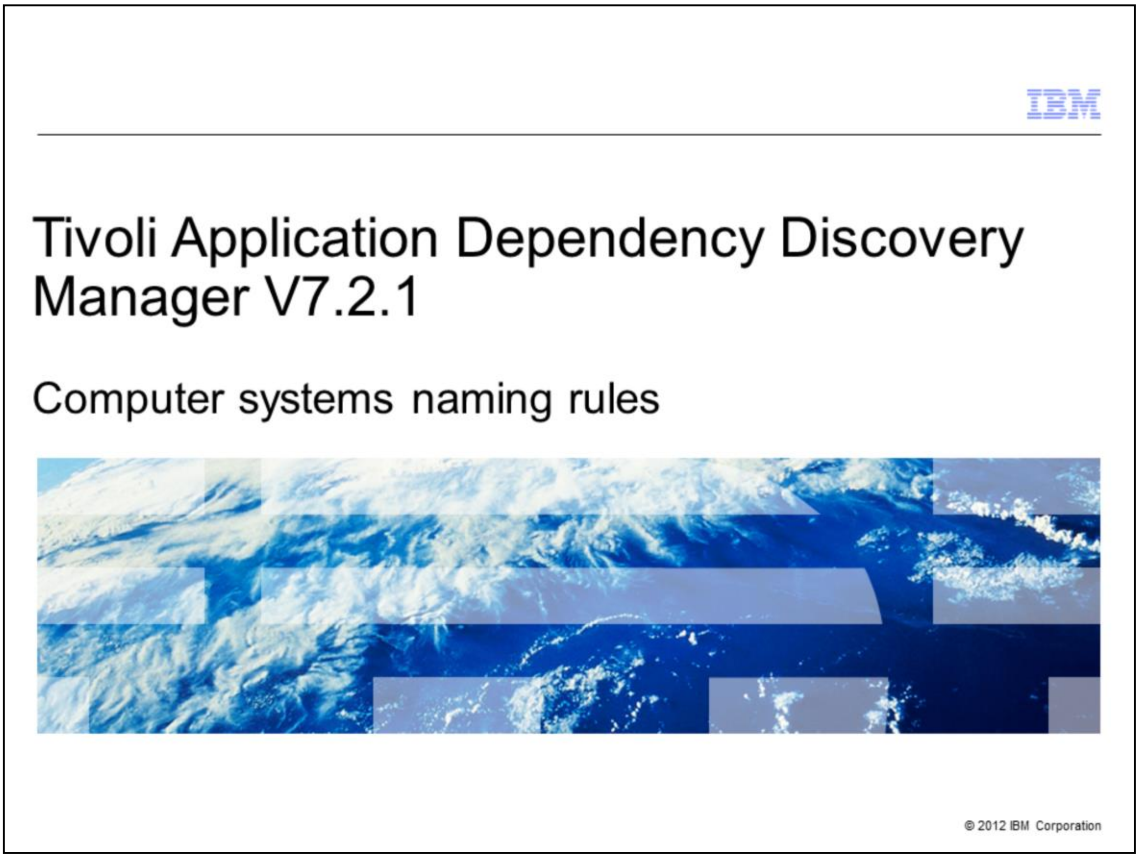
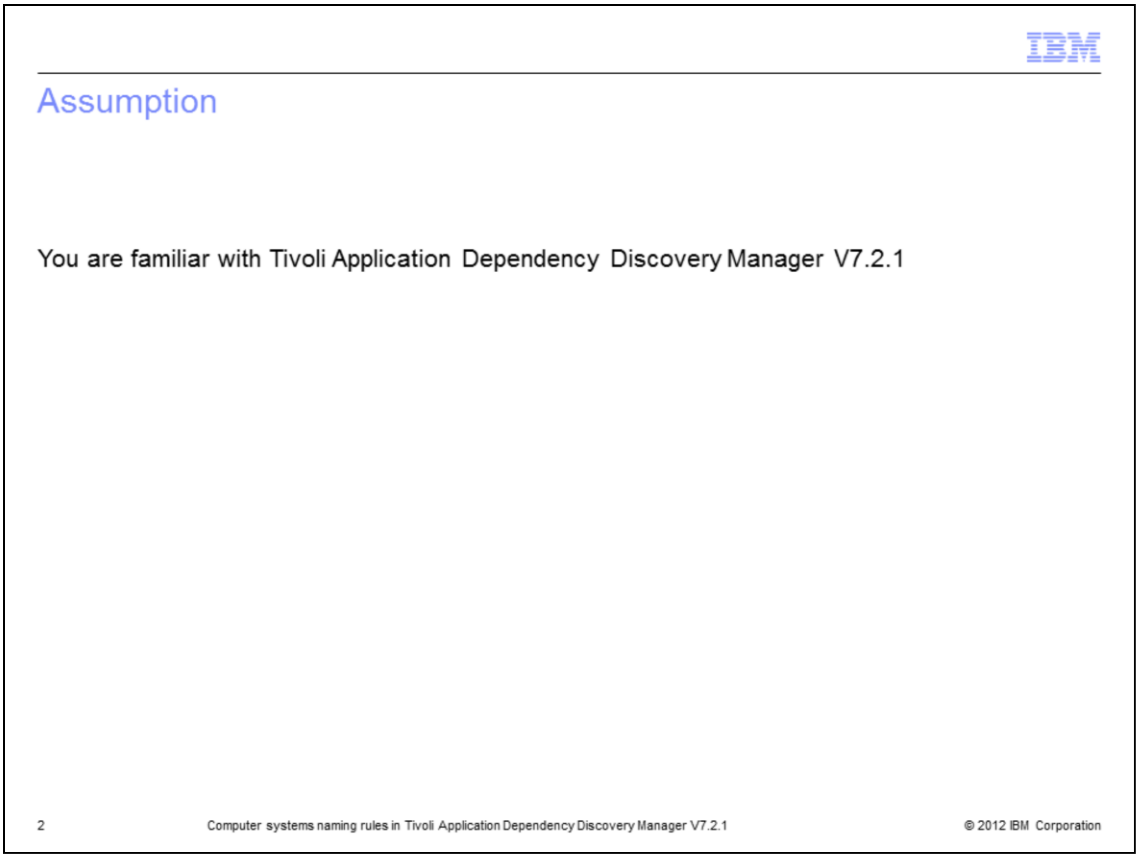


Tivoli Application Dependency Discovery Manager V7.2.1

Computer systems naming rules

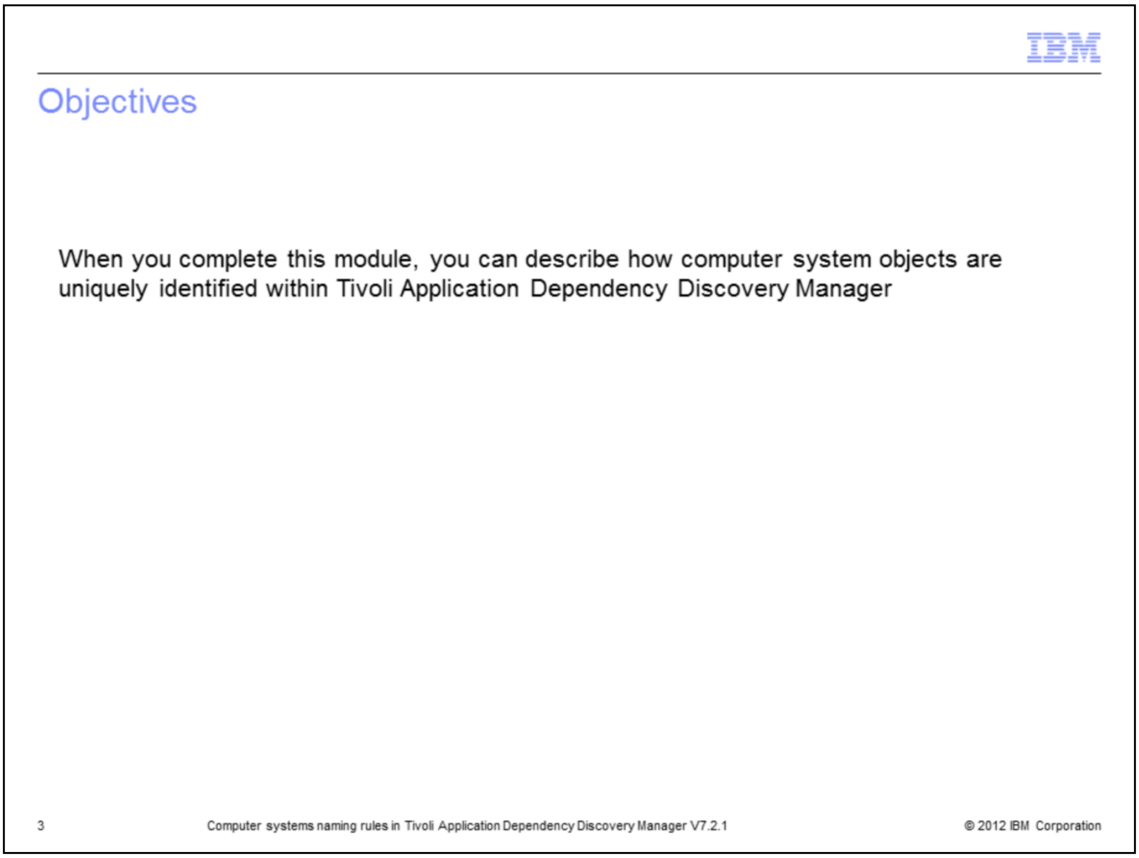
In this module, you learn about computer system naming rules in Tivoli Application Dependency Discovery Manager V7.2.1.



