BMC Atrium CMDB 7.5.00

Data Modeling Guide

January 2009

Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address	BMC SOFTWARE INC 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA	Telephone	713 918 8800 or 800 841 2031	Fax	713 918 8000

Outside United States and Canada

Telephone	(01) 713 918 8800	Fax	(01) 713 918 8000

If you have comments or suggestions about this documentation, contact Information Development by email at doc_feedback@bmc.com.

Restricted Rights Legend

Customer Support

You can obtain technical support by using the Support page on the BMC Software website or by contacting Customer Support by telephone or email. To expedite your inquiry, please see "Before Contacting BMC Software."

Support Website

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at `http://www.bmc.com/support_home`. From this website, you can

- Read overviews about support services and programs that BMC Software offers.
- Find the most current information about BMC Software products.
- Search a database for problems similar to yours and possible solutions.
- Order or download product documentation.
- Report a problem or ask a question.
- Subscribe to receive email notices when new product versions are released.
- Find worldwide BMC Software support center locations and contact information, including email addresses, fax numbers, and telephone numbers.

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813 or send an email message to `customer_support@bmc.com`. (In the Subject line, enter `SupID:<yourSupportContractID>`, such as `SupID:12345`.) Outside the United States and Canada, contact your local support center for assistance.

Before Contacting BMC Software

Have the following information available so that Customer Support can begin working on your issue immediately:

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack
  - System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level
- Sequence of events leading to the problem
- Commands and options that you used
- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system, such as `file system full`
  - Messages from related software

Contents

User's Guide

Data Modeling Guide

The BMC Atrium Configuration Management Database (CMDB) enables you to store and manage information about products and services that are in your environment. This white paper describes how to model business entities available in BMC Atrium 7.5.00 by using the Common Data Model (CDM) and extensions to that model, and explores recommended practices for using new entities effectively.

The BMC Atrium CMDB uses the term *class* to describe a configuration item (CI) or relationship classification. Each CI is partially classified using some common *attributes* that describe the *base class* (`BMC_BaseElement`). Specific details about each class of CI are described by attributes of *subclasses* of `BMC_BaseElement`. Relationships are also modeled as a base relationship class (`BMC_BaseRelationship`) with subclasses for different types of relationships.

This paper illustrates how to use the classes that BMC provides for BMC Atrium CMDB to model a particular business entity, focusing on how you use the entire model rather than on general information about a class or attribute. Although descriptions of classes and attributes are provided to give you context when determining how to model CIs, detailed information such as syntax and the type of attribute is not specified. For that level of information, see the BMC Atrium 7.5.00 Data Model Help.

As a provider of BMC Atrium CMDB data, BMC Discovery products can discover large amounts of configuration data for use by data consumers. BMC Discovery products are natural enablers for the creation of service models because they can discover many of the components, or CIs, that ultimately make up the service models. These components include:

- Computer systems (including servers, routers, physical and virtual systems, and operating systems)
- Applications
- Software servers (including specialized elements such as SAP®, Sun™, Siebel, and mainframe infrastructure components)
- Databases
- Business process definitions
- Network elements

- Relationships between these CIs

Content and format of this guide

This user's guide originated and was released as a white paper for BMC Atrium CMDB 2.1.00. It has been formatted as a user's guide primarily so that it can be included in the documentation set for users of BMC Atrium CMDB 7.5. As such, the formatting may not mirror that of standard BMC user's guides. This will be fixed in a future version.

Also, although the content of the guide reflects version 7.5 of the CDM, there are some modeling concepts that have not yet been documented in support of the latest version. BMC Software plans to update the content of this guide in a future version to include a greater number of best practices that take advantage of the current data model.

Conventions used in this guide

This user's guide applies the following conventions to explain BMC Atrium CMDB concepts in both textual and graphical formats.

Terminology

In many cases you will be modeling an entity using classes from the CDM, but you might also model part of that same entity using an extension to the CDM. For models that require extensions to the CDM, the term *data model* is used. This paper is organized so that the entities are introduced first in each section, including the recommended practice for that implementation. Any classes and attributes that can be included in the deployment of these business entities in an IT infrastructure are described in an architectural diagram. Where appropriate, recommendations are provided for setting specific attributes for a given class.

Attributes are defined as either *key* or *additional*. Key attributes are those that BMC recommends that you populate for a given class to model a specific CI. Additional attributes are optional attributes that you can populate to further classify a CI or relationship.

Differentiating Name and ShortDescription attributes

A common misconception is that the caption for the CI on user interfaces and reports is represented by the `Name` attribute, when it is actually the `ShortDescription` attribute. In diagrams in this paper, the names that appear are not from the `Name` attribute, they are the `ShortDescription` attribute (which is usually just a user interface caption). Also, in modeling recommendations, `ShortDescription` is the more user-friendly label, and should always be provided and set with a value that makes sense to an end user.

Diagrams

Illustrative model diagrams help explain the concepts and modeling recommendations in this paper, and also show how you might model an entity in a real-world business environment. In these diagrams, CIs are represented by single-line boxes that contain attributes of the class or its parent class. Where applicable, key attributes are shown in the box that depicts a specific class and, in some cases, include the recommended value of those attributes.

NOTE

Illustrative diagrams are just examples, and might not reflect every possible class, attribute, or relationship that you would use for modeling all types of the represented object.

Relationships represented in illustrative model diagrams

In the diagrams, boxes illustrate how CIs in your environment should be mapped to the CDM, or how to extend the CDM to create your own data model. Lines are used to represent the type and direction of the relationship.

NOTE

Relationships in the diagrams in this white paper are illustrated using Unified Modeling Language (UML) standards. The UML notation may not be consistent with the BMC Atrium 7.5.00 user interfaces (UI). Some of this discrepancy is due to the absence of a direct UML equivalent to the relationships represented, and some of it is the lack of alignment between the CDM, the UI, and UML standards.

Although discrepancies may exist between the UML standards and the BMC Atrium CMDB UI, changes in the UI for future releases of BMC Atrium CMDB will enable the UI to more closely align with UML. In this white paper, the conventions applied to the diagrams enable you to easily distinguish which relationship is used in a modeling scenario, regardless of how one might view them in the product. For example, one major difference between UML standards and the BMC Atrium CMDB UI is that, in the UI, an arrow is always used to represent the source and destination of the relationship, whereas in UML, it is not. Therefore, in this white paper, the diagrams more closely align with UML so that you can understand the semantic of the modeling scenario in the context of the corresponding best-practice modeling recommendations.

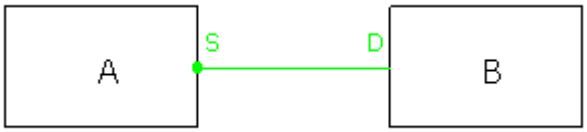
Although UML does not standardize colors in its rendering of relationships, they are used in the diagrams to help you easily distinguish at a glance which relationship type is recommended to model an example business object. Additionally, the source and destination of each relationship are represented by the letters S and D, respectively. The following section illustrates examples of each relationship type.

Examples of dependency relationships (arrow)

Dependency relationships are represented by dashed red lines, and contain an arrow to show the direction of the relationship. In a BMC_Dependency relationship, the arrow starts at A, the dependent (Destination), and ends at B, the antecedent (Source) of the relationship. Entity A is dependent on Entity B.

Example of a collection relationship (circle)

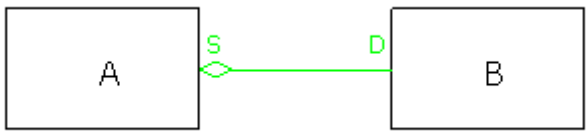
A BMC_MemberOfCollection relationship is represented by green lines with circle tips, as illustrated in the following diagram:



A is the collection class (Source), and B is the member class (Destination). The circle represents a collection relationship, where the collection class uses properties of the member class.

Example of a component relationship (diamond)

In a component relationship, the source CI is a group that has a component or part, its destination. Entity A is a group (Source) that has a component B (Destination). In diagrams, component relationships are represented by green lines with diamond tips.



Cardinality in relationships

Every relationship class has a cardinality that defines how many instances of the source class can be related to each instance of the destination class and vice versa. Where cardinality is specified in the diagrams, it is shown at the ends of the relationship lines as one of the following types:

- 1:1 (one to one)

- 1:* (one to many)
- *:* (many to many)

Weak relationships

Where a *weak relationship* exists between two instances, that relationship is indicated by the letter W in the illustrative model diagrams. If the relationship is a weak relationship, its destination member, called the *weak* member, cannot exist without its source member, called the *strong* member. A weak relationship creates a logical composite object consisting of both member CIs.

BMC Atrium Core documentation

The following table lists the documentation available for BMC Atrium Core.

Unless otherwise noted, softcopy documentation is available on the BMC Atrium Core documentation media (DVD or Electronic Product Download bundle) and on the BMC Customer Support site at http://www.bmc.com/support_home.

Title	Document provides	Audience	Format
BMC Atrium CMDB 7.5.00 Administrator's Guide	Information about setting permissions, configuring federation, modifying the data model, configuring an impact model, and other administrative tasks in BMC Atrium CMDB.	Administrators	Print and PDF
BMC Atrium CMDB 7.5.00 Common Data Model Diagram	Hierarchical diagram of all classes in the Common Data Model (CDM) including unique attributes and applicable relationships.	Administrators	Print and PDF
BMC Atrium CMDB 7.5.00 Data Model Help	Description and details of superclasses, subclasses, attributes, and relationship classes for each class. Contains only information about the Common Data Model at first, but you can update it to include information about data model extensions that you install.	Administrators	HTML (product media only)
BMC Atrium CMDB 7.5.00 Data Modeling Guide	Best practices for using the classes that BMC provides for BMC Atrium CMDB (both the CDM and extensions) to model complex business entities, focusing on the use of multiple related CIs to model an entity rather than on general information about a class or attribute.	Administrators	Print and PDF
BMC Atrium CMDB 7.5.00 Javadoc™ Help	Information about Sun™ Java™ classes, methods, and variables that integrate with BMC Atrium CMDB.	Programmers	HTML (product media only)
BMC Atrium CMDB 7.5.00 Normalization and Reconciliation Guide	Information about normalizing data in BMC Atrium CMDB and reconciling CIs from different data providers into a single production dataset.	Administrators	Print and PDF

Title	Document provides	Audience	Format
BMC Atrium CMDB 7.5.00 Online Help	Help for using and configuring BMC Atrium CMDB, available by clicking Help in the product interface.	Users and administrators	Product Help (available from Help links after installed)
BMC Atrium CMDB 7.5.00 User's Guide	Information about using BMC Atrium CMDB, including searching for and comparing CIs and relationships, relating CIs, viewing history, running impact simulations, and viewing federated data.	Users	Print and PDF
BMC Atrium Core 7.5.00 Concepts and Planning Guide	Information about CMDB concepts and high-level steps for planning and implementing your BMC Atrium Core implementation.	IT leaders and administrators	Print and PDF
BMC Atrium Core 7.5.00 Developer's Reference Guide	Information about creating API programs using C and web services API functions and data structures.	Administrators and programmers	Print and PDF
BMC Atrium Core 7.5.00 Installation Guide	Information about installing, upgrading, and uninstalling BMC Atrium Core features.	Administrators	Print and PDF
BMC Atrium Core 7.5.00 Master Index	Combined index of all guides.	Everyone	Print and PDF
BMC Atrium Core 7.5.00 Product Catalog and DML Guide	Information about configuring the Product Catalog and DML, updating vendor data, and connecting the Product Catalog to BMC Configuration Management and BMC Remedy AR System.	Administrators	Print and PDF
BMC Atrium Core 7.5.00 Product Catalog and DML Online Help	Help for using and configuring BMC Atrium Product Catalog, available by clicking Help in the product interface.	Users and administrators	Product Help (available from Help links after installed)
BMC Atrium Core 7.5.00 Release Notes	Information about new features, known issues, and other late-breaking information.	Everyone	Print and PDF
BMC Atrium Core 7.5.00 Troubleshooting Guide	Information about resolving issues with BMC Atrium Core components, including API, filter and console error messages and their solutions.	Administrators, programmers, and BMC Support personnel	Print and PDF
BMC Atrium Integration Engine 7.5.00 ADK Developer's Guide	Information about how to build adapters that can transfer information between an external data store and either AR System forms or BMC Atrium CMDB.	Developers	Print and PDF
BMC Atrium Integration Engine 7.5.00 Online Help	Information about how to create data exchanges and data mappings and to configure administrative options, available by clicking Help in the product interface.	Users and Administrators	Product Help (available from Help links after installed)

Title	Document provides	Audience	Format
BMC Atrium Integration Engine 7.5.00 User's Guide	Information about how to create data exchanges, data mappings, define rules and queries, activate event-driven data exchanges, define connection settings, and conceptual information about BMC Atrium Integration Engine.	Users and Administrators	Print and PDF
Mapping Your Data to BMC Atrium CMDB 7.5.00 Classes	Mappings of common IT objects to the appropriate class in which to store them, whether part of the Common Data Model or an extension. Also includes information about further categorizing instances using key attributes and best practices for creating normalized relationships.	Administrators	Spreadsheet, print and PDF

You can view PDF documents using Acrobat Reader, which is available through Adobe at www.adobe.com.

Modeling computer systems, servers, workstations, and network devices

This section describes how to use the CDM to model computer systems (servers, workstations, and network nodes such as routers, switches, and hubs). It details the classes, relationships, and attributes used to model computer systems, operating systems, hardware components, software inventory and patches, access points, and network interfaces.

For information on modeling applications, including modeling runtime versus installed aspects of applications, see “Modeling applications” on page 31.

Logical identity of BMC_ComputerSystem

BMC_ComputerSystem is a class that stores CIs relating to collections of managed system elements. This is the primary class that you use to model the computers in your organization. You can use the attributes in this class to identify the purpose of each computer CI in your organization.

For example, the class contains several attributes that represent any network-addressable system, such as a server, a workstation, or a network device (router, switch, hub, load balancer, firewall, and so forth), as well as mainframes, printers, and virtual systems.

The following key attributes identify an instance of BMC_ComputerSystem.

