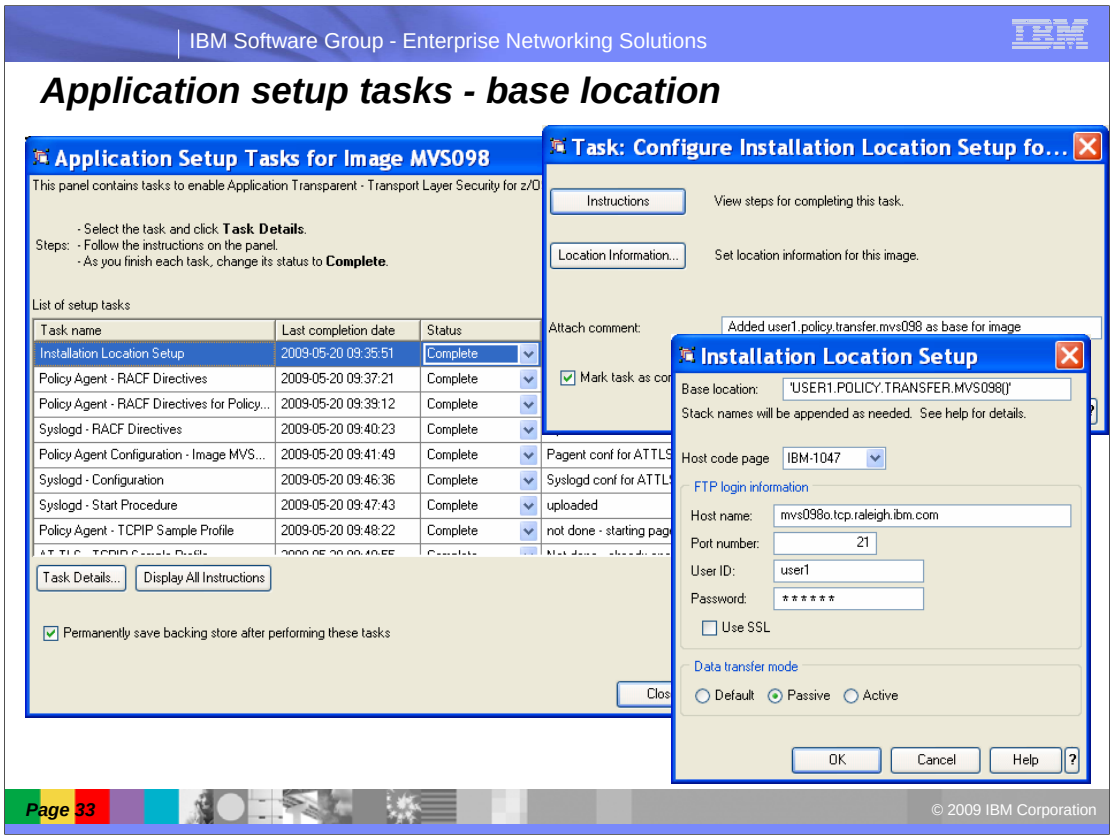
Connectivity rules exists per stack, and stacks exist on LPARs (or images to be generic).



This example shows how you can work with the Configuration Assistant and a z/OS LPAR. The sample uses MVS PDS(E) libraries for all policy components, and it also uses z/OS for storing the backing store file. The backing store file is stored as a sequential MVS data set, while the policy configuration data components are stored as members of PDS(E) libraries.

The sample uses two stacks (TCPCS and TCPCS2) on an LPAR (image name MVS098).

The sample shows a suggested structure for honoring traditional change management procedures: having a transfer library set into which the Configuration Assistant stores new and changed definitions. A production library set from which the system reads the current definitions, and a back out library set for use in case of emergency back out of changed definitions. The simple method for production cutover to copy the content of the production library set to the back out library set, followed by copying the transfer library set to the production library set – and either restarting the policy components or issue modify commands for them to refresh their configurations



The Installation Location Setup panel provides a means to set a base location for the staging area (the tranfer library set in our sample setup). When you provide a base location for each image, the Configuration Assistant will use this to create names for the files. The files will be organized by image name and within that by stack name for files that are specific to the stack.

