V5R1 OA06856 Readme
Message Automation for Intrusion Detection Services (IDS)

Version 5 Release 1
V5R1 OA06856 Readme
Message Automation for Intrusion Detection Services (IDS)

Version 5 Release 1

This edition applies to Version 5, Release 1 of Tivoli NetView for z/OS (product number 5697-ENV) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2004. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

About this Program Temporary Fix ........................................... v
New in this PTF ................................................................. v
Prerequisites ........................................................................ v
Backward compatibility. ....................................................... v
Recommended Publications ................................................... v

Intrusion Detection Services ................................................. 1
Message Automation for Intrusion Detection Services (IDS). ....... 2
   Notifications .................................................................... 2
   Data gathering. ................................................................ 2
   Timer events ..................................................................... 2

Installation considerations. ............................................... 5

Enabling towers ............................................................... 7

Customize AON/TCP policy definitions ..................................... 9
TCP930 ............................................................................. 9
NOTIFY ............................................................................. 9
NTFYOP. .......................................................................... 9
INFORM ........................................................................... 10

Customize CNMSTYLE ................................................... 11
CNMSTYLE statements ............................................................ 12
   IDS.Attack_Cmd .......................................................... 12
   IDS.Attack_CmdType .................................................. 13
   IDS.Auto_Intvl ........................................................... 13
   IDS.Auto_Thresh ........................................................ 14
   IDS.ClearStat_Day ....................................................... 14
   IDS.ClearStat_Inform ................................................... 15
   IDS.ClearStat_Log ......................................................... 16
   IDS.ClearStat_Log_File ............................................... 16
   IDS.ClearStat_Time ....................................................... 16
   IDS.CONSOLEMSG ...................................................... 17
   IDS.DSIPARM .............................................................. 17
   IDS.Event_Inform ........................................................ 18
   IDS.Event_Limit ........................................................... 18
   IDS.Event_Log .............................................................. 19
   IDS.Event_Log_File ....................................................... 19
   IDS.Flood_Cmd ............................................................ 20
   IDS.Flood_CmdType ..................................................... 20
   IDS.probeid ................................................................. 21
   IDS.Report_Cmd ........................................................... 21
   IDS.Report_CmdType .................................................... 22
   IDS.Report_Inform ......................................................... 23
   IDS.Report_Log ............................................................. 23
   IDS.Report_Log_File ...................................................... 24
   IDS.Scan_Cmd ............................................................. 24
   IDS.Scan_CmdType ....................................................... 25
   IDS.SYSLOG.FILENAME .................................................... 26
   IDS.SYSLOGMSG .......................................................... 26
   IDS.TCP_Cmd ............................................................... 27
   IDS.TCP_CmdType ......................................................... 27
   IDS.UDP_Cmd ............................................................... 28
About this Program Temporary Fix

This document provides important information about message automation for Intrusion Detection Services (IDS) being added to Tivoli® NetView® for z/OS® Version 5 Release 1. This document contains the most current information for the product and takes precedence over all other documentation.

Please review these notes thoroughly before installing or using this product.

New in this PTF

This PTF enables message automation for Intrusion Detection Services (IDS).

Prerequisites

This PTF can be used with Communications Server on z/OS Version 1 Release 2 or later. Communications Server must be configured to provide Intrusion Detection Services (IDS). The Communications Server publications listed in "Recommended Publications" provide information on configuring Communications Server for IDS.

NetView Event/Automation Services (E/AS) must be installed if you want to route SYSLOG messages to T/EC or the Tivoli Risk Manager.

The message adapter service of the NetView for z/OS event/automation services must be customized in order to send events to the Tivoli Risk Manager. Communications Server provides a message format file that provides the conversion from syslog messages to the appropriate T/EC events. This Communications Server message format file can be downloaded from http://www.ibm.com/support/docview.wss?uid=swg24006973

In order to receive e-mail notifications, you must use SMTP or an equivalent interface.

Backward compatibility

This PTF applies only to NetView for z/OS Version 5 Release 1.

Recommended Publications

In addition to the NetView publications referred to in this document, you will also need to refer to these books to fully utilize Intrusion Detection Services:

- z/OS Communications Server: IP Configuration Guide (SC31-8775)
  Contains detailed information about Intrusion Detection Services (IDS) policies.

- z/OS Communications Server: IP Configuration Reference (SC31-8776)
  Contains a list of LDAP object classes and attributes used to define IDS policy objects.

- z/OS Communications Server: IP and SNA Codes (SC31-8791)
  Defines probeids and provides a detailed discussion of probeids.
z/OS Communications Server: IP Messages Volume 4 (SC31–8786)
Roster of TCP/IP messages, including messages used for Intrusion Detection Services.

z/OS Communications Server: IP System Administrator’s Commands (SC31–8781)
Detailed description of the trmdstat (Traffic Regulation Management Daemon) commands that generate reports based on IDS messages.
Intrusion Detection Services

It is becoming increasingly important to not merely protect systems from attacks, but also to detect patterns of usage that might indicate impending attacks. Many attacks follow a sequence of information gathering, unauthorized access to resources (information, applications, storage) and denial of service. It can be difficult, or at times, impossible, to determine the originator of denial of service attacks. Correlating information gathering activities with access violations may help identify an intruder before they succeed.

Intrusion Detection Services provides support, detection, and regulation of:

Scan detection and reporting
Scans are recognized as the result of multiple information gathering events from a single source Internet Protocol (IP) within a defined period of time. Scanning in and of itself it not harmful. However, many serious attacks, especially access violation attacks, are preceded by information gathering scans.

Attack detection, reporting, and prevention
An attack can be a single packet designed to crash or hang a system. An attack can also consist of multiple packets designed to consume a limited resource, causing a network, system, or application to be unavailable to its intended users (that is, denial of service). IDS attack policy allows you to turn on attack detection for one or more categories of attacks independently of each other. Actions that can be specified for an attack policy are event logging, statistics gathering, packet tracing, and discarding of the attack packets.