Attribute	Description
Name, NameFormat	Use this attribute to identify a computer system. The Name attribute should be a unique instance identifier that may not be Human Readable. Because multiple valid naming conventions may exist and can be used according to specific contexts, set the NameFormat attribute with a value indicating the Heuristic used to generate the Name value. For example, in some cases, an instance of BMC_ComputerSystem will be identified by an external DNS name (a name configured in a DNS server). In other cases, a static IP address will be used. The naming conventions for NameFormat are: ■ IP—a valid IP address (decimal bytes delimited with dots). ■ DNS—a fully qualified host name, formatted as a HostName and a DomainName delimited with dots (the DomainName can also be made of multiple components delimited with dots). ■ TOKEN—Name holds a value defined by the TokenId (see "Additional Attributes for BMC_ComputerSystem" for more information on the TokenId).
Domain	Use this attribute to set the domain name of the computer, as known by the end points. This value must be set according to BMC N11n guidelines that specifies the algorithms and methods required to obtain the correct values.
HostName	Use this attribute to set the local name of the computer, as known by the end points. This value must be set according to BMC N11n guidelines that specifies the algorithms and methods required to obtain the correct values.
SerialNumber	Use this attribute to set the serial number of the computer.

Key attributes of BMC_ComputerSystem

The following attributes further describe the role of an instance of BMC_ComputerSystem.

<table>
<tr><th>Attribute</th><th colspan="2">Description</th></tr>
<tr><td>CapabilityList</td><td colspan="2">Use this attribute to list the main functions that the computer can perform.
This is a character attribute in which you can enter any value listed in the description. You can enter more than one of these values; however, make sure that multiple values are delimited by commas. A computer system can be dedicated to a single function, such as printing, routing, or switching packets, or it can perform several functions. Typically, the PrimaryCapability attribute is set to the first value specified in CapabilityList.
The following list illustrates the functions and values to assign to a CapabilityList attribute depending on the function of the computer.</td></tr>
<tr><td></td><td>■ Not Dedicated—0
■ Unknown—1
■ Other—2
■ Storage—3
■ Router—4
■ Switch—5
■ Layer 3 Switch—6
■ Central Office Switch—7
■ Hub—8
■ Access Server—9
■ Firewall—10
■ Print—11</td><td>■ I/O—12
■ Web Caching—13
■ Server—14
■ Management —15
■ Block Server —16
■ File Server —17
■ Mobile User Device —18
■ Repeater—19
■ Bridge/Extender—20
■ Gateway —21
■ LoadBalancer—22</td></tr>
<tr><td>PrimaryCapability</td><td colspan="2">Use this attribute to describe the main function that the computer performs.
By convention, PrimaryCapability is the first item in the CapabilityList attribute.</td></tr>
<tr><td>ShortDescription</td><td colspan="2">Use this attribute to specify a short description for the instance when the value of the Name attribute is encoded. ShortDescription should always be provided and set with a value that makes sense to an end user.</td></tr>
</table>

For example, a server with active firewall capabilities could have the values 14 (Server) or 10 (Firewall) for CapabilityList. PrimaryCapability would be set to Server if this is the main function of the system. However, a switch device would have CapabilityList = 5 (Switch) and PrimaryCapability = 5.

Additional attributes for BMC_ComputerSystem

The following table describes attributes that provide additional information about an instance of BMC_ComputerSystem.

Attribute	Description
Description	Use this attribute to specify all the functions that the computer system can perform.
DHCPUse	Use this attribute to indicate whether the system is configured to use DHCP: Enabled = configured to use DHCP Disabled = not configured to use DHCP
ManufacturerName	Use this attribute to describe the company that manufactured the computer.
Model	Use this attribute to describe the model of the computer.
OwnerContact	Use this attribute to specify information about how the primary system owner can be reached (such as phone number or email address).
OwnerName	Use this attribute to specify the name of the primary system owner.
TokenId	Use this attribute to specify a unique identifier populated by BMC Discovery products and used by the Reconciliation Engine (of the BMC Atrium CMDB) to identify instances.
TotalPhysicalMemory	Total physical memory, in kilobytes.

IMPORTANT

The Dimensions attribute, also available for the BMC_ComputerSystem class, should not be used. This attribute will be deprecated in a future version of BMC Atrium CMDB.

See the BMC Atrium 7.5.00 Data Model Help for more information about specific attributes.

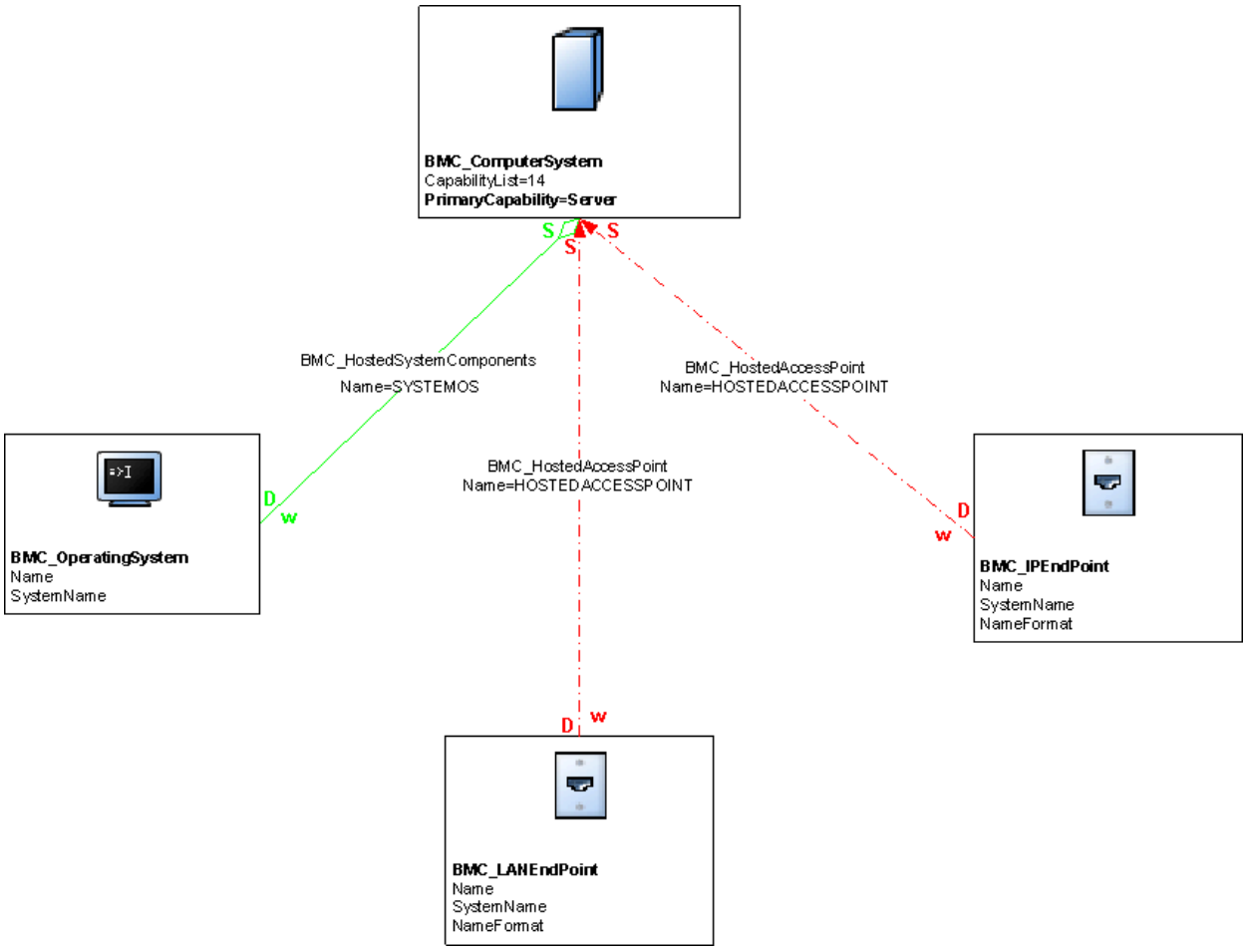
Modeling computer systems

Computer systems are parent objects that may be represented as an aggregation of component parts (such as operating systems, hardware, software inventory, or network addresses) that are child instances related to the BMC_ComputerSystem instance.

Systems provide computing capabilities and aggregate one or more of the following elements: file systems, operating systems, processors, and memory (including volatile and nonvolatile storage). Therefore, additional information about a computer system might not be part of a BMC_ComputerSystem instance but be available from instances of other classes connected to the BMC_ComputerSystem instance through relationships.

For example, Figure 1-1 represents a model for a server, a network-addressable computer system.

Figure 1-1: Illustrative model of a server



Servers, workstations, network devices (such as routers, switches, hubs, load balancers, or firewalls) are all instances of `BMC_ComputerSystem`, the class representing all network addressable systems. `BMC_ComputerSystem` represents an entity made up of component parts that operate as a functional whole.

The `PrimaryCapability`attribute is crucial to identifying whether a specific instance is a server, a router, or something else. BMC Atrium CMDB planners might use the `PrimaryCapability` attribute to define a vendor-specific switch used in their network, making it easy to import this data from a vendor's environment as an industry-standard item in their BMC Atrium CMDB.

Modeling software inventory and patches

Software inventory represents the products, packages, and patches that are installed on a computer. BMC_Product represents instances of installed products, whereas BMC_Patch represents instances of patches (operating system patches and product patches). BMC_Package stores information about containers or collections of related software. As a subclass of BMC_Product, BMC_Package can be used to represent suite of products. For example, you can represent Microsoft Office as an instance of BMC_Package, whereas Microsoft Word would be an instance of BMC_Product. Figure 1-2 illustrates an example model of a server with two installed products.

Figure 1-2: Illustrative model of a software inventory containing two installed products

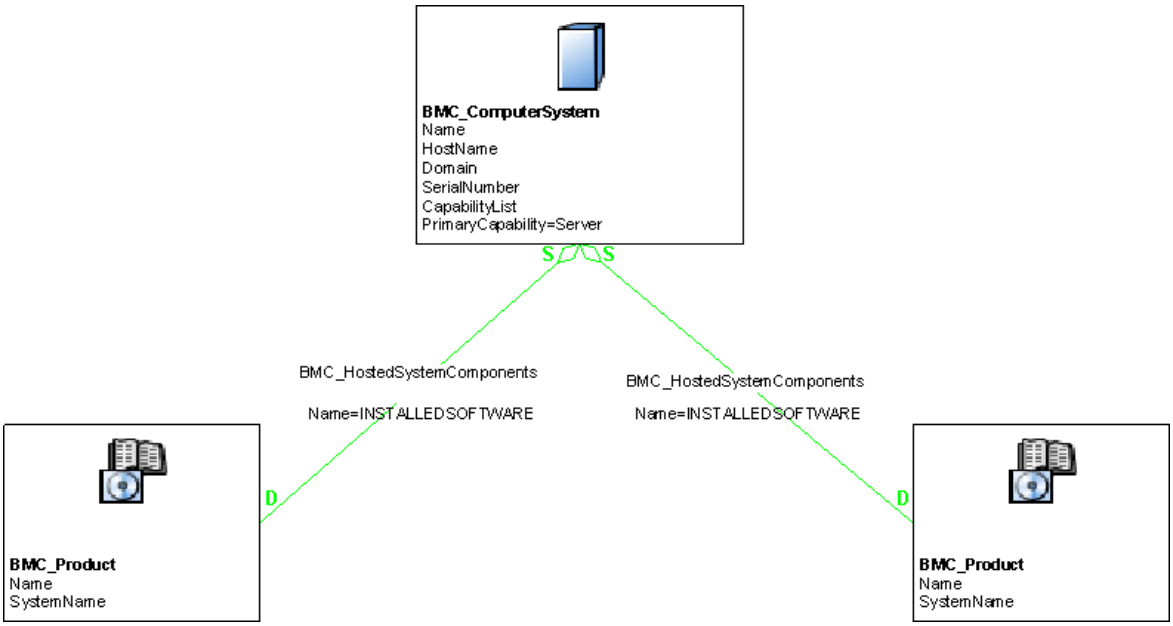
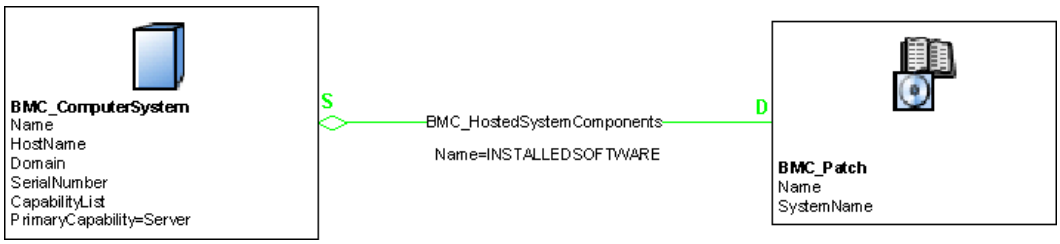


Figure 1-3 illustrates an example model of a server with one installed patch.

Figure 1-3: Illustrative model of a software inventory with a patch



You might not want to model patches in your BMC Atrium CMDB in all cases. For example, you might store patches for servers in the BMC Atrium CMDB, but it might not be necessary to do so for desktops and routers.

Both BMC_Product and BMC_Patch are subclasses of BMC_Software and BMC_SystemComponent. You should associate each instance of a product or of a patch to the parent instance of BMC_ComputerSystem by the BMC_HostedSystemComponents relationship. When modeling software inventory, be aware that the BMC_Product class captures only installed products or applications, not runtime aspects.

For more information about modeling runtime applications and using instances of BMC_Product for application modeling, see “Modeling applications” on page 31.

Logical identity of BMC_ComputerSystem (for products or patches)

Like any child instance of BMC_ComputerSystem, a product or a patch is identified by the Name attribute in conjunction with the SystemName attribute that represents the name of the computer instance. Thus, the Name attribute represents the local name of the CI in the context of the computer that is hosting it, as described in the following table.

Attribute	Description
Name	Use this attribute to specify the name of the child instance in the context of the parent instance of BMC_ComputerSystem.
SystemName	Use this attribute to specify the name of the computer instance. This must be the same as the parent instance of the BMC_ComputerSystem Name attribute. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the product or patch.

