Understanding the Application Management Model

CIM Version 2.7, Document Version 1.2

June 17, 2003

Abstract

The DMTF Common Information Model (CIM) is a conceptual information model for describing computing and business entities in enterprise and Internet environments. It provides a consistent definition and structure of data, using object-oriented techniques. The CIM Schema establishes a common conceptual framework that describes the managed environment.

The CIM Application Management Model is an information model that describes the details commonly required to manage software products and applications. This model can describe applications with various structures – ranging from standalone desktop applications to a sophisticated, multi-platform distributed, Internet-based application. Likewise, the model can be used to describe a single software product as well as a group of interdependent software products that form a business system.

Notices

DSP0140 *Status: Preliminary*

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents for uses consistent with this purpose, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

Table of Contents

1. Introduction

The CIM Application Management Model is an information model that describes the details commonly required to manage software products and applications. Figure 1 summarizes some of the ways this information model can be used.

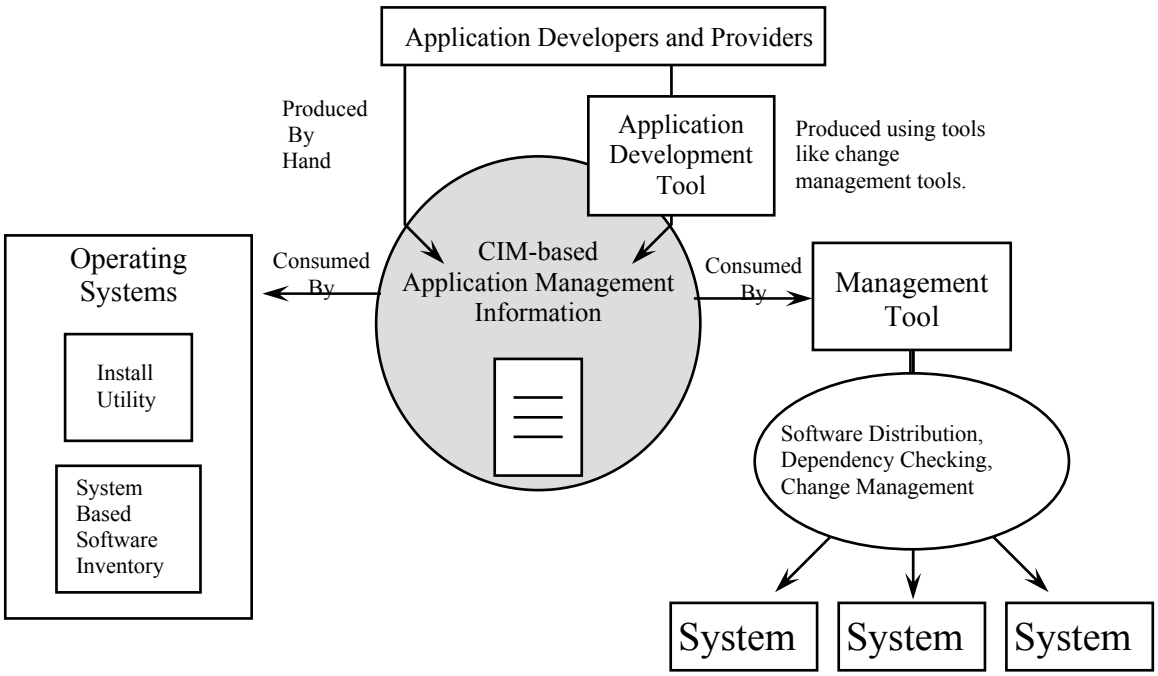


Figure 1. Developing a Management-Ready Application

In most cases, developers start with application/software providers, producing instances of the classes in the model to describe the manageable aspects of their product. The manageable aspects include details such as installable or deployable units, the files these units are composed of, the memory required to properly execute the software, etc. These providers can either become familiar with the details of the information model, so they can produce the details by hand, or application development tool vendors can enhance their tools to produce the information based on this open standard. As an example of the latter, a software change management tool typically knows the files that are produced from a build and can automatically record this information in the CIM application management model format.

By packaging an instance of the application management information model with an application or other software, operating systems utilities and/or management tools can consume the model in order to simply the work of their users. In the case of operating systems, utilities that install software or maintain databases of installed software can consume the information model, so users do not need to type the details in. On the other hand, management tools have instances of the information model to eliminate the need for users to type in details. For example, a software distribution tool can learn what needs to be done when the software is distributed to various operating system platforms. So, the information model can be used by a wide variety of tools.

Version 2 of the CIM Application Management Model is not expected to capture all the information these various tools require to accomplish their missions. The model provides a base upon which additional modeling concepts can be added. For example, management tools that monitor or measure applications can use the basic concepts of this model to describe the components of a software product that can be monitored, but details of how these components can be measured (e.g., output queue length) must be added. Similarly, management tools that deal with software maintenance issues can use the Version 2 model to capture the details of complete products, but must add additional modeling constructs to capture fixes or patches to be applied to this base.

In addition to the basic model application model lifecycle components, a special category of classes related to BIOS software which inherits from the application model are now included in the application model. This includes classes such as CIM_BiosFeature which is subclassed from CIM_SoftwareFeature. These components are really part of the Bios model and originally existed within the System devices group and SYSDEV model. They are now part of the application model mof because they use the application model classes as superclasses and therefore are only usable if the application model is installed. Please see the System model for more information on the use of these classes.

2. Basic Concepts

The application management model uses the concepts of a software product, software feature, software element, and application system. These concepts are defined as follows:

- A ***Software Product*** is a collection of software features that can be acquired as a unit. Acquisition implies an agreement between the consumer and supplier, which may have implications in terms of licensing, support, or warrantee.
- A ***Software Feature*** is a collection of software elements that performs a particular function or role of a software product. This level of granularity is intended to be meaningful to a consumer or user of the application to choose. This concept allows software products or application systems to be decomposed into units that have a meaning to users rather than units that reflect how the product or application was built (i.e., software elements).
- A ***Software Element*** is a collection of one or more files and associated details that are individually managed on a particular platform. It represents the level of granularity at which software features are managed.
- An ***Application System*** is a collection of software features that can be managed as an independent unit that supports a particular business function.

As the description indicates, the concepts of software product, application system, and software feature group or organize software elements. Note that a *Software Element* represents the level of granularity that is the fundamental building block of the application management information model.

These four concepts and the relationships between them are represented in the information using the classes and associations shown in Figure 2.

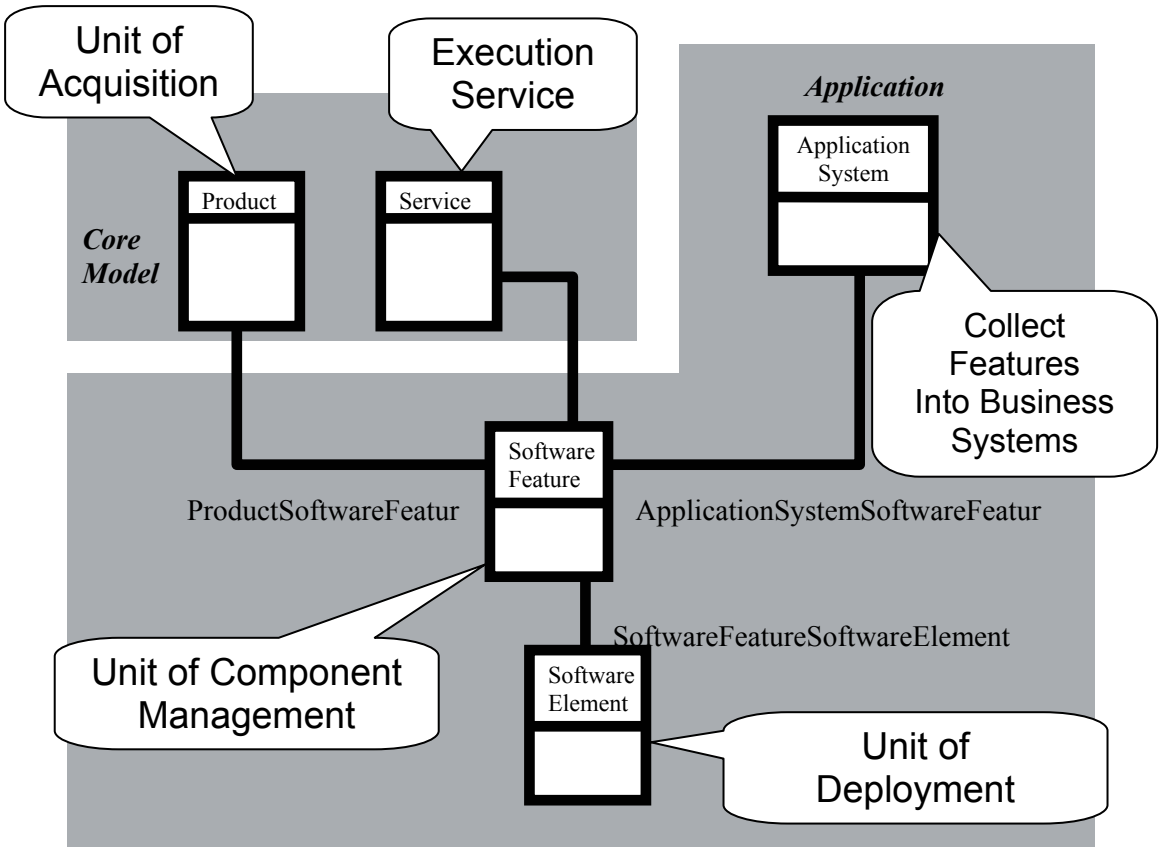


Figure 2. Base Classes for Application Management

A software product is defined using the Product class in the CIM Core Model. This class is a collection of software features that can be acquired as a unit. Acquisition implies an agreement between the consumer and supplier, which may have implications in terms of licensing, support, or warranties.

Also noted above, Application System is a subclass of the System Model. The software element and software feature concepts are subclasses of Logical Element, known as SoftwareElement and SoftwareFeature.

2.1 Software Elements Details

The model uses three additional concepts to organize more details about software elements. These include:

- software element life cycle or state
- environmental conditions that software elements depend on, and
- software element actions.

Figure 3 shows a conceptual picture of how these fit together. The next three sections provide a detailed description.

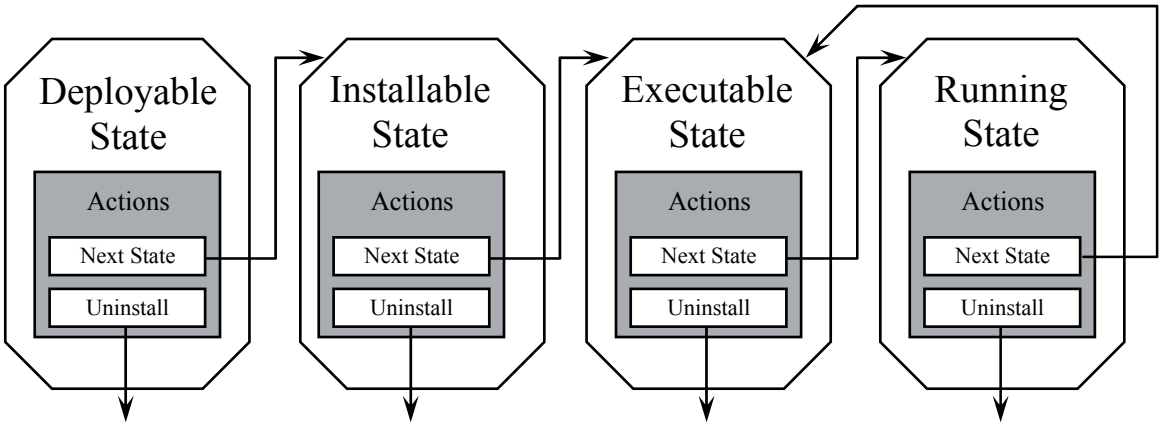


Figure 3. Software Element States, Conditions, and Actions

2.1.1 Software Element Life Cycle or State

The most basic aspect of managing software elements is managing its transitions through its life cycle. The life cycle can be segmented into four activities:

1. deploying a software feature
2. installing and configuring a software feature
3. starting a software feature
4. monitoring and operating a running software feature

As the software performs these tasks, the details of a particular software element changes. For example, the files that make up a software element when it is deployed are different from the files that make up a software element when it is running on a machine. Therefore, managing a software element through the act of performing these four tasks requires a management tool to understand what the particular details are before and after the tasks are performed. Using the value of the SoftwareElementState property in the SoftwareElement class, the following four states can be captured:

- A software element in the **deployable state** describes the details necessary to successfully distribute it, as well as the details (conditions and actions) required to create a software element in the installable state (i.e., the next state).
- A software element in the **installable state** describes the details necessary to successfully install it, as well as the details (conditions and actions) required to create a software element in the executable state (i.e., the next state).
- A software element in the **executable state** describes the details necessary to successfully start it, and the details (conditions and actions) required to create a software element in the running state (i.e., the next state).
- A software element in the **running state** describes the details necessary to monitor and operate on a start element.

