BDNA

DOCUMENTATION

BDNA Discover 7.7.2

Administrator Guide

November 11, 2016

LEGAL NOTICES

BDNA Corporation
339 North Bernardo Avenue, Suite 206
Mountain View, CA 94043
USA
Phone +1 650 625 9530
Fax +1 650 625 9533

http://www.bdna.com

02500010101

Contents

Chapter 1. Overview 15

About this Book 15

Applications for Discovery and Analysis. 15

Application Administration 15

Scan Administration 15

Analytics. 16

New Feature Updates 16

How BDNA Discover Works. 18

Level 1 18

Level 2 18

Level 3 19

BDNA Discover Application Architecture 19

Components 20

MBUS. 21

Agenda Manager. 21

Rule Engines. 21

PerlCS. 22

WinCS 22

Other Components 22

Oracle 22

Configuring BDNA for a Distributed Installation 22

Specifying the MBUS Host 22

Using BDNA’s Interactive Shell 22

Scripting batchbdna.sh 23

Example Configurations. 24

Small, Distributed Installation 24

Medium, Distributed Installation 25

Chapter 2. Pre-Initialization Setup 27

About this Chapter 27

BDNA Discover and Oracle Overview 27

Verifying Oracle Database Features. 27

Verifying and Creating Tablespaces for BDNA Discover. 27

Chapter 3. Starting and Stopping BDNA Discover. 29

About this Chapter 29

Starting BDNA Discover 29
Viewing a Running Configuration 30
Starting and Stopping Components 31
Starting Component-Server Components at Boot 31
Creating and Modifying Components 32
Starting and Stopping WinCS 32
Stopping BDNA Discover 33

Chapter 4. Configuring and Running Level 1 Discovery 35

About this Chapter 35
Overview of Level 1 Discovery Operations. 35
Launching the BDNA Discover Client 35
Logging in to the BDNA Discover Client 37
Accessing Scan Administration 38
Creating Level 1 Scan Tasks 39
Deleting Scan Tasks 45
Working with Networks 46
Creating New Networks with the New Network Wizard 47
Modifying Networks. 51
Deleting Networks 53
Creating IP Exclusions (Optional) 55
Troubleshooting Level 1 Scanning: Firewalls 56
Level 1 Scanning of Oracle VM and XenServer Systems 56
Level 1 Scanning of Windows 7 Systems 56

Chapter 5. Configuring and Running Level 2 Discovery 57

About this Chapter 57
Routable Access Requirements 57
SNMP Level 2 Access Requirements. 57
UNIX Level 2 Access Requirements 57
Windows Level 2 Access Requirements 58
Working with Credentials 59
General Credential Requirements. 59
Mac OS Credential Requirements 59
SNMP Credential Requirements 59
UNIX Credential Requirements 59
VMware ESX Credential Requirements. 60
Windows Credential Requirements 60
Windows Local Administrators 60

Windows Domain Administrators 61
Windows non-Administrator Accounts 61
Additional Client-Configuration Requirements for Windows 7, Vista, and 2008 62
Requirements for Oracle VM Discovery 62
Discover Rollout Tools for Windows and Mac OS X Level 2 Data Collection 62
Task 1: Enabling Classic Sharing and Security Model 63
Task 2: Allowing Incoming Network Connections on TCP Ports 135 and 445. 63
Task 3: Allowing Remote Administration Using Administrative Tools Such as WMI 64
Task 4: Enabling Remote DCOM Access 65
Task 5: Granting WMI Privileges and Network Access to a User Account 65
Task 6: Granting Registry Read Permissions to a User Account 66
Rollout Tools Information and Syntax 66
wmisd.exe 66
regsd.exe 68
dcomsd.exe 70
Applying Remote DCOM Access Manually 71
Required Windows Services. 71
Known Issues 72
Running the Rollout Tools 73
Windows Level 2 Credential Rollout Methodologies Compared .73
SNMP Credential Requirements 75
BDNA Discover Rollout Tools for Mac OS X. 77
GUI 77
Rollout Script 77
Automation 78
Testing Credentials. 78
BDNA Discover Credential Checker Overview. 78
Other Methods for Testing Credentials 79
Applying Credentials Selectively. 79
Managing Credentials. 80
Creating New UNIX Credentials for Level 2 Discovery 80
Creating New Windows Credentials for Level 2 Discovery 83
Creating New SNMP Credentials for Level 2 Discovery. 84
Creating New SMIS Credentials for Level 2 Discovery 85
Modifying Existing Credentials 87
Credential Precedence 87
Creating Level 2 Scan Tasks 88

Chapter 6. Configuring and Running Level 3 Discovery 95

About this Chapter 95

Overview of Level 3 Discovery Operations 95
Credentials Requirements. 95
Oracle User vs. OS User Database Authentication 96
Managing Level 3 Credentials 96
Creating a New Credential for Level 3 Discovery 96
Modifying an Existing Credential for Level 3 Discovery 98
Creating Level 3 Scan Tasks 99
Windows SQL Server Level 3 Discovery Requirements 103

Chapter 7. Troubleshooting Discovery 105

About this Chapter 105
Troubleshooting for Level 1. 105
No Substitute for Experience 106
BDNA Discover Support Can Help 106
General Tools for Troubleshooting 106
Troubleshooting for Level 2. 107
Solving Level 2 Issues 107
Level 2 Negative Impact on Target Host Issues. 107
Level 2 Access Failures 108
Discovering Level 2 Access Failures 109
Finding Patterns in a Complete Access Failure 109
Working with the Credential Usage History Report. 110
Finding Patterns in a Partial Access Failure 110
Remedying Failures 111
Incomplete Level 2 Data. 112
Discovering Incomplete Level 2 Data 112
Causes of Incomplete Level 2 Data 112
Preventing Incomplete Level 2 Data 113

Chapter 8. Configuring BDNA for Secure Operations 115

About this Chapter 115
Level 1 Network Security Requirements 115
Port Scanning 115
Potential Risks of Port Scanning 115
Port Fingerprinting 115
Potential Risks of Port Fingerprinting 116
Level 2 Network Security Risks. 117
UNIX Level 2 Network Security 117
Potential Risks of Level 2 on UNIX. 117

Windows Level 2 Network Security 117
Potential Risks of Level 2 on Windows 118
Configuring the BDNA Discover Interface for SSL 118
To Enable SSL in BDNA 118

Chapter 9. Managing Groups 121

About this Chapter 121
Creating, Modifying, and Removing Groups. 122
Creating a New Group 122
Modifying an Existing Group. 128
Removing a Group 129

Chapter 10. User Management and Access Control 131

About this Chapter 131
Role Types 131
Managing Native Users 131
Creating a Native BDNA Analyst User 132
Modifying a Native BDNA Analyst User 135
About Access Control 136
Roles and Access Control. 137
Using Groups for Access Control. 137
Mixing Group Types for Access Control 137
Access Control Example Illustration 138
Users and Multiple Roles 138
Only Groups with Active IPs are Available for Grants 138
Workflow for Configuring Access Control 139
Creating Roles 139
Enabling Access Control 140
Disabling Access Control 140
Modifying Roles. 140
Exporting and Importing Users and Roles 140
Using BDNA with LDAP 141
LDAP Users 141
Distinguished Name 141
Working with LDAP Groups 141
Overview of BDNA Authentication Using LDAP. 142
Connecting BDNA to LDAP 142
Setting BDNA Properties for LDAP through BDNA Discover .143
Setting BDNA Properties for LDAP through the Command Line. 145

Setting Authentication Mode 149
Connecting BDNA to LDAP 151

Chapter 11. Monitoring Discovery 153

About this Chapter 153
What to Look For from a Scan 153
Scan Status and Scan Progress 154
Scan Status 154
Scan Status Messages 154
Monitoring the Progress of a Scan 155
Scan Monitoring Reports 160
Report Drilldown 161
Scan Coverage Reports. 162
Scan Summary by Network 162
Level 1 Scan Summary by Network 163
Level 2 Scan Summary by Network 163
IP/Host Elements 163
Level 2 Coverage 164
Most Recently Modified Elements 164
Element Creation and Modification Over Time 165
Collection Tasks Reports 165
Task Summary 166
Task Summary by Type of Task. 167
Task Flow Over Time 168
Currently Queued Tasks Over Time. 168
Currently Queued Tasks 168
Task Backlog 168
Long-Running Tasks. 169
Pending Unscheduled Discovery Tasks 169
Diagnostics Reports 169
Host-Based Analysis. 169
Components Reports. 170
Rule Execution Status 170
How Current Is Agenda Manager. 171
Database Reports 172
Free Tablespace 172
Accessing the Event Viewer. 172
SanityCheck Command in the BDNA Shell 172
Other Monitoring and Administrative Commands in the BDNA Shell 173
Refreshanalytics 173

refreshpub 173
Key Log Files 174
cle.CLE*.log 174
dw.log 175
Maintenance Tasks 175

Chapter 12. Finishing a Scan and Generating Reports 177

About this Chapter 177
Snapshot Discovery Data 177
Building Analytics Reports 177

Chapter 13. BDNA Discover Utility Reference 179

About this Chapter 179
Installing and Running the BDNA Discover Credential Checker 179
Launching BDNA Discover Credential Checker 180
Using BDNA Discover Credential Checker 181
Importing 182
Windows Error Messages 183
SSH/Telnet Error Messages 184
Advanced Usage 184
Known Limitations 185
BDNA Discover Network Range Comparison Tool 185
Installing BDNA Discover Network Range Comparison Tool 185
Launching BDNA Discover Network Range Comparison Tool 185
Preparing Input Files 185
Defining Input File Format 186
Analyzing Input Files 186
Known Limitations 187
BDNA Discover Bulk Import Validation Tool 187
Installing BDNA Discover Bulk Import Validation Tool 188
Launching BDNA Discover Bulk Import Validation Tool 188
Preparing Input Files 189
Creating a Template 189
Validating Input Files 191
References 193
Networks/Logical Sets 193
Exclusions 195
Credentials 195
SNMP Credentials 196

SSH Key Credentials 197

Chapter 14. Working with Collection Stores 199

About this Chapter 199

Starting the Application Agent. 199

Starting the BDNA Shell 200

Initializing the Collection Store 200

Chapter 15. Advanced Topics 203

About this Chapter 203

Directory Exclusion 203

Enabling the Display of Microsoft Office/Microsoft OS License Keys 204

Modular Collection Infrastructure (MCI) for UNIX Level 2. 205

Steps for Using MCI. 205

Running the Level 2 Scan. 206

Customer-Provided Attributes 207

How CPA-Assisted Discovery Works 208

Setting up Customer-Provided Attributes. 208

Driver-File Schema. 209

INI Files 210

Sample Configuration Files 210

UNIX 210

Windows 212

Appendix A. Working with Bulk Load Files 215

About this Appendix 215

Command-Line Usage 215

General Format of Input File 216

Importing Bulk Load Files 216

Generating Bulk Load Files 217

Credential Bulk Load Files. 217

Credential Bulk Load Input File Format 217

Bulk Task Load Files 218

Bulk Scan Task Input File Format 218

Exporting Bulk Load Files 218

Using Bulk Load From The Command Line 219

BulkLoader Arguments 219

BulkCommon Arguments. 220

Examples: Bulk Import. 220

Scan Tasks 221
Individual Description of Each Input Column 221
Column 1: Task Type 221
Column 2: Packages 222
Column 3: Locations 223
Column 4: Start_Scan 223
Column 5: Stop_Scan 223
Column 6: Shift 223
Column 7: Scan_interval 224
Column 8: Credentials 224
Column 9: Scan_Name 225
Column 10: Scan_Desc 225
Column 11: priority 225
Networks 225
IP Exclusions 226
Groups 227
Credentials 228
Namespaces 228

Appendix B. BDNA Shell Command Reference 229

About this Appendix 229
Command Summary 229
bounce 229
buildPresRls 229
checkpoint 230
cleanDeadHost 230
connMethod 230
dedup 231
exit 231
expcoll 231
halt 232
help 232
idle 232
impcoll 233
info 233
initdb 233
initfb 234
list 234
lscomp 235
lshost 235

mkcomp 235
mkhost 236
mkipspace 236
module 236
partition. 237
property. 237
purge 238
rebalance 238
reset. 239
rmcomp 239
rmhost 240
rules 240
sanityCheck 240
scanStatus 241
shutdown. 241
start 242
startup 242
status 243
stop 243
updcomp 243
updhost 244

Appendix C. BDNA Inventory Component Reference 245

About this Appendix 245
Component Summary. 245
Application Monitor 245
Agent 246
Message Bus 246
Database Watcher 246
Agenda Manager. 246
Collection Manager 247
Collection Engine 247
PerlCS. 249
bdnaPerlCS 250
bdnaCLE. 250
WinCS 250
Rules Engine. 251
Remote Services 251