Assumption

You are familiar with Tivoli Application Dependency Discovery Manager V7.2.1

An assumption for this module is that you are familiar with Tivoli Application Dependency Discovery Manager.

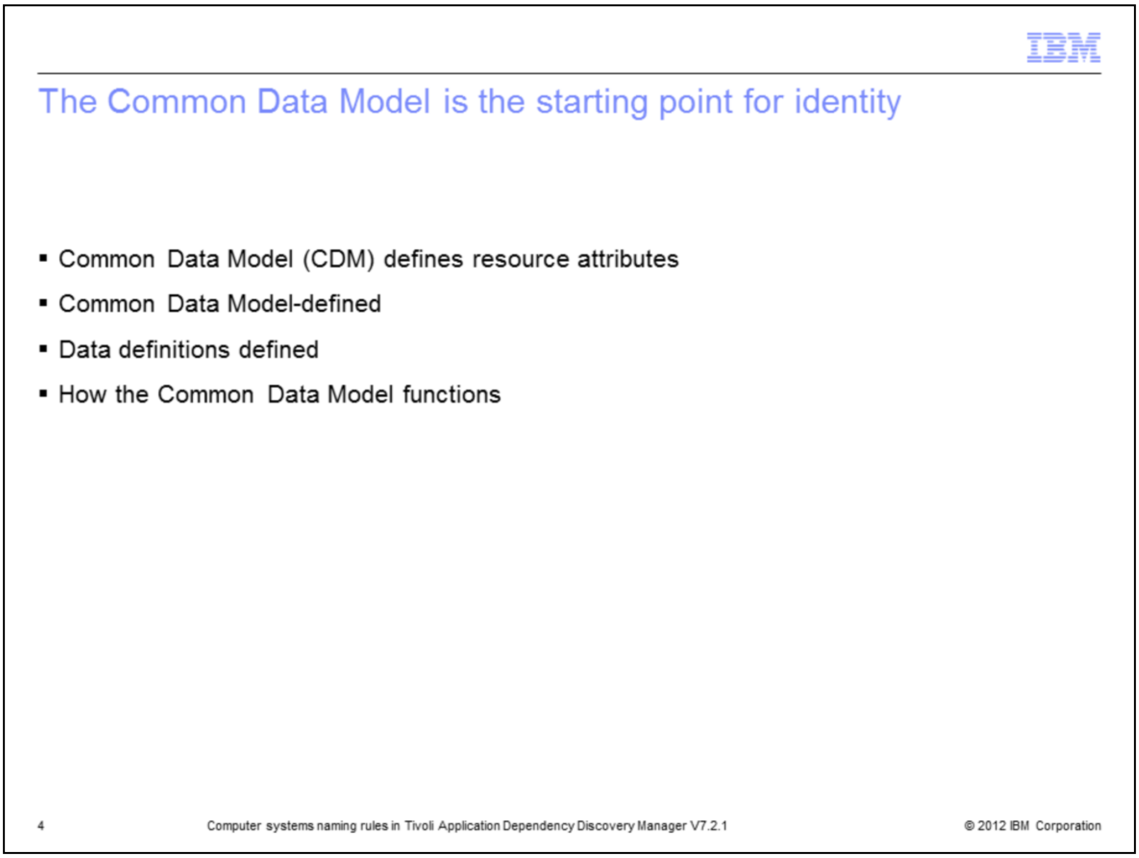


Objectives

When you complete this module, you can describe how computer system objects are uniquely identified within Tivoli Application Dependency Discovery Manager

3 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

The objective for this module is to describe how computer systems are uniquely identified within Tivoli Application Dependency Discovery Manager. This is a high level introduction intended to prepare customers for future education based on this module.



The Common Data Model is the starting point for identity

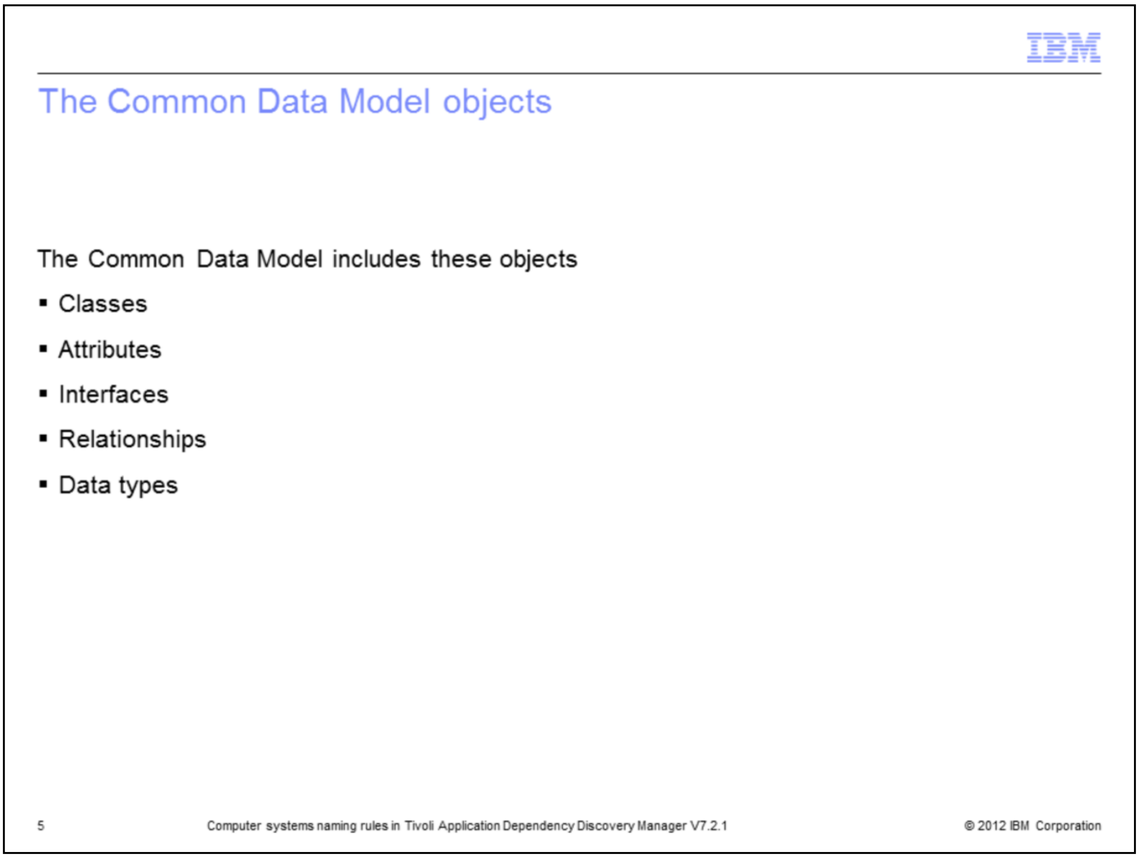
- Common Data Model (CDM) defines resource attributes
- Common Data Model-defined
- Data definitions defined
- How the Common Data Model functions

To understand how computer systems are uniquely identified in Tivoli Application Dependency Discovery Manager, you must first understand the basics of the Common Data Model (CDM) it uses to define resource attributes.

