

#### 4. Find and document as much as possible for the Microsoft Management Console (MMC)

MMC is a Windows-based multiple document interface (MDI) application that heavily uses Internet technologies. Both Microsoft and ISVs extend the console by writing MMC snap-ins, which perform management tasks.

The MMC programmatic interfaces permit the snap-ins to integrate with the console. These interfaces deal only with user interface extensions—how each snap-in actually performs tasks is entirely up to the snap-in. The relationship of the snap-in to the console consists of sharing a common hosting environment, and cross-application integration. The console itself offers no management behavior. Snap-ins always reside in a console; they do not run by themselves.

Both Microsoft and third-party software vendors can develop management tools to run in MMC, as well as write applications to be managed by MMC administrative tools. MMC is part of the Microsoft Platform Software Development Kit (SDK) and is available for general use.

##### Windows Management Services and MMC

Windows management services are provided as a standard part of the operating system. Together, these services act as the management infrastructure, providing a highly scalable foundation on which sophisticated management tools can be built, and a base level of common management functions. On top of these services, Microsoft, other software vendors, and corporate developers can layer any number of management tools, using the underlying functions of the Windows management services. MMC is a core component in this model.

This section briefly discusses the roles of MMC, Active Directory™ directory service, and Group Policy in the management structure. These particular technologies are highlighted because they are the means by which custom MMC consoles are deployed in an Active Directory environment, as explained in the "Active Directory-Based Deployment of MMC" section in this document.

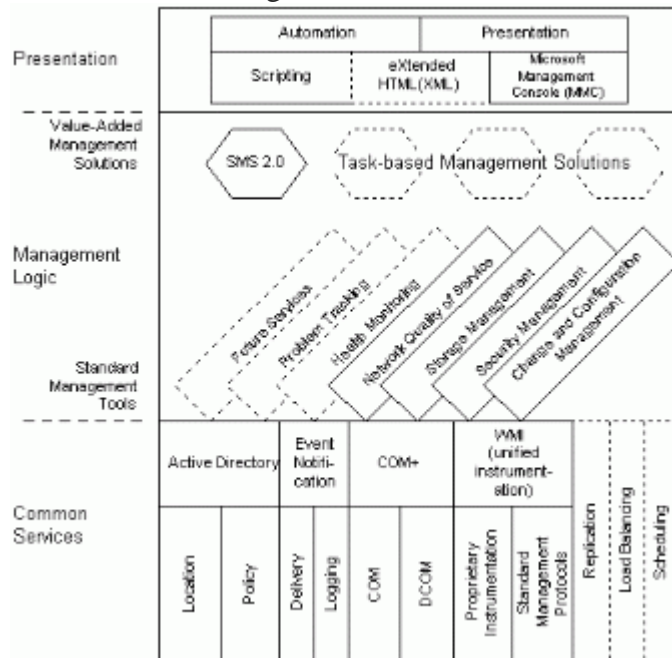
For more detailed information on the Windows management services, see the "Introduction to Windows Management Services" white paper on the Windows 2000 Server Web site or Microsoft TechNet.

The Windows Management services are divided into three logical layers:

- **The Common Services.** These are the low-level operating system services that form the basis of Windows management services. This layer includes base services such as Active Directory, unified instrumentation, and event notification to name but a few.
- **Management Logic.** This middle layer has two major areas. The first area consists of the standard management tools that are necessary for change and configuration management, security management, and problem tracking. The second area is the value-added management solutions, that is, the task-based management solutions that both Microsoft and third-party software vendors will provide. The Group Policy MMC snap-in is one of the standard management tools included in the Windows 2000 operating system.

- **Presentation.** This represents the high-level, common services that allow people to tie the other services together and allow people and processes to interact with the services. MMC is part of this layer.

The Windows management services are illustrated in Figure 1.