Note: "Flood" events are usually considered a subcategory of Attacks. Flood events are "denial of service" events which occur when a server is flooded with connection requests from incorrect or nonexistent source Internet Protocol (IP) addresses.

Traffic regulation for UDP (User Datagram Protocol) events
Intrusion Detection Services (IDS) Traffic regulation (TR) policies for UDP based applications limits the amount of storage that can be consumed by inbound datagrams for any single bound port.

Traffic regulation for TCP (Transmission Control Protocol) connections
Intrusion Detection Services (IDS) Traffic regulation (TR) policies for TCP ports limits the total number of connections an application has active at one time.

The Communications Server for z/OS Version 1 Release 2 and later can detect TCP/IP intrusions and generate messages that can be passed on to NetView and the automation table. The Communications Server can generate IDS console messages, IDS syslog messages, or both IDS console messages and syslog messages. Both types of messages begin with EZZ and are defined in the z/OS Communications Server: IP Messages Volume 4.

These Intrusion Detection Services events are monitored:
• TCP traffic
• UDP traffic
• Scan detection
• Attack detection (and Attack flood events)

Message Automation for Intrusion Detection Services (IDS)

With this PTF, NetView (via AON/TCP) will automate Communications Server IDS system console or syslog messages and provide the functions of notification, of data gathering, and of timer events. These are described more completely in the following sections.

Notifications

This PTF can generate these forms of notifications:

• Generate an e-mail using SMTP (this is the default) to route IDS messages, along with commands and responses, to IDSOPERS. This can be customized in the INFORM policy definitions.

• Generate a message to the NTFYOPs based on the NOTIFY IDSAUTO policy setting for the MSG parameter. The IBM Tivoli NetView for z/OS Administration Reference contains information about possible settings for the MSG parameter. The NOTIFY IDSAUTO MSG setting will apply to all Communications Server IDS messages. The z/OS Communications Server: IP Messages Volume 4 (SC31–8786) contains detailed information about these messages.

• Generate an alert to NPDA based on the NOTIFY IDSAUTO policy setting for the ALERT parameter. The IBM Tivoli NetView for z/OS Administration Reference contains information about possible settings for the ALERT parameter.

• Generate a Tivoli Enterprise Console (T/EC) event if ALERT=TEC is set in the NOTIFY IDSAUTO statement. This function is only available for IDS syslog messages (IDS.SYSLOGMSG=Y in CNMSTYLE definitions). SYSLOG uses the event/automation service (E/AS) Message Adapter, but will also generate an alert consistent with all other NOTIFY ALERT=TEC commands. Use the T/EC event to route the event to the IBM Tivoli Risk Manager. Additional information about the IBM Tivoli Risk Manager can be found in the IBM Tivoli Risk Manager Administrator’s Guide and the IBM Tivoli Risk Manager User’s Guide, SC23–4822.

Data gathering

Data gathering is accomplished by issuing commands for each of the event types (TCP traffic, UDP traffic, scan detection, attack and flood). The commands can be customized by the use of CNMSTYLE statements. That data will be logged to a file in DSIPARM along with any messages. The data will also be sent via e-mail to operators that have been designated to receive this data.

Timer events

This PTF also provides a mechanism for timer-generated processing of IDS information. Summary statistics are kept, based on the probeids that are in the IDS messages. Based on definitions set in CNMSTYLE, timer-driven commands, such as TRMDSSTAT, can be issued, and the output from such commands can be appended to the summary statistics being kept for each probeid. The reports will be e-mailed and logged to a file in DSIPARM.

Each stack can contain a timer to generate probe/event summary reports. Each stack contains an additional timer to clear the probe/event summary statistics on a regulated basis.

Threshold values can be set in CNMSTYLE to protect NetView from being overrun with a flood of IDS messages. If the IDS message automation threshold is
exceeded, message FXX301I is issued. Processing of any additional IDS messages received within the threshold time interval will be limited; this includes the updating of the probe summary statistics.
Installation considerations

This is a summary of what must be done to set up Intrusion Detection Services:

- Define your stack via a TCP390 Policy Definition in DSIPARM member FXXCFG01 (see "Customize AON/TCP policy definitions" on page 9). At a minimum, code IDSAUTO=Y for that stack. Consider specifying a value for IDSINTVL. You might review the NOTIFY, NTFYOP, and INFORM statements to see if you want to use these statements.
- Uncomment AON in the CNMSTYLE TOWER statement (see "Enabling towers" on page 7).
- Ensure that TOWER.AON=TCP in the CNMSTYLE (see "Enabling towers" on page 7).
- Uncomment TOWER.AON.TCP=IDS in the CNMSTYLE TOWER statement (see "Enabling towers" on page 7).
- Customize the IDS autotask statements as appropriate (see "Customize AON/TCP policy definitions" on page 9).
- Customize the IDS.* statements as appropriate (see "Customize CNMSTYLE" on page 11).
Enabling towers

Specify TOWER=AON in CNMSTYLE
CNMSTYLE contains TOWER=*AON. At a minimum, remove the asterisk (*) so that this now reads TOWER=AON in order to make AON active.

Specify TOWER.AON=TCP in CNMSTYLE
The AON tower enables network automation, and contains the SNA subtower (for SNA automation) and the TCP subtower (for TCP automation). The TCP subtower contains the IDS subtower, which provides automation Intrusion Detection Services support. A hierarchic view of this would be

AON (tower)
SNA (subtower)
TCP (subtower)
IDS (subtower)

Specify TOWER.AON.TCP=IDS in CNMSTYLE
CNMSTYLE contains *TOWER.AON.TCP = IDS. At a minimum, remove the asterisk (*) so that this now reads TOWER.AON.TCP = IDS.
Customize AON/TCP policy definitions

The Automated Operations Network (AON) provides a way to provide automation across multiple network protocols. AON intercepts alerts and messages that indicate problems with network resources and attempts to recover failed resources. The components of AON are:
• TCP/IP automation (AON/TCP)
• SNA automation (AON/SNA)

To enable Intrusion Detection Services (IDS), review the IDS statements in CNMSTYLE %INCLUDE member CNMSTIDS and specify AON policy for the TCP390 definition, and optionally, modify the NOTIFY, NTFYOP, and INFORM policy definitions.

