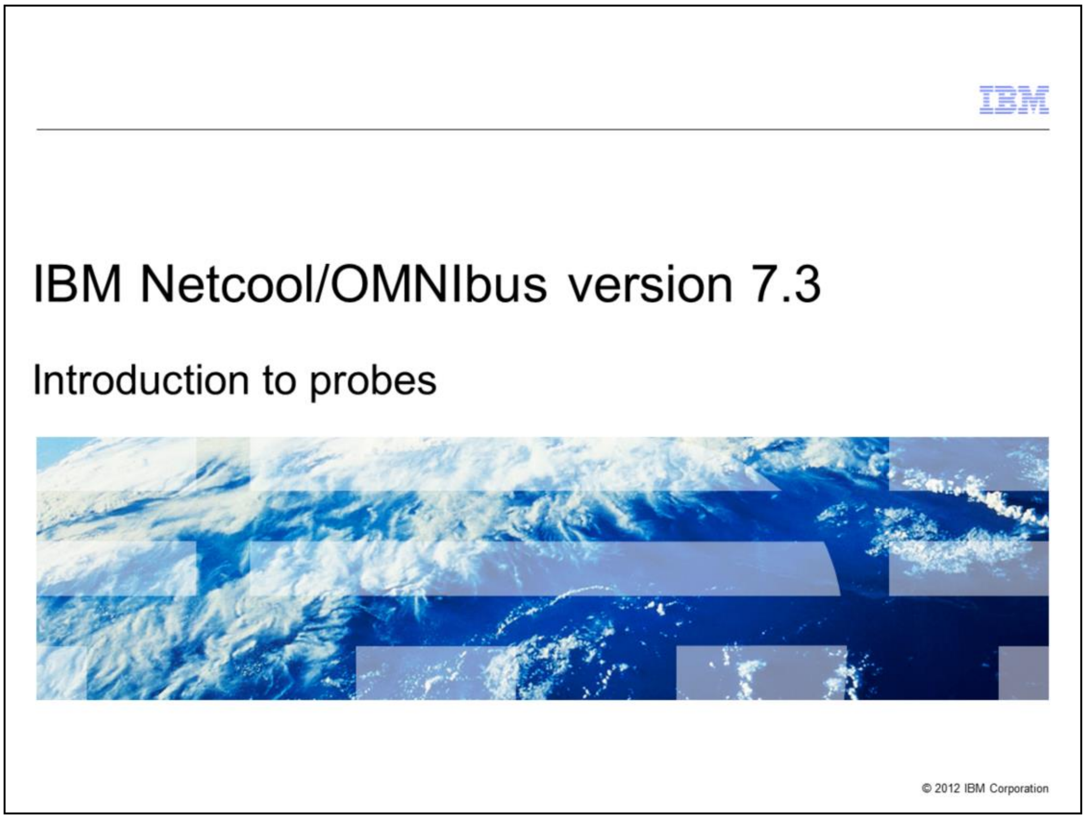
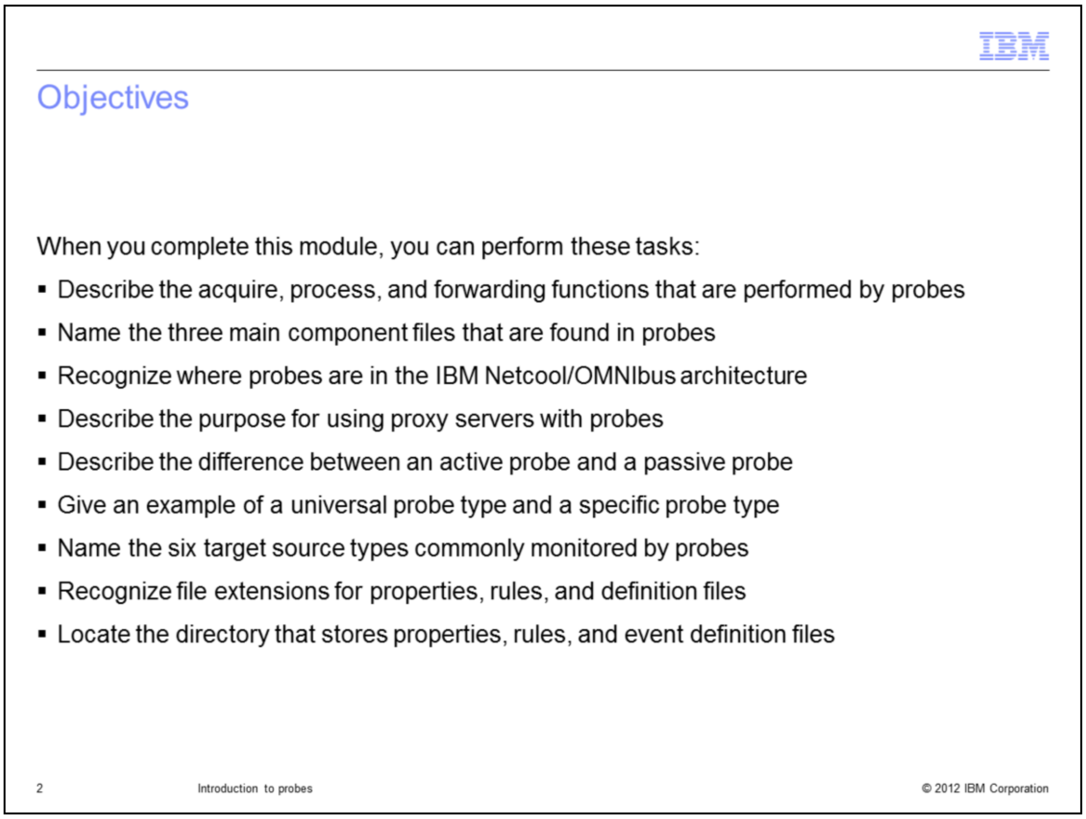


IBM Netcool/OMNIbus version 7.3

Introduction to probes

This training module provides an introduction to IBM Netcool/OMNIbus version 7.3 probes.



Objectives

When you complete this module, you can perform these tasks:

- Describe the acquire, process, and forwarding functions that are performed by probes
- Name the three main component files that are found in probes
- Recognize where probes are in the IBM Netcool/OMNIbus architecture
- Describe the purpose for using proxy servers with probes
- Describe the difference between an active probe and a passive probe
- Give an example of a universal probe type and a specific probe type
- Name the six target source types commonly monitored by probes
- Recognize file extensions for properties, rules, and definition files
- Locate the directory that stores properties, rules, and event definition files

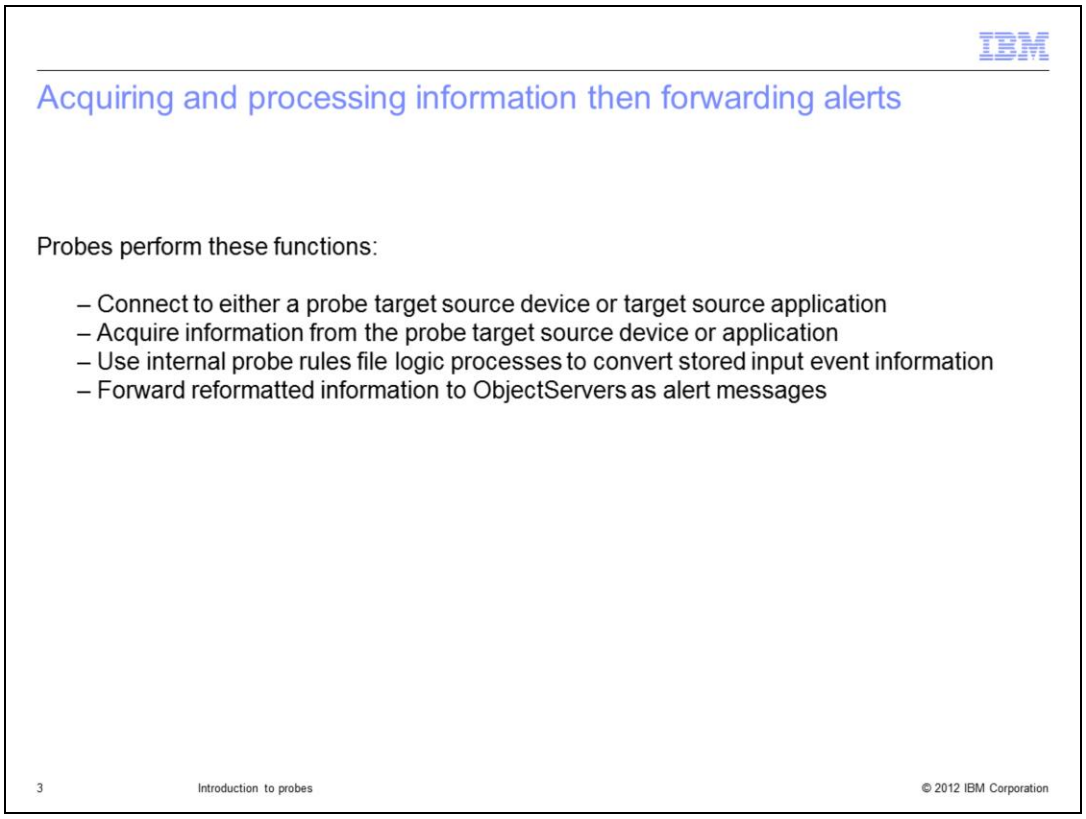
2 Introduction to probes © 2012 IBM Corporation

When you complete this module, you can perform these tasks:

Understand probe acquisition, processing, and forwarding. Name the three main IBM Netcool/OMNIbus probe files. Place probes logically within a IBM Netcool/OMNIbus architecture diagram.