**Figure 1: Windows management services**

#### Common Services

The common services represent the necessary building blocks on which Microsoft and others can build value-added solutions. This layer includes the Active Directory, a secure, distributed, partitioned, and replicated directory service that provides two key common management services. Active Directory provides a standardized location service; that is, it provides a standard way to locate resources within the computer systems. Active Directory service also provides the basis for applying Group Policy to the objects managed by the Active Directory.

The Windows 2000 Server operating system includes several Active Directory MMC snap-ins: Active Directory Sites and Services, Active Directory Users and Computers, and Active Directory Domains and Trusts. Additionally, administrators can install the Active Directory Schema snap-in included in the Windows 2000 Administration Tools Setup wizard, which is available in the Windows 2000 Server CD, in the I386 folder (the file name is Adminpak.msi). The Group Policy snap-in, described next, extends both the Active Directory Sites and Services and the Active Directory Users and Computers snap-ins.

In addition to a standardized location service, the Windows 2000 operating system includes Windows Management Instrumentation (WMI), which provides a uniform model through which management data from any source can be accessed and managed in a standard way.

#### Management Logic

The management logic layer provides two distinct classes of service. It provides standard management tools that are built from the common services, and it enables development of high-end, full function, value-added management solutions. Administrators can maximize

users' productivity by using Group Policy—one of the standard Windows 2000 Server management tools—to ensure that people have the necessary data, applications, operating system and settings available, optimally configured for their respective job tasks. Administrators use the Group Policy MMC snap-in to specify options for managed desktop configurations for groups of computers and users. Group Policy provides options for registry-based policy settings, security settings, software installation, scripts, and folder redirection. The Group Policy settings that administrators create are contained in a Group Policy object (GPO) that is in turn associated with selected Active Directory containers: sites, domains, and organizational units (OUs).

Administrators can also set local Group Policy for computers that are not members of a domain. To set local Group Policy, administrators use the Group Policy snap-in focused on the local computer.

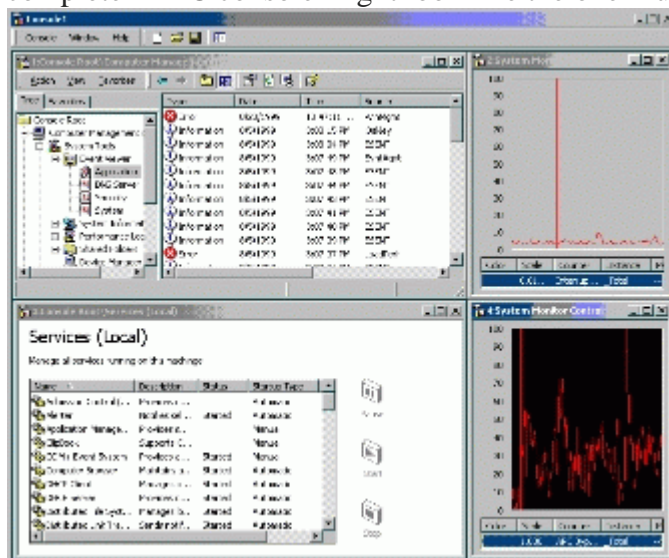
### Common Presentation

Common presentation services such as MMC are also included in the operating system. MMC provides an open, extensible, common hosting environment for management applications (snap-ins). MMC provides a unified user interface for hosting administrative tools, including snap-ins, to administer networks, computers, services, and other system components.

The following section discusses the MMC user interface (UI), and presents descriptions of UI elements and the MMC namespace.

### The MMC USER Interface

At first glance, a MMC UI looks much like an MDI version of Windows Explorer. A complete MMC console might look like the one illustrated in Figure 2.

