Peregrine

Network Discovery

Data Export Guide

PART NO: PND-5.0-ENG-01013-00249

Peregrine SYSTEMS

The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document.

The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental.

If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by email at support@peregrine.com.

If you have comments or suggestions about this documentation, contact Peregrine Systems, Inc. Technical Publications by email at doc_comments@peregrine.com.

This edition applies to version 5.0 of the licensed program.

Peregrine Systems, Inc.
Worldwide Corporate Campus and Executive Briefing Center
3611 Valley Centre Drive San Diego, CA 92130
Tel 800.638.5231 or 858.481.794-7428
Fax 858.481.1751
www.peregrine.com

Data Export Guide

Contents

1 About Exporting Data

CHAPTER

This book is for people who want to create custom reports of Network Discovery data.

Note that Network Discovery also has reports you can access from within its own interface (see “A Tour of Inventory Mapping” in the *Network Discovery User Guide*).

Topics in this chapter include:

- *What you need to know before you start* on page 6
- *About the Network Discovery database* on page 6
- *What’s in this book* on page 8

What you need to know before you start

There are some skills you should have before attempting to follow the procedures in this book.

For Microsoft Word

To export data into a Microsoft Word document, you should already:

- be comfortable with the environment in which you are working (Windows 95/98/2000/NT4.0)
- have a high level of expertise with Microsoft Word 97 or 2000

For data access applications

To create custom reports with your own data access application, you should already be comfortable with:

- the environment in which you will be working (Windows 95/98/2000/NT/XP or UNIX)
- Microsoft Access 2000
- the Open Database Connectivity Standard (ODBC)
- the reporting application you want to use

About the Network Discovery database

The schema of the database is optimized for reporting performance and human understandability rather than for real-time access and statistic storage. In particular, the tables and columns are clearly named and the schema is fully normalized.

The database includes inventories of:

- devices
- ports
- connections
- events

Peregrine's Express Inventory (WMI) collector gathers information about Windows workstations using Windows Management Instrumentation (WMI). This WMI information can also be added to the Network Discovery database. For information on setting up and using the WMI Collector, see your ServiceCenter Essentials documentation.

Open database access

The Reports database is open. That means the schema is published in standard description languages and you can access and export the data in a number of ways by means of standard protocols.

The schema of the reports database is described using English, SQL and XML DTDs. All descriptions are available from the appliance.

The data can be accessed directly on the appliance (queries only, no updates) through the ODBC protocol. MySQL provides an ODBC driver for Windows. The ODBC driver is available on the appliance.

Using this driver, Windows-based applications can remotely access the Reports database.

Creating your own reports

There are three ways to use Network Discovery data.

1) Network Discovery has its own reports accessible through the Network Discovery interface.

2) You can export Network Discovery data into Comma Separated Values (CSV) or into XML format for use with Microsoft Word documents or spreadsheets (see "*Exporting data into Microsoft Word documents* on page 9").

3) A third option—the option most of this book is about—is to export Network Discovery data into a data access application that operates on the Open Database Connectivity (ODBC) standard. You can use a third-party data access application to customize the presentation of Network Discovery data, if your application operates on the ODBC standard. ODBC applications into which you can export Network Discovery data comprise (but are not limited to):

- Microsoft Access

- Crystal
- Cognos Impromptu
- PowerPlay OLAP tool

Note: The data imported into the data access application is read-only. You cannot manipulate Network Discovery or your network from within the application.

What's in this book

The second chapter tells you how to import data into Microsoft Word documents. The third chapter tells you how to export data and make it available to data access applications. The remainder of the book describes the data that you have available to work with.

Enjoy creating your own reports!

CHAPTER 2

Exporting data into Microsoft Word documents

You can create custom Microsoft Word report documents with any text you want and include any graphs and pie charts that appear in Network Discovery. Because the graphs are actually linked to the Network Discovery appliance, they update automatically.

Topics in this chapter include:

What you need before you start

You must have either Word 97 or Word 2000 on one of the following operating systems:

- Windows 95
- Windows 98
- Windows 2000
- Windows NT 4.0

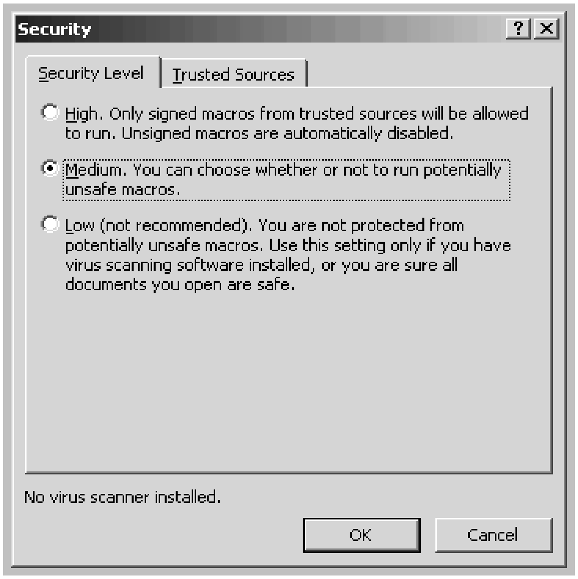
Note: You must have Internet Explorer (IE) on the computer you're working with. Even if you do not use IE with Network Discovery, you must have it loaded onto your management workstation for the Word file to work properly.

You must enable macros in Word.

To enable macros

1. For example, in Word 2000, click **Tools** > **Macro** > **Security**.
2. In the **Security Level** tab, select one of the following security levels to enable macro execution:
   - **Medium**
   - **Low**

Figure 2-1: Word Security Level tab



○ High. Only signed macros from trusted sources will be allowed to run. Unsigned macros are automatically disabled.

● Medium. You can choose whether or not to run potentially unsafe macros.

○ Low (not recommended). You are not protected from potentially unsafe macros. Use this setting only if you have virus scanning software installed, or you are sure all documents you open are safe.

No virus scanner installed.

Two Word files are included with Network Discovery

Microsoft Word can import diagrams from URIs. Since most of the components of Network Discovery are web-based, they have URIs that Microsoft Word can reference. See your Microsoft Word documentation for a full explanation of how to create and manipulate a document with URI references.

To make it easier to prepare your Word report, Network Discovery comes with two documents for Microsoft Word that you can download.

The two files are:

- SumRepExample.doc, a sample Microsoft Word file
- SumRepTemplate.dot, a Microsoft Word template file

The first document is an example report framework for the intermediate Microsoft Word user. To use it, you use the Microsoft Cut and Paste commands to rearrange the built-in graphs. The second document, the template, is for an advanced Microsoft Word user who is comfortable with Word field codes and macro substitution.

Each document contains links to graphs on your Peregrine appliance. Once you customize the report with the name of your Peregrine appliance, you can easily update the graphs—to present at weekly meetings for example.

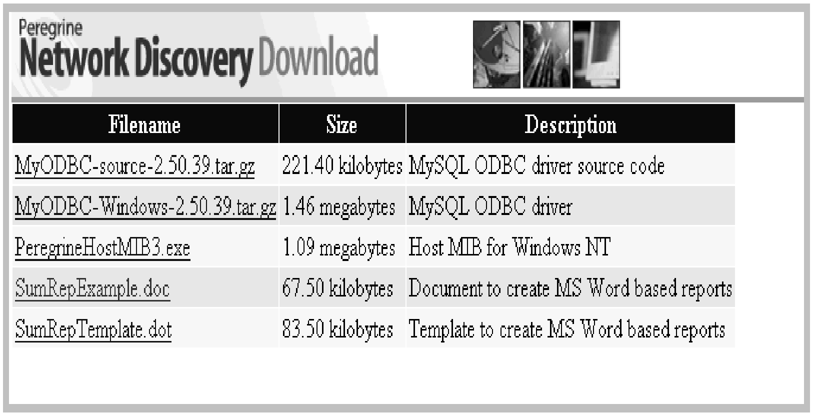
Note: To use the SumRepExample.doc report framework, you must download both files. You need the template file to be able to update the example file.

Download the Word files

To download the example file

1. In Network Discovery, click **Home** > **Download.**
2. Right-click **SumRepExample.doc** > **Save Target As**. Save the example file in any directory that is convenient to you.

Figure 2-2: The Network Discovery Download page



Filename	Size	Description
MyODBC-source-2.50.39.tar.gz	221.40 kilobytes	MySQL ODBC driver source code
MyODBC-Windows-2.50.39.tar.gz	1.46 megabytes	MySQL ODBC driver
PeregrineHostMIB3.exe	1.09 megabytes	Host MIB for Windows NT
SumRepExample.doc	67.50 kilobytes	Document to create MS Word based reports
SumRepTemplate.dot	83.50 kilobytes	Template to create MS Word based reports

Remember you also need the template file to use the example file.

To download the template file

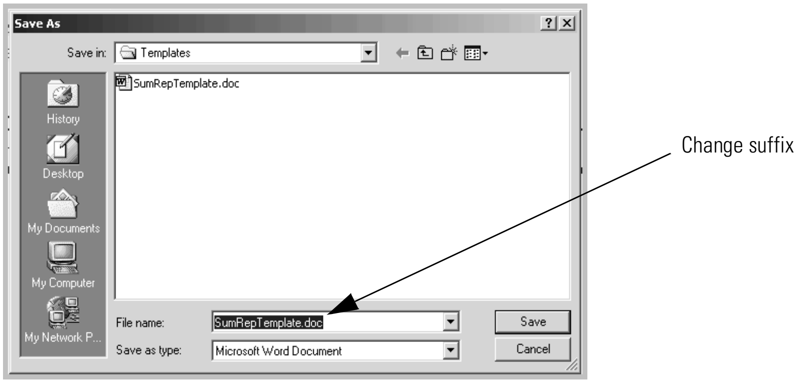
You can use the template file either on its own or with the example file.

1 In Network Discovery, click **Home** >**Download.**

2 Right-click **SumRepTemplate.dot** > **Save Target As.** Save the template file in your local Word template directory. If you don't know where this directory is on your PC, see your Microsoft documentation.

Important: Before you save the file, change the file name suffix from .doc to .dot

Figure 2-3: Change .doc to .dot when you save the template



Save in: Templates

SumRepTemplate.doc

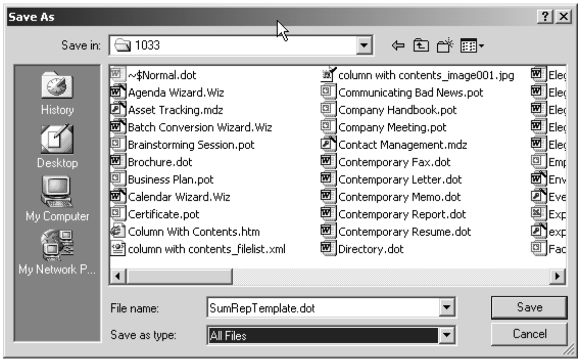
File name: SumRepTemplate.doc

Save as type: Microsoft Word Document

Change suffix

In this example (Figure 2-4 on page 13, below), the path for saving the file is C:\Program Files\Microsoft Office\Templates\1033. This is using Microsoft Word 2000 on Windows NT 4.0.

Figure 2-4: Word 2000 on Windows NT 4.0 path for saving template file



Save in: 1033

~$Normal.dot
Agenda Wizard.Wiz
Asset Tracking.mdz
Batch Conversion Wizard.Wiz
Brainstorming Session.pot
Brochure.dot
Business Plan.pot
Calendar Wizard.Wiz
Certificate.pot
Column With Contents.htm
column with contents_filelist.xml
column with contents_image001.jpg
Communicating Bad News.pot
Company Handbook.pot
Company Meeting.pot
Contact Management.mdz
Contemporary Fax.dot
Contemporary Letter.dot
Contemporary Memo.dot
Contemporary Report.dot
Contemporary Resume.dot
Directory.dot
Ele
Ele
Ele
Ele
Ele
Em
Env
Eve
Exp
exp
Fac

File name: SumRepTemplate.dot

Save as type: All Files

In Word XP, the directory is C:\Documents and Settings\Application Data\Microsoft\Templates.

Your template directory may be in a different location, so refer to your Microsoft documentation.

3 Click **Save**.

The example file is a report you can customize

The example file includes all the graphical reports available on your Peregrine appliance. You can use the example report to create your own reports by deleting what you do not need, and by adding your own descriptive text.

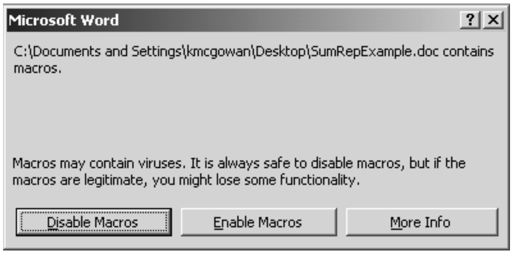
To work with the Example file

1 In Microsoft Word, click **File** > **Open**.

2 Select the file SumRepExample.doc.

A dialog asks if you want to enable macros.