Understand the purpose for using proxy servers. Describe the difference between active and passive probes. Give an example of both a universal probe and a specific probe.

Name six common probe target source types. Recognize file extensions for and locate the directory for properties, rules, and event definition files.



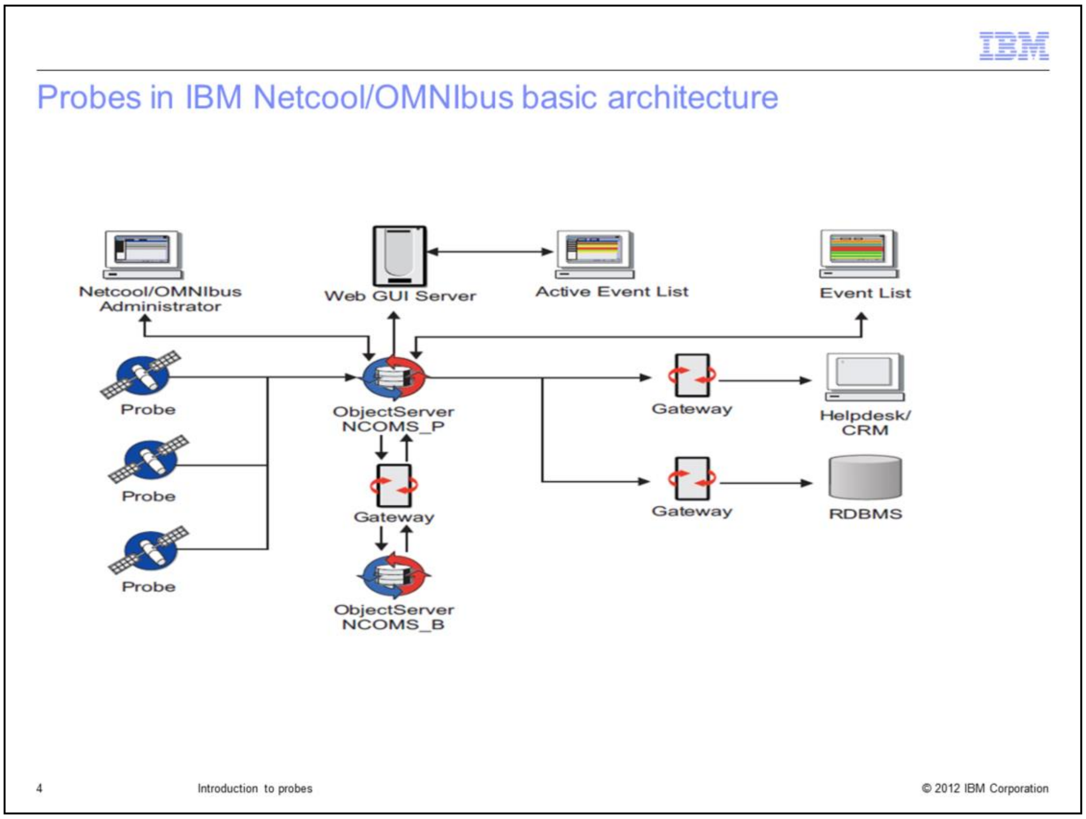
Acquiring and processing information then forwarding alerts

Probes perform these functions:

- Connect to either a probe target source device or target source application
- Acquire information from the probe target source device or application
- Use internal probe rules file logic processes to convert stored input event information
- Forward reformatted information to ObjectServers as alert messages

3 Introduction to probes © 2012 IBM Corporation

IBM Netcool/OMNIbus probes perform four basic functions. Probes first connect to target devices and target applications and then acquire event information. The probes are configured to process the event information. After they process the information, they forward the event information to another IBM Netcool/OMNIbus component called the ObjectServer.



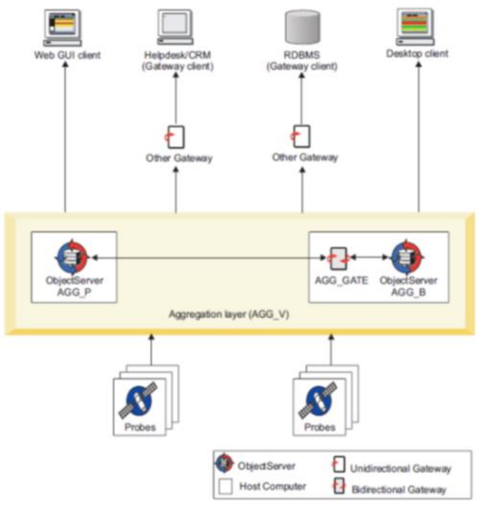
Probes in IBM Netcool/OMNIbus basic architecture

4 Introduction to probes © 2012 IBM Corporation

On this slide, you can see where probes are located within the basic IBM Netcool/OMNIbus architecture. Notice the unidirectional interconnection between probes and ObjectServers.

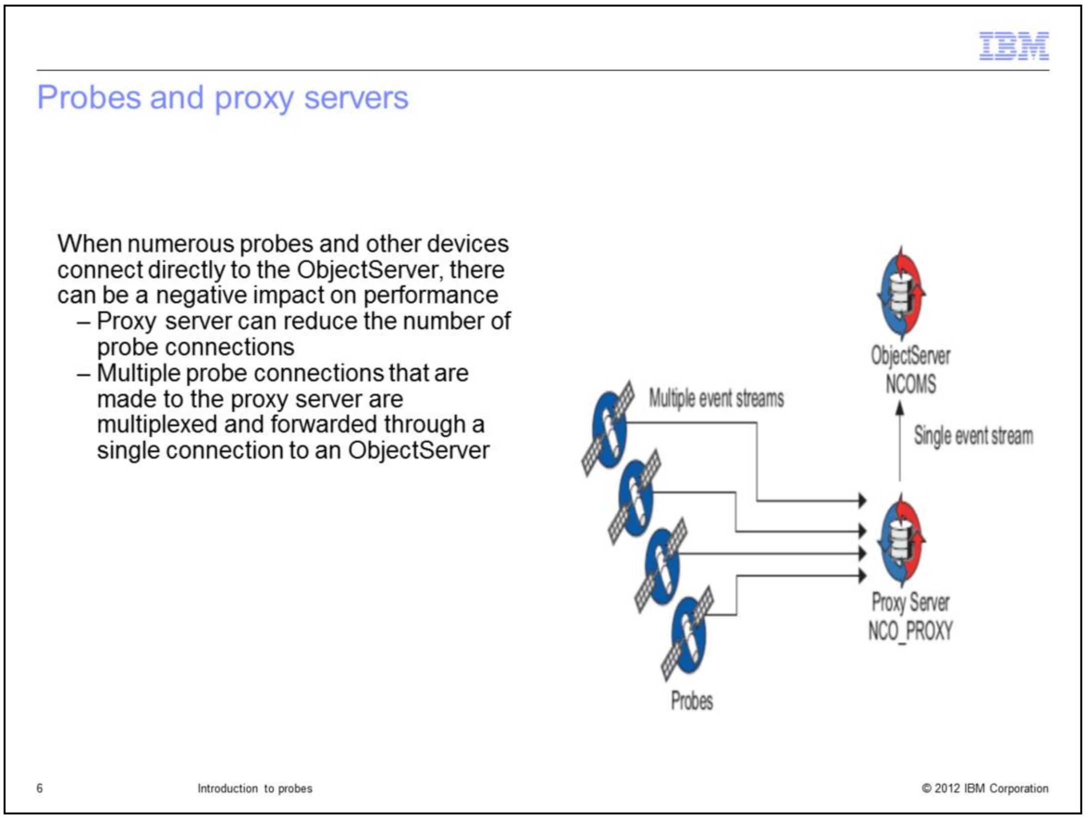
Probes in IBM Netcool/OMNIbus basic failover architecture