The CDM is the definitional language used to integrate understanding and the exchange of data between Tivoli management products concerning customer business resources and components. The CDM is entirely composed of data definitions.

These data definitions are characteristics or attributes that identify resources, their meanings, and any restrictions on their lengths or values. The content of the CDM is obtained by merging of applicable industry information, data model standards, and the data models used by your current products into a single converged model.

The CDM is a definitional language for resources and components of a data center environment. This language is used among Tivoli products so that they can integrate and share information. The CDM is really data definitions; it is the characteristics that can be associated with resources as defined by industry standards and Tivoli data models.



The Common Data Model objects

The Common Data Model includes these objects

- Classes
- Attributes
- Interfaces
- Relationships
- Data types

5 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

The CDM contains the objects ***classes***, ***attributes***, ***interfaces***, ***relationships***, and ***data types***.

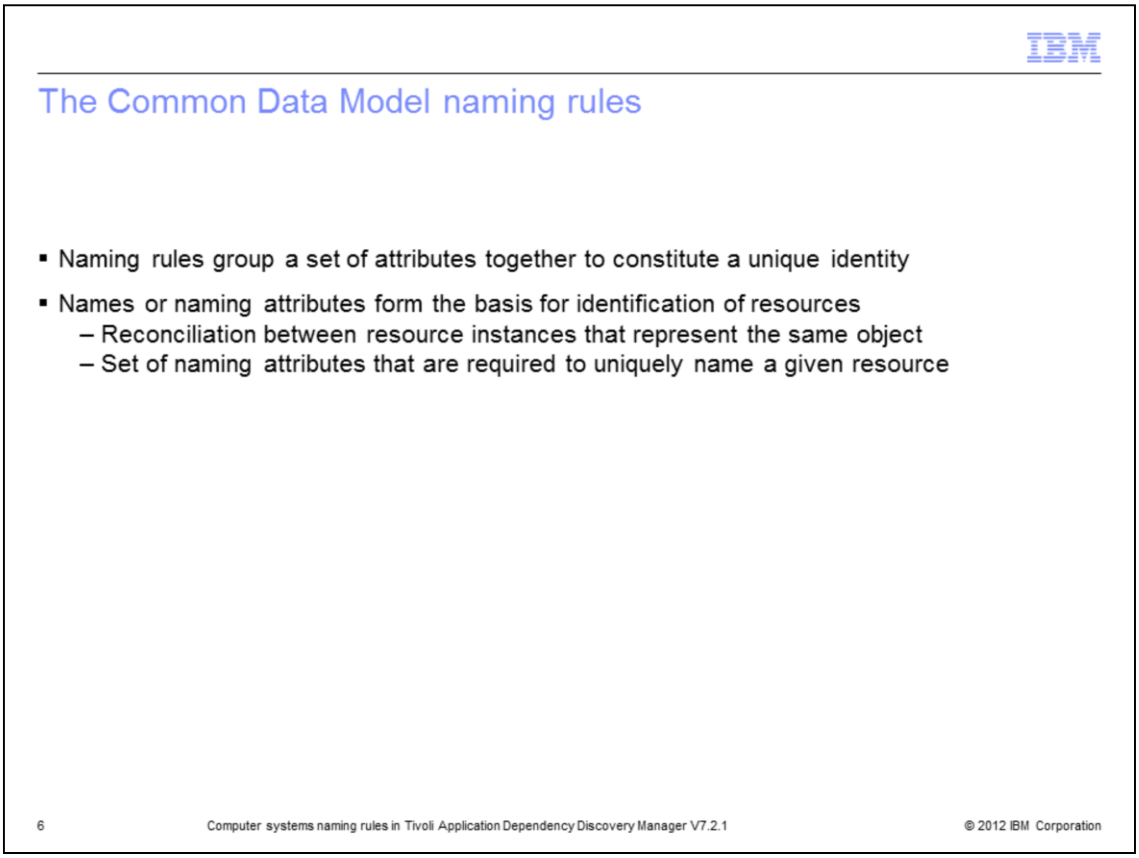
A ***class*** is a construct used to group related attributes, for this presentation, you are looking closely at the ComputerSystem class. An attribute is a particular property that is valid for the class.

When a resource instance is created, there is the ability to store data for any ***attribute*** valid for a resource instance. Not all attributes are required to contain a value, however, there are some attributes that are in use to represent a unique identity for a resource instance. These attributes are often referred to as ***identity or naming attributes*** and are included in the naming rules presented in a moment. The remaining object types are not generally required to name a computer system, however, the object types are described briefly for general knowledge.

Interfaces enable the convenient reuse of a set of attributes and provide increased flexibility in relationship definitions. For example, the attribute **VersionString** is a valid attribute for several different class types of resource instances. Rather than duplicating the attribute across multiple classes in the CDM, an interface is created to represent the set of attributes that pertain to version data.

Relationships contain associations between two resource instances that show how resource instances are related to each other. Relationships can only be between classes.

The information contained in attributes and measurements must be presented in a well-known syntax, and for this purpose the CDM defines a set of ***data types*** that are used for representing entity information.



IBM

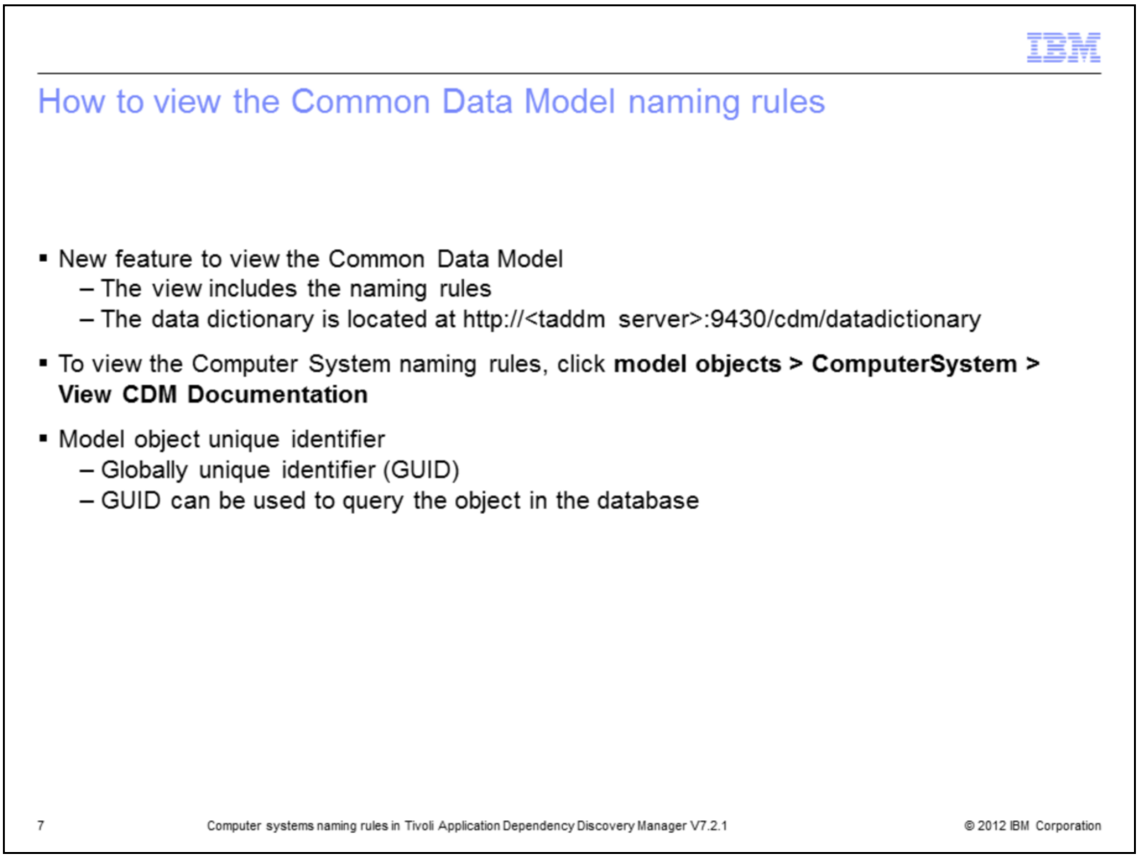
The Common Data Model naming rules

- Naming rules group a set of attributes together to constitute a unique identity
- Names or naming attributes form the basis for identification of resources
  - Reconciliation between resource instances that represent the same object
  - Set of naming attributes that are required to uniquely name a given resource

6 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

Tivoli Application Dependency Discovery Manager uses **Naming Rules** to uniquely identify objects in Tivoli Application Dependency Discovery Manager. To organize the method of generating a unique name, the Common Data Model uses the concept of naming rules to group a set of attributes together that constitutes a unique identity.