Note: For more information on AON customization, refer to IBM Tivoli NetView for z/OS Automated Operations Network Customization Guide.

TCP390

Use the TCP390 policy statement to define a TCP/IP stack to the NetView program. You must specify a TCP390 entry for each local stack that you want to manage. For each local stack that you define, specify IDSAUTO=Y in the TCP390 statement if you want IDS support for that stack. In the TCP390 statement, you can also specify a value for the IDSINTVL parameter to specify the time interval during which to schedule the IDS-report generation timer.

NOTIFY

You may choose to modify the NOTIFY IDSAUTO policy definition. By default, the NOTIFY IDSAUTO policy definition is set up to issue alerts and messages for IDS events. If you want IDS events to be forwarded to the Tivoli Enterprise Console (T/EC), then you should modify the NOTIFY policy definition. For example:

```
NOTIFY IDSAUTO,
    ALERT=TEC,
    MSG=YES,
    DDF=NO
```

Note: If you choose to have events forwarded to the T/EC, you must also modify the MSG.TECROUTE statement to specify the name of the program-to-program interface (PPI) receiver associated with the event automation service. This is described in “MSG.TECROUTE” on page 29.

NTFYOP

You must define which operators can see IDS-related messages. You can do this by specifying NTFYOP policy definitions. Create a NTFYOP definition with CLASS=64 for each operator who can view messages related to IDS events. For example:

```
NTFYOP OPER1,
    OPER='IDS-AUTO-SVCS',
    CLASS=(64), HELDMSG=(I)
```
INFORM

An integral part of the IDS support is the ability to e-mail reports and responses to commands once an IDS event has occurred. If you are using e-mail alerts to notify operators, you must identify each operator or operators who are to receive e-mail alerts. This is customized via the INFORM policy definition. A sample, EZLINSMP, is provided to help you with this. For example:

GROUP IDSOPERS,LIST=OPER1,OPER2,OPER3;
INFORM OPER1;
   CONTACT CONNECTION=EMAIL,
   INTERFACE=EZLESMTP,
   ROUTE=IBPERSC@US.IBM.COM,
   NAME=C. PERSON;
INFORM OPER2,...
INFORM OPER3,...

For more information, refer to sample EZLINSMP.
Customize CNMSTYLE

When CNMSTYLE initializes, the CNMSTIDS member is included. This member includes Intrusion Detection Services statements which must be reviewed if you are enabling IDS support.

**Note:** A complete list of all of the members that are included when CNMSTYLE is initialized can be found in IBM Tivoli NetView for z/OS Installation: Migration Guide. The CNMSTIDS member is specific to Intrusion Detection Services, and the only member discussed in this document.

Detailed descriptions of the statements specific to Intrusion Detection Services are contained in “CNMSTYLE statements” on page 12, but they can be summarized as follows:

- **Source events for Intrusion Detection Services**
  - IDS.CONSOLEMSG
  - IDS.SYSLOGMSG
  - IDS.SYSLOG.FILENAME

- **Log specifications**
  - IDS.Event_Inform
  - IDS.Event_Limit
  - IDS.Event_Log
  - IDS.Event_Log_File

- **Scan detection**
  - IDS.Scan_Cmd
  - IDS.Scan_CmdType

- **Attack detection events**
  - IDS.ATTACK_Cmd
  - IDS.ATTACK_CmdType
  - IDS.FLOOD_Cmd
  - IDS.FLOOD_CmdType

- **TCP events**
  - IDS.TCP_Cmd
  - IDS.TCP_CmdType

- **UDP events**
  - IDS.UDP_Cmd
  - IDS.UDP_CmdType

- **Event threshold values and automation table threshold values**
  - IDS.Auto_Intvl
  - IDS.Auto_Thresh

- **Specify T/EC receiver**
  - MSG.TECROUTE

- **Probe summary statistics**
  - IDS.ClearStat_Day
  - IDS.ClearStat_Inform
  - IDS.ClearStat_Log
  - IDS.ClearStat_Log_File
  - IDS.ClearStat_Time
  - IDS.DSIPARM
  - IDS.probeid
  - IDS.Report_Cmd
Note: If you choose to log reports for any of the events of the Probe Summary Statistics Reports, you will need to compress the DSIPARM data set periodically to prevent reports from being lost.

**CNMSTYLE statements**

These are CNMSTYLE statements that either are new or have been changed to support Intrusion Detection Services.

NetView initialization definitions are contained in DSIPARM member CNMSTYLE and its included members. To make changes to a CNMSTYLE statement, copy the statement to CxxSTGEN and then make any necessary updates. For more information on NetView initialization and CNMSTYLE, refer to [IBM Tivoli NetView for z/OS Installation: Getting Started](#).

**IDS.Attack_Cmd**

The IDS.Attack_Cmd statement specifies the command to be issued when an IDS attack detection event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Attack_Cmd statement is:

```
IDS.Attack_Cmd.suffix=command
```

*Where:*

*suffix*

A numerical suffix. Increment each suffix by one for each command used. In the example that follows, two commands, a UNIX command and a NetView command, are defined.

IDS.Attack_CmdType.1 = UNIX
IDS.Attack_Cmd.1 = /bin/trmdstat -A -D IDS.SYSLOG.FILENAME
IDS.Attack_CmdType.2 = NETV
IDS.Attack_Cmd.2 = WHO

*Note:* Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Attack_Cmd.4 statement will be ignored:

IDS.Attack_Cmd.1
IDS.Attack_Cmd.2
IDS.Attack_Cmd.4

*command*

Specifies the command to be issued.

*Usage Note:*

- The IDS.Attack_CmdType statement specifies the command environment.