In failover architecture, you configure the ObjectServer by using a primary and backup failover pair



5 Introduction to probes © 2012 IBM Corporation

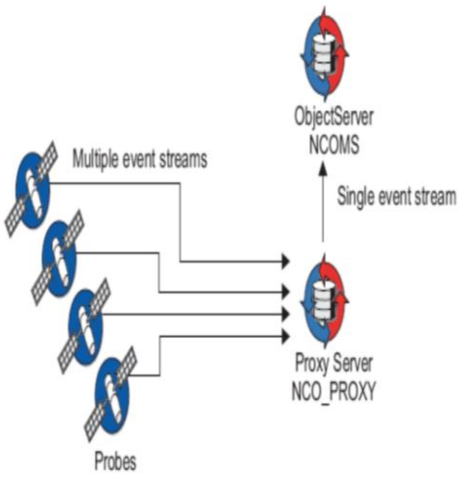
For high availability, you can implement the ObjectServer in pairs. Each ObjectServer in the pair receives a connection to network probes. Probes forward identical alert messages to each ObjectServer in the pair.



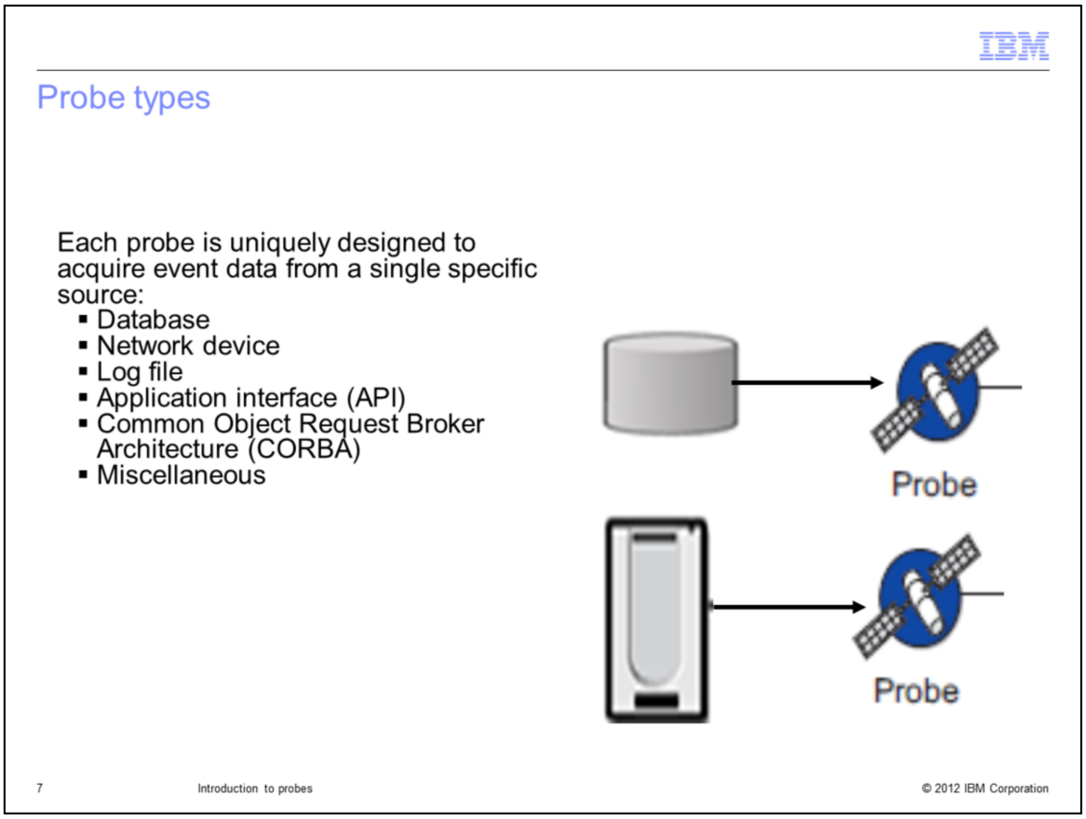
Probes and proxy servers

When numerous probes and other devices connect directly to the ObjectServer, there can be a negative impact on performance

- Proxy server can reduce the number of probe connections
- Multiple probe connections that are made to the proxy server are multiplexed and forwarded through a single connection to an ObjectServer



Alerts are typically forwarded directly to an ObjectServer. For high-volume probe architecture, a proxy server can be used. Proxy servers take some of the interconnection workload away from ObjectServers.



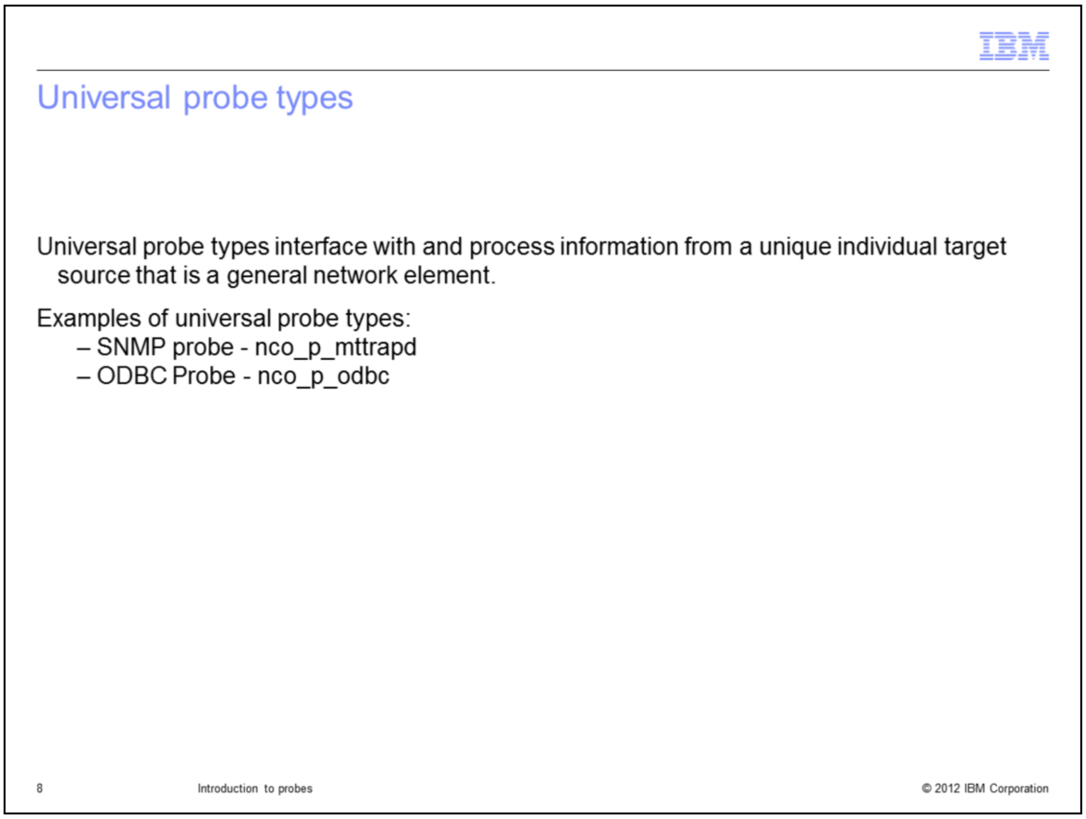
Probe types

Each probe is uniquely designed to acquire event data from a single specific source:

- Database
- Network device
- Log file
- Application interface (API)
- Common Object Request Broker Architecture (CORBA)
- Miscellaneous

Probe
Probe

Probes connect to event sources to acquire event data. The data is passed to ObjectServers in the form of alerts. Probes use function logic specified in a rules files to map the received data into the format of the event or alert. Every probe type is uniquely designed for monitoring a particular type of database, application, log file, or network device. Probes can be categorized based on how they acquire events. The types of probes are: device, log file, database, API, CORBA, and miscellaneous. The probe type is determined by the method in which the probe detects events. Two examples are Probe for Agile ATM Switch Management and Probe for Oracle.



