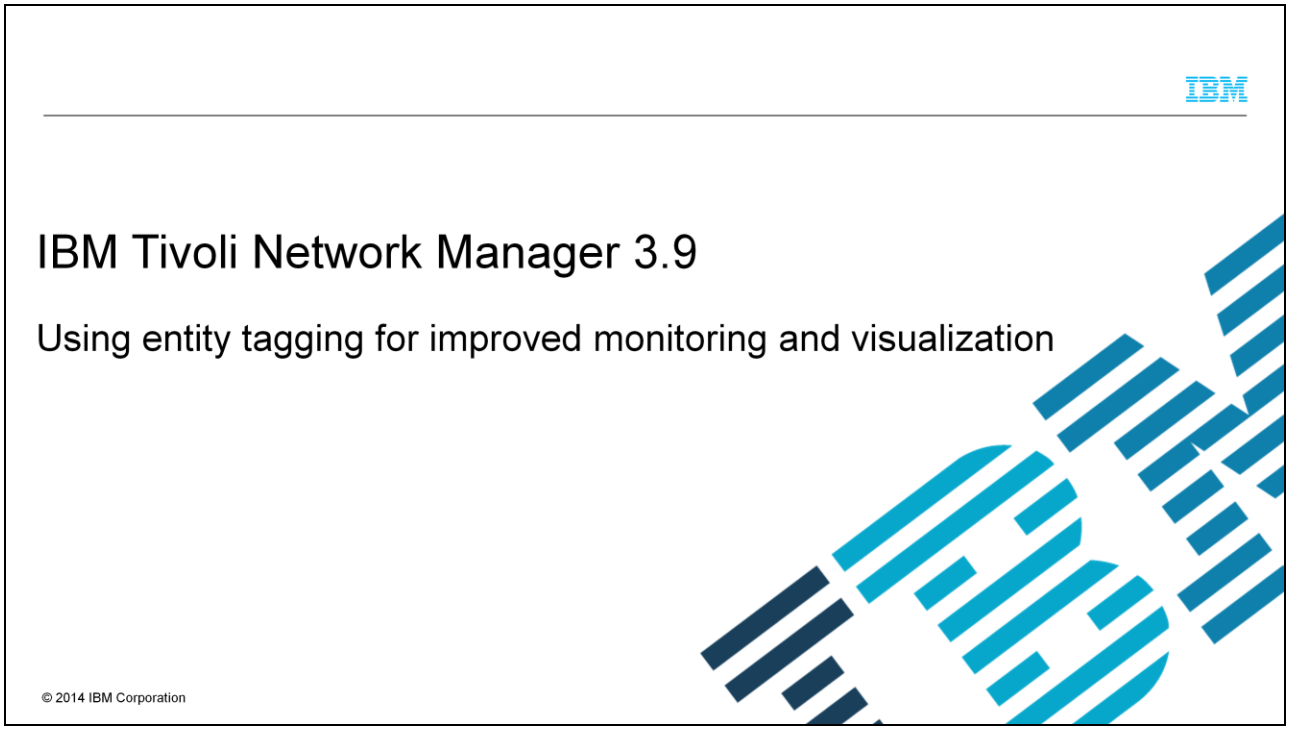


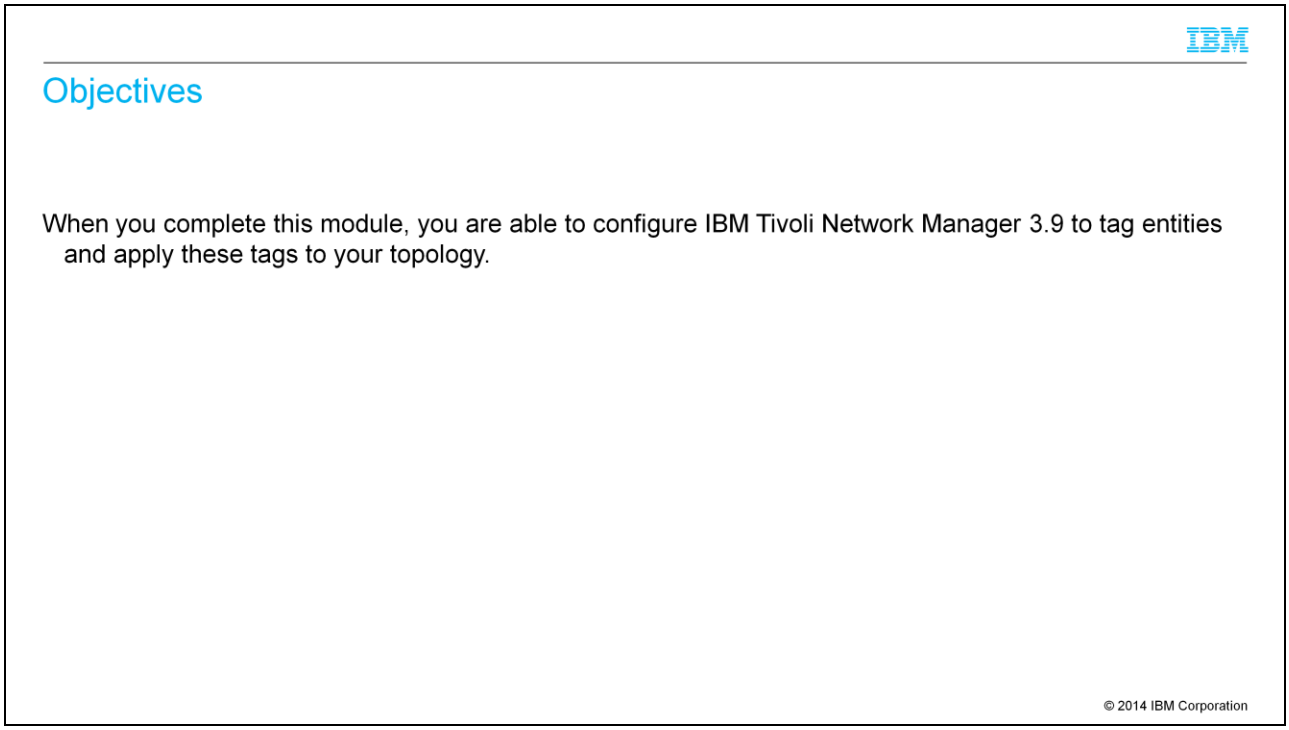
This module covers the use of entity tagging for IBM Tivoli® Network Manager version 3.9. Entity tagging creates more powerful ways to monitor and visualize the IBM Tivoli Network Manager topology.

Prerequisites

As a prerequisite, you must have one of these:

- IBM Tivoli Network Manager 3.9 GA with the latest fix pack applied
- IBM Tivoli Network Manager 3.9 Refresh with the latest fix pack applied

This module shows configuration changes for IBM Tivoli Network Manager 3.9. Install the latest fix pack. To use the tags, you also need access to the IBM Tivoli Network Manager graphical user interface.



When you complete this module, you are able to configure IBM Tivoli Network Manager 3.9 to tag entities in three different ways. Once IBM Tivoli Network Manager is tagged, you are able to use the IBM Tivoli Network Manager GUI to create either a tagged network view or a unique polling profile.

Definitions

Tag: A tag is a label that is associated with an entity. The entity is mostly likely an IP address that is discovered.

Tagging: The process of applying the tag to the entity.

A tag is a label that is created by you to associate with an entity. Most likely the entity is an IP address. During the IBM Tivoli Network Manager discovery phase, the process of tagging occurs.

Methods of tagging

There are three tagging methods:

- Tagging entities in a flat file, as part of a file finder discovery
- Tagging individual IP addresses
- Tagging subnets of IP addresses

Most configuration changes can be seen in **$NCHOME/etc/precision**.

There are three methods to tag entities before the discovery. You can modify a flat file as used in a file finder discovery. You can tag individual IP addresses. You can also mark an entire subnet with a tag. Like other configuration files with IBM Tivoli Network Manager, these changes reside in **$NCHOME/etc/precision**.

Creation of tags in a flat file

```
[root@ncServer precision]# cat /etc/precision/myhosts
ServerWest,192.168.0.1,okpoll
ServerCentral,192.168.0.2
ServerEast,192.168.0.3,okpoll
[root@ncServer precision]#
```


The first example is tagging applied to a flat file. Here the discovery includes three devices, ServerWest, ServerCentral, and ServerEast. In this case, the file finder discovery is delimited by commas. But here, a tag is applied of the customer's choosing, marking ServerWest and ServerEast as **okpoll**.