Appendix D. Completion-Code Reference 255

About this Appendix 255

Appendix E. Configuration-File Reference 261

About this Appendix 261

Configuration File Directory 261

adminCommand.properties. 261

adminQuery.properties 261

agenda.properties 261

agent.properties. 261

app.dtd 261

appLogging.properties 262

archiver.properties 262

bdna.Linux.properties 262

cle.xml 262

clm.properties 262

CLMServiceDeploy.wsdd 263

CLMServiceUndeploy.wsdd 263

component.properties 263

ConfigureMessages.properties 263

connection.properties 263

content.properties 264

ExternalMessages.properties 264

glue-config.xml. 264

j4bundle.properties 264

.keystore 264

label.properties 265

license.bin 265

license.txt 265

logging.properties 265

mbus.properties. 265

MessagesBundle.properties 266

nm.properties 266

notifierLoadTest.properties 267

precomputeQueries.xml 267

properties.xml 267

replay.config 267

ReplayScript.perl 267

ReplayScript.wincs 267

rs.properties 267

safepw 268

sanity.xml 268
scmap.conf 268
trimContainmentType.conf. 268
xsiCommand.properties 268
Configuration File Subdirectories 268
$BDNA_HOME/conf/components 268
$BDNA_HOME/conf/jiffy. 269
$BDNA_HOME/conf/sql 269

Appendix F. Viewing East-Asian Languages 271

About this Appendix 271
Viewing Double-Byte Characters in Microsoft Excel 271

Appendix G. Blackberry Server L3 Fingerprint Report Columns 273

About this Appendix 273
Blackberry Enterprise Server Report Columns 274
Blackberry Enterprise Server Component Report Columns 275
Blackberry Handheld Devices Report Columns 275

Appendix H. Deploying Enterprise Best Practices. 277

About this Appendix 277
Clearing Log Files and File Store 277
Post-Discovery Operations. 277
Exporting Schemas. 277
Archiving Log Files and File Store 278
Other Operations 279
Restoring Schemas 279

Overview 1

About this Book

Welcome to the *BDNA Discover 7.7.2 User Guide*. This guide documents the concepts and processes related to the discovery and analysis of IT assets through the use of BDNA Discover applications. The instructions in this guide assume that you have already completed all of the contingent tasks, such as installing the appropriate servers and running the Discover client, as documented in the *BDNA Discover 7.7.2 Installation Guide*.

This chapter provides an introduction to the Discover applications that enable the discovery and analysis of your IT assets.

Applications for Discovery and Analysis

BDNA Discover captures information on all networked IT hardware assets, software installed on those assets, and the utilization of key software packages such as ERP and database applications. BDNA Discover also identifies non-traditional devices such as manufacturing equipment and biomedical devices.

The discovery and analysis of your IT assets involves the following three BDNA Discover applications:

- Application Administration
- Scan Administration
- Analytics

Application Administration

Application Administration contains the tools that you use to execute administrative functions associated with your data discovery and analysis tasks.

You can also perform administrative tasks from the command line using the BDNA shell: `bdna.sh`. For detailed information about BDNA Shell commands, see “BDNA Shell Command Reference.”

Scan Administration

Scan Administration is a comprehensive tool that BDNA Discover administrators use to monitor and report on all aspects of the scan process, from start to finish and beyond. It includes the following administrative functions:

- Scan Setup

  Tasks that comprise the scan, networks included in the scan, IP addresses excluded from the scan, groups included in the scan, credentials that must be supplied to execute Level 2 and Level 3 scan tasks

- Scan Monitoring

  Scan progress, IP/Host elements, Level 2 scan coverage, element creation, collection tasks, components, tablespace available to BDNA Discover on the database server

- Performance

  Graphical performance monitoring with individual charts for Collection Activity (CLE, CLM), RULEs Engines, Agenda Manager, and Remote Services

- Event Viewer

Fatal events, errors, warnings, and information generated during the scan

- System Setup

  Displays managed properties, components, and hosts

- Maintenance Tasks

  General maintenance including credential coverage, SSH credentials, UNIX level 2 and BDNA namespace dedupping, removal of Windows profiles, and clearing of BDNA UNIX cache

  For an example of how Scan Administration is used to monitor a scan in progress, refer to Chapter 11, "Monitoring Discovery".

Analytics

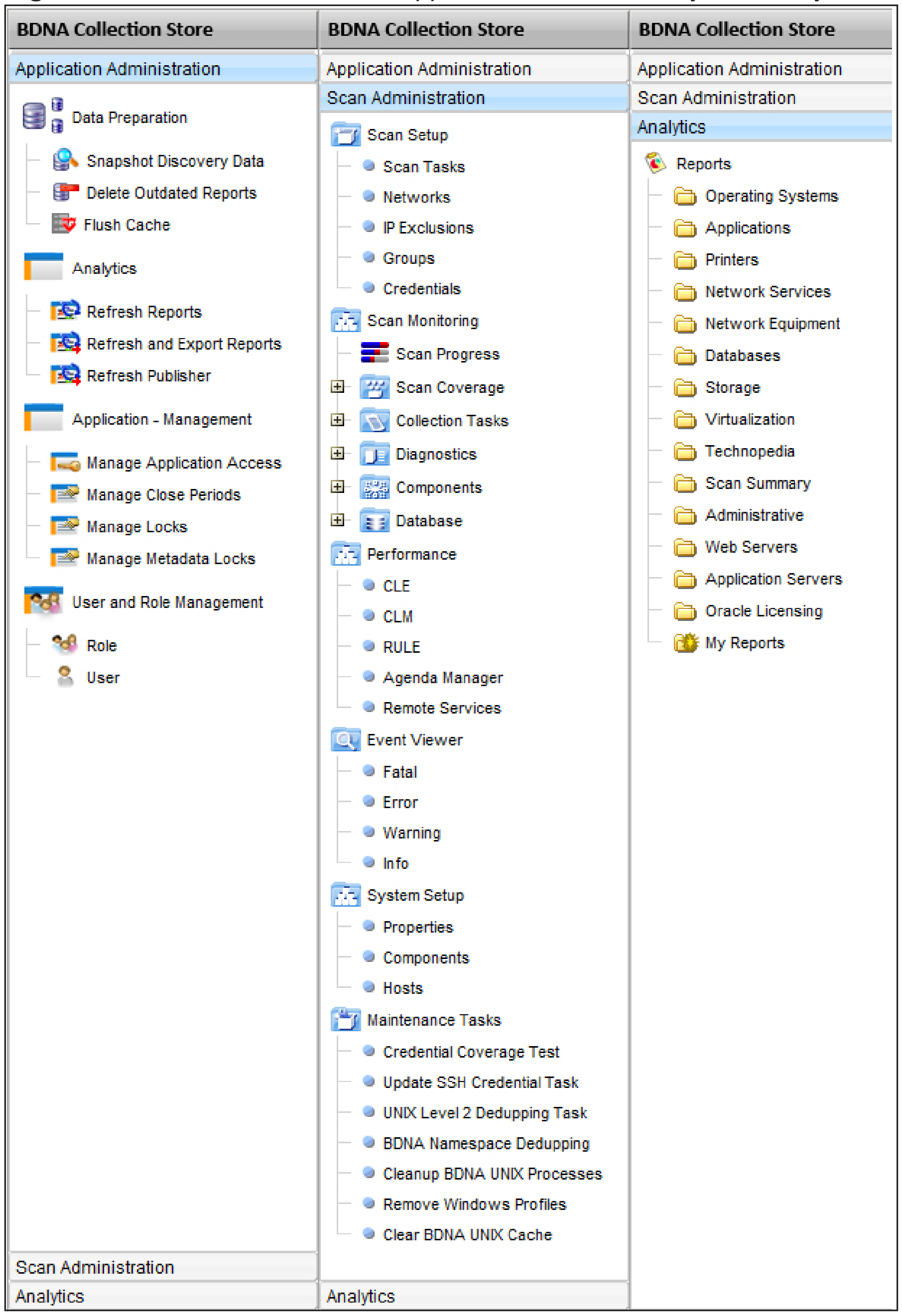
The Analytic application provides the tools needed to view and analyze the collected data. While Analytics defines many standard reports, it also provides the ability to group, filter, rollup and drill down through the collected data. The Analytics application lets you analyze virtually any specific set of assets by any set of attributes, thereby maximizing your ability to realize the benefits available from the findings in the data collected.

New Feature Updates

A new time session feature with the Application UI has been added. When the UI detects inactivity after a certain interval (usually 10 minutes), the user will receive a timeout warning and then be redirected back to the login dialog. Properties files/features for a user include:

- Conf/sessiontimeout.properties
- bdna.enableTimeout - Values are set to 'yes' or 'no.'
- bdna.inactivityInterval - Set to 'yes' if greater than 2 minutes after the user has been warned of a timeout session.

Figure 1: BDNA Discover Client: Applications for Discovery and Analysis



Within the context of the BDNA Discover applications, the process of discovery involves the following elements:

- Collection System

  The Collection System is designed to gather data using a wide variety of mechanisms. For example, the Collection System knows how to schedule and initiate a collection task and capture the data from that collection. The Collection System relies upon the Content to determine what data to collect, how to collect it, and how to interpret it.

- Content

  The Content portion of BDNA defines what data should be collected and how to collect that data. In other words, it provides the method and rules for collecting data. For example, the Content may define that the Service Pack of a Windows system should be collected and that the data should be collected using the Windows Management Instrumentation (WMI) protocol. While BDNA provides several types of Content, the majority of the Content describes this type of collection and is broadly referred to as *fingerprints*. Additional Content may include metadata about the collected data (such as an end-of-life date for an operating system) or BDNA's product catalog.

How BDNA Discover Works

BDNA Discover provides three distinct levels of discovery:

- Level 1 discovery: network-based discovery
- Level 2 discovery: system-based discovery
- Level 3 discovery: application-based discovery

As discussed above, the BDNA fingerprints define how to search and what to search for. Thus, at each level, BDNA applies fingerprints to search for hardware, software, and application level information.

Level 1

Level 1 discovery finds and classifies all accessible TCP/IP-based devices and their associated high-value network services in an organization.

The requirement for Level 1 discovery is:

- An IP address range to search
- Routable access to that range.

At Level 1, BDNA Discover identifies:

- Type of device
- Operating system
- Critical network services running on the device.

BDNA Discover can also often identify version, model, and manufacturer data. Since each device is configured uniquely, BDNA Discover will provide varying levels of detail on the devices discovered.

For example, one device might be classified as "Solaris" or "Cisco Switch" while another may be classified as "Solaris, Version 5.8, Ultra-Enterprise" or "Cisco Catalyst 2450, 12 daughter boards, 256MB memory." The level of detail returned is a function of each device's configuration. BDNA puts no requirements on customers to install agents, provide SNMP access, or otherwise alter their current environment to gather this information.

Level 2

Level 2 discovery provides detailed system and software information.

To perform Level 2 discovery on a device, BDNA Discover requires a:

- Non-administrative user account.

Specifically, BDNA Discover captures:

- System configuration (number of CPUs, memory size, machine type, etc.)
- Serial numbers (when available)
- Local attached storage
- File system layout
- Operating system version
- Operating system patch level
- Installed patches, and more.

BDNA Discover will also verify attributes collected during Level 1 Discovery and provide additional detail where appropriate.

BDNA Discover also collects information about installed software packages during this phase of discovery. BDNA develops and maintains software fingerprints for finding and collecting attributes about key installed software packages. Fingerprint development effort, focused on high-value software (such as databases, application servers, and packaged applications), provides the greatest opportunity for cost savings or risk reduction.

Level 3

Similar to Level 2 Discovery, BDNA Discover develops and maintains Level 3 fingerprints for databases and packaged applications.

Level 3 requires:

- An application-specific credential.

Level 3 discovery provides:

- Detailed discovery of targeted applications.

