



Administering DFM (Advanced)

These topics are intended for system administrators who will perform Device Fault Manager (DFM) administrative functions. The topics include:

- [Ports and Protocols that DFM Uses, page 11-1](#)
- [Security Considerations, page 11-3](#)
- [Device Support, page 11-5](#)
- [System Administration, page 11-5](#)

Ports and Protocols that DFM Uses

Table 11-1 DFM Ports

Protocol	Port Number	Service Name	Direction (of establishment) of Connection
ICMP	—	Ping	Server to Device
UDP	161	Simple Network Management Protocol (SNMP)	Server to Device, Device to Server
UDP	162	SNMP Traps (Standard Port)—Default port number used by DFM for receiving traps	Server to Device, Device to Server
UDP	9000	Used for trap receiving	Client to Server

Table 11-1 DFM Ports (continued)

Protocol	Port Number	Service Name	Direction (of establishment) of Connection
UDP	9002	Used by DFM broker (if port 162 is occupied)	Client to Server
TCP	49	TACACS+ and ACS	Server to ACS
TCP	15000	Used by log server	Server internal
TCP	43445	Used by Fault History database engine (dfmFH)	Server internal
TCP	43446	Used by inventory service database engine (DFMInv)	Server internal
TCP	43447	Used by event processing database engine (dfmEPM)	Server internal
TCP	43500-43520	Used by Common Transport Mechanism for internal application messaging	Server internal

Table 11-2 CiscoWorks Common Services Incoming Ports

Port Number/Type	Usage
42343/tcp	Jrun
57860/tcp	JRun Server Manager ControlServer: Used for Jrun Administration
42344/tcp	ANI HTTP server
43441–43459	Used as database ports: 43441—Used by CiscoWorks Common Services

Security Considerations

These topics address some important DFM security issues:

- [File Ownership and Protection](#), page 11-3
- [Secure Socket Layer \(SSL\)](#), page 11-3
- [SNMPv3](#), page 11-4
- [Working with Firewalls](#), page 11-4

File Ownership and Protection

Security for DFM files is based on the same standards used for CiscoWorks.



Caution

Do not change the protection of any file or directory to be less restrictive. You may, if you wish, make the protections more restrictive.

All DFM files are installed with owner CASUSER. Only CASUSER can create, delete, or modify the files installed in *NMSROOT*. *NMSROOT* is the directory where CiscoWorks is installed on your system. If you selected the default directory during installation, on Windows it is C:\Program Files\CSCOpX. On Solaris, it is /opt/CSCOpX.



Note

File protections are not enforced on FAT partitions.

Secure Socket Layer (SSL)

SSL is an application-level protocol that enables secure transactions of data through privacy, authentication, and data integrity. It relies upon certificates, public keys, and private keys. You can enable or disable SSL depending on the need to use secure access.

DFM supports SSL between clients and the server. By default, DFM is not SSL-enabled. For information on enabling SSL, refer to the Common Services online help.

SNMPv3

Like CiscoWorks Common Services, DFM supports SNMPv3 (authentication and access control but no data encryption) between server and devices to eliminate leakage of confidential info. This provides packet-level security, integrity protection, and replay protection, but does not encrypt the packets.

Working with Firewalls

DFM will work across firewalls, but you must perform the following two tasks:

- Configure the DFM server to use a specific port (outgoing connection)
- Configure the firewall to use an automatic established connection (incoming connection)

Step 1 Configure the DfmServer process so it binds to a privileged port, using the `pdcmd --port` option (see [Table 11-6 on page 11-25](#) for more `pdreg` options):

- a. Check the flags that are currently set for the DfmServer process, and write them down (you will need to reset them later):


```
# NMSROOT/bin/pdreg -l DfmServer
```
- b. Unregister the DfmServer process:


```
# NMSROOT/bin/pdcmd -u DfmServer
```
- c. Re-register the DfmServer process with all the flags found in Step a and the following `sm_server` flags, as needed:

<code>--port=port</code>	Specifies port (for example, on a firewall) on which DfmServer will run
<code>--privopen=protocol:port</code>	Specifies privileged port to which DfmServer has access (for example, UDP:162)

```
# NMSROOT/bin/pdcmnd -r DfmServer -e NMSROOT/objects/smarts/bin/sm_server --output -n DFM  
-c icf --privopen=UDP:162 --bootstrap=DFM_bootstrap.conf --subscribe=default"
```

Use the following command to list all sm_server flags:

```
NMSROOT/objects/smarts/bin/sm_server --help
```

Step 2 Configure the *established connection* keyword in the firewall to be automatic.

For additional information on using the privopen option, see [Example: Configuring the DFM Server to Use a Privileged Port, page 11-28](#).

Device Support

When support for new devices becomes available for DFM, Incremental Device Updates (IDUs) will be announced on the planner page for DFM on Cisco.com. Visit the planner page for announcements, downloads, and installation instructions for IDUs as they become available.