2.1.2 Software Element Conditions

Conditions describe situations that may or may not exist in the environment where a software element also may or may not exist. The environment is typically a computer system. Conditions are organized into two groups: in-state conditions and next-state conditions.

The ***in-state conditions*** describe the characteristics of an environment that contains a software element in a particular state. Some examples of these characteristics are the directories that should exist, as well as the files that should be in these directories. For an existing software element, in-state conditions can be used to verify that the software element is complete or whole.

The ***next-state conditions*** describe the minimum set of characteristics that must be true in the target environment for the next-state actions (See Section 2.1.3 Software Element Actions) to successfully execute. The next-state conditions are the minimum pre-conditions required for the next-state actions to be successful.

Figure 4 shows the Application Model constructs that model both in-state and next-state conditions. The Check class is used to represent a condition, and the SoftwareElementChecks association links particular conditions with the appropriate Software Element. The value of the Phase property on the SoftwareElementChecks association determines whether a particular instance of a Check class is an in-state or next-state condition. The details of a particular condition (e.g., expected amount of disk space) are captured in a subclass of the Check class. The conditions covered in Version 2 of the Application Management Model and their interpretation is summarized in Table 1.

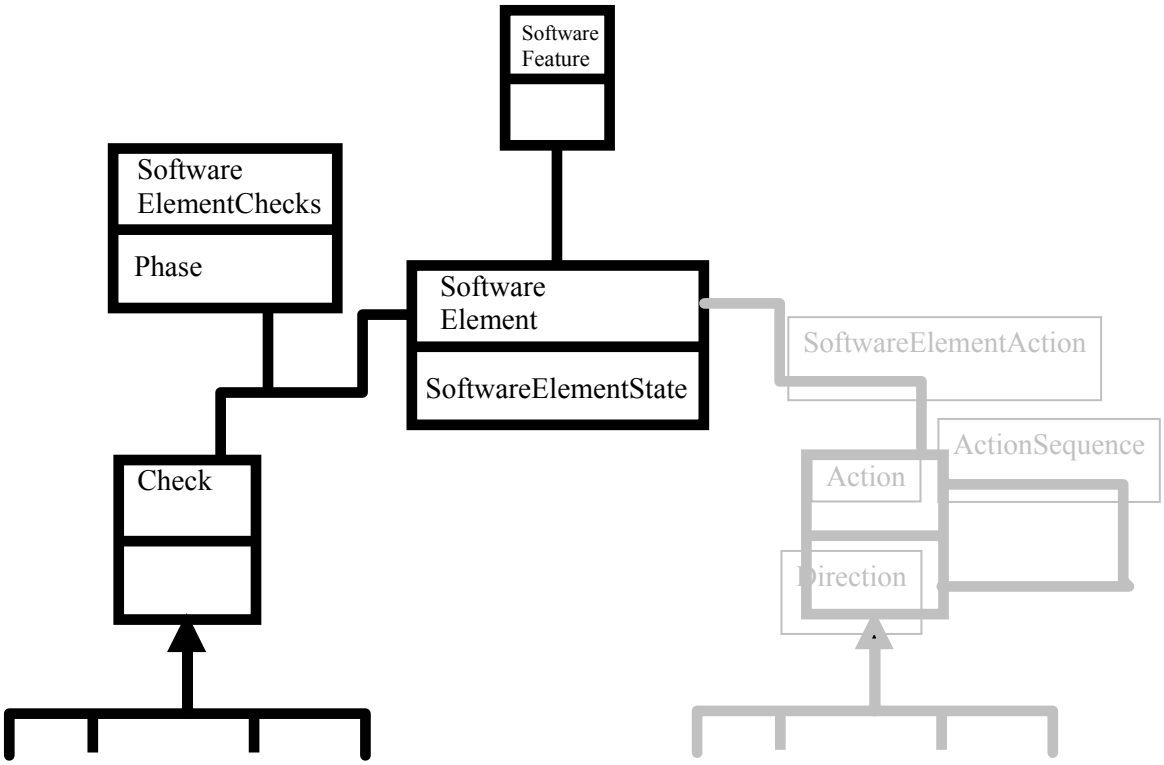


Figure 4. Using the Check Class for Conditions

Condition	In-State Interpretation	Next-State Interpretation
Memory Requirements	Minimum Amount of memory required to transition into the ***current*** state.	Minimum amount of memory required to transition into the ***next*** state
Disk Space	Minimum amount of disk space required to transition into the ***current*** state.	Minimum amount of disk space required to transition into the ***next*** state.
Swap Space	Minimum amount of swap space required to transition into the ***current*** state.	Minimum amount of swap space required to transition into the ***next*** state.
Architecture	The architecture required by a software element in the ***current*** state.	The architecture required by the software element to transition into the ***next*** state.
Files	A file that is expected to exist or not exist when a software element is in the ***current*** state.	A file that is expected to exist or not exist before a software element transitions into the ***next*** state.

Directories	A directory that is expected to exist or not exist when a software element is in the ***current*** state.	A directory that is expected to exist or not exist before a software element transitions into the ***next*** state.
OS Version	The version or ranges of versions a software element requires in its ***current*** state.	The version or ranges of versions a software element requires before it transitions into the ***next*** state.
Software Elements	A software element that is expected to exist or not exist when a software element is in the ***current*** state.	A software element that is expected to exist or not exist before a software element transitions into the ***next*** state.
Setting	An entry in a "setting" file that is expected to exist or not exist when a software element is in the ***current*** state.	An entry in a "setting" file that is expected to exist or not exist when a software element is in the ***next*** state.
Version Compatibility	Indicates whether it is permissible to create the software element in its next state.	

Table 1. Conditions (the "Check" Class) for Application Deployment

The model does not require all the aspects to be consistent. For example, there can be an in-state condition for one file to exist, and there can be a next-state condition for a different file to exist. In this case, the next-state condition says the files must be there before executing the next-state actions, where the in-state condition says this file should be there after the software element in the new state is created.

A user of the model cannot assume that the difference between the in-state conditions and the previous state's in-state condition will equal the sequence of actions. There may or may not have been an action to copy/move the file, depending on whether it is an explicit action or a side effect of executing another action (e.g., running setup.exe).

An important special case deals with the Runnable state. Since what follows the running state is the executable state, knowing uninstall actions for a software element in the running state has no meaning. The next-state conditions of a software element in a runnable state captures the details that lead to a successful shut down.

2.1.3 Software Element Actions

Actions are operations that either create a new software element or remove an existing software element. Actions are organized into two categories: next-state actions and uninstall actions.

The ***next-state actions*** describe a sequence of actions that will create a software element in the next state using a software element in the current state. These are the actions that perform the task described earlier (unpackage, install, start). The next-state actions do not destroy the software element in the previous state. The next-state conditions are pre-conditions that need to be true about the environment in order for this sequence of actions to succeed. These are not the only preconditions, since the in-state conditions can also be thought of as preconditions — the next-state actions assume that a software element exists in the current state.

The ***uninstall actions*** describe a sequence of actions that will properly remove a software element in a particular state from a machine. This is commonly used to uninstall software elements. The in-state conditions can be thought of as pre-conditions for the uninstall action. As shown in Figure 3, the uninstall actions remove or delete software elements. They do not create a software element in the previous state.

The class and associations used to capture action sequences in the Application Model are shown in Figure 5. The Action class represents a particular action. The value of the Direction property in the Action class determines whether the action is a next-state action or an uninstall action. SoftwareElementActions associations link the actions to a particular software element. The ActionSequence association is used to order the actions for a particular software element when there is more than one. The details of a particular type of action are defined in a set of subclasses. Table 2 summarizes the action defined in the version 2.0 model.

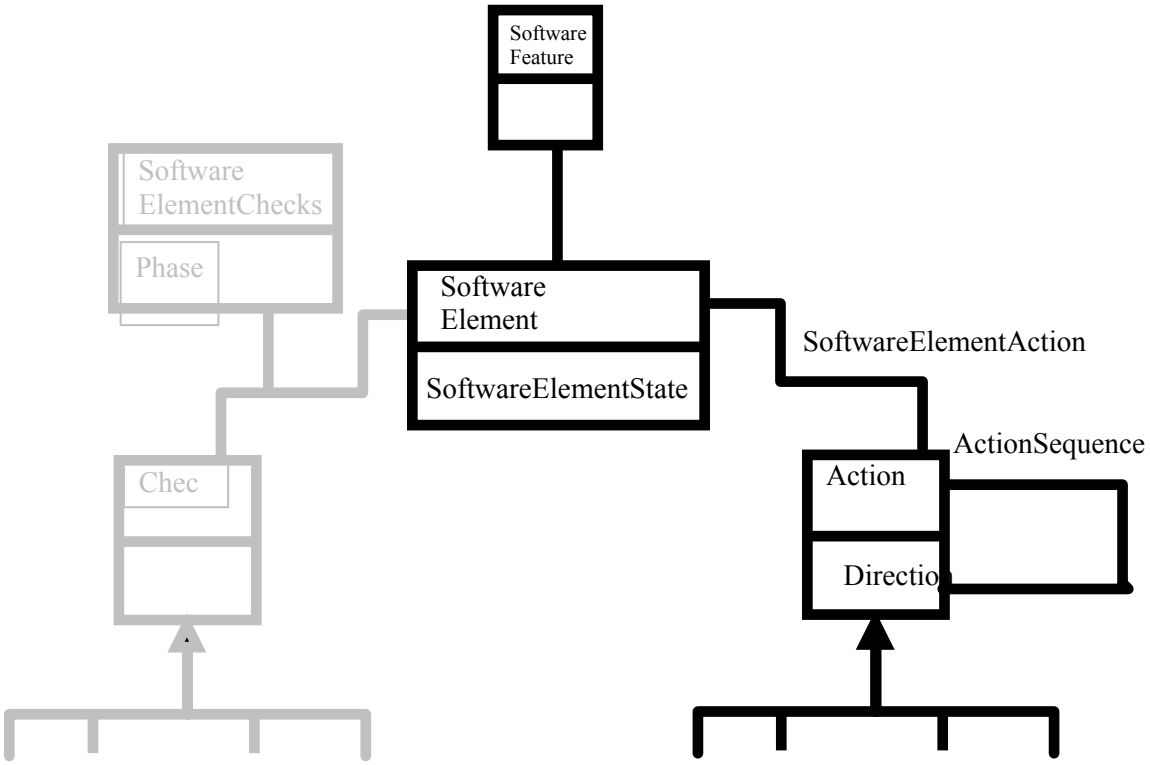


Figure 5. Classes and Associations for Actions

Actions	Description
Directory	An action to create or remove a directory.
File	An action to create or remove a file.
Reboot	An action that signals the need to reboot the computer system.
Execute Program	An action that executes a program. This can be the install script or program (e.g., setup.exe) when a software element in the installable state transitions to the executable state.
Modify Setting	An action to modify an entry in a "setting" file.

Table 2. Actions Defined in Application Deployment

State	Subpart	Description
Deployable	Next-State Actions	These actions perform the “unpackage” task to create a software element in the installable state.
	Uninstall Actions	These actions remove the software element in the deployable state.
Installable	Next-State Actions	These actions perform the install task to create a software element in the executable state.
	Uninstall Actions	These actions remove the software element in that state.
Executable	Next-State Actions	These actions start the software element.
	Uninstall Actions	These actions uninstall the executable state.
Runnable	Next-State Actions	These actions stop or terminate the running software element.
	Uninstall Actions	These are not typically used for a software element in a runnable state.

Table 3. "Change State" Actions Defined in Application Deployment

3. Application Management Model Use Case

This section illustrates how to use the CIM Application Management model for the fictional *CustomerFirst* software product. This is a customer support and problem tracking application developed by the First Always Company. Typical *CustomerFirst* users are technical support staff in a computer-related products company. They rely on *CustomerFirst* to log customer call incidents, research resolutions, and assign and track ownership of problems.

CustomerFirst is a client/server application with a thick client and a database server configuration. The clients run on various MS-Windows platforms and communicate with relational databases running on servers. Since companies typically have some users tracking customer calls or incidents, and have another set of users managing the resolution of problems, the First Always Company ships their products with two different client configurations: an Incident Manager client and a Problem Tracker client.

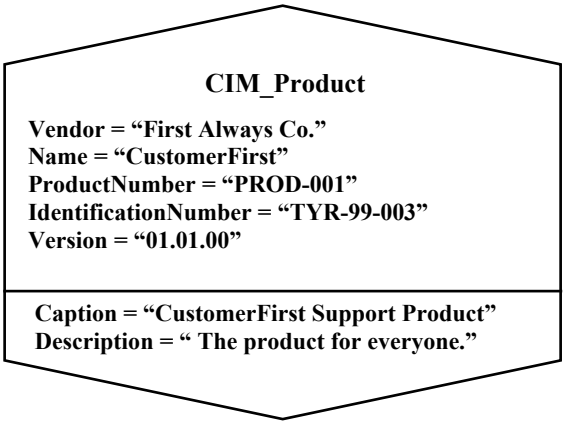
The CustomerFirst product uses other products like an Oracle Relational DBMS to deliver its functionality. In addition, Customer First can deliver additional capabilities when a customer has additional software products. These include:

1. The Problem Tracker client can use external e-mail services to notify its users of events. It supports a variety of mail systems and mail protocols.
2. Users with Microsoft Excel can create graphs and charts by accessing data from the *CustomerFirst* database. This feature is distributed and installed as an add-in to Excel.