Figure 2-5: Enable macros dialog



Microsoft Word

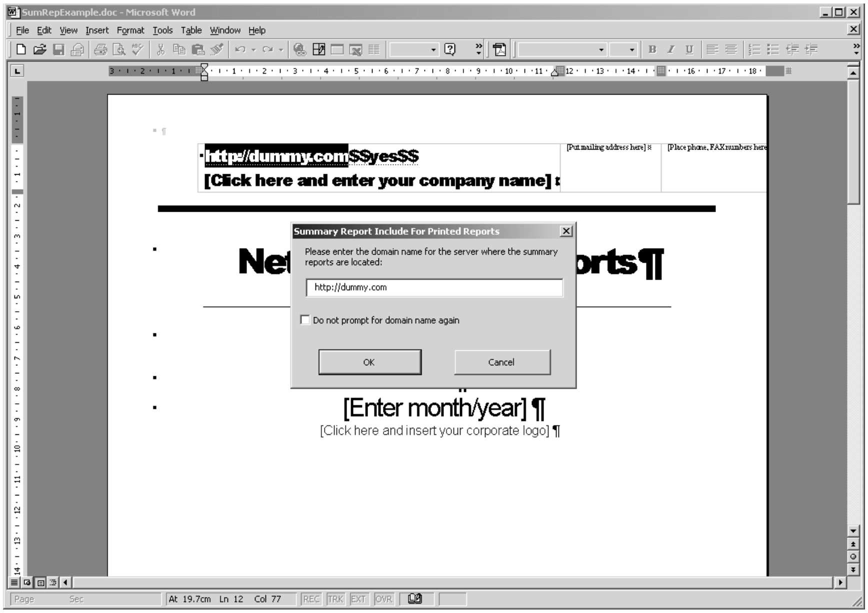
C:\Documents and Settings\kmcgowan\Desktop\SumRepExample.doc contains macros.

Macros may contain viruses. It is always safe to disable macros, but if the macros are legitimate, you might lose some functionality.

3 Click **Enable Macros**.

A dialog box appears, called “Summary Report Include for Printed Reports.” The dialog allows you to connect to your Peregrine appliance to import the Network Discovery graphs.

Figure 2-6: “Summary Report Include for Printed Reports” dialog



SumRepExample.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
At 19.7cm Ln 12 Col 77 REC TRK EXT OVR

http://dummy.com$$yes$$

[Click here and enter your company name]

[Put mailing address here]

[Place phone, FAX numbers here

[Enter month/year] ¶

[Click here and insert your corporate logo] ¶

Summary Report Include For Printed Reports

Please enter the domain name for the server where the summary reports are located:

http://dummy.com

□ Do not prompt for domain name again

OK Cancel

4 Enter the IPv4address or domain name for your Peregrine appliance.

5 Click **OK**.

A dialog asks if you want to continue searching at the beginning.

6 Click **Yes**.

You are prompted for your user name and password for the appliance.

7 Enter your name and password.

Four more dialogs appear, asking if you want to search the document.

8 Click **Yes** to all of the dialogs.

The Word document is now full of graphs from your appliance. Use Word’s Cut and Paste commands to make the example file your own report.

Start a report from scratch with the template file

The template file allows you to create a report from scratch rather than cutting and pasting within a prepared document as you can do with the example file. It is for expert Microsoft Word users. You should be comfortable with Word field codes and macro substitution.

To start a new Word report file

1 Open Word.

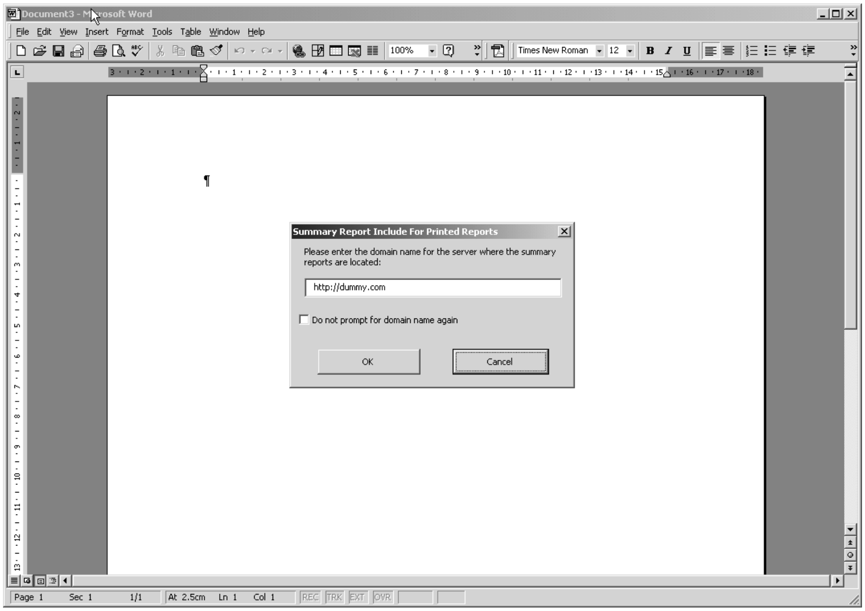
2 Click **File** > **Open**.

3 Select the file **sumreptemplate.dot**.

4 Click **Open**.

A dialog box, called "Summary Report Include for Printed Reports", appears.

Figure 2-7: "Summary Report Include for Printed Reports" dialog



Summary Report Include For Printed Reports

Please enter the domain name for the server where the summary reports are located:

http://dummy.com

☐ Do not prompt for domain name again

Document3 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
100%
Times New Roman 12
OK
Cancel
Page 1 Sec 1 1/1 At 2.5cm Ln 1 Col 1 REC TRK EXT OVR

5 Enter the URI or domain name for your Peregrine appliance.

6 Click **Save**.

You can now create your document from scratch.

You can use shortcuts to include graphs automatically

Instead of typing the exact URI for the graph stored on your appliance, you type a shortcut that translates automatically into the exact URI that INCLUDEPICTURE requires. The shortcut takes this form:

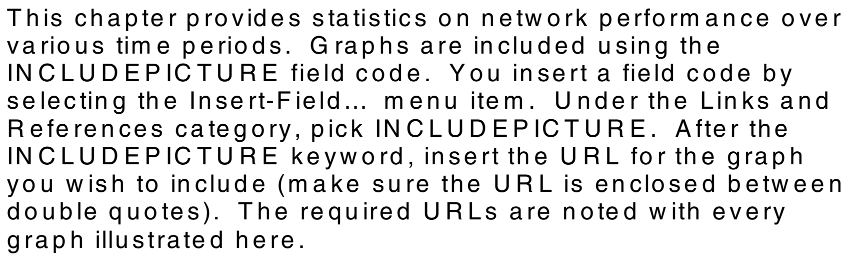
```
?graph(group,attr,stat,period,legend)?
```

Spaces are permitted between each element, but not within elements.

Once you have typed the shortcut into the template, you run a pre-recorded macro to translate the shortcut into a field code that loads a fresh graph right from the Network Discovery appliance.

a The available shortcuts and what graphs they map to are in the original example file.

Figure 2-8: From the Word example report:



This chapter provides statistics on network performance over various time periods. Graphs are included using the INCLUDEPICTURE field code. You insert a field code by selecting the Insert-Field... menu item. Under the Links and References category, pick INCLUDEPICTURE. After the INCLUDEPICTURE keyword, insert the URL for the graph you wish to include (make sure the URL is enclosed between double quotes). The required URLs are noted with every graph illustrated here.

To translate a shortcut;

1 Select the shortcut.

Do one of the following:

a Click **Tools** > **Macro** > **Macros.**

b Press **Alt** + **F8.**

Update the graphs in your report

To update the graphs

1 In your Word report, click **Tools** >**Macro** > **Macros.**

(The shortcut is **Alt** + **F8.**)

2 Select **SumRepUpdate.**

3 Click **Run.**

A dialog asks you to confirm the name of the server you want the data to update from (the domain name of the Peregrine appliance).

4 Click **Yes.**

You are prompted for your user name and password for the appliance.

5 Enter your name and password.

Four more dialogs appear, asking if you want to continue searching at the beginning.

6 Click **Yes** to all of the dialogs.

The graphs in your report have updated.

7 Select the “SumRepUpdate” macro, then click **Run**.

When you open a document created with the template, the graphs update automatically.

3 CHAPTER Exporting Data into Data Access Applications

This chapter contains a tutorial that walks you through a simple example of how to connect to the Network Discovery database from Microsoft Access 2000 by means of ODBC; how to link in the tables and perform two basic queries.

Topics in this chapter include:

Step 1: Set up Network Discovery to export data

To set up Network Discovery to export data:

1 In Network Discovery, click **Administration** >**Appliance Services** >**MySQL Access**.

2 Click **Enabled**.

Step 2: Putting data into CSV format first works better

Putting Network Discovery data into "comma-separated values" (CSV) format first, allows you to export it into a Microsoft Word document or into a spreadsheet more successfully.

To export Reports data into CSV format

1 In Network Discovery click on the [Export] link in a Report list.

A dialog box asks if you want to export the data to a file, or open the file from the Peregrine appliance.

2 Select **Export**.

3 Click **OK**.

- If you select **Open the file from its current location**, the data is displayed in a spreadsheet using the default program designated by your system to display CSV files, such as Microsoft Excel.
- If you select **Save this file to disk**, follow the prompts to save the CSV file for opening later.

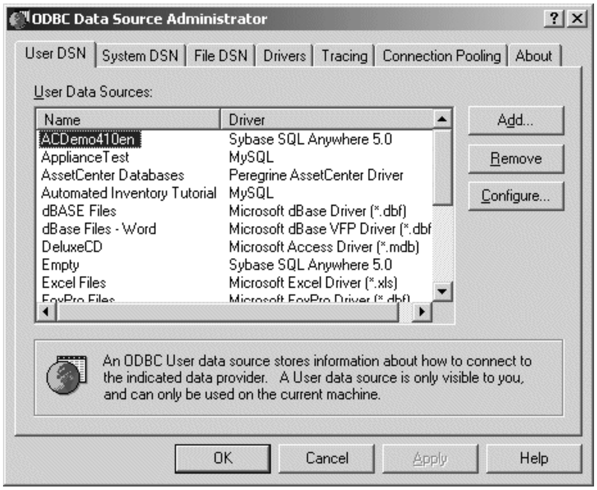
Step 3: Select MYSQL as the data source (create an ODBC alias)

Before you can use the Network Discovery data with Microsoft Access 2000 you need to create an ODBC alias for the database.

1 From the Windows **Control Panel**, select **Data Sources** (ODBC).

Note: In Windows 2000, select **Setting|Control Panel|Administrative Tools|Data Sources** (ODBC)

The **ODBC Data Source Administrator** appears.



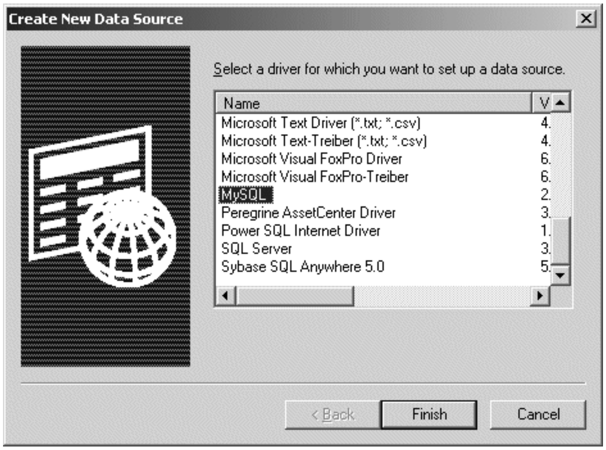
User Data Sources:

Name	Driver
ACDemo410en	Sybase SQL Anywhere 5.0
ApplianceTest	MySQL
AssetCenter Databases	Peregrine AssetCenter Driver
Automated Inventory Tutorial	MySQL
dBASE Files	Microsoft dBase Driver (*.dbf)
dBase Files - Word	Microsoft dBase VFP Driver (*.dbf
DeluxeCD	Microsoft Access Driver (*.mdb)
Empty	Sybase SQL Anywhere 5.0
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files	Microsoft FoxPro Driver (*.dbf)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

2 In the **User DSN** tab, click **Add.**

The **Create New Data Source** dialog appears.



Select a driver for which you want to set up a data source.

Name	V
Microsoft Text Driver (*.txt; *.csv)	4.
Microsoft Text-Treiber (*.txt; *.csv)	4.
Microsoft Visual FoxPro Driver	6.
Microsoft Visual FoxPro-Treiber	6.
MySQL	2.
Peregrine AssetCenter Driver	3.
Power SQL Internet Driver	1.
SQL Server	3.
Sybase SQL Anywhere 5.0	5.

3 In the list box select **MySQL.**

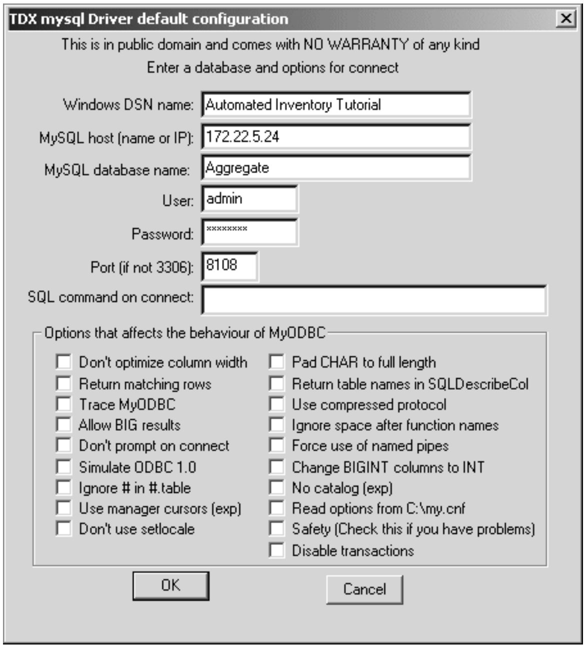
4 If your computer does not already have a MySQL driver, you can do one of two things:

a Go to http://www.mysql.com. Follow the directions on the web site to download the ODBC driver.

b Download the MYSQL driver from the appliance. From **Home Base** click **Download.** On the **Download** page click the **MyODBC-Windows-2.50.39.tar.gz** link.

5 Click **Finish.**

The **TDX MySQL Driver default configuration** dialog appears.