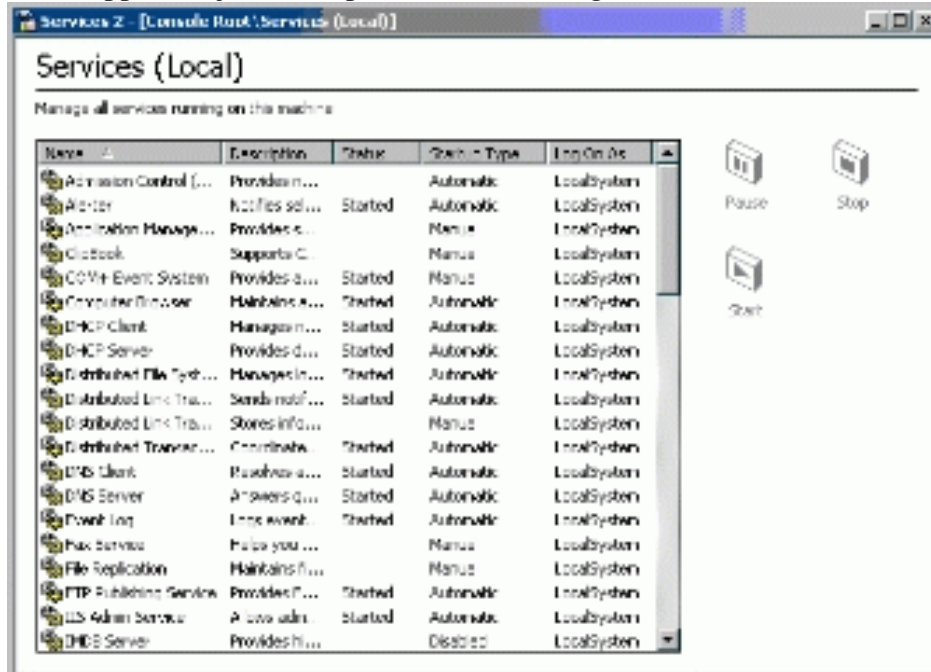


**Figure 2: An MMC console**

The MDI child windows offer many differing views. Each of these child views includes a command bar, a *console tree* (the left pane), and a *details pane* (the right pane)<sup>1</sup> The command bar contains both drop-down menus and buttons. The console tree is a hierarchical structure that displays the items available in a console. This tree-formatted listing shows all visible nodes, each of which represents a manageable container, object, task, or view. The console tree need not be visible in all views—in this example, it is visible in only the top left child window.

Each child window's details pane displays the result of selecting a node in the console tree. In many cases it is a list of the contents of a folder, but in other cases it is a management related view (such as the performance graph in this example), which can be Web- or ActiveX® control-based. The MMC namespace consists of the console tree and the details pane. It is described in more detail in the "MMC Namespace" section in this document.

MMC, as pictured above, can be configured to represent powerful management tools. MMC is also designed to offer a scaled-down view that can be less daunting to less-experienced administrators. For example, a console, such as the one illustrated in Figure 3, can appear as just a taskpad view containing task-oriented information.



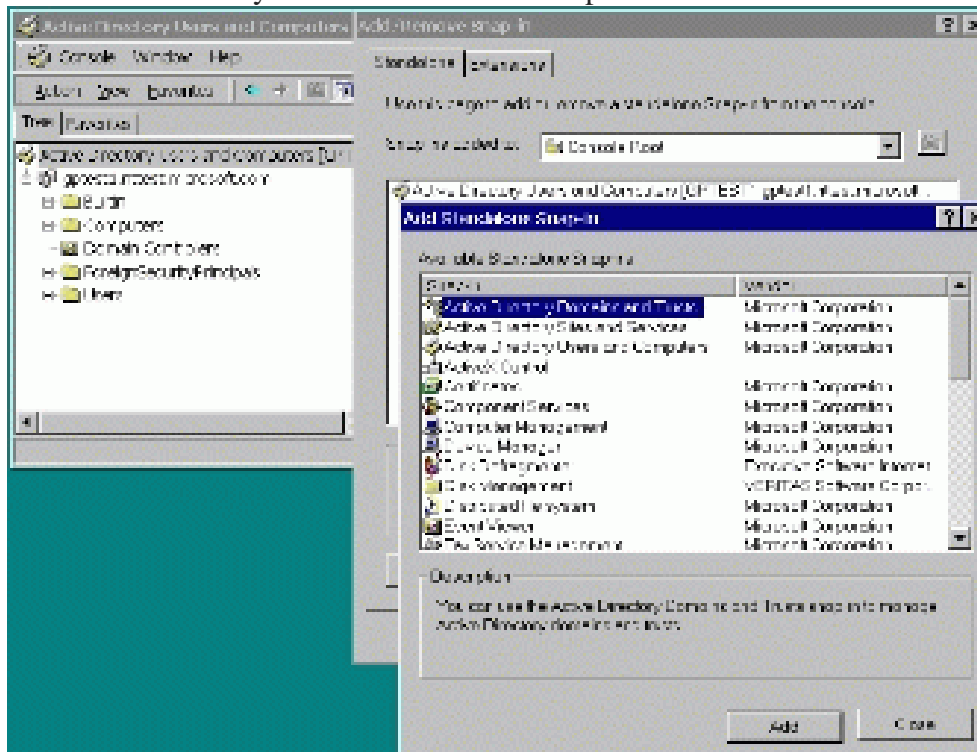
**Figure 3: A console appearing as a taskpad view only**