In the following sections, we present a step-by-step discussion to create the appropriate Application Management model objects for CustomerFirst. When more details about the product are needed for the model (for example, memory and disk space requirements), this will be introduced.

3.1 Product or Application System

CustomerFirst is a software product purchased from the First Always Company. The CustomerFirst application uses other software products, like Oracle, to accomplish its tasks. Therefore, a Product class needs to be created for the CustomerFirst product. A graphical representation of the Product object and the matching MOF statement are:




```
instance of CIM_Product as $prod001
{
   Vendor = " First Always Co.";
   Name = "CustomerFirst";
   ProductNumber = "PROD-001";
   IdentifyingNumber = " TYR-99-003";
   Caption = " CustomerFirst Support
Product";
   Description = "The product for everyone ";
   Version = "01.01.00";
};
```

Table 4. CustomerFirst Product

Since the CustomerFirst product uses other software products, developers can combine its components with these products to form an application system. The section titled Application System Details illustrates this concept for the WeServiceIt company.

3.2 Software Feature or Software Elements

The Product object captures high-level details about the vendor of the product. The next step is to decompose the CustomerFirst product into its software features. From the overview of the CustomerFirst product we derive the following three features:

1. Incident Manager
2. Problem Tracker
3. Advance Graphics Package

For each of these, the developer must create a SoftwareFeature object, and these objects must be associated with the Product object using the ProductSoftwareFeature association. These associations are required because SoftwareFeature objects are named within the context of Product objects, meaning SoftwareFeature objects are weak with respect to Product objects. A graphical representation of the software features is shown in Figure 6.

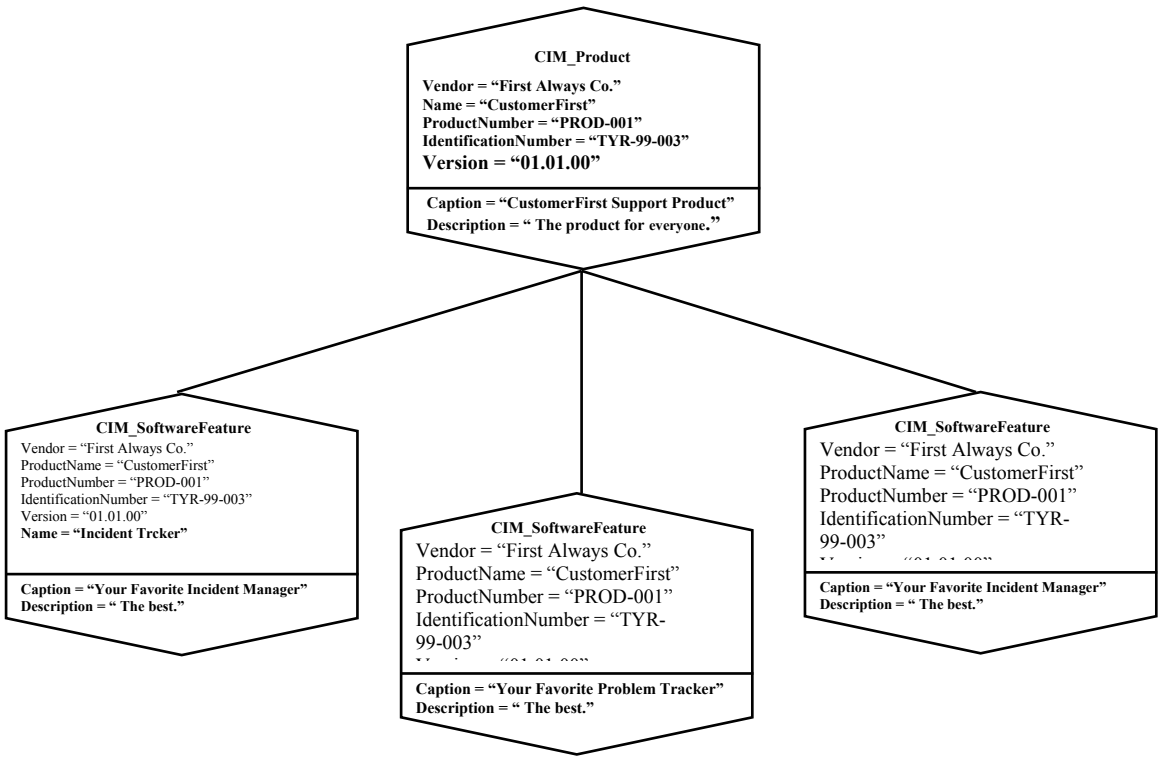


Figure 6. CustomerFirst Software Features

```
// ***********************************************
// Software Feature:
// ***********************************************
instance of CIM_SoftwareFeature as $feat001
{
   Vendor = "First Always Co." ;
   ProductName = "CustomerFirst" ;
   ProductNumber = "PROD-001" ;
   IdentificationNumber = "TYR-99-003" ;
   Version = "01.01.00" ;
   Name = "Incident Manager";
   Caption = " Your Favorite Incident Manager";
   Description = "The best." ;
};


instance of CIM_ProductSoftwareFeatures
{
   Product = $prod001 ;
   Component = $feat001 ;
};
```

The values for Vendor, ProductName, ProductNumber, IdentificationNumber, and Version are propagated keys from the CIM_Product object referenced by the CIM_ProductSoftwareFeatures asssocation.

Table 5. CustomerFirst SoftwareFeature MOF

3.3 Decomposing Software Features into Software Elements

Software Features are decomposed into manageable units known as software elements. The software elements for a particular software feature are organized by two dimensions: platform and state. A particular Software Feature can have one or more software element for each platform and state combination. Table 6 summarizes the eight software elements needed for the Incident Management features. This software feature has a single software element for each of the platform/state combinations.

	States			
Platform	Deployable	Installable	Executable	Running
NT	1	1	1	1
Windows 95	1	1	1	1

Table 6. Software Elements for Incident Management Feature

Figure 7 shows a UML instance diagram for the four software elements defined for the NT platform. There is one SoftwareElement object for each software element.

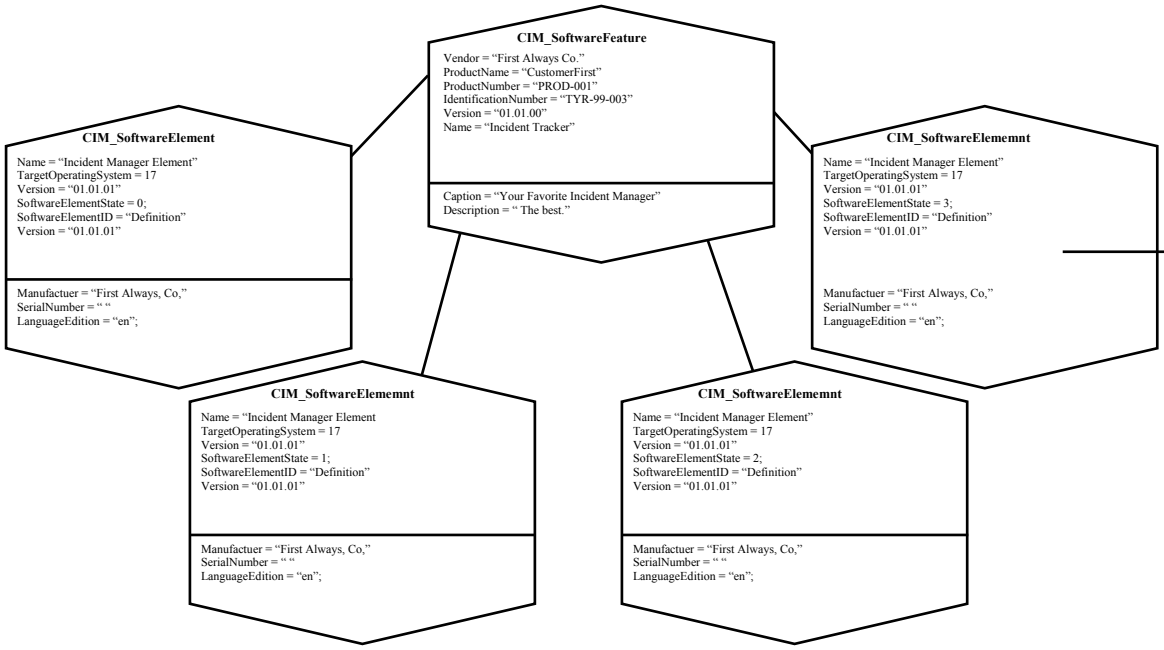


Figure 7. Software Elements for Incident Management Feature on NT

The MOF used to define and link one of these software elements (i.e., the installable state element) to the Incident Management feature is shown in Table 7. The MOF for the other

three software elements would be similar, except for the Name and the SoftwareElementState properties.

```
instance of CIM_SoftwareElement as $elmt001_01
{
   Name = "Incident Manager Element"; // key
   TargetOperatingSystem = 17; // NT
   Version = "01.01.01 "; // key
   SoftwareElementState = 0 ; // Deployable
   SoftwareElementID = " Definition"; // key
   Version = "01.01.01 "; // key
   Manufacturer = "First Always, Co. ";
   SerialNumber = " ";
   LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
   GroupComponent = $feat001 ;
   PartComponent = $elmt001_01 ;
};
```

Table 7. MOF Example of a Software Element Linked to a Software Feature

3.4 Capturing Software Element Conditions

For each software element, the Application Management model allows more details about items like disk space requirements, operating system version, etc. For example, Table 8 below shows the details of the Software Elements for the Incident Management software feature. The table captures some of the details that can be specified for each software element. These details include the version of the OS, the amount of disk space required to transition into the next state, and the amount of memory required for a software element within a particular state.

Feature	Platform	Details	Deployable	Installable	Executable	Running
Incident Manager		Version	3.5.1	3.5.1	3.5.1	3.5.1
	Win. 95	Disk Space	.75 MB	2 MB	15 MB	3 MB
		Memory			8 MB	32 MB
		Software			Oracle	Oracle

	NT	Disk Space	.75 MB	3 MB	25 MB	3 MB
		Memory			8 MB	48 MB
		Software			Oracle	Oracle
		Other			SMTP/POP3 Mail Server Notes Server	
Problem Tracker		Version				
	Win. 95	Disk Space	.75 MB	2 MB	15 MB	3 MB
		Memory			8 MB	48 MB
		Software			Oracle	Oracle
	NT	Disk Space	1 KB	5 KB	1 MB	
		Memory				
		Software			Excel	
Advance Graphics Package		Version				
	Win. 95	Disk Space	1 KB	5 KB	1 MB	
		Memory				
		Software			Excel Probl. Tracker	

Table 8. Incident Management Feature Condition Summary

These details are captured using Check objects. Recall that checks or conditions are organized into two lists: in-state conditions and next-state conditions. This distinction must be considered when mapping the details summarized in Table 8. According to the definitions in Table 1, the details can be interpreted as in-state conditions, so:

- the version row refers to the operating system version required by the software element in the specified state,
- the disk space row refers to the minimum amount of disk space required to transition in the specified state,
- the memory row refers to the minimum amount of memory required to transition into the specified state,
- the software row refers to the expected software element(s) that is expect to exist for the software element in the specified state.

It is not necessary, but it is reasonable to consider defining next-state conditions for conditions such as version, disk space, and memory on the previous state. For example, the 15 MB of disk space required to transition the Incident Manager on NT software element into its executable state can also be specified as a next-state condition for the same software element in the installable state. Figure 8 shows the Check objects required for the Incident Manager software element targeted for NT in the executable state using for the OS Version details. Similar objects need to be created for disk space and memory.