When a new IDU becomes available, you can download it from Cisco.com by selecting **Products and Solutions > Network Management > CiscoWorks LAN Management Solution > CiscoWorks Device Fault Manager > Software Center**. (You will be prompted to log into Cisco.com.)

System Administration

DFM system administration can be performed only by the following types of users:

- Users in a System Administrator role. These users can perform system administration tasks that can be started from the CiscoWorks desktop. These tasks include:
 - Configuring users
 - Backing up and restoring data
 - Configuring logging
 - Starting and stopping CiscoWorks processes

- Users who log in as local administrator to the system where DFM is installed. These users can view log files.

If the DFM server is using CiscoSecure Access Control Server (ACS) mode, these CiscoWorks roles are mapped to ACS roles.

Registering Additional DFM Servers with the CiscoWorks Home Page

You can register additional DFM servers so that they appear on the CiscoWorks home page. There is no limit to the number of servers you can register, since device limits are enforced from the DFM server side; the CiscoWorks home page is simply a portal for the different applications. However, you will probably want to limit your home page to two or three DFM servers. The local DFM server name is always listed first on the CiscoWorks home page.

If you have multiple instances of DFM on your home page, you can always map a DFM instance to its Common Services instance by the server hostname (DFM@server, CS@server).



Note

When you use a remote version of DFM, CiscoWorks will prompt you to reauthenticate yourself.

-
- Step 1** From the Common Services home page, select **Home Page > Application Registration**. The Application Registration Status page appears.
 - Step 2** Click **Registration**. The Registration Location page opens.
 - Step 3** Activate the Import from Other Servers radio button, and click **Next**. The Import Server's Attributes page opens.
 - Step 4** In the Import Server's Attributes page, enter the following information:
 - Server Name—Host name or IP address.
 - Server Display Name—A user-specified name that will be displayed on the CiscoWorks home page, and as the DFM home page title when you select that DFM instance.
 - Port—1741.

- Step 5** Click **Next**. CiscoWorks verifies that the remote server is reachable.
- Step 6** When you select the new DFM server instance from the CiscoWorks home page, you will have to authenticate by entering a user name and password for the remote host.
-

Configuring Users (ACS and Non-ACS)

The CiscoWorks server provides the mechanism for authenticating and authorizing users for CiscoWorks applications. What users can see and do is determined by their user role. System Administrators can configure user roles by selecting **Server > Security > User Management**. From here you can add, modify, or delete users.

The CiscoWorks server provides two different mechanisms or *modes* for authenticating users for CiscoWorks applications:

- CiscoWorks Local Mode—By default, the CiscoWorks server uses CiscoWorks Local mode, or *non-ACS mode*. In CiscoWorks Local mode, CiscoWorks assigns roles, along with privileges associated with those roles, as described in the Common Services Permission Report. (You can generate a Permission Report from the Common Services home page by selecting **Server > Reports > Permission Report** and clicking **Help**.) For more information, refer to [Configuring Users Using CiscoWorks Local Mode, page 11-8](#).
- CiscoSecure Access Control Server (ACS) Mode—ACS specifies the privileges associated with roles; however, ACS also allows you to perform device-based filtering, so that users only see authorized devices. Using ACS, which is called *ACS mode*, is supported when ACS is installed on your network and DFM is registered with ACS. For more information, refer to [Configuring Users Using ACS Mode, page 11-8](#).

If Common Services is using ACS mode, DFM must also use ACS mode; otherwise, DFM users will not have any permissions. However, if another instance of DFM is already integrated with ACS, the new DFM will also be integrated with ACS.

Configuring Users Using CiscoWorks Local Mode

To add a user and specify their user role using CiscoWorks Local Mode, select **Server > Security > User Management** from the Common Services home page. Click the Help button for information on the configuration steps.

Use the CiscoWorks Permission Report to understand how each user role relates to tasks in DFM. From the Common Services home page, select **Server > Reports > Permission Report** and scroll down until you find Device Fault Manager.

Configuring Users Using ACS Mode

To use this mode for DFM, Cisco Secure ACS must be installed on your network, and DFM must be registered with ACS.

-
- Step 1** Verify which mode the CiscoWorks server is using. From the Common Services home page, select **Server > Security** and check what is listed in the Current Settings table. Either CiscoWorks Local or TACACS (ACS) will be displayed.
- Step 2** Verify whether DFM is registered with ACS (if ACS Mode is being used) by checking the ACS server.
- Step 3** To modify ACS roles:
- Refer to the ACS online help (on the ACS server) for information on modifying roles.
 - Refer to the Common Services online help for information on the implications of ACS on the DCR (specifically, role dependencies).



Note If you modify DFM roles using ACS, your changes will be propagated to all other instances of DFM that are using Common Services servers which are registered with the same ACS server.