TDX mysql Driver default configuration

This is in public domain and comes with NO WARRANTY of any kind

Enter a database and options for connect

Windows DSN name: Automated Inventory Tutorial

MySQL host (name or IP): 172.22.5.24

MySQL database name: Aggregate

User: admin

Password: ********

Port (if not 3306): 8108

SQL command on connect:

Options that affects the behaviour of MyODBC

- ☐ Don't optimize column width
- ☐ Return matching rows
- ☐ Trace MyODBC
- ☐ Allow BIG results
- ☐ Don't prompt on connect
- ☐ Simulate ODBC 1.0
- ☐ Ignore # in #.table
- ☐ Use manager cursors (exp)
- ☐ Don't use setlocale
- ☐ Pad CHAR to full length
- ☐ Return table names in SQLDescribeCol
- ☐ Use compressed protocol
- ☐ Ignore space after function names
- ☐ Force use of named pipes
- ☐ Change BIGINT columns to INT
- ☐ No catalog (exp)
- ☐ Read options from C:\my.cnf
- ☐ Safety (Check this if you have problems)
- ☐ Disable transactions

6 Enter the following information:

- The Windows DSN name (Data Source Name). In the following example we have called it **Network Discovery Tutorial.**
- The name or IP address of the Peregrine appliance. (Ask your network administrator if you are not sure of the IP address or name of your appliance).
- In the MYSQL database name field, enter the name of the database. Enter **Aggregate**.

7 For the name of the user, enter one of the following:

- "demo" for anyone
- "admin" for anyone who has the administrator password or
- the account name of anyone who has been set up with a user account

8 Enter the password for the above user.

9 For the number of the port, always enter **8101**.

10 Once you have entered these fields, click **OK**.

Now you are returned to the **UserDSN** tab in the **ODBC Data Source Administrator** dialog box.

11 Click **OK** to exit.

You are now ready to connect to the Network Discovery database with applications such as MS Access 2000 by means of ODBC.

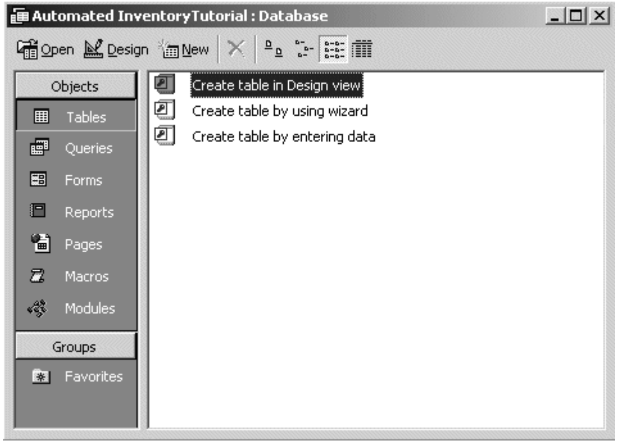
Step 4: Create a new database in Microsoft Access 2000

12 Start Microsoft Access 2000.

13 Create a new blank database. Give it a name and save it.

The following screen is displayed. In this example, the database has been named **Network DiscoveryTutorial.mdb** and saved in the following directory

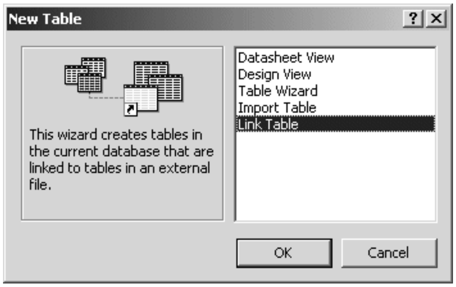
C:\Program Files\Peregrine\Network Discovery\
The following dialog is displayed.


Automated InventoryTutorial : Database
Open
Design
New
Objects
Tables
Queries
Forms
Reports
Pages
Macros
Modules
Groups
Favorites
Create table in Design view
Create table by using wizard
Create table by entering data

Step 5: Link in the Network Discovery tables

14 In the Objects menu, select **Tables** and click **New**.

The following dialog appears.

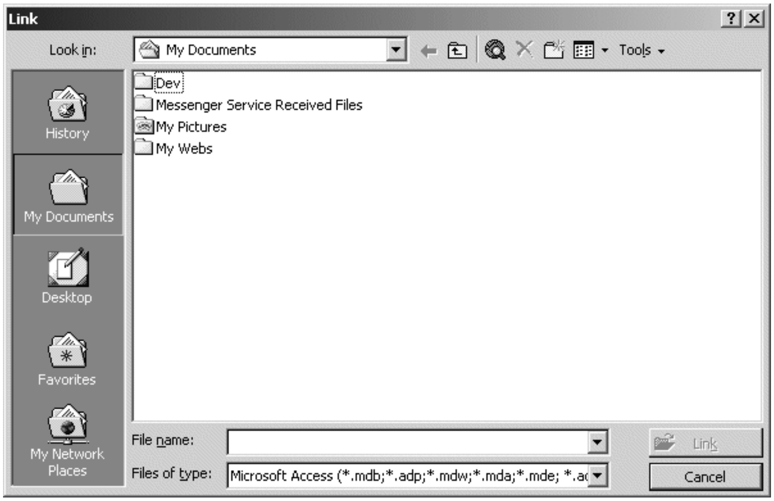


New Table

This wizard creates tables in the current database that are linked to tables in an external file.

15 Select **Link Table** and click **OK.**

The following screen appears.



Link

Look in: My Documents

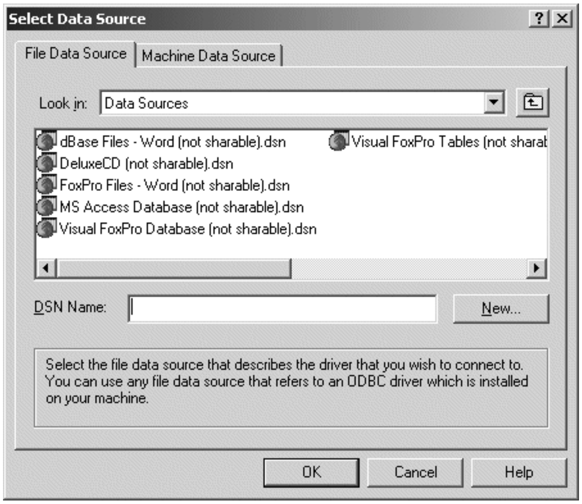
File name:

Files of type: Microsoft Access (*.mdb;*.adp;*.mdw;*.mda;*.mde; *.ac

16 In the **Files of type** pull-down list select **ODBC Databases.**

The following dialog appears.

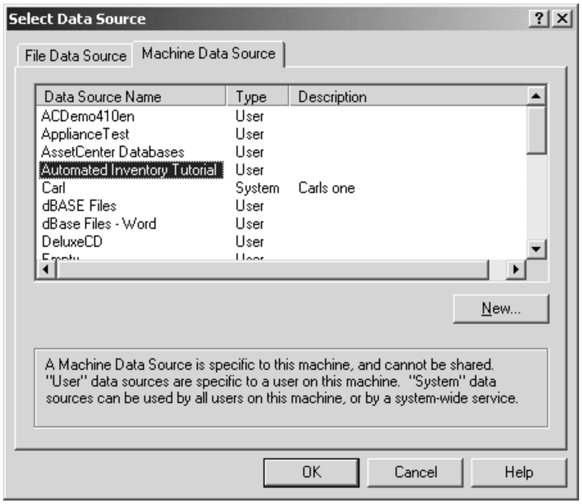
Note: Note, that the **Network DiscoveryTutorial.mdb** file is not supplied with Network Discovery, but is the file that you created in step 13 on page 23.



dBase Files - Word (not sharable).dsn
DeluxeCD (not sharable).dsn
FoxPro Files - Word (not sharable).dsn
MS Access Database (not sharable).dsn
Visual FoxPro Database (not sharable).dsn
Visual FoxPro Tables (not sharat

Select the file data source that describes the driver that you wish to connect to. You can use any file data source that refers to an ODBC driver which is installed on your machine.

17 Click the **Machine Data Source** tab.



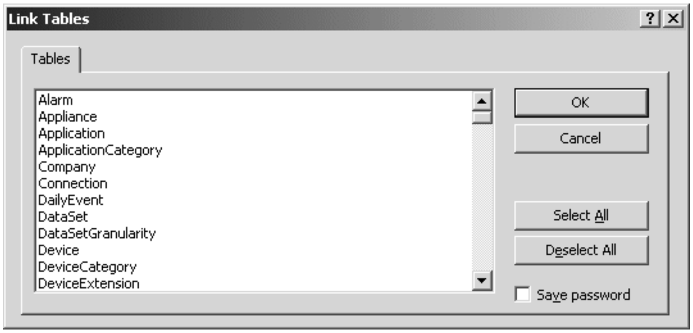
Data Source Name	Type	Description
ACDemo410en	User	
ApplianceTest	User	
AssetCenter Databases	User	
Automated Inventory Tutorial	User	
Carl	System	Carls one
dBASE Files	User	
dBase Files - Word	User	
DeluxeCD	User	
Empty	User	

A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

18 Select your entry (in this case **Network Discovery Tutorial**) and click **OK**. The **Logon to InterBase** dialog appears.

Note: Note, that this is the Tutorial data source name that you created in step 6 on page 22.

The following **Link Tables** dialog is displayed.



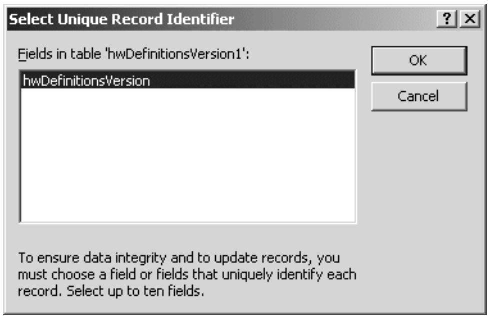
Alarm
Appliance
Application
ApplicationCategory
Company
Connection
DailyEvent
DataSet
DataSetGranularity
Device
DeviceCategory
DeviceExtension

19 Click **Select All.**

All the entries are now highlighted.

20 Click **OK.**

The **Select Unique Record Identifier** dialog for the 'hwDefinitionsVersion1' table appears.



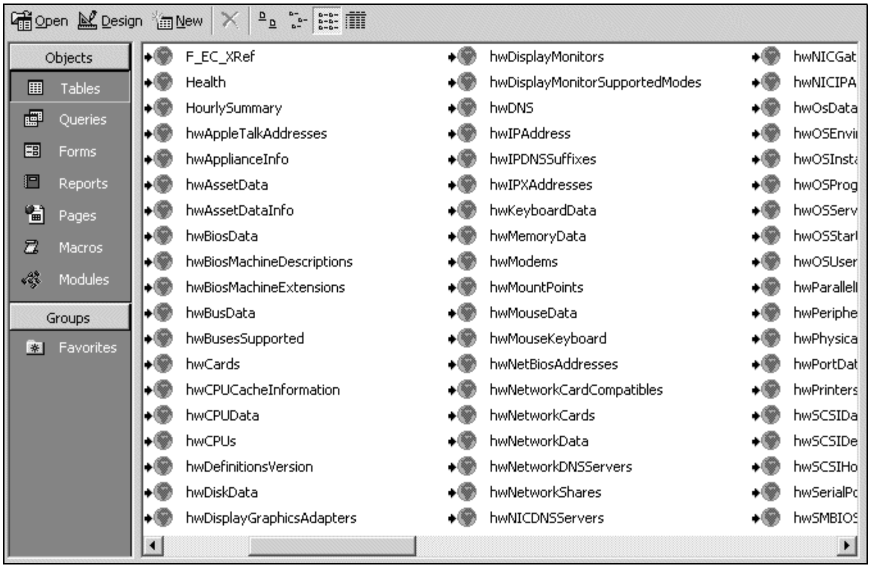
Fields in table 'hwDefinitionsVersion1':

hwDefinitionsVersion

To ensure data integrity and to update records, you must choose a field or fields that uniquely identify each record. Select up to ten fields.

21 Select **hwDefinitionsVersion** and click **OK.**

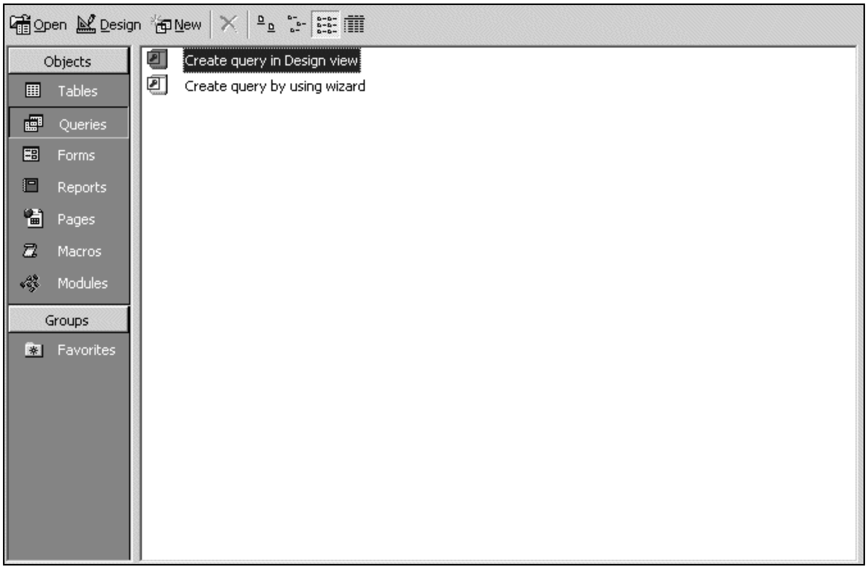
You are returned to the **Tables** Tab which shows the newly linked Network Discovery tables.