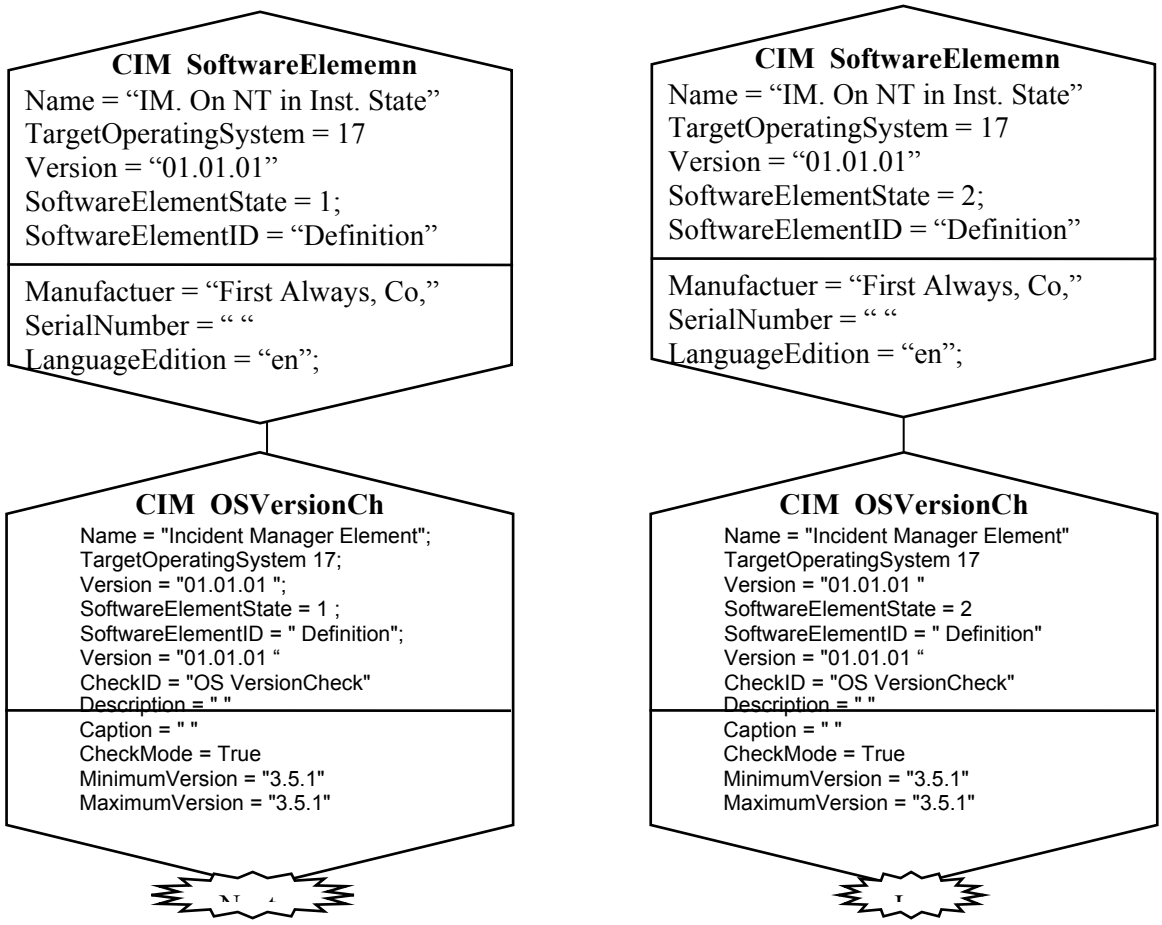


Figure 8. Check Object for NT/Executable Software Element

The following is the MOF for the two OSVersionCheck classes. Remember that the Phase property of the SoftwareElementChecks association is what determines whether a check is an in-state or next-state condition.

```
instance of CIM_OSVersionCheck as $chck001_02_N01
{
   Name = "Incident Manager Element"; // key
   TargetOperatingSystem = 17;
   Version = "01.01.01 ";
   SoftwareElementState = 1 ;
   SoftwareElementID = " Definition";
```

```
    Version = "01.01.01 "; // key
    CheckID = "OS VersionCheck";
    Description = " ";
    Caption = " ";
    CheckMode = True;
    MinimumVersion = "3.5.1";
    MaximumVersion = "3.5.1";
};
Instance of CIM_SoftwareElementChecks
{
    Element = $elmt001_02 ;
    Check = $chck001_02_N01 ;
    Phase = 1 ; // next-state
};

instance of CIM_OSVersionCheck as $chck001_03_I01
{
    Name = "Incident Manager Element"; // key
    TargetOperatingSystem = 17; // NT
    Version = "01.01.01 "; // key
    SoftwareElementState = 2 ; // Executable
    SoftwareElementID = " Definition"; // key
    Version = "01.01.01 "; // key
    CheckID = "OS VersionCheck";
    Description = " ";
    Caption = " ";
    CheckMode = True;
    MinimumVersion = "3.5.1";
    MaximumVersion = "3.5.1";
};

Instance of CIM_SoftwareElementChecks
{
     Element = $elmt001_03 ;
     Check = $chck001_03_I01 ;
     Phase = 0 ; // in-state
};
```

Table 9. Example SoftwareElementChecks

3.5 Decomposing Software Elements

The application management model allows details about the software element's bill of materials - a list of all files that make up the component – including executables, data files, scripts, and so on. These files are organized according to the target directory where they will reside in the target environment or system.

We model two aspects of the bill of materials: the directory structure of the software element using the DirectorySpecification class, and the file content of these directories using the FileSpecification class. The software element files are modeled in the context of

a directory structure so that it is easier to relocate these files. Rather than changing the path of every file, only the path for the directory must be changed.

The DirectorySpecification and FileSpecification classes are subclasses of Check, since they described conditions that can and must exist within the computer system environment.

The DirectorySpecification class has three properties: Index, Location Type, and Path. The Index property is a unique number assigned to a particular directory. The Location Type property describes the type of directory. The possible directory types are summarized in Table 10. The Path property allows an application provider to describe a default or recommended path for a particular directory. The value can be changed for a particular building block on a particular system.

Product Directories	**Shared Directories**	**System Directories**	**Miscellaneous Directories**
Product base directory Product executables directory Product library directory Product include directory Product configuration directory Product log directory Product working directory	Shared base directory Shared executables directory Shared library directory Share include directory	System base directory System executables directory System library directory System include directory System configuration directory System log directory	Unknown Other

Table 10. Directory Types for the Location Type Property

The files that are in a particular target directory are captured using the FileSpecification class and are referenced by the *DirectorySpecificationFile* association.

The following are files prepared by Installshield, ready to be copied to a distribution CD-ROM. This represents the software element for the Incident Management Feature in its Installable state.

```
04/20/97 10:48a 30,218 Custfir.bmp
03/24/98 07:15p 136 DATA.TAG
03/24/98 07:16p 12,281,942 data1.cab
04/20/97 04:03p 79 Install.ini
03/13/97 10:12a 352 lang.dat
03/24/98 07:16p 376 layout.bin
03/26/98 04:00p 0 list
03/19/97 01:52a 419 os.dat
03/24/97 04:21p 59,392 SETUP.EXE
03/24/98 07:15p 71 SETUP.INI
12/05/97 07:15p 92,007 setup.ins
03/24/98 07:16p 47 setup.lid
03/24/97 04:44p 316,220 _INST32I.EX_
03/24/97 04:21p 8,192 _ISDEL.EXE
03/24/97 04:20p 11,264 _SETUP.DLL
03/24/98 07:15p 181,841 _sys1.cab
03/24/98 07:15p 7,646 _user1.cab
```

This list would be modeled as a single directory with these 19 files, as shown in the Figure 9.

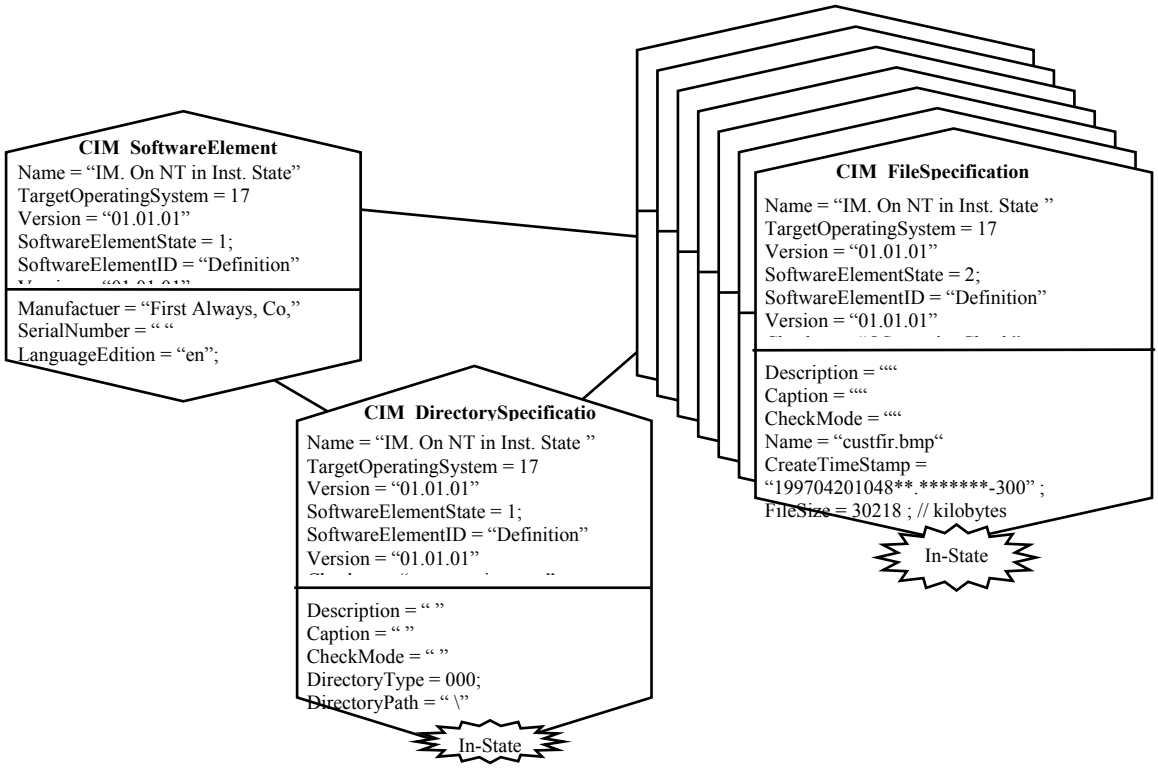


Figure 9. File List modeled as a single Directory

```
instance of CIM_SoftwareElementChecks
{
   Element =
   "
   Name = "IM. On NT in Depl. State",
   TargetOperatingSystem = 17,
   Version = "01.01.01 ",
   SoftwareElementState = 0 ,
   SoftwareElementID = " Definition",
   Version = "01.01.01 "
   ";

   Check = "
   Name = \" PT Primary Element\",
   TargetOperatingSystem = 17,
   Version = \"01.01.01\",
   SoftwareElementState = 0 ,
   SoftwareElementID = \"Definition\",
   Version = "01.01.01\",
   CheckID = \"OS VersionCheck\"
   ";
   Phase = 0 ; // in-state
};

 instance of CIM_DirectorySpecification as $dirspec_01
{
   Name = "IM. On NT in Depl. State"; // key
   TargetOperatingSystem = 17; // NT
   Version = "01.01.01 "; // key
   SoftwareElementState = 0 ; // Deployable
   SoftwareElementID = " Definition"; // key
   Version = "01.01.01 "; // key
   CheckID = "OS VersionCheck";
   Description = " ";
   Caption " ";
   CheckMode = True;

   DirectoryType =??;
   DirectoryPath = " " ;
};

instance of CIM_FileSpecification as $filespec_01
{
   Name = " ";
   CreateTimeStamp = "xx/xx/xx";
   FileSize=123; //kilobytes
   CheckSum = 00000;
   CRC1 = 0000;
   CRC2 = 0000 ;
   MD5Checksum = "1234567890123456";
};
```

```
class CIM_DirectorySpecificationFile
{
   DirectorySpecification = $dirspec_01;
   FileSpecification = $filespec_01;
};
```

Table 11. MOF describing a file list modeled as a single Directory

Although the model allows an application provider to capture directories and files in ways independent of physical location, there are situations in which the application provider will want to "hard code" the physical location of a file. This can be accomplished in the model by providing the complete file name in the file specification.

Component (Features)	Dependency	Main Executables
Advance GraphicsPackage part of	Requires Microsoft Excel Requires Problem Tracker feature	CFSRPT.XLA (installed as Excel add-in) DYNMCSQL.DLL LOADQDB.DLL

Table 12. CustomerFirst Client-side Features

3.6 Discovery

Sometimes, a developer must discover or detect software components on a set of machines. Developers typically accomplish this using distinguishing characteristics of critical files that a software component must have installed on the target system to function. These characteristics are referred to as a software signature. The file size, file time stamp, file checksum, CRC1 and CRC2 properties of the FileSpecification class form the software signature defined by the DMTF Software Standard Group Definition, Version 2.0.

The *CRC 1* property is determined by running an algorithm on a particular 512K bytes of the file. The CRC algorithm is specified by the ITU recommendation V.41.

For the CRC 1 value, the algorithm is run on the middle 512K bytes of the file. To determine the middle 512K bytes, do the following:

1. Identify the file's midpoint by dividing the file size by two and rounding down to the next whole number.
2. Take the 256K bytes on either side of the file's midpoint. The resulting 512K bytes is the portion of the file on which you should run the algorithm.

If the file is less than 512K bytes, then run the algorithm on the entire file.

For the CRC 2 value, the algorithm is run on the 512K bytes that result from the following procedure:

1. Divide the file by three and save the remainder.
2. Add the remainder to the beginning of the file (starting from zero). The result is the midpoint that is used to establish the 512K bytes.
3. Take the 256K bytes on either side of the midpoint determined in step 2. The resulting 512K bytes is the portion of the file on which you should run the algorithm.