Names or naming attributes form the basis for identification of resources and reconciliation between resource instances that represent the same object within the data center. Naming rules contain a set of naming attributes that are required in order to uniquely name a given resource. Tivoli Application Dependency Discovery Manager determines which objects are the same versus which are different with the names. Naming rules are typically based on an industry standard.



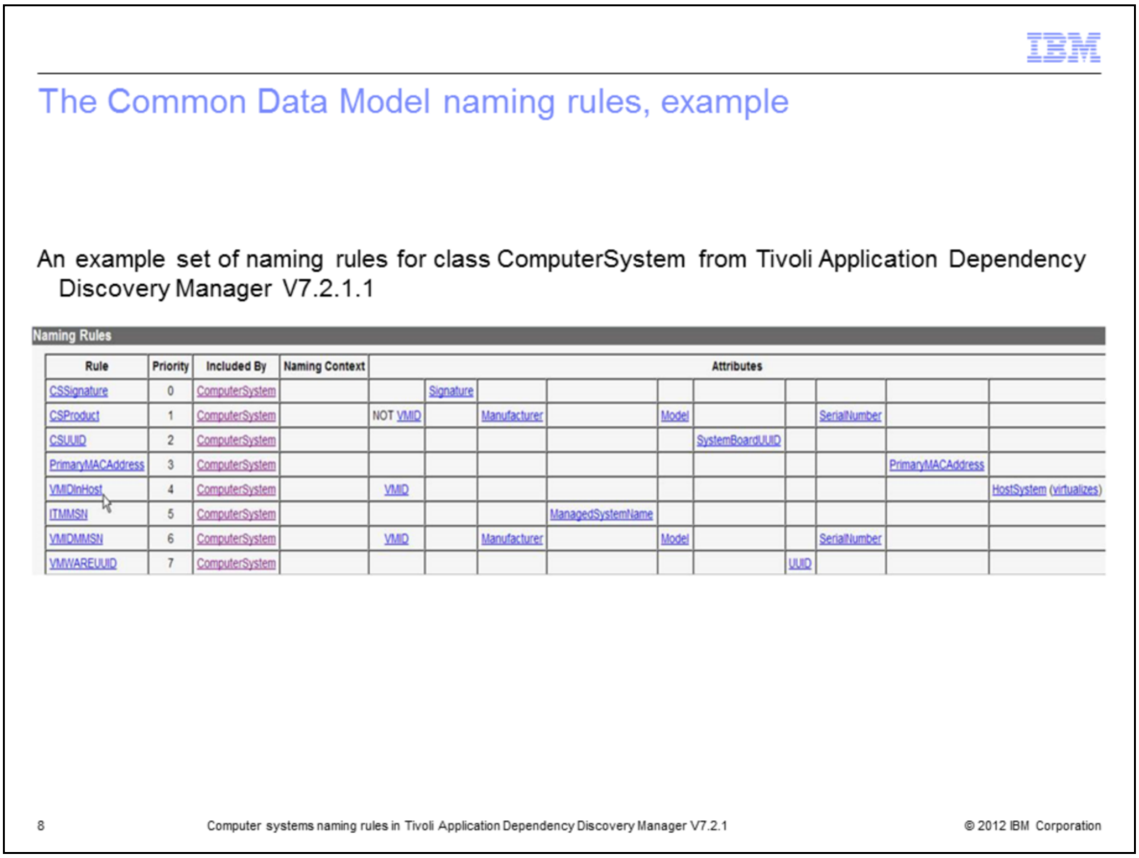
How to view the Common Data Model naming rules

- New feature to view the Common Data Model
  - The view includes the naming rules
  - The data dictionary is located at http://<taddm server>:9430/cdm/datadictionary
- To view the Computer System naming rules, click **model objects > ComputerSystem > View CDM Documentation**
- Model object unique identifier
  - Globally unique identifier (GUID)
  - GUID can be used to query the object in the database

Tivoli Application Dependency Discovery Manager V7.2.1 has a new Data Dictionary feature so you can look at the Common Data Model (CDM). This documentation includes the naming rules for all the available classes in Tivoli Application Dependency Discovery Manager. You can view the dictionary by appending **cdm/datadictionary** to your current Tivoli Application Dependency Discovery Manager server web address.

To view the Computer System naming rules from this web address, click **model objects > ComputerSystem > View CDM Documentation**. A screen capture is on the next slide.

After an object is uniquely identified it is assigned a ***Globally unique identifier*** (GUID). You can use this GUID to query the object in the Tivoli Application Dependency Discovery Manager database.



The Common Data Model naming rules, example

An example set of naming rules for class ComputerSystem from Tivoli Application Dependency Discovery Manager V7.2.1.1

Naming Rules

Rule	Priority	Included By	Naming Context	Attributes									
CSSignature	0	ComputerSystem			Signature								
CSProduct	1	ComputerSystem		NOT VMID		Manufacturer		Model			SerialNumber		
CSUUID	2	ComputerSystem							SystemBoardUUID				
PrimaryMACAddress	3	ComputerSystem										PrimaryMACAddress	
VMIDInHost	4	ComputerSystem		VMID									HostSystem (virtualizes)
ITMMSN	5	ComputerSystem					ManagedSystemName						
VMIDMMSN	6	ComputerSystem		VMID		Manufacturer		Model			SerialNumber		
VMWAREUUID	7	ComputerSystem								UUID			

8 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

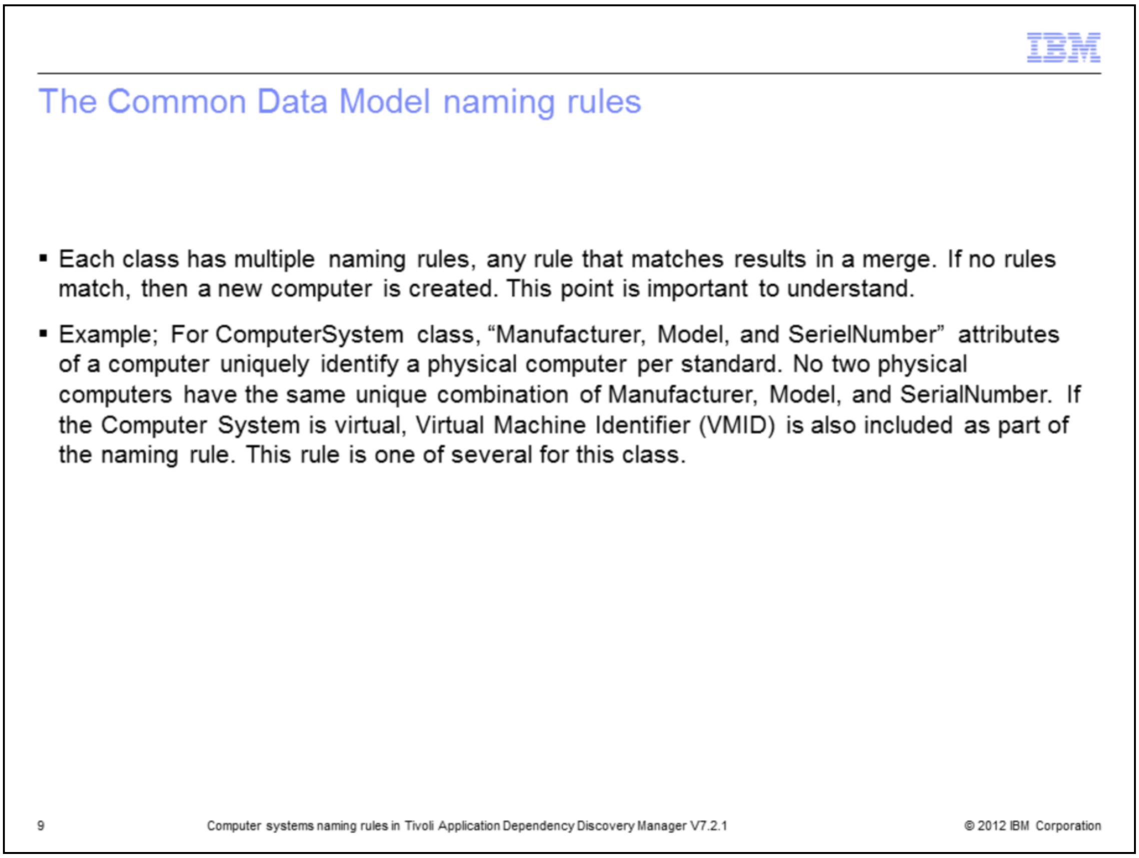
Here is a screen capture of the **ComputerSystem class** naming rules taken from a Tivoli Application Dependency Discovery Manager 7.2.1.1 CDM. There are eight rules defined. The Rule column identifies the name of the rule. The Priority column indicates the order in which they are evaluated. The Naming Context is used to identify a relationship to a second resource. It is used when there is minimal amount of information available to uniquely name the instance based on the attributes that are available on the class.