Another simple view is created by condensing the view to a single tool, like the services management window illustrated in Figure 4.



**Figure 4: A console view condensed to a single view**

Because MMC permits customization, multiple tools can be created and saved; each of the views in the preceding examples can be saved to separate files as different tools. When one of these files is sent to another person, that person can open the file, and the corresponding tool is loaded as configured. For example, a senior administrator could create the view in Figure 4 (a list of services on a computer), and send that view to an operator who will manage only the services on that computer. The operator receives—and can access—only the UI pictured. The subjects of console customization, and tool delegation and deployment are discussed in more detail in the sections in this document on "Creating Custom Consoles: Examples" and "Active Directory-Based Deployment of MMC." Customization is quite simple. An administrator can use the *Snap-in Manager* to dynamically load and unload snap-ins. (The Snap-in Manager provides a list of available snap-ins, allowing users to add or remove snap-ins to the console). To access the Snap-in Manager, administrators click **Add/Remove Snap-in** on the MMC **Console** menu. In the following example (Figure 5), the **Add/Remove Snap-in** dialog box is used to add the Active Directory Domains and Trusts snap-in to the current console.



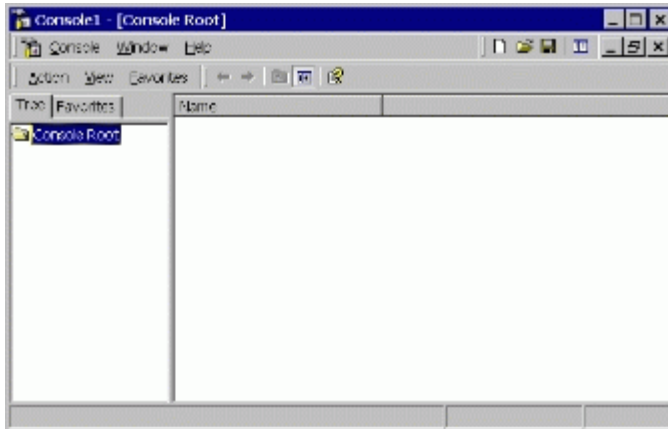
**Figure 5: Adding Active Directory Domains and Trusts snap-in**  
User Interface Elements

This section introduces the main graphical user interface (GUI) elements of MMC. The majority of these are standard Windows 2000-based software GUI elements, tailored for use with MMC.

### Menus

The MMC console has a main menu, an Action band, and a toolbar. The main menus are **Console**, **Window**, and **Help**; these menus provide commands that affect the entire console.

A new console is illustrated in Figure 6.