If the file is less than 512K bytes, then run the algorithm on the entire file.

3.7 Shared Software Elements

After closer scrutiny, we determined that the executable software elements have shared files, so a shareable feature can be created.

The *CustomerFirst* application is mostly represented by components executing on the client (desktop) system. Typical setup is only one copy of the runtime components on a LAN shareable directory. All clients systems load the software from the shared area when *CustomerFirst* or its features are invoked.

Observe that some software elements are shared across software features. You can determine this from the model by locating software elements that reference more than one Software Feature.

Software Element	Dependency	Main Executables	Distribution
Core Functions including Database Access	Oracle Call Interface; Windows 95/NT	CSMDBIO.DLL CSMSUBS.DLL	CustomerFirst installation disks
CustomerFirst call tracking and defect tracking (consists of over 300 .dll's / the main executable loads the appropriate .dll as the feature is invoked. Each .dll represents a main UI panel and subordinate panels and dialog boxes.)	Core Functions; Windows 95/NT	CSF.EXE INCIDENT.DLL PROBLEM.DLL INCSERCH.DLL …	CustomerFirst installation disks
CustomerFirst e-mail facilities	CustomerFirst call tracking; one of the following client components: •MAPI-compliant mail software such as Microsoft Mail; Microsoft Exchange; Windows Messaging; Microsoft Outlook •VIM-compliant mail software such as Lotus CC:Mail •Lotus Notes Mail	RTIMAPI.DLL VIMMAIL.DLL LOTMAIL.DLL	CustomerFirst installation disks
CustomerFirst e-mail facilities (SMTP mail)	CustomerFirst call tracking; SMTP mail server	SMTPMAIL.DLL	CustomerFirst installation disks

CustomerFirst Excel Pivot Tables			CustomerFirst installation disks
CustomerFirst Notes Remote Administration	Lotus Notes client software; Core Functions	CFSNOTES.EXE RTINOTES.DLL	CustomerFirst installation disks
CustomerFirst Notes Remote	Lotus Notes client software	*.NSF and *.NTF files to be copied to Notes server	CustomerFirst installation disks
Lotus Notes	Lotus Notes server	<Notes client exe>	Lotus Notes CD

Table 13. Shared Software Elements

Client Component	**Dependency**	**Main Executables**	**Distribution**
VIM-compliant mail (CC:Mail)	Windows; supported network protocol such as IP.	??	Lotus CC:Mail CD or CustomerFirst installation disks.
MAPI-compliant mail software	E-Mail server supporting the MAPI software	??	E-mail software from the appropriate vendor.
Oracle Call Interface	Oracle IP Protocol Adapter	??	Oracle Client Software CD
Oracle IP Protocol Adapter	IP stack provided by Windows; IP network hardware	??	Oracle Client Software CD

Table 14. Client Components

3.8 Dependencies

We have learned how to describe dependencies that a software element might have on an operating system, such as amount of memory, disk space, etc. The application management information model also provides mechanisms to express higher-level dependencies. We can use the Advance Graphics Package feature to illustrate some of

these. In Table 12, the Advance Graphics Package feature has two dependencies: it requires Microsoft Excel and it requires the Problem Tracker feature. These represent inter-component or inter-feature dependencies. When application management models are available for both, a developer can use an instance of the SoftwareFeatureCheck class. However, in many cases an application model may not be available, as in the case of Microsoft Excel in this example. In this case, you can express dependencies using the DirectorySpecification and FileSpecification classes to locate one or more significant files for the Microsoft Excel product.

TBD	
Notes 1. The values for Vendor, ProductName, ProductNumber, IdentificationNumber, and Version are propagated keys from the CIM_Product object referenced by the CIM_ProductSoftwareFeatures asssocation.	

Table 15. Dependencies expressed by Directory and File Spec Checks

3.9 Application System Details

The CustomerFirst product might use several other products. These include a relational database product, an e-mail system, and Microsoft's Excel. Each of them would have a similar CIM_Product.

Instance diagram and MOF TBD

Acknowledgments

The Application Management Work Group's ability to develop this model was greatly enhanced by work that already existed in the industry. We would like to recognize these sources since they served as a source of ideas and provided experiences that separate concepts from practical implementation. These sources include the Posix 13786 standard, the DMTF's Software Standard Group Definitions, Microsoft's software and installation work for Windows 2000, and Tivoli's Application Management Specification Version 2.0.

Appendix A – Change History

Version 1.0	May 17, 1998	Initial Version
Version 1.1	May 25, 2002	Update to reflect changes to the model effective CIM 2.6. Note that changes (See the application MOF for details) are not reflected in the model descriptions in this version of this document. Also removed the section which was to have contained the complete Application Model UML.
Version 1.2	June 12, 2003	Update Document to CIM 2.7 Clean up graphics and text. Add comments on the new components of the model (bios) and the relationship between the model and identity.

Appendix B – List of Files for Full-Featured Installation

The following is a list of files of a full-featured installation of *CustomerFirst.* This represents the application in its executable state.

01/01/80 12:00a <DIR> .
01/01/80 12:00a <DIR> ..
01/30/98 10:43a 82,944 PRODPGS.DLL
03/24/98 07:48p 5,632 RTIOBJ.DLL
01/30/98 10:36a 43,008 INCOMORD.DLL
01/30/98 11:05a 52,736 MSRCRES.DLL
01/30/98 02:16p 26,112 RTIEDIT.DLL
03/24/98 08:04p 129,024 ADM.EXE
01/09/98 02:57p 8,192 CSMUSER.DLL
03/24/98 09:09p 4,608 MHSHIELD.DLL
03/24/98 07:48p 5,632 RTIOUT.DLL
02/15/95 01:11a 17,920 IMPLODE.DLL
08/24/96 11:11a 326,656 MSVCRT40.DLL
07/19/96 04:25p 267,536 MSVCRT.DLL
01/09/98 02:52p 4,096 RTIDLG.DLL
01/09/98 04:09p 3,507,458 SYSADMIN.HLP
01/30/98 10:47a 28,672 R0006.DLL
01/10/98 04:48p 8,249,538 CFS.HLP
03/24/98 09:26p 49,152 ORDPRT.DLL
03/24/98 07:46p 886,784 CSMSUBS.DLL
09/30/97 06:42p 46,320 MAGMACTL.DLL
03/16/98 01:10p 18,432 RTITOOLS.DLL
01/30/98 10:20a 60,416 EQMODEL.DLL
01/30/98 09:59a 52,736 SFSALHIS.DLL
01/30/98 10:11a 376,832 PFIX.DLL

01/30/98 10:19a 63,488 CUSTALIS.DLL
03/24/98 08:03p 195,584 TRUECFSD.DLL
03/24/98 08:09p 122,880 ALERT.EXE
03/24/98 08:02p 151,552 CFS.EXE
03/24/98 09:22p 437,760 CFSNOTES.EXE
01/30/98 11:02a 55,296 ENNCTIME.DLL
01/30/98 09:59a 40,960 SFLTSENT.DLL
03/13/98 10:22p 34,304 CONVERT.EXE
03/24/98 08:04p 128,512 CTLADM.EXE
01/30/98 10:58a 131,072 WORKITEM.DLL
03/24/98 08:09p 122,880 CTLALRT.EXE
04/16/97 11:01a 8,192 CSMUSER.BAK
01/30/98 02:23p 41,984 SUBSYSCL.DLL
09/24/97 04:13p 83,216 ALTV3-0.OR7
05/29/97 08:39a 52,224 ISRCHRES.DLL
08/13/97 01:59p 81,142 ALTV3-0.SAV
08/20/97 08:59a 134,144 LAYOUT.BAK
03/24/98 08:02p 151,040 CTLCFS.EXE
11/17/97 11:52a 329 CFS1.LNK
03/24/98 08:14p 135,680 CTLMAIL.EXE
03/24/98 08:06p 125,440 CTLRPT.EXE
03/24/98 08:11p 128,512 CTLSSRV.EXE
03/24/98 08:11p 128,512 ESSERVER.EXE
03/24/98 08:10p 124,928 HTMLGEN.EXE
03/24/98 09:13p 152,064 INCIPT.EXE
03/24/98 08:13p 135,680 INMAIL.EXE
03/16/98 12:37p 108,032 MAILSTAT.EXE
03/24/98 08:07p 154,624 OPS.EXE
09/27/96 11:22a 580 LCABOX.EXP
03/24/98 08:06p 125,440 RPT.EXE
12/19/96 09:53p 2,054 RTILOAD.LIB
01/30/98 10:41a 55,296 SELCUST.DLL
01/30/98 10:32a 127,488 CHARMOD.DLL
01/30/98 10:31a 82,944 CHARTSET.DLL
01/30/98 10:33a 145,408 CHARVSET.DLL
01/30/98 11:08a 49,152 CNTPDINT.DLL
03/24/98 09:31p 263,680 RTIIMP.EXE
03/24/98 09:19p 47,104 RTICONV.DLL
03/13/98 10:43p 120,320 RTINOTES.DLL
03/24/98 09:40p 134,144 LAYOUT.DLL
03/16/98 12:42p 41,984 RTILIC.EXE
03/18/98 07:37p 97,792 MNTSERCH.DLL
07/28/97 03:52p 34,816 CFSSCI.DLL
06/09/96 03:40p 22,478 CTLFIR.BMP
01/30/98 10:37a 61,952 QUOFCUST.DLL
01/30/98 11:04a 59,392 CNTRYLST.DLL
03/17/98 11:17a 52,736 CONSRCR.DLL
07/09/97 01:48a 49,664 TSKSRES.DLL
01/30/98 10:22a 74,752 CONTEQU.DLL
01/30/98 10:46a 19,968 R0003.DLL
09/23/94 06:07p 115,462 NEWCFS.BMP
01/30/98 11:15a 39,936 RTIDATA.DLL
03/17/98 11:16a 167,936 CONTSRCH.DLL
01/30/98 10:57a 102,400 CPERNOTE.DLL
01/30/98 11:07a 59,904 MCACTIVE.DLL
01/30/98 10:25a 68,608 CPUCHAR.DLL
03/24/98 09:26p 12,288 CSMLOGON.DLL
03/24/98 09:09p 119,296 RTILOAD.EXE
03/24/98 08:05p 186,880 CTLADMD.DLL
03/24/98 08:09p 152,064 CTLALRTD.DLL
03/24/98 08:03p 207,872 CTLCFSD.DLL
01/30/98 11:09a 53,760 MSCHRGRN.DLL
04/16/97 10:59a 1,988 CSMUSER.LIB
03/24/98 09:15p 152,576 RTIUPG.EXE
03/24/98 08:14p 145,920 SLS.EXE
07/04/97 11:26a 254,857 CFSUSER.HLP
03/04/98 03:09p 329 CFS2.LNK
03/24/98 08:09p 122,880 TRUALRT.EXE
07/09/97 01:57a 46,592 PFSRCRES.DLL
01/30/98 10:25a 75,264 LICALL.DLL
01/30/98 10:34a 78,848 CNTRYDEF.DLL
03/24/98 08:06p 159,744 CTLRPTD.DLL
01/30/98 10:17a 47,616 DEBUG.DLL
03/24/98 08:04p 115,712 TRUEADM.EXE
07/09/97 01:47a 56,320 CSRCRES.DLL