Additional attributes of BMC_ComputerSystem (for products or patches)

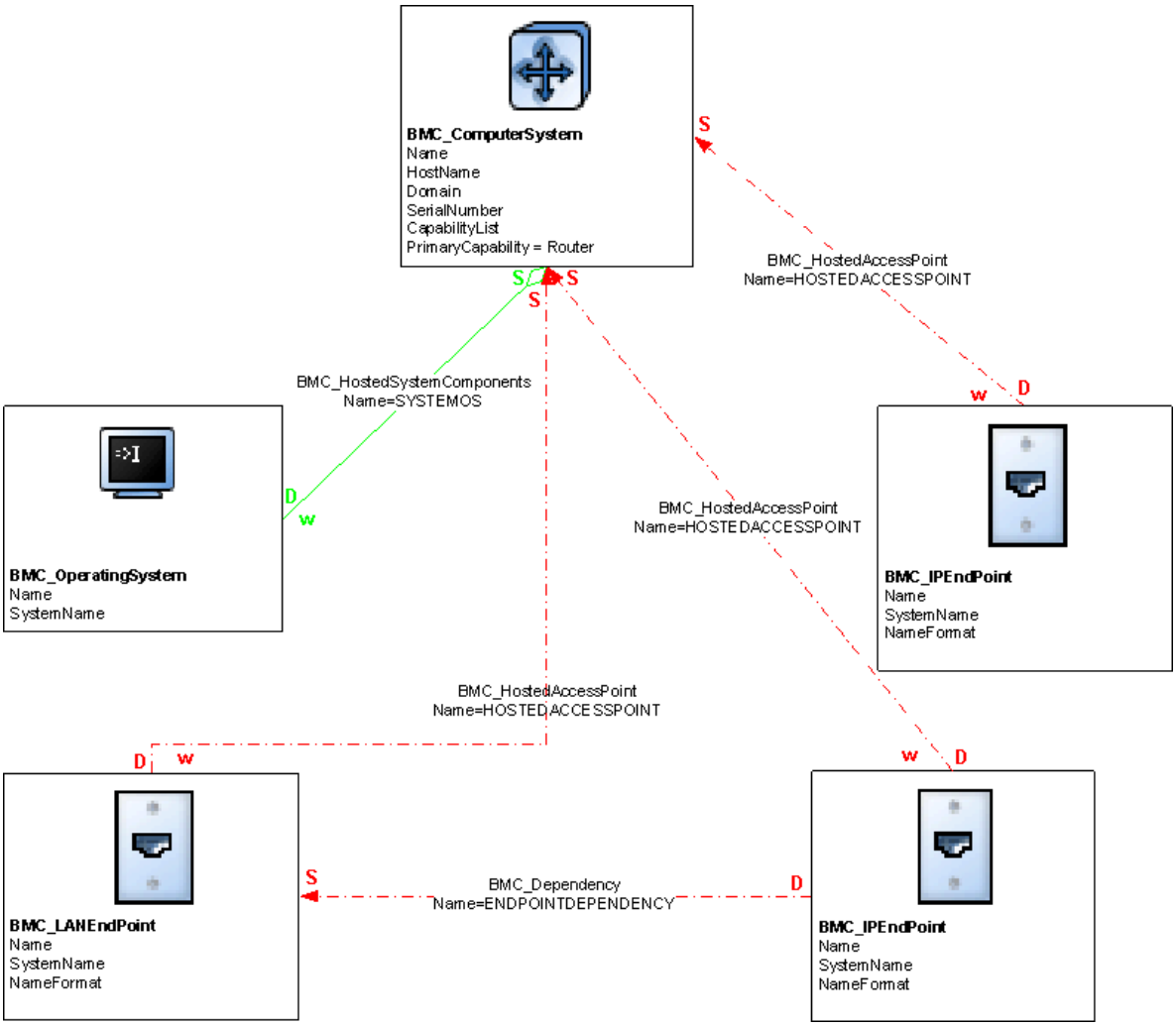
The following table describes attributes that provide additional information about products and patches.

Attribute	Description
Description	Use this attribute to specify the description for the component.
ManufacturerName	Use this attribute to specify the company that manufactured the component.
SerialNumber	Use this attribute to specify the serial number of the component.
ShortDescription	Use this attribute to specify the short description for the component.
PatchNumber	Use this attribute to specify the version number of the patch.
VersionNumber	Use this attribute to specify the version number of the component.

Modeling routers

Routers are modeled using the `BMC_ComputerSystem` class by setting the `PrimaryCapability` attribute to Router. Figure 1-4 illustrates an example model of a network router.

Figure 1-4: Illustrative model of a router



In the model, two `BMC_IPEndpoint` classes are used to represent the interfaces to the router.

Modeling virtual systems

Virtual systems represent one or more virtual machines that are hosted by a physical computer. A virtual system has the same relationships to subcomponents and applications that a physical system does. In other words, a virtual system has an operating system (such as Windows or UNIX®), network addresses, and software. The major difference is that these subcomponents, although captured as regular CIs, are all virtual.

When modeling virtualization in your environment, represent the physical computer system using the BMC_ComputerSystem class, and the virtualization software (such as Hypervisor or virtualization software), using the BMC_VirtualSystemEnabler class. Use BMC_VirtualSystemSettingData and BMC_ResourceAllocationSettingData (both new classes introduced in BMC Atrium 7.5.00) to represent virtual system settings, and BMC_ResourcePool (also new in this version) to model resource pools. More information on these classes and example illustrative models will be provided in a future version of this guide.

Logical identity of BMC_ComputerSystem (for virtual systems)

You model virtual systems as instances of BMC_ComputerSystem. You should follow the same naming rules as for an instance of BMC_ComputerSystem class. For more information about this class, see "Logical identity of BMC_ComputerSystem" on page 13. Key attributes for defining virtual systems using the BMC_ComputerSystem class are described in the following table.

Attribute	Description
VirtualSystemType	Use this attribute to specify the type of virtual machine. Values are Other (0), Unknown (1), PR/SM (2), z/VM (3), VMWare (4), Zen (10), Hyper-V (15), Solaris Container (20), , VPar (25), NPar (30) and LPar (35) .
isVirtual	Use this attribute to specify whether the instance is virtual or physical. Values are No (0) or Yes (1).

NOTE

To ensure correct reconciliation with data created by BMC Software products, use NULL instead of No for the isVirtual attribtue to represent an instance that is not virtual.

Logical identity of BMC_VirtualSystemEnabler

The BMC_VirtualSystemEnabler class stores information about software that enables a collection of virtual computer systems to run on a single physical computer system (for example, VMware). This class is used to capture the virtualization OS, such as operating systems that run virtual machines (including VMware images, Solaris zones, AIX logical partitions, HP-UX virtual partitions, and so forth).

The BMC_VirtualSystemEnabler class is associated to the parent computer system instance by the BMC_HostedSystemComponents relationship. As a subclass of BMC_ComputerSystem, any instance (representing a new business CI) of the BMC_VirtualSystemEnabler class is identified, at minimum, by the Name and SystemName attributes.

EnablerType attribute for BMC_VirtualSystemEnabler

In addition to setting the Name attribute, you can also define the type of software or OS, by populating the EnablerType attribute with the appropriate value. This attribute is an actual hypervisor type. The possible values are 0 (Other, the default), 1 (Unknown), 2 (PR/SM), 3 (z/VM), 4 (VMWare Server), 5 (Solaris Resource Manager), 6 (LPar), 7 (VPar), 20 (HP nPartitions, 25 (Integrity VM), 30 (Microsoft Hyper-V), 35 (VMWare ESX Server), 40 (VMWare Workstation), 45 (Xen Hypervisor), and 50 (LDOM Hypervisor).

For a complete list of attributes for the BMC_VirtualSystemEnabler class, see the BMC Atrium 7.5.00 Data Model Help.

Logical identity of BMC_VirtualSystemSettingData

The BMC_VirtualSystemSettingData class (derived from BMC_Settings) defines the virtual aspects of a virtual system through a set of virtualization-specific properties. BMC_VirtualSystemSettingData is also used as the top level class of virtual system configurations that model configuration information about virtual systems and their components. A virtual system configuration consists of one top-level instance of the BMC_VirtualSystemSettingData class that aggregates a number of instances of the BMC_ResourceAllocationSettingData class using the BMC_Component association.

For example, virtual system configurations may be used to reflect configurations of virtual systems that are defined at a virtualization platform virtual systems that are currently active input requests to create new virtual systems input requests to modify existing virtual systems snapshots of virtual systems.

This class was introduced in BMC Atrium 7.5.00. More information on it (and additional classes new to this version), including example illustrative model diagrams, will be provided in a future version of this guide.

Logical identity of BMC_ResourceAllocationSettingData

The BMC_ResourceAllocationSettingData class (derived from BMC_Settings) represents settings that specifically relate to an allocated resource that is outside the scope of the CIM class (which is typically used to represent the resource itself). These settings contain information specific to the allocation that may not be visible to the consumer of the resource itself. For example, a virtual processor may look like a 2 ghz processor to the consumer (as a virtual computer system); however, the virtualization system may use time-slicing to schedule the virtual processor so it can only use 1 ghz of CPU speed.

This class was introduced in BMC Atrium 7.5.00. More information on it (and additional classes new to this version), including example illustrative model diagrams, will be provided in a future version of this guide.

Logical identity of BMC_ResourcePool

The BMC_ResourcePool class (derived from BMC_LogicalEntity) serves as a logical entity (with associated controls) provided by the host system to allocate and assign resources. A resource pool may be used to allocate resources of a specific type. Hierarchies of resource pools may be created to provide administrative control over allocations. In cases where resources are subdivided, multiple resource pools may exist (for example, nodal boundaries in NUMA-like systems).

In systems that support over-commitment, pools represent the reservable capacity, not an upper bound or limit on the maximum amount that can be allocated. Admission control during power-on may detect and prevent systems from powering due to resource exhaustion. For example, over-commitment on a resource pool with ResourceType=Memory would require that sufficient space be available in a backing store that might be managed through a storage resource pool.

This class was introduced in BMC Atrium 7.5.00. More information on it (and additional classes new to this version), including example illustrative model diagrams, will be provided in a future version of this guide.

Deprecated classes for virtual systems

The following classes have been deprecated in BMC Atrium 7.5.00, and are no longer used for modeling virtualized environments:

- BMC_VirtualSystem (including all subclasses)
- BMC_VMWare
- BMC_VMWareVirtualSystem
- BMC_UnixVirtualSystem
- BMC_MFVirtualSystem
- BMC_MFVirtualSystemEnabler
- BMC_LPAR

Relationships used for virtual systems

The following table describes the relationships for virtual systems.

Relationship	Relationship class	Value of Name attribute
Resource allocation and resource pool	BMC_Dependency	ResourceAllocationFromPool
Virtual system and resource pool	BMC_Component	HostedResourcePool
Allocated resource and resource pool	BMC_Dependency	ElementAllocatedFromPool

Relationship	Relationship class	Value of Name attribute
Managed elements and setting data	BMC_SettingsOf	ElementSettingData
Allocated resource and resource pool	BMC_MemberOfCollection	IsMemberOfPool

Modeling operating systems

An operating system is software or firmware that controls the operation of a computer and directs the processing of programs. This section describes how to model Windows and UNIX operating systems.

To model a Windows or UNIX operating system, create an instance of the BMC_OperatingSystem class. Associate the instance to the parent BMC_ComputerSystem instance by a BMC_HostedSystemComponents relationship.

NOTE

This class is not reserved for servers and workstations only but is used to capture any type of operating system, such as the IOS for a Cisco network switch or router.

Logical identity of BMC_OperatingSystem

As with any related component of BMC_ComputerSystem, an operating system is identified by the Name attribute in conjunction with the SystemName attribute that represents the name of the parent instance of the computer. Therefore, the Name attribute represents the local name of the operating system CI in the context of the computer that is hosting it, as described in the following table.

Attribute	Description
Name	Use this attribute to specify the name of the child instance in the context of the parent instance of BMC_ComputerSystem. If multiple operating systems are installed on the same computer, Name must be structured so that the multiple instances have different names.
SystemName	Use this attribute to specify the name of the system. This must be the same as the parent instance of the BMC_ComputerSystem Name attribute. This attribute is automatically populated from the related CI when a weak relationship is created between the operating system and a computer system.

Additional attributes for BMC_OperatingSystem

The following table describes additional attributes of BMC_OperatingSystem.

Attribute	Description
Description	Use this attribute to specify the description for the operating system.
ManufacturerName	Use this attribute to specify the company that manufactured the operating system.

Attribute	Description
SerialNumber	Use this attribute to specify the serial number of the operating system.
ShortDescription	Use this attribute to specify a short description for the operating system.
VersionNumber	Use this attribute to specify the version number of the operating system.

Modeling hardware components

The hardware components that make up a computer system are captured by subclasses of BMC_HardwareSystemComponent. Generally, one subclass represents one type of hardware component. Examples of hardware components include:

- Disk drive—Machine that reads data from and writes data to a disk.
- Disk partition—Logical allocation of space on a disk drive.
- Monitor—Video device attached to computer systems that displays computer operations.
- Keyboard—Set of typewriter-like keys that enables you to enter data into a computer.
- Memory—Stores information about internal storage areas in a computer.
- Processor—Device that interprets a machine instructions in a computer.
- Network port—Interfaces that connect network drives to computer systems.

For example, you might identify a specific processor as an instance of BMC_HardwareSystemComponent. Each instance representing a hardware component is associated to the parent BMC_ComputerSystem instance by the BMC_HostedSystemComponents relationship.

Logical identity of BMC_HardwareSystemComponent

Like any child instance of BMC_ComputerSystem, a hardware component is identified by the Name attribute in conjunction with the SystemName attribute that represents the name of the parent computer instance. Therefore, the Name attribute represents the local name of the hardware CI in the context of the computer that is hosting it, as described in the following table.

Attribute	Description
Name	Use this attribute to specify the name of the child instance in the context of the parent instance of BMC_ComputerSystem.
SystemName	Use this attribute to specify the name of the system. This must be the same as the parent instance of BMC_ComputerSystem Name attribute. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the operating system.

Additional attributes for BMC_HardwareSystemComponent

The following table describes additional attributes of BMC_HardwareSystemComponent.

Attribute	Description
Description	Use this attribute to specify the description for the component.
ManufacturerName	Use this attribute to specify the company that manufactured the component.
SerialNumber	Use this attribute to specify the serial number of the component.
ShortDescription	Use this attribute to specify the short description for the component.
VersionNumber	Use this attribute to specify the version number of the component.

Because each hardware component contains attributes specific to its type, see the BMC Atrium 7.5.00 Data Model Help for a complete list of BMC_HardwareSystemComponent types and attributes to ensure you can accurately and completely represent your specific hardware CIs.

Modeling access points

A computer provides functions for other entities to use. Access points represent those available functions. Each access point represents the configuration of access to a function or the ability to invoke a service and is modeled by the BMC_AccessPoint class. This characteristic is further defined by BMC_ProtocolEndpoint, the only direct subclass of BMC_AccessPoint. Among other types of access points, a network address such as an IP address, MAC address, or IPX address, is captured as a subclass of BMC_AccessPoint.

Instances of the BMC_AccessPoint class are related to a computer system through a BMC_HostedAccessPoint dependency relationship. Access points exist within the context of a computer system, and are associated to their parent instance of the system through the BMC_HostedAccessPoint dependency relationship.

For example, Figure 1-4 on page 20 illustrates an example of a computer system's relationship to an IP endpoint, in the context of modeling a router.