See the following for other information related to ACS:

- To register applications with ACS, and for information on supported ACS versions, refer to *Installation and Setup Guide for Device Fault Manager*.
- To see which DFM user interfaces are affected by ACS device-based filtering, refer to [Device-Based Filtering, page 11-11](#).
- For information on the implications of ACS custom roles on the DCR, see the online help for Common Services.

Using DFM in ACS Mode

Before performing any tasks that are mentioned here, you must ensure that you have successfully completed configuring Cisco Secure ACS with the CiscoWorks server. If you have installed DFM after configuring the CiscoWorks Login Module to the ACS mode, then DFM users are not granted any permissions. However, the DFM application is registered to Cisco Secure ACS.



Note

The System Identity Setup user that is defined in the CiscoWorks server must be added to the Cisco Secure ACS, and this user must have the Network Administrator privilege.

CiscoWorks login modules allow you to add new users using a source of authentication other than the native CiscoWorks server mechanism (that is, the CiscoWorks Local login module). You can use the Cisco Secure ACS services for this purpose.

The following topics provide information on how to use DFM in the ACS mode:

- [Modifying CiscoWorks Roles and Privileges, page 11-11](#)
- [Device-Based Filtering, page 11-11](#)

By default, the CiscoWorks server authentication scheme has five roles in the ACS mode. They are listed here from least privileged to most privileged:

Help Desk	User with this role has the privileges to access network status information from the persisted data. User does not have the privilege to contact any device or schedule a job that will reach the network. Example: Using the Alerts and Activities display.
Approver	User with this role has the privilege to approve all DFM tasks. User can also perform all the Help Desk tasks. Example: Searching the Fault History database.
Network Operator	User with this role has the privilege to perform all tasks that involve collecting data from the network. User does not have write access on the network. User can also perform all the Approver tasks. Example: Configuring logging parameters.
Network Administrator	User with this role has the privilege to change the network. User can also perform Network Operator tasks. Example: Adding devices to DFM from the DCR.
System Administrator	User with this role has the privilege to perform all CiscoWorks system administration tasks. See the Permission Report on CiscoWorks server (Common Services > Server > Reports > Permission Report). Example: Changing device polling from suspended to resumed (Configuration > Polling and Thresholds > Apply Changes).

Cisco Secure ACS allows you to modify the privileges to these roles. You can also create custom roles and privileges that help you customize Common Services client applications to best suit your business workflow and needs.

To modify the default CiscoWorks privileges, see Cisco Secure ACS online help. (On Cisco Secure ACS, click **Online Documentation > Shared Profile Components > Command Authorization Sets.**)

**Note**

See the Common Services online help for important information on how ACS custom roles affect the DCR.

Modifying CiscoWorks Roles and Privileges

If another instance of DFM is registered with the same Cisco Secure ACS, your instance of DFM will inherit those role settings. Furthermore, any changes you make to DFM roles will be propagated to other instances of DFM through Cisco Secure ACS. If you reinstall DFM, your Cisco Secure ACS settings will automatically be applied upon DFM restart.

-
- Step 1** Select **Shared Profile Components > DFM** and click on the DFM roles that you want to modify.
- Step 2** Select or deselect any of the DFM tasks that suit your business workflow and needs.
- Step 3** Click **Submit**.
-

Device-Based Filtering

You can configure ACS to restrict access to all DFM displays. However, device-based filtering can only be performed on the following DFM displays:

**Note**

ACS does not perform any filtering on VLANs.

- **Alerts and Activities**—All displays.
- **Device Management**—All displays.
- **Notification Services > Notification Groups.** (If ACS filtering is changed, Notification Services will not update running notifications.)

- **Fault History > Alert Filtering > Search by Device**
Fault History > Alert Filtering > Search by Group
Fault History > Event Filtering > Search by Device
Fault History > Event Filtering > Search by Group
- **Configuration > Polling and Thresholds:** Only the Polling Parameters Summary and the Thresholds Parameters Summary pages are filtered.
- **Configuration > Other Configurations > Group Administration**—All displays.



Note If any user starts the rediscovery process, all devices managed by DFM are rediscovered (not just those for which the user has access).

Most of the DFM tasks are device-centric. The devices listed for you while performing the DFM tasks are based on your role and associated privileges, defined in Cisco Secure ACS.



Note Refer to the Common Services online help for important information on how ACS custom roles affect the DCR and device-based filtering.

Creating Self-Signed Security Certificates Yearly

When you install DFM, DFM creates a self-signed security certificate on the server. Users on some client systems must install the certificate; see [Responding to Security Alerts, page 2-12](#). Self-signed security certificates expire one year from the date of creation.

Create a new self-signed security certificate yearly, before the certificate expires. You can also do so after the certificate expires; however, users might not be able to access DFM until you complete this task.