*Related Statements:* IDS.Attack_CmdType
IDS.Attack_CmdType

The IDS.Attack_CmdType statement defines the command environment to be used when an IDS attack detection event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your system. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Attack_CmdType statement is:

```
IDS.Attack_CmdType.suffix=UNIX
IDS.Attack_CmdType.suffix=MVS
IDS.Attack_CmdType.suffix=NETV
```

Where:

**suffix**
A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, two command environments, UNIX and NETV, are defined.

IDS.Attack_CmdType.1 = UNIX
IDS.Attack_CmdType.2 = NETV

**Note:** Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Attack_CmdType.3 statement will be ignored:

IDS.Attack_CmdType.1 = UNIX
IDS.Attack_CmdType.2 = NETV
IDS.Attack_CmdType.3 = MVS

**MVS**
Indicates to issue an MVS command.

**NETV**
Indicates to issue a NetView command or command list.

**UNIX**
Indicates to issue a command to the UNIX® System Services using the PIPE UNIX command. This is the default setting.

**Usage Note:**
- The IDS.Attack_Cmd statements define the commands to be issued.

**Related Statements:** IDS.Attack_Cmd

IDS.Auto_Intvl

The IDS.Auto_Intvl statement defines the time interval to be used when defining IDS event threshold values. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Auto_Intvl statement is:

```
IDS.Auto_Intvl=hh:mm:ss
```

Customize CNMSTYLE 13
IDS.Auto_Intvl

Where:

hh:mm:ss
Indicates the interval in hours (24-hour clock), minutes, and seconds for an
IDS event threshold. The initial default value is 5 minutes, which a setting of
00:05:00 would provide.

Related Statements: IDS.Auto_Thresh

IDS.Auto_Thresh
The IDS.Auto_Thresh statement defines the automation table threshold value for
the number of IDS events. When the threshold value is exceeded, no event reports
are generated. Code this statement in CNMSTYLE %INCLUDE member
CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Auto_Thresh statement is:

IDS.Auto_Thresh

Where:

nnn
Indicates the threshold for the number of IDS events. The number must be in
the range of 1–999. The initial setting is 100.

Usage Note:
• The IDS.Auto_Thresh and IDS.Auto_Intvl statements set the IDS event threshold
used by the automation table. In the following example, if 101 or more IDS
events are detected within 5 minutes, no IDS event reports are generated:

IDS.Auto_Thresh = 100
IDS.Auto_Intlvl = 00:05:00

Related Statements: IDS.Auto_Intvl

IDS.ClearStat_Day
The IDS.ClearStat_Day statement defines when to clear the probe summary
statistics. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS.
Stop and restart the NetView program to implement the change.

The syntax for the IDS.ClearStat_Day statement is:

IDS.ClearStat_Day

Where:

ALL
Indicates that the probe summary statistics are to be cleared every day. This is
the default setting.
**IDS.ClearStat_Day**

*day*

Indicates that the probe summary statistics are to be cleared on one or more of the following days:

- ALL - every day
- MON - Monday
- TUE - Tuesday
- WED - Wednesday
- THU - Thursday
- FRI - Friday
- SAT - Saturday
- SUN - Sunday

**Usage Notes:**

- If you specify more than one day, separate the days with a blank space.
- Specify quotation marks before and after the days you specify.


---

**IDS.ClearStat_Inform**

The IDS.ClearStat_Inform statement defines the inform policies to be used when probe summary statistics are cleared. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.ClearStat_Inform statement is:

```
  IDS.ClearStat_Inform.suffix=inform_policy
```

**Where:**

*suffix*

A numerical suffix. Increment the remaining suffixes by one for each policy defined. In the example that follows, there is one inform policy (IDSOPERS) that is defined:

```
  IDS.ClearStat_Inform.1 = IDSOPERS
```

**Note:** Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

*inform_policy*

Specifies a predefined inform policy.

**Usage Note:**

- The *inform_policy* must be predefined in sample EZLINSMP. No checking is performed to verify that this policy is defined.

For information on defining inform policies, see the section on "Inform Policy Member" in the [IBM Tivoli NetView for z/OS Administration Reference](https://www.ibm.com/support/docview.wss?uid=swg21403980)

IDS.ClearStat_Log

The IDS.ClearStat_Log statement defines whether to log a report when the probe summary statistics are cleared. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.ClearStat_Log statement is:

**IDS.ClearStat_Log**

```
IDS.ClearStat_Log= N
```

Where:

N  Indicates that a report is not to be generated. This is the default setting.

Y  Indicates that a report is to be generated when the probe summary statistics are cleared.

Usage Note:
- The report is written to the DSIPARM data set member defined by the IDS.ClearStat_Log_File statement.


IDS.ClearStat_Log_File

The IDS.ClearStat_Log_File statement defines the DSIPARM data set member to which IDS probe summary statistics are written when the log is cleared. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.ClearStat_Log_File statement is:

**IDS.ClearStat_Log_File**

```
IDS.ClearStat_Log_File=member
```

Where:

**member**

Defines the DSIPARM data set member name. The initial setting is FXXSTATS.


IDS.ClearStat_Time

The IDS.ClearStat_Time statement defines when to clear probe summary statistics. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.ClearStat_Time statement is:
**IDS.ClearStat_Time**

```
IDS.ClearStat_Time
```

Where:

```
hh:mm
```

 Indicates the time of day in hours (24-hour clock) and minutes when the statistics are to be cleared. The initial default value is 00:00, indicating that the probe summary statistics are to be cleared at midnight.


**IDS.CONSOLEMSG**

The IDS.CONSOLEMSG statement specifies whether system console messages should be used as the source of events for IDS automation services. Code this statement in CNMSTYLE %.INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.CONSOLEMSG statement is:

```
IDS.CONSOLEMSG
```

Where:

```
Y   Selects system console messages as the source of events.
N   Specifies that system console messages are not used as the source of events.
```

Usage Notes:

- For IDS automation services to be active, you must set either IDS.CONSOLEMSG or IDS.SYSLOGMSG to Y.
- For performance reasons, do not set both IDS.CONSOLEMSG and IDS.SYSLOGMSG to Y.