Specifically, BDNA Discover queries application installations and running instances to extract relevant information for asset governance such as:

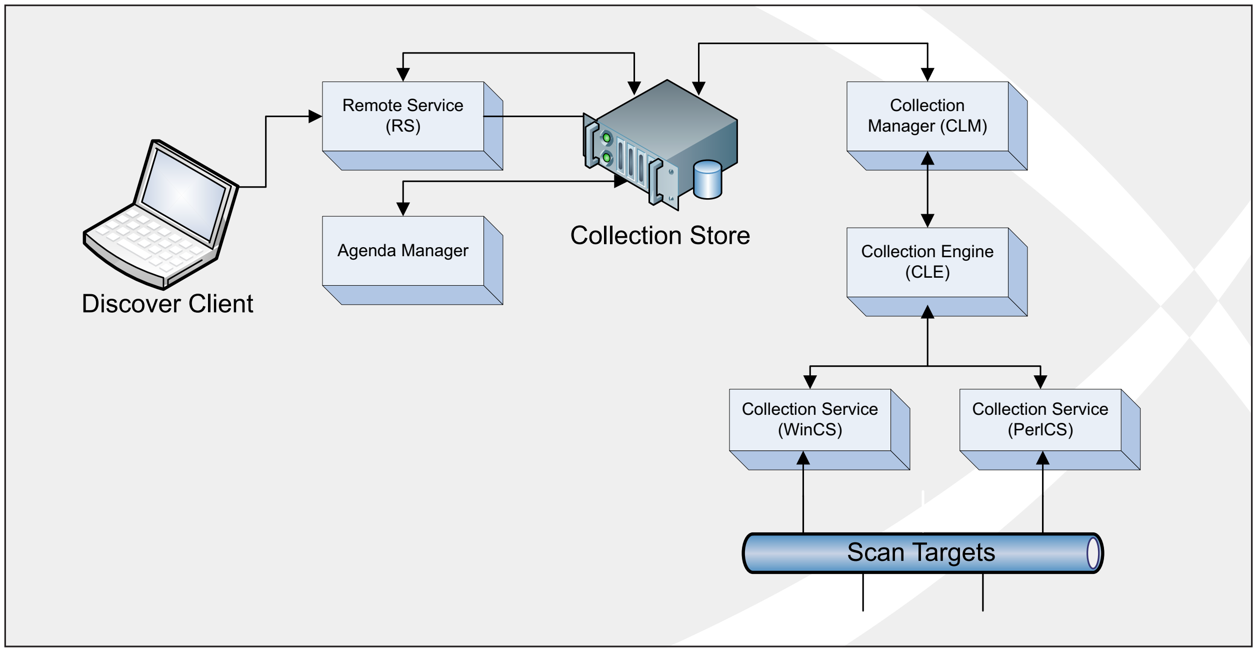
- User counts
- Storage usage
- Other application-specific data.

BDNA Discover Application Architecture

The BDNA Discover application architecture allows for scalable and distributed deployments while preserving centralized management. BDNA Discover applications implement a Web Services architecture. The system is comprised of several software components, each of which provide a specific service to the system. All components communicate across a software Message Bus using the SOAP protocol. This architecture enables BDNA Discover applications to provide horizontal scaling for enterprise implementations while using low-cost hardware.

Figure 2 shows the various components of BDNA Discover applications:

Figure 2: BDNA Discover Application Software Architecture

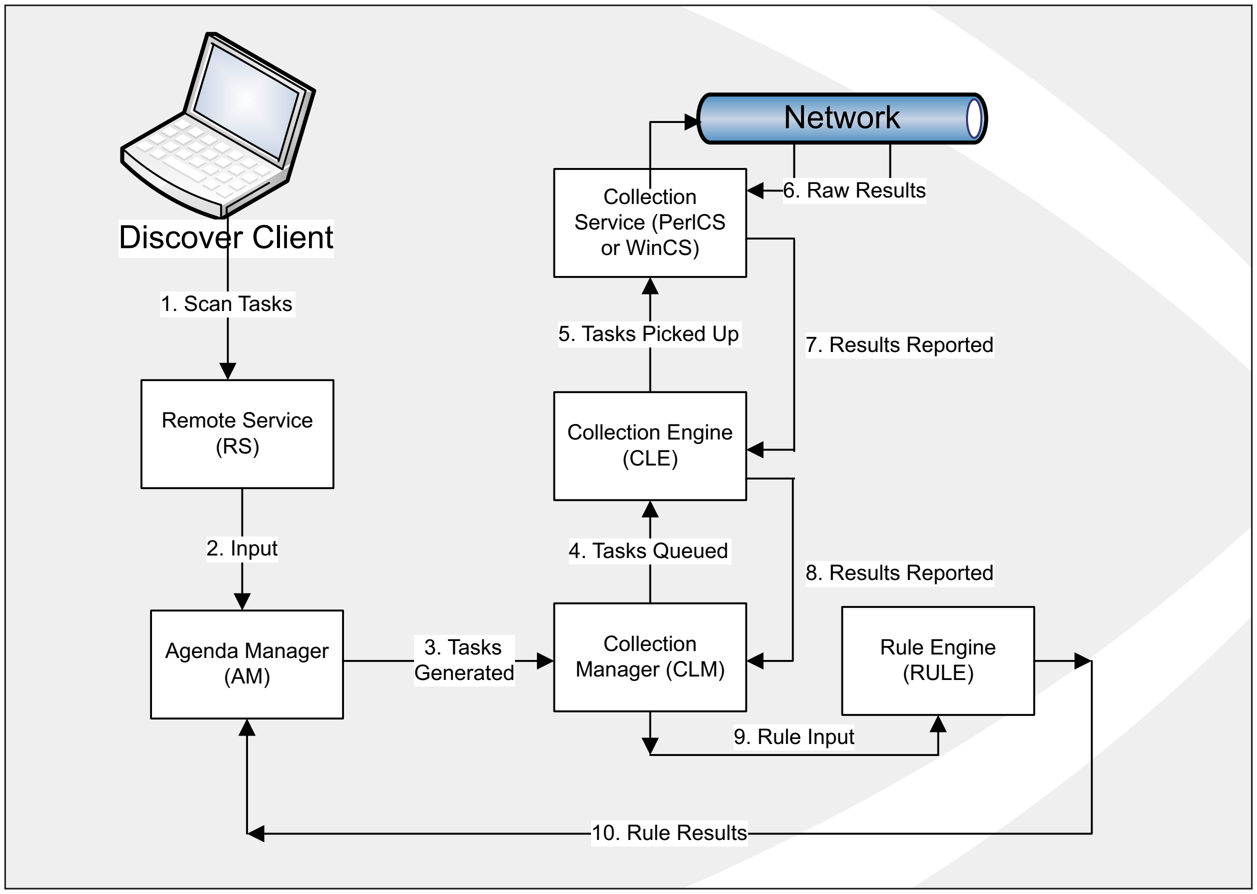


Components

The following list contains a summary description of the major components of the BDNA Discover application. Figure 3 illustrates how BDNA Discover Applications perform collections, and the interaction between the BDNA Discover application components:

See "BDNA Inventory Component Reference" for more detailed information.

Figure 3: BDNA Discover Scan Workflow Diagram



MBUS

In BDNA, the Message Bus or MBUS plays the role of the main coordinating component. The MBUS is the central point of a distributed installation. It is usually placed on the host intended to be the main host in a cluster. The MBUS Component on the main host and each component system to be controlled by the main host can be started using the command:

```
>sh startagent.sh &
```

Agenda Manager

The Agenda Manager, which is the task scheduler of BDNA, is database intensive. It is frequently placed on the same host as the MBUS in distributed installations. BDNA uses a single Agenda Manager.

Rule Engines

The Rule Engines (RULEs) do the heavy lifting of BDNA. They are memory intensive, processor intensive, and database intensive. As such, for large distributed installations an entire host or possibly several hosts in the cluster are dedicated to running RULEs.

In its default configuration, BDNA has two RULEs predefined – RULE0 and RULE1. RULE0's job is to partition work to the Rule Engines. RULE1 is the worker Rule Engine that does the processing.

A general rule of thumb is to consider creating a third RULE after the system is expected to discover more than 5,000 active IP addresses in any given scan.

PerlCS

PerlCS handles all discovery tasks except Windows Levels 2 and 3 collections. This component is very CPU and Network intensive. Within a single PerlCS, there is an adjustable level of parallelism. Parallelism may be increased to maximize throughput from any given PerlCS. Unlike other components, the memory usage is not specified by direct configuration. Memory usage can be indirectly affected by adjusting the level of parallelism.

In its default configuration, BDNA has one PerlCS component - PerlCS1. Large distributed installations have dedicated hosts to running PerlCS to maximize throughput.

WinCS

WinCS is not an MBUS controlled component. Therefore, it does not appear as a BDNA component. The WinCS component is installed on a Windows server and it is responsible for Level2 and Level3 discovery on Windows target systems. Similar to the PerlCS, WinCS connects to the CLE and has an adjustable parallelism level. Multiple WinCSs can be installed and configured to maximize throughput.

Other Components

The remaining components, i.e. the Collection Manager (CLM), Collection Engine (CLE), Reporting Service (RS), and Administrative Service (Admin), are only moderately CPU and database intensive. They can easily share a reasonably configured Linux host. They are often placed on the same host as the MBUS.

Oracle

Although Oracle is not a BDNA component, an Oracle database server is an integral part of any BDNA installation. For all distributed installations, Oracle should be running on a dedicated machine. The rest of this chapter assumes that this is so.

Configuring BDNA for a Distributed Installation

Specifying the MBUS Host

The file `$BDNA_HOME/conf/mbus.properties` defines which host is running the MBUS component. The MBUS host is set by editing the value of `bdna.mbus.url` in this file so that the hostname in the url matches the name of the MBUS host. The mbus.properties file must be edited on every machine in a cluster.

The file `$BDNA_HOME/conf/connection.properties` should be edited on the MBUS host and on the host running the bdnaPerlCS component.

Note: For more information on editing `$BDNA_HOME/conf/connection.properties`, refer to "Initializing BDNA Inventory" on page 15.

Using BDNA's Interactive Shell

Using BDNA's interactive shell is the recommended method for specifying component layout.

- Create a BDNA host entry

  BDNA must know about a host before the host can connect to the MBUS. Therefore, a host entry must be created for every non-MBUS host in a cluster. To make a host entry, run the BDNA Shell command `mkhost`, which takes a single argument—a hostname. The following command issued from within bdna.sh would create a host entry for a specific host in a cluster:

  ```
  mkhost bdna-2.foo.com
  ```

- Allocate more memory to a component

  Change the amount of memory allocated to a component (except for PerlCS/WinCS components) with the BDNA Shell command `updcomp`. The following command issued from within bdna.sh will allocate 1 gigabyte of memory to the Agenda Manager:

  ```
  updcomp –Dmx=1g agendaManager
  ```

  Note that it is possible to specify megabytes instead of gigabytes, so “1024m” would be equivalent to “1g”. Decimal points may be used, so “1.1g” would be a valid value as well.

- Move RULE1

  For large distributed installations, each RULE may be installed on a dedicated machine. To do this, move RULE1 off of the MBUS host with the following command issued from within `bdna.sh`:

  ```
  updcomp -Dhost=bdna-2.foo.com -Dmx=1.7g RULE1
  ```

  Note that this will also set the memory allocation for RULE1 at the same time.

- Create Another Rule Engine

  For those instances when a second RULE is needed, the following command issued from with `bdna.sh` will create a second RULE on the host machine specified:

  ```
  mkcomp --active -Dmx=1.7G
  -Dhost=bdna-2.foo.com -Ddc=discovery rule RULE2
  ```

- Create Another PerlCS

  To create a second PerlCS, issue the following command from with `bdna.sh`:

  ```
  mkcomp -Dhost=bdna-2.foo.com --active PerlCS PerlCS2
  ```

Scripting batchbdna.sh

The BDNA Shell has two modes; interactive and batch mode. Batch mode allows for the scripting of routine tasks and configuration. Command result error codes also help the administrator quickly determine whether or not the execution of a command was successful.

BDNA provides a simple wrapper script in `$BDNA_HOME/bin` named `batchbdna.sh`. This script calls `bdna.sh` in batch mode. Assume there is a script in the current directory named `customSettings.sh` containing these lines:

```
#!/bin/bash

sh batchbdna.sh 2>/home/bdna/cmdResult.log << EOF >> /dev/null
mkhost bdna-2.foo.com
updcomp –Dmx=1g agendaManager
updcomp -Dhost=bdna-2.foo.com -Dmx=1.7g RULE1
mkcomp --active -Dmx=1.7G -Dhost=bdna-2.foo.com \
-Ddc=discovery rule RULE2
mkcomp -Dhost=bdna-2.foo.com --active PerlCS PerlCS2

exit

EOF
```

All of the actions above could be performed by issuing:

```
./customSettings.sh
```

The result of those commands would then be located in `cmdResult.log`:

```
Command-Result-Code: 1
Command-Result-Code: 0
Command-Result-Code: 1
Command-Result-Code: 1
Command-Result-Code: 1
```

The interpretation of the command result code is binary in nature; 0 indicates a successful execution of the associated command and 1 indicates that an exception has occurred.

Example Configurations

Small, Distributed Installation

The following example configuration should be suitable for performing discovery on up to 5,000 active IPs with moderate Level 2 coverage, within 12 hours. The cluster is made up of three hosts: two Linux hosts and one Windows host. The Linux hosts are named bdna-1.foo.com (bdna-1) and bdna-2.foo.com (bdna-2). The Windows host is named bdna-3.foo.com (bdna-3).

The Windows machine is dedicated to running WinCS. The rest of the components are divided between bdna-1 and bdna-2 as follows:

Table 1: Example of Distribution of Components on a Small, Distributed Installation

bdna-1	bdna-2	bdna-3
MBUS agendaManager CLM1 CLE1 RS bdnaPerlCS	PerlCS1 RULE0 RULE1	WinCS

Also, the following components will have their memory allocations configured:

Table 2: Example of Memory Allocated to Components on a Small, Distributed Installation

Component	Default MX	Updated MX
agendaManager	128m	512m
CLE1	256m	512m
RULE1	348m	1.7g

Note that the agendaManager and CLE1 memory allocations are very conservative in this example.