DiscoFileFinderParseRules.cfg

```
insert into fileFinder.parseRules(
      m_FileName,
      m_Delimiter,
      m_ColDefs
)
values
(
      "/etc/precision/myhosts",
      "[,]+",
     [
          {
                m_VarName="m_Name",
                m_ColNum=1
           },
           {
                m_VarName="m_UniqueAddress",
                m_ColNum=2
           },
          {
               m_VarName="m_CustomTags->pollflag",
               m_ColNum=3
          }
     ]
);
```


The **$NCHOME/etc/precision DiscoFileFinderParseRules.cfg** file exists today to show the delimiters for the file finder. Add a column to handle the new field, which is a tag of the entity. So in addition to Name and Unique Address, a custom tag field should be added called poll flag. This file modification is only necessary for the file finder, flat file tagging discovery.

DbEntityDetails.cfg

```
insert into dbModel.entityDetails
(
   EntityType,
   EntityDetails
)
values
(
   1,  -- chassis
   {
      Pollflag = "eval(text, '&ExtraInfo->m_CustomTags->pollflag')",
      NetworkEdge = "eval(text, '&ExtraInfo->m_NetworkEdge')"
   }
);
```


The **$NCHOME/etc/precision DbEntityDetails.cfg** file is also modified, adding an entry for poll flag. This change is required for all three methods in this module. So using file finder tagging, tagging by IP address, or tagging by subnet each needs this change.

Results can be seen in service model

```
|ncServer:1.> select ObjectId, Address, ExtraInfo->m_CustomTags, ExtraInfo->m_BaseName from master.entityByName where Address(2) like '192.168.0';
|ncServer:2.> go
{
    ObjectId=8565501;
    Address=['','','192.168.0.1']
    m_CustomTags={
        pollflag='okpoll';
    };
    m_BaseName='ServerWest';
}
{
    ObjectId=8565503;
    Address=['','','192.168.0.3'];
    m_CustomTags={
        pollflag='okpoll';
    };
    m_BaseName='ServerEast';
}
{
    ObjectId=8565505;
    Address=['','','192.168.0.2'];
    m_BaseName='ServerCentral';
}
```


When the IBM Tivoli Network Manager discovery is complete, you can look in the NCP OQL query of service model. You can see that the tagging was successful.

Examination of the tags in NCIM database

```
[root@ncServer bin]# ./ncp_oql -username ncim -password ncim -service ncim -domain SCOM39
ncp_oql ( IBM Tivoli Network Manager OQL Interface )
Copyright (C) 1997 - 2010 By IBM Corporation. All Rights Reserved. See product license for details.

IBM Tivoli Network Manager Version 3.9 (Build 97) created by ncpbuild at 17:09:54 Fri Feb  8 GMT 2014

Connected to INFORMIX DB schema NCIM
|ncServer:1.> select * from ncim.entityDetails;
|ncServer:2.> go
{
      ENTITYID=8565501;
      KEYNAME='Pollflag';
      KEYVALUE='okpoll';
}
{
      ENTITYID=8565503;
      KEYNAME='Pollflag';
      KEYVALUE='okpoll';
}
 2 rows(s)
|ncServer:1.>
```


The discovered topology is passed to model, and finally the Netcool® Common Information Model or NCIM. The tags are visible in the NCIM database, entity details table. A user can log in to the IBM Tivoli Network Manager GUI and create views and polls that are based on the tag.

Tagging individual IP addresses in discovery

- Tagging entities in a flat file, as part of a file finder discovery / COMPLETE
- Tagging individual IP addresses
- Tagging subnets of IP addresses

The first method for tagging is complete. Before reviewing the options available in the GUI for using tags, tagging individual IP addresses and tagging subnets are configured. Remember, in your discovery, any or all three methods are available.

Add individual addresses to DiscoConfig.cfg

```
insert into disco.ipCustomTags
(
        m_UniqueAddress,
        m_CustomTags
 )
values
(
        '192.168.0.2',
        {
       pollflag="okpoll"
        }
);
```


The **$NCHOME/etc/precision DiscoConfig.cfg** file can be modified to tag an individual IP address. This would be necessary if the ping finder was used for discovery and not file finder. Like the previous example, the **DbEntityDetails.cfg** file must also be modified to ready model and NCIM for the update.