Example: If all that is known about a particular instance of an operating system is the type, Tivoli Application Dependency Discovery Manager uses the parent computer system as an additional required relationship to uniquely identify the operating system.

The remainder of the chart includes the attributes that comprise each naming rule for the class ComputerSystem.

In summary, for computer systems, the rules are **Signature**; “**Manufacturer, Model, SerialNumber**”; **Universally Unique Identifier** (UUID); **PrimaryMACAddress**; and **SystemBoardUUID**. There are additional rules that include VMID for virtual systems too.

Note: The Fully Qualified Domain Name (FQDN) is not a naming rule. There is no industry standard that ensures an FQDN is unique, therefore it cannot be used to name computer systems.



The Common Data Model naming rules

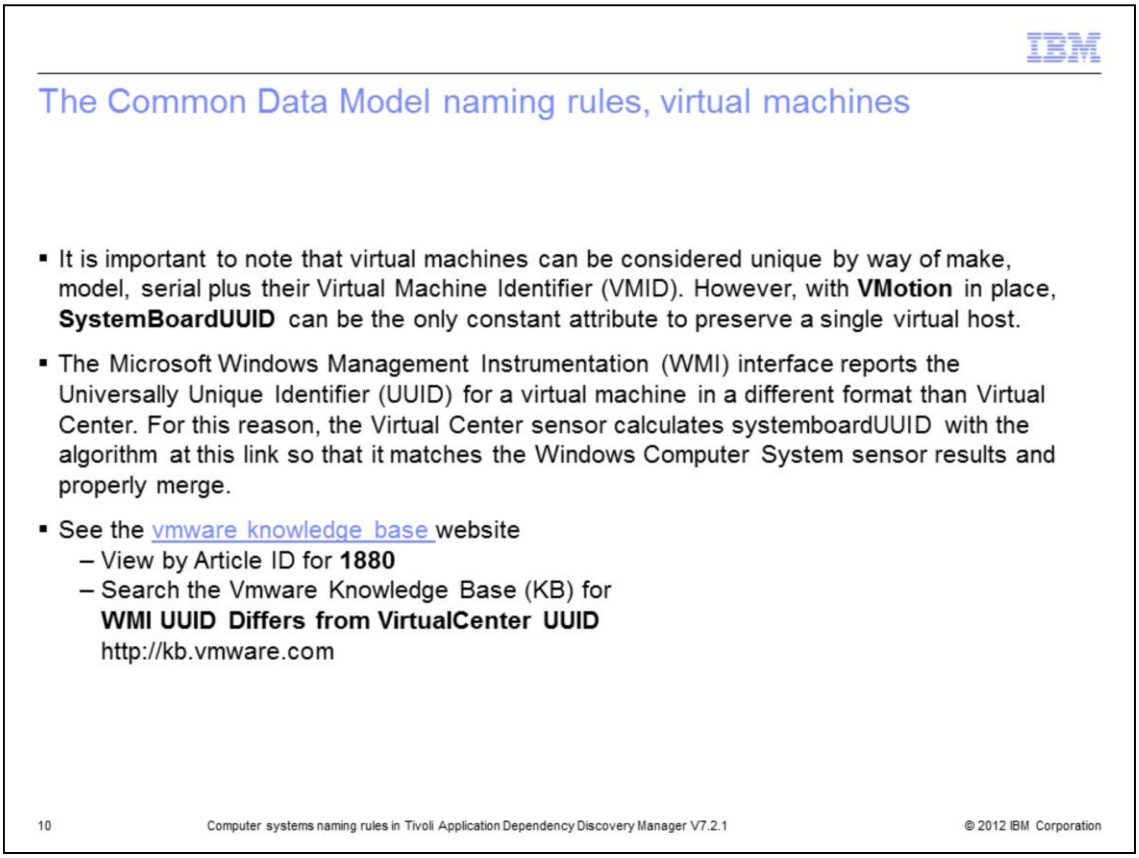
- Each class has multiple naming rules, any rule that matches results in a merge. If no rules match, then a new computer is created. This point is important to understand.
- Example; For ComputerSystem class, “Manufacturer, Model, and SerielNumber” attributes of a computer uniquely identify a physical computer per standard. No two physical computers have the same unique combination of Manufacturer, Model, and SerialNumber. If the Computer System is virtual, Virtual Machine Identifier (VMID) is also included as part of the naming rule. This rule is one of several for this class.

9 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

When reviewing naming rules, you must understand that when Tivoli Application Dependency Discovery Manager evaluates the rules against the attributes of a discovered ComputerSystem. If it finds an instance already in the database with a matching rule, the two instances are merged. If there is no match found on any rule, a new instance is created.

Example: For ComputerSystem class, “Manufacturer, Model, SerialNumber” attributes of a computer uniquely identify a physical computer per standard. No two physical computers can have the same unique combination of Manufacturer, Model, and SerialNumber.

An exception can be virtual machines, but then Virtual Machine Identifier (VMID) is added to the naming rule to make it unique. This is just one of the many rules for this class as each class has multiple rules. It is important to note that if any rule in the set matches a merge occurs. If no rules match, then a **new computer** is created.