Logical identity of BMC_IPEndpoint

An IP address is modeled as an instance of BMC_IPEndpoint. Like any child instance of BMC_ComputerSystem, an instance of BMC_IPEndpoint is identified by the Name attribute in conjunction with the SystemName attribute that represents the name of the parent instance of the computer. Therefore, the Name attribute represents the local name of the CI in the context of the computer that is hosting it, as described in the following table.

Attribute	Description
Name	Use this attribute to specify the name of the child instance in the context of the parent instance of the BMC_ComputerSystem. Name must be an IPv4 or IPv6 address, and must be formatted as decimal numbers delimited by a period, with no leading zeros.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value. This value must be set to IP.
SystemName	Use this attribute to specify the name of the system. This must be the same as the parent instance of the BMC_ComputerSystem Name attribute. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the operating system.

Additional attributes for BMC_IPEndpoint

The following table details additional attributes of BMC_IPEndpoint.

Attribute	Description
Address	Use this attribute to specify the IP address. This value must be compliant with AddressType.
AddressType	Use this attribute to specify the enumeration that defines the type of address. This value must be set to either 0 (Unknown), 1 (IPv4), or 2 (IPv6).
DNSName	Use this attribute to search a system based on its DNS name. The DNS name corresponds to the IP address; therefore, when you want to search a system by DNS name, you should look up the IPEndpoint CIs, and then the parent computer.
ProtocolType	Use this attribute to specify the enumeration that categorizes and classifies instances of this class.
ShortDescription	Use this attribute to specify the short description of the IP address.
SubnetMask	Use this attribute to specify the IP address subnet mask.

Logical identity of BMC_LANEndpoint

A MAC address is modeled as an instance of BMC_LANEndpoint. Like any child instance of BMC_ComputerSystem, a MAC address is identified by the Name attribute in conjunction with the SystemName attribute that represents the name of the parent instance of the computer. Therefore, the Name attribute represents the local name of the CI in the context of the computer that is hosting it, as described in the following table.

Attribute	Description
Name	Use this attribute to specify the name of the address. This must be an address suffixed by an index. The index uniquely identifies a MAC address for situations where multiple identical MAC addresses are configured within the same system. The index is generally the index of the MAC address entry in the SNMP MIB.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value. This value must be set to MACAddress:Index.
SystemName	Use this attribute to specify the name of the system. This must be the same as the parent instance of the BMC_ComputerSystem Name attribute. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the operating system.

Additional attributes of BMC_LANEndpoint

The following table describes additional attributes of BMC_LANEndpoint.

Attribute	Description
Address	Use this attribute to specify the MAC address.
ProtocolType	Use this attribute to specify the enumeration that categorizes and classifies instances of this class, such as for 14 (for Ethernet).
ShortDescription	Use this attribute to specify the short description of the MAC address.

Access point binding

Some access points use the services provided through another access point. You can use access point binding to establish a layering of two protocols, with the upper layer represented by the dependent and the lower layer represented by the antecedent.

This binding is modeled in the CDM by the BMC_Dependency relationship with the Name attribute set to BindsTo.

Modeling network interfaces and addresses

Network interfaces are captured by instances of BMC_NetworkPort. Although model extensions might define subclasses (like a FiberChannel port), the class that you should use for network interfaces is BMC_NetworkPort. Like other hardware components, each instance of a network port is associated to the parent instance of the BMC_ComputerSystem by the BMC_HostedSystemComponents relationship.

Network addresses are captured by BMC Discovery products as access points (inherited from BMC_AccessPoint) and therefore must always be associated to their parent instance of the computer through the BMC_HostedAccessPoint relationship. Also, a network address can have a relationship to the network interface for which it is configured. This relationship is modeled by a BMC_Dependency relationship in which the network interface is the antecedent (source) and the network address is the dependent (destination).

For more information on modeling network addresses, including an illustration of the relationships used in the model, see “Modeling network topology” on page 54.

Logical identity of BMC_NetworkPort

Like any child instance of BMC_ComputerSystem, a network port is identified by the Name attribute in conjunction with the SystemName attribute that represents the name of the parent instance of the computer. Therefore, the Name attribute represents the local name of the CI in the context of the computer that is hosting it, as described in the following table.

Attribute	Description
Index	Use this attribute to specify the index, which must be a valid SNMP index relative to the SNMP IF Table of the computer.
Name	Use this attribute to specify the Name of the network address. This must be an address suffixed by an index. The index uniquely identifies a MAC address for situations where multiple identical MAC addresses are configured within the same system. The index is generally the index of the MAC address entry in the SNMP MIB.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value. For instance, in many cases, network interfaces are best discovered and identified using SNMP information.
SystemName	Use this attribute to specify the name of the system. This must be the same as the parent instance of the BMC_ComputerSystem Name attribute. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the operating system.

Additional attributes for BMC_NetworkPort

The following table describes additional attributes of BMC_NetworkPort.

Attribute	Description
AutoSense	Use this attribute to specify the Boolean value that indicates whether the port can automatically determine the speed or other communications characteristics of the connected network media.
Description	Use this attribute to specify the description for the component.
FullDuplex	Use this attribute to specify the Boolean value that indicates whether the port is operating in full duplex mode (carrying signals in both directions).
LinkTechnology	Use this attribute to specify the enumeration of the types of link technologies, with values such as Unknown, Other, Ethernet, IB, FC, FDDI, ATM, Token Ring, Frame Relay, Infrared, BlueTooth, or Wireless LAN.
ManufacturerName	Use this attribute to specify the company that manufactured the component.
MaxSpeed	Use this attribute to specify the maximum bandwidth of the port in bits/second.
NetworkAddresses	Use this attribute to specify the list of strings specifying the network addresses for the port.
PermanentAddress	Use this attribute to specify the network address hard-coded into the port. This address can be changed by firmware upgrade or software reconfiguration. If it is changed, update this field. If no hard-coded address exists for the network port, leave this attribute blank.
PhysicalDescription	Use this attribute to specify the physical description or location for the port, such as slot3/port4.
PortType	Use this attribute to specify the enumeration of the types of ports, with values such as Ethernet, FDDI, Token Ring, WAN, or Unknown.
ShortDescription	Use this attribute to specify the short description for the component.
SerialNumber	Use this attribute to specify the serial number of the component.
SpeedConfigured	Use this attribute to specify the maximum bandwidth of the port in bits/second.

Modeling applications

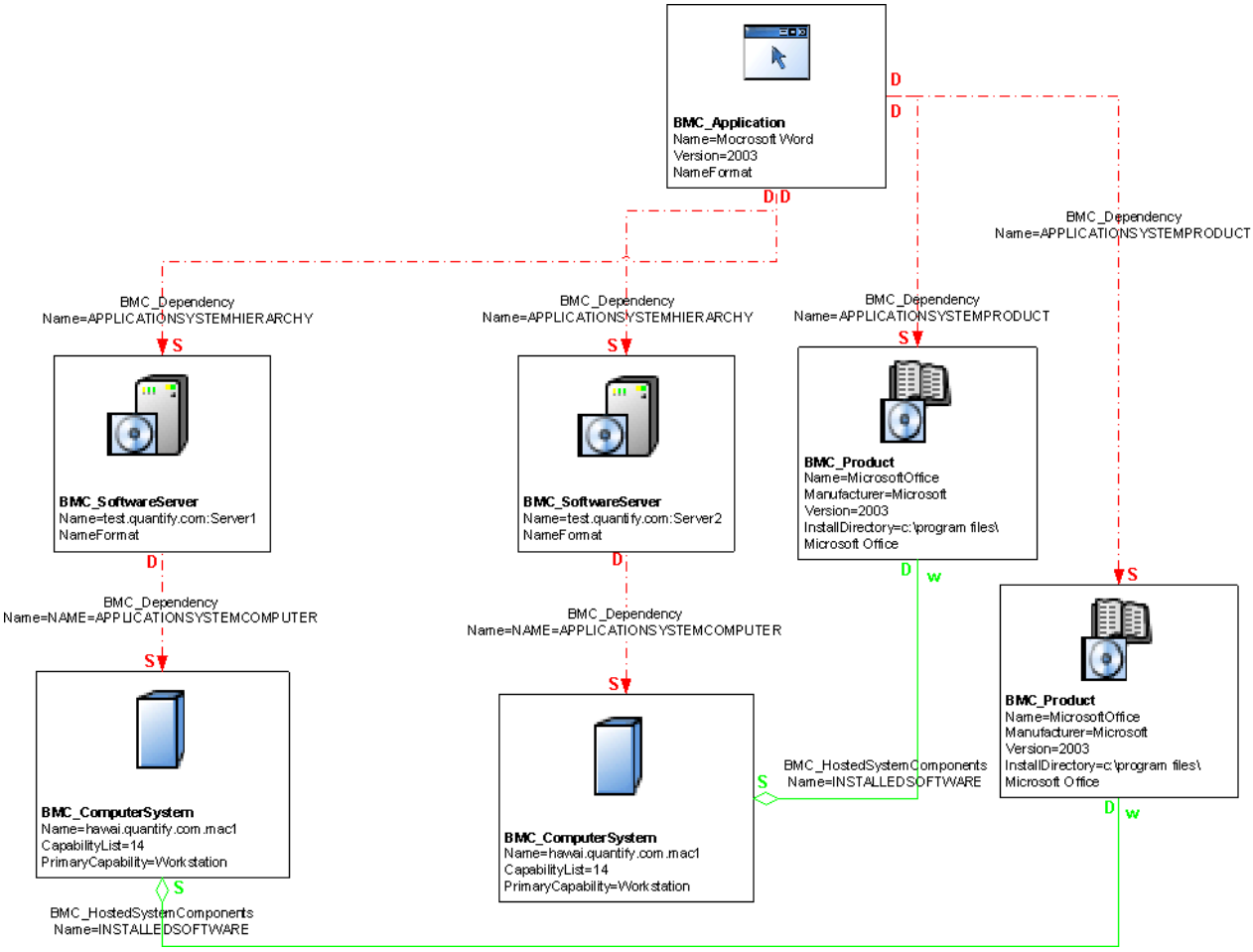
This section describes how to model software business entities, including applications, software servers, databases, and middleware. Applications have characteristics that help you determine how to best use the CDM in your modeling strategy. The following table maps the characteristics of an application to the type of class you would use to model that application.

Note that not all objects and relationships are required to model certain types of applications. For example, patch information may not be required in the case of Software License Management.

Characteristic	Description	Class
Runtime aspect	Running instances of applications and software servers	BMC_SoftwareServer BMC_Application BMC_ApplicationInfrastructure
Installation aspect	Identifies the product that is installed, its version, and any patch	BMC_Product
Service aspect	Business applications. (For business applications supporting a particular function such as payroll and trading, use the BMC_BusinessService class.)	BMC_Application BMC_BusinessService

Figure 1-5 on page 32 illustrates how the installed, runtime, and service aspects of an application relate to each other.

Figure 1-5: Illustrative representation of multiple application aspects



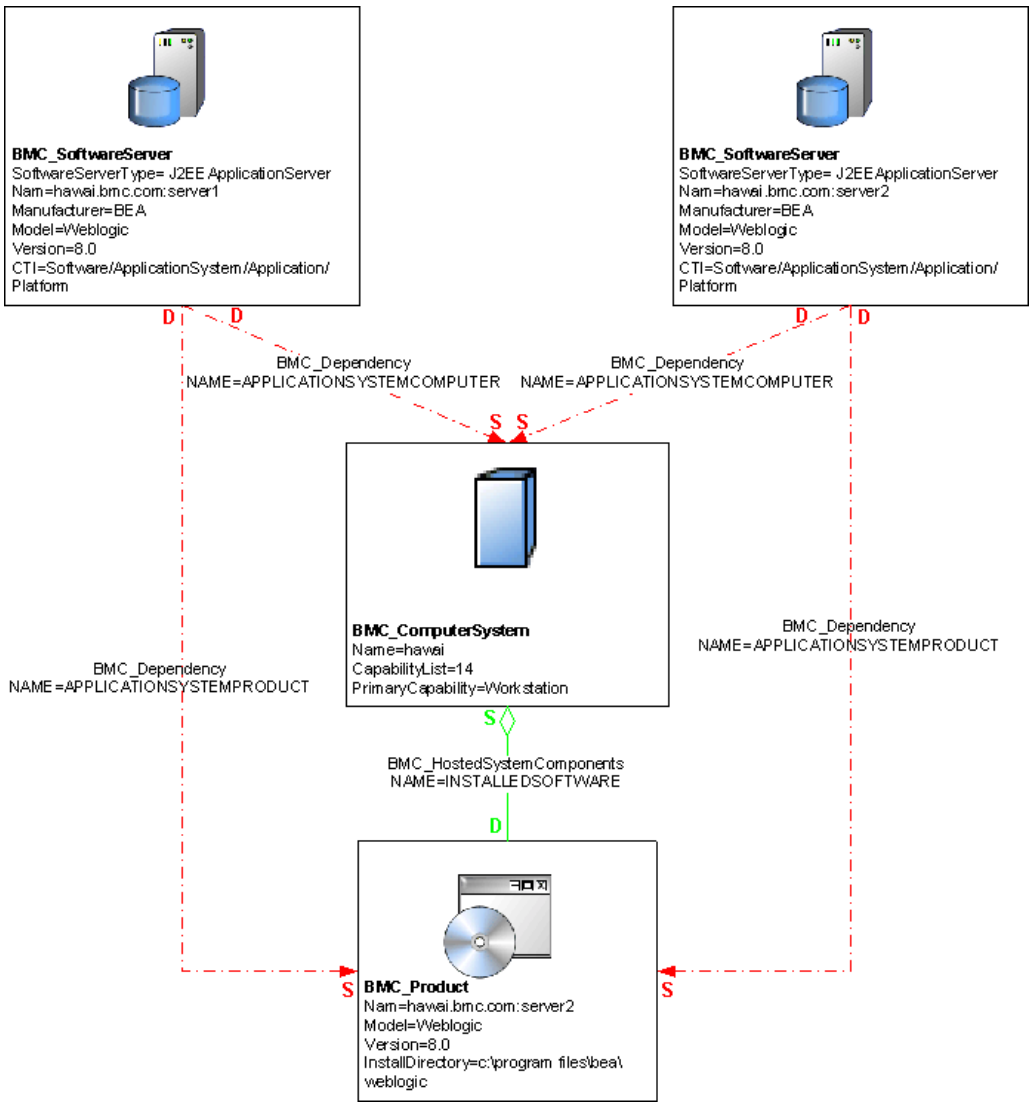
The BMC_SoftwareServer class represents the deployed, runtime aspects of applications; in other words, the instances of software actually running on a server. You instantiate this class to capture long-lived, server-type applications in your environment. When modeling applications, you must remember this distinction. To model static, installed components such as Microsoft Excel or Microsoft Word, create a BMC_Product instance.