Step 1 From the Common Services home page, select **Server > Security > Self Signed Certificates**. The Create Certificates page appears.

Step 2 Enter the values for the fields described in the following table.

Field	Description	Usage Notes
Country Name	Name of your country	Use two-character country code.
State or Province	Name of your state or province	Use two-character state or province code or complete name of state or province.
Locality	Name of your city or town	Use two-character city or town code or complete name of city or town.
Organization Name	Name of your organization	Use complete name or abbreviation for your organization.
Organization Unit Name	Name of department in your organization	Use complete name or abbreviation for your department.
Host Name	Name of server on which DFM is installed	Use the DNS name of the server. Note Use the proper domain name, which should already be displayed in the Host Name field.
Email Address	Your e-mail address	—

Step 3 Click **Apply**.

Backing Up and Restoring DFM Data

Use the Common Services home page to perform immediate backups or schedule backups of DFM data. Common Services provides a command line script that restores data, including data from previous versions of Common Services and DFM.

- For backing up data, select **Server > Admin > Backup**, click Help, and follow the instructions.
- For restoring data, select **Server > Admin > Backup**, click Help, and click the Help link to the Restoring Data topic.

If you are restoring data from DFM 1.2.x or earlier, you will see a warning message and should follow the instructions in the message.

-
- Step 1** On the DFM 1.2.x or earlier server, run the following command on Solaris. (*NMSROOT* is the folder where DFM is installed on the server. If you selected the default directory during installation, it is C:\Program Files\CSCOpX on Windows and /opt/CSCOpX on Solaris.)

```
NMSROOT/objects/smarts/bin/sm_tpmgr -s DFM --dump-agents >
seedfile.txt
```

Run this command on Windows:

```
NMSROOT\objects\smarts\bin\sm_tpmgr.exe -s DFM --dump-agents >
seedfile.txt
```

- Step 2** Copy seedfile.txt to a temporary location on your upgraded server.
- Step 3** Use the CiscoWorks pdshow command to verify that the daemon manager is running (crmdmgtd on Windows and dmgttd on Solaris).
- Step 4** Import the DFM 1.2.x or earlier information, using this command on Solaris:

```
NMSROOT/bin/dfmimport fn=fullpath/seedfile.txt
```

Run this command on Windows:

```
NMSROOT\bin\dfmimport.exe fn=fullpath\seedfile.txt
```

Database files are stored using the backup directory structure described in [Table 11-3](#).

- Format—`/generation_number/suite[/directory]/filename`
- Example—`/1/dfm/dfmFh.db`

Table 11-3 DFM Backup Directory Structure

Option	Description	Usage Notes
generationNumber	Backup number	For example, 1, 2, and 3, with 3 being the latest database backup.
suite	Application, function, or module	When you perform a backup, data for all suites is backed up. The CiscoWorks Common Services suite is cmf. The DFM application suite is dfm.
directory	What is being stored	Suite applications (if applicable).
filename	Specific file that has been backed up	Files include database (.db), log (.log), version (DbVersion.txt), manifest (.txt), tar (.tar), and data files (datafiles.txt). For DFM, the following files are listed directly under <i>generationNumber/suite</i> : dfmEpm.db dfmInv.db dfmFh.db filebackup.tar The file backup.tar contains the following directories and file: <i>NMSROOT</i> /objects/smarts/conf <i>NMSROOT</i> /objects/smarts/local/repos <i>NMSROOT</i> /objects/smarts/local/logs <i>NMSROOT</i> /objects/smarts/local/conf <i>NMSROOT</i> /setup/dfm.info

Changing the Password for DFM Databases

Before You Begin

The procedure in this topic enables you to change the password for the following DFM databases. All DFM databases must use the same password.

- dfmEpm—Event promulgation
- dfmFh—Fault History
- dfmInv—Inventory

Step 1 At the command prompt on the DFM server, stop the daemon manager by entering the following command:

- On Windows:


```
net stop crmdmgmt
```
- On Solaris:


```
/etc/init.d/dmgt d stop
```

Step 2 Change directory to *NMSROOT*/conf/dfmDb/bin. For example, on Windows:

```
cd Program Files\CSCOpX\conf\dfmDb\bin
```

On Solaris:

```
cd /opt/CSCOpX/conf/dfmDb/bin
```



Note *NMSROOT* is the folder where DFM is installed on the server. If you selected the default directory during installation, it is C:\Program Files\CSCOpX on Windows and /opt/CSCOpX on Solaris.