The following `configuration.txt` file will create the host entry for the non-MBUS host and create the components when fed to `bdna.sh` (refer to "Scripting batchbdna.sh" on page 23 for instructions on how to do this):

```
mkhost bdna-2.foo.com
updcomp –Dmx=1g agendaManager
updcomp –Dmx=1g CLE1
updcomp –Dhost=bdna-2.foo.com RULE0
updcomp –Dhost=bdna-2.foo.com -Dmx=1.7g RULE1
updcomp –Dhost=bdna-2.foo.com PerlCS1
```

Medium, Distributed Installation

The following example configuration should be suitable for performing discovery on between to 5,000 to 20,000 active IPs with moderate Level 2 coverage, within 12 hours. The cluster is made up of 6 hosts: four Linux hosts and two Windows hosts. The Linux hosts are named bdna-1.foo.com (bdna-1), bdna-2.foo.com (bdna-2), bdna-3.foo.com (bdna-3), and bdna-4.foo.com (bdna-4). The Windows hosts are named bdna-5.foo.com (bdna-5) and bdna-6.foo.com.

The Windows machines will be divided to running WinCS. The rest of the components will be divided between bdna-1, bdna-2, bdna-3, and bdna-4 as shown:

Table 3: Example of Distribution of Components on a Medium, Distributed Installation

bdna-1	bdna-2	bdna-3	bdna-4
MBUS agendaManager CLM1 CLE1 RS SYSTEM_SERVICE bdnaPerlCS	RULE0 RULE1 RULE2	PerlCS1	PerlCS2

Also, the following components will have their memory allocations configured:

Table 4: Example of Memory Allocated to Components on a Medium, Distributed Installation

Component	Default MX	Configured MX
agendaManager	128m	512m
CLE1	256m	512m
RULE1	348m	1.7g
RULE2	NA	1.7g

The following `configuration.txt` file will create the host entry for the non-MBUS host and create the components when fed to `bdna.sh` (refer to “Scripting batchbdna.sh” on page 23 for instructions on how to do this):

```
mkhost bdna-2.foo.com
mkhost bdna-3.foo.com
mkhost bdna-4.foo.com
updcomp –Dmx=1g agendaManager
updcomp –Dmx=1g CLE1
updcomp –Dhost=bdna-2.foo.com RULE0
updcomp –Dhost=bdna-2.foo.com -Dmx=1.7g RULE1
updcomp –Dhost=bdna-3.foo.com PerlCS1
mkcomp -Dhost=bdna-4.foo.com --active PerlCS \ PerlCS2
```

Pre-Initialization Setup 2

About this Chapter

This chapter provides general information about how BDNA Discover stores data in an Oracle database repository. It also provides information about preliminary steps necessary to set up an Oracle database for use with BDNA Discover.

The Oracle database repository stores all configuration, transactional, and analytical information collected by BDNA Discover. While many organizations may have a separate Oracle database administrator (DBA), some sites will rely on the BDNA Discover Administrator to provide guidance for setting up and managing the Oracle database.

BDNA Discover and Oracle Overview

BDNA Discover is a highly database-intensive application. During data collection, the database is accessed rapidly and frequently by many of the core BDNA Discover components. As such, an Oracle database supporting BDNA Discover should provide sufficient CPU and disk performance to minimize database bottlenecks.

Refer to the *BDNA Discover 7.7.2 Release Notes* for the Oracle requirements associated with the version of BDNA Discover that you are planning to deploy. Additionally, refer to the *BDNA Discover 7.7.2 Installation Guide* for verification steps related to the Oracle installation.

Verifying Oracle Database Features

BDNA Discover uses a number of standard features of the Oracle database that Oracle installs by default. However, some organizations may have chosen to remove these features. BDNA provides an installer-based check that will verify your database. Use the `-o` flag on the BDNA Discover installer to verify your Oracle database setup. Refer to the *BDNA Discover 7.7.2 Installation Guide* for more information.

Verifying and Creating Tablespaces for BDNA Discover

Prior to creating the BDNA Discover user, the BDNA Discover Administrator must verify that there is sufficient storage space available in the database. The BDNA Discover user account must be created in the USERS tablespace. The USERS tablespace is created by Oracle during a default installation. The size of the USERS tablespace varies for every organization and depends on the number of assets being discovered, the amount of data discovered for each asset, and the number of inventories retained in the FactBase.

Some organizations may have chosen to remove the USERS tablespace, or the existing USERS tablespace may not have the parameters recommended for BDNA Discover. If a USERS tablespace does not exist, the Oracle DBA will need to create one. BDNA Discover also requires a second tablespace named BDNATEMP to be created. The BDNATEMP tablespace is used to store temporary data during discoveries.

Refer to the *BDNA Discover 7.7.2 Installation Guide* for a list of recommended tablespace and related parameters and the *Discover FactBase User Guide* for more detailed information about FactBase.

Starting and Stopping BDNA Discover 3

About this Chapter

This chapter describes the basic steps for starting the BDNA Discover system, viewing what components are running, starting and stopping individual components, creating and managing components, and shutting down the system.

Starting BDNA Discover

Once the Collection Store is initialized, BDNA Discover can be started. During the initialization, several default components are created and their configurations are stored in the Collection Store. These default components are defined in the configuration file, `boot.xml`, in the $BDNA_HOME/conf directory. It is not recommended for BDNA Discover Administrators to edit the `boot.xml` file to change the initial component layout. Rather, the component layout should be altered via BDNA Shell commands. For the purposes of this section, the default component configuration will be assumed.

For a complete list of BDNA Shell commands, see “BDNA Shell Command Reference” on page 229.

To start BDNA Discover:

1. Issue the following command on the Message Bus server (you can skip to step 2 if you already have a running instance of BDNA Discover):

   ```
   $ sh startagent.sh &
   ```

2. Start the BDNA shell by issuing the following command on the Message Bus server:

   ```
   $ sh bdna.sh
   ```

3. Launch all BDNA Discover components by issuing the following command:

   ```
   bdna> startup
   ```

The `startup` command can take around five minutes to complete, depending upon the number of components to be started and the system state. The `startup` command launches all components that are configured to start on bootstrap.

Typically, this configuration will include:

- Reporting Service
- Agenda Manager
- Two Rule Engines
- Collection Manager
- Collection Engine
- Perl Collection Service

These components represent the default BDNA Discover components.

Viewing a Running Configuration

Once the startup command has completed, it is easy to view the running configuration. From the BDNA Shell, issue the following command:

```
bdna> list
agendaManager on localhost, state = Running
bdnaCLE on localhost, state = Running
bdnaPerlCS on localhost, state = Running
CLE1 on localhost, state = Running
CLM1 on localhost, state = Running
PerlCS1 on localhost, state = Running
PERL_SERVER_localhost on localhost, state = Running
RS on localhost, state = Running
RULE0 on localhost, state = Running
RULE1 on localhost, state = Running
bdna>
```

Notice that there are a number of components listed above that have not yet been discussed, such as `bdnaCLE and bdnaPerlCS`. The functions of each of these components are discussed later. Moreover, components are not listed by their type, but rather by their name. For example, this configuration has two Rule Engines (RULE0 and RULE1) and a single Perl Collection Service (PerlCS1) running. Each component has both a type and a name. By convention, components are named by their type followed by the component number (PerlCS1, PerlCS2, etc.).

Note: For further information about components and their functions, refer to "BDNA Inventory Component Reference" on page 245.

The BDNA shell can provide additional information on the system as well. To determine what Oracle connection is being used, issue the info command:

```
bdna> info
Database: BDNA1_DB as BDNA_USER
System state: Running
Tablespace free: 37.2GB (last updated: 2008-10-03 16:01:28)
bdna>
```

Note: Many other commands for viewing running configurations exist. For further information, refer to "BDNA Shell Command Reference."

Starting and Stopping Components

Individual components can be started and stopped in a running system. Typically, this is done when a component needs configuration changes or is no longer needed for the system. Components are started and stopped by their name. For example:

```
bdna> stop PerlCS1
```

This command would stop PerlCS1. It would prevent further tasks from being scheduled to the Perl Collection Service named PerlCS1 and rollback all existing work the PerlCS is performing. As such, it may take a few minutes for the service to actually stop. In the default configuration, it would also stop all active Level 1 and UNIX Level 2 and Level 3 discovery, since PerlCS is the main worker component.

However, to restart the discovery, the Discovery administrator could issue:

```
bdna> start PerlCS1
```

BDNA Discover will pick up where it had left off.

Starting Component-Server Components at Boot

For installations that require it, the Application Agent and other BDNA Discover components can be configured to start at boot. The Red Hat Linux OS provides several conventions for doing this. The easiest method is to modify the `rc.local` file in `/etc`. The `rc.local` is used to execute scripts during the final stages of boot. This is done by adding the fully qualified path to a script in this file. To illustrate, assume there is a script located in `/home/bdna` named `startupInventory.sh`. The script contains the following lines:

```
#!/bin/bash
SLOG=$BDNA_HOME/logs/startupInventory-`date "+%Y%m%d"`.log
sh startagent.sh 1> $SLOG 2>&1 &
sleep 60
sh bdna.sh 1>> $SLOG 2>&1 << "FIN"
startup
list
exit
FIN
```

By inserting the line

```
su – bdna -c “/home/bdna/startupInventory.sh”
```

in `/etc/rc.local`, `startupInventory.sh` will execute as the BDNA user, effectively bringing up the entire BDNA system at boot.

Note: In order for the script to execute the aforementioned `su` command, use the `chmod` command to make `startupInventory.sh` executable.

Note: For distributed installations, the component server running the MBUS must always be started first.

Creating and Modifying Components

While not typically necessary or recommended for single node installations, the BDNA Discover Administrator can create and modify components. The most frequently created components are Rule Engines and Perl Collection Service components. While the default configuration has two Rule Engines, there is only a single Perl Collection Service (PerlCS). PerlCS components perform the majority of the actual data collection, so BDNA Discover Administrators often want to use more than one PerlCS to improve throughput.

To create a new PerlCS component, the BDNA Discover Administrator would issue the following command:

```
bdna> mkcomp –Dmx=536870912 –Dhost=localhost CLE CLE2
```

This command creates a new CLE component named CLE2, sets the maximum memory size to 512MB, and sets the host that will manage it to localhost.

Alternatively, the BDNA Discover Administrator could clone an existing component:

```
bdna> mkcomp –Dlike=CLE1 CLE CLE2
```

This command creates CLE2 with all the attributes of CLE1.

Running components can also be updated to reflect new parameters. For example, to increase the memory of CLE2 to 768M, the BDNA Discover Administrator could issue the command:

```
bdna> updcomp –Dmx=768M CLE2
```

Notice that the memory size parameter can be specified by either a full value or with the M megabyte modifier.

Components may also be removed from a running system. A component must be stopped prior to removing it from the system. For example:

```
bdna> stop CLE2
```

```
bdna> rmcomp CLE2
```

Starting and Stopping WinCS

The Windows Collection Service (WinCS) is the only component of BDNA Discover that must be run on a Microsoft Windows operating system. WinCS is used solely to perform Level 2 and Level 3 discovery on remote Windows systems. It is started and stopped like any other Windows service and is named *BDNA Discover Collection Service*.

WinCS connects via the Message Bus to a Collection Engine (CLE), so the CLE must be running prior to starting WinCS. WinCS must be configured with the DNS hostname or IP address of the host running the CLE to which it will connect. This value is typically set during installation, but it may be altered by editing the configuration file, WinCS.exe.config. This file is typically found in C:\Program Files\BDNA\WinCS on the system running WinCS. To change the target CLE address, open the file with a text editor and search for the string “cleHostName”. There should be a line in the file similar to the following:

```
<add key="cleHostName" value="192.168.2.100" />
```

Replace the value with the DNS hostname or IP address of the target CLE. The BDNA Discover Administrator must restart the BDNA Discover Windows Collection Service to make this change take effect.

Note: For more information on how to configure WinCS components, refer to “Configuration-File Reference.” .

Stopping BDNA Discover

Stopping BDNA Discover is a two-step process. First, all running components must be shut down, then the Message Bus and Application Agent must be stopped. To shut down all running components, issue the following command:

```
bdna> shutdown
```

This operation may take from 1 to 5 minutes, depending upon current system activity. When the command returns, the BDNA Discover administrator can run the list command to verify the action:

```
bdna> list
agendaManager on localhost, state = Init
bdnaCLE on localhost, state = Init
bdnaPerlCS on localhost, state = Init
CLE1 on localhost, state = Init
CLM1 on localhost, state = Init
PerlCS1 on localhost, state = Init
PERL_SERVER_localhost on localhost, state = Init
RS on localhost, state = Init
RULE0 on localhost, state = Init
RULE1 on localhost, state = Init
bdna>
```

All components should be in the Init state after a shutdown. To stop the Message Bus and Application Agent, the BDNA Discover Administrator should issue:

```
bdna> halt
```

```
bdna> exit
```

The `halt` command will stop the Message Bus and the Application Agent, and the `exit` command will close the BDNA Shell.