Universal probe types

Universal probe types interface with and process information from a unique individual target source that is a general network element.

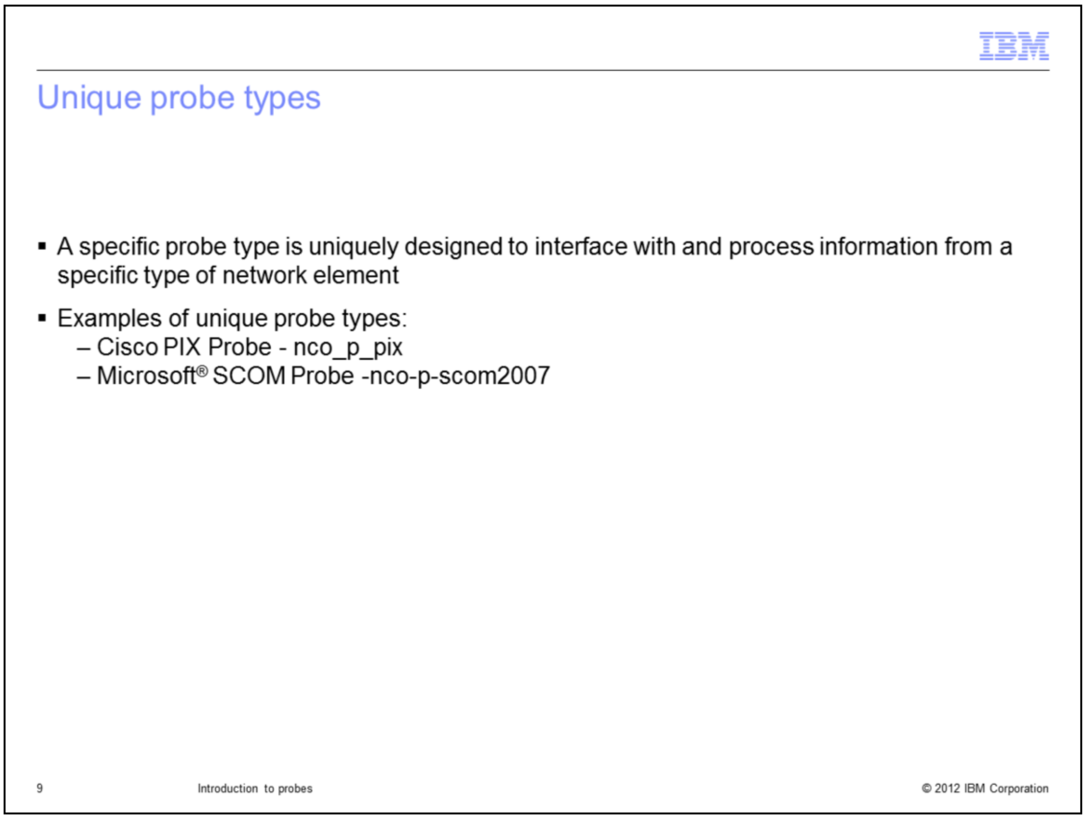
Examples of universal probe types:

– SNMP probe - nco_p_mttrapd

– ODBC Probe - nco_p_odbc

8 Introduction to probes © 2012 IBM Corporation

A universal probe type is designed to interface with and process information from a general network element type. Two examples of universal probes types are the SNMP probe and the Open Database Connectivity probe. The SNMP probe is compatible with source devices that create SNMP standards-compliant MIBs. Likewise, the OBDC probe is compatible with ODBC standards-compliant database connectivity devices.

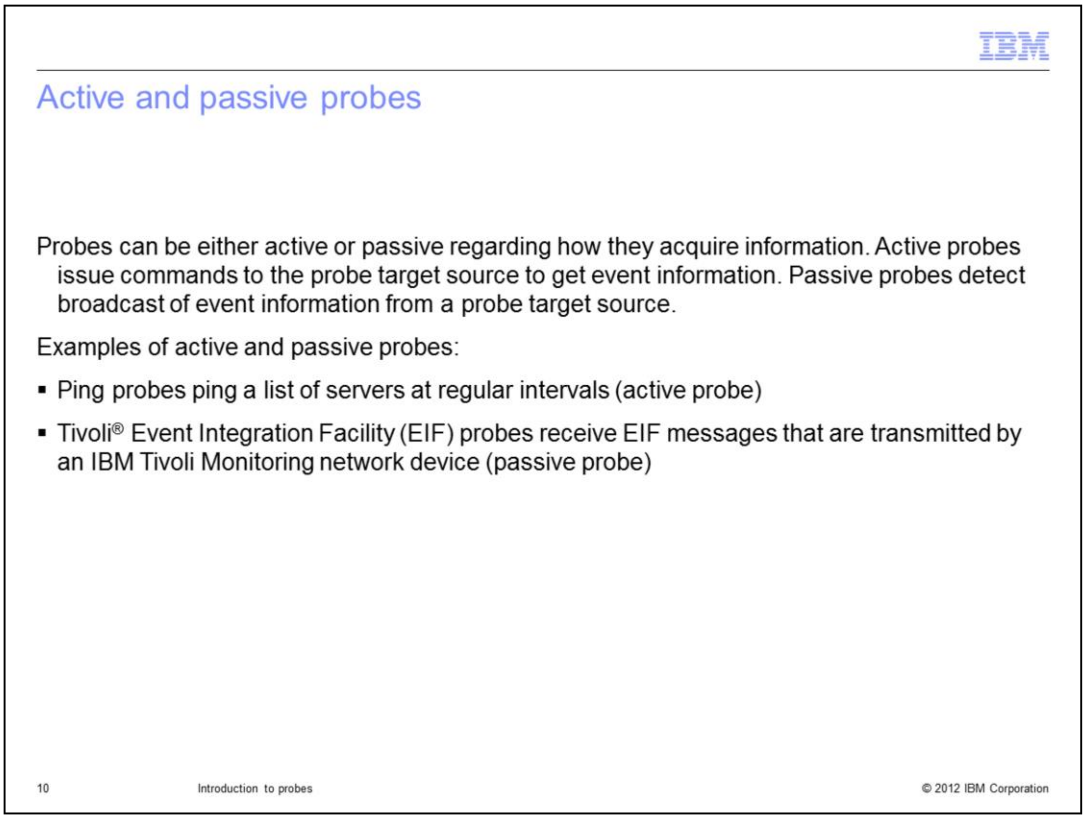


Unique probe types

- A specific probe type is uniquely designed to interface with and process information from a specific type of network element
- Examples of unique probe types:
  - Cisco PIX Probe - nco_p_pix
  - Microsoft® SCOM Probe -nco-p-scom2007

9 Introduction to probes © 2012 IBM Corporation

A specific probe type is uniquely designed to interface with and process information from a specific type of network element. Two examples of specific probes types are the Cisco PIX probe and the Microsoft SCOM probe.