These file names may then be used when the configuration materials are transferred to the target host. You may override the provided file names at that time, but this base location provides a suggested set of names for you to use.

The base location may be a zFS file path, a dataset HLQ, or a partitioned dataset name. Whatever base location is specified, the administrator user id should have the authority to write files at that location.

The Location Information panel also provides a place to set FTP login information to be used by the FTP delivery process. This FTP login information will be the default for delivering files for the current image. At the final delivery stage, the administrator may override the file name, the installation method (the browser client allows save to disk selection), or any of the FTP login details.

The base location may be a zFS directory:

/etc/cfgasst/v1r11/TEST9/

It may be an MVS sequential data set hlq:

USER1.CFGASST.TEST9.

Or it may be PDS(E) library name:

USER1.CFGASST.TEST9()

The Configuration Assistant will assume PDS(E) libraries have been created before using them. If the sample LPAR (in the context of the above sample LPAR name of TEST9) has a stack named STACK1 then two PDS(E) libraries must exist: one named USER1.CFGASST.TEST9 and one named USER1.CFGASST.TEST9.STACK1. If the LPAR has multiple stacks, a PDS(E) library per stack must be created.

Content of base locations after application setup tasks performed

Image PDS(E) library members

Component	Description
DMDCONF	DM configuration file
DMDPROC	DM JCL start procedure
DMDPROF	TCP/IP Profile sample IPSECURITY stmts.
IKEDCONF	IKE configuration file
IKEDPROC	IKE JCL start procedure
IMGPAG`	Image PAGENT configuration file
IPSPROF	TCP/IP Profile sample IPSECURITY stmts.
PAGPROC	Pagent JCL start procedure
RDMD	DM RACF setup commands
RIKED	IKE RACF setup commands
RIPSEC	RACF setup commands for ipsec cmd.
RPAGENT	Pagent RACF setup commands
RPOLICY	RACF setup commands for Policy data import
RSYSLOGD	Syslogd RACF setup commands
RTRMD	TRMD RACF setup commands
SYSLOCONF	Snippets for Syslogd configuration file
SYSLOGD	Syslogd JCL start procedure

Stack PDS(E) library members

Component	Description
IDSPOL	IDS policy
IPSPOL	IPSec policy
QOSPOL	QoS policy
STKPAG	Stack Pagent configuration
TLSPOL	ATTLS policy
TRMDPROC	TRMD JCL procedure

- ***Start PAGENT before any stacks are started***
  - ***Pagent will start Syslogd and other LPAR-wide components, such as IKED***
- ***When a stack is started, PAGENT notices it***
  - ***Pagent will then start the stack-specific TRMD***
  - ***Pagent will load all the relevant policies into that stack***

In this example, we defined policies for IPSec, ATTLS, QoS, and IDS.