Configuring and Running Level 1 Discovery 4

About this Chapter

Configuring and running a discovery operation with BDNA Discover is the first core function of the BDNA Discover Administrator. This chapter covers the basic tasks a BDNA Discover Administrator performs to run a successful Level 1 scan such as loading networks and creating Level 1 scan tasks. This chapter also introduces the Scan Administration application, which is used to define and start scans.

Overview of Level 1 Discovery Operations

Level 1 discovery has two technical requirements:

- A set of network IP address ranges on which to discover assets
- Routable access from the BDNA Discover Collection Engines to the specified network IP address ranges

Level 1 discovery uses hundreds of ports and services across TCP, UDP, and ICMP protocols. Where firewalls are involved, the recommended configuration for routable access is to pass all traffic from the known BDNA Discover Collection Engines to the target devices in both directions.

Level 1 discovery is configured and initiated using Scan Administration in the BDNA Discover Client.

As described in "How BDNA Discover Works," , Level 1 discovery is the baseline discovery performed by BDNA Discover to inventory devices across TCP/IP networks.

To perform a Level 1 discovery, nearly all of the BDNA Discover components must be running, except for RS and WinCS. While it is not required to stop unnecessary components before beginning a Level 1 scan, BDNA Discover can be optimized for performance by running only the components required for the completion of the discovery tasks.

Note: For information on how to start and stop components, refer to "Starting and Stopping Components."

Launching the BDNA Discover Client

Before launching the BDNA Discover client, you must install a Java Runtime Environment (JRE) and configure it for the browser. (For a list of supported JRE and browser versions, refer to the *BDNA Discover 7.7.2 Release Notes*.) If the client machine is missing the proper JRE, BDNA Discover displays an error message and directs the browser to the JRE download site. The client platform requirements are documented in the *BDNA Discover 7.7.2 Installation Guide*.

Note: The BDNA Discover Client opens in safe mode if the Remote Services component is not running. In safe mode, you can login, view a list of components and their status, and start any component from the UI.

If you directed the installer to copy the template files to the Apache directory, you can launch BDNA Discover through a web browser on a client machine by opening the following URL:

http://RSHostName/

where `RSHostName` is the DNS hostname or IP address of the system running the BDNA Discover RS Component.

If you directed the installer not to copy the template files, then you need to use the fully qualified URL in order to launch BDNA Discover:

```
http://RSHostName:8080/bdna/AppConsole.jnlp
```

Note: If you access this URL directly and don't use the recommended http://hostname, you will bypass the browser and JRE checking step. Consequently, use the .jnlp URL only as a troubleshooting step. Refer to the *BDNA Discover 7.7.2 Installation Guide* for details.

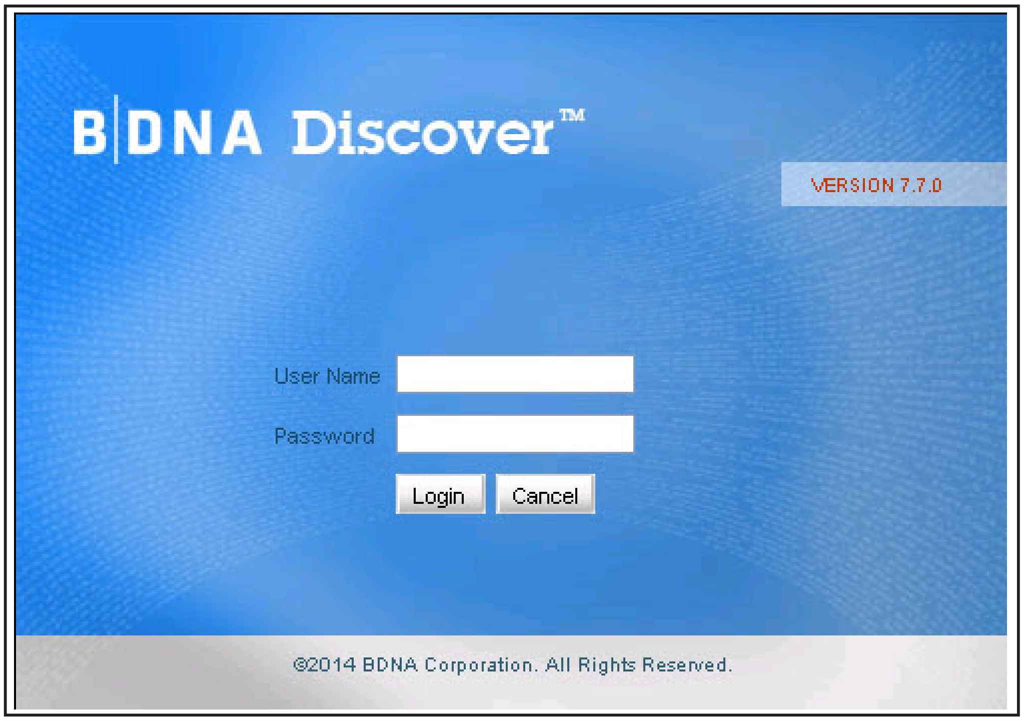
The BDNA Discover user interface is downloaded as a Java application and starts shortly after download.

Logging in to the BDNA Discover Client

The BDNA Discover login must have administrative privileges for BDNA Discover. The default BDNA Discover user "root" is a user with administrative privileges.

You must enter the user name and password you specified during execution of `configure.sh`.

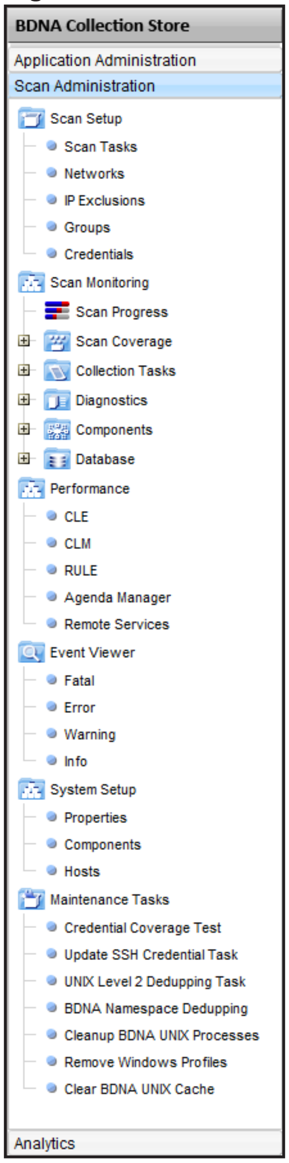
Figure 4: Login Dialog for BDNA Discover Client



Accessing Scan Administration

Scan Administration is one of the applications available through BDNA Discover. When you log in to BDNA Discover with administrative privileges—and assuming that you have also logged in to a Collection Store—the Scan Administration tab is visible on the left navigation pane of the application interface. Click Scan Administration to access the application.

Figure 5: BDNA Discover Scan Administration



Creating Level 1 Scan Tasks

Before a Level 1 scan can be executed, it is necessary to define the scan task and the target network for the scan (including any IP addresses that should be excluded from the scan).

All levels of discovery are driven by scan tasks. Scan tasks have the following three major dimensions:

- Scan level

  The level of discovery to be performed. A scan task may contain instructions to perform a scan at multiple levels of discovery, including Level 1, 2 or 3. A scan task may also specify multiple combinations of packages (Windows, UNIX or SNMP) and levels, since levels 2 and 3 support multiple types of discovery. The scan task example in this chapter is configured to perform only Level 1 discovery.

Note: For more information on discovery levels, refer to "How BDNA Discover Works".

- Networks to scan

  Target networks or groups to include for discovery. This is configured as a listing of IP address targets that this scan task should scan. The targets are defined by one or multiple Network definitions created either during scan task creation or by using Scan Administration (Scan Setup feature, Network wizard).

- Scan schedule

  When to begin and end discovery tasks. The scan task may be configured to perform discovery over general time windows (e.g. any time starting now) or for a very specific time windows (e.g. between 9:30am to 4:30pm on all weekdays during the last two weeks of the month).

Scan task creation involves many options and requires that the BDNA Discover Administrator have accurate, specific network and environment configuration data available. The remainder of this section will focus on creating a general Level 1 scan task, with the following characteristics:

- The scan task will perform Level 1 discovery on particular network(s) or group(s).
- It will start immediately.

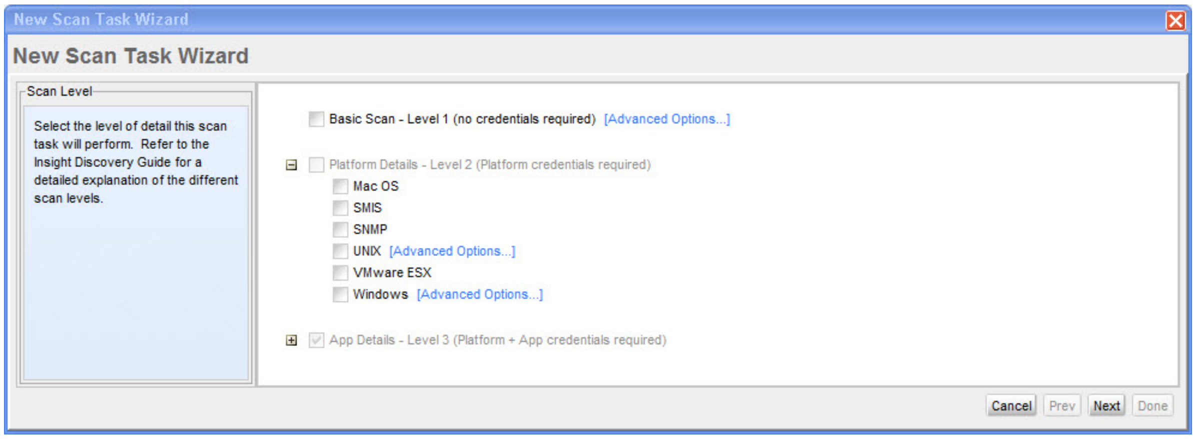
The steps in this section are intended to aid in configuring a scan task for the first time, and do not discuss all available options.

To create a Level 1 scan task:

1. Log in to BDNA Discover and select Scan Administration.
2. Under Scan Setup, click Scan Tasks.
3. In the toolbar underneath the application menu, click New Task.

   Scan Administration starts the New Task wizard.

Figure 6: New Scan Task Wizard: Scan Level



New Scan Task Wizard

Scan Level

Select the level of detail this scan task will perform. Refer to the Insight Discovery Guide for a detailed explanation of the different scan levels.

- Basic Scan - Level 1 (no credentials required) [Advanced Options...]
- Platform Details - Level 2 (Platform credentials required)
  - Mac OS
  - SMIS
  - SNMP
  - UNIX [Advanced Options...]
  - VMware ESX
  - Windows [Advanced Options...]
- App Details - Level 3 (Platform + App credentials required)

4. Specify the desired Level 1 scan option.

BDNA Discover offers several options for Level 1 scanning. For normal inventory scanning purposes, the default Complete Level 1 scan option is appropriate. Several additional Level 1 scan options are provided, however, for specific circumstances.

Light Level 1	ICMP ping scanning only. Choose this setting when you are testing the behavior of your firewalls or other network security infrastructure with respect to discovery. Some firewalls falsely report that all IP addresses behind them are active and thus distort your inventory results with spurious data. This option displays the active IP addresses in the IP ranges for the networks or logical groups that have been selected for the scan; BDNA Discover terminates discovery at this point. (If a firewall falsely reports that every IP address is active, continuing with normal discovery could generate a significant amount of activity that could impact that firewall.)
Complete Level 1 (default)	ICMP ping scanning plus OS typing. Choose this setting unless your scan fits one of the scenarios described in the other two options.
Rigorous Level 1	ICMP ping scanning and TCP SYN scanning plus OS typing. Choose this setting if you suspect that your network-based firewall, network-based security infrastructure, or host-based firewalls are blocking ICMP and thus preventing BDNA Discover from discovering active IP addresses that are in use. With this option, BDNA Discover also employs TCP SYN port scanning on a few TCP ports to discover active IP addresses that are in use.

To specify the default Basic Level 1 scan, click Next and proceed to step 5.