You can also use the BMC_Product class to model noncommercial products, such as in-house software. One application can be installed once, yet have multiple instances running. For example, you can create a BMC_Product instance to represent the installed version of WebServer and create several BMC_SoftwareServer instances to represent actual instances of WebServer, one listening on port 80, another on port 8000, and a third on port 8080.

As another example, you would model Weblogic first by instantiating the BMC_Product class (to indicate where it is installed, the number of licenses, product name, and version). To add the runtime aspect, you would instantiate a BMC_SoftwareServer class.

Figure 1-6 illustrates an example of this model, where two instances of a Weblogic application server (server1 and server2) are actually instances of the same installed product.

Figure 1-6: Illustrative model of a Weblogic application



IMPORTANT

Accounting for the runtime aspect of the application in this context is very important for understanding the impact of an application on a business service. You must consider capturing Weblogic patches (using the `BMC_Patch` class), because the patch will then be connected to the service through the installed product, runtime, applications and, ultimately, the service and its relationships. Consequently, an IT administrator responsible for updating patches on Weblogic would understand how the change relates to the business that Weblogic supports.

For complete descriptions of the classes described in this section for modeling applications, including examples of usage, see the BMC Atrium 7.5.00 Data Model Help. For more information on using the BMC_Product class to model components, see “Modeling software inventory and patches” on page 18.

Application infrastructure and hosting environment

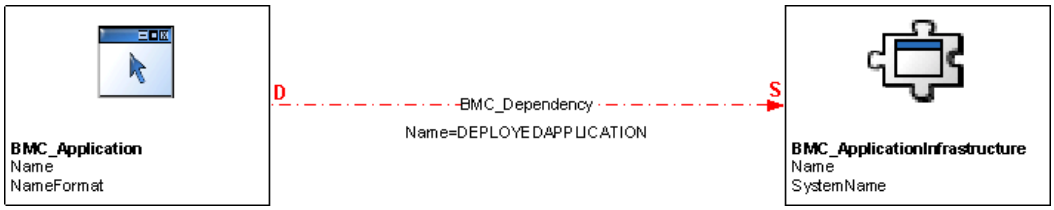
The BMC_Application class stores information about standalone applications, applications deployed on servers (such as SAP), and applications deployed on distributed systems (such as SAP).

The BMC_ApplicationInfrastructure class stores information about the framework that supports applications in a distributed or composite system. This class represents the platform to model your applications. For example, you would model SAP as an instance of BMC_ApplicationInfrastructure. After an application is deployed in that platform, it can run on any application server in the SAP environment. An application can be hosted by different types of environments: an application server or application system, or a physical or virtual system. Both of these environments are detailed in the following sections.

Applications running on application servers or application systems

To model applications to run directly on top of an application server or application system, relate an instance of the BMC_Application class to a hosting BMC_ApplicationInfrastructure instance. In this model, the application has only one relationship: a dependency on the application infrastructure hosting the application. This dependency is modeled by a BMC_Dependency relationship, as illustrated in Figure 1-7. When using the relationship, set the Name value to DEPLOYEDAPPLICATION.

Figure 1-7: Illustrative model of applications running on application systems



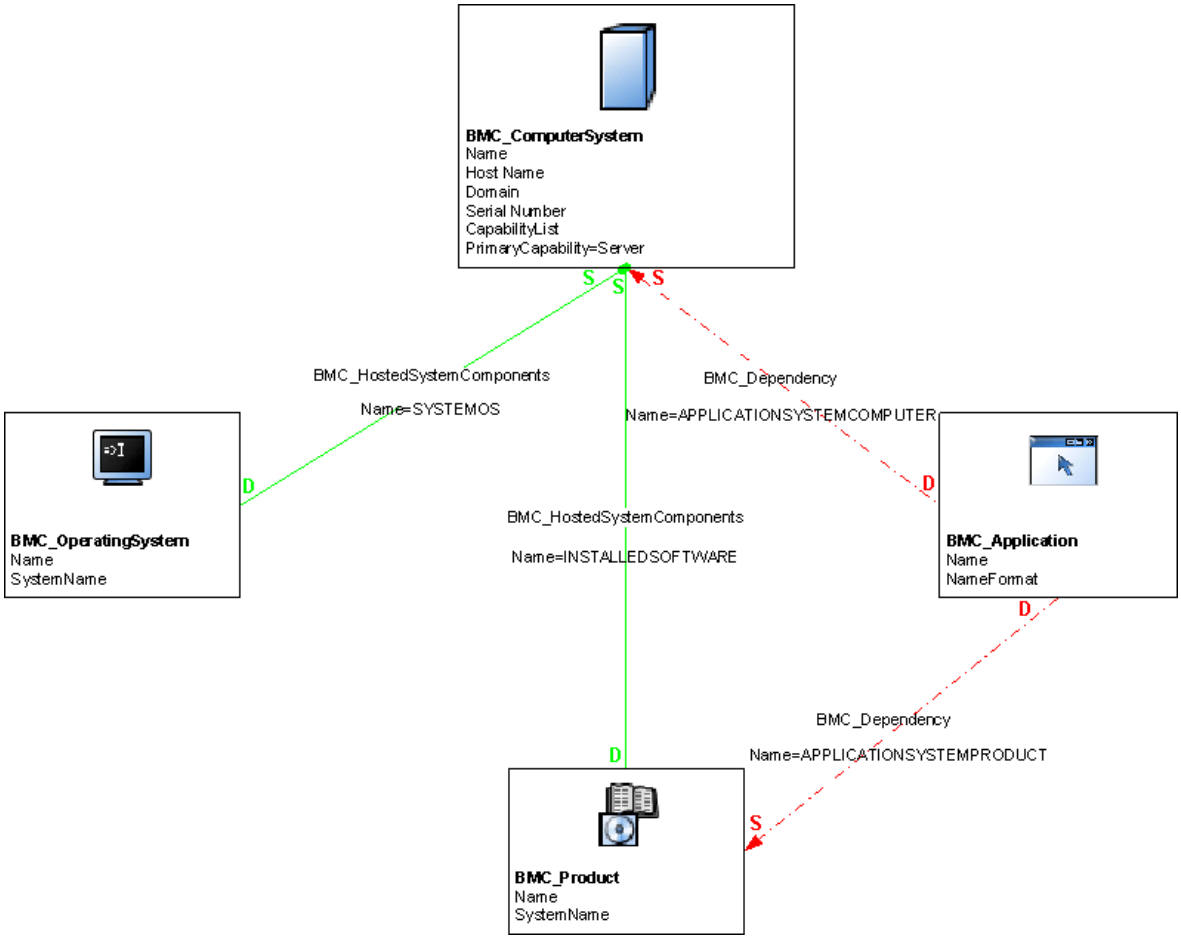
An application infrastructure cannot have any direct relationship to computers. Only applications and software servers have relationships to computers.

This model can also be applied to an application or set of applications that support or collaborate to provide a particular business function. For example, an Oracle® application infrastructure supports two applications, TimeCard and HR personal data, both stored in the BMC_Application class. The two classes relate to each other through the BMC_Dependency relationship, meaning that both the TimeCard and HR personal data applications are dependent on the supporting Oracle application infrastructure. To decompose the system into its functional components, relate an instance of this class to its component BMC_SoftwareServer instance with the BMC_Dependency class.

Applications running on computer systems

To model applications to run on computer systems (physical or virtual), relate an instance of the BMC_Application class to a hosting physical or virtual BMC_ComputerSystem instance. Figure 1-8 illustrates this model.

Figure 1-8: Illustrative model of applications running on computer systems



Relationships for applications

The relationships for modeling applications are described in the following table.

Relationship	Relationship class	Value of Name attribute
Application infrastructure hosting the application.	BMC_Dependency	DEPLOYEDAPPLICATION
System hosting the application (mandatory).	BMC_Dependency	APPLICATIONSYSTEMCOMPUTER

Relationship	Relationship class	Value of Name attribute
Operating system running the application (optional).	BMC_Dependency	APPLICATIONSYSTEMOS
Product representing the installed software of which this application is an instance (optional).	BMC_Dependency	APPLICATIONSYSTEMPRODUCT

Business applications and services

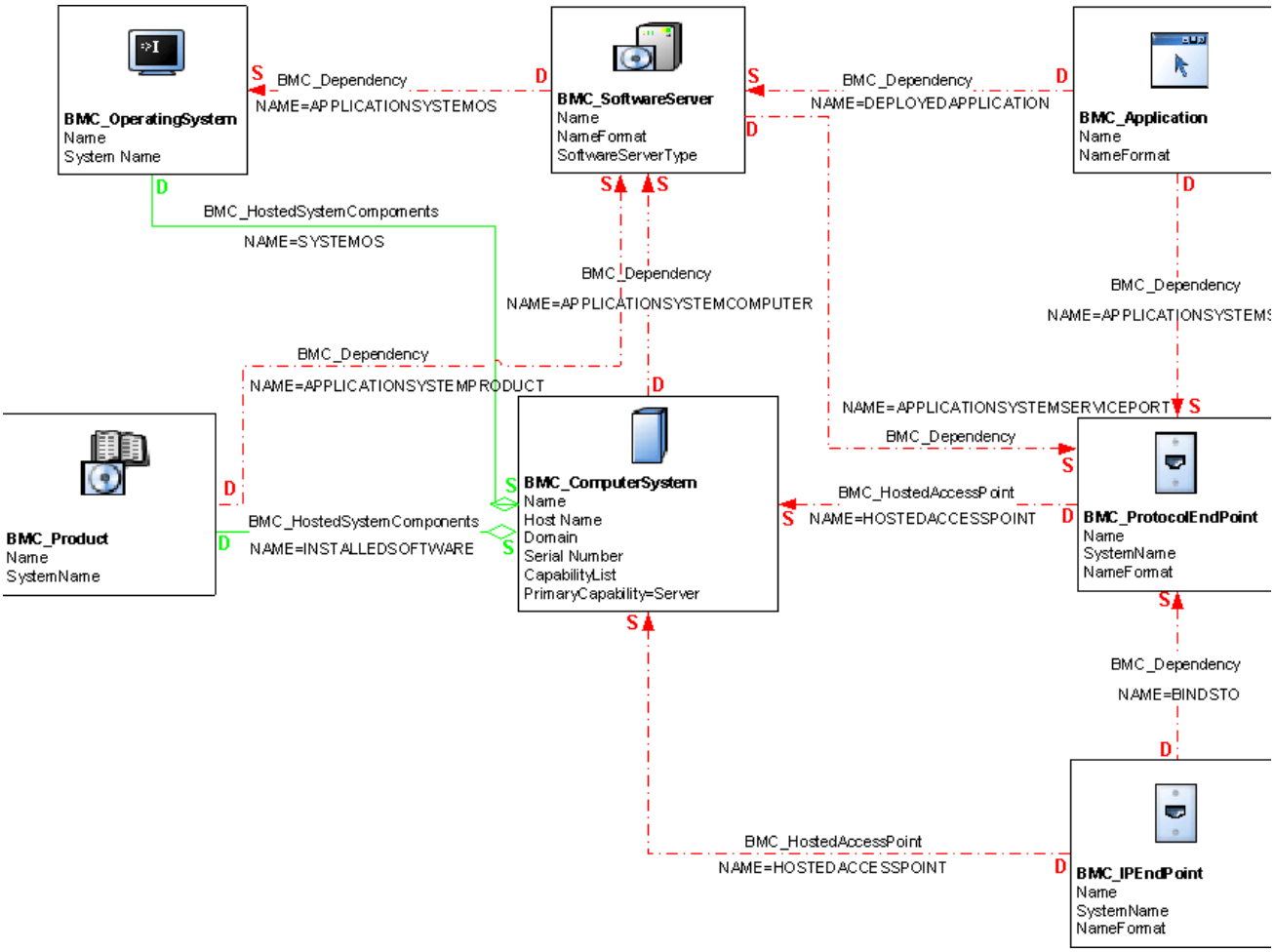
To model the business aspect of applications, use the BMC_BusinessService class. Business applications support a particular business function (such as payroll or trading) and are, generally, made up of a set of applications, servers, and databases that collaborate to provide a particular service.

Modeling software servers

A software server is a system that provides services to client applications and other servers, runs on top of a physical or virtual system, and is modeled using the BMC_SoftwareServer class.

Figure 1-9 on page 37 illustrates a software server model.

Figure 1-9: Illustrative model of software servers



Logical identity of BMC_SoftwareServer

The `BMC_SoftwareServer` class stores information about a server that provides a single service to client applications or other systems. Database servers, web servers, DNS servers, mainframe servers, and directory servers can be represented by this class.

The following table details the key attributes used in the `BMC_SoftwareServer` class. When modeling software servers, you identify the unique server type by specifying its name in the `SoftwareServerType` attribute.

For example, for database servers, set the SoftwareServerType attribute to DatabaseServer.

Attribute	Description
Name	Use this attribute to specify the name of the software server.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value. Set this attribute to name the installation directory of the server.
SoftwareServerType	Use this attribute to specify the type of software server (for example, DatabaseServer).

Additional attributes for BMC_SoftwareServer

The following table describes additional attributes for BMC_SoftwareServer.

Attribute	Description
ShortDescription	Use this attribute to specify a short description of the software server.
TokenId	Use this attribute to specify the unique identifier populated by BMC Discovery products and used by the Reconciliation Engine (of BMC Atrium CMDB) to identify instances.

Relationships for software servers

The following table describes the relationships for software servers.