01/30/98 10:27a 97,280 PERLIST.DLL
03/24/98 08:02p 138,752 TRUECFS.EXE
01/30/98 11:15a 116,736 LITSEND.DLL
03/24/98 08:06p 159,744 TRUERPTD.DLL
09/30/97 04:52p 840,192 CSMSUBS.BAK
03/24/98 09:41p 31,232 RTITRUE.DLL
01/30/98 10:18a 55,296 CUSTLIC.DLL
01/30/98 10:57a 102,400 CUSTNOTE.DLL
02/25/98 01:14p 137,216 CUSTPGS.DLL
02/25/98 01:10p 156,672 CUSTPRDS.DLL
03/24/98 09:35p 31,744 RTIMAPI.DLL
01/30/98 10:37a 62,464 QUOFCONT.DLL
06/11/97 09:15a 834 SERCHGEN.SKL
01/30/98 10:28a 86,016 DEPTLIST.DLL
03/16/98 02:19p 131,584 ITEM.DLL
01/30/98 10:31a 71,680 DFDEPT.DLL
01/30/98 10:35a 140,800 DISCSCHD.DLL
01/30/98 10:08a 70,656 DOCREF.DLL
03/18/98 07:38p 253,440 PDFAMILY.DLL
01/30/98 11:03a 78,848 ENINTIME.DLL
01/30/98 11:03a 75,776 ENPRTIME.DLL
01/30/98 11:00a 164,864 ENTYTASK.DLL
03/18/98 07:36p 80,896 ENTYTIME.DLL
01/30/98 10:24a 60,928 EQIPHIST.DLL
01/30/98 10:23a 53,248 EQSRCRES.DLL
05/29/97 08:29a 85,504 SEARCH.DLL
01/30/98 10:58a 66,048 EQUCHAR.DLL
01/30/98 10:23a 433,152 EQUIP.DLL
01/30/98 10:49a 27,136 R0010.DLL
01/30/98 10:22a 130,560 EQUQUERY.DLL
03/18/98 07:08p 7,680 ERROBJ.DLL
05/14/97 11:16a 254,072 TRUETREK.BMP
01/14/97 04:49p 136,704 RTITIMED.DLL
03/24/98 08:12p 271,872 ESSERVED.DLL
01/30/98 10:29a 80,896 SETUP.DLL
07/16/97 10:51a 4,648 RTILOAD.LOG
01/30/98 10:59a 72,704 SUMPTASK.DLL
01/30/98 10:26a 93,184 FLDVAL.DLL
01/30/98 11:14a 71,680 FORMDEF.DLL
03/13/98 09:10p 110,592 GRPSETUP.DLL
03/06/98 12:59p 101,306 MAINTCST.LOG
01/30/98 10:46a 26,624 R0002.DLL
01/30/98 10:57a 49,152 USERPRVL.DLL
03/24/98 08:10p 151,040 HTMLGEND.DLL
03/24/98 09:30p 144,896 IMPFUNCS.DLL
01/30/98 10:07a 70,656 INCCORR.DLL
01/30/98 10:16a 46,592 INCEQU.DLL
03/24/98 08:15p 288,768 SFSALOPP.DLL
03/24/98 09:32p 265,216 TRUEIMP.EXE
03/24/98 09:24p 73,728 INCFUNC.DLL
01/30/98 10:36a 61,952 ORDFCUST.DLL
01/30/98 10:21a 73,216 INVCHECK.DLL
03/13/98 09:04p 689,664 INCIDENT.DLL
01/30/98 11:16a 53,760 LETRCOMP.DLL
01/30/98 10:09a 175,616 INCSERCH.DLL
03/24/98 08:14p 208,896 INMAILD.DLL
02/25/98 01:13p 88,064 INPDPGS.DLL
01/30/98 11:14a 46,592 SHOWUSER.DLL
01/30/98 10:38a 49,664 INVQUERY.DLL
03/24/98 09:24p 140,800 ORDER.DLL
03/24/98 09:27p 8,704 OBJFUNC.DLL
03/24/98 09:13p 95,232 LCABOX.DLL
03/13/98 09:05p 50,688 LINKUNL.DLL
01/30/98 11:00a 97,792 LISTTASK.DLL
01/30/98 10:40a 51,200 PRSKUQRY.DLL
01/30/98 10:51a 29,184 R0015.DLL
01/30/98 09:58a 69,632 SFSALSTG.DLL
01/30/98 10:53a 18,944 R0018.DLL
01/30/98 10:48a 31,232 R0007.DLL
03/24/98 09:42p 6,144 LOADQDB.DLL
03/24/98 09:33p 47,104 LOTMAIL.DLL
01/30/98 10:53a 20,480 R0020.DLL
03/13/98 09:14p 81,920 MAINTCST.DLL
06/12/97 08:35a 3,291 CHARGEN.SKL
01/30/98 11:08a 518,656 MCCONT.DLL

03/24/98 09:35p 74,240 SMTPMAIL.DLL
01/30/98 10:12a 45,568 MODHIST.DLL
01/30/98 10:39a 122,880 OPDEF.DLL
03/24/98 08:07p 198,656 OPSD.DLL
09/29/95 10:25p 439,296 MSVCR40D.DLL
01/30/98 10:36a 62,464 ORDFCONT.DLL
01/30/98 10:15a 52,736 OWNINCS.DLL
01/30/98 10:16a 48,128 OWNPROB.DLL
01/30/98 10:44a 42,496 PDCTPGS.DLL
03/24/98 07:47p 6,144 PDFSTR.DLL
01/30/98 10:55a 39,936 PERSONCL.DLL
01/30/98 10:14a 99,328 PFIXSRCH.DLL
01/30/98 10:53a 19,456 R0021.DLL
01/30/98 10:39a 154,112 PPSCHED.DLL
03/24/98 09:25p 20,992 PRICING.DLL
01/30/98 10:56a 67,072 PRIVILCL.DLL
01/30/98 10:34a 49,664 PROBEXT.DLL
01/30/98 10:12a 52,736 PROBINC.DLL
03/24/98 09:28p 43,008 TRUELIC.EXE
01/30/98 10:42a 69,632 ALERTLST.DLL
01/30/98 10:04a 72,704 ALRTINFO.DLL
01/30/98 10:29a 39,424 CPRDLIC.DLL
01/30/98 10:19a 88,064 CUSTCPU.DLL
01/30/98 10:59a 59,904 SUMDTASK.DLL
03/24/98 08:18p 223,232 CUSTINFO.DLL
03/24/98 08:09p 152,064 TRUALRTD.DLL
01/30/98 10:18a 299,520 LICENSE.DLL
01/30/98 11:06a 122,368 MCTYPES.DLL
09/26/96 04:48p 15,360 PKWDCL.DLL
01/30/98 10:03a 331,264 PROBLEM.DLL
04/14/97 10:38a 18,944 DYNMCSQL.DLL
01/30/98 10:13a 145,920 PROBSRCH.DLL
01/30/98 09:57a 101,376 SFSCOMP.DLL
03/25/98 06:51p 132,884 ESSERVER.LOG
02/25/98 01:09p 264,704 PRODUCT.DLL
03/24/98 09:10p 121,856 TRUELOAD.EXE
01/30/98 10:45a 34,816 R0001.DLL
01/30/98 10:47a 24,064 R0004.DLL
12/19/96 09:53p 366 RTILOAD.EXP
01/30/98 10:47a 21,504 R0005.DLL
01/30/98 10:49a 23,040 R0009.DLL
01/30/98 10:10a 122,880 TSKSRCH.DLL
07/09/97 01:56a 54,784 PSRCHRES.DLL
01/30/98 10:48a 23,040 R0008.DLL
01/30/98 10:30a 58,880 SETUP2.DLL
01/30/98 10:50a 23,552 R0011.DLL
03/18/98 07:31p 159,232 TEMPFIX.DLL
01/30/98 10:50a 24,064 R0012.DLL
01/30/98 10:50a 25,600 R0013.DLL
01/30/98 10:51a 29,184 R0014.DLL
03/18/98 07:35p 20,480 R0016.DLL
01/30/98 10:52a 19,456 R0017.DLL
01/30/98 10:54a 30,720 R0022.DLL
01/30/98 10:54a 25,088 R0023.DLL
01/30/98 10:55a 35,328 R0024.DLL
01/30/98 11:11a 27,648 R0025.DLL
01/30/98 11:11a 32,768 R0026.DLL
01/30/98 11:12a 20,992 R0027.DLL
01/30/98 10:10a 92,672 RECENT.DLL
01/30/98 10:11a 88,576 RECENTC.DLL
01/30/98 10:28a 64,000 REFRCUST.DLL
01/30/98 10:12a 40,448 REFRESH.DLL
04/16/97 10:59a 323 CSMUSER.EXP
01/30/98 10:07a 52,736 RELATED.DLL
01/30/98 10:24a 57,856 REPLEQU.DLL
02/10/98 05:02p 659,968 REPORTS.DLL
01/30/98 10:03a 41,984 SFLTPEND.DLL
01/30/98 10:08a 108,544 RESSRCH.DLL
06/02/96 02:30p 30,218 SALEFIR.BMP
03/13/98 09:00p 46,592 SFALLSOP.DLL
03/16/98 12:41p 26,624 ATTBOX.DLL
01/30/98 10:17a 32,768 RMASRCH.DLL
03/24/98 08:06p 159,744 RPTD.DLL
03/16/98 01:05p 424,448 RTICTL.DLL
03/24/98 08:06p 125,440 TRUERPT.EXE

01/30/98 10:05a 111,616 WIQYIVW.DLL
03/24/98 09:37p 34,304 RTITIMER.DLL
01/30/98 10:40a 85,504 SKUINFO.DLL
01/30/98 10:42a 144,384 RULELIST.DLL
02/10/98 10:53a 44,032 SCANCHAR.DLL
01/30/98 10:04a 52,736 USERSET.DLL
03/24/98 08:14p 199,168 SLSD.DLL
01/30/98 10:01a 49,152 SERCHRES.DLL
04/13/97 05:14a 35,840 MCMCHIS.DLL
01/30/98 10:02a 75,264 SFSALFOR.DLL
01/30/98 11:13a 62,976 TFIXCUST.DLL
02/16/98 07:35p 96,256 TMHLTASK.DLL
01/30/98 10:00a 59,904 SFSALLED.DLL
03/24/98 08:05p 174,592 TRUEADMD.DLL
04/13/97 05:37a 56,320 MNTCPER.DLL
01/30/98 10:01a 130,048 SFSAOPSH.DLL
03/24/98 08:17p 87,040 SFSLPCAL.DLL
10/17/96 07:09p 42,496 MCTERM.DLL
01/30/98 11:15a 49,152 SFSRECNT.DLL
03/24/98 08:14p 208,896 TRUMAILD.DLL
03/24/98 08:12p 271,872 TRUSSRVD.DLL
01/30/98 10:26a 39,936 USERCL.DLL
01/30/98 10:20a 70,144 VENDOR.DLL
03/24/98 09:35p 39,936 VIMMAIL.DLL
06/09/96 03:46p 31,494 OPS.BMP
06/02/96 02:30p 30,218 CUSTFIR.BMP
03/24/98 07:14p 3,784,192 CSMDBIO.DLL
01/30/98 10:42a 49,152 ALERTOPT.DLL
03/24/98 08:03p 208,384 CFSD.DLL
03/24/98 08:12p 271,872 CTLSSRVD.DLL
01/30/98 10:21a 83,968 CUSTEQU.DLL
03/24/98 09:28p 39,424 MAILSTAD.DLL
03/13/98 09:12p 81,920 NEWCUST.DLL
01/30/98 10:00a 92,672 SFFILMNT.DLL
01/30/98 10:35a 62,976 CTYPDEF.DLL
01/30/98 10:14a 133,120 CUSTCONT.DLL
01/30/98 11:07a 50,688 MCCHRGI.DLL
01/30/98 11:09a 53,760 MSCHRGCT.DLL
01/30/98 10:33a 38,400 SCANKWD.DLL
03/26/98 12:21p 184,320 CONTLIST.DLL
03/24/98 09:17p 154,112 TRUEUPG.EXE
03/24/98 08:14p 135,680 TRUMAIL.EXE
03/24/98 08:12p 128,512 TRUSSRV.EXE
01/30/98 11:01a 86,016 ACTYTASK.DLL
03/24/98 08:04p 187,392 ADMD.DLL
03/24/98 08:14p 208,896 CTLMAILD.DLL
01/30/98 10:41a 403,968 ORDENTRY.DLL
03/24/98 08:09p 152,064 ALERTD.DLL
02/16/98 07:34p 99,328 EVHLTASK.DLL
01/30/98 10:09a 145,408 CSTSERCH.DLL
03/13/97 03:38a 59,904 WIQYPVW.DLL
03/13/97 03:39a 52,736 WIQYTVW.DLL
03/13/97 04:05a 57,856 DEPTINC.DLL
03/13/97 04:06a 51,712 DEPTPRB.DLL
03/13/97 04:06a 46,592 DEPTPFX.DLL
332 File(s) 51,331,914 bytes
469,073,920 bytes free

Oracle 8 File List

The following is a complete list of server exe & dll's for Oracle 8 on a WindowsNT system.

Directory of C:\ORANT\BIN

03/25/98 12:44p <DIR> .
03/25/98 12:44p <DIR> ..