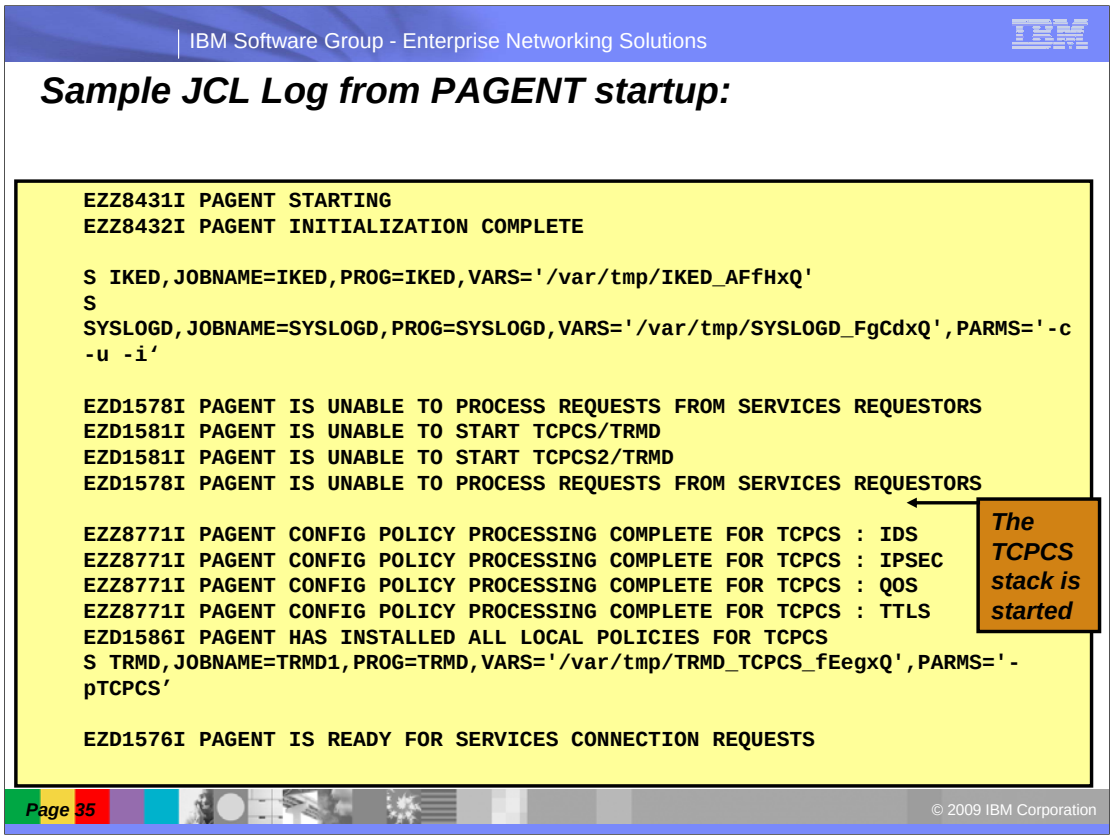
After having defined all the policy disciplines and policies in the Configuration Assistant – and having performed all the suggested installation setup tasks, the sample PDS(E) libraries will contain a number of members. The names used here are the default names – they can be overridden.

There is still a little setup work to do on z/OS after having defined everything and transferred everything to z/OS:

The RACF jobs needs to be analyzed and executed

The JCL procedures need to be copied to a valid procedure library (the RACF jobs will create STARTED TASK profiles for the procedures)

The IMGPAG and STKPAG configuration files need to be checked an extra time and any necessary modifications made to them.

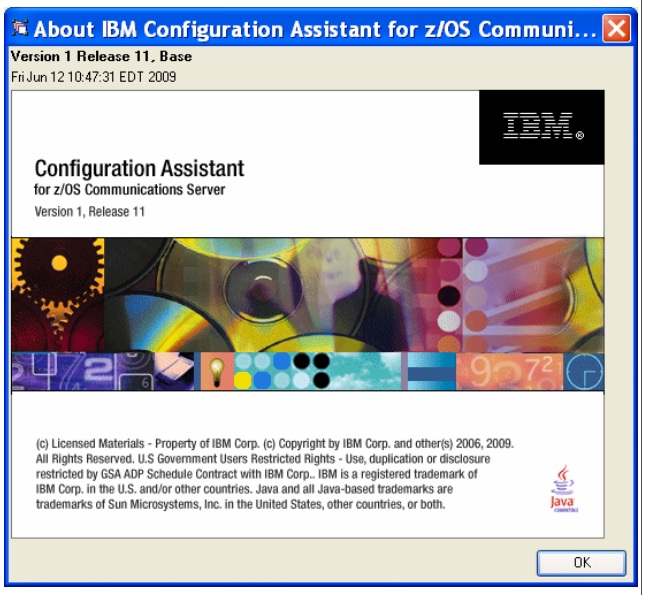


This is the PAGENT job log from a start-up sequence on an LPAR where one stack (TCPCS) is started after PAGENT has started.