Active and passive probes

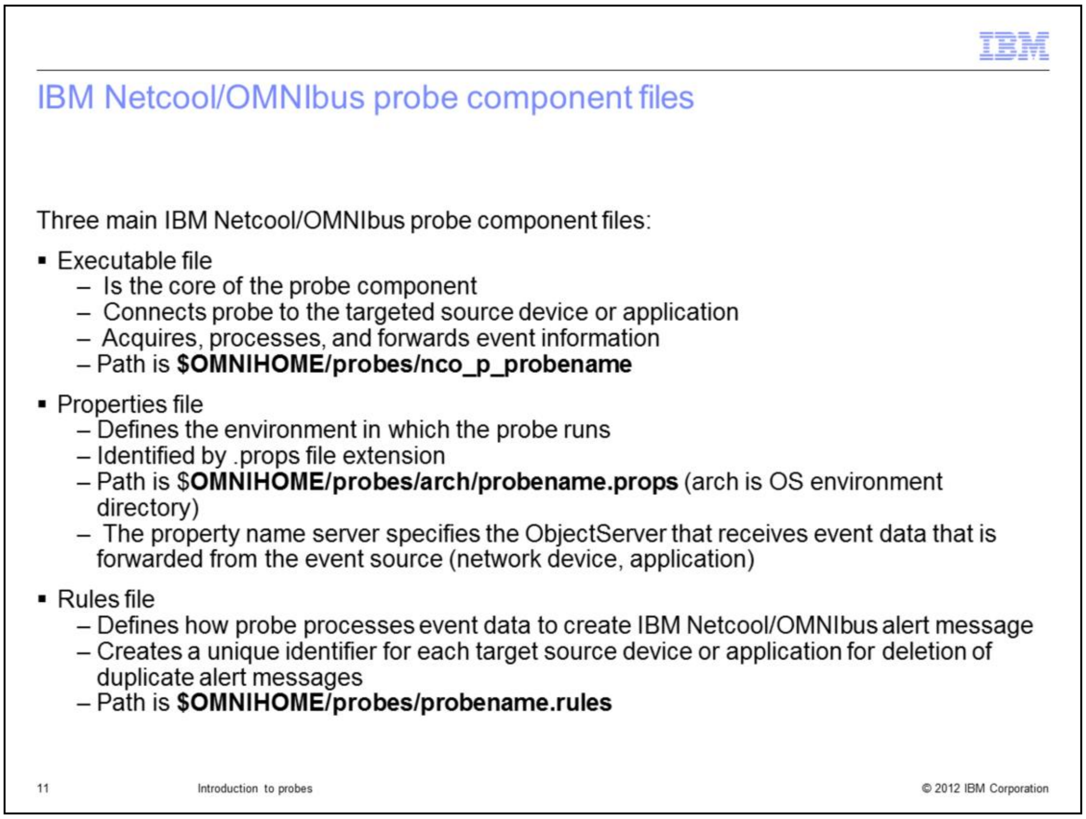
Probes can be either active or passive regarding how they acquire information. Active probes issue commands to the probe target source to get event information. Passive probes detect broadcast of event information from a probe target source.

Examples of active and passive probes:

- Ping probes ping a list of servers at regular intervals (active probe)
- Tivoli® Event Integration Facility (EIF) probes receive EIF messages that are transmitted by an IBM Tivoli Monitoring network device (passive probe)

10 Introduction to probes © 2012 IBM Corporation

Probes are either active or passive. Active probes issue commands to trigger the pulling of event information from source targets. Passive probes wait for and detect source target device broadcast of event messages. A Ping probe is an example of an active probe. Ping probes are configured to ping servers at regular intervals to ensure connectivity. Server availability is reported by the Ping probe to an ObjectServer. A Tivoli EIF probe is an example of a passive probe. The Tivoli EIF probe awaits transmission of EIF messages from source network devices.



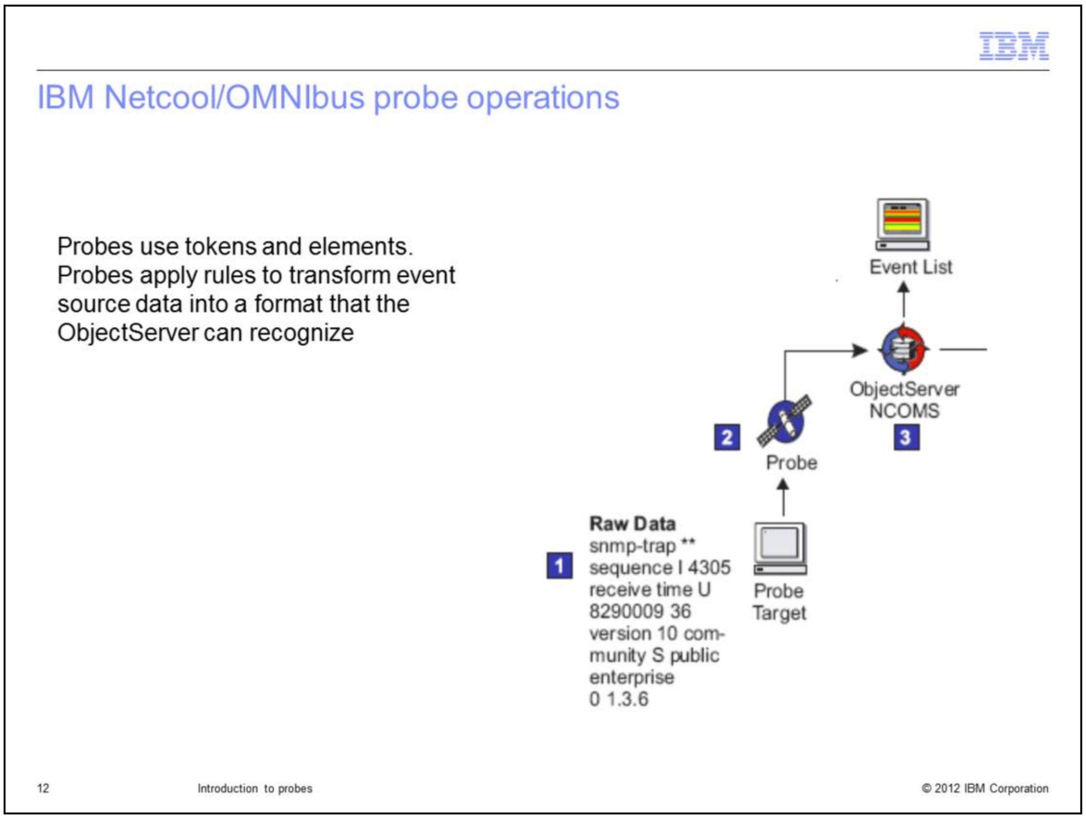
IBM Netcool/OMNIbus probe component files

Three main IBM Netcool/OMNIbus probe component files:

- Executable file
  - Is the core of the probe component
  - Connects probe to the targeted source device or application
  - Acquires, processes, and forwards event information
  - Path is **$OMNIHOME/probes/nco_p_probename**
- Properties file
  - Defines the environment in which the probe runs
  - Identified by .props file extension
  - Path is $**OMNIHOME/probes/arch/probename.props** (arch is OS environment directory)
  - The property name server specifies the ObjectServer that receives event data that is forwarded from the event source (network device, application)
- Rules file
  - Defines how probe processes event data to create IBM Netcool/OMNIbus alert message
  - Creates a unique identifier for each target source device or application for deletion of duplicate alert messages
  - Path is **$OMNIHOME/probes/probename.rules**

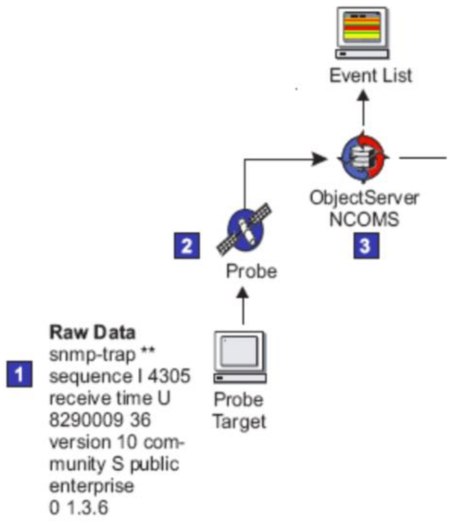
11 Introduction to probes © 2012 IBM Corporation

IBM Netcool/OMNIbus probes typically consist of an executable file, properties file, and rules file. The executable file is the core of the probe. It connects to the event source, processes stored data, and forwards the events to either ObjectServers or ProxyServers as alerts. The properties file specifies the destination ObjectServer that the probe forwards alerts to. The rules file creates a unique identifier for the destination ObjectServer. It also maps event information into the alert message format accepted by ObjectServers.