**Figure 6: Console interface elements**

### Action Band

A band is a rectangular area that contains menus and icons. The Action band contains the following menus:

- **Action.** Includes the same contents as a context menu in Windows (a context menu is accessed by right-clicking an object or container).
- **View.** Controls how information is displayed in the details pane.
- **Favorites.** This tab is displayed when you open a new console in author mode, or when an item has already been added to the Favorites list in a console. The Favorites list can include shortcuts to tools, items in the console, or tasks. When you use MMC in author mode, you gain full access to all MMC functionality. You can add or remove snap-ins, create new windows, create taskpad views and tasks, add items to the Favorites list, and view all parts of the console tree.

### Toolbars

The MMC console contains two standard Windows toolbars:

- **Console toolbar.** Contains the main menu and author mode band (when used in author mode).
- **Snap-in toolbar.** Includes the Action band, common commands bands, and one or more snap-in specific bands.

### Property Sheets

A *property sheet* is a window that users can use to view and edit the properties of an item. A property sheet contains one or more overlapping *property pages*, which are child windows that contain controls for setting a group of related properties. Each page has a tab that the user can click to bring the page to the foreground of the property sheet. For example, the **Computer Management Properties** page contains the controls for setting properties such as a textual description of the computer, setting environment variables, performance options, and startup and recovery options.

### Wizards

A *wizard* is a type of property sheet that is designed to present single property pages in a sequence controlled by the application. To the user, a wizard is a set of windows that present a sequence of steps to complete a particular task. Users can navigate through the sequence by using the **Back** and **Next** buttons located at the bottom of each property page.

Some MMC snap-ins use wizards to automate and simplify tasks for users. For example, the Active Directory Users and Computers snap-in includes the Delegation of Control



wizard that administrators use to delegate control of Active Directory objects. Administrators use this wizard to grant other users permission to manage users, groups, computers, organizational units, and other objects stored in the Active Directory service. MMC snap-ins can add both standard and Wizard 97 pages; however, the preferred method is to use Wizard 97 pages.

### **Dialog Boxes**

MMC uses dialog boxes, which are secondary windows that are mainly used to obtain information from the user to complete a given task. Dialog boxes are usually *modal*, which means the dialog box must be closed before the user can access another window. However, MMC is primarily modeless, which allows users to move between open windows.

The following sections describe the MMC namespace.

### **MMC Namespace**

The MMC namespace represents the hierarchy of objects and containers that are displayed in the console window. The window consists of two panes; the left pane contains the console tree, and the right pane contains the details pane. The left pane also includes the Favorites tab, explained below.

The console tree contains a hierarchy of containers, most of which are represented by folder icons. Some containers are displayed as unique icons that graphically represent the type of items that they contain.

The details pane displays the item selected in the console tree according to a selected view type.

### **Console Tree Pane**

The console tree uses a standard Windows-based tree control to represent a set of containers and objects as an indented outline based on their hierarchical relationship. In this document, the term *container* refers to an item in the console tree that displays child containers below it in the console tree when expanded, and displays its child containers in the details pane (subject to the selected view) when selected.

*Object* refers to an item in the console tree that does not have child items displayed beneath it in the console tree; an object displays information in the details pane when it is selected in the console tree.

### **Containers**

Most of the items in the console tree are containers that hold other containers, objects, or both.

### **Console Root**

The console root is a container that holds the snap-in root nodes.

### **Snap-in Root Node**

The snap-in root node is the uppermost node in the snap-in; it is labeled according to the product or task that it manages. Only one snap-in root node exists for each stand-alone snap-in. MMC supports stand-alone and extension snap-ins. A stand-alone snap-in provides management functionality without requiring support from another snap-in, whereas an extension snap-in requires a parent snap-in above it in the console tree. Extension snap-ins extend the functionality provided by other snap-ins.

### **Favorites List**

When you create a console in author mode, the left pane of the new console includes the **Favorites** tab. This tab is also displayed if an item has been added to the Favorites list in a console.