Note: If you specify both IDS.CONSOLEMSG=Y and IDS.SYSLOGMSG=Y, neither system console messages or system log messages will be used.

Related Statements: IDS.SYSLOGMSG

**IDS.DSIPARM**

The IDS.DSIPARM statement defines the DSIPARM data set to which all TCP/IP Intrusion Detection Services (IDS) reports are written. Code this statement in CNMSTYLE %.INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.DSIPARM statement is:
IDS.DSIPARM

IDS.DSIPARM

IDS.DSIPARM

Where:

data_set

Defines the DSIPARM data set name. The IDSAUTO task must have write access to this data set.

IDS.Event_Inform

The IDS.Event_Inform statement defines the inform policies to be used when an IDS event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Event_Inform statement is:

IDS.Event_Inform

Where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each policy used. In the example that follows, there is one inform policy (IDSOPERS) that is defined:

IDS.Event_Inform.1 = IDSOPERS

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

inform_policy

Specifies a predefined inform policy.

Usage Note:

- The inform_policy must be predefined in sample EZLINSMP. No checking is performed to verify that this policy is defined.

For information on defining inform policies, see the section on "Inform Policy Member" in the IBM Tivoli NetView for z/OS Administration Reference.

IDS.Event_Limit

The IDS.Event_Limit statement defines the number of event reports to keep. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Event_Limit statement is:

IDS.Event_Limit

Where:
Specifies the number of event reports to keep. The number must be in the range of 1–99. The initial setting is 99.

**Related Statements:** IDS.Event_Log, IDS.Event_Log_File

### IDS.Event_Log

The IDS.Event_Log statement defines whether to log an event and associated commands and responses to a DSIPARM data set. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Event_Log statement is:

```
IDS.Event_Log
```  

```
N
Y
```

**Where:**

- **N** Indicates not to log the event and associated commands and responses. This is the default setting.
- **Y** Indicates to log the event, commands, and responses to the DSIPARM data set defined by the IDS.Event_Log_File statement.

**Related Statements:** IDS.Event_Log_File, IDS.Event_Limit

### IDS.Event_Log_File

The IDS.Event_Log_File statement defines the DSIPARM data set member to which IDS events and their associated commands and responses are written. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Event_Log_File statement is:

```
IDS.Event_Log_File
```  

```
member
```

**Where:**

- **member** Defines the DSIPARM data set member name. The name must be in the range of 1–6 characters. The initial setting is FXXIDS.

**Usage Note:**

- If event reporting is enabled, each report is logged to a DSIPARM member named FXXIDSnn (or to the member name specified in the IDS.Event_Log_File statement), where nn is determined by the setting of the IDS.Event_Limit statement.

**Related Statements:** IDS.Event_Log, IDS.Event_Limit
IDS.Flood_Cmd

IDS.Flood_Cmd
The IDS.Flood_Cmd statement specifies the command to be issued when an IDS attack detection flood event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Flood_Cmd statement is:

```
IDS.Flood_Cmd.suffix=command
```

Where:
suffix
A numerical suffix. Increment the remaining suffixes by one for each command used. In the example that follows, there is one command defined.

IDS.Flood_CmdType.1 = UNIX
IDS.Flood_Cmd.1 = /bin/trmdstat -F -D IDS.SYSLOG.FILENAME

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

command
Specifies the command to be issued.

Usage Note:
• The IDS.Flood_CmdType statement specifies the command environment.

Related Statements: IDS.Flood_CmdType

IDS.Flood_CmdType
The IDS.Flood_CmdType statement defines the command environment to be used when an IDS attack detection flood event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your system. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Flood_CmdType statement is:

```
IDS.Flood_CmdType.suffix=
```

Where:
suffix
A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, there is only one command environment (UNIX) defined.

IDS.Flood_CmdType.1 = UNIX

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

MVS
Indicates to issue an MVS command.
NETV
Indicates to issue a NetView command or command list.

UNIX
Indicates to issue a command to the UNIX System Services using the PIPE UNIX command.

Usage Note:
- Attack detection flood events have a probeid that begins with 0407.
- The IDS.Flood_Cmd statements define the commands to be issued.

Related Statements: IDS.Flood_Cmd

IDS.probeid
The IDS.probeid statement defines the supported IDS probeids. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.probeid statement is:

```
IDS.probeid.suffix=probeid
```

Where:

- `suffix`
  A numerical suffix. Increment the remaining suffixes by one for each probeid defined.

- `probeid`
  Specifies the hexadecimal value of the probeid.

Note: Do not leave gaps in the suffix numbers because a gap will cause any statements with a higher sequence number to be ignored.

Usage Note:
- The sample CNMSTIDS file contains the following note:

  Note: To ignore a probe id, add (NONE) to the end of the probeid without any blanks, e.g. 01002200 (NONE)
  Do NOT delete the probeid!

- If you add or delete probeids, renumber the suffix values.

  Note: Do not leave any gaps in the sequence numbers, because a gap will cause statements with a higher sequence number to be ignored.

IDS.Report_Cmd
The IDS.Report_Cmd statement specifies the command to be issued when the probe statistics report is generated. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Report_Cmd statement is:
IDS.Report_Cmd

Where:
suffix
A numerical suffix. Increment the remaining suffixes by one for each command used. In the example that follows, four commands are defined.

* IDS.Report_CmdType.1 = UNIX
  IDS.Report_Cmd.1 = /bin/trmdstat -I IDS.SYSLOG.FILENAME
* IDS.Report_CmdType.2 = UNIX
  IDS.Report_Cmd.2 = /bin/trmdstat -A -S IDS.SYSLOG.FILENAME
* IDS.Report_CmdType.3 = UNIX
  IDS.Report_Cmd.3 = /bin/trmdstat -U -S IDS.SYSLOG.FILENAME
* IDS.Report_CmdType.4 = UNIX
  IDS.Report_Cmd.4 = /bin/trmdstat -T -S IDS.SYSLOG.FILENAME

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

command
Specifies the command to be issued.