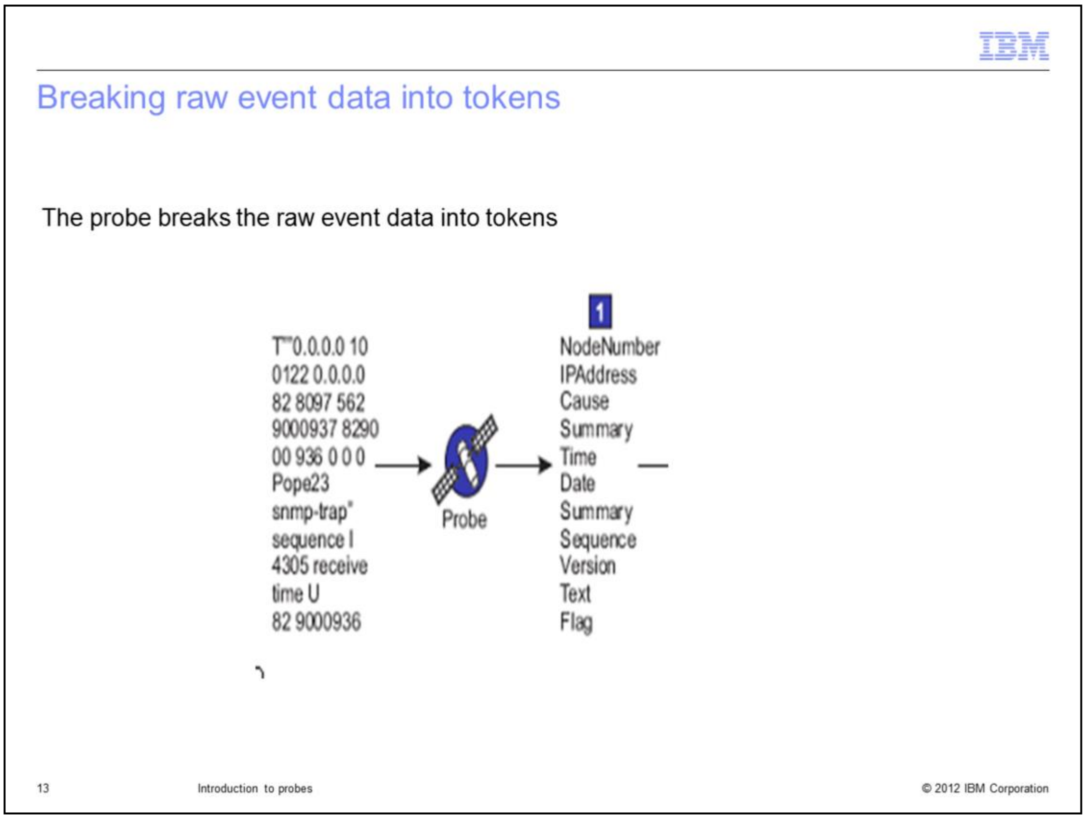
IBM Netcool/OMNIbus probe operations

Probes use tokens and elements. Probes apply rules to transform event source data into a format that the ObjectServer can recognize



12 Introduction to probes © 2012 IBM Corporation

Probes use tokens and elements when applying rules. These rules convert event information into a format that is recognized and accepted by ObjectServers. The ObjectServer is an in-memory database server at the core of the IBM Netcool/OMNIbus system.



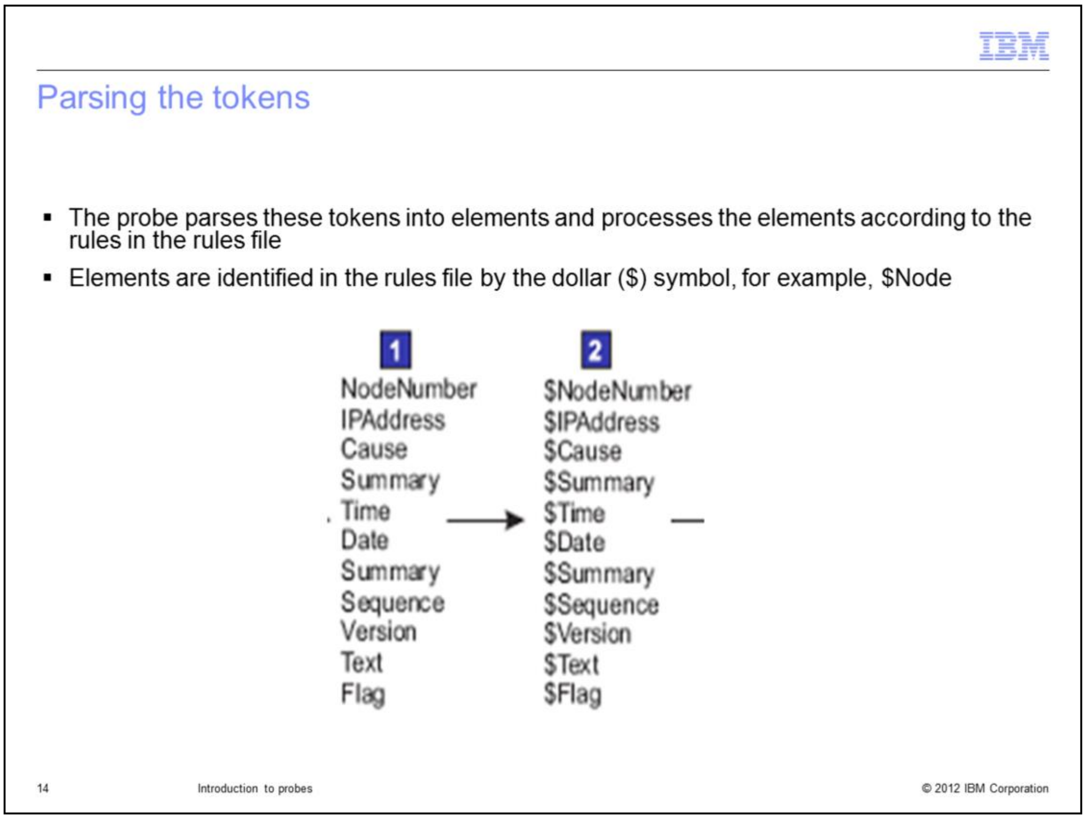
Breaking raw event data into tokens

The probe breaks the raw event data into tokens

T"0.0.0.0 10
0122 0.0.0.0
82 8097 562
9000937 8290
00 936 0 0 0
Pope23
snmp-trap"
sequence I
4305 receive
time U
82 9000936
Probe
1
NodeNumber
IPAddress
Cause
Summary
Time
Date
Summary
Sequence
Version
Text
Flag

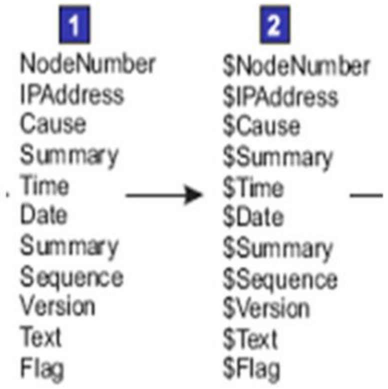
13 Introduction to probes © 2012 IBM Corporation

In this slide, raw target source information is mapped into organized tokens.



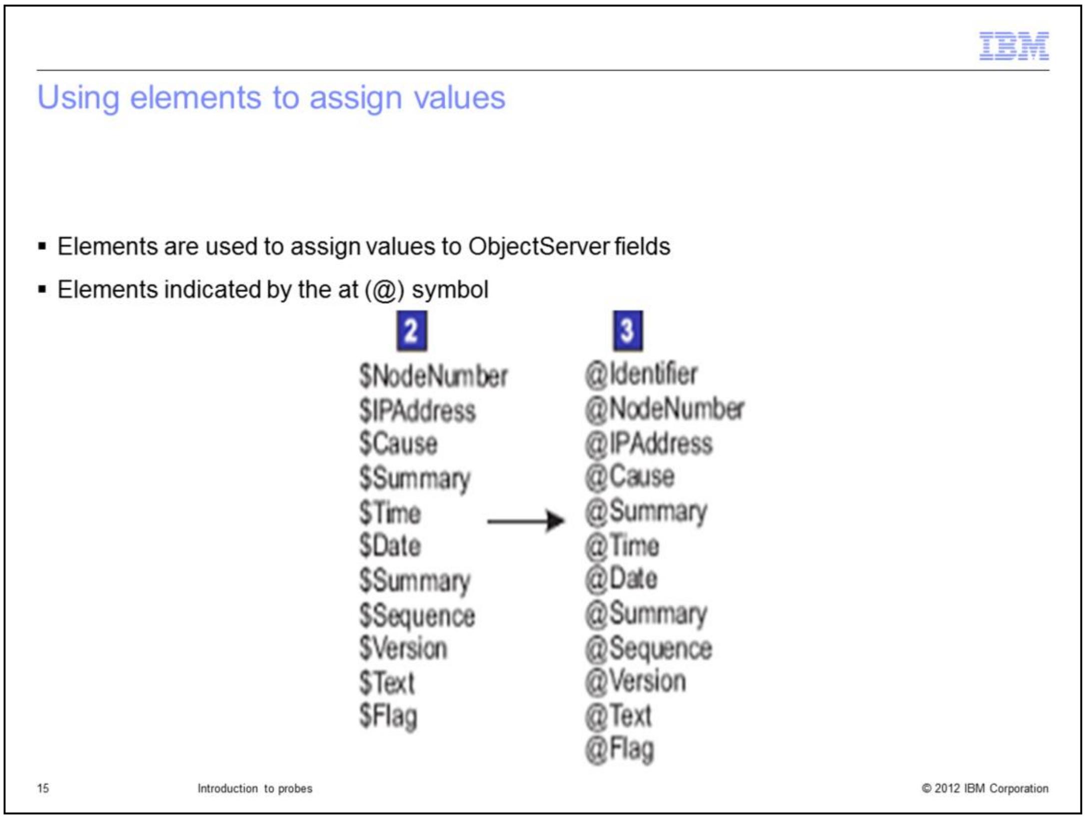
Parsing the tokens

- The probe parses these tokens into elements and processes the elements according to the rules in the rules file
- Elements are identified in the rules file by the dollar ($) symbol, for example, $Node



14 Introduction to probes © 2012 IBM Corporation

Tokens are mapped into elements, which are processed according to probe rules file instructions. Elements in the rules file have the dollar sign prefix.