Step 3 Enter `ChangeDfmDbPasswd.pl`, providing a new password as input. For example:

```
ChangeDfmDbPasswd.pl newpassword
```

Step 4 Restart the daemon manager by entering the following command:

- On Windows:

```
net start crmdmgmt
```

- On Solaris:

```
/etc/init.d/dmgt stop
```

Configuring Logging

DFM writes application log files for all major functional modules. By default, DFM writes only error and fatal messages to these log files; DFM saves the previous three logs as backups. You cannot disable logging. However, you can:

- Collect more data when needed by increasing the logging level
- Return to the default logging level as the norm

This task can be performed by a user logged in to DFM in any of the following roles:

- System Administrator
 - Network Administrator
 - Network Operator
-

Step 1 From the DFM home page, select **Configuration > Logging**. The Logging: Level Configuration page is displayed.



Note You cannot disable logging. DFM will always write error and fatal messages to application log files.

Step 2 For each DFM functional module, the Error check box is always selected; you cannot deselect it.

To set all modules to Error, the default logging level:

- a. Click the **Default** button. A confirmation page is displayed.
- b. Click **OK**.

To change the logging level for individual modules:

- a. For each module that you want to change, select one (or deselect all) of the following logging levels:
 - Warning—Log error messages and warning messages
 - Informational—Log error, warning, and informational messages
 - Debug—Log error, warning, informational, and debug messages



Note Deselecting all check boxes for a module returns it to Error, the default logging level.

- b. Review your changes. To cancel your changes, click the **Cancel** button. Otherwise, click the **Apply** button. Clicking the **Apply** button starts immediately resetting the changed logging levels for the DFM functional modules.
-

Viewing and Maintaining Log Files

Each DFM module writes log files to its own folder within the *NMSROOT*/log/dfmLogs folder. [Table 11-4](#) lists each DFM module, the name of the folder where the log files are stored, the related log files, the maximum log size, and the number of backup logs that are saved.



Note

NMSROOT is the folder where DFM is installed on the server. If you selected the default directory during installation, it is C:\Program Files\CSCOpX. On Solaris it is /opt/CSCOpX.

When a log file reaches its maximum size, the module backs up the file and starts writing to a new log file. The module appends a number to the backup file, until it reaches the maximum allowed backups. In the following example, the oldest file is TISServer.log.2, and TISServer.log is the current log file.

```
02:42 PM      4,481,607      TISServer.log
10:22 AM      5,120,447      TISServer.log.1
03:17 AM      5,120,105      TISServer.log.2
```

By default, DFM writes error messages only to log files. You can change the logging level and thereby affect the amount of information stored in log files. To do so, see [Configuring Logging, page 11-17](#).

Table 11-4 DFM Log Files by Module

Function/Module	Folder in <i>NMSROOT</i> \log\dfmLogs	Log Files	Maximum Size (KB)	No. of Backup Files
Alerts and Activities Display	AAD	AAD.log	1000	3
Inventory Interactor	cfi	Interactor.log	1000	5
Inventory Collector	cfi	InventoryCollector.log	35000	5
Polling and Threshold Adapter	cfi	PollingThresholdAdapter.log	10000	5
Detailed Device View	DDV	DDV.log	1000	2
Daily Purging Schedule	DPS	DPS.log	100	2
Event Processing Adapters	epa	adapterServer.log dfmEvents.log	1000	5
Event Promulgation Module	EPM	EPM.log	15000	5
Fault History	FH	FHCollector.log FHUI.log	1000	2
Logging Services	LogService	DfmLogService.log	500	2
Processes with multiple threads	LogService	MultiProcLogger.log	10000	5

Table 11-4 DFM Log Files by Module (continued)

Function/Module	Folder in <i>NMSROOT</i> \log\dfmLogs	Log Files	Maximum Size (KB)	No. of Backup Files
License (device limit)	license	licenseCheck.log	100	2
Notification Services	NOS	nos.log	5000	2
Polling and Threshold Manager	PTM	PTMClient.log PTMServer.log	1000	5
Polling and Threshold Manager (database)	PTM	PTMDB.log	1000	5
Polling and Threshold Manager (grouping services)	PTM	PTMOGS.log	1000	5
Polling and Threshold Manager (Polling and Threshold Adapter)	PTM	PTMPTA.log	1000	5
Rediscovery Schedule	Rediscovery	Rediscovery.log	100	2
Device and Credentials Repository Adapter	TIS	DCRAAdapter.log	1000	2
Device Management	TIS	DeviceManagement.log	1000	2
Inventory Service	TIS	TISServer.log	1000	2
View Group Management	VGM	vgm.log	1000	3

Starting and Stopping DFM Processes


Note

You cannot stop or unregister a process if any process that depends on it is running. You must first stop or unregister all dependent processes, and then stop or unregister the process.

- Step 1** Log in to DFM as a system administrator.
- Step 2** Select **Server Configuration > Administration > Process Management > Stop Process**. The Stop Process page appears.


Note

If a process is not listed, it has not yet been started.