You can use the Favorites list to:

- **Create shortcuts.** To tools or items in the console tree.
- **Create tasks for novice users.** For example, you can create custom tasks by including shortcuts only to the tasks the user needs to perform, providing a simplified view of a console.
- **Organize taskpad views.** For example, if a console has multiple taskpad views that are distributed in several places in the console tree, you can add these views to the Favorites list, allowing users to access all the views from a single location.

**Note:** MMC version 1.1 does not support user-created taskpads.

### Details Pane

The details pane displays the view of the selected item in the console tree. It has also been referred to as the result pane. The details pane can display information in a variety of formats: a list view, a taskpad view, as ActiveX controls (OCXs), or as an HTML page.

#### List View

The list view displays a collection of items, each consisting of an icon and a label, and provides several ways to display and organize the items. For example, additional information about each item can be displayed in columns to the right of the icon and label. The following view type modes are supported:

- Large icons
- Small icons
- List
- Detail
- Filtered

### Column Customization

MMC supports column configuration, which allows users to customize the configuration of columns in a details list view. The changes the user makes to the column configuration are saved, or persisted, by the console. Not all of the snap-ins support column customization.

By making changes to the column configuration, you can:

- **Customize the display of columns and rows.** For instance, you can rearrange or hide columns. Or you can click the column heading to reorder rows alphabetically or chronologically.
- **Filter columns based on particular attributes.** This applies to some snap-ins only. If you enable this feature, a row of drop-down list boxes that contain options for filtering are displayed beneath the column headings.

**Note:** MMC version 1.1 does not support column configuration.

### Taskpad view

A taskpad view is a dynamic HTML (DHTML) page that presents shortcuts to commands available for a selected item in the scope pane; the taskpad view is displayed in the details pane. Each command is represented as a task that consists of an image, a label, a description, and a mechanism for instructing the snap-in to run that command. Users can run the commands by clicking a task.



You can use taskpad views to:

- Include shortcuts to all the tasks a specific user may need to perform.
- Group tasks by function or user by creating multiple taskpad views in a console.
- Create simplified lists of tasks. For example, you can add tasks to a taskpad view and then hide the console tree. This way, users can begin using tools before knowing the location of particular items in the console tree or operating system, making it easier for novice users to perform their jobs.
- Simplify complex tasks. For example, if a user frequently performs a given task involving several snap-ins and other tools, you can organize shortcuts to those tasks in a single location that run the appropriate property pages, command lines, dialog boxes, or scripts.

### **Custom ActiveX Controls (OCX Controls)**

An ActiveX control is a COM-based object that can draw itself in its own window, respond to events (such as mouse clicks), and can be managed through an interface that includes properties and methods similar to those in Automation objects. You can insert custom ActiveX controls (objects or components) into a Web page or an application to reuse their functionality. Snap-ins can launch an ActiveX control in the details pane.

### **Custom Web Page**

The details pane of a console can host HTML pages that are on the local computer or hosted on a Web server.

MMC is a point of integration between a Web UI and a Win32®-based UI. You can display Web pages within a saved console file by choosing **Add/Remove Snap-in** on the **Console** menu, and clicking **Add** to add a **Link to Web Address**. This permits you to mix and match Web-based administration programs and MMC snap-ins.

### **Export List**

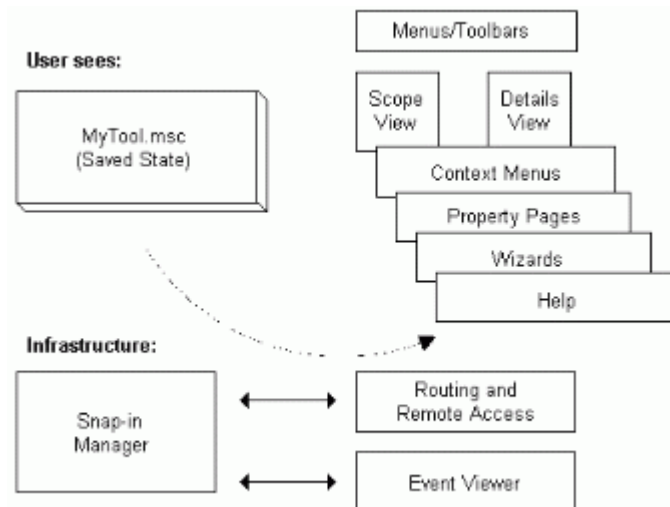
MMC supports exporting the data displayed in all standard list views to a text file. When the user clicks the **Export List** context menu item while a scope pane node with a list view is selected, the visible columns in the list view are exported to a text file in the order in which they appear in the view. You can choose to export only specific rows by selecting them.