Using elements to assign values

- Elements are used to assign values to ObjectServer fields
- Elements indicated by the at (@) symbol

2

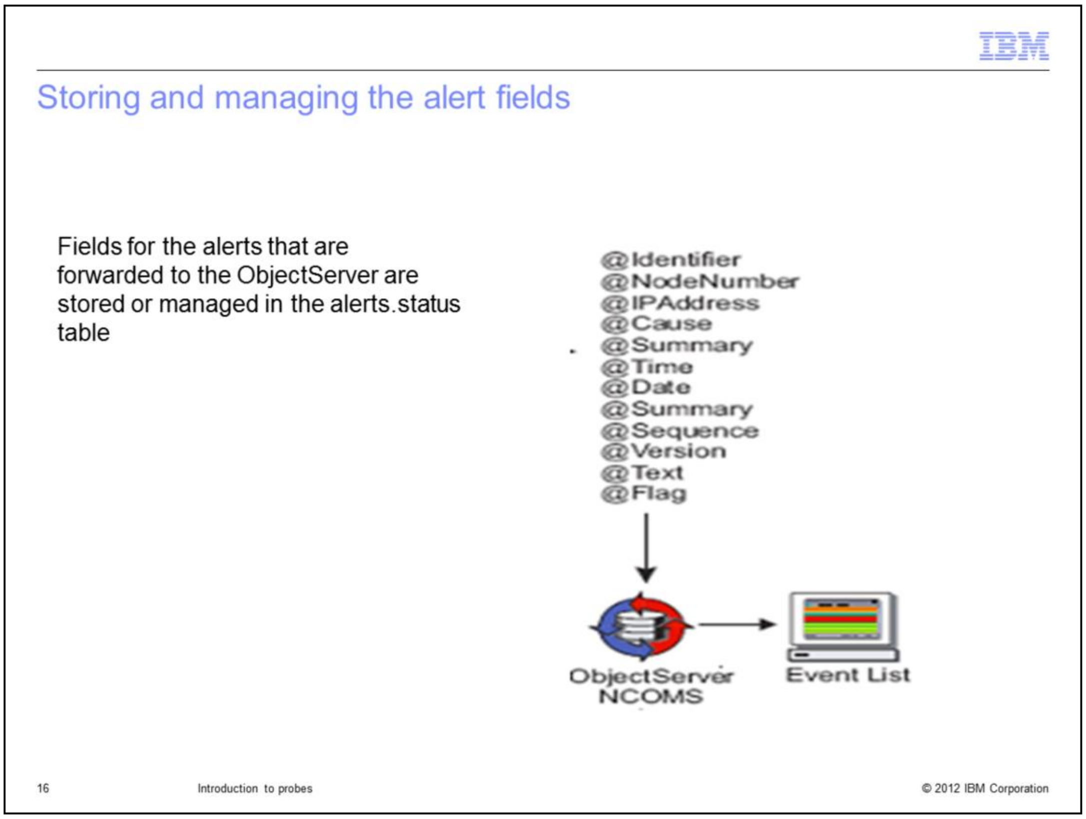
$NodeNumber
$IPAddress
$Cause
$Summary
$Time
$Date
$Summary
$Sequence
$Version
$Text
$Flag

3

@Identifier
@NodeNumber
@IPAddress
@Cause
@Summary
@Time
@Date
@Summary
@Sequence
@Version
@Text
@Flag

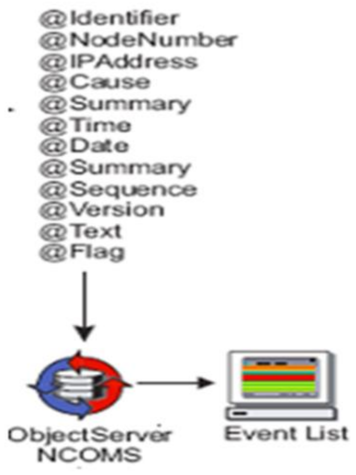
15 Introduction to probes © 2012 IBM Corporation

The rules file elements are mapped into ObjectServer alert field elements. At this stage, the elements have the at (@) sign prefix.



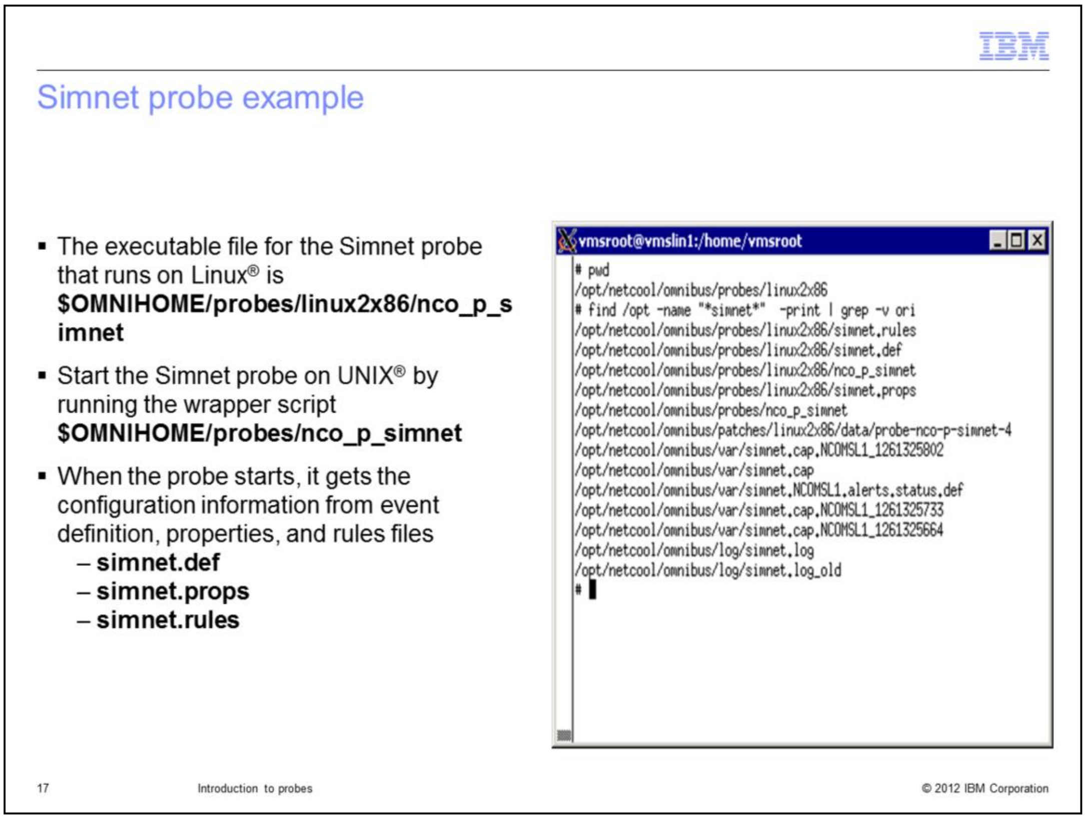
Storing and managing the alert fields

Fields for the alerts that are forwarded to the ObjectServer are stored or managed in the alerts.status table



16 Introduction to probes © 2012 IBM Corporation

The elements that are ready for the ObjectServer are forwarded to the ObjectServer and stored in the alerts.status table on the ObjectServer.