- Step 3** On the Stop Process page, locate the process you want to stop in the Process list.
- Step 4** Select the process you want to stop and click the **Finish** button.
- Step 5** To restart the process, select **Server Configuration > Administration > Process Management > Start Process**. The Start Process page appears.
- Step 6** On the Start Process page, locate the process you want to start in the Process list.
- Step 7** Select the process you want to start and click the **Finish** button.

[Table 11-5](#) provides a complete list of DFM-related CiscoWorks processes. Logs for most of these processes are provided in [Table 11-4 on page 11-19](#).

Table 11-5 DFM-Related CiscoWorks Processes

Name	Description	Dependency
AdapterServer	Event adapter takes events from backend servers.	None
DataPurge	Data Purge—Starts as scheduled in the GUI and purges the Fault History database.	jrm

Table 11-5 DFM-Related CiscoWorks Processes (continued)

Name	Description	Dependency
DfmBroker	<p>DFM Broker maintains a registry about DFM domain managers, which register the following information with the broker when its initialization is complete:</p> <ul style="list-style-type: none"> • Application name of the domain manager • Hostname where the domain manager is running • TCP port at which the HTTP server is listening <p>When a client needs to connect to the domain manager, it first connects to the broker to determine the hostname and TCP port where that server's HTTP service is listening. It then disconnects from the broker and establishes a connection to the domain manager.</p> <p>The DfmBroker log file is located at <i>NMSROOT/objects/smarts/local/logs/brstart.log</i>.</p>	None
DFMLogServer	Controls DFM logs.	None
DFMMultiProcLogger	Handles processes with multiple threads.	None
DFMOGSServer	DFM Object Grouping Service Server evaluates group membership.	CmfDbEngine, ESS
DfmServer	Infrastructure device domain manager, a program that provides backend services for DFM. Services include SNMP data retrieval and event analysis. The DfmServer log is <i>NMSROOT/objects/smarts/logs/DFM.log</i> .	DfmBroker
DFMCTMStartup	Handles interprocess communication.	None
EPMDBEngine	Event Promulgation Module (EPM) database engine—Repository for the EPM module.	None
EPMDBMonitor	EPM database monitor.	EPMDBEngine
EPMServer	Sends events to notification services.	EPMDBEngine

Table 11-5 DFM-Related CiscoWorks Processes (continued)

Name	Description	Dependency
FHDbEngine	Fault History database engine—Repository for alerts and events.	None
FHDbMonitor	Fault History database monitor.	FHDbEngine
FHPurgeTask	Fault History purge task.	None
FHServer	Fault History server, a program that runs backend services for Fault History.	EPMServer, EPMDbEngine, FHDBEngine, FHDbMonitor
Interactor	Provides inventory and device information to the Detailed Device View (DDV); updates the DDV with events.	InventoryCollector
InventoryCollector	Synchronizes voice device inventory with infrastructure device inventory. Handles all inventory events, such as adding and deleting devices.	ESS, TISServer, DFMOGSServer
INVDbEngine	Inventory database engine—Repository for devices.	None
INVDbMonitor	Inventory database monitor.	INVDbEngine
NOSServer	Notification Server monitors alerts and sends notifications based on subscriptions.	EPMDbEngine, EPMServer, INVDbEngine, DFMOGSServer
PTMServer	Polling and thresholds server.	DFMOGSServer
TISServer	Inventory server.	EssMonitor, INVDbEngine

Registering and Unregistering DFM Processes

You can use `pdcmd` to manually unregister and reregister DFM processes with the CiscoWorks daemon manager. This is useful when you want to do any of the following:

- Specify clients that can connect to DFM.
- Configure adapters to restart automatically whenever the DFM server stops and restarts.
- Configure the DFM server to use a privileged port.

Because these commands are complex, be sure to refer to the examples in these sections:

- [Example: Specifying Clients That Can Connect to DFM, page 11-27](#)
- [Example: Configuring the DFM Server to Use a Privileged Port, page 11-28](#)

Before registering a process, you must unregister the related processes in this order:

1. Unregister any processes that depend on the `DfmServer` process.
2. Unregister the `DfmServer` process.
3. Unregister the `DfmBroker` process.

Use the following syntax when unregistering DFM processes (for Windows, the command is `pdcmd.exe`):

```
NMSROOT/bin/pdcmd -u process
```

When you reregister the process, specify all options in the same command instance. If you enter the `pdcmd` multiple times, only the last instance is used.

Register the processes in the following order:

1. Register the `DfmBroker` process.
2. Register the `DfmServer` process.
3. Register any processes that depend on `DfmServer`.

Use the following syntax to reregister the DFM processes. (Refer to [Table 11-6](#) for information about the options and arguments).

```
NMSROOT/bin/pdcmd -r DfmBroker -e path -f arguments
NMSROOT/bin/pdcmd -r DfmServer -e path -d depends -f arguments
NMSROOT/bin/pdcmd -r dependent_process -d DfmServer
```



Note To view the default settings for a process, enter `NMSROOT/bin/pdreg -l process`.