**Note:** NNC version 1.1 does not support the export list functionality.

### **How MMC Works**

The MMC console is a Windows-based MDI application that makes use of Internet technologies. The console itself has no management behavior; it is a host that contains other software—snap-ins—that extends the console to offer the actual management capabilities.

The MMC model is illustrated in Figure 7.



**Figure 7: The MMC model**

The UI elements of the tool interact with the *MMC snap-in Manager*, which interacts with the various snap-ins. The snap-in Manager is accessed by clicking **Add/Remove Snap-in** on the MMC **Console** menu. The snap-in Manager also deals with saving settings into a document (Management Saved Console or .msc file). The items at the top of the picture, the .msc file and the UI elements, are all that a user interacts with. The items at the bottom (the Snap-in Manager, the Routing and Remote Access, and Event Viewer snap-ins) are the elements that the developers interact with.

When a MMC tool is loaded, one or more snap-ins are initialized. These snap-ins are integrated to create the tool's namespace—the hierarchy of objects and containers that are displayed in the console tree, and the details view, which displays the view of a selected item in the console tree. The namespace is a master tree that represents what the tool can do. It appears similar to a tree view of the files and folders on a hard disk. The namespace can include all manageable aspects of a network—computers, users and groups, and so on. The details pane can display information as a list view, taskpad view, ActiveX control, or an HTML page.

The child windows in MMC are views into this master namespace. This is akin to having multiple instances of Windows Explorer looking at the same hard disk. Each view may be rooted at a different portion of the tree but they all point to the same master data source (as the examples in Figures 8, 9, and 10 illustrate). If data is currently displayed in multiple child windows, when that data is deleted in one view it will also disappear from the other views.