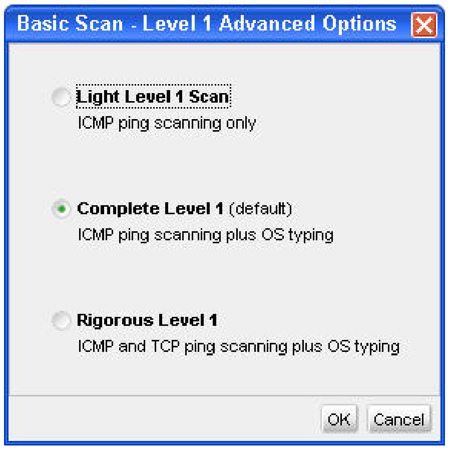
or

To specify the Light Level 1 or the Rigorous Level 1 option:

4.1. Click Advanced Options adjacent to Basic Scan - Level 1.

The Basic Scan - Level 1 Advanced Options dialog box displays.

Figure 7: Basic Scan - Level 1 Advanced Options


Basic Scan - Level 1 Advanced Options

( ) **Light Level 1 Scan**
ICMP ping scanning only

(●) **Complete Level 1** (default)
ICMP ping scanning plus OS typing

( ) **Rigorous Level 1**
ICMP and TCP ping scanning plus OS typing

4.2. Check the radio button adjacent to the desired option.

4.3. Click OK.

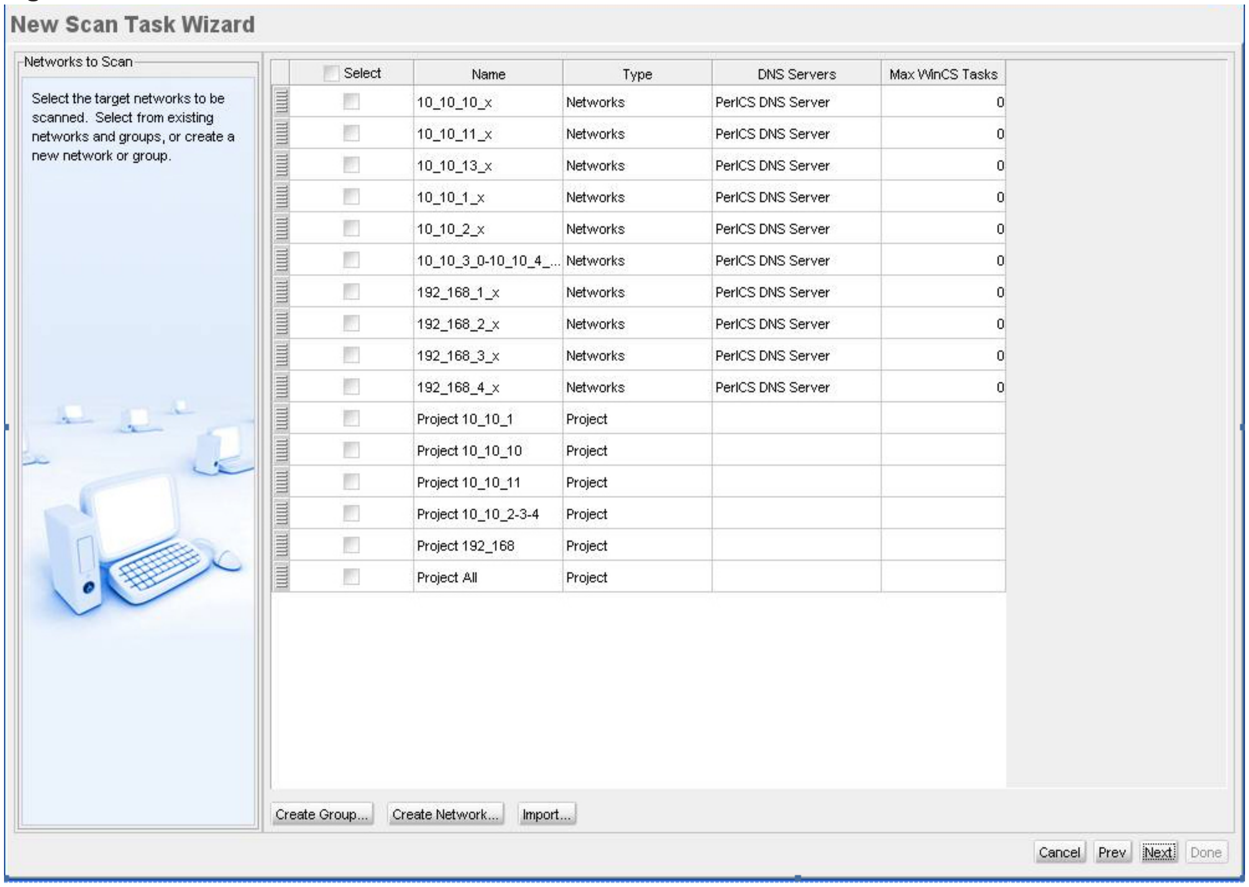
The New Scan Task wizard displays your selection.

4.4. Click Next.

5. Specify Scan Credentials: A Level 1 scan requires no special credentials, so make no changes on this page; click Next.

Scan Administration displays the Networks to Scan page of the New Scan Task wizard.

Figure 8: New Scan Task Wizard: Networks to Scan



Select the target networks to be scanned. Select from existing networks and groups, or create a new network or group.

Select	Name	Type	DNS Servers	Max WinCS Tasks
☐	10_10_10_x	Networks	PerlCS DNS Server	0
☐	10_10_11_x	Networks	PerlCS DNS Server	0
☐	10_10_13_x	Networks	PerlCS DNS Server	0
☐	10_10_1_x	Networks	PerlCS DNS Server	0
☐	10_10_2_x	Networks	PerlCS DNS Server	0
☐	10_10_3_0-10_10_4_...	Networks	PerlCS DNS Server	0
☐	192_168_1_x	Networks	PerlCS DNS Server	0
☐	192_168_2_x	Networks	PerlCS DNS Server	0
☐	192_168_3_x	Networks	PerlCS DNS Server	0
☐	192_168_4_x	Networks	PerlCS DNS Server	0
☐	Project 10_10_1	Project		
☐	Project 10_10_10	Project		
☐	Project 10_10_11	Project		
☐	Project 10_10_2-3-4	Project		
☐	Project 192_168	Project		
☐	Project All	Project		

6. Select the network(s) or group(s) that are to be targeted for discovery by this task.

If networks or groups have already been defined, associate them to this scan task by clicking on the checkbox in the Select column next to individual networks or groups.

It is also possible to select all of the available entries by clicking the checkbox next to Select in the column header. If this is the first time Scan Setup has been performed, this list is empty; in order to proceed, you must create at least one network. Clicking the Create Network... button on this page of the wizard launches the New Network wizard and then returns you to this wizard.

Note: For more information on groups and how to set them up, refer to the *BDNA Discover Administration Tutorial, Module 1*.

To create additional networks, repeat the following steps:

6.1. Click Create Network…

6.2. Follow the steps in the section "Creating New Networks with the New Network Wizard" on page 47.

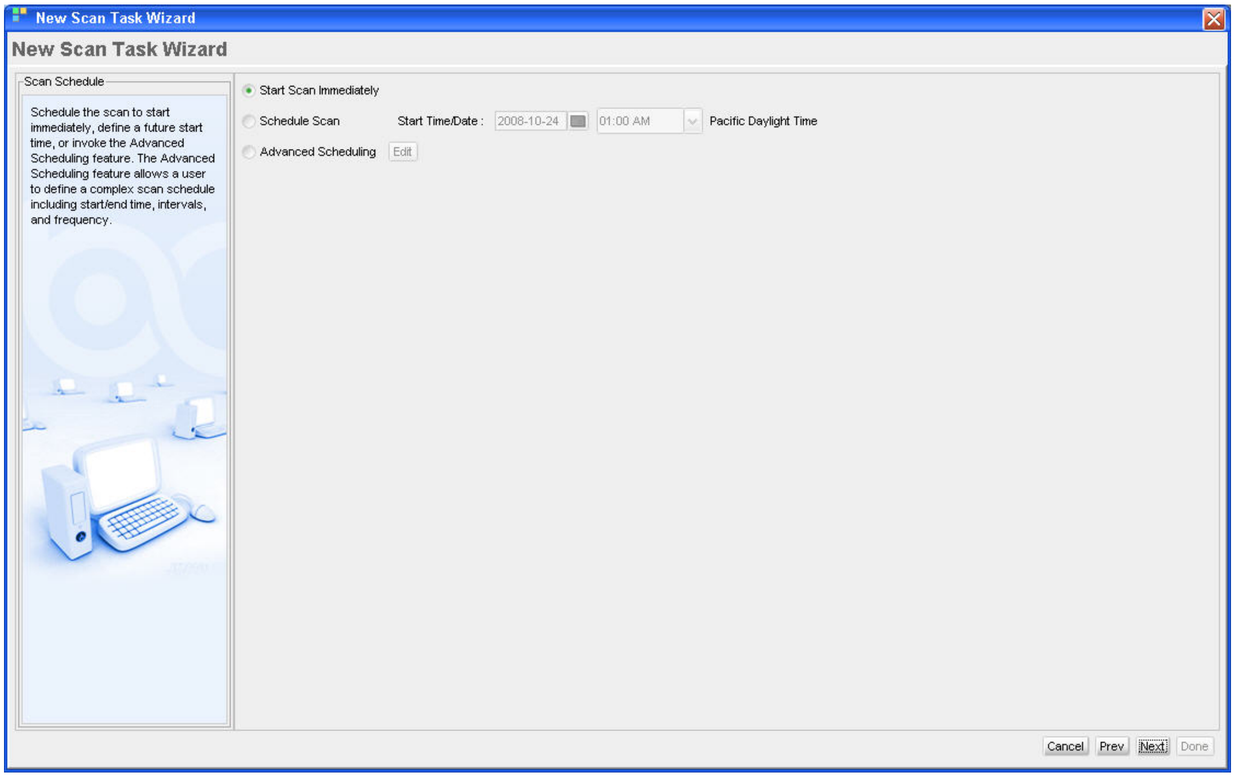
6.3. When the network has been created, click OK.

BDNA Discover saves the network definition and adds the new network to the list of available networks and groups. The new network is highlighted in the list.

7. Having selected at least one network or group, click Next.

Scan Administration displays the Scan Schedule page of the New Scan Task wizard.

Figure 9: New Scan Task Wizard: Scan Schedule



Schedule the scan to start immediately, define a future start time, or invoke the Advanced Scheduling feature. The Advanced Scheduling feature allows a user to define a complex scan schedule including start/end time, intervals, and frequency.

[✓ Start Scan Immediately]

Schedule Scan Start Time/Date : 2008-10-24 01:00 AM Pacific Daylight Time

Advanced Scheduling Edit

8. Assign a start time for the scan:

To start the scan immediately (the default setting), click Next and proceed to step 9 on page 44.

or

To schedule a single scan to start at a particular time and date, check the radio button adjacent to Schedule Scan, specify the start date and time, click Next, and proceed to step 9 on page 44.

or

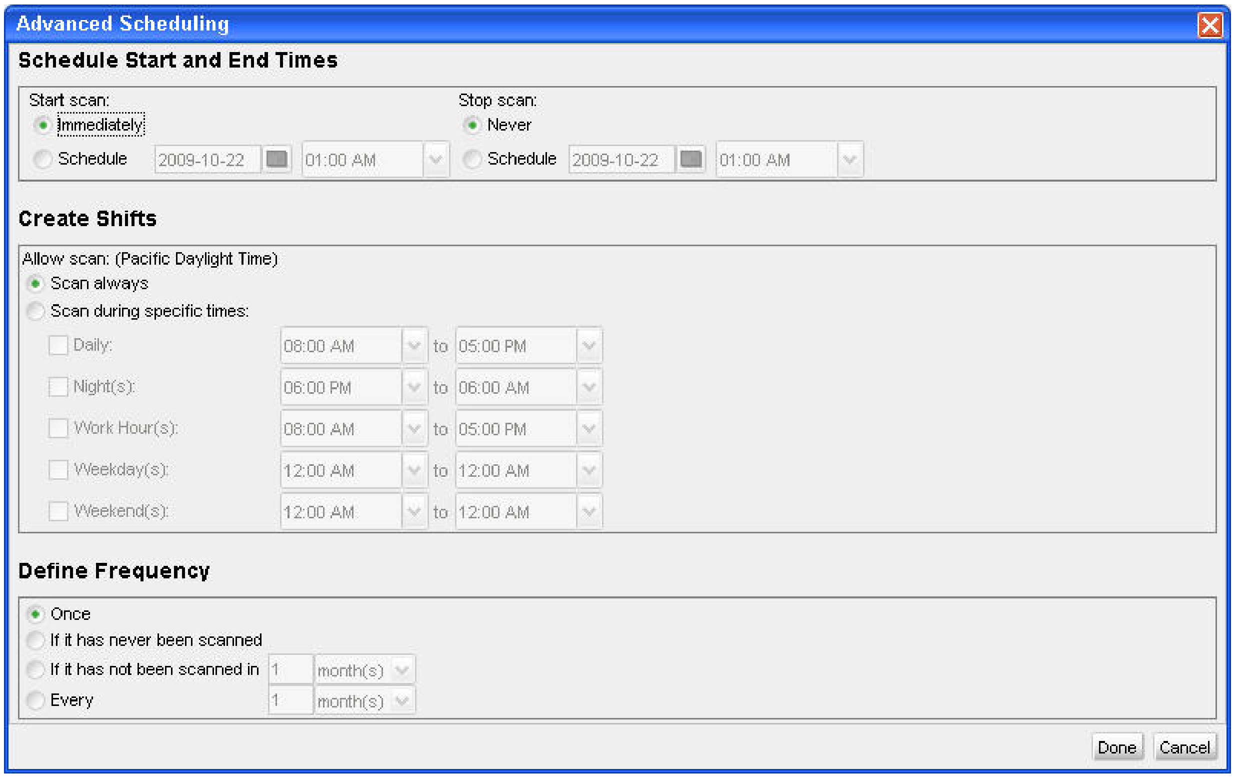
To schedule scans on a recurring basis:

8.1. Check the radio button adjacent to Advanced Scheduling.

8.2. Click Edit.

The Advanced Scheduling dialog box displays.