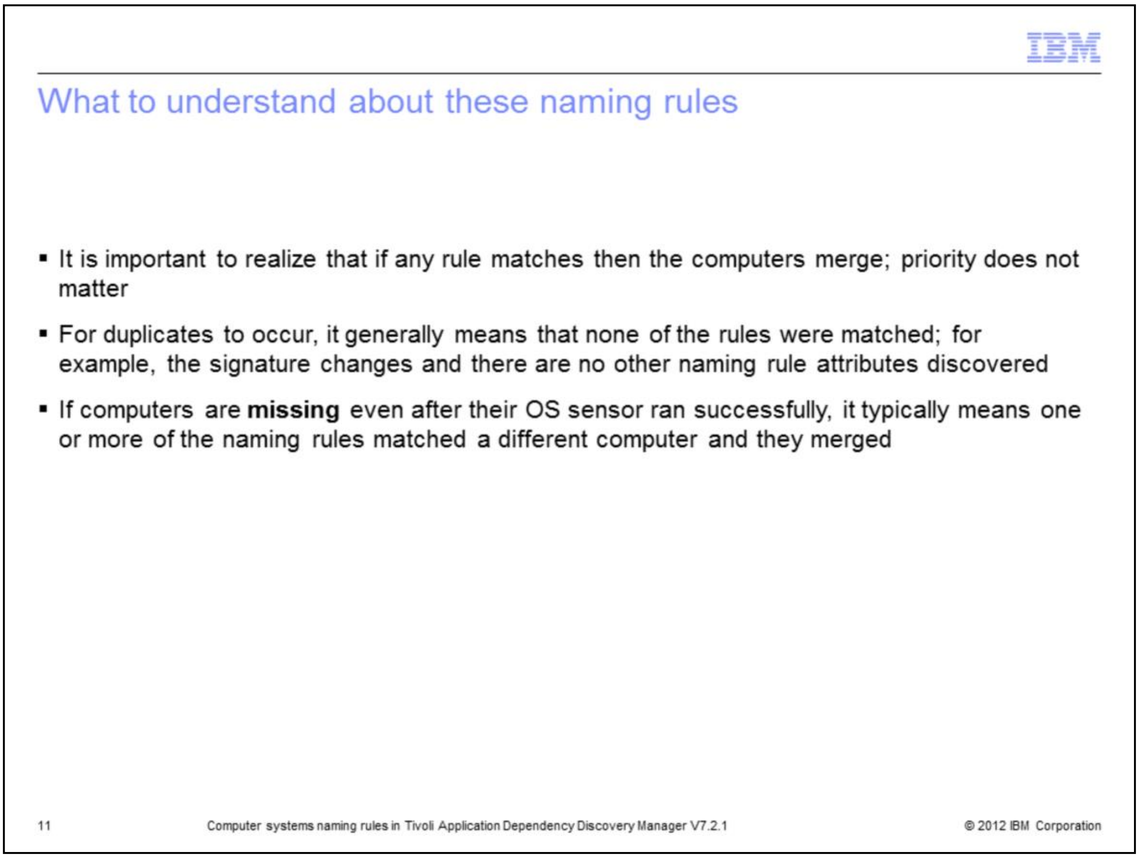
IBM

The Common Data Model naming rules, virtual machines

- It is important to note that virtual machines can be considered unique by way of make, model, serial plus their Virtual Machine Identifier (VMID). However, with **VMotion** in place, **SystemBoardUUID** can be the only constant attribute to preserve a single virtual host.
- The Microsoft Windows Management Instrumentation (WMI) interface reports the Universally Unique Identifier (UUID) for a virtual machine in a different format than Virtual Center. For this reason, the Virtual Center sensor calculates systemboardUUID with the algorithm at this link so that it matches the Windows Computer System sensor results and properly merge.
- See the vmware knowledge base website
  - View by Article ID for **1880**
  - Search the Vmware Knowledge Base (KB) for **WMI UUID Differs from VirtualCenter UUID** http://kb.vmware.com

10 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

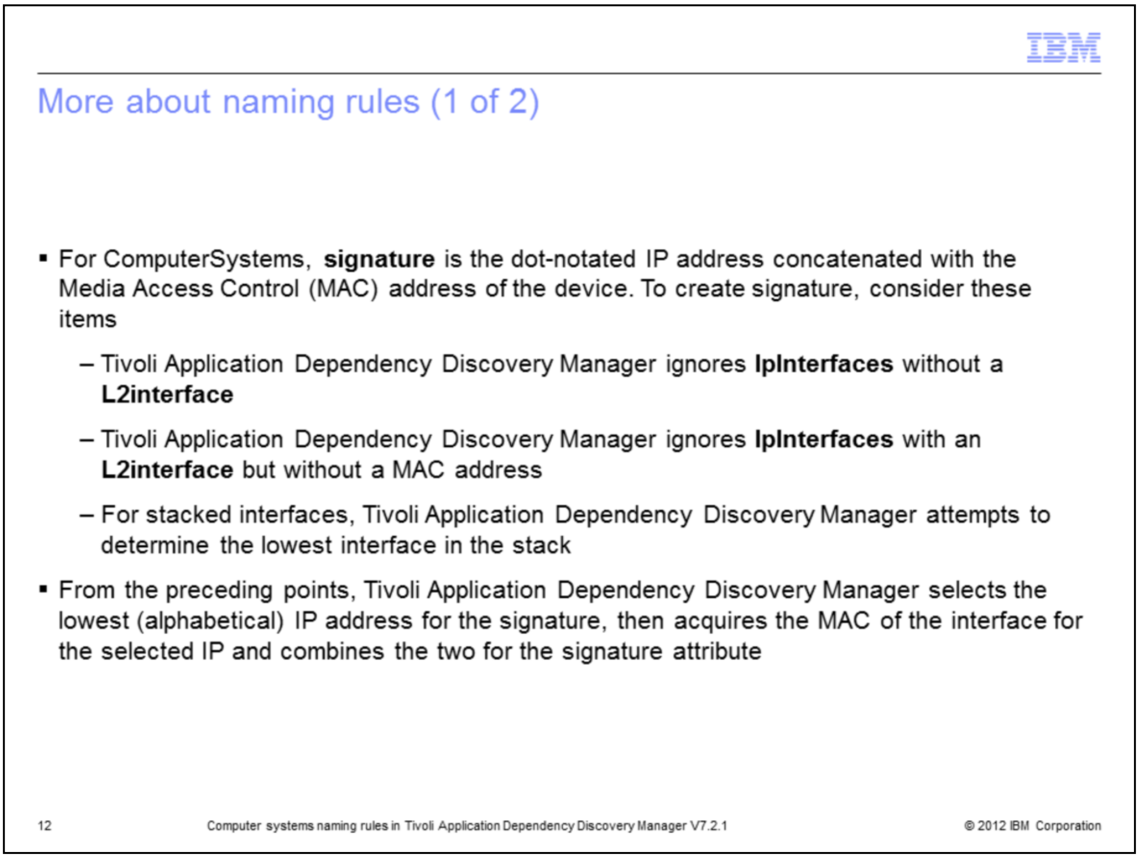
It is important to note that virtual machines can be considered unique by way of make, model, serial number, plus their VMID (Virtual Machine ID). However, with **Vmotion** in place, **SystemBoardUUID** can be the only constant attribute to preserve a single virtual host. This attribute is available on the target system as UUID but does not match the Virtual Center value for the UUID because of format differences between the Microsoft interface and Virtual Centers. For this reason, the Virtual Center sensor calculates **systemboardUUID** with the algorithm at the link shown so that it matches the Computer System sensor results and properly merges.



What to understand about these naming rules

- It is important to realize that if any rule matches then the computers merge; priority does not matter
- For duplicates to occur, it generally means that none of the rules were matched; for example, the signature changes and there are no other naming rule attributes discovered
- If computers are **missing** even after their OS sensor ran successfully, it typically means one or more of the naming rules matched a different computer and they merged

There are a few things you need to understand about naming rules. If the first point on this slide sounds like a broken record it is because there is a common confusion that priority plays a part in merge and reconciliation activity. As noted earlier, priority is just an order of evaluation, objects can merge if any rule matches. Duplicates occur when none of the rules match. If computers are missing after successful discovery, this is often because one or more naming rules matched a different computer and they merged.