01/31/97 04:18p 55,296 AUTO32W.DLL
06/13/97 05:36p 37,376 BCKRSUS.DLL
06/12/97 09:19p 26,624 CheckerClass.dll
06/13/97 08:21p 344,064 CORE40.DLL
03/31/96 10:44a 27,136 CTL3D32.NEW
06/09/97 04:44p 1,113,600 CTXADMIN.EXE
06/08/97 08:40p 12,288 CTXCOM80.DLL
06/08/97 08:39p 17,920 CTXCTL80.EXE
06/08/97 06:16p 27,984,384 CTXDAT80.DLL
06/09/97 04:44p 43,520 CTXIO32.EXE
06/09/97 04:44p 63,488 CTXIO32L.DLL
06/08/97 08:21p 3,073,024 CTXLDR80.EXE
06/08/97 06:17p 10,752 CTXPIP80.EXE
06/08/97 08:20p 3,080,704 CTXSRV80.EXE
06/08/97 08:23p 2,368,000 CTXSRX80.EXE
06/08/97 08:40p 17,920 CTXSVC80.EXE
06/09/97 04:44p 906 CTXV32.CNT
06/09/97 04:44p 38,153 CTXV32.HLP
06/09/97 04:44p 78,336 CTXV32.OCX
06/09/97 04:44p 4,697 CTXV32.TLB
06/16/97 01:45p 22,016 DBAssist.exe
06/13/97 10:58a 18,944 dbmig.exe
06/18/97 01:50p 36,352 DBVERF80.EXE
06/09/97 04:44p 50,688 DRDS.DLL
06/17/97 05:45p 7,680 EVR80US.DLL
06/18/97 01:38p 177,664 EXP80.EXE
06/18/97 01:44p 34,816 EXTPROC.EXE
06/18/97 01:45p 164,864 HOAGTOCI.EXE
06/09/97 04:27p 18,944 IFA80.EXE
06/18/97 01:55p 78,336 IMP80.EXE
05/22/97 02:31p 31,232 JNLSLib3.dll
05/06/97 11:21a 23,040 JNLSTool.dll
06/17/97 10:16a 70,144 LSNRCTL80.EXE
01/31/97 01:24p 941,840 MFC42.NEW
06/20/97 12:16p 72,704 MIG80.EXE
06/16/97 09:21p 426 MMDX.LIC
06/17/97 12:32a 70,656 MMDX32.OCX
06/16/97 09:21p 423 MMTX.LIC
06/16/97 09:21p 514,048 MMTX32.OCX
08/09/96 01:00a 74,752 MSVCIRT.NEW
01/22/97 11:07p 271,632 MSVCRT.NEW
02/20/96 01:03a 326,656 MSVCRT40.NEW
09/27/96 05:30p 435,200 MTLD.DLL
05/28/97 07:22p 19,968 N8A.EXE
06/04/97 02:38p 19,968 N8SW.EXE
06/14/97 11:42p 179,712 NAMES80.EXE
06/14/97 03:44p 53,248 NAMESCTL80.EXE
06/20/97 05:46p 252,928 NASNS80.DLL
06/14/97 03:43p 8,704 NAUN80.DLL
06/14/97 03:43p 16,384 NAUNTS80.DLL
06/14/97 03:43p 337,920 NCR80.DLL
05/16/97 05:39p 34,816 NDWSI80.DLL
06/19/97 04:25p 64,512 NI80.DLL
06/19/97 04:44p 136,192 NL80.DLL
06/13/97 07:54p 223,232 NLSRTL33.DLL
06/16/97 09:21p 88,064 NLSWIN.DLL

06/14/97 03:43p 14,336 NMP80.DLL
06/14/97 03:43p 10,752 NMS80.DLL
06/14/97 03:43p 45,056 NNCI80.DLL
06/14/97 03:43p 5,120 NNFD80.DLL
06/14/97 03:43p 29,696 NNFG80.DLL
06/14/97 03:43p 16,384 NNFN80.DLL
06/14/97 03:43p 121,344 NNG80.DLL
06/14/97 03:43p 17,408 NPL80.DLL
06/14/97 03:43p 119,296 NR80.DLL
06/14/97 03:41p 294,400 NS80.DLL
06/17/97 09:09p 27,648 NT80.DLL
06/18/97 03:41p 25,088 NTN80.DLL
06/18/97 03:41p 17,920 NTP80.DLL
06/14/97 03:44p 34,816 NTS80.DLL
06/17/97 09:08p 53,248 NTT80.DLL
06/18/97 03:41p 25,088 NTUS80.DLL
06/14/97 03:43p 346,624 NZ80.DLL
06/17/97 11:03p 6,656 O80DBNOP.DLL
06/17/97 11:03p 6,656 O80DISOP.DLL
06/17/97 11:03p 6,656 O80HETOP.DLL
06/17/97 11:03p 6,656 O80OBJOP.DLL
06/17/97 11:03p 6,656 O80PAROP.DLL
06/17/97 11:03p 6,656 O80REPOP.DLL
06/08/97 06:10a 6,656 O80SPAOP.DLL
06/20/97 06:45p 6,656 O80VSNOP.DLL
06/16/97 02:14p 41,984 OCI803JDBC.DLL
06/20/97 01:30p 17,920 OCIW32.DLL
06/18/97 03:29a 17,408 OCOPY80.EXE
06/20/97 09:10p 92,160 OIBKND32.DLL
06/11/97 05:24p 211,968 OICORNLS.DLL
06/13/97 04:59p 192,512 OICTRL32.DLL
06/11/97 05:30p 85,504 OIDRSUS.DLL
06/20/97 11:30a 76,800 OIEXEC.DLL
06/12/97 12:31p 284,160 OIIMPL32.DLL
06/20/97 01:36p 27,136 OIIPI.DLL
06/11/97 05:30p 14,848 OIMRSUS.DLL
05/09/97 03:38p 290,816 OIP21.DLL
05/07/97 04:21p 24,981 OIP21.TLB
05/21/97 05:08a 21,504 OISBRUS.DLL
05/22/97 07:59a 13,824 OISFP10.EXE
05/21/97 05:09a 124,928 OISIIFAC.DLL
05/21/97 05:07a 17,408 OISIIFC.DLL
05/22/97 07:59a 93,696 OISIIMP.DLL
05/22/97 07:52a 22,016 OISLRUS.DLL
05/21/97 05:06a 44,544 OISOIFC.DLL
05/22/97 07:59a 151,040 OISOIMP.DLL
05/22/97 07:59a 34,816 OISOMIMP.DLL
05/22/97 07:59a 86,528 OISSAC.DLL
05/22/97 07:50a 115,200 OISWIZ.DLL
06/19/97 12:12p 151,040 OIUIMFC.DLL
06/13/97 12:23p 13,824 OMRESUS.DLL
06/14/97 11:42p 141,312 ONRSD80.EXE
06/16/97 01:10p 24,576 OPERF80.DLL
06/20/97 12:49p 1,646,592 ORA803.DLL
11/16/95 09:53a 148,992 ORAANSI.DLL
06/20/97 06:44p 8,371,200 ORACLE80.EXE

04/16/96 04:28p 386 ORADC.LIC
04/08/97 03:56p 87,040 ORADC.OCX
06/18/97 02:58p 161,280 ORADIM80.EXE
06/12/97 12:32p 76,800 ORAINST.EXE
06/18/97 03:29a 7,168 ORAKILL.EXE
06/17/97 11:03p 11,264 ORAPWD80.EXE
06/17/97 11:02p 8,192 ORASPAWN.DLL
06/16/97 07:56p 9,216 ORASTACK.EXE
06/14/97 07:42a 255,488 ORDIMG80.DLL
06/17/97 11:00a 5,510 OSM.CNT
06/14/97 12:46a 1,042,432 OSM.EXE
06/17/97 11:07a 91,319 OSM.HLP
06/14/97 12:47a 729,600 OSM.OCX
07/09/96 06:01p 552 OSMOCX.CNT
07/09/96 06:01p 26,339 OSMOCX.HLP
06/14/97 03:44p 6,656 OSSLOGIN.EXE
06/18/97 04:53p 118,272 OTRACE80.DLL
06/17/97 02:47a 18,944 OTRCCOL.EXE
06/17/97 02:37a 4,608 OTRCCREF.EXE
06/17/97 02:41a 6,144 OTRCFMT.EXE
06/17/97 02:44a 36,864 OTRCREP.EXE
06/02/97 10:47p 33,792 OTT.EXE
06/18/97 05:49p 72,192 OTT80.EXE
05/29/97 02:33p 33,280 OWASMUS.DLL
06/03/97 12:02p 164,352 OWAST.EXE
06/20/97 12:10p 51,200 OWASTCON.EXE
06/20/97 12:10p 58,880 OWASTS.EXE
06/20/97 12:10p 31,744 OWASTSVR.DLL
06/20/97 12:10p 104,448 OWASTSVR.EXE
06/03/97 12:00p 373,760 OWASTUS.DLL
10/01/96 05:40p 2,495 OWSMGR.REG
06/17/97 12:24p 9,728 OWSMGR.WRI
06/20/97 04:10p 1,717,248 PLS803.DLL
06/19/97 04:18p 285,696 PLUS80.EXE
06/18/97 07:57p 329,216 PLUS80W.EXE
06/17/97 05:35a 51,712 PRST.OCX
06/17/97 05:34a 10,240 PRSTRUS.DLL
06/17/97 01:53a 63,488 PRT.OCX
06/17/97 05:31p 626,688 PSTD803.DLL
06/13/97 05:51p 41,472 RCVRSUS.DLL
10/06/95 12:02a 24,064 REGSVR32.EXE
06/01/97 04:17p 7,203 REPMGR.CNT
06/13/97 05:29p 736,256 REPMGR.EXE
06/04/97 02:26p 322,227 REPMGR.HLP
03/13/96 04:18p 690 REPMGR.REG
06/18/97 03:44p 10,240 REPMGR.WRI
06/13/97 05:14p 646,656 REPMUS.DLL
06/14/97 04:24p 11,776 RHSLF80.EXE
06/18/97 01:45p 778,752 RMAN80.EXE
06/14/97 04:24p 144,896 SDCONV80.EXE
06/14/97 04:24p 175,616 SDLOAD80.EXE
06/14/97 04:24p 76,800 SDO80.DLL
06/14/97 04:24p 9,216 SDPWD80.EXE
06/14/97 03:43p 8,192 SNAUM080.DLL
06/18/97 01:29p 202,752 SQLLDR80.EXE
06/19/97 02:47p 98,816 SQLLIB80.DLL

06/18/97 02:46p 38,400 SQLPLUS.EXE
06/05/97 02:29p 5,632 STRTDB80.EXE
06/18/97 03:00a 128,000 SVRMGR30.EXE
06/13/96 05:30p 187,392 TCL73.DLL
03/25/98 12:44p 6,144 testexpdat.txt
03/25/98 01:10p 6,144 testexpdata.dmp
06/18/97 03:28a 52,736 TKPROF80.EXE
06/17/97 10:16a 124,928 TNSLSNR80.EXE
06/14/97 03:44p 16,384 TNSPING80.EXE
06/14/97 03:47p 86,016 TRCASST.EXE
06/13/97 03:28p 1,460 TRCFMT.CMD
06/14/97 01:25a 477,184 UIRESUS.DLL
10/07/96 10:32p 92,160 UNZIP.EXE
06/17/97 03:26a 107,008 VAC.EXE
06/17/97 03:22a 20,480 VACRUS.DLL
06/17/97 03:47a 58,368 VAD.EXE
06/17/97 03:44a 16,896 VADRUS.DLL
06/17/97 12:11a 427,008 VAEASQL.DLL
06/17/97 12:11a 8,192 VAERUS.DLL
06/17/97 03:33a 87,040 VAG.EXE
06/17/97 03:30a 22,016 VAGRUS.DLL
06/17/97 03:36a 78,848 VAI.EXE
06/17/97 03:33a 21,504 VAIRUS.DLL
06/17/97 12:13a 67,584 VAO.DLL
06/17/97 12:39a 39,936 VAOB.DLL
06/17/97 12:38a 7,680 VAOBRUS.DLL
06/17/97 12:37a 215,040 VAOC.DLL
06/17/97 12:32a 22,016 VAOCRUS.DLL
06/17/97 03:44a 374,272 VAOD.DLL
06/17/97 03:37a 219,648 VAODRUS.DLL
06/17/97 12:41a 117,248 VAOG.DLL
06/17/97 12:39a 12,288 VAOGRUS.DLL
06/17/97 01:08a 303,104 VAOI.DLL
06/17/97 01:01a 103,936 VAOIRUS.DLL
06/17/97 01:00a 900,608 VAOS.DLL
06/17/97 12:41a 204,800 VAOSRUS.DLL
06/17/97 03:22a 59,904 VAP.DLL
06/17/97 03:20a 22,016 VAPRUS.DLL
06/17/97 04:31a 872,960 VAR.EXE
06/17/97 03:51a 799,232 VARRUS.DLL
06/17/97 03:30a 160,768 VAS.EXE
06/17/97 03:27a 18,432 VASRUS.DLL
06/17/97 03:51a 92,672 VAW.EXE
06/17/97 03:47a 31,744 VAWRUS.DLL
06/17/97 12:31a 173,056 VAXA.DLL
06/17/97 12:28a 30,720 VAXARUS.DLL
06/17/97 12:00a 126,976 VAXC.DLL
06/17/97 12:13a 24,064 VAXCRUS.DLL
06/17/97 12:16a 211,456 VAXCT.DLL
06/16/97 11:59p 65,536 VAXN.DLL
06/17/97 12:17a 53,248 VAXS.DLL
06/17/97 01:51a 67,072 VAXT.DLL
06/17/97 01:50a 7,680 VAXTRUS.DLL
06/16/97 11:58p 28,672 VAXX.DLL
01/31/97 04:18p 469,504 VIEW32W.DLL
06/17/97 05:33a 60,416 VMEM.EXE