Figure 10: Advanced Scheduling Dialog Box


Advanced Scheduling

Schedule Start and End Times

Start scan:
[✓ Immediately]
Schedule 2009-10-22 01:00 AM

Stop scan:
[✓ Never]
Schedule 2009-10-22 01:00 AM

Create Shifts

Allow scan: (Pacific Daylight Time)
[✓ Scan always]
Scan during specific times:

Daily:	08:00 AM	to	05:00 PM
Night(s):	06:00 PM	to	06:00 AM
Work Hour(s):	08:00 AM	to	05:00 PM
Weekday(s):	12:00 AM	to	12:00 AM
Weekend(s):	12:00 AM	to	12:00 AM

Define Frequency

[✓ Once]
If it has never been scanned
If it has not been scanned in 1 month(s)
Every 1 month(s)

8.3. Assign a scan cycle that meets your needs.

Note: Be sure that you *always* specify a scan stop time.

8.4. Click Done.

8.5. Click Next.

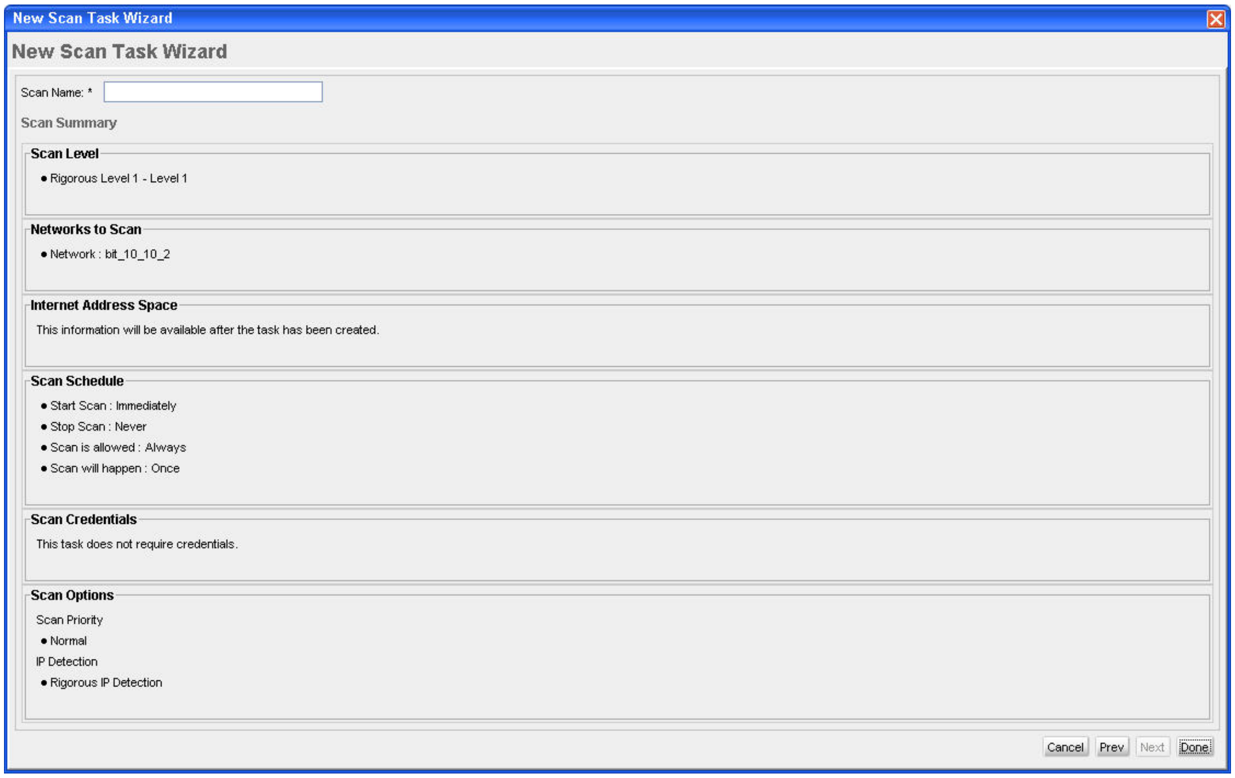
9. Specify a priority for the scan:

Normal:	Collection happens in the order in which the task was queued, but behind any scheduled high-priority task.
High:	Collection happens in the order in which the high-priority task was queued, ahead of normal-priority task.

10. Click Next.

The New Scan Task wizard displays a summary of your previous selections.

Figure 11: New Scan Task Wizard: Scan Summary Page



Scan Name: *

Scan Summary

Scan Level
- Rigorous Level 1 - Level 1

Networks to Scan
- Network : bit_10_10_2

Internet Address Space
This information will be available after the task has been created.

Scan Schedule
- Start Scan : Immediately
- Stop Scan : Never
- Scan is allowed : Always
- Scan will happen : Once

Scan Credentials
This task does not require credentials.

Scan Options
Scan Priority
- Normal

IP Detection
- Rigorous IP Detection

11. On the Scan Summary page, assign a unique and descriptive name for the new scan task.

12. Examine the overview of all of the configuration settings chosen on the previous pages of the Wizard, and make sure they are as intended; to return to a previous page and reset a configuration item, click Prev.

13. When you are satisfied with the configuration settings for the new scan task, click Done to save the scan task and execute the scan.

The new scan task appears in the Scan Tasks list along with any previously defined scan tasks. To monitor the progress of your scan follow the instructions provided in Chapter 11, “Monitoring Discovery” .

Deleting Scan Tasks

To delete a scan task:

Caution: Removing a scan task during or after discovery could result in invalid, incomplete or inconsistent information. Remove scan tasks with extreme caution.

---

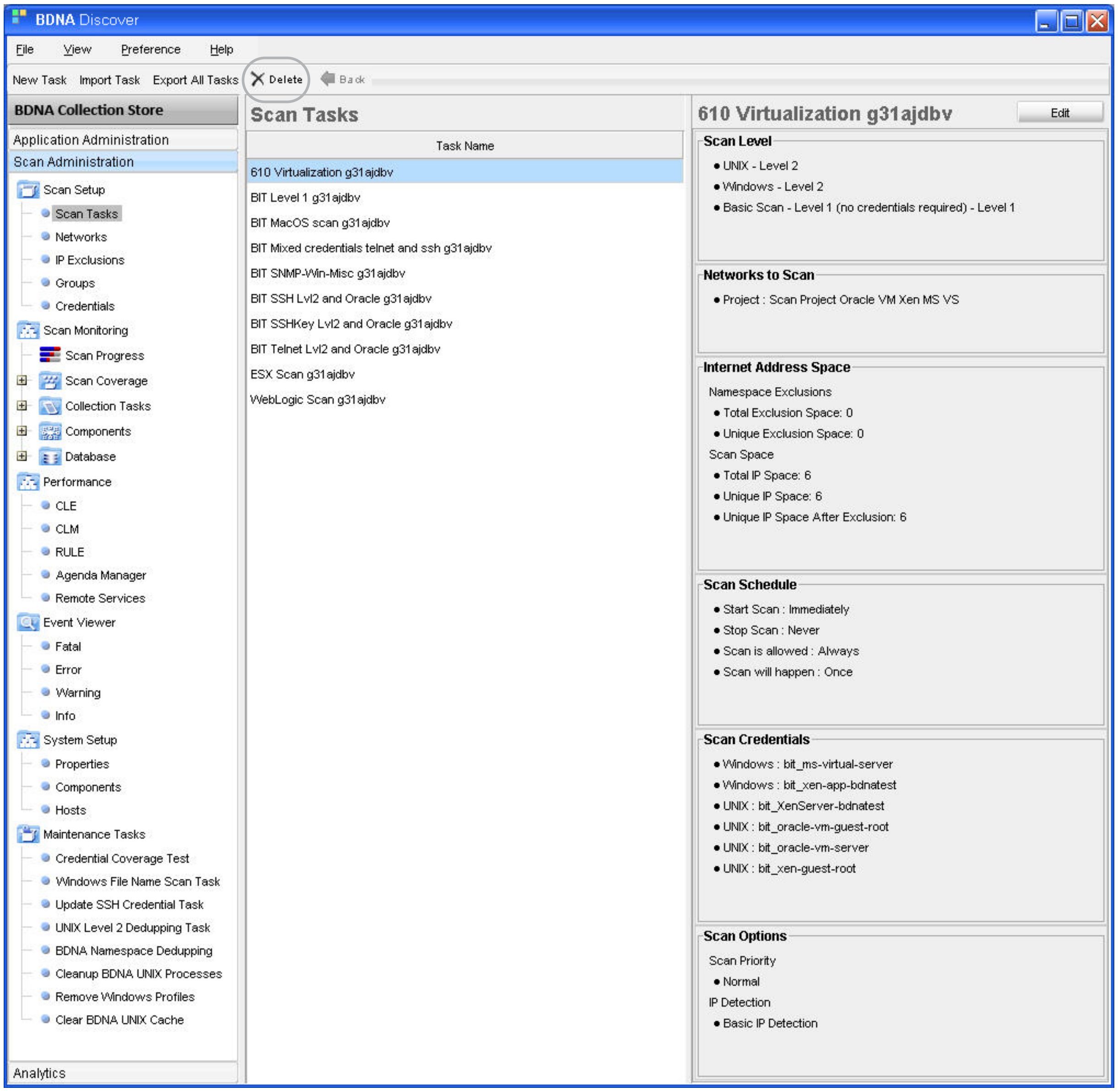
1. Log in to BDNA Discover and select Scan Administration.

2. Under Scan Setup, click Scan Tasks.

3. Highlight the scan task to delete in the Scan Tasks list by clicking on it.

4. In the toolbar under the application menu next to Import Task, click the X.

Figure 12: Delete Scan Task



Scan Tasks

Task Name
610 Virtualization g31ajdbv
BIT Level 1 g31ajdbv
BIT MacOS scan g31ajdbv
BIT Mixed credentials telnet and ssh g31ajdbv
BIT SNMP-Win-Misc g31ajdbv
BIT SSH Lvl2 and Oracle g31ajdbv
BIT SSHKey Lvl2 and Oracle g31ajdbv
BIT Telnet Lvl2 and Oracle g31ajdbv
ESX Scan g31ajdbv
WebLogic Scan g31ajdbv

610 Virtualization g31ajdbv

Scan Level
- UNIX - Level 2
- Windows - Level 2
- Basic Scan - Level 1 (no credentials required) - Level 1

Networks to Scan
- Project : Scan Project Oracle VM Xen MS VS

Internet Address Space

Namespace Exclusions
- Total Exclusion Space: 0
- Unique Exclusion Space: 0

Scan Space
- Total IP Space: 6
- Unique IP Space: 6
- Unique IP Space After Exclusion: 6

Scan Schedule
- Start Scan : Immediately
- Stop Scan : Never
- Scan is allowed : Always
- Scan will happen : Once

Scan Credentials
- Windows : bit_ms-virtual-server
- Windows : bit_xen-app-bdnatest
- UNIX : bit_XenServer-bdnatest
- UNIX : bit_oracle-vm-guest-root
- UNIX : bit_oracle-vm-server
- UNIX : bit_xen-guest-root

Scan Options

Scan Priority
- Normal

IP Detection
- Basic IP Detection

5. Click Yes when prompted.

Note: The deletion of a scan task may take some time to complete.

The scan task will no longer appear in the Scan Tasks list.

Working with Networks

Networks are groupings of IP addresses that can include IP address lists and/or ranges of IP addresses in various formats using the following guidelines:

- The IP addresses and IP ranges within a network can be contiguous (e.g. 192.168.0.0/24, 192.168.1.0/24) or non-contiguous (e.g. 192.168.0.0/24, 192.168.23.0/24).
- Individual IP addresses may be separated by commas, or each one may be entered on its own line.
- IP ranges can be entered using host address ranges (e.g. 192.168.0.1–192.168.0.255) or subnet mask notation (e.g. 192.168.0.0 & 255.255.255.0).