Tagging an entire subnet

- Tagging entities in a flat file, as part of a file finder discovery / COMPLETE
- Tagging individual IP addresses/ COMPLETE
- Tagging subnets of IP addresses

Often an entire subnet might be discovered as part of the ping finder. The third method is to tag all IP addresses that fit in the configured subnet. This is the preferred configuration rather than tagging hundreds of individual IP addresses.

Add a subnet to DiscoConfig.cfg

```
insert into disco.filterCustomTags
(
        m_Filter,
        m_CustomTags
 )
values
(
        "m_UniqueAddress LIKE '192.168.0'",
          {
          pollflag="okpoll"
           }
);
```


The **$NCHOME/etc/precision DiscoConfig.cfg** file can be modified to tag an entire subnet. Here, discovered IP addresses that are part of the 192.168.0 subnet are all tagged as **okpoll**. Again the **DbEntityDetails.cfg** file was modified to prepare model and NCIM for the data.

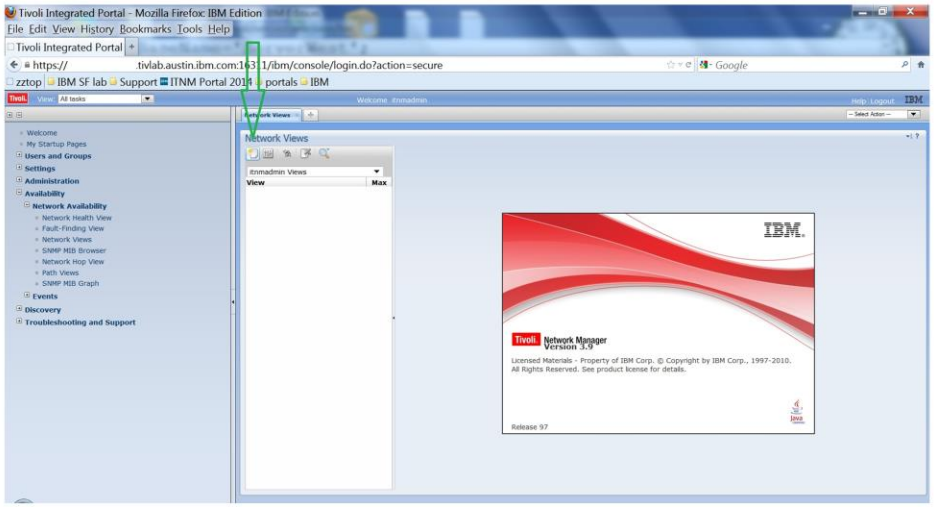
Example of creating a network view from a tag

```
|ncServer:1.> select ObjectId, Address, ExtraInfo->m_CustomTags, ExtraInfo->m_BaseName from master.entityByName where Address(2) like '192.168.0';
|ncServer:2.> go
{
    ObjectId=8565501;
    Address=['','','192.168.0.1'];
    m_CustomTags={
        pollflag='NoTickets';
    };
    m_BaseName='ServerWest';
}
{
    ObjectId=8565503;
    Address=['','','192.168.0.2'];
    m_CustomTags={
        pollflag='NoTickets';
    };
    m_BaseName='ServerCentral';
}
{
    ObjectId=8565505;
    Address=['','','192.168.0.3'];
    m_BaseName='ServerEast';
}
( 3 record(s) : Transaction complete )
```


Here is an example of creating a network view based on tags. The administrator has chosen the custom tag to be called poll flag **no tickets**. The administrator can use file finder tagging or individual IP tagging, either is able to tag some entities within a subnet, but not all. The poll flag, **no tickets** will become a network view, which operators will know to mean they are not to open any tickets.