Note If you specify registration options using `pdcmd`, you must re-run your command whenever the daemon manager restarts.

Table 11-6 Options to `pdcmd`

Option	Description and Arguments
<code>-u process</code>	Unregister <i>process</i> . The processes are listed in Table 11-5 on page 11-21 .
<code>-r process</code>	Register <i>process</i> to CiscoWorks daemon manager and start <i>process</i> whenever the dependent (parent) process starts (as described in the <code>-d depends</code> option). The processes are listed in Table 11-5 on page 11-21 .
<code>-e path</code>	Process binary path. <i>path</i> should be:
	• DFM broker: <code>NMSROOT/objects/smarts/bin/brstart</code>
	• DFM server: <code>NMSROOT/objects/smarts/bin/sm_server</code>
<code>-d depends</code>	Process dependency. For <code>DfmServer</code> , <i>depends</i> should be <code>DfmBroker</code> .

Table 11-6 Options to `pdcmd` (continued)

Option	Description and Arguments		
-f “arguments”	DFM-specific arguments, enclosed in one set of quotes. <i>arguments</i> can be the following:		
	<table border="1"> <tr> <td data-bbox="360 358 623 659">--accept <i>host1,host2...</i></td> <td data-bbox="623 358 1244 659">(Optional.) Comma-separated list of hostnames or IP addresses specifying clients which can connect to the server. (The DFM server does not use reverse lookups to determine names of connecting host. If you specify clients as hostnames, be sure the hostname is in DNS, especially if you are using DHCP. If you want to specify localhost, use the host name or IP address, not <i>localhost</i>; refer to the “Example: Specifying Clients That Can Connect to DFM” section on page 11-27.)</td> </tr> </table>	--accept <i>host1,host2...</i>	(Optional.) Comma-separated list of hostnames or IP addresses specifying clients which can connect to the server. (The DFM server does not use reverse lookups to determine names of connecting host. If you specify clients as hostnames, be sure the hostname is in DNS, especially if you are using DHCP. If you want to specify localhost, use the host name or IP address, not <i>localhost</i> ; refer to the “Example: Specifying Clients That Can Connect to DFM” section on page 11-27.)
	--accept <i>host1,host2...</i>	(Optional.) Comma-separated list of hostnames or IP addresses specifying clients which can connect to the server. (The DFM server does not use reverse lookups to determine names of connecting host. If you specify clients as hostnames, be sure the hostname is in DNS, especially if you are using DHCP. If you want to specify localhost, use the host name or IP address, not <i>localhost</i> ; refer to the “Example: Specifying Clients That Can Connect to DFM” section on page 11-27.)	
	<table border="1"> <tr> <td data-bbox="360 659 623 951">--privopen=<i>open-list</i></td> <td data-bbox="623 659 1244 951">(Optional.) Specify the privileged ports and protocol which DfmBroker or DfmServer may open (see Working with Firewalls, page 11-4, for an example). <i>open-list</i> can be comma-separated list of the following (IP:protocol is always required): TCP:<i>port</i>, UDP:<i>port</i>, IP:<i>protocol</i> The defaults for <i>open-list</i> depend on whether DFM is using a reserved port:</td> </tr> </table>	--privopen= <i>open-list</i>	(Optional.) Specify the privileged ports and protocol which DfmBroker or DfmServer may open (see Working with Firewalls , page 11-4, for an example). <i>open-list</i> can be comma-separated list of the following (IP:protocol is always required): TCP: <i>port</i> , UDP: <i>port</i> , IP: <i>protocol</i> The defaults for <i>open-list</i> depend on whether DFM is using a reserved port:
	--privopen= <i>open-list</i>	(Optional.) Specify the privileged ports and protocol which DfmBroker or DfmServer may open (see Working with Firewalls , page 11-4, for an example). <i>open-list</i> can be comma-separated list of the following (IP:protocol is always required): TCP: <i>port</i> , UDP: <i>port</i> , IP: <i>protocol</i> The defaults for <i>open-list</i> depend on whether DFM is using a reserved port:	
	--privopen=IP:1	Default if reserved port is not being used.	
	--privopen=IP:1, UDP: <i>reserved_port</i>	Default if reserved port is being used (normally 162).	
--ouput= <i>file</i>	(Required.) Name of process output file. For DfmServer, <i>file</i> should be DFM.		
--port= <i>port</i>	(DfmBroker only.) DFM broker port. <i>port</i> should always be 9002.		
--restore= <i>file</i>	(DfmBroker only.) Restore broker state from backup file. <i>file</i> should always be: --restore=NMSROOT/objects/smarts/conf/broker.rps		
-n	Do not restart process when DfmServer is stopped and restarted.		