IBM Software Group - Enterprise Networking Solutions

Configuration Assistant for z/OS Communications Server

- **Configuration Assistant for z/OS V1R11 Communications Server is shipped with the z/OSMF product**
  - Runs on z/OS
  - Accessed from a Web browser
  - Support is via normal IBM support channels
  - Same basic functions as the Windows-based version
- **The Windows-based standalone version remains available for z/OS V1R11, and can be downloaded from the web:**
  - Versions for z/OS V1R7, V1R8, V1R9, V1R10, and V1R11 are available for download
  - 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
  - Support is "informal" via a forum



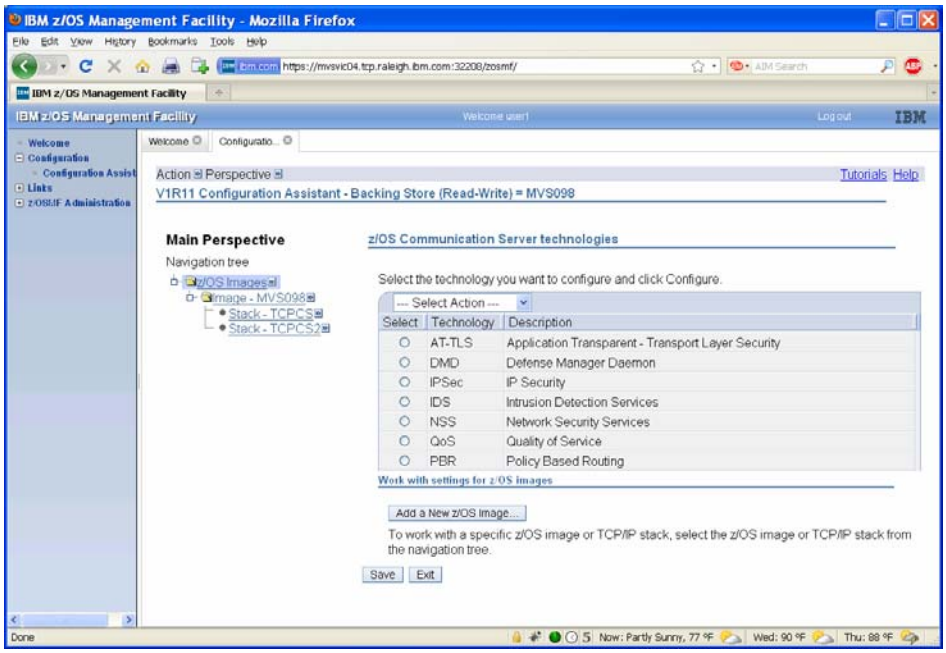
Tired of that long URL above – try this one instead: http://tinyurl.com/cgoqsa

Page 36

The IBM Configuration Assistant for z/OS V1R11 Communications Server will in due time be made available at the same location as shown above.

The Configuration Assistant in z/OS V1R11 is delivered both as a windows-based standalone application and with the new z/OS Management Facility offering on z/OS. The z/OSMF version is a normal IBM product fully supported via the normal IBM support channels.

Configuration Assistant in z/OSMF



This is a screen shot of the Configuration Assistant running in z/OSMF. The panels look slightly different and some of the navigation follows different rules, but in general, if one is familiar with the Windows-based version, the z/OSMF version comes very naturally.

For more information

URL	Content
http://www.ibm.com/servers/eserver/zseries	IBM eServer zSeries Mainframe Servers
http://www.ibm.com/servers/eserver/zseries/networking	Networking: IBM zSeries Servers
http://www.ibm.com/servers/eserver/zseries/networking/technology.html	IBM Enterprise Servers: Networking Technologies
http://www.ibm.com/software/network/commserver	Communications Server product overview
http://www.ibm.com/software/network/commserver/zos/	z/OS Communications Server
http://www.ibm.com/software/network/commserver/z_lin/	Communications Server for Linux on zSeries
http://www.ibm.com/software/network/ccl	Communication Controller for Linux on zSeries
http://www.ibm.com/software/network/commserver/library	Communications Server products - white papers, product documentation, etc.
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/	Technical support documentation (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)

For pleasant night-time reading ….