Relationship	Relationship class	Value of Name attribute
Computer system hosting the software server (mandatory).	BMC_Dependency	APPLICATIONSYSTEMCOMPUTER
Communication endpoint (one relationship per endpoint) that the software server is listening on.	BMC_Dependency	APPLICATIONSYSTEMSERVICEENDPOINT
Operating system running the application. This dependency is actually on the OS that is running the server (as opposed to the computer).	BMC_Dependency	APPLICATIONSYSTEMOS
Installed software of which this software server is an instance (optional).	BMC_Dependency	APPLICATIONSYSTEMPRODUCT

Modeling database servers

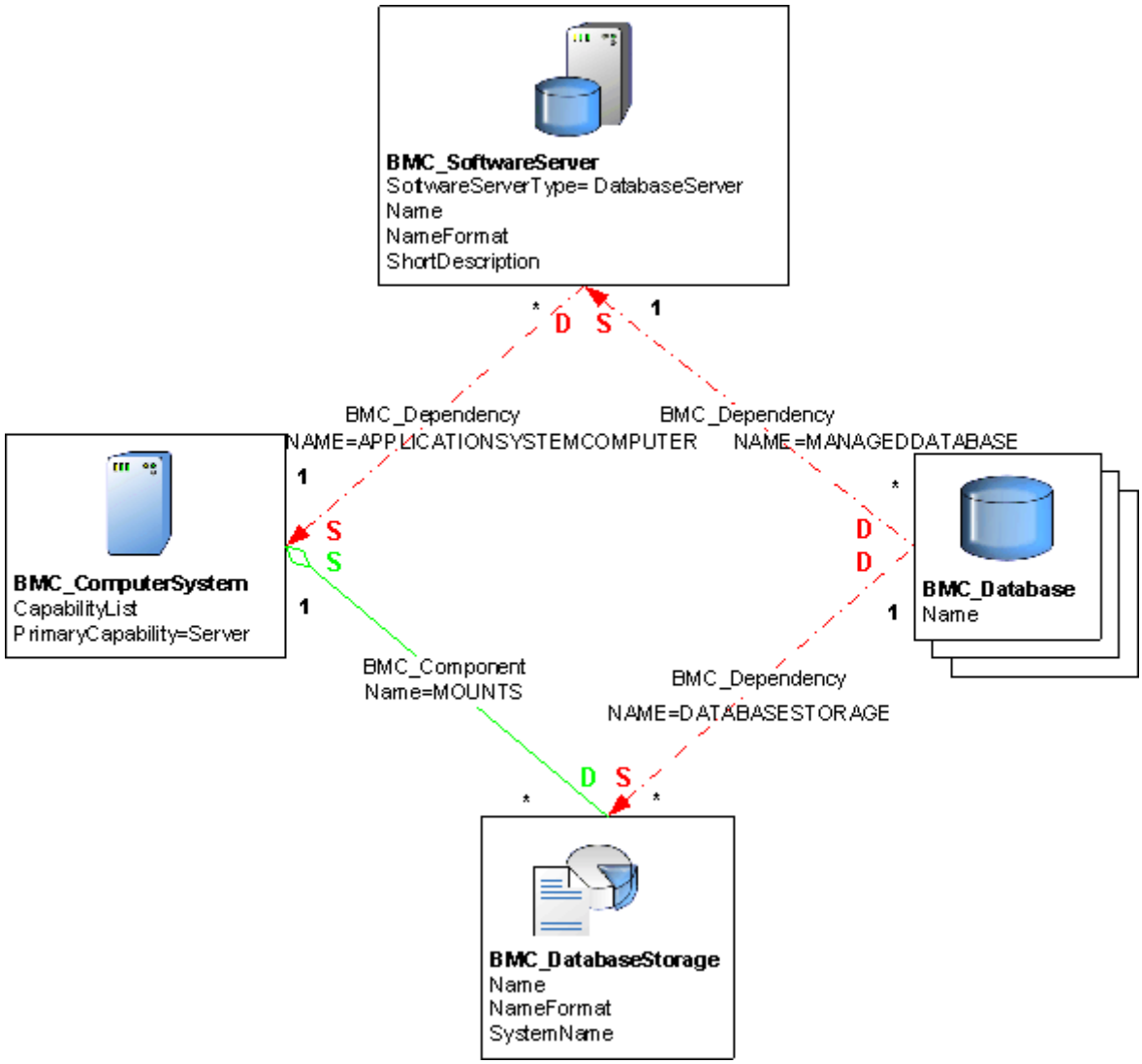
A database server is a form of software server that, like all software servers, must be uniquely named in the context of the CDM. A database server is modeled as an instance of the `BMC_SoftwareServer` class (derived from the `BMC_ApplicationSystem` class) and is identified by its `Name` attribute.

IMPORTANT

The key attribute for this class is `SoftwareServerType`, which must be set to DatabaseServer.

Figure 1-10 illustrates how to model a database server.

Figure 1-10: Illustrative model of a database server environment



Modeling databases

A database is a collection of interrelated data that is treated as a unit and that is organized into one or more schemas. Databases are dependent on software servers and, therefore, are dependent on database servers.

Logical identity of BMC_DataBase

A database is modeled as an instance of the BMC_DataBase class (derived from the BMC_LogicalEntity class) and is identified by its Name attribute.

The following table details the description and syntax for the Name and NameFormat attributes used in the BMC_DataBase class.

Attribute	Description
Name	Use this attribute to specify the name of the database
NameFormat	Use this attribute to specify the directory in which the server is installed. As multiple valid naming conventions can be used in specific contexts, the NameFormat attribute must be set with a value indicating the Heuristic used to generate the Name value. For example, in some cases, a computer system will be identified by an external DNS Name (a name configured in a DNS Server). In other cases, a static IP Address will be used. In any case, the values for NameFormat should be: ■ HostName.IP: The name must be a valid IP Address, decimal bytes delimited with dots ('.') ■ HostName.DNS: The name must a fully qualified host name, a host name and a domain name delimited with dots (the domain name can also consist of multiple components delimited with dots).

The BMC_DataBase class defines the properties that are common across database models and vendor implementations for the database entity that is represented by the unit of interrelated data. Create an instance of this class for each managed database. You can use this class to specify the software that belongs to the database, perform system-wide database management operations (such as stopping all the databases that were created by the system for maintenance purposes), or view runtime statistics for the database.

To represent database storage areas, use the BMC_DataBaseStorage class. The key to a BMC_DataBase instance in an enterprise environment is its Name attribute. For more information about database storage, see “Modeling database storage entities” on page 42.

Additional attributes for BMC_DataBase

Although databases are primarily defined by the Name attribute, the following attributes provide additional information about an instance of BMC_DataBase:

Attribute	Description
ShortDescription	Use this attribute to specify a short description of the database.
TokenId	Use this attribute to specify a unique identifier populated by BMC Discovery products and used by the Reconciliation Engine (of BMC Atrium CMDB) to identify instances.

Modeling an Oracle Listener

The Oracle Listener manages network communications for one or more database instances. An Oracle Listener is modeled as an instance of the BMC_SoftwareServer class (derived from BMC_ApplicationSystem) and is identified by both its Name attribute (set to Oracle Listener) and SoftwareServerType attribute (set to Other).

The following table details the attributes used to model Oracle Listeners.

Attribute	Description
Name	Use this attribute to define the name of the Oracle Listener.
NameFormat	Use this attribute to define the Heuristic used to generate the Name value.
SoftwareServerType	Use this attribute to define the type of server. This value must be set to Other.
ShortDescription	Use this attribute to define the description of the database.
TokenId	Use this attribute to define the unique identifier populated by BMC Discovery products and used by the Reconciliation Engine to identify instances.

Relationships for database servers and databases

The BMC_Dependency class is a generic association used to establish dependency relationships between instances in the BMC Atrium CMDB. This association allows you to establish dependency relationships between endpoints, including the roles of the endpoints.

The following table describes the relationships for database servers and databases.

Relationship	Relationship class	Value of Name attribute
Computer system (source) and database server (destination)	BMC_Dependency	APPLICATIONSYSTEMCOMPUTER
Computer system (source) and Oracle Listener (destination)	BMC_Dependency	APPLICATIONSYSTEMCOMPUTER
Oracle Listener (source) and database server (destination)	BMC_Dependency	DEPENDENCY
Installed software of which this software server is an instance (optional).	BMC_Dependency	APPLICATIONSYSTEMPRODUCT
Database server (source) and database (destination)	BMC_Dependency	MANAGEDDATABASE

Relationship	Relationship class	Value of Name attribute
Computer system (source) and the file system (destination)	BMC_HostedSystemComponents	HOSTEDSYSTEMCOMPONENT
Database storage (source) and database (destination)	BMC_Dependency	DATABASEDATASTORAGE

Modeling database storage entities

Database storage entities are an extension of file system CIs in a database environment and are modeled using the BMC_DataBaseStorage class.

Logical identity of BMC_DataBaseStorage

The BMC_DataBaseStorage class stores information about a collection of logical storage areas that hold and retain data. You model a database storage CI as an instance of the BMC_DataBaseStorage class (derived from the BMC_FileSystem class) and identify the instance by its Name and SystemName attributes. The BMC_DataBaseStorage class extends a file system CI and uses its inherited associations to represent the internal structure of the database.

The following table details the attributes used to model database storage CIs.

Attribute	Description
Name	Use this attribute to specify the storage area (tablespace) name.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value. Set it with the storage name used to generate the Name value.
SystemName	Use this attribute to specify the name of the parent BMC_Database CI. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the operating system.

Additional attributes for BMC_DataBaseStorage

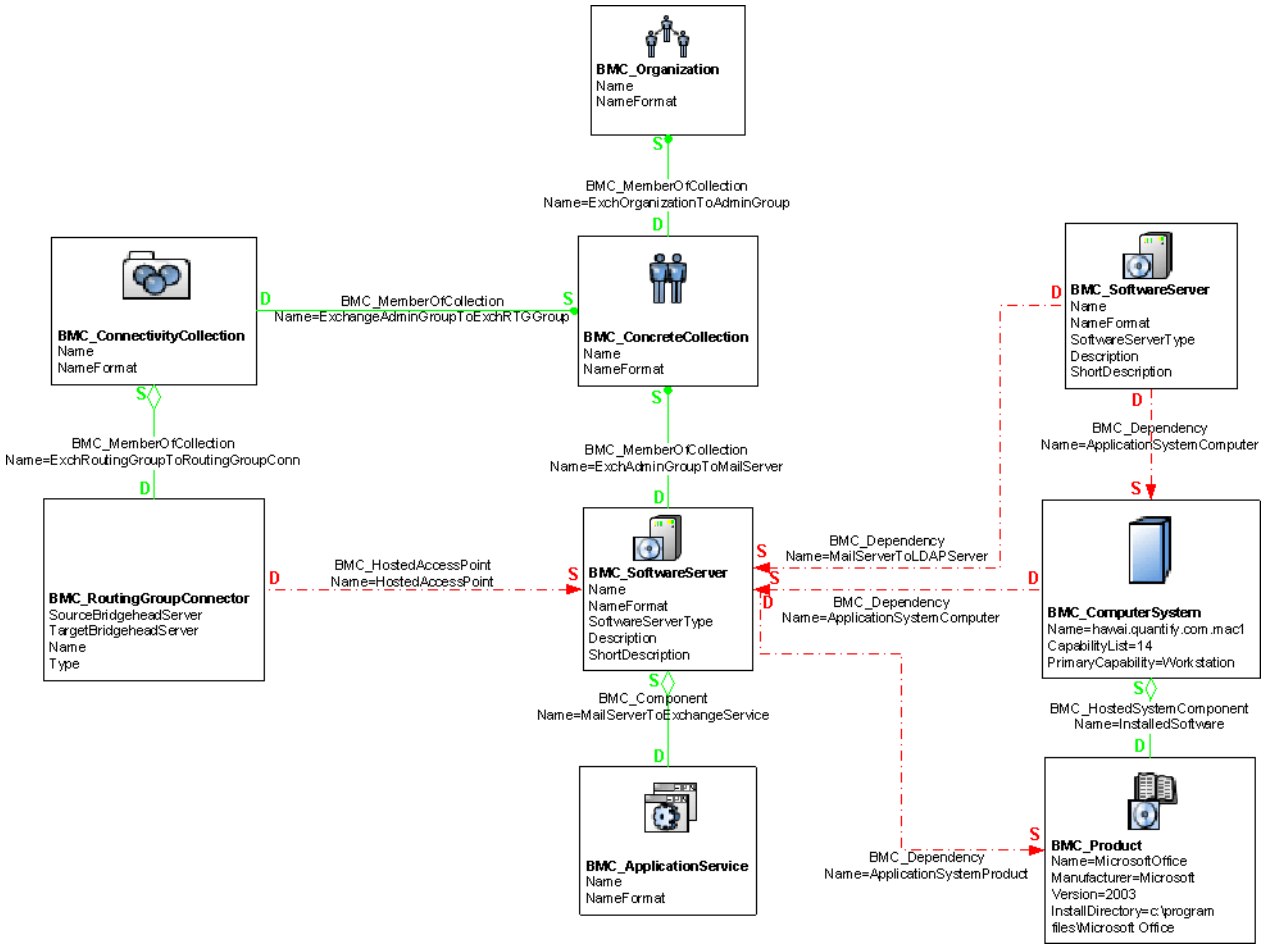
The following table describes attributes that provide additional information about an instance of BMC_DataBaseStorage.

Attribute	Description
IsSystemArea	Use this attribute to specify the owner of the storage area.
ShortDescription	Use this attribute to specify the description of the database.
TokenId	Use this attribute to specify the Token ID.

Modeling Microsoft Exchange business entities

This section describes how to use the CDM to model Microsoft Exchange. Exchange objects are contained in an organization, as illustrated in Figure 1-11.

Figure 1-11: Illustrative model of an Exchange environment



In the diagram, an Exchange server organization is represented by `BMC_Organization` (the container), which includes the objects defined in the following table:

Object	Function and class used to model
Exchange server organization administrative groups	Collect Exchange servers that are grouped to manage permissions. You model these groups using the `BMC_ConcreteCollection` class.
Exchange servers	Provide a single service to client applications or other systems. You model Exchange servers using the `BMC_Software` class.

Object	Function and class used to model
Exchange server services	Represent MS Exchange SA Service, MS Exchange MTA Service and MS Exchange IS Service. You model services using the BMC_BusinessService class.
Exchange server software	Represents the installed version of Exchange server. You model Exchange server software using the BMC_Product class.
Routing groups	Logically organize Exchange servers. You model routing groups using the BMC_ConnectivityCollection class.
Routing group connectors	Communicate between routing groups. You model connectors using the BMC_RoutingGroupConnector class.

Modeling an Exchange server organization

The BMC_Organization class represents the Exchange server organization and is the container that includes all administrative groups deployed within it.

Logical identity of BMC_Organization

An Exchange server organization is modeled as an instance of BMC_Organization and is identified by its Name attribute. The NameFormat attribute should be set to the name of the organization.

Attribute	Description
Name	Use this attribute to specify the name of the organization.
NameFormat	Use this attribute to specify the name of the organization in the format MSEXCHANGE:ObjectGUID. ObjectGUID is a unique identifier. As multiple valid Naming Conventions may exist and be used according to specific contexts, the NameFormat attribute must be set with a value indicating the Heuristic used to generate the Name value. For example, in some cases, you identify a computer system by an external DNS mame (a name configured in a DNS Server). In other cases, you will use a static IP address.

Additional attributes for BMC_Organization

The following table describes additional attributes that you can use to model an Exchange organization.

Attribute	Description
Description	Use this attribute to specify the description for the organization.
ShortDescription	Use this attribute to specify a short description for the organization.

Relationships for an Exchange organization

The following table describes relationships for an Exchange organization.