F_EC_XRef
Health
HourlySummary
hwAppleTalkAddresses
hwApplianceInfo
hwAssetData
hwAssetDataInfo
hwBiosData
hwBiosMachineDescriptions
hwBiosMachineExtensions
hwBusData
hwBusesSupported
hwCards
hwCPUCacheInformation
hwCPUData
hwCPUs
hwDefinitionsVersion
hwDiskData
hwDisplayGraphicsAdapters
hwDisplayMonitors
hwDisplayMonitorSupportedModes
hwDNS
hwIPAddress
hwIPDNSSuffixes
hwIPXAddresses
hwKeyboardData
hwMemoryData
hwModems
hwMountPoints
hwMouseData
hwMouseKeyboard
hwNetBiosAddresses
hwNetworkCardCompatibles
hwNetworkCards
hwNetworkData
hwNetworkDNSServers
hwNetworkShares
hwNICDNSServers
hwNICGat
hwNICIPA
hwOsData
hwOSEnvi
hwOSInsta
hwOSProg
hwOSServ
hwOSStar
hwOSUser
hwParallel
hwPeriphe
hwPhysica
hwPortDat
hwPrinters
hwSCSIDa
hwSCSIDe
hwSCSIHo
hwSerialPc
hwSMBIOS

Step 6: Create a basic license query

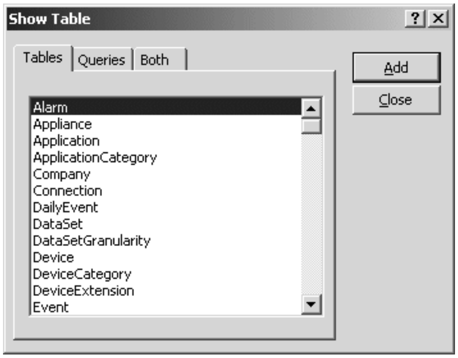
22 From the **Objects** list, select **Queries**.



Create query in Design view
Create query by using wizard

23 Double click **Create query in Design view.**

The **Show Table** dialog appears.

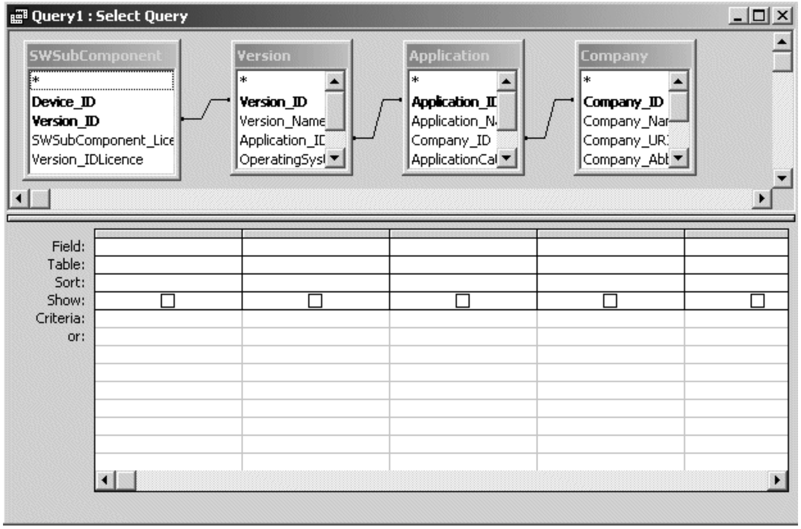


24 In the **Tables** tab page, select:

- SWSubComponent
- Version
- Application
- Company

25 With the table selected, click **Add**, then **Close**.

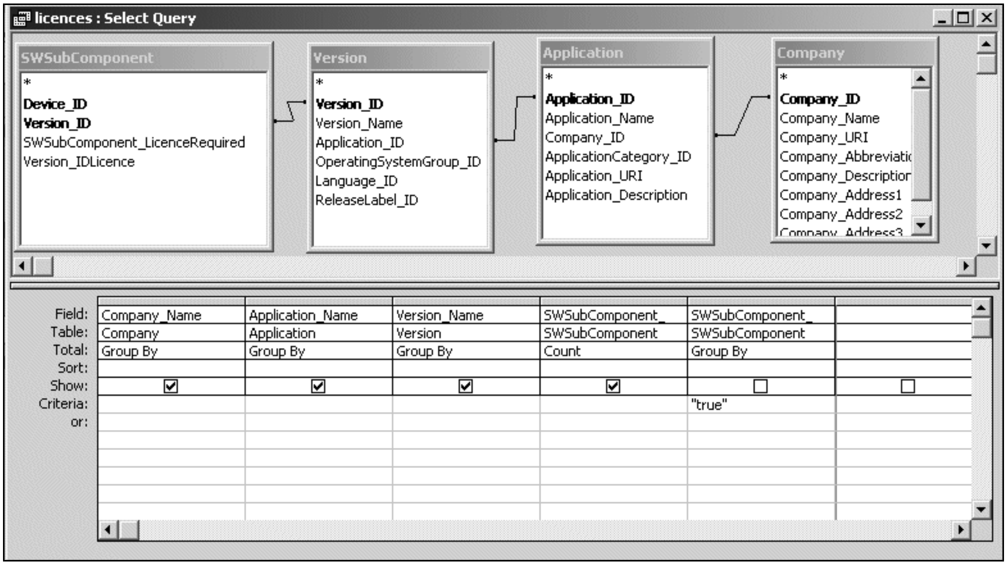
The table displayed is similar to this:



26 Go to **File** > **Save As** and save the query

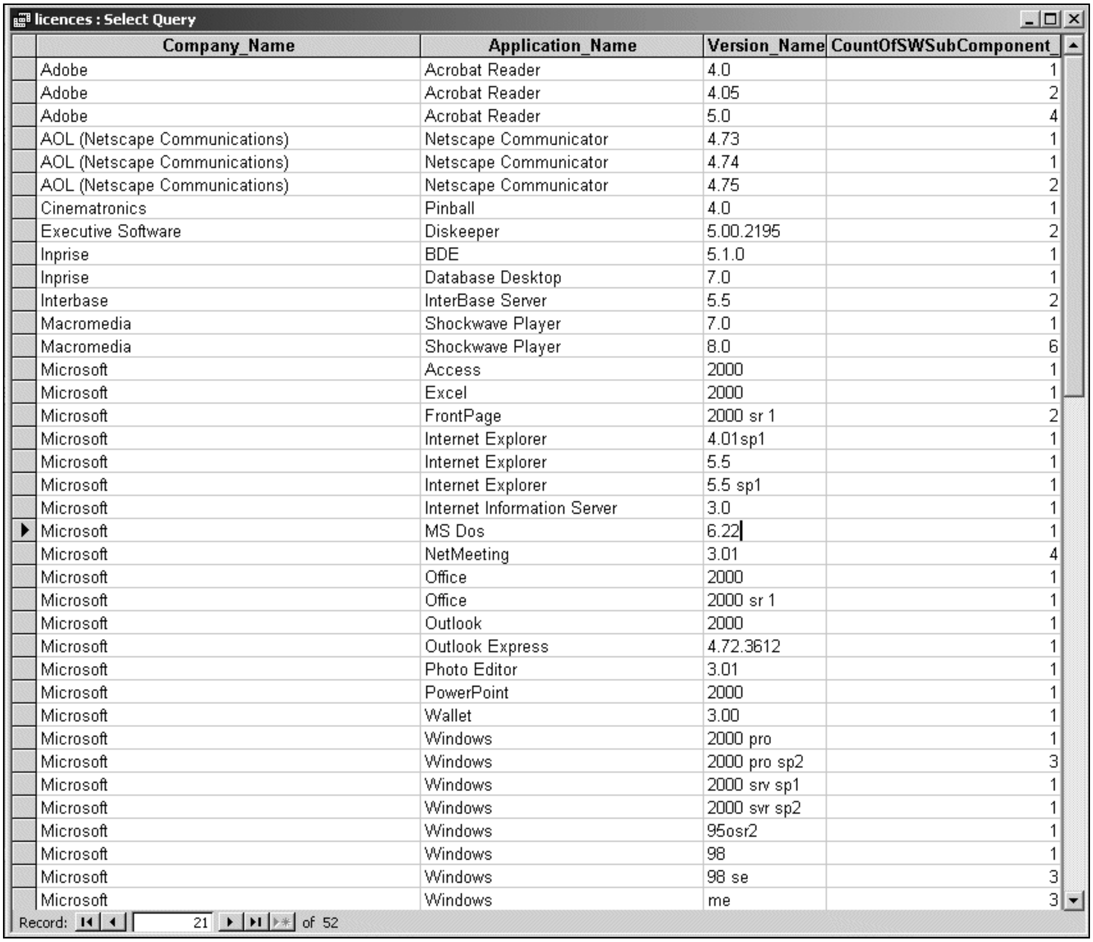
In this example, we have called it **Licenses.**

27 Enter the query field parameters as shown below:



Field:	Company_Name	Application_Name	Version_Name	SWSubComponent_	SWSubComponent_	
Table:	Company	Application	Version	SWSubComponent	SWSubComponent	
Total:	Group By	Group By	Group By	Count	Group By	
Sort:						
Show:	☑	☑	☑	☑	□	□
Criteria:					"true"	
or:						

28 Run the Query. From the **Query** pull down menu, select the **Run** option. A query is generated, showing license data from the inventory scans in the Inventory Database.



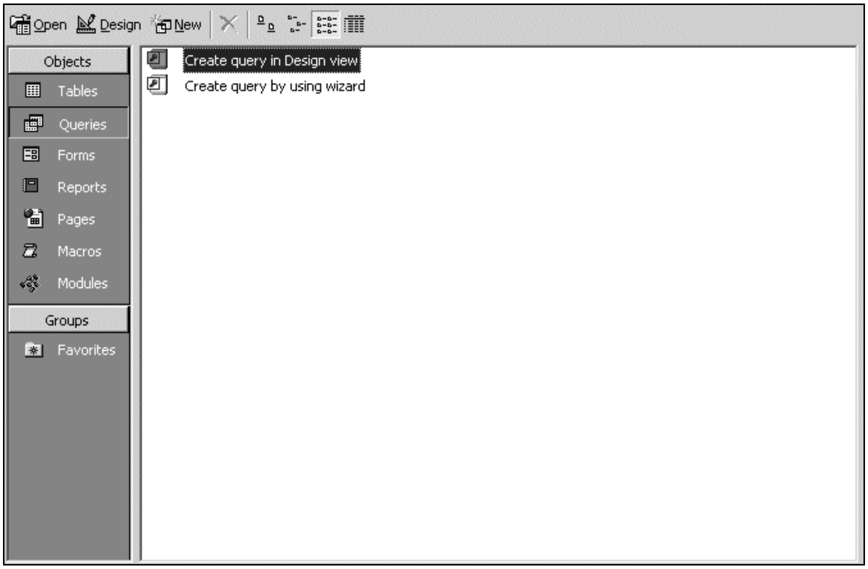
licences : Select Query

Company_Name	Application_Name	Version_Name	CountOfSWSubComponent
Adobe	Acrobat Reader	4.0	1
Adobe	Acrobat Reader	4.05	2
Adobe	Acrobat Reader	5.0	4
AOL (Netscape Communications)	Netscape Communicator	4.73	1
AOL (Netscape Communications)	Netscape Communicator	4.74	1
AOL (Netscape Communications)	Netscape Communicator	4.75	2
Cinematronics	Pinball	4.0	1
Executive Software	Diskeeper	5.00.2195	2
Inprise	BDE	5.1.0	1
Inprise	Database Desktop	7.0	1
Interbase	InterBase Server	5.5	2
Macromedia	Shockwave Player	7.0	1
Macromedia	Shockwave Player	8.0	6
Microsoft	Access	2000	1
Microsoft	Excel	2000	1
Microsoft	FrontPage	2000 sr 1	2
Microsoft	Internet Explorer	4.01sp1	1
Microsoft	Internet Explorer	5.5	1
Microsoft	Internet Explorer	5.5 sp1	1
Microsoft	Internet Information Server	3.0	1
Microsoft	MS Dos	6.22	1
Microsoft	NetMeeting	3.01	4
Microsoft	Office	2000	1
Microsoft	Office	2000 sr 1	1
Microsoft	Outlook	2000	1
Microsoft	Outlook Express	4.72.3612	1
Microsoft	Photo Editor	3.01	1
Microsoft	PowerPoint	2000	1
Microsoft	Wallet	3.00	1
Microsoft	Windows	2000 pro	1
Microsoft	Windows	2000 pro sp2	3
Microsoft	Windows	2000 srv sp1	1
Microsoft	Windows	2000 svr sp2	1
Microsoft	Windows	95osr2	1
Microsoft	Windows	98	1
Microsoft	Windows	98 se	3
Microsoft	Windows	me	3

Record: 21 of 52

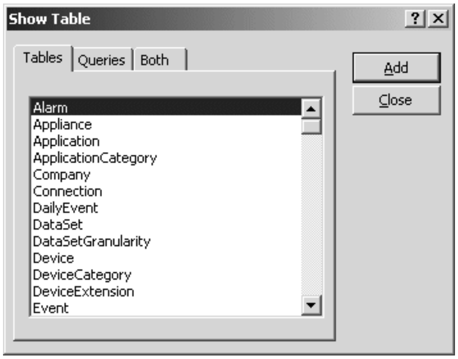
Step 7: Create a basic assets and recognition query

29 From the **Objects** list, select **Queries**.



30 Double click **Create query in Design view**.

The **Show Table** dialog appears.