Simnet probe example

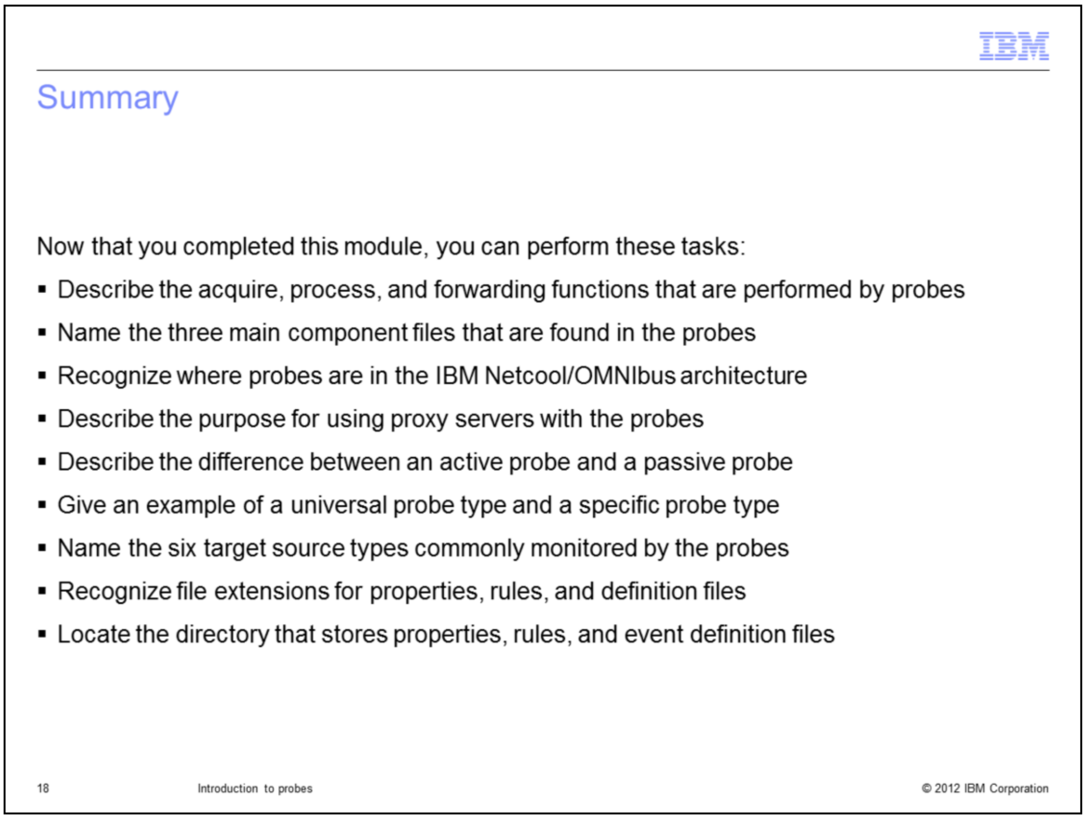
- The executable file for the Simnet probe that runs on Linux® is **$OMNIHOME/probes/linux2x86/nco_p_simnet**
- Start the Simnet probe on UNIX® by running the wrapper script **$OMNIHOME/probes/nco_p_simnet**
- When the probe starts, it gets the configuration information from event definition, properties, and rules files
  - **simnet.def**
  - **simnet.props**
  - **simnet.rules**

vmsroot@vmslin1:/home/vmsroot

```
# pwd
/opt/netcool/omnibus/probes/linux2x86
# find /opt -name "*simnet*"  -print | grep -v ori
/opt/netcool/omnibus/probes/linux2x86/simnet.rules
/opt/netcool/omnibus/probes/linux2x86/simnet.def
/opt/netcool/omnibus/probes/linux2x86/nco_p_simnet
/opt/netcool/omnibus/probes/linux2x86/simnet.props
/opt/netcool/omnibus/probes/nco_p_simnet
/opt/netcool/omnibus/patches/linux2x86/data/probe-nco-p-simnet-4
/opt/netcool/omnibus/var/simnet.cap.NCOMSL1_1261325802
/opt/netcool/omnibus/var/simnet.cap
/opt/netcool/omnibus/var/simnet.NCOMSL1.alerts.status.def
/opt/netcool/omnibus/var/simnet.cap.NCOMSL1_1261325733
/opt/netcool/omnibus/var/simnet.cap.NCOMSL1_1261325664
/opt/netcool/omnibus/log/simnet.log
/opt/netcool/omnibus/log/simnet.log_old
# 
```

17 Introduction to probes © 2012 IBM Corporation

In the illustration on this slide, you can see the path to the rules, props, log, and definition files.



Summary

Now that you completed this module, you can perform these tasks:

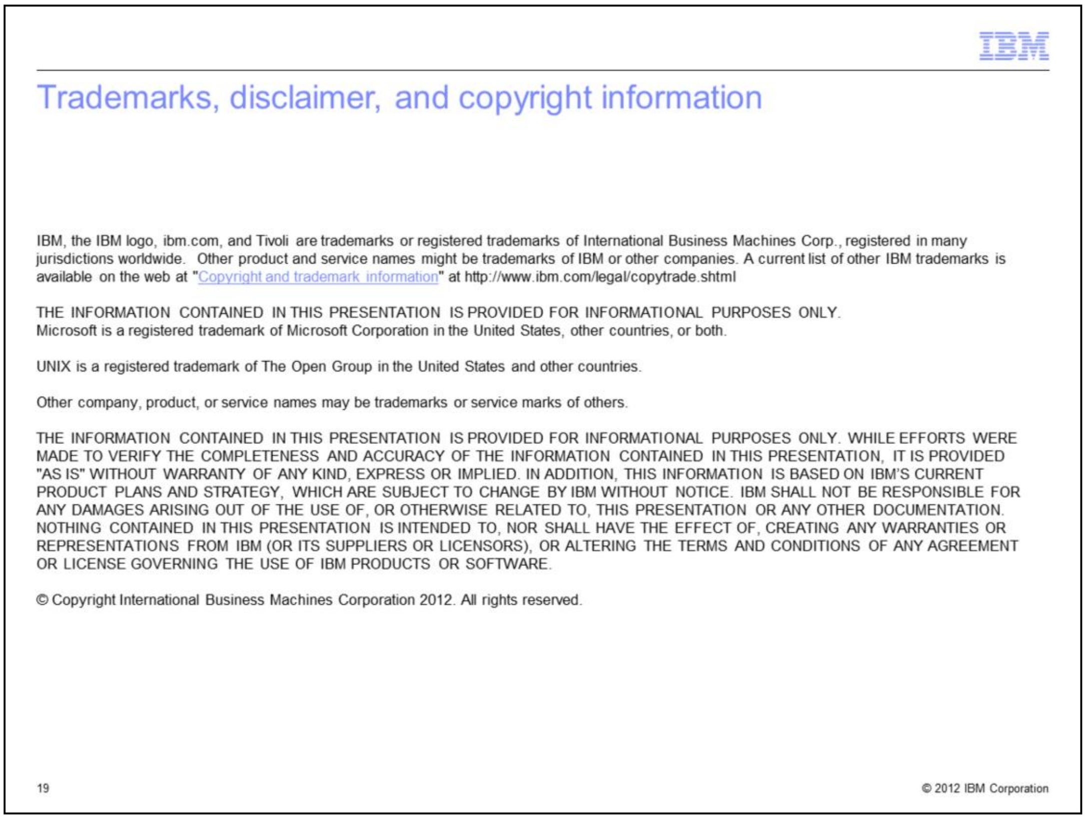
- Describe the acquire, process, and forwarding functions that are performed by probes
- Name the three main component files that are found in the probes
- Recognize where probes are in the IBM Netcool/OMNIbus architecture
- Describe the purpose for using proxy servers with the probes
- Describe the difference between an active probe and a passive probe
- Give an example of a universal probe type and a specific probe type
- Name the six target source types commonly monitored by the probes
- Recognize file extensions for properties, rules, and definition files
- Locate the directory that stores properties, rules, and event definition files

18 Introduction to probes © 2012 IBM Corporation

Now that you completed this module, you can perform these tasks:

Understand probe acquisition, processing, and forwarding. Name the three main IBM Netcool/OMNIbus probe files. Place probes logically within a IBM Netcool/OMNIbus architecture diagram. Understand the purpose for using proxy servers. Describe the difference between active and passive probes. Give an example of both a universal probe and a specific probe.

Name six common probe target source types. Recognize file extensions for and locate the directory for properties, rules, and event definition files.



Trademarks, disclaimer, and copyright information