Relationship	Relationship class	Value of Name attribute
Exchange organization and administrative groups	BMC_MemberOfCollection	EXCHORGANIZATIONTOADMINGROUP
Administrative groups and Exchange server	BMC_MemberOfCollection	EXCHADMINGROUPTOMAILSERVER
Administrative groups and routing groups	BMC_MemberOfCollection	EXCHADMINGROUPTOEXCHRTGGROUP

Modeling Exchange server organization administrative groups

An Exchange server organization administrative group is a collection of Exchange servers that are grouped together for the purposes of managing permissions.

Logical identity of BMC_ConcreteCollection

An Exchange server organization administrative group is modeled as an instance of BMC_Collection. This class stores a generic and instantiable collection, such as a pool of hosts available for running jobs. It is defined as a subclass of BMC_Collection and is identified by its Name attribute. The NameFormat attribute is set to HostName.OrganizationName:Name of the administrative group.

Attribute	Description
Name	Use this attribute to specify the name of the organization administrative group.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value. Specify the name of the organization administrative group in the format MSEXCHANGE:ObjectGUID. ObjectGUID is a unique identifier.

Additional attributes for BMC_ConcreteCollection

The following table describes additional attributes that you can use to model an Exchange organization administrative group.

Attribute	Description
Description	Use this attribute to specify the description for the organization administrative group.
ShortDescription	Use this attribute to specify a short description for the organization administrative group.

Modeling Exchange servers

An Exchange server is an extension to the BMC_SoftwareServer class that stores information about a server and provides a single service to client applications or other systems.

Logical identity of BMC_SoftwareServer

An Exchange server is modeled as an instance of the class BMC_SoftwareServer (derived from BMC_ApplicationSystem) and is identified by its Name attribute. The NameFormat attribute is set to HostName.DNS:Manufacturer.

Additional attributes for BMC_SoftwareServer

The following table describes additional attributes that you can use for modeling Exchange servers.

Attribute	Description
Description	Use this attribute to specify the description for the software server.
ManufacturerName	Use this attribute to specify the manufacturer name for the Exchange Server. Always set this attribute to Microsoft Corporation.
Model	Use this attribute to specify the model name of the Exchange server. Always set this attribute to Microsoft Exchange Server.
ShortDescription	Use this attribute to specify a short description for the software server.
SoftwareServerType	Use this attribute to define the type of Exchange server. Set this attribute to MailServer for Microsoft Exchange servers or LDAPServer for an LDAP server.

Relationships for an Exchange server

The following table describes relationships for an Exchange server.

Relationship	Relationship class	Value of Name attribute
Exchange server and its services whose class is BMC_BusinessService	BMC_Component	MAILSERVERTOEXCHANGESERVICE
Exchange server and connector	BMC_HostedAccessPoint	HOSTEDACCESSPOINT
Exchange server and product	BMC_HostedSystemComponents	INSTALLEDSOFTWARE
Mail server and LDAP server	BMC_Dependency	MAILSERVERTOLDAPSERVER
Computer system to mail server	BMC_Dependency	APPLICATIONSYSTEMCOMPUTER
Computer system and LDAP server	BMC_Dependency	APPLICATIONSYSTEMCOMPUTER
Mail server and product	BMC_Dependency	APPLICATIONSYSTEMPRODUCT

Modeling Exchange server services

Services describe what is delivered by an organization, process or system and cannot be performed or executed. Exchange server services run on Exchange servers and include message transfer agents, information stores, and system attendants. Application services can be used to capture modules of an application, or services that are deployed on a server.

Logical identity of BMC_BusinessService

Exchange server services are captured as an instance of `BMC_ApplicationService` and are identified by the `ServiceType` attribute. The `BMC_BusinessService` class stores information about business or technical services. For MS Exchange, this class is used to represent MS Exchange SA Service, MS Exchange MTA Service and MS Exchange IS Service.

Attribute	Description
`ServiceType`	Use this attribute to specify the type of service that the instance of `BMC_BusinessService` represents. Supported values are Unknown (10), BusinessService (20), and TechnicalService (30). The default value is Unknown.
`NameFormat`	Use this attribute to specify the Heuristic used to generate the `Name` value. Specify the name of the service in the format MSEXCHANGE:ObjectGUID. ObjectGUID is a unique identifier.

Modeling Exchange server software

This section details how to model installed versions of Microsoft Exchange software.

Logical identify of BMC_Product

Exchange server software is modeled using the `BMC_Product` class, and is identified by the `Name` attribute. Set the `NameFormat` attribute to InstallDirectory:Model:Version.

Additional attributes for BMC_Product

The following table describes additional attributes that you can use for modeling Exchange server installations.

Attribute	Description
`Description`	Use this attribute to specify the description for the product.
`InstallLocation`	Use this attribute to specify the location where the Exchange software is installed.
`ManufacturerName`	Use this attribute to specify the manufacturer name for the Exchange Server installation. Always set this attribute to Microsoft Corporation.

Attribute	Description
Model	Use this attribute to specify the model name of the Exchange server installation. Always set this attribute to Microsoft Exchange Server.
ShortDescription	Use this attribute to specify a short description for the product.
VersionNumber	Use this attribute to specify the version number of the Exchange server.

Modeling Routing groups

Routing groups are used to logically organize Exchange servers. They represent a set of protocol endpoints of the same type that can communicate with each other. They can also group related systems, users, or other base elements.

Logical identity of BMC_ConnectivityCollection

A routing group is modeled as an instance of BMC_ConnectivityCollection (derived from BMC_Collection) and is identified by the Name attribute. The NameFormat attribute is set to MSEXCHANGE:ObjectUID.

Additional attributes for BMC_ConnectivityCollection

The following table describes additional attributes for BMC_ConnectivityCollection.

Attribute	Description
Description	Use this attribute to specify the description for the routing group.
ShortDescription	Use this attribute to specify a short description for the routing group.

Modeling routing group connectors

Routing group connectors are used for communication between routing groups. Each routing group has a BridgeHead Exchange server that communicates with other routing groups through the connectors.

Logical identity BMC_RoutingGroupConnector

You model a routing group connector as an instance of the class BMC_RoutingGroupConnector (derived from BMC_AccessPoint) and identify the class by its Name attribute.

The name of the connector is set with the NameFormat attribute as SystemName:Name.

Additional attributes for BMC_RoutingGroupConnector

The following table describes additional attributes of BMC_RoutingGroupConnector for modeling connectors.

Attribute	Description
SourceBridgeHeadServer	Use this attribute to specify the source BridgeHead server.
TargetBridgeHeadServer	Use this attribute to specify the target BridgeHead server.

Relationships for routing groups and connectors

Routing groups and connectors have one relationship: a BMC_MemberOfCollection relationship with the Name attribute set to EXCHROUTINGGROUPTOROUTINGGROUPCONN.

Modeling storage entities and devices

The following sections detail how to model storage entities and devices, which might include business CIs such as tape drives, disk drives, and storage subsystems.

Tape drives

A tape drive is modeled as an instance of BMC_TapeDrive (derived from BMC_Media).

Logical identity of BMC_TapeDrive

The following table describes key attributes of a tape drive.

Attribute	Description
Name	Use this attribute to specify the unique name of the instance; not necessarily human-readable.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value.
SystemName	Use this attribute to specify the name of the system in which the component resides. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the operating system.

Additional attributes for BMC_TapeDrive

The following table describes additional attributes of a tape drive.

Attribute	Description
ComponentAliases	Use this attribute to specify the array of strings indicating the unique identifier of the SIM.
Description	Use this attribute to specify the description of the instance.
ManufacturerName	Use this attribute to specify the name of the vendor.
MediaType (not inherited)	Use this attribute to specify the type of media. Its value is always Removable Media.
Model	Use this attribute to specify the tape drive model.
ShortDescription	Use this attribute to specify a short description of the instance.
TokenID	Use this attribute to specify a unique identifier populated by BMC Discovery products and used by the Reconciliation Engine (of BMC Atrium CMDB) to identify instances.

Tape drive instance

The following table illustrates an example instance.

Attribute	Value
ComponentAlias	003590.B1A.IBM.13.000000044832.0080
Description	003590.B1A.IBM.13.000000044832.0080
ManufacturerName	IBM
MediaType	Removable Media
Model	3590-1
Name	003590.B1A.IBM.13.000000044832.0080
NameFormat	Mainframe
ShortDescription	003590.B1A.IBM.13.000000044832.0080
SystemName	003590.B1A.IBM.13.000000044832
TokenID	003590.B1A.IBM.13.000000044832.0080

DASD

A DASD is modeled as an instance of BMC_DiskDrive (derived from BMC_Media).

Logical identity of BMC_DiskDrive

The following table describes key attributes of a DASD.

Attribute	Description
Name	Use this attribute to specify the unique name of the instance; not necessarily human-readable.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value.
SystemName	Use this attribute to specify the name of the system in which the component resides. This attribute is automatically populated from the related CI when a weak relationship is created between the computer system and the operating system.

Additional attributes for BMC_DiskDrive

The following table describes additional attributes of a DASD.

Attribute	Description
ComponentAliases	Use this attribute to specify the array of strings indicating the unique identifier of the SIM.
Description	Use this attribute to specify the description of the instance.
ManufacturerName	Use this attribute to specify the name of the mainframe vendor.
MediaType (inherited from BMC_Media)	Use this attribute to specify the type of media. Its value is always Fixed Hard Disk.
Model	Use this attribute to specify the DASD model.
SerialNumber	Use this attribute to specify the manufacturer-allocated number used to identify the instance.
ShortDescription	Use this attribute to specify a short description of the instance.
TokenID	Use this attribute to specify a unique identifier populated by the BMC Discovery products and used by the Reconciliation Engine (of BMC Atrium CMDB) to identify instances.

DASD instance

The following table illustrates an example DASD instance.

Attribute	Value
Name	002105.000.IBM.13.000000025559.0B46
NameFormat	Mainframe
SystemName	002105.000.IBM.13.000000025559
ComponentAlias	002105.000.IBM.13.000000025559.0B46
Description	002105.000.IBM.13.000000025559.0B46
ManufacturerName	IBM
MediaType	Fixed Hard Disk

Attribute	Value
Model	3390
SerialNumber	ADR071
ShortDescription	ADR071
TokenID	002105.000.IBM.13.000000025559.0B46

Storage subsystems

A storage subsystem is modeled as an instance of BMC_StorageSubsystem (derived from BMC_ComputerSystem).

Logical identity of BMC_StorageSubsystem

The following table describes key attributes of a storage subsystem.

Attribute	Description
Name	Use this attribute to specify the unique name of the instance; not necessarily human-readable.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value.

Additional attributes of BMC_StorageSubsystem

The following table describes additional attributes of a storage subsystem.

Attribute	Description
ComponentAliases	Use this attribute to specify the array of strings indicating the unique identifier of the SIM.
Description	Use this attribute to specify the description of the instance.
ManufacturerName	Use this attribute to specify the name of the vendor.
Model	Use this attribute to specify the mainframe model.
ShortDescription	Use this attribute to specify a short description of the instance.
TokenID	Use this attribute to specify a unique identifier populated by BMC Discovery products and used by the Reconciliation Engine (of BMC Atrium CMDB) to identify instances.

Example subsystem instances

The following tables illustrate instances of storage subsystems.

Tape drive subsystem instance

Attribute	Value
CapabilityList	StorageSubsystem
ComponentAlias	003590.B1A.IBM.13.000000044832
Description	003590.B1A.IBM.13.000000044832
ManufacturerName	IBM

Attribute	Value
Model	TapeSubsystem
Name	003590.B1A.IBM.13.000000044832
NameFormat	Mainframe
PrimaryCapability	StorageSubsystem
ShortDescription	003590.B1A.IBM.13.000000044832
TokenID	003590.B1A.IBM.13.000000044832

DASD subsystem instance

Attribute	Value
CapabilityList	StorageSubsystem
ComponentAlias	002105.000.IBM.13.000000025559
Description	002105.000.IBM.13.000000025559
ManufacturerName	IBM
Model	DASDSubsystem
Name	002105.000.IBM.13.000000025559
NameFormat	Mainframe
PrimaryCapability	StorageSubsystem
ShortDescription	002105.000.IBM.13.000000025559
TokenID	002105.000.IBM.13.000000025559

Relationships for a storage subsystem

The following table describes the relationships for a storage subsystem.

Relationship	CI class	Relationship class	Value of Name attribute
Storage subsystem and an operating system	BMC_OperatingSystem	BMC_Dependency	STORAGESUBSYSTEMOS
Storage subsystem and a DASD	BMC_DiskDrive	BMC_HostedSystemComponents	STORAGESUBSYSTEMDASD
Storage subsystem and a tape drive	BMC_TapeDrive	BMC_HostedSystemComponents	STORAGESUBSYSTEMTAPE

Modeling software and hardware clusters

Use the BMC_Cluster class to classify or update groups of software or hardware. Clusters are modeled using the BMC_Cluster class, which stores information about the cluster in relation to the BMC_System component.

Clusters help increase the performance of resources by storing groups of two or more computer systems or applications so that they operate together as a functional whole. Using clusters helps to improve and maintain the reliability, serviceability, and availability of your operating-system environment.

For example, an accounting department might create a cluster and link it with relationships to specific computer systems to obtain a complete picture of their business environment and assess the performance of computers individually and collectively. To do so, they could use the BMC_Component relationship to connect Computer A, Computer B, and Computer C to a performance measurement CI in the BMC_Cluster class.