31 In the **Tables** tab page, from the list, select:

- hwAssetData
- Device

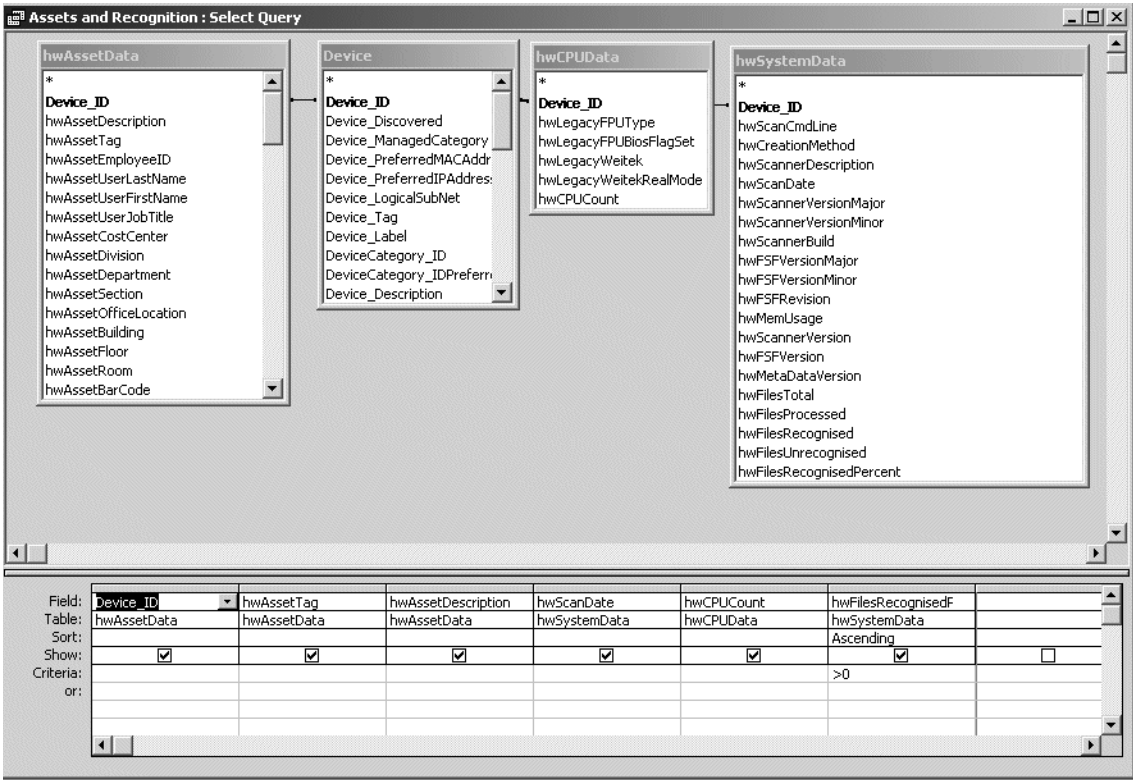
- hwCPUData
- hwSystemData

32 With the table selected, click **Close**.

The table appears.

33 Save the query. In this example we have called it **Assets and Recognition**.

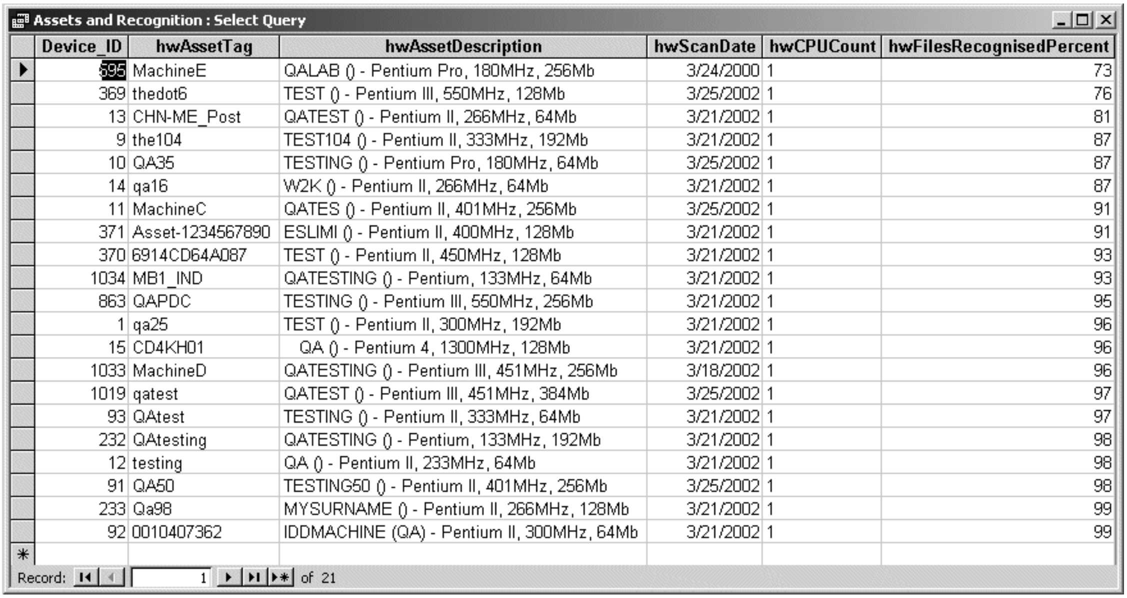
34 Enter the query field parameters as shown below:



Field:	Device_ID	hwAssetTag	hwAssetDescription	hwScanDate	hwCPUCount	hwFilesRecognisedF	
Table:	hwAssetData	hwAssetData	hwAssetData	hwSystemData	hwCPUData	hwSystemData	
Sort:						Ascending	
Show:	☑	☑	☑	☑	☑	☑	□
Criteria:						>0	
or:							

35 Run the Query. From the **Query** pull-down menu, select **Run**.

A query is generated, showing asset and recognition data from the inventory scans in the Inventory Database.



Assets and Recognition : Select Query

Device_ID	hwAssetTag	hwAssetDescription	hwScanDate	hwCPUCount	hwFilesRecognisedPercent
595	MachineE	QALAB () - Pentium Pro, 180MHz, 256Mb	3/24/2000	1	73
369	thedot6	TEST () - Pentium III, 550MHz, 128Mb	3/25/2002	1	76
13	CHN-ME_Post	QATEST () - Pentium II, 266MHz, 64Mb	3/21/2002	1	81
9	the104	TEST104 () - Pentium II, 333MHz, 192Mb	3/21/2002	1	87
10	QA35	TESTING () - Pentium Pro, 180MHz, 64Mb	3/25/2002	1	87
14	qa16	W2K () - Pentium II, 266MHz, 64Mb	3/21/2002	1	87
11	MachineC	QATES () - Pentium II, 401MHz, 256Mb	3/25/2002	1	91
371	Asset-1234567890	ESLIMI () - Pentium II, 400MHz, 128Mb	3/21/2002	1	91
370	6914CD64A087	TEST () - Pentium II, 450MHz, 128Mb	3/21/2002	1	93
1034	MB1_IND	QATESTING () - Pentium, 133MHz, 64Mb	3/21/2002	1	93
863	QAPDC	TESTING () - Pentium III, 550MHz, 256Mb	3/21/2002	1	95
1	qa25	TEST () - Pentium II, 300MHz, 192Mb	3/21/2002	1	96
15	CD4KH01	QA () - Pentium 4, 1300MHz, 128Mb	3/21/2002	1	96
1033	MachineD	QATESTING () - Pentium III, 451MHz, 256Mb	3/18/2002	1	96
1019	qatest	QATEST () - Pentium III, 451MHz, 384Mb	3/25/2002	1	97
93	QAtest	TESTING () - Pentium II, 333MHz, 64Mb	3/21/2002	1	97
232	QAtesting	QATESTING () - Pentium, 133MHz, 192Mb	3/21/2002	1	98
12	testing	QA () - Pentium II, 233MHz, 64Mb	3/21/2002	1	98
91	QA50	TESTING50 () - Pentium II, 401MHz, 256Mb	3/25/2002	1	98
233	Qa98	MYSURNAME () - Pentium II, 266MHz, 128Mb	3/21/2002	1	99
92	0010407362	IDDMACHINE (QA) - Pentium II, 300MHz, 64Mb	3/21/2002	1	99
*					

Record: 1 of 21

CHAPTER 4 How the Network Discovery schema is organized

This chapter explains how the data has been named and how and why the data refreshes at different rates. It also gives a brief descripion of each type of data available.

Topics in this chapter include:

Naming conventions

The following conventions have been used to name tables and fields in the Aggregate schema to ease reading and understanding.

Table Names

Table names start with a capital letter followed by lower case letters—“Device”, “Port”, or “Connection”—unless the name itself is an abbreviation. If the table name is an abbreviation, capital letters are kept for the acronym part of the name—“IPv4”, “MAC”, and so on.

Table names always use the singular form— “Connection” not “Connections”.

If the name of a table is formed by putting more than one word together, the words making up the name start with capitals to differentiate them— “DeviceCategory” or “PortCategory”.

Field Names

As with table names, field names start with a capital letter followed by lowercase letters—“Index” or “Description”—unless the name itself is an abbreviation. If the name is an abbreviation, the acronym part of the name keeps its capital letters— “CIR” or “URI”.

If the name of a field is formed by putting more than one word together, the words forming the name start with capitals to differentiate them—“SpeedIn”, “SpeedOut”, or “IsFullDuplex”.

Field names are always prefixed by the name of the table to which they belong, followed by an underscore character, followed by the name of the field—“Device_Label”, “Port_Speed”, “MAC_Address”.

Usually, primary keys of tables are simply sequential numbers (positive integers). In these cases, the name of the primary key field is “ID”, so the complete field (key) name becomes “<table name>_ID – Device_ID” (or “Port_ID” or “Connection_ID” and so on).

If a primary key field enters a table as a foreign key, it keeps its original name, thus indicating its origin. When two or more primary keys from the same table are imported into one table, a suffix follows each foreign key name to indicate the different purpose of the fields—“Company_IDHardware” and “Company_IDSoftware” or “DeviceCategory_ID” and DeviceCategory_IDPreferred”.

Abbreviations

Use of abbreviations has been limited to a few widely spread conventions, such as using **Dt** to abbreviate Date. Also, as explained above, industry wide acronyms were kept where it was felt that they would easily be understood:

- BIOS (Basic Input/Output System)
- CIR (Committed Information Rate)
- DNS (Domain Name System)
- FDDI (Fiber Distributed Data Interface)
- IP (Internet Protocol)
- MAC (Media Access Control)
- NMID (Network Management Identifier—the ID (key) of the Device or Port in the Appliance)
- PVCS (Permanent Virtual Connection)
- URI (Universal Resource Identifier)

An entry can come from another table—foreign keys

The Network Discovery database is normalized. That means that information occurs only in one place so that it doesn't have to be updated more than once. For instance, if a company changes its name, the new name does not have to be updated in every table that uses the company name. Company_Name is only "real" in the Company table. Any other table that refers to Company_Name is using the Company table primary key, "Company_ID".

Data refreshes at different rates

It is important to understand that Network Discovery collects and refreshes different kinds of data at different rates. Different layers of the underlying software are responsible for refreshing different areas of the data at suitable intervals. The data falls into four groups (or sub-schemas[1]), depending on the refresh rate:

- Classification
- Network Configuration
- Statistics
- Control

[1] A schema or a sub-schema is a group of database objects such as tables that are logically connected

The data sorted by refresh rate category

The sub-schemas only indicate how and how frequently the data is refreshed. They do not affect how *you* access and retrieve data. Consequently, chapter 5 presents the tables in alphabetical order, so you can find them easily.

However, the following list arranges the tables by their four sub-schemas, so that you can have a quick overview of the data available and understand which tables change more frequently than others.

The Classification sub-schema

The Classification sub-schema contains generic information common to any network. For example, it contains the list of hardware or software manufacturers (Company), the various categorization of devices or ports (DeviceCategory & PortCategory) and so on. This information is quite stable. It only changes when you upgrade Network Discovery with a new Rulebase software component

The Classification sub-schema contains the following tables:

- **Application**: This table lists all the various applications that can be found on an end-user workstation or that can be run on network devices, such as Word, Excel, and Visio.
- **ApplicationCategory**: This table lists all the categories used to regroup applications, such as Word Processors, Spread Sheets, and Accounting Packages.
- **Company**: This table lists information on companies that manufacture computer equipment and/or produce software. Other tables (for example, Device, Application) reference this table to indicate which company manufactures certain kinds of hardware (Device) or which company produces certain software (Application).
- **DeviceCategory**: This table lists various categories of devices, such as printers, workstations, switches, routers, and web servers. It also stores some attributes of these categories.
- **Family**: This table lists product families, such as hubs, routers, workstations, switches, and so on. The Device table is related to this table so that a specific device can be related to information about the product family to which it belongs.
- **Language**: This table lists all the various languages that exist. It tells what language an application supports.
- **MediaCategory**: This table lists the media category of the line, such as Ethernet, CSMA / CD, Token Ring.
- **Model**: This table lists various models of devices, such as Cisco’s Catalyst X series, HP’s LaserJet X series, Compaq’s (DEC) AlphaServer X series.