Creation of a network view based on a tag

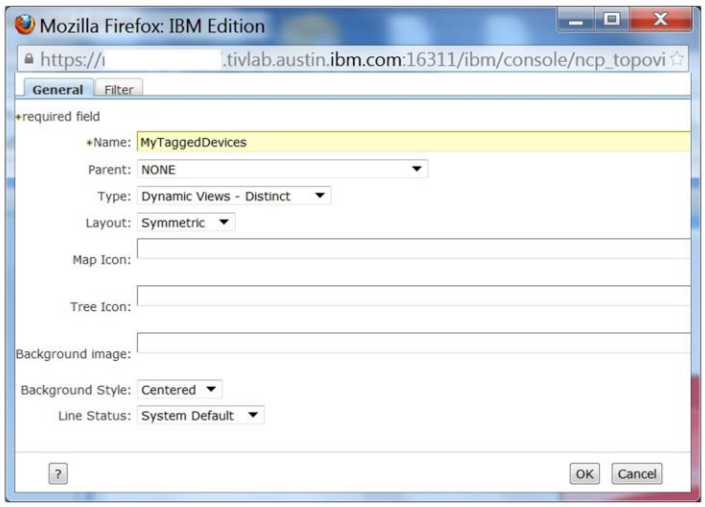
Click the icon to create a new view (green arrow).



The administrator logs in to the IBM Tivoli Network Manager GUI, clicks **Availability**, **Network Availability**, **Network Views**. Finally, you click the icon to create a new view that is highlighted by the green arrow.

Create a dynamic view – distinct (1 of 2)

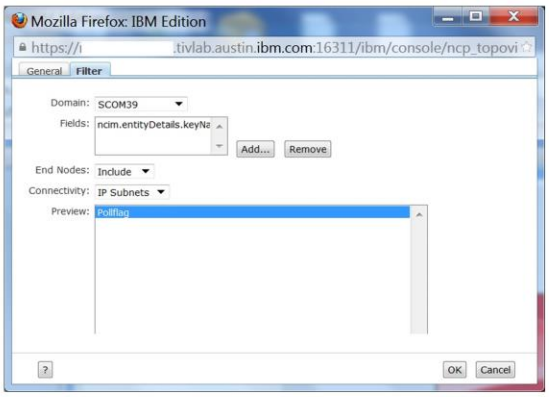
This view is named MyTaggedDevices of type Dynamic Views – Distinct.



In the new window, type a name for the view. In the example, the name is **MyTaggedDevices**. Select **Dynamic Views – Distinct** from the **Type** drop-down list.

Create a dynamic view – distinct (2 of 2)

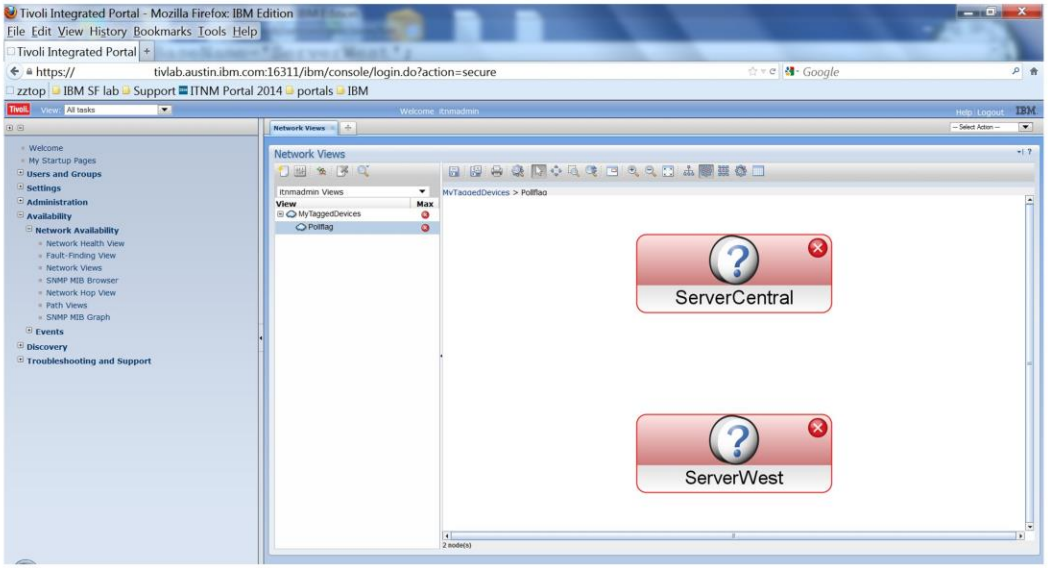
Click **Filter** to begin selecting your domain.



Click the **Filter** tab and select your domain. Add the field for **entityDetail**s, the location in NCIM that holds that tagged information. The view can be key name or key value. In this example, you look for **poll flag**, but you could have also configured the view for **no tickets**.