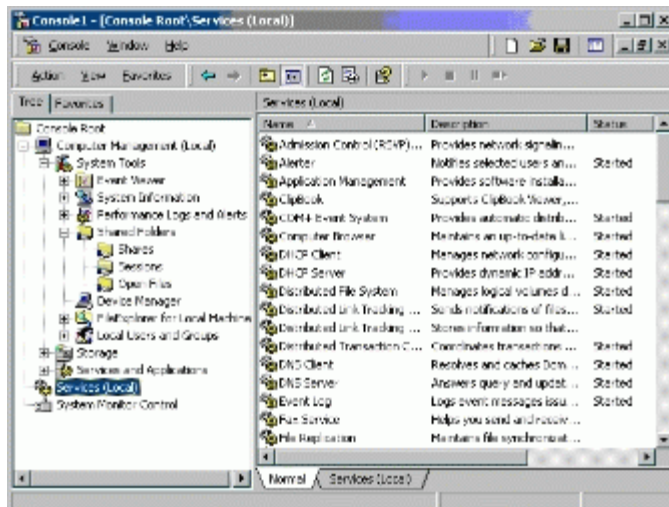


Figure 8: View of the Services node

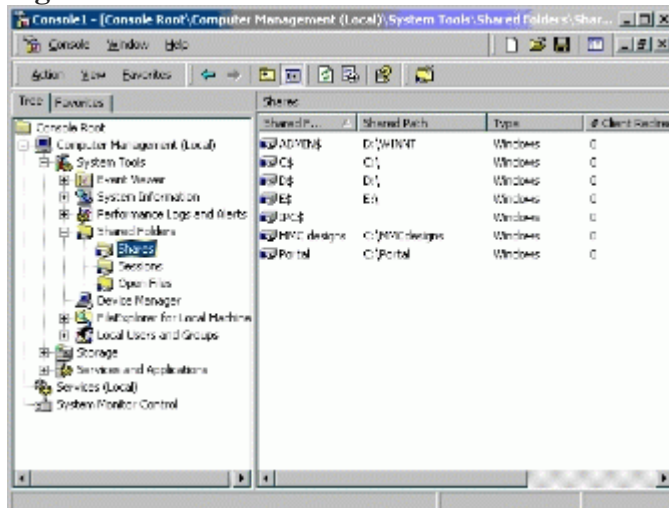


Figure 9: View of the Shares node under Shared Folders

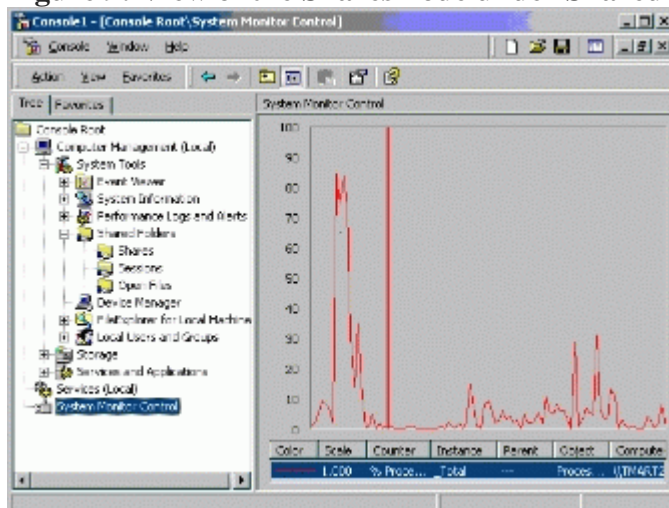


Figure 10: View of the System Monitor Control node Snap-ins

Each MMC tool is built of a collection of instances of smaller tools called MMC *snap-ins*. One snap-in represents one unit of management behavior. A snap-in is the smallest unit of console extension. Technically, a snap-in is an OLE In Process (InProc) server that runs in the process context of MMC. (An In Process server is a server implemented as a dynamic link library (DLL) that runs in the same process as the client.)

The snap-in may call on other supporting controls and dynamic link libraries (DLLs) to accomplish its task.

Snap-ins extend MMC by adding and enabling management behavior. This behavior may be provided in a number of ways. For example, a snap-in might add elements to the viewable node namespace, or it might simply extend a tool by adding context menu items, toolbars, property pages, wizards, or Help to an existing snap-in.

#### Creating Custom Tools from Snap-ins

MMC provides functionality for creating custom management tools. This allows administrators to create, save, and then delegate management tools tailored for specific tasks.

Administrators can assemble multiple snap-ins, from multiple vendors, into a *tool* (also called a *document*). An administrator can create multiple tools, and load and unload them when needed; these tools are what the administrator actually uses to manage the network. After assembling a tool from various snap-ins, the administrator can save the tool in an .msc file, and then reload the file later to instantly recreate the tool. The .msc file can also be e-mailed to another administrator, who can then load the file and use the resulting tool. (If the second administrator does not have all the necessary snap-ins installed on his or her computer, MMC automatically downloads the required snap-ins when the second administrator loads the .msc file).

Administrators can also distribute saved consoles and snap-ins to other administrators by using the Group Policy snap-in and its Software Installation extension. See the "Active Directory-Based Deployment of MMC" section for more information on this topic.

It is possible to run multiple tools simultaneously on one computer, but each tool requires its own instance of MMC to be running.

Note that with MMC, a single tool (a saved console file) does not necessarily have only a single purpose. It is likely that the one tool that an administrator creates and uses regularly will contain management functionality for all aspects of the network—Active Directory, replication topology, file sharing, and so on. It is called a tool because it runs in one instance of MMC, and can be saved in one .msc file.