Example: Specifying Clients That Can Connect to DFM

This example shows how to configure DFM to only accept client connections from the hostnames `lucy` and `ethel`. In this case you must unregister and reregister the DFM broker, server, and notification adapter processes.



Note

To allow connections from processes running on the same host, specify the host's name—do not use “localhost.” This is because connections made using the DFM Broker will appear to come from the DFM Broker's host. Only connections that explicitly specify “localhost” as the target address will appear to come from localhost. Such target addresses may result in configurations that forward incoming connections (such as through software that provides an encrypted tunnel).

Step 1 Unregister the processes.

a. Unregister the DFM notification adapters:

```
# NMSROOT/bin/pdcmnd -u DfmFileNotifier
# NMSROOT/bin/pdcmnd -u DfmMailNotifier
# NMSROOT/bin/pdcmnd -u DfmTrapNotifier
```

b. Unregister the DFM server process:

```
# NMSROOT/bin/pdcmnd -u DfmServer
```

c. Unregister the DFM broker process:

```
# NMSROOT/bin/pdcmnd -u DfmBroker
```

Step 2 Re-register the processes, specifying the clients that can connect to the broker and server:

a. For the DFM broker (the following command is one line):

```
# NMSROOT/bin/pdcmnd -r DfmBroker -e NMSROOT/objects/smarts/bin/brstart -f "--output
--port=9002 --accept=lucy,ethel --restore=NMSROOT/objects/smarts/conf/broker.rps"
```

- b. For the DFM server (the following command is one line):

```
# NMSROOT/bin/pdcmnd -r DfmServer -e NMSROOT/objects/smarts/bin/sm_server -d DfmBroker -f
"--bootstrap=DFM_bootstrap.conf --accept=lucy,ethel --output --name=DFM"
```

**Note**

When specifying other options (such as `--privopen`) for `DfmServer`, use one `pdcmnd` instance. See the [“Example: Configuring the DFM Server to Use a Privileged Port”](#) section on page 11-28.

- c. For DFM notification adapters (the following commands are each one line):

```
# NMSROOT/bin/pdcmnd -r DfmFileNotifier -d DfmServer -e
NMSROOT/objects/smarts/bin/sm_notify -f "--adapter=filelog --output=sm_file_notifier"
```

```
# NMSROOT/bin/pdcmnd -r DfmMailNotifier -d DfmServer -e
NMSROOT/objects/smarts/bin/sm_notify -f "--adapter=mail --output=sm_mail_notifier"
```

```
# NMSROOT/bin/pdcmnd -r DfmTrapNotifier -d DfmServer -e
NMSROOT/objects/smarts/bin/sm_notify -f "--adapter=trap --output=sm_trap_notifier"
```

Example: Configuring the DFM Server to Use a Privileged Port

This example shows how to configure DFM to use a privileged port.

- Step 1** Unregister any processes that depend on the `DfmServer` (such as the notification adapters).

```
# NMSROOT/bin/pdcmnd -u DfmFileNotifier
# NMSROOT/bin/pdcmnd -u DfmMailNotifier
# NMSROOT/bin/pdcmnd -u DfmTrapNotifier
```

- Step 2** Unregister the `DfmServer` process:

```
# NMSROOT/bin/pdcmnd -u DfmServer
```

- Step 3** Re-register the `DfmServer` process to use UDP port 162 and the IP protocol 1:

```
# NMSROOT/bin/pdcmnd -r DfmServer -e NMSROOT/objects/smarts/bin/sm_server -d DfmBroker -f
"--bootstrap=DFM_bootstrap.conf --privopen=IP:1,UDP:162 --output --name=DFM"
```

Step 4 Reregister any processes that depend on DfmServer:

```
# NMSROOT/bin/pdcmnd -r DfmFileNotifier -d DfmServer -e
NMSROOT/objects/smarts/bin/sm_notify -f "--adapter=filelog --output=sm_file_notifier"

# NMSROOT/bin/pdcmnd -r DfmMailNotifier -d DfmServer -e
NMSROOT/objects/smarts/bin/sm_notify -f "--adapter=mail --output=sm_mail_notifier"

# NMSROOT/bin/pdcmnd -r DfmTrapNotifier -d DfmServer -e
NMSROOT/objects/smarts/bin/sm_notify -f "--adapter=trap --output=sm_trap_notifier"
```

If you also want DFM to accept only specific client connections, you must specify the `--accept` option when registering the DfmServer process (you do not have to do this for the adapter processes). The following example registers the DfmServer process to use UDP port 162 and IP protocol 1, *and* specifies that DFM can accept connections from hostnames `lucy` and `ethel`:

```
# NMSROOT/bin/pdcmnd -r DfmServer -e NMSROOT/objects/smarts/bin/sm_server -d DfmBroker -f
"--bootstrap=DFM_bootstrap.conf --accept=lucy,ethel --privopen=IP:1,UDP:162 --output
--name=DFM"
```