Usage Note:
• The IDS.Report_CmdType statement specifies the command environment.


IDS.Report_CmdType

The IDS.Report_CmdType statement defines the command environment to be used when the probe statistics report is generated. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your system. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Report_CmdType statement is:

---IDS.Report_CmdType_suffix=command---

Where:
suffix
A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, one command environment (UNIX) is defined.

IDS.Report_CmdType.1 = UNIX

Note: If you specify more than one command environment, do not leave gaps in the suffix numbers because such a gap would cause statements with a higher sequence number to be ignored.
IDS.Report_CmdType

MVS
Indicates to issue an MVS command.

NETV
Indicates to issue a NetView command or command list.

UNIX
Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage Note:
• The IDS.Report_Cmd statements define the commands to be issued.


IDS.Report_Inform
The IDS.Report_Inform statement defines the inform policies to be used when the probe statistics report is generated. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Report_Inform statement is:

```
IDS.Report_Inform.suffix=inform_policy
```

Where:

suffix
A numerical suffix. Increment the remaining suffixes by one for each policy used. In the example that follows, there is one inform policy (IDSOPERS) defined:

```
IDS.Report_Inform.1 = IDSOPERS
```

Note: If you specify more than one inform policy, do not leave gaps in the suffix numbers because such a gap would cause statements with a higher sequence number to be ignored.

inform_policy
Specifies a predefined inform policy.

Usage Note:
• The inform_policy must be predefined in sample EZLINSMP. No checking is performed to verify that this policy is defined. For information on defining inform policies, see the section on "Inform Policy Member" in the IBM Tivoli NetView for z/OS Administration Reference


IDS.Report_Log
The IDS.Report_Log statement defines whether to log a timer-generated IDS probe statistics report to the DSIPARM data set member defined by the
IDS.Report_Log

IDS.Report_Log_File statement. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Report_Log statement is:

IDS.Report_Log

Where:
N Indicates not to log the report. This is the default setting.
Y Indicates to log the report to the DSIPARM data set.


IDS.Report_Log_File

The IDS.Report_Log_File statement defines the DSIPARM data set to which the IDS probe statistics reports are written. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Report_Log_File statement is:

IDS.Report_Log_File

Where:
member Defines the DSIPARM data set member name. The initial setting is FXXREPO.


IDS.Scan_Cmd

The IDS.Scan_Cmd statement specifies the command to be issued when an IDS scan detection event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Scan_Cmd statement is:

IDS.Scan_Cmd

Where:
suffix
A numerical suffix. Increment the remaining suffixes by one for each command used. In the example that follows, one command is defined.

IDS.Scan_CmdType.1 = UNIX
IDS.Scan_Cmd.1 = /bin/trmdstat -N -D IDS.SYSLOG.FILENAME

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

command
Specifies the command to be issued.

Usage Note:
• The IDS.Scan_CmdType statement specifies the command environment.

Related Statements: IDS.Scan_CmdType

IDS.Scan_CmdType
The IDS.Scan_CmdType statement defines the command environments to be used when an IDS scan detection event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your system. Stop and restart the NetView program to implement the change.

The syntax for the IDS.Scan_CmdType statement is:

```
IDS.Scan_CmdType.suffix = UNIX
```

Where:

suffix
A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, one command is defined.

IDS.Scan_CmdType.1 = UNIX

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

MVS
Indicates to issue an MVS command.

NETV
Indicates to issue a NetView command or command list.

UNIX
Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage Note:
• The IDS.Scan_Cmd statement defines the command to be issued.

Related Statements: IDS.Scan_Cmd
IDS.SYSLOG.FILENAME

The IDS.SYSLOG.FILENAME statement defines the UNIX System Services file to which IDS automation services is listening. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

**Note:** Specify the IDS.SYSLOG.FILENAME statement only if IDS.SYSLOGMSG=Y.

The syntax for the IDS.SYSLOG.FILENAME statement is:

```
IDS.SYSLOG.FILENAME
```

Where:

**filename**

Defines the fully-qualified UNIX System Services file name. You can use the following shell variables in the file name:

**Table 1. Supported Variables**

<table>
<thead>
<tr>
<th>Shell Variable</th>
<th>Definition</th>
<th>Substitution</th>
</tr>
</thead>
<tbody>
<tr>
<td>%D</td>
<td>Day</td>
<td>dd</td>
</tr>
<tr>
<td>%M</td>
<td>Month</td>
<td>mm</td>
</tr>
<tr>
<td>%Y</td>
<td>Year</td>
<td>yyyy</td>
</tr>
</tbody>
</table>

Usage Note:

- Changes to the UNIX System Services syslog file, including clearing it, should be synchronized with the NetView program. The UNIX System Services syslog is generally updated using the *cron* command. You must recycle the NetView program after any updates have been made to the syslog file name.

IDS.SYSLOGMSG

The IDS.SYSLOGMSG statement defines the system log as the source of events for IDS automation services. Code this statement in CNMSTYLE %INCLUDE member CNMSTIDS. Stop and restart the NetView program to implement the change.

The syntax for the IDS.SYSLOGMSG statement is:

```
IDS.SYSLOGMSG
```

Where:

**Y** Selects the system log as the source of events.

**N** Specifies not to use the system log as the source of events. This is the default setting.

Usage Notes:
For IDS automation services to be active, you must set either
IDS.CONSOLEMSG or IDS.SYSLOGMSG to Y.
For performance reasons, do not set both IDS.CONSOLEMSG and
IDS.SYSLOGMSG to Y.

Note: If you specify both IDS.CONSOLEMSG=Y and IDS.SYSLOGMSG=Y,
neither system console messages or system log messages will be used.

Related Statements: IDS.CONSOLEMSG

**IDS.TCP_Cmd**
The IDS.TCP_Cmd statement specifies the command to be issued when an IDS
TCP traffic regulation event occurs. Code as many of these statements in
CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment.
Stop and restart the NetView program to implement the change.