As many subnets as desired may be grouped into a single network definition, or they may be spread across multiple definitions. There can be a 1 to many relationship between networks and subnets. Multiple networks may be defined to aid in post-scan analysis and scan task scheduling, or all of the IP and subnet information added into a single network. Creating multiple networks is beneficial if you are generating tasks that scan multiple subnets or that are used in conjunction with Groups or Access Control (refer to Chapter 9, "Managing Groups" for more information.)

Scan Administration is used to create, modify, and remove networks. Networks may be created before scan tasks are created by manually creating the Network(s) or by Bulk Importing the Network(s). Networks may also be defined during the creation of a scan task. There is no difference between a Network created through a scan task and a Network created manually or via Bulk Import.

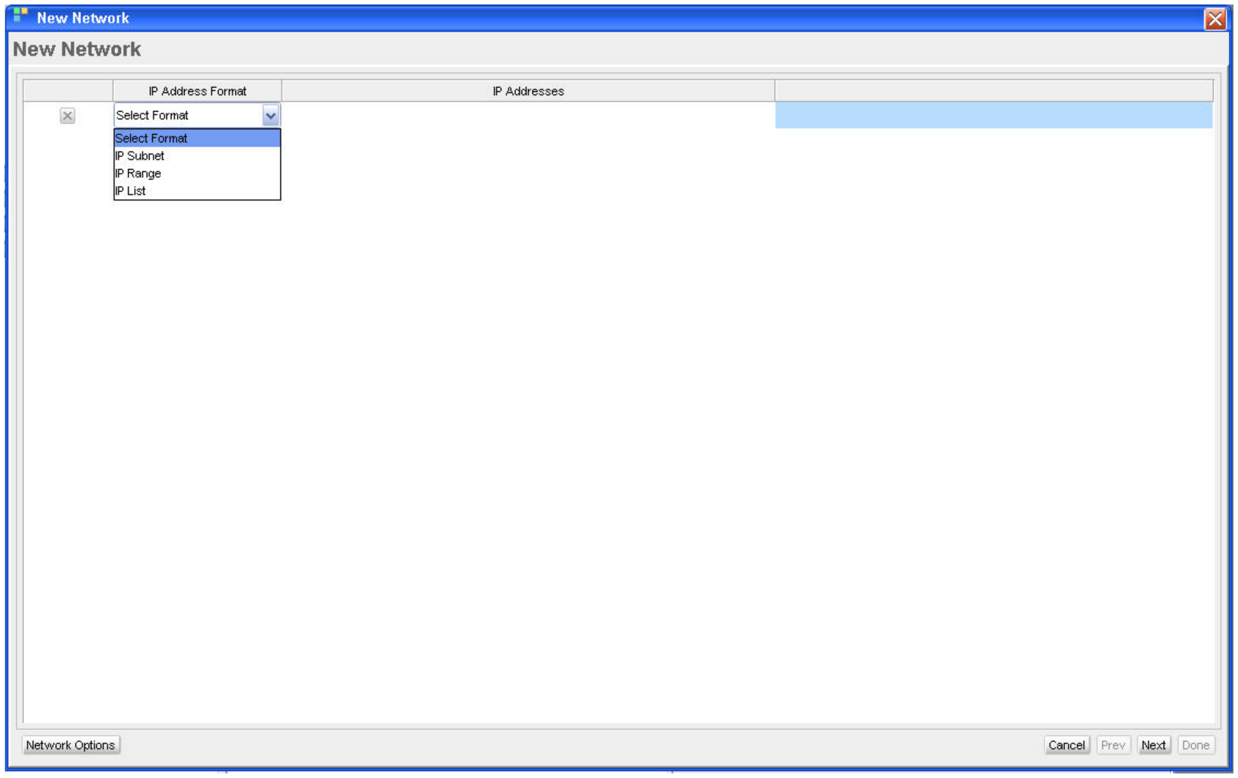
Creating New Networks with the New Network Wizard

To create a new network definition by using the New Network wizard:

1. Log in to BDNA Discover and select Scan Administration.
2. Under Scan Setup, click Networks.
3. In the toolbar under the application menu click the button New Network.

   The New Network wizard appears.

Figure 13: New Network Wizard



4. In the New Network wizard, select the IP Address Format from the Select Format drop-down list.
5. Enter the data appropriate to the type of format selected.

   Any errors in input will be immediately flagged in the Error column on the right side of the wizard when the Next button is clicked.

   5.1. An IP Subnet requires a Subnet Address (e.g. 192.168.1.0) and a Subnet Mask (e.g. 255.255.255.0).

   Some errors in input will be auto-corrected. For example, an input of 192.168.1.0 with a subnet mask of 255.255.0.0, which is invalid, would cause the Subnet Address to be auto-corrected to 192.168.0.0. If the Subnet Mask does not begin with 255.255, an error message will be generated, as subnets larger than a class B cannot be input.

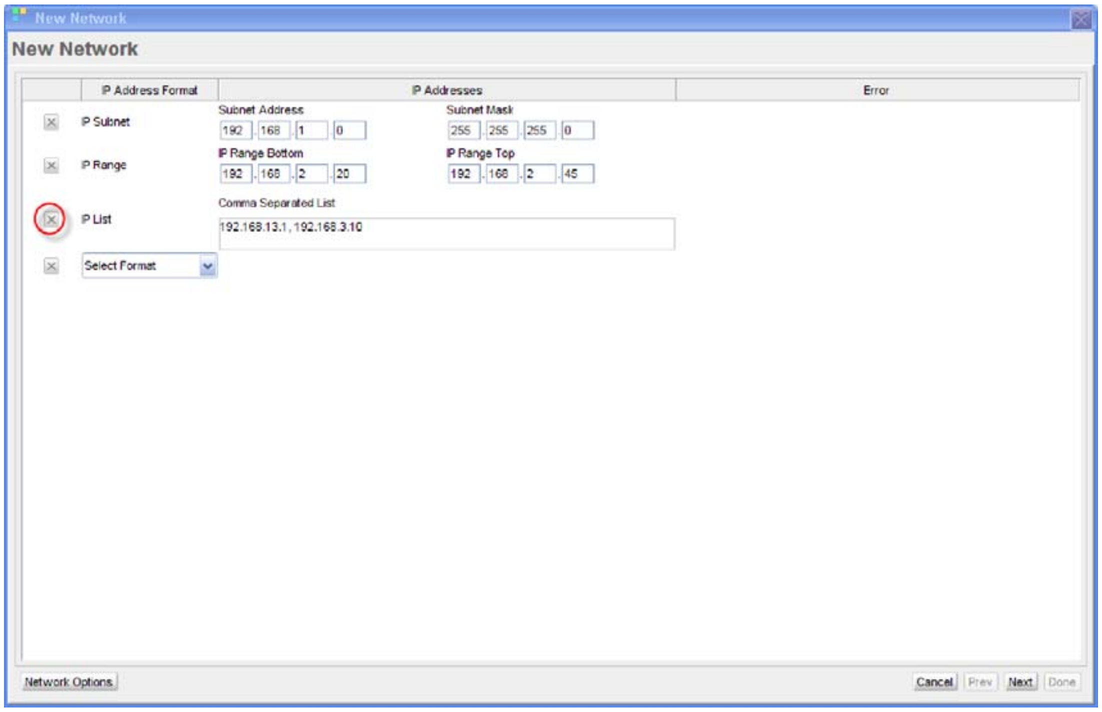
   5.2. An IP Range requires an IP Range Bottom (e.g. 192.168.2.20) and IP Range Top (e.g. 192.168.2.45).

   If the bottom is larger than the top, an error message will be generated.

   5.3. An IP List accepts multiple individual IP addresses, separated by commas (e.g. 192.168.3.1, 192.168.3.10) up to a maximum of 250 addresses.

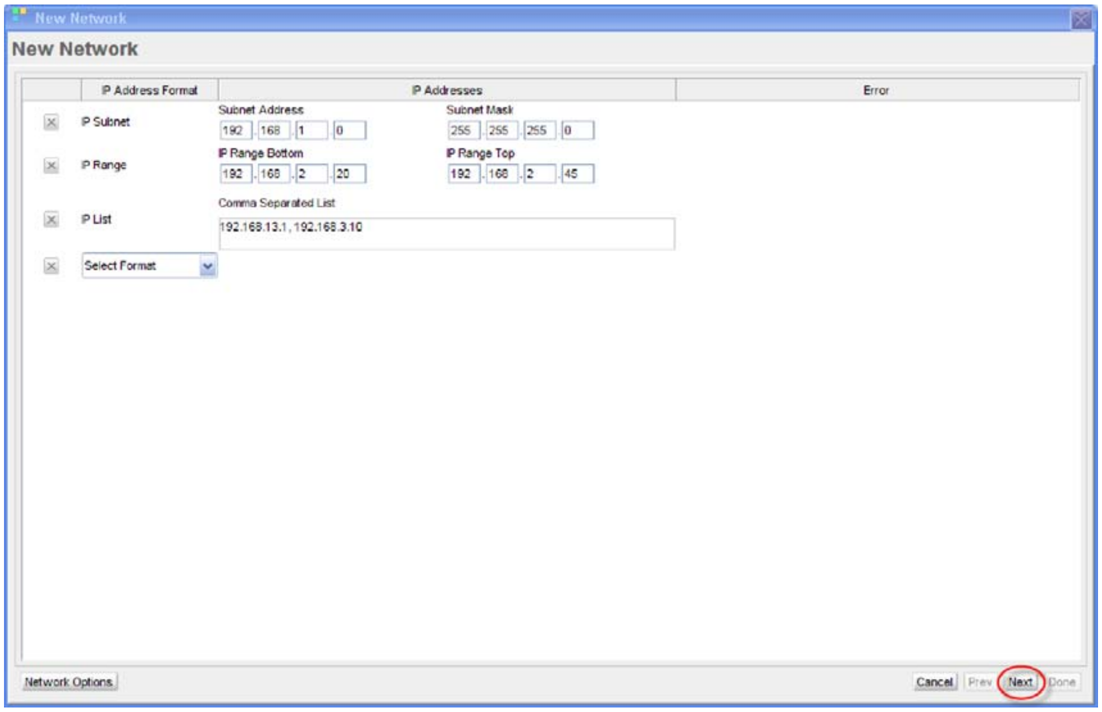
   If the list is empty or improperly formatted, an error message will be generated.
6. Keep selecting input types from the Select Format drop-down list until all of the IP information for this Network has been added; click the X next to any subnet definition to remove it from the Network definition.

Figure 14: Delete Network Entry



7. When satisfied, click Next at the bottom right of the Wizard; if any errors are detected, correct them and click Next again until all errors are resolved.

Figure 15: Create New Network Definitions



8. On the second page of the New Network wizard, give the network a unique name.

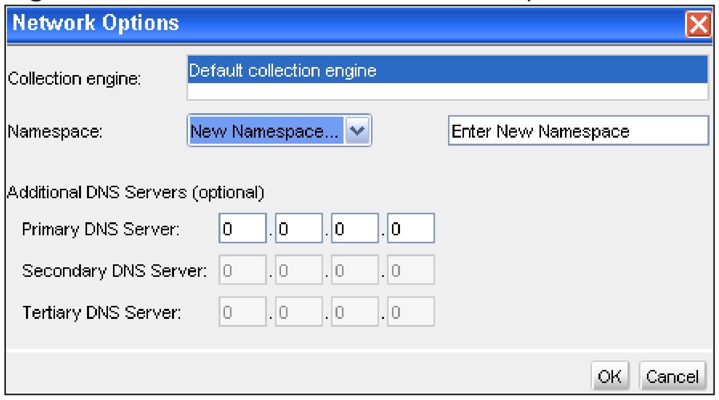
9. Use the Prev button to go to the first page in the New Network wizard at any time.
10. Optional: On either page of the New Network wizard, click the button Network Options in the bottom left corner of the Wizard page to assign which CLE(s) and which Namespace to use for the new network.
11. Specify a Namespace:

    Select an existing Namespace from the drop list.

    or

    Select New Namespace and enter a name for the new Namespace that is to be defined.

Figure 16: Create New Network: Network Options Details


Collection engine: Default collection engine

Namespace: New Namespace... Enter New Namespace

Additional DNS Servers (optional)

Primary DNS Server: 0 . 0 . 0 . 0

Secondary DNS Server: 0 . 0 . 0 . 0

Tertiary DNS Server: 0 . 0 . 0 . 0

Note: Namespaces are used to handle the rare situation where different machines in a single scan may have the same IP addresses. For example, an organization may have two organizational departments, both of which utilize the non-routable 192.168.x.x IP space. (Granting routable access to these otherwise non-routable subnets is beyond the scope of this document.) In the vast majority of cases, the default namespace Global Namespace should be selected.