Modeling network topology

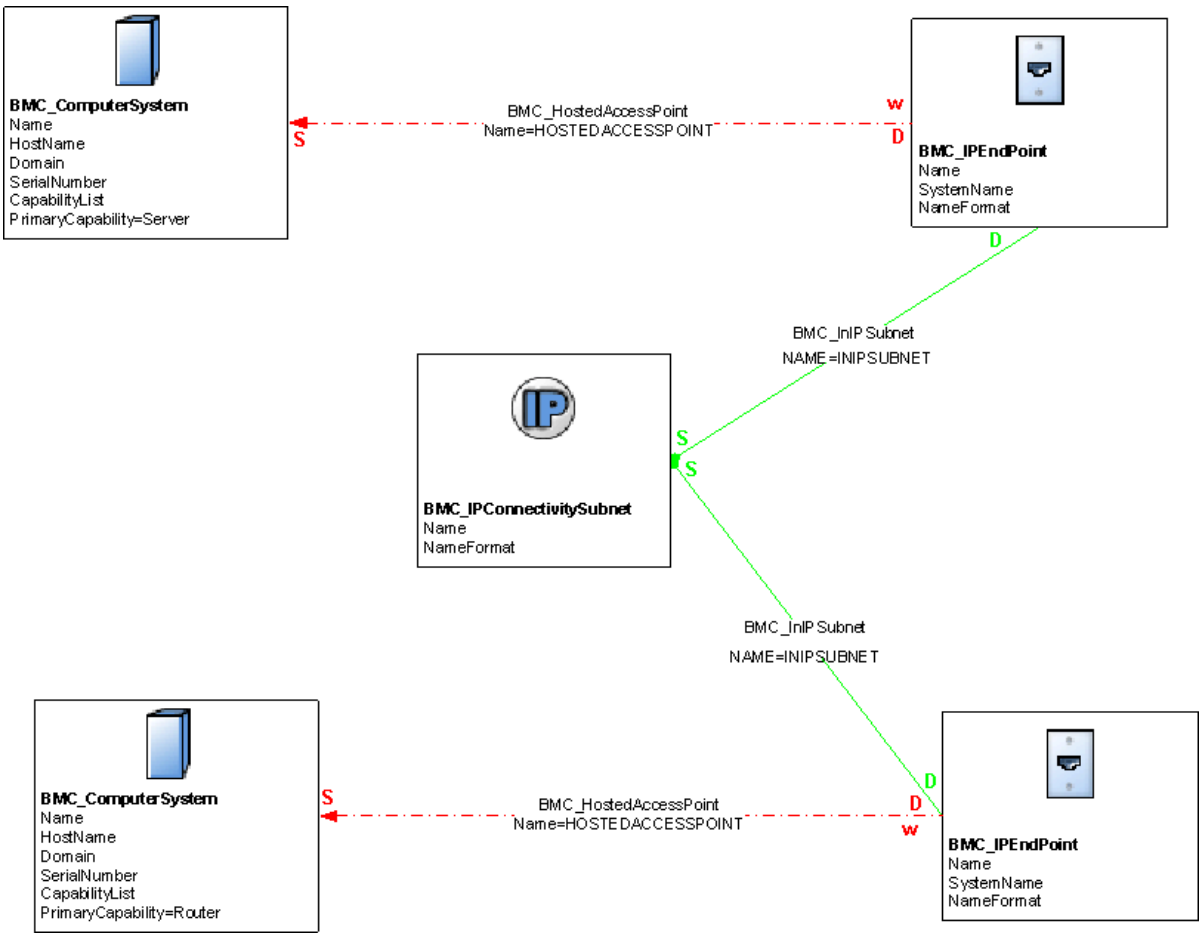
Topologies are based on the BMC_ConnectivityCollection class, which are collections of BMC_ProtocolEndpoint (communication points from which data may be sent or received) of the same type and which can communicate with each other. Logical groupings of these connectivity collections enable users to define the scope of LAN and WAN networks.

Modeling L3 topology and IP connectivity

A BMC_IPConnectivitySubnet instance represents a group of related BMC_IPEndpoint instances that can communicate with each other as members of a subnet and describes the characteristics of the subnet.

Figure 1-12 on page 55 illustrates a server and a router that belong to the same subnet.

Figure 1-12: Illustrative model of components on a subnet



Logical identity of BMC_IPConnectivitySubnet

The following table describes key attributes of `BMC_IPConnectivitySubnet`.

Attribute	Description
`Name`	Use this attribute to specify the IP address of the entire subnet, formatted according to the appropriate convention as defined in the `AddressType` attribute. When `AddressType` is 1 (IPV4), the `Name` must be built by concatenating the `SubnetNumber` and `SubnetMask` separated by a `/`.
`NameFormat`	Use this attribute to specify the Heuristic used to generate the `Name` value, which must be set to IP.

Additional attributes for BMC_IPConnectivitySubnet

The following table describes attributes that provide additional information about BMC_IPConnectivitySubnet.

Attribute	Description
AddressType	Use this attribute to specify an enumeration that describes the format of the Name and SubnetNumber properties in IPConnectivitySubnet: ■ 0—Unknown ■ 1—IPv4 ■ 2—IPv6
PrefixLength	Use this attribute to specify a prefix length for IPv6 addresses in the IP subnet (AddressType property is 2).
SubnetMask	Use this attribute to specify the mask for the starting address of the IPv6 IP subnet (AddressType is 1).
SubnetNumber	Use this attribute to specify the IP address of the entire subnet; must be equal to the Name attribute.

Relationships for components on a subnet

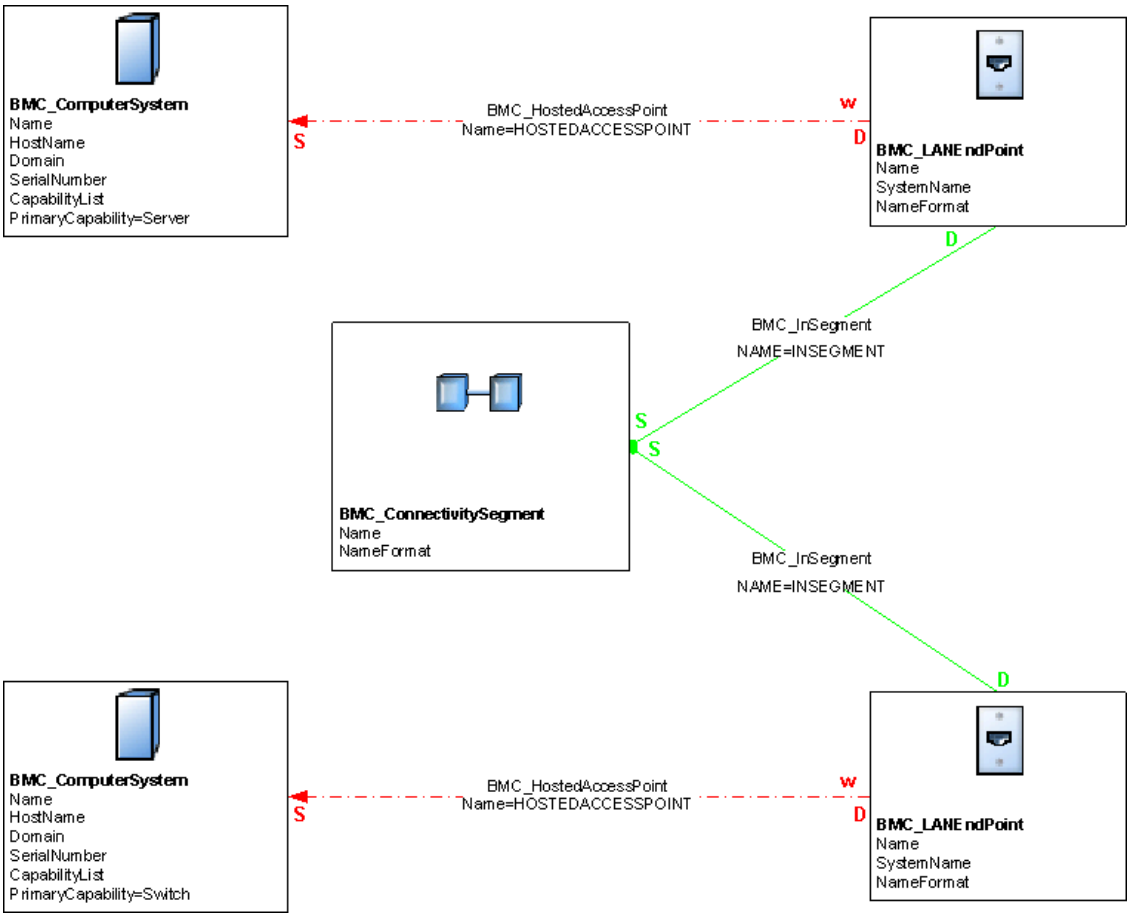
BMC_IPEndpoint instances are associated to the BMC_IPConnectivitySubnet to which they belong through the BMC_InIPSubnet relationship.

Modeling L2 topology and physical connectivity

A BMC_ConnectivitySegment instance represents a group of related instances of BMC_LANEndpoint of a particular type (such as Ethernet, token ring, or fiber channel) that can intercommunicate without the assistance of bridging or routing services. They are sometimes referred to as members of the same collision domain. The class describes the characteristics of the group, or segment.

Figure 1-13 illustrates a server and a switch, with the server having one NIC directly connected to a network interface of the switch.

Figure 1-13: Illustrative model of a network interface



Logical identity of BMC_ConnectivitySegment

The following table describes key attributes of `BMC_ConnectivitySegment`.

Attribute	Description
`Name`	Use this attribute to specify the name of the connectivity segment, which uses the following information and generates a hash code as resulting value: ▪ The list of physical addresses that belong to the segment. ▪ The name of the LAN instance to which the segment belongs.
`NameFormat`	Use this attribute to specify the Heuristic used to generate the `Name` value, which must be set to OID.

Additional attributes for BMC_ConnectivitySegment

The following table describes attributes that provide additional information about BMC_ConnectivitySegment.

Attribute	Description
ConnectivityType	Use this attribute to specify an enumeration that describes the type of technology used: ■ 0—Unknown ■ 1—Other ■ 2—Ethernet ■ 3—Token Ring ■ 4—FDDI ■ 5—Fiber Channel
Count	Use this attribute to specify the current number of endpoints connected to this segment. Count equals 2 indicates a direct connection between the two network ports interconnected by means of the segment.

Relationships for network interfaces

Instances of BMC_LANEndpoint are associated to the BMC_ConnectivitySegment to which they belong through the BMC_InSegment relationship.

Modeling network topology and LAN and WAN networks

LAN and WAN networks do not have any well-known identifier, such as an IP address or mask for an IP subnet. These networks are characterized by the list of machines can intercommunicate at the physical level without crossing the boundaries of gateways. This description includes the infrastructure network devices (switches, hubs) that enable these machines to communicate. In the CDM, LANs and WANs are captured by entities that aggregate that list of IP subnets.

Figure 1-14 illustrates a LAN that aggregates IP subnets.

Figure 1-14: Illustrative model of a LAN

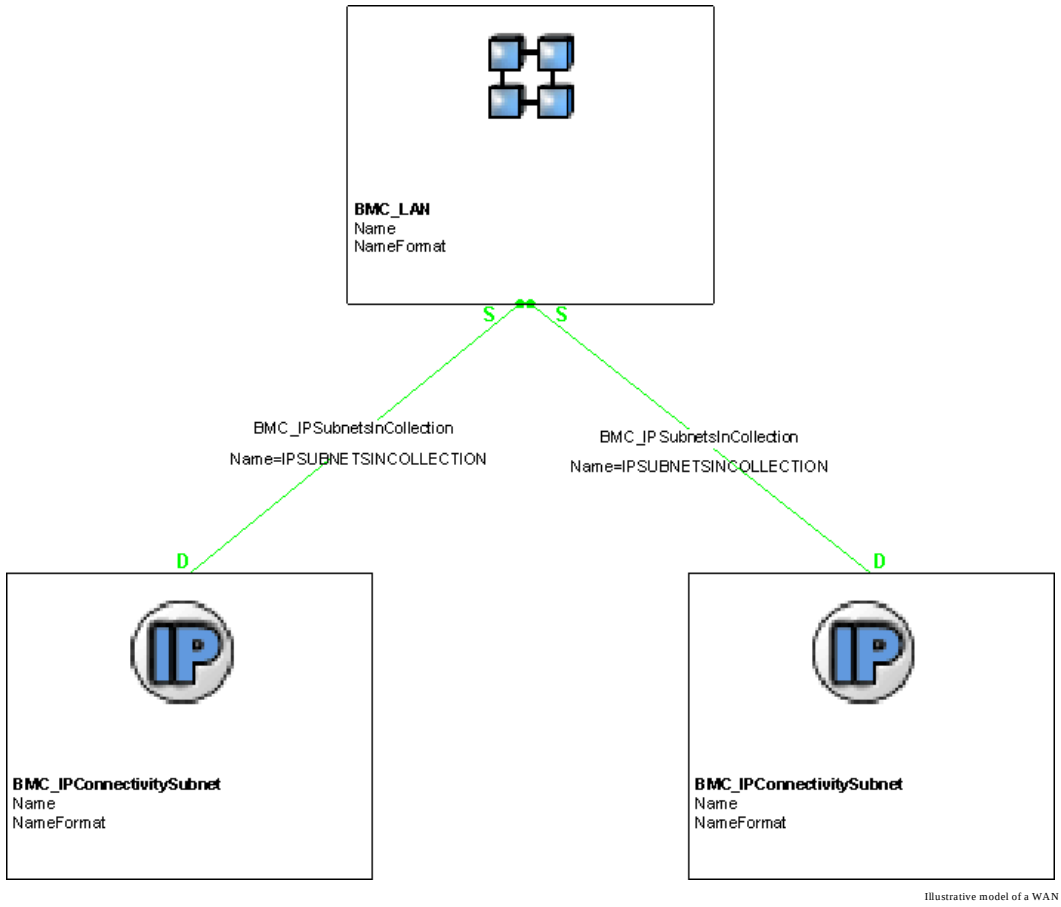
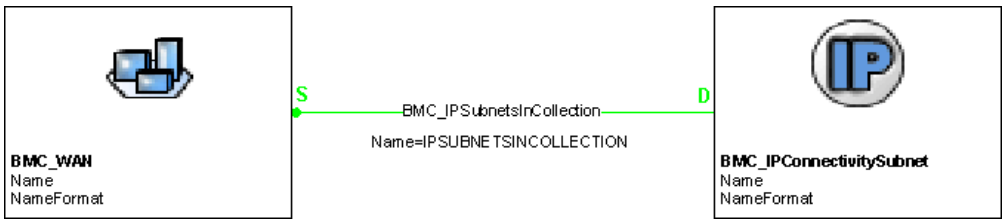


Figure 1-15 illustrates an example of a WAN that aggregates IP subnets.

Figure 1-15: Illustrative model of a WAN



Logical identity of BMC_LAN

The following table describes key attributes of BMC_LAN.

Attribute	Description
Name	Use this attribute to specify the name of the LAN, computed from the list of IP subnets that make up the LAN. The name for the LAN is the lexicographically lower value of the names of IP subnets.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value, which must be set to OID.

Logical identity of BMC_WAN

The following table describes key attributes of a BMC_WAN.

Attribute	Description
Name	Use this attribute to specify the name of the WAN, computed from the list of IP subnets that make up the WAN. The name for the WAN is the lexicographically lower value of the names of IP subnets.
NameFormat	Use this attribute to specify the Heuristic used to generate the Name value, which must be set to OID.
WANType	Use this attribute to specify the enumeration that describes the type of technology used: ■ 0—Unknown ■ 1—Other ■ 2—ATM ■ 3—Frame relay

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

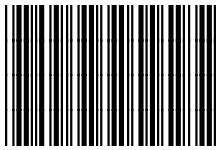
S

T

U

V

W



97740