The syntax for the IDS.TCP_Cmd statement is:

```plaintext
IDS.TCP_Cmd.suffix=command
```

Where:

* **suffix**
  A numerical suffix. Increment the remaining suffixes by one for each command
  used. In the example that follows, there is one command defined.
  ```plaintext
  IDS.TCP_CmdType.1 = UNIX
  IDS.TCP_Cmd.1 = /bin/trmdstat -T -D IDS.SYSLOG.FILENAME
  ```
  Note: Do not leave gaps in the suffix numbers because this gap causes
  statements with a higher sequence number to be ignored.

* **command**
  Specifies the command to be issued.

Usage Note:
* The IDS.TCP_CmdType statement specifies the command environment.

Related Statements: IDS.TCP_CmdType

**IDS.TCP_CmdType**
The IDS.TCP_CmdType statement defines the command environment used when
an IDS TCP traffic regulation event occurs. Code as many of these statements in
CNMSTYLE %INCLUDE member CNMSTIDS as needed for your system. Stop
and restart the NetView program to implement the change.

The syntax for the IDS.TCP_CmdType statement is:

```plaintext
IDS.TCP_CmdType.suffix=netv
```

Where:
IDS.TCP_CmdType

suffix

A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, there is one command environment (UNIX) defined.

IDS.TCP_CmdType.1 = UNIX

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

MVS
Indicates to issue an MVS command.

NETV
Indicates to issue a NetView command or command list.

UNIX
Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage Note:
• The IDS.TCP_Cmd statements define the commands to be issued.

Related Statements: IDS.TCP_Cmd

IDS.UDP_Cmd

The IDS.UDP_Cmd statement specifies the command to be issued when an IDS UDP traffic regulation event occurs. Code as many of these statements in CNMSTYLE %INCLUDE member CNMSTIDS as needed for your environment. Stop and restart the NetView program to implement the change.

The syntax for the IDS.UDP_Cmd statement is:

```
IDS.UDP_Cmd.suffix=command
```

Where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command defined. In the example that follows, there is one command defined.

IDS.UDP_CmdType.1 = UNIX
IDS.UDP_Cmd.1 = /bin/trmdstat -U -D IDS.SYSLOG.FILENAME

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

command

Specifies the command to be issued.

Usage Note:
• The IDS.UDP_CmdType statement specifies the command environment.

Related Statements: IDS.UDP_CmdType

IDS.UDP_CmdType

The IDS.UDP_CmdType statement defines the command environment used when an IDS UDP traffic regulation event occurs. Code as many of these statements in
CNMSTYLE %INCLUDE member CNMSTIDS as needed for your system. Stop and restart the NetView program to implement the change.

The syntax for the IDS.UDP_CmdType statement is:

```
IDS.UDP_CmdType.suffix=UNIX
```

Where:

- `suffix` A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, there is one command environment (UNIX) defined.
  
  ```
  IDS.UDP_CmdType.1 = UNIX
  ```

  **Note:** Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

- `MVS` Indicates to issue an MVS command.
- `NETV` Indicates to issue a NetView command or command list.
- `UNIX` Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

**Usage Note:**
- The IDS.UPD_Cmd statements define the commands to be issued.

**Related Statements:** IDS.UPD_Cmd

### MSG.TECROUTE

The MSG.TECROUTE statement specifies the name of the PPI receiver associated with the event/automation service. This statement is coded in CNMSTYLE %INCLUDE member CNMSTIDS.

The syntax for the MSG.TECROUTE statement is:

```
MSG.TECROUTE=name
```

Where:

- `name` The PPI receiver name. The default value is IHSATEC.
Commands specific to IDS

These are NetView commands that either are new or have been changed to support Intrusion Detection Services.

**Note:** For online information on any NetView command, enter the following command:
```
HELP cmd
```

where *cmd* is the name of the command.
FKXEIDSC

Syntax

```
FKXEIDSC — sp_name
```

Purpose of Command
The FKXEIDSC command generates a Probe Summary Statistics report and then clears the existing statistics kept for each probe (since the last time they were cleared) for the specified service point.

Operand Descriptions

`sp_name`

The name of the service point as defined on a TCP390 policy definition.

Usage Notes

The IDS.ClearStat statements in CNMSTYLE %INCLUDE member CNMSTIDS define inform policies, time interval, and logging actions to be used when clearing the probe summary statistics.

Examples

To generate a Probe Summary Statistics report for the NMPANY10 service point and reset the statistics being kept, enter the following command:

```
FKXEIDSC NMPANY10
```

The IDS.ClearStat_Log statements in CNMSTIDS determine if report data is to be logged, and if so, where it is to be logged. The IDS.ClearStat_Inform statements determine the inform policies that are invoked when the probe summary statistics are cleared. The IDS.ClearStat_Day and IDS.ClearStat_Time statements determine when to clear the statistics.

If the IDS.ClearStat_Inform statements have been properly configured, the following example message is sent to the operator console: EZL460I EMAIL ACTION WAS SUCCESSFULLY ISSUED FOR POLICY DAYOPS BY OPERATOR OPER1

This message indicates that report data was logged, the statistics were reset, and an e-mail was sent to DAYOPS.
FKXEIDSR

Syntax

FKXEIDSR — sp_name

Purpose of Command

The FKXEIDSR command generates a Probe Summary Statistics report for the specified service point.

Operand Descriptions

sp_name

The name of the service point as defined on a TCP390 policy definition.

Usage Notes

The IDS.Report statements in CNMSTYLE %INCLUDE member CNMSTIDS define commands and command types as well as inform policies to be issued when the Probe Summary Statistics report is generated.

Examples

To generate a Probe Summary Statistics report for the NMPANY10 service point, enter the following command:

FKXEIDSR NMPANY10

The IDS.Report_Log_File statements in CNMSTIDS determine if report data is to be logged, and if so, where it is to be logged. The IDS.Report_Inform statements determine the inform policies that are invoked when the report is generated.