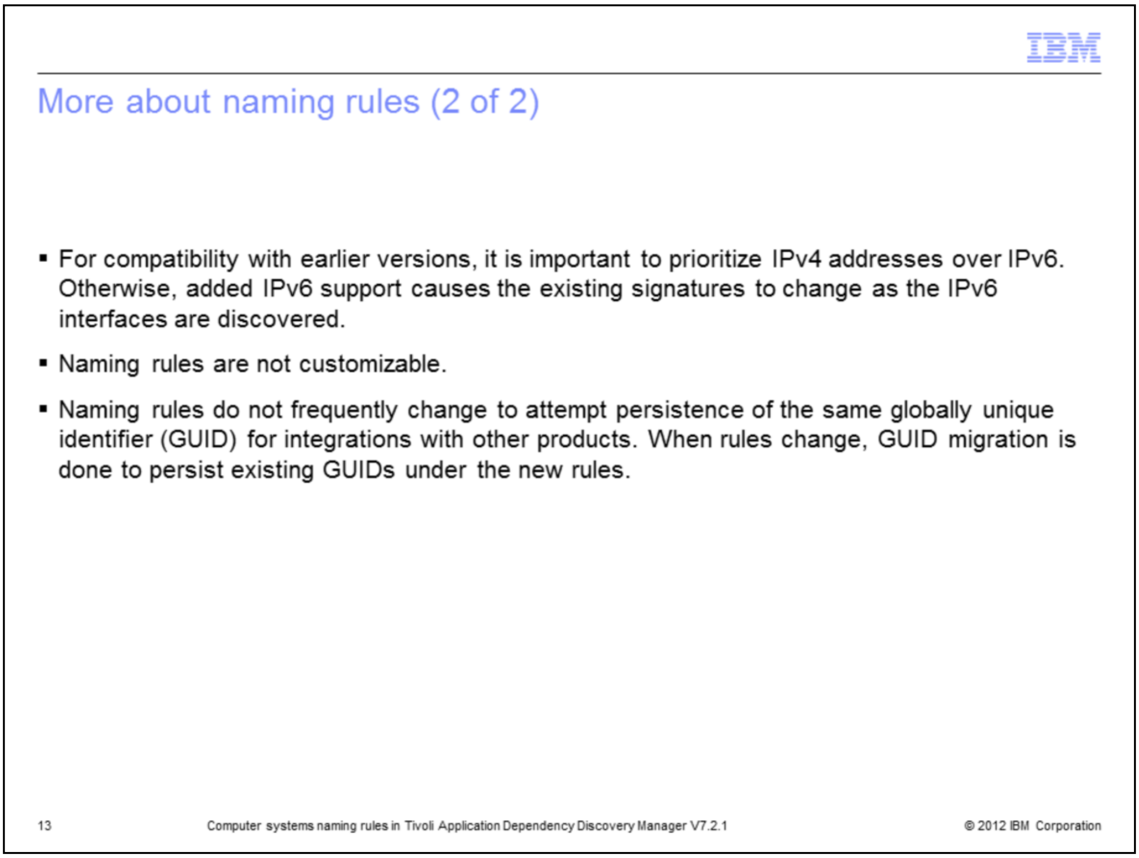


IBM

More about naming rules (1 of 2)

- For ComputerSystems, **signature** is the dot-notated IP address concatenated with the Media Access Control (MAC) address of the device. To create signature, consider these items
  - Tivoli Application Dependency Discovery Manager ignores **IpInterfaces** without a **L2interface**
  - Tivoli Application Dependency Discovery Manager ignores **IpInterfaces** with an **L2interface** but without a MAC address
  - For stacked interfaces, Tivoli Application Dependency Discovery Manager attempts to determine the lowest interface in the stack
- From the preceding points, Tivoli Application Dependency Discovery Manager selects the lowest (alphabetical) IP address for the signature, then acquires the MAC of the interface for the selected IP and combines the two for the signature attribute

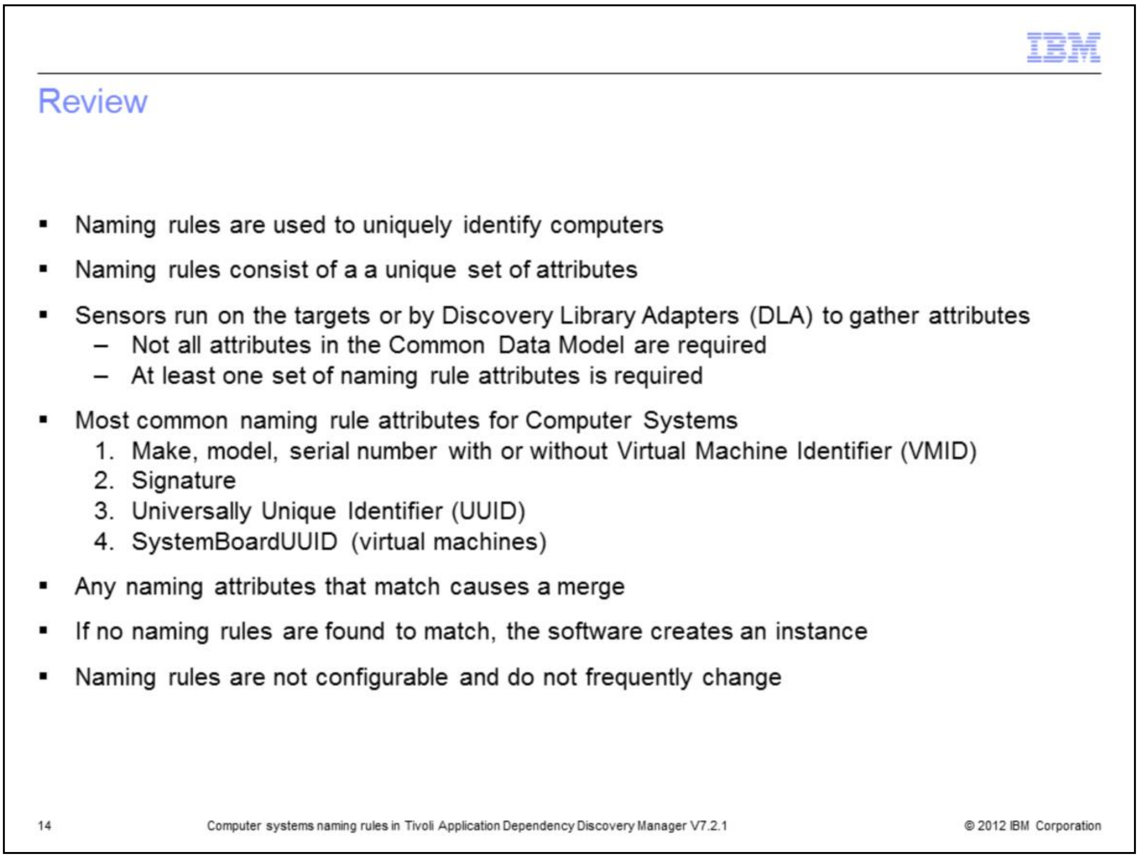
One of the rules that is frequently an issue with duplicates is signature. If Tivoli Application Dependency Discovery Manager is unable to obtain the serial number or UUID with the discovery, often signature is the only naming attribute collected. This is especially true for Solaris targets where there is no default method to obtain serial number. Given this, it is important to understand what signature represents. Signature is the dot-notated IP address concatenated with the MAC address of the device. For the signature, Tivoli Application Dependency Discovery Manager gathers interface data with WMI (Windows) or ifconfig (UNIX). There is a relationship from an ipInterface to an l2Interface. If that relationship is not there, or if it is there but the related L2interface does not have a hardware address (MAC), then Tivoli Application Dependency Discovery Manager does not consider the IP address of that interface as a candidate for signature. If the IpInterfaces are stacked Tivoli Application Dependency Discovery Manager attempts to determine the lowest interface in the stack. From these, Tivoli Application Dependency Discovery Manager selects the lowest (alphabetic) IP address. Tivoli Application Dependency Discovery Manager then gets the MAC of the L2Interface of the IpInterface of the IP that was selected and combines the two to be the signature.



More about naming rules (2 of 2)

- For compatibility with earlier versions, it is important to prioritize IPv4 addresses over IPv6. Otherwise, added IPv6 support causes the existing signatures to change as the IPv6 interfaces are discovered.
- Naming rules are not customizable.
- Naming rules do not frequently change to attempt persistence of the same globally unique identifier (GUID) for integrations with other products. When rules change, GUID migration is done to persist existing GUIDs under the new rules.

There are just a few more points to understand about naming rules. Tivoli Application Dependency Discovery Manager does prioritize IPv4 over IPv6 for compatibility. Naming rules are not customizable within Tivoli Application Dependency Discovery Manager. Naming rules do not change frequently as changes require a GUID migration to maintain consistency with external integrations. Such changes are typically done in major releases.



Review

- Naming rules are used to uniquely identify computers
- Naming rules consist of a a unique set of attributes
- Sensors run on the targets or by Discovery Library Adapters (DLA) to gather attributes
  - Not all attributes in the Common Data Model are required
  - At least one set of naming rule attributes is required
- Most common naming rule attributes for Computer Systems
  1. Make, model, serial number with or without Virtual Machine Identifier (VMID)
  2. Signature
  3. Universally Unique Identifier (UUID)
  4. SystemBoardUUID (virtual machines)
- Any naming attributes that match causes a merge
- If no naming rules are found to match, the software creates an instance
- Naming rules are not configurable and do not frequently change

14 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

To summarize: Naming rules are used to uniquely identify computers within Tivoli Application Dependency Discovery Manager.