Network view output

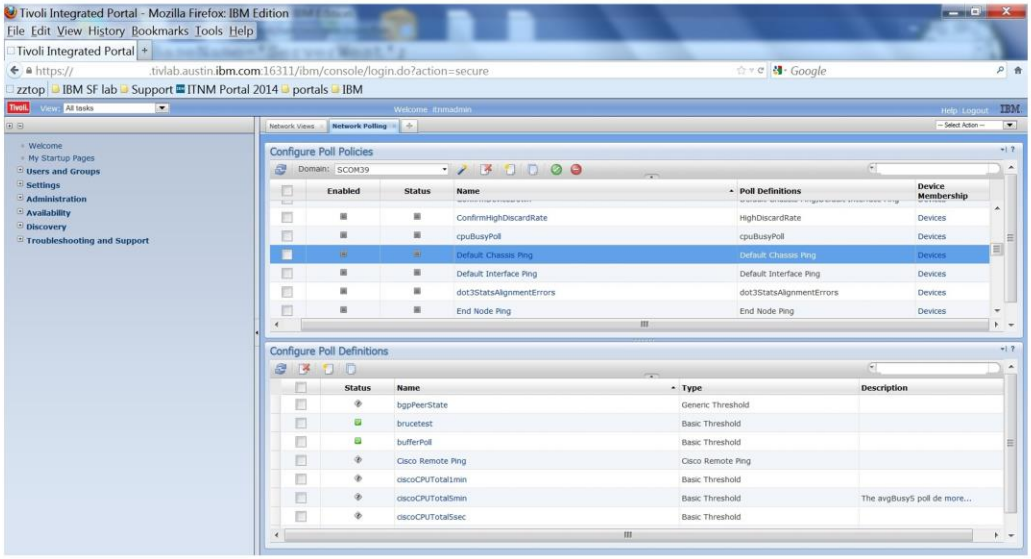
Network view output



Your new view opens based on key name poll flag. These are the original two devices that were tagged within discovery as **poll flag**, **no tickets**. This simple example illustrates that a customized network view can exist based on discovery data. More devices can be tagged as **poll flag**, **no tickets**, and those devices also appear in this view.

Polling based on tagged entities

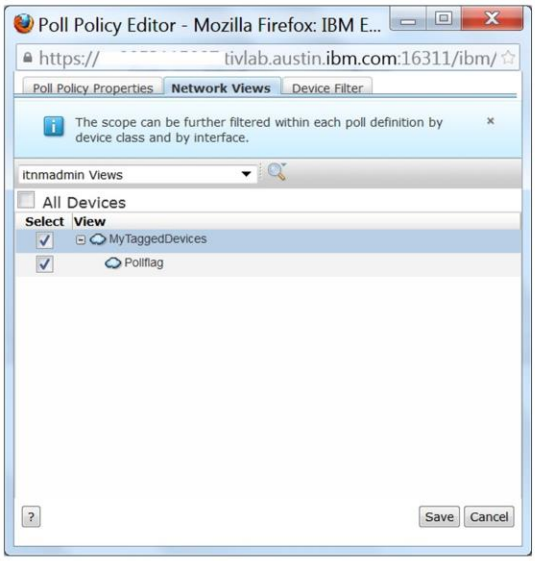
Open the default chassis ping policy.



The tagged entities can also be polled. Rather than change a device filter for a network poll, devices can be added to the created network view. The last section created a network view called **poll flag**. Open the default chassis ping policy, highlighted in blue.

Click the Network Views tab to select a custom view to poll

Click the **Network Views** tab.



For this default chassis ping, rather than poll all devices, or create a device filter, the **Network Views** tab is chosen. Now that a view exists called **poll flag**, that box can be ticked, and IBM Tivoli Network Manager poller will only use the default chassis ping to poll devices in this view. Since the view was created from the discovery tagging, the administrator has greater control over the polling targets.

Summary

- Now that you have completed this module, you can configure IBM Tivoli Network Manager 3.9 to tag entities and apply these tags to your topology.

Now that you have completed this module, you can configure IBM Tivoli Network Manager 3.9 to tag entities in three different ways. Once IBM Tivoli Network Manager 3.9 is tagged, you are able to use the IBM Tivoli Network Manager GUI to create either a tagged network view or a unique polling profile.

Trademarks, disclaimer, and copyright information