- **NetworkFunction:** Sometimes you need workstations or server platforms set up with commercial operating systems for the specific purpose of deploying a software application or a set of applications. In such cases NetworkFunction identifies the software application that is identified by the SNMP MIB II (RFC 1213, RFC 2011, RFC 2012, RFC 2013) data fields within the device.
- **OperatingSystem**: This table lists all the various operating systems that exist for networked devices—such as Windows, Unix, or Mac OS.
- **Operating System Group:** This table groups all variants (releases) of a particular generic operating system—such as Windows 95, Windows 97, Windows 2000, or Windows NT.
- **PortCategory**: This table lists categories (and subcategories) of ports, such as Token Ring, Serial, Ethernet, FDDI. Records in the Port table are related to records in this table so that each port is associated with a specific port category.
- **ReleaseLabel**: This table lists major releases of applications—Word 97, Word 2000, Word XP are Release Labels of the Word application.
- **Version**: This table lists all the various versions that exist for a specific release. For example, the release, Word 97, could have the following versions—Word 97, Word 97sr1, Word 97sr2, Word 97 sr2a/v, Word 97.

Note: An application can have Versions, without necessarily having ReleaseLabels.

The Network Configuration sub-schema

The Network Configuration sub-schema contains information about the configuration of a specific network. It is concerned with what equipment this network contains and how it is interconnected. This information is refreshed by the Network Discovery sub-system on an ongoing basis, but information is copied to the database on a daily basis (or sometimes on request, if it is determined that a more current view of that data is required).

The Network Configuration sub-schema contains the following tables:

- **Connection**: This table lists all the connections between devices or ports within a network. In other words, it represents the connectivity between nodes in a network.
- **Device**: This table lists all the hardware equipment found in a network, including end-user workstations and interconnecting devices.
- **DeviceExtension:** This table lists additional information about devices. It is used for support purposes and should be ignored.
- **Health**: This table lists all the devices showing alarms or warnings found in the Health Panel
- **IPv4**: This table lists IPv4 addresses linked to a particular device or port. Both may have several addresses.

- **MAC**: This table lists MAC or IEEE802 style addresses linked to a particular device or port. Both may have several addresses.
- **Metric**: This tables lists all the statistics shown in the statistical panel at the bottom of the Health Panel—such as Frames / sec, Errors / sec, Availability.
- **Port**: This table lists information about all the ports discovered on each of the network devices.

The Statistics sub-schema

The Statistics sub-schema contains network statistics collected on an ongoing basis and summarized either hourly or daily to minimize the amount of disk space needed to hold them.

The data retention period is one year or 500 M Bytes of data—whichever limit is reached first—except for the event table where the limit is 45 days or 100,000 records—whichever limit is reached first.

The Statistics sub-schema contains the following tables:

- **Alarm**: This table gives objects' (device or port) alarms. It describes any alarm conditions that currently exist. If no alarms exist, the table is empty. To see alarms (and other events) that have occurred in the past, look at the Event table.
- **Event**: This table contains details about recent events for the specified object (device or port). To spare disk space, events are removed from this table after a certain period of time (currently set at 45 days).
- **HourlySummary**: This table contains summaries of certain statistics for certain pre-defined groups of devices summarized on an hourly basis. To spare disk space, HourlySummary records will be deleted after a certain period of time (currently set as 1 year).

The Control sub-schema

The Control sub-schema is relevant when you have an Aggregator. It contains tables that control how the database is refreshed by multiple appliances.

The Control sub-schema contains the following tables:

- **Appliance**: This table lists all the appliances contributing to the Aggregate Database. It describes all remote appliances that are aggregated by this system. If the system is not licensed as an aggregator, then this table is empty. Note that an appliance acting as an aggregator has entries for all remote appliances in the Appliance table but does not have an entry for itself. Instead, a value of zero for Appliance_ID is used to refer to the local appliance.

- **DataSet:** This table lists all the data components that can be imported into the Aggregate database along with their characteristics. The information found here closely matches the sub schemas listed in this section.
- **DataSetGranularity:** This table lists all the possible granularities (resolutions) (certain) data sets can have.
- **LastTime:** This table contains the last time a load operation was performed for a given appliance and data set.
- **NMID**: This table lists all the internal Network Manager IDs that contribute to populate the Aggregate database, along with their matching Appliance_IDs and the corresponding Object_IDs to which they are mapped. It also contains information stating when the last update was performed on a specific object as well as the status this object has in the source appliance.
- **Object**: This table lists all the Object_IDs and which table (Connection, Device, IPv4, MAC, Port, SubComponent) they belong to. The Object table serves as a foreign key exported to these tables (Device, Ports, and so on.). This means that the primary keys of these tables can not overlap. For example, if the value 1 is assigned to the Device_ID (key) of the Device table, then that value can not appear as the value of any primary key found in any of the other tables (such as Port_ID, Connection_ID).
- **RefreshRate:** Stores the various possible values for the data set refresh rate configuration parameter.
- **Schedule:** This table contains information related to scheduled jobs (jobs under the control of the sch_schedule jobs scheduler).
- **ScheduleLog:** This table contains information and statistics about the status of jobs that have run or are currently running.
- **UpdateHistory**: This table gives the time a Network Configuration Sub-Schema update run was performed as well as how many objects were updated by that run. Note that update runs are performed on a per appliance basis.

5 CHAPTER The data listed alphabetically

This chapter shows you the tables of data available to you for use in your data access application.

For each table the following information is given:

- the sub-schema to which the table belongs
- the primary key
- the names of the fields in the table
- the type of the field
- a brief description of the field.

Alarm

The Alarm table describes any alarm conditions that currently exist. If no alarms exist, the table will be empty. To see alarms (and other events) that have occurred in the past, look at the Event table.

Primary key

- Appliance_ID, Event_ID

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Appliance table. It identifies the appliance on which the alarm occurred. A value of zero implies the alarm has occurred on the local appliance.
Event_ID	int	Foreign key into the Event table. This field identifies the event associated with this alarm.
DeviceOrPort_ID	int	Foreign key into either the Device or Port table. It identifies the device or port on which the alarm occurred. It can also be used to join to the Object table (using DeviceOrPort_ID = Object.Object_ID). The field Object.Object_Table indicates whether the object is a device or port.
DeviceCategory_ID	int	Foreign key into the DeviceCategory table. It associates the alarm with a specific kind of device.
Alarm_Priority	int	This field contains the priority for the device associated with this alarm.
Alarm_Category	enum	This field indicates what type of alarm is being described. Possible values are 'break','drop','delay','usage','error','broadcast', and ‘collision'. An alarm is defined by the device or port on which the alarm occurred (DeviceOrPort_ID), the Alarm category or type (Alarm_Category), and the alarm state (Alarm_State).
Alarm_State	enum	This field indicates what the latest state of the alarm is. The possible values are 'info' (for information only), 'ok' (the alarm condition no longer exists), 'warn' (the condition has passed the warning threshold), or 'alarm' (the condition has exceeded the alarm threshold). This field can be used together with Alarm_Category to specifically identify the alarm condition.
Alarm_StartTimeStamp	datetime	This is the time that the alarm condition was first detected.
Alarm_EndTimeStamp	datetime	This field indicates when the specific alarm condition ended.
Alarm_MAC	varchar	This field replicates the device or port (preferred) MAC address.

Field Name	Type	Description
Alarm_IP	varchar	This field replicates the device or port (preferred) IPv4 address.
Alarm_Information	varchar	This field provides descriptive information about the Alarm.

Appliance

The Appliance table describes all remote appliances that are aggregated in the Aggregate database. If the system is not licensed as an Aggregator, then this table will be empty. Note that an appliance acting as an aggregator will have entries for all remote appliances in the Appliance table but will not have an entry for itself. Instead, a value of zero for Appliance_ID is used to refer to the local appliance.

Primary key

- Appliance_ID

Field Name	Type	Description
Appliance_ID	int	This is a unique identifier for every aggregated appliance. Appliances are always given an ID greater than zero. In other tables that have an Appliance_ID field, a value of zero means the record is associated with the local appliance.
Appliance_Name	varchar	This is a descriptive name given to the remote appliance when an administrator added this appliance to the list of appliances to be aggregated.
Appliance_IPv4Address	varchar	This field contains the IPv4 address of the remote appliance and is used to access it.
Appliance_LoginUserID	varchar	This field contains the user ID that this appliance will use to access the remote server. In order for an aggregator to access a remote appliance, it needs the appliance's IP address (Appliance_IPv4Address) and a login name and password. The password is obtained by looking up this login name in the User Preferences Accounts table.
Appliance_Reachable	enum	This field indicates whether we can currently access this appliance (true) or not (false)
Appliance_ProxyConfig	int	This field is used internally.

Application

The Application table provides descriptions of various applications, such as Word, Excel or Visio, that can be found on workstations or network devices.

Primary key

- Application_ID

Field Name	Type	Description
Application_ID	int	This field is a unique identifier for the application.
Application_Name	varchar	This field contains the name of the application.
Application_URI	varchar	This field contains an address for a web site (usually the vendor's web site) that describes the application in more detail.
Company_ID	int	Foreign key in the Company table. Indicates which company is the main source of this application.
ApplicationCategory_ID	int	Foreign key in the ApplicationCategory table. Indicates the category of this application.
Application_Description	varchar	Brief description of the application.

ApplicationCategory

The ApplicationCategory table defines types of applications that may exist on a device. The Application table contains a reference to this table so that a specific application can be identified to be of a certain type.

Primary key

- ApplicationCategory_ID

Field Name	Type	Description
ApplicationCategory_ID	int	This is the unique identifier for an application category record.
ApplicationCategory_Name	varchar	This field contains a description of the application category.

Company

This table contains information on companies that manufacture computer equipment and/or produce software. Other tables (such as Application, Device) reference this table to indicate, for example, who manufactures certain kinds of hardware or who publishes certain software.

Primary key ■ Company_ID

Field Name	Type	Description
Company_ID	int	This is the unique identifier for a company record.
Company_Name	varchar	This is the name of the company. When a company changes name (for example, if it is acquired by another company), this name may indicate both the new and old names.
Company_URI	varchar	This is an address you can use to reach the company web site.
Company_Abbreviation	varchar	This field contains a brief label for the company that is shorter than the verbose name stored in Company_Name.
Company_Decription	varchar	Brief description of the company.
Company_Address1	varchar	Mailing address of the company head office.
Company_Address2	varchar	Mailing address of the company head office.
Company_Address3	varchar	Mailing address of the company head office.

Connection

The Connection table defines connections between a device or port and another device or port. In other words, it represents the connectivity between nodes in your network.

Note: Connection records are symmetric. If device A port 1 is connected to device B port 2, then there will be two records in the connection table: one indicating the device A port 1 connection to device B port 2 and the other indicating the device B port 2 connection to device A port 1.

Primary key ■ Connection_ID

Field Name	Type	Description
Connection_ID	int	This is a unique identifier of a Connection record.
Device_IDConnectedFrom	int	This field is a foreign key into Device and defines the source Device for the Connection.
Port_IDConnectedFrom	int	This field is a foreign key into Port and defines the source Port (on the source Device) for the Connection. If this information is not available, then this field is NULL.

Field Name	Type	Description
Device_IDConnectedTo	int	This field is a foreign key into Device and defines the target device for the connection.
Port_IDConnectedTo	int	This field is a foreign key into Port and defines the target port (on the target device) for the connection. If this information is not available, then this field is NULL

DataSet

This table list all the data components that can be imported into the aggregate database along with their characteristics.

Primary key

- DataSet_ID

Field Name	Type	Description
DataSet_ID	int	This is a unique identifier of a DataSet record.
DataSet_Name	varchar	Name of the data set
DataSet_ExtractCommand	varchar	Command executed to build the data set import file
DataSet_TransferCommand	varchar	Command executed to transfer the data set import file to the Aggregator. Obviously, to have anything to transfer, the Extract command must first be executed on the remote appliance.
DataSet_ImportCommand	varchar	Command executed to upload the data set to the Aggregate database. To have anything to import, the Extract & Transfer commands must first be executed.
DataSet_GranularityRequired	enum	Whether a granularity value is required for this data set (true) or not (false).
DataSet_Size	enum	Relative sizes of data sets to give the user a rough idea of how much data will be generated / transferred. Possible values are: Small, Medium, Large.
DataSet_IDRequired	int	Foreign key into the DataSet table that identifies a data set which is required to support the current data set. For example, the import of the Classification Data Set is a pre-requisite to the import of the Network Configuration Data Set.

DataSetGranularity

This table lists all the possible granularities (resolutions) (certain) data sets can have.

Primary key

- DataSetGranularity_ID

Field Name	Type	Description
DataSetGranularity_ID	int	This is the unique identifier of a Data Set Granularity record.
DataSetGranularity_Value	int	Granularity in hours

Device

The Device table describes network devices.

Primary key

- Device_ID

Field Name	Type	Description
Device_ID	int	This is a unique identifier of a device record.
Device_Discovered	enum	This field indicates whether the device was automatically discovered (true) or not (false)
Device_ResourceManaged	enum	This field is true if and only if the device is resource managed. Examples of resource managed devices are workstations and printers. Resource managed devices usually have associated records in the SubComponent table (for CPU's, disks, toner cartridges, paper trays, for example).
Device_Department	varchar	This is the name of the department that is responsible for the device. It is not currently populated.
Device_ManagedCategory	enum	This field indicates what kind of device management exists. Possible values are: ■ SNMP - Managed ■ Unmanaged ■ Cloud
Device_PreferredMACAddress	varchar	Although a device may have many MAC addresses, this is the address that is preferred when referencing this device.