06/17/97 01:11a 62,464 VMEP.EXE
06/17/97 01:09a 29,696 VMEPC.DLL
06/19/97 05:05p 83,968 VOBMGR.DLL
06/20/97 02:22p 10,240 VOBSH.EXE
06/17/97 02:16a 996,864 VOC.EXE
06/17/97 01:12a 313,856 VOCRUS.DLL
06/19/97 03:53p 16,849 VOCUS.CNT
06/17/97 07:05p 508,065 VOCUS.HLP
06/17/97 01:49a 292,352 VOD.EXE
06/17/97 05:31a 178,176 VODA.EXE
06/17/97 05:32a 14,848 VODARUS.DLL
06/17/97 01:20a 433,152 VOJT.OCX
06/17/97 01:12a 45,568 VOJTRUS.DLL
06/17/97 05:37a 103,424 VOM.EXE
06/17/97 05:35a 80,384 VOMRUS.DLL
06/17/97 01:52a 110,592 VOT.OCX
06/17/97 12:28a 39,424 VOX.DLL
06/17/97 05:42a 65,024 VOXD.DLL
06/19/97 09:56a 225,280 VSBCK80.EXE
06/19/97 10:45a 223,232 VSRCV80.EXE
01/31/97 04:18p 90,112 W001F32W.DLL
01/31/97 04:18p 62,464 W003F32W.DLL
01/31/97 04:18p 97,792 W4W909F.DLL
06/17/97 05:22p 388,608 WRAP80.EXE
06/03/97 03:48p 38,400 WRQUICK.EXE
06/17/97 03:18p 64,512 XA80.DLL
06/19/97 05:06p 40,448 XPCRT.DLL
06/19/97 05:03p 1,086,304 XPFC.DLL
06/19/97 05:01p 859,796 XPFCUS.DLL
06/19/97 05:04p 57,344 XPGUI.DLL
06/19/97 05:01p 38,400 XPGUIUS.DLL
06/19/97 05:04p 394,240 XPLBSQO.DLL
10/07/96 10:32p 91,136 ZIP.EXE
259 File(s) 82,332,701 bytes
2,308,807,680 bytes free

CustomerFirst Management Object Format (MOF)

```
// ***********************************************
// Product: Customer First
// ***********************************************
instance of CIM_Product as $prod001
{
Vendor = " First Always Co.";
Name = "CustomerFirst";
ProductNumber = "PROD-001";
IdentifyingNumber = " TYR-99-003";
Caption = " CustomerFirst Support Product";
Description = "The product for everyone ";
Version = "01.01.00";
};


// ***********************************************************
```

```
// The product in decomposed into three software features.
// ***********************************************************
//
// Software Feature 001 : Incident Manager
//
instance of CIM_SoftwareFeature as $feat001
{
Vendor = "First Always Co." ;
ProductName = "CustomerFirst" ;
ProductNumber = "PROD-001" ;
IdentificationNumber = "TYR-99-003" ;
Version = "01.01.00" ;
Name = "Incident Management";
Caption = " Your Favorite Incident Manager";
Description = "The best." ;
};

// ***********************************************

instance of CIM_ProductSoftwareFeatures // Link Incident Manager with Product
{
Product = $prod001 ;
Component = $feat001 ;
};

//
// Software Feature 002 : Problem Tracker
//

instance of CIM_SoftwareFeature as $feat002
{
Vendor = "First Always Co." ;
ProductName = "CustomerFirst" ;
ProductNumber = "PROD-001" ;
IdentificationNumber = "TYR-99-003" ;
Version = "01.01.00" ;
Name = "Problem Tracker";
Caption = " Your Favorite Problem Manager";
Description = "The best." ;
};
// ***********************************************

instance of CIM_ProductSoftwareFeatures // Link Problem Tracking with Product
{
Product = $prod001 ;
Component = $feat002 ;
};

//
// Software Feature 003 : Advance Graphics Package
//

instance of CIM_SoftwareFeature as $feat003
{
Vendor = "First Always Co." ;
ProductName = "CustomerFirst" ;
```

```
ProductNumber = "PROD-001" ;
IdentificationNumber = "TYR-99-003" ;
Version = "01.01.00" ;
Name = "Advance Graphics Package";
Caption = " Cool Graphics Package";
Description = "The best." ;
};
// ***********************************************
instance of CIM_ProductSoftwareFeatures // Link Advance Graphics with Product
{
Product = $prod001 ;
Component = $feat003;
};


// ***********************************************
// Each Software Feature is decomposed into a set of software elements.
// ***********************************************
//
// The software elements for the Incident Management Features are:
//
// Software Element 1 (Depl/NT) for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_01
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 0 ; // Deployable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
GroupComponent = $feat001 ;
PartComponent = $elmt001_01 ;
};

//
// Software Element 2 (Inst/NT) for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_02
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 1 ; // Installable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
```

```
};

instance of CIM_SoftwareFeatureSoftwareElements
{
GroupComponent = $feat001 ;
PartComponent = $elmt001_02 ;
};

//
// Software Element 3 (Exec/NT) for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_03
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 2 ; // Executable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
GroupComponent = $feat001 ;
PartComponent = $elmt001_03 ;
};
//
// Software Element 4 (Run/NT) for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_04
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 3 ; // Running
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
GroupComponent = $feat001 ;
PartComponent = $elmt001_04 ;
};

// Software Element 5 (Depl/WIN95) for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_05
{
Name = "Incident Manager Element"; // key
```

```
TargetOperatingSystem = 15; // WIN95
Version = "01.01.01 "; // key
SoftwareElementState = 0 ; // Deployable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
GroupComponent = $feat001 ;
PartComponent = $elmt001_05 ;
};

//
// Software Element 6 (Inst/WIN95) for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_06
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 15; // WIN95
Version = "01.01.01 "; // key
SoftwareElementState = 1 ; // Installable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
GroupComponent = $feat001 ;
PartComponent = $elmt001_06 ;
};

//
// Software Element 7 (Exec/WIN95) for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_07
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 15; // WIN95
Version = "01.01.01 "; // key
SoftwareElementState = 2 ; // Executable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
```

```
GroupComponent = $feat001 ;
PartComponent = $elmt001_07 ;
};
//
// Software Element 8 (Run/WIN95 for Incident Management Feature
//
instance of CIM_SoftwareElement as $elmt001_08
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 15; // WIN95
Version = "01.01.01 "; // key
SoftwareElementState = 3 ; // Running
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
Manufacturer = "First Always, Co. ";
SerialNumber = " ";
LanguageEdition = "en";
};

instance of CIM_SoftwareFeatureSoftwareElements
{
GroupComponent = $feat001 ;
PartComponent = $elmt001_08 ;
};

// ***********************************************
//
// Each of the software elements have a series of checks defined for them.
// These are organized into the in-state checks and the next-state checkes.
//
//
// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | In-State
// Element | Deployable |
//

// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | Next-State
// Element | Deployable |
//

// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | In-State
// Element | Installable |
```

```
//


// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | Next-State
// Element | Installable |
//
instance of CIM_OSVersionCheck as $chck001_03_N01
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 1 ; // Installable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
CheckID = "OS VersionCheck";
Description = " ";
Caption " ";
CheckMode = True;
MinimumVersion = "3.5.1";
MaximumVersion = "3.5.1";
};
Instance of CIM_SoftwareElementChecks
{
Element = $elmt001_03 ;
Check = $chck001_03_N01 ;
Phase = 1 ; / next-state
};

// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | In-State
// Element | Executable |
//
instance of CIM_OSVersionCheck as $chck001_03_I01
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 2 ; // Executable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
CheckID = "OS VersionCheck";
Description = " ";
Caption = " ";
CheckMode = True;
MinimumVersion = "3.5.1";
MaximumVersion = "3.5.1";
```

```
};
Instance of CIM_SoftwareElementChecks
{
Element = $elmt001_01 ;
Check = $chck001_03_I01 ;
Phase = 0 ; // in-state
};
instance of CIM_DiskSpaceCheck as $chck001_03_I02
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 2 ; // Executable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
CheckID = "OS VersionCheck";
Description = " ";
Caption = " ";
CheckMode = True;
AvailableDiskSpace = 15000; // kilobytes
};
Instance of CIM_SoftwareElementChecks
{
Element = $elmt001_01 ;
Check = $chck001_03_I02 ;
Phase = 0 ; // in-state
};

instance of CIM_MemoryCheck as $chck001_03_I03
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 2 ; // Executable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
CheckID = "OS VersionCheck";
Description = " ";
Caption = " ";
CheckMode = True;
MemorySize = 32000; // kilobytes
};
Instance of CIM_SoftwareElementChecks
{
Element = $elmt001_01 ;
Check = $chck001_03_I03 ;
Phase = 0 ; // in-state
};

instance of CIM_SoftwareElementVersionCheck as $chck001_03_I04
{
Name = "Oracle"; // key
TargetOperatingSystem = 17; // NT
Version = "07.01.01 "; // key
SoftwareElementState = 2 ; // Executable
SoftwareElementID = " Definition"; // key
```

```
CheckID = "RelationalDatabaseCheck";
Description = " ";
Caption = " ";
CheckMode = True;
SoftwareElementName = "Problem Tracker" ;
LowerSoftwareElementVersin = "01.01.00" ;
UpperSoftwareElementVersion = "01.01.00" ;
SoftwareElementState = 2 ;
TargetOperatingSystem = 17 ;
};
Instance of CIM_SoftwareElementChecks
{
Element = $elmt001_01 ;
Check = $chck001_03_I04 ;
Phase = 0 ; // in-state
};

// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | Next-State
// Element | Executable |
//

instance of CIM_MemoryCheck as $chck001_01_03
{
Name = "Incident Manager Element"; // key
TargetOperatingSystem = 17; // NT
Version = "01.01.01 "; // key
SoftwareElementState = 2 ; // Executable
SoftwareElementID = " Definition"; // key
Version = "01.01.01 "; // key
CheckID = "MemoryCheck";
Description = " ";
Caption " ";
CheckMode = True;
MemorySize = 32000; // units ("KiloBytes"),


};
instance of CIM_SoftwareElementChecks
{
Element = $elmt001_01 ;
Check = $chck001_01_03 ;
Phase = 1 ; / next-state
};


// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | In-State
```

```
// Element | Running |
//

// ***********************************************
//
// Feature Platform/State Phase
// _____________|_____________|_____________
// Incident | NT |
// Manager | | Next-State
// Element | Runing |
//
```

Application Management Object Format (MOF)

```
// ***********************************************
// Application System
// ***********************************************
instance of CIM_ApplicationSystem
{
CreationClassName = "CIM_ApplicationSystem";
Name = "WeServiceIt Customer Support";
};

// ***********************************************
// Application System Software Feature Associations
// ***********************************************

instance of CIM_ApplicationSystemSoftwareFeatures
{
GroupComponent = "
CIM_ApplicationSystem.CreationClassName = \"CIM_ApplicationSystem\" ,
CIM_ApplicationSystem.Name = \"WeServiceIt Customer Support\"
";
PartComponent = "
CIM_Product.Vendor = \"First Always Co.\" ,
CIM_Product.ProductName = \"CustomerFirst\" ,
CIM_Product.ProductNumber = \"PROD-001\" ,
CIM_Product.IdentificationNumber = \"TYR-99-003\" ,
CIM_Product.Version = \"01.01.00\" ,
CIM_SoftwareFeature.Name = \"Incident Management\"
";
};
instance of CIM_ApplicationSystemSoftwareFeatures
{
GroupComponent = "
CIM_ApplicationSystem.CreationClassName = \"CIM_ApplicationSystem\" ,
CIM_ApplicationSystem.Name = \"WeServiceIt Customer Support\"
";
PartComponent = "
CIM_Product.Vendor = \"First Always Co.\" ,
CIM_Product.ProductName = \"CustomerFirst\" ,
CIM_Product.ProductNumber = \"PROD-001\" ,
CIM_Product.IdentificationNumber = \"TYR-99-003\" ,
CIM_Product.Version = \"01.01.00\" ,
CIM_SoftwareFeature.Name = \"Problem Management\"
";
};

instance of CIM_ApplicationSystemSoftwareFeatures
{
GroupComponent = "
CIM_ApplicationSystem.CreationClassName = \"CIM_ApplicationSystem\" ,
CIM_ApplicationSystem.Name = \"WeServiceIt Customer Support\"
";
PartComponent = "
CIM_Product.Vendor = \"First Always Co.\" ,
```

```
CIM_Product.ProductName = \"CustomerFirst\" ,
CIM_Product.ProductNumber = \"PROD-001\" ,
CIM_Product.IdentificationNumber = \"TYR-99-003\" ,
CIM_Product.Version = \"01.01.00\" ,
CIM_SoftwareFeature.Name = \"Database Server\"
";
};

instance of CIM_ApplicationSystemSoftwareFeatures
{
GroupComponent = "
CIM_ApplicationSystem.CreationClassName = \"CIM_ApplicationSystem\" ,
CIM_ApplicationSystem.Name = \"WeServiceIt Customer Support\"
";
PartComponent = "
CIM_Product.Vendor = \"First Always Co.\" ,
CIM_Product.ProductName = \"CustomerFirst\" ,
CIM_Product.ProductNumber = \"PROD-001\" ,
CIM_Product.IdentificationNumber = \"TYR-99-003\" ,
CIM_Product.Version = \"01.01.00\" ,
CIM_SoftwareFeature.Name = \"Advance Graphics Package \"
";
};
```