Naming rules consist of a set of attributes which are considered unique by industry standard.

For Tivoli Application Dependency Discovery Manager systems, such attributes are gathered either by sensors or by a discovery library adapter (**DLA**). Not all attributes in the CDM are required, but at least one set of naming rule attributes is required.

For Computer Systems, these are the most common attributes gathered to uniquely identify a computer system:

1. **Make, model, serial number** (this includes **VMID** if the computer is virtual).

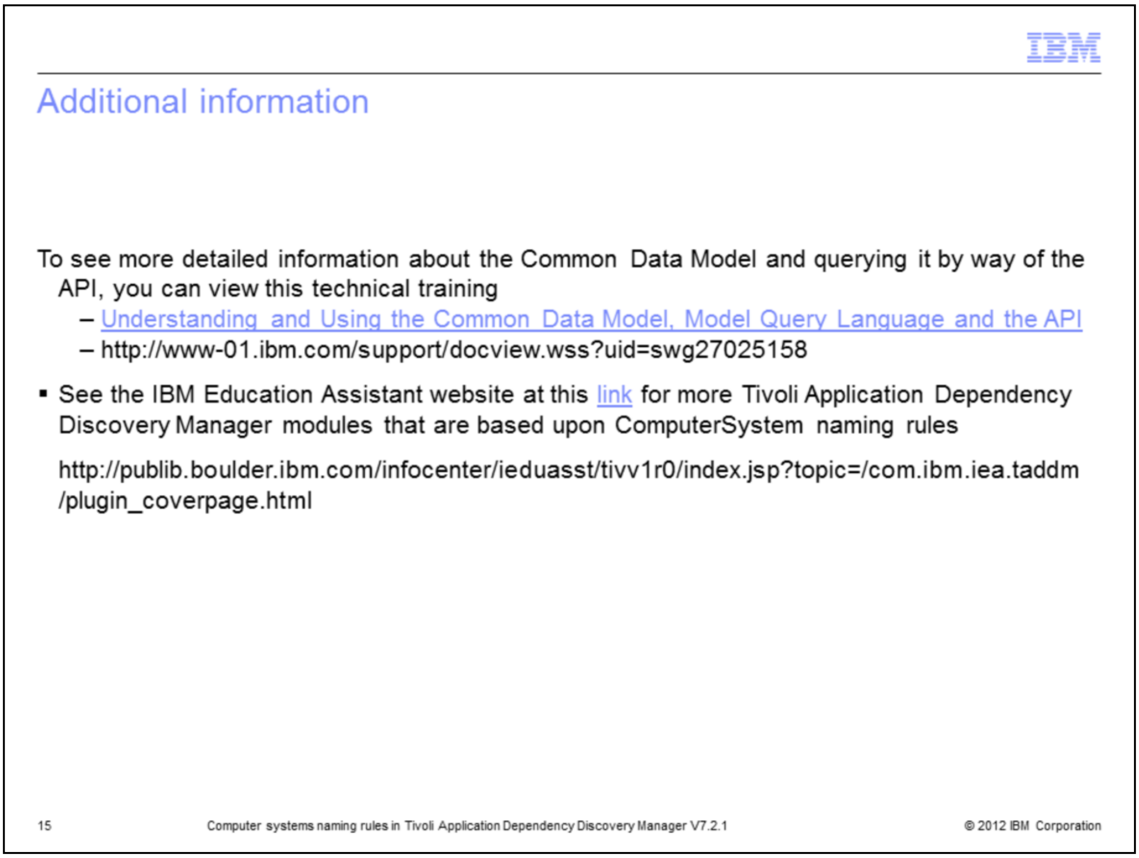
2. The **computer system signature** which was described in detail earlier in this presentation.

3. The **UUID**, universally unique identifier assigned by the manufacturer.

4. The **systemboardUUID**, typically used by virtual machines.

If the naming attributes of a computer system match a previously discovered computer system, the two merge together. If no naming rules match, the software creates a new instance.

Naming rules are not configurable and do not frequently change.



IBM

Additional information

To see more detailed information about the Common Data Model and querying it by way of the API, you can view this technical training

- Understanding and Using the Common Data Model, Model Query Language and the API
- http://www-01.ibm.com/support/docview.wss?uid=swg27025158

- See the IBM Education Assistant website at this link for more Tivoli Application Dependency Discovery Manager modules that are based upon ComputerSystem naming rules

  http://publib.boulder.ibm.com/infocenter/ieduasst/tivv1r0/index.jsp?topic=/com.ibm.iea.taddm/plugin_coverpage.html

15 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

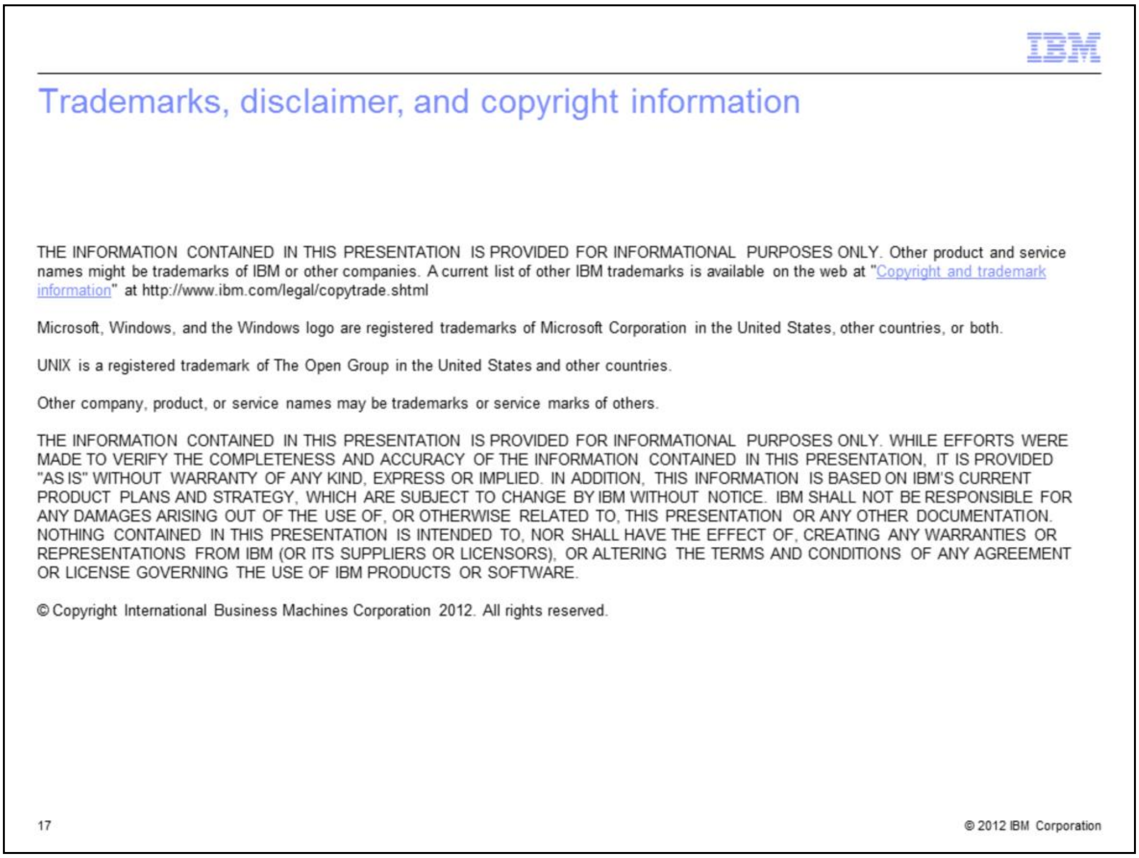
You can find additional detailed training on this topic and usage of the API on the Tivoli Application Dependency Discovery Manager Support site in technote 7025158. There are additional training modules related to this topic on this IBM Education Assistant website.

Summary

Now that you have completed this module, you can describe how computer system objects are uniquely identified within Tivoli Application Dependency Discovery Manager

16 Computer systems naming rules in Tivoli Application Dependency Discovery Manager V7.2.1 © 2012 IBM Corporation

Now that you have completed this module, you can describe how computer system objects are uniquely identified within Tivoli Application Dependency Discovery Manager.



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.