Device_PreferredIPv4Address	varchar	Although a device may have many IPv4 addresses, this is the address that is preferred when referencing this device.
Device_LogicalSubNet	varchar	The logical subnet IP address.
Device_Tag	text	A short descriptive string assigned to this type of device.
Device_Label	varchar	A label used to describe this specific device. It may be a domain name or an IP address.
DeviceCategory_ID	int	This is a foreign key into the DeviceCategory table which associates this device with a specific category of devices (for example, workstations). This association is automatically performed by the system.
DeviceCategory_IDPreferred	int	Users may change the above category to something more appropriate. This is the preferred category. It is also a foreign key into DeviceCategory. If nothing is set by the user, it defaults to the value of the Device Category ID field.
Device_Description	varchar	This is a longer description of the device.
Device_ObjectID	varchar	This is the object ID (OID) from the Device's MIB.
Device_Contact	varchar	This is a contact person name and is taken from the Device's MIB.
Device_Name	varchar	Name (DNS) of the device
Device_Location	varchar	This is the device location and is taken from the device's MIB.
Device_NetBIOS	varchar	This is the NetBIOS name for the device.
Device_NetBIOSWorkGroup	varchar	This is the NetBIOS work group name for the device.
Company_IDHardware	int	This field is a foreign key into the Company table and indicates the company that manufactures the device hardware.
Company_IDSoftware	int	This field is a foreign key into the Company table and indicates the company that manufactures the software used by the device.
Family_ID	int	This field is a foreign key into the Family table and indicates the product family to which this device belongs.
Model_ID	int	This field is a foreign key into the Model table and indicates the product model of this device.

OperatingSystem_ID	int	This field is a foreign key into OperatingSystem and indicates the operating system running on the device.
Device_Priority	int	This is the user preferred priority for this device.
Device_DefaultPriority	int	This is the default numeric priority assigned to devices of this type. Users may override this default; the user value is in Device_Priority.
Device_PriorityLabel	int	A label used to represent the device priority on the map.
Device_NumberRealPorts	int	This field contains the number of actual ports that this device has.
Device_FirstFoundDt	datetime	This is the date and time when this device was first discovered.
Device_AddedToMapDt	datetime	This is the date and time when the device was added to the map.
Device_UpdatedDt	datetime	After a device is discovered, it will be revisited from time to time to check for changes. This is the date and time when it was last visited.
Device_Connections	int	Device_Connections provides a count of the number of actual connections to this device that have been detected. The field Device_NumberRealPorts is a count of the number of ports on the device but not all of these necessarily have something connected.
Device_UnusedEthernetInterface	int	This field contains a count of the Ethernet ports for this device that have nothing connected to them. A high number of unused ports may indicate that the device is under utilized.
Device_UnusedFDDIInterfaces	int	This field contains a count of the FDDI ports for this device that have nothing connected to them. A high number of unused ports may indicate that the device is under utilized.
Device_UnusedFrameRelayInterfaces	int	This field contains a count of the frame relay ports for this device that have nothing connected to them. A high number of unused ports may indicate that the device is under utilized.
NetworkFunction_ID	int	Foreign key into the Network Function table indicating the primary applications (purpose) of this device.
Device_ServicePack	varchar	This field is the name of the last service pack (if any) that has been applied to the operating system running on the device.

DeviceCategory

The DeviceCategory table defines a number of categories or types of devices and the attributes of these types.

Primary key

- DeviceCategory_ID

Field Name	Type	Description
DeviceCategory_ID	int	This field provides a unique identifier for each Device Category record.
DeviceCategory_Description	varchar	This field provides a description of the category (Router, Workstation, etc.).
DeviceCategory_SortOrder	int	This field provides a sort order value for this device type. If device reports are sorted by this field, they will be grouped by device type and presented in the order of categories defined by this field.
DeviceCategory_NetworkClassification	varchar	This field classifies the device in terms of its basic position in a network. Allowed values are Network Device and End Node.
DeviceCategory_FunctionClassification	varchar	This field classifies the device in terms of its function within a network. It has values such as Server, Router, Switch, and so on.
DeviceCategory_DefaultPriority	int	This is the default priority value associated with devices of this category.
DeviceCategory_GraphicFileName	varchar	This field provides the name of a graphic file that contains an icon to represent this type of device in the user interface.
DeviceCategory_IconSortOrder	int	Similar to Device Category Sort Order but linked to the Icon.
DeviceCategory_IconDescription	varchar	Similar to Device Category Description but linked to the Icon.
DeviceCategory_GroupID	int	Node Main (Icon) Id

DeviceExtension

This table lists additional information about devices.

Primary key

- Device_ID

Field Name	Type	Description
Device_ID	int	Foreign key into the device table. Indicates which device this information refers to.
DeviceExtension_MiscInfo	varchar	This field is used for support purposes and should be ignored.
DeviceExtension_Rule	int	This field is used for support purposes and should be ignored.

Event

The Event table contains details about recent events. In order to manage the usage of disk space, events are removed from this table after a certain period of time (currently set at 45 days).

Primary key

- Appliance_ID, Event_ID

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Appliance table indicating which appliance produced this Event. A value of zero indicates the local appliance.
Event_ID	int	This field provides a unique identifier for an event on a specific appliance.
DeviceOrPort_ID	int	Foreign key into Device or Port and indicating which device or port produced the event.
DeviceCategory_ID	int	This field replicates the Device.DeviceCategory_ID field.
Event_Priority	int	This field indicates the priority assigned to this event.
Event_Category	enum	This field indicates the type of event being described. Some possible values are: ■ Break ■ Drop ■ Delay ■ Usage ■ Error ■ Broadcast

Field Name	Type	Description
Event_State	enum	This field indicates the state of event being described. Possible values are: ■ Info ■ OK ■ Warn ■ Alarm
Event_IsCheckPoint	enum	This field indicates whether the event is a checkpoint event (true) or not (false).
Event_TimeStamp	datetime	This field indicates when the event occurred.
Event_MAC	varchar	This field replicates the device or port MAC address.
Event_IP	varchar	This field replicates the device or port IPv4 address.
Event_Information	varchar	This field provides descriptive information about the event.

Family

The Family table defines product families. The Device table is related to this table so that a specific device can be related to information about the product family to which it belongs.

Primary key

- Family_ID

Field Name	Type	Description
Family_ID	int	This is a unique identifier for a Family record.
Family_Name	varchar	This field contains a standard name for the product family.
Family_URI	varchar	This field contains a web address where more information about the product family can be found.

Health

This table lists all the devices with alarms or warnings found in the Health Panel.

Primary key

- Appliance_ID, Health_NMID, Health_DeviceOrPort, Health_Statistic

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Appliance table indicating which appliance this record relates to.
Health_NMID	int	NMID assigned by the appliance to this device (port).
Health_DeviceOrPort	enum	Indicates whether the Health_NMID is a Device or Port ID (on appliances, device and port IDs may have the same value).
Health_Statistic	enum	May be: ■ In Errors ■ Out Errors ■ Collisions ■ In Broadcasts ■ Out Broadcasts ■ In Utilization ■ Out Utilization, etc.
Device_IDCurrentConnection	int	Appliance NMID of the device this Health_NMID is connected to.
Device_IDCurrentConnectionVia	int	Appliance NMID of the cloud through which this Health_NMID is connected.
Port_IDCurrentConnection	int	Application NMID of the port this Health_NMID is connected to.
Device_IDPreviousConnection	int	Appliance NMID of the device through which this Health_NMID used to be connected.
Device_IDPreviousConnectionVia	int	Appliance NMID of the cloud through which this Health_NMID used to be connected.
Port_IDPreviousConnection	int	Appliance NMID of the port this Health_NMID used to be connected to.
Health_Value	real	Value of the statistics.
Health_State	enum	■ Alarm ■ Warning
Health_UpdatedDt	datetime	Time

HourlySummary

The HourlySummary table contains summaries of certain statistics for certain pre-defined groups of devices for a specific hourly period. In order to manage disk space usage, records in this table will be deleted after a certain period of time (currently set at one year).

Primary key

- Appliance_ID, HourlySummary_Type, HourlySummary_Statistic, HourlySummary_UnixTimeStamp,

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Appliance table indicating which Appliance the summary applies to. A value of zero, implies the summary applies to the local Appliance.
HourlySummary_Type	enum	This field indicates what sort of device grouping the summary applies to. For example, a value of “NET” indicates the summary is for network devices. A value of “LANBB” indicates the summary is for LAN backbone devices. Possible values are: ■ Point ■ Token ■ WAN ■ FDDI ■ ATMF ■ DSL ■ FRM ■ SSPN ■ LANBB ■ ATMC ■ NET
HourlySummary_Statistic	enum	This field indicates what statistic is summarized. Note that some statistics such as Downtime have a value for a given period while others (such as Volume) have a mean, peak and mean peak for the given hour. Possible values are: ■ Utilization ■ Downtime ■ Volume

Field Name	Type	Description
HourlySummary_UnixTimeStamp	int	This field indicates the hour when the summary begins (in Unix timestamp format).
HourlySummary_TimeStamp	datetime	This field indicates the hour when the summary begins.
HourlySummary_Value	real	This field indicates the value for the statistic over the indicated hour. If the statistic does not have a single value (for example, Volume), then this field is NULL.
HourlySummary_ValuePeriod	int	For statistics that only have a value (for example, Downtime), this field indicates the number of seconds over which the statistics was measured. Normally, this will be 3600 (the number of seconds in an hour). But, as a result of network or other failures, statistics are sometimes only measured over shorter time periods. If you wish to compute the value for this statistic over several hours, you should add individual hourly values weighted by this field.
HourlySummary_ValueObjects	int	How many devices have contributed to the calculation of the value statistics.
HourlySummary_ValueSamples	int	How many samples were used to compute the value statistics.
HourlySummary_Mean	float	This field indicates the mean value for the statistic over the indicated hour. If the statistic does not have a mean (for example, Downtime), then this field is NULL.
HourlySummary_MeanObjects	int	For statistics that have Means (for example, Volume) this field indicates how many devices have contributed to the calculation of the mean statistics.
HourlySummary_MeanSamples	int	For statistics that have means (for example, Volume), this field indicates the number of individual samples that were used to compute the mean statistic over the hour. If you wish to compute a mean over several hours, you should average the mean for the hours weighted by this field.
HourlySummary_Peak	float	This field indicates the peak value for the statistic over the indicated hour. If the statistic does not have peaks (for example, Downtime), then this field is NULL.
HourlySummary_PeakTime	datetime	This field indicates the time (during the hour) when the peak occurred. It is NULL for statistics that have no peak (for example, Downtime).

Field Name	Type	Description
HourlySummary_MeanPeak	float	This field indicates the mean peak value for the statistic over the indicated hour. Statistics (mean and peak) are sampled frequently during an hour. This value indicates the average of the individual peak values. If the statistic does not have mean peaks (for example, Downtime), then this field is NULL.
HourlySummary_MeanPeakObjects	int	For statistics that have Mean Peaks (for example, Volume) this field indicates how many devices have contributed to the calculation of the mean peak statistic.
HourlySummary_MeanPeakSamples	int	For statistics that have mean peaks (for example, Volume), this field indicates the number of individual samples of mean peaks that were used to compute the mean peak statistics over the hours. If you wish to compute a mean peak over several hours, you should average the mean peak for the hours weighted by this field.

IPv4

This table contains version 4 IP addresses for devices and ports. Both may have several addresses.

Primary key

- IPv4_ID

Field Name	Type	Description
IPv4_ID	int	A unique identifier of an IPv4 record.
DeviceOrPort_ID	int	This field is a foreign key into the Device or Port table.
IPv4_Address	varchar	This field contains the version 4 IP address in the usual form (that is xxx.xxx.xxx.xxx where each xxx is between one and three digits).
IPv4_PaddedAddress	varchar	This field contains the IP address in padded form suitable for sorting. That is, the address is in the form xxx.xxx.xxx.xxx where each xxx is exactly three digits long (including leading zeroes if necessary).
IPv4_NetMask	varchar	This field contains the network mask associated with the address.
IPv4_DNS	varchar	This field contains the address (either an IP address or a domain name) for the domain name server.

Language

This table contains a list of languages.The Version table links to it allowing the software version language to be identified.

Primary key

- Language_ID

Field Name	Type	Description
Language_ID	int	This is a unique identifier for a Language record.
Language_Name	varchar	Language name
Language_Code	varchar	Code used to represent the language.

LastTime

This table contains the last time a given operation was performed for a given appliance and data set. It is used internally to control the data set extraction, transfer and import processes.

Primary key

- Appliance_ID

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Appliance table indicating which appliance this record relates to.
DataSet_ID	int	Foreign key into the Data Set table indicating which data set this record relates to.
LastTime_IsFor	enum	The operation performed; Extract, Transfer, Import, Transferred File or Record Extraction.
LastTime_UnixTimeStamp	int	The time (in Unix time stamp format) of the last operation performed.

MAC

This table contains MAC or IEEE804 style addresses for both devices and ports.

Primary key

- MAC_ID

Field Name	Type	Description
MAC_ID	int	A unique identifier of a Mac record.
DeviceOrPort_ID	int	This field is a foreign key into the Device or Port table.
MAC_Address	varchar	This field contains the MAC address for the Device or Port table.
Company_ID	int	This field is a foreign key into the Company table and relates this MAC address to the Company to which the address has been assigned.

MediaCategory

This table lists the media category of the line, such as Ethernet, CSMA / CD, Token Ring.

Primary key

- MediaCategory_ID

Field Name	Type	Description
MediaCategory_ID	int	This is a unique identifier of a Media Category record.
MediaCategory_Description	varchar	Media category description
MediaCategory_DefaultDuplex	enum	Full/Half Duplex information

Metric

This table lists all the statistics shown in the Statistics panel at the bottom of the Health Panel. For example, Frames/sec, Errors/sec, Availability, and so on.

Primary key

- Appliance_ID, Metric_Name

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Application table indicating which application this statistic is linked to.
Metric_Name	varchar	The name of the statistic reported: ■ Errors ■ Exceptions ■ (number of) Devices ■ (number of) Ports etc.
Metric_Value	double	The value of the statistic reported.