If the IDS.Report_Inform statements have been properly configured, the following example message is sent to the operator console: EZL460I EMAIL ACTION WAS SUCCESSFULLY ISSUED FOR POLICY DAYOPS BY OPERATOR OPER1

This message indicates that report data was logged and an e-mail was sent to DAYOPS.
Messages

This section describes NetView messages that are new or changed for the Intrusion Detection Automated Services.

For online help on messages, enter the command that follows.
HELP msgid

where msgid is the identifier of the NetView message for which a help panel is to be displayed.

BNH180I INTRUSION DETECTION MESSAGE RECEIVED.

Explanation: An Intrusion Detection Services (IDS) message has been forwarded from the UNIX System Services syslog. This message is the first line of an MLTWO. The next line of the message contains the complete text of the syslog message, including the syslog header. This message is used by NetView automation.

System Action: Command processing continues.

BNH193I ERROR MONITORING SYSLOG 'filename'

Explanation: NetView has encountered an error attempting to receive updates from the specified USS syslog file. Possible problems fall into these general areas: problems with the NetView Unix pipe stage, problems with the OMVS tail command, or problems with the format of a syslog record. Additional messages accompany message BNH193 that further identify the problem.

Message Variables:
filename The fully qualified name of the USS syslog file that is being monitored for updates.

System Action: For Unix pipe stage or tail command errors, NetView stops monitoring the file. For record format errors, processing continues, but the invalid syslog entry is ignored and logged.

Operator Response: Notify the system programmer.

System Programmer Response: Examine the additional messages that accompany BNH193 to determine the cause of the problem.

DSI665I KEYWORD CONFLICT BETWEEN keyword1 AND keyword2. KEYWORD keyword2 IGNORED

Explanation: Keywords keyword1 and keyword2 can not be specified in the same invocation of the command.

Message Variables:
keyword1 The first keyword specified.
keyword2 The second keyword specified.

System Action: The NetView program ignores keyword2. Keyword1 is processed normally.

Related Commands: AUTOTEST, START, IDS automation services

FKX300I IDS EVENT RECEIVED : DIPADDR=dest_ip_addr, SIPADDR=src_ip_addr, CORRELATOR=nnnn. MEMBER=member_name

Explanation: This message indicates that an Intrusion Detection Automation Services (IDS) event has been processed by IDS automation services. One or more actions could have occurred. The responses to those actions are listed in DSIPARM member member_name. If member_name is NONE, there is no data to view.

Message Variables:
**dest_ip_addr**
The destination IP address for the IDS event as determined from the message that started IDS automation services. If the dest_ip_addr is UNKNOWN, the destination IP address could not be determined.

**src_ip_addr**
The source IP address

**nmmm** The IDS correlator associated with the event

**member_name**
The name of the DSIPARM member containing the command responses. This name is listed in the following CNMSTYLE definition:

```plaintext
IDS.Event_Log_File = DSIPARM.member_name
```

If NONE is specified, event logging is not enabled.

If **ERROR** is specified, no data could be written to the file. For more information, browse DSILog for message CNM236.

**System Action:** Processing continues.

**Operator Response:** For additional information on the IDS event, browse DSIPARM member member_name.

**System Programmer Response:** If necessary, grant the operator access to browse DSIPARM member member_name.

---

**FKX301I** IDS EVENT THRESHOLD OF nnn EVENTS DURING INTERVAL interval REACHED FOR STACK stackname

**Explanation:** There have been a number of IDS events detected within the system for which the threshold defined in CNMSTYLE %INCLUDE member CNMSTIDS has been matched or exceeded. This message is sent to the IDS NTFYOPs.

**Message Variables:**
- **nnn** The threshold for the number of IDS events as defined in the following CNMSTIDS statement:
  ```plaintext
  IDS.Auto_Thresh = nnn
  ```
- **interval** The time interval used to determine the threshold as defined in the following CNMSTIDS statement:
  ```plaintext
  IDS.Auto_Interval = hh:mm:ss
  ```
- **stackname** The name of the stack as defined on a TCP390 policy definition.

**System Action:** IDS events are ignored until enough time has passed to no longer trigger this threshold.

**System Programmer Response:** Check your CNMSTYLE definitions and update as needed.

---

**FKX303I** IDS EVENT DETECTED CONTAINS UNKNOWN probeid

**Explanation:** An IDS event was sent to IDS automation services. However, IDS automation services did not recognize the probeid contained in the event. The event containing the unknown probeid is logged immediately following this message.

**Message Variables:**
- **probeid** The probeid from the IDS event.

**System Action:** The event is discarded.

**System Programmer Response:** Browse CNMSTIDS and ensure that the probeid is defined in the IDS.PROBEID.* list.
**FKX305I**  
IDS EVENT DETECTED FOR DESTINATION IP ADDRESS *dest_ip_addr* BUT COULD NOT CORRELATE TO A KNOWN TCP/IP STACK.

**Explanation:** An IDS event was sent to IDS automation services. However, IDS automation services was unable to determine the TCP/IP stack associated with the destination IP address.

**Message Variables:**

*dest_ip_addr*  
The destination IP address for the IDS event as determined from the message that started IDS automation services. If the *dest_ip_addr* is UNKNOWN, the destination IP address could not be determined.

**System Action:** Processing continues. Some data (such as Summary Probe Statistics) might not include this event.

**System Programmer Response:** For more information related to the event, browse DSILog for EZZ* messages.

**Note:** The IP Messages Volume 4 (SC31–8786) contains detailed information about EZZ* messages.
Contacting Customer Support

If you have a problem with any Tivoli product, refer to the following IBM® Software Support Web site:


If you want to contact software support, see the IBM Software Support Guide at the following Web site:

http://techsupport.services.ibm.com/guides/handbook.html

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

• Registration and eligibility
• Telephone numbers and e-mail addresses, depending on the country in which you are located
• Information you must have before contacting IBM Software Support
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX  78758  U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to
IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

Trademarks

IBM, the IBM logo, Tivoli, the Tivoli logo, MVS™, NetView, and z/OS are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.