Model

This table lists various models of devices, such as Cisco's Catalyst X series, HP's LaserJet X series, Compaq's (DEC) AlphaServer X series.

Primary key

- Model_ID

Field Name	Type	Description
Model_ID	int	This is a unique identifier of a Model record.
Model_Name	varchar	Model name
Model_URI	varchar	Web address where more information can be obtained for a model.
Company_IDPresentManufacturer	int	Foreign key into the Company table. Indicates which company presently manufactures this model.
Company_IDManufacturer	int	Foreign key into the Company table. Indicates which company originally manufactured this model.

NetworkFunction

Sometimes you need workstations or server platforms set up with commercial operating systems for the specific purpose of deploying a software application or a set of applications. In such cases NetworkFunction identifies the software application that is identified by the SNMP MIB II (RFC 1213, RFC 2011, RFC 2012, RFC 2013) data fields within the device.

Primary key

- NetworkFunction_ID

Field Name	Type	Description
NetworkFunction_ID	int	This is a unique identifier of a Network Function record.
NetworkFunction_Name	varchar	Name of the application.
NetworkFunction_URI	varchar	This is the web address where more information can be found on the application.
NetworkFunction_Descriptio n	varchar	A more detailed description of the network function.
Company_ID	int	Foreign key into the company table indicating which company produces this application.
ApplicationCategory_ID	int	Foreign key into the application category table indicating the category of or type of this application.

NMID

This table lists all the internal Network Manager IDs that contribute to populate the Aggregate database, along with their matching Appliance_IDs and the corresponding Object_IDs. It also contains information stating when the last update was performed on a specific object as well as the status this object has in the source appliance.

The remapping of NMIDs to Object_IDs enables the merger of two NMIDs originating from two different appliances pointing to the same object, an issue that can occur when two appliances perform discovery on overlapping networks.

Primary key

- Appliance_ID, Object_ID

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Appliance table indicating where this object information was generated.
Object_ID	int	The Object_ID corresponding to NMID_NMID in the Network Configuration database for a given appliance;
NMID_NMID	int	DeviceID, PortID or sub component ID assigned on the remote appliance.

Field Name	Type	Description
NMID_InsertedFlag	enum	Flag set to 'true' if the object pointed to by this record has just been inserted by the last update run; otherwise ‘false’
NMID_UpdatedFlag	enum	Flag set to 'true' if the object pointed to by this record has just been updated by the last update run; otherwise ‘false’.
NMID_LastUpdateDt	datetime	Time this object was last updated. This should also be a foreign key (in combination with Appliance_ID) into the UpdateHistory table.
NMID_StatusInAppliance	enum	This indicates which status this object has in the remote appliance. Possible values are: ■ Active ■ Trashed - Manual ■ Trashed - Auto ■ Deleted

Object

The Object table maps the key for a device, port, IPv4 address, MAC address, connection or sub component. Each of these types of object has a unique identifier (that is an identifier used for an object will not be re-used for any other object type). Some tables such as IPv4 contain a field (such as IPv4.DeviceOrPort_ID) that could be a reference back to more than one possible table. By joining to the Object table, you can determine which table is actually referenced.

Primary key

- Object_ID

Field Name	Type	Description
Object_ID	int	This is the identifier of an object record (such as a device, a port, IPV4 address, and so on).
Object_Table	enum	This indicates which table uses this specific identifier. Possible values are Device, Port, IPv4, MAC, Connection or SubComponent.

OperatingSystem

The OperatingSystem table provides names and descriptions for operating systems. This table is linked to from the Device table and is used to identify the operating system a particular device runs.

Primary key

- OperatingSystem_ID

Field Name	Type	Description
OperatingSystem_ID	int	This is a unique identifier of an Operating System record.
OperatingSystem_Name	varchar	This field contains the standard name for this operating system.
OperatingSystem_URI	varchar	This is the web address where more information about the operating system can be found.
OperatingSystem_ShortName	varchar	This is a shorter name associated with the operating system.

OperatingSystemGroup

This table groups releases of a specific Operating Systems contained in the OperatingSystem table. This table groups all variants (releases) of a particular generic *Operating System*—such as Windows 95, Windows 97, Windows 2000, or Windows NT.

Primary key

OperatingSystemGroup_ID

Field Name	Type	Description
OperatingSystemGroup_ID	int	This is a unique identifier of an Operating System Group record.
OperatingSystemGroup_Name	varchar	The name of the Operating System group.
OperatingSystemGroup_Short Name	varchar	A short name for the Operating System group

Port

The Port table contains information about each port that is discovered on each device.

Primary key

- Port_ID

Field Name	Type	Description
Port_ID	int	The unique identifier of a port record.
Port_Index	varchar	Each port on a device is assigned a unique index string which is stored in this field.
Port_SpeedIn	real	This field contains the maximum incoming speed for this port.
Port_SpeedOut	real	This field contains the maximum outgoing speed for this port.
Port_Speed	real	This field contains the maximum of both Port_SpeedIn and Port_SpeedOut and is used when a single speed value for the port is required.
Port_IsFullDuplex	enum	This field indicates whether the port is full duplex ('true') or not ('false').
Port_PreferredMACAddress	varchar	Although a port may have many MAC addresses, this is the address that is preferred when referencing this port.
Port_PreferredIPAddress	varchar	Although a port may have many IP addresses, this is the address that is preferred when referencing this port.
Port_Description	varchar	This field contains a description of the port.
MediaCategory_ID	int	This is a foreign key into the MediaCategory table and relates this port with a specific media category.
PortCategory_ID	int	This is a foreign key into the PortCategory table and indicates the type of port.
Port_Label	varchar	This is a label used to describe a port and is often used in inventory reports to identify a Port in a user understandable way.
Port_Status	tinyint	This field indicates the status of the port.
Port_CIR	real	This field contains the CIR value for frame relay ports.
Port_UniqueLabel	varchar	This field is used internally.
Port_FirstFoundDt	datetime	This is the time when the port was first discovered by the system.
Port_AddedToMapDt	datetime	This field contains the time at which this port was added to the map.
Port_HasPVCs	enum	Indicates whether this port has PVC (true) or not (false).
Device_ID	int	This is a foreign key into the device table and indicates the device that owns this port.

PortCategory

The PortCategory table defines types (and subtypes) of ports. Records in the Port table are related to records in this table so that each port is associated with a specific port type.

Primary key

- PortCategory_ID

Field Name	Type	Description
PortCategory_ID	int	This is a unique identifier of a Port Category record.
PortCategory_Description	varchar	A detailed description of the port category.
PortCategory_SortOrder	int	This field can be used to sort ports into groups related to port categories.
PortCategory_LineType	varchar	This field defines the broad category of port to which this category belongs. Typical values are “WAN” (a WAN port), “LAN”(a local area network port), and so on.
PortCategory_LineSubType	varchar	This field provides a finer categorization of the broad type defined in PortCategory_LineType and contains values such as 'Point to Point Serial', 'Ethernet', 'FDDI', and so on.

RefreshRate

Stores the various possible values for the data set refresh rate configuration parameter.

Primary key

- RefreshRate_ID

Field Name	Type	Description
RefreshRate_ID	int	This field is a unique identifier of a Refresh Rate record.
RefreshRate_Value	enum	Possible values are: ■ Daily ■ Hourly ■ Every Minute

Field Name	Type	Description
RefreshRate_StartTime	enum	Determines the time (in hh:mm:ss format) at which the refresh would take place if the schedule start time offset parameter value would be equal to 0. The refresh period start time is determined by the refresh rate: Daily->beginning of the day (i.e., 00:00:00), Hourly->beginning of the hour (i.e., 00:00), every minute-> beginning of the minute (i.e. 00)
RefreshRate_MaxStartTimeOffset	enum	Maximum possible offset (in HH:MM:SS format) from the refresh period start time for this refresh rate. The maximum offset can therefore have the following values: Daily->23:59:59, Hourly->59:59, every minute -> 59.
RefreshRate_MaxExcludedPeriodIndex	int	Maximum value for a schedule excluded period index. An excluded period index is a zero based index that determines which period are to be excluded from the scheduled operation: 0 = first period, 1 = second period, etc. The maximum value is governed by the refresh rate: Daily->6 (0 = Monday, 1 = Tuesday, etc.), Hourly->23 (0 = 00:00, 1 = 01:00, …, 23 = 23:00), every minute -> 59 (0 = 00, 1 = 01, ..., 59 = 59).

ReleaseLabel

This table lists categories of versions and is linked to from the Version table. For example, 97 is the Release Label for Versions of Microsoft Word: 97, 97 Sp1, 97 Sp1 Hot Fixes, 97 Sp2, etc.

Primary key

- ReleaseLabel_ID

Field Name	Type	Description
ReleaseLabel_ID	int	This is a unique identifier of a Release Label record.
Application_ID	int	Foreign key into the Application table. Indicates which application this release label is for.
ReleaseLabel_Name	varchar	Name of the release.

Schedule

This table contains information related to scheduled jobs. It records the schedule that is used to populate the Network Discovery database for a given data set.

Primary key

- Schedule_ID

Field Name	Type	Description
Schedule_ID	int	This field is a unique identifier of a Schedule record.
Appliance_ID	int	Foreign key in the Appliance table specifying which appliance this row is related to. 0 means the local appliance;
DataSet_ID	int	Foreign key in the DataSet table that specifies which data set this row is related to. A given DataSet record can have 0 or 1 corresponding Schedule record for a given set of Appliance_ID and Schedule_IsFor values;
Schedule_IsFor	enum	Determines to which operation this schedule applies. Possible values are: Extract, Transfer, Import;
Schedule_Command	varchar	Which command to execute to perform the above operation.
RefreshRate_ID	int	Foreign key in the RefreshRate table that specifies the refresh rate to apply to this scheduled operation;
Schedule_StartTimeOffset	varchar	Offset in HH:MM:SS format from the refresh period start time. The refresh period start time and maximum possible offset values are determined by the refresh rate. The actual start time of the schedule will be compiled as: Refresh Rate_Start Time + Schedule_Start Time Offset

Field Name	Type	Description
Schedule_ExcludedPeriods	varchar	Comma separated list identifying the period(s) to exclude from the scheduled operation. The values in that list are integer specifying the zero-based indices of the period(s) to exclude from the refresh. Valid values are determined by the refresh rate: Daily->[0-6] (0 = Monday, 1 = Tuesday, etc.), Hourly->[0-23] (0 = Midnight, 1 = 1:00 AM, etc.), every minute -> [0-59] (0 = first minute, 1 = second minute etc.).
DataSetGranularity_ID	int	Foreign key in the DataSetGranularity table that specifies the granularity configuration parameter to apply to this scheduled operation;
Schedule_CurrentRunControlDt	datetime	Date used internally to control the scheduling process. Default->2000-01-01 00:00:00;
Schedule_Retries	int	How many times a failing run has been attempted. Default: 0;
Schedule_MaxRetries	int	How many times the scheduler should attempt to run a failing job. Default: 0;
Schedule_Comment	varchar	Comment on / description of this schedule record;
Schedule_UpdateDt	datetime	When that Schedule record was last updated.
Schedule_Suspended	enum	Whether the current schedule record is suspended (not active) or not. Possible values are: true, false. Default -> false;
Schedule_IsPresentlyRunning	enum	Whether the scheduled job is currently running (true) or not (false).

ScheduleLog

This table contains information about status of jobs that have run (or are currently running).

Primary key

- Schedule_ID, ScheduleLog_RunStartDt

Field Name	Type	Description
Schedule_ID	int	foreign key into the Schedule table record associated with this ScheduleLog record;
ScheduleLog_RunStartDt	datetime	When the scheduled job started;

Field Name	Type	Description
ScheduleLog_RunEndDt	datetime	When the scheduled job ended;
ScheduleLog_RunOutcome	enum	Whether the last run completed successfully or not. Possible values are: Succeeded, Failed, or Unknown. Default->Unknown;
ScheduleLog_ParentProcessID	int	Process ID of the parent processes of this job. Used internally by the parent process to write job statistics of its child process (this job) when it terminates.
ScheduleLog_RunExecutionStartDT	datetime	When the scheduled jon effectively started to run Jobs may have to wait for a while for other concurrent jobs to terminate before they can actually start executing.
ScheduleLog_SystemTimeSec	float	Number of seconds used by the operating system to control the job executions.
ScheduleLog_UserTimeSec	float	Number of seconds consumed by the job while executing.
ScheduleLog_	float	Peak amount of memory (in KByte) used by the job while executing.

UpdateHistory

This table provides a brief history of when updates to network configuration information on each appliance were done. However, you cannot determine exactly what changes or additions were made.

Primary key

- Appliance_ID, UpdateHistory_StartDt

Field Name	Type	Description
Appliance_ID	int	Foreign key into the Appliance table indicating the Appliance to which this update record applies. A value of zero implies the local appliance.
UpdateHistory_StartDt	datetime	This field contains the time at which the update started.
UpdateHistory_EndDt	datetime	This field contains the time at which the update ended.
UpdateHistory_RecordCount	int	This field contains the number of records processed during the update.

Version

This table lists all the various versions that have been identified for a specific application.

Primary key

- Version_ID
- Operating System Group ID

Field Name	Type	Description
Version_ID	int	This is a unique identifier of a version record.
ReleaseLabel_ID	int	Foreign key into the release label table indicating which release this version belongs to.
Version_Name	varchar	Name of the version
Language_ID	int	Foreign key into the language table indicating which language this version supports.
Application_ID	int	Foreign key into the application table indicating which application this version is for.
OperatingSystemGroup_ID	int	Foreign key into the Operating System Group table indicating which operating systems this version of the application is able to run on.

Peregrine
SYSTEMS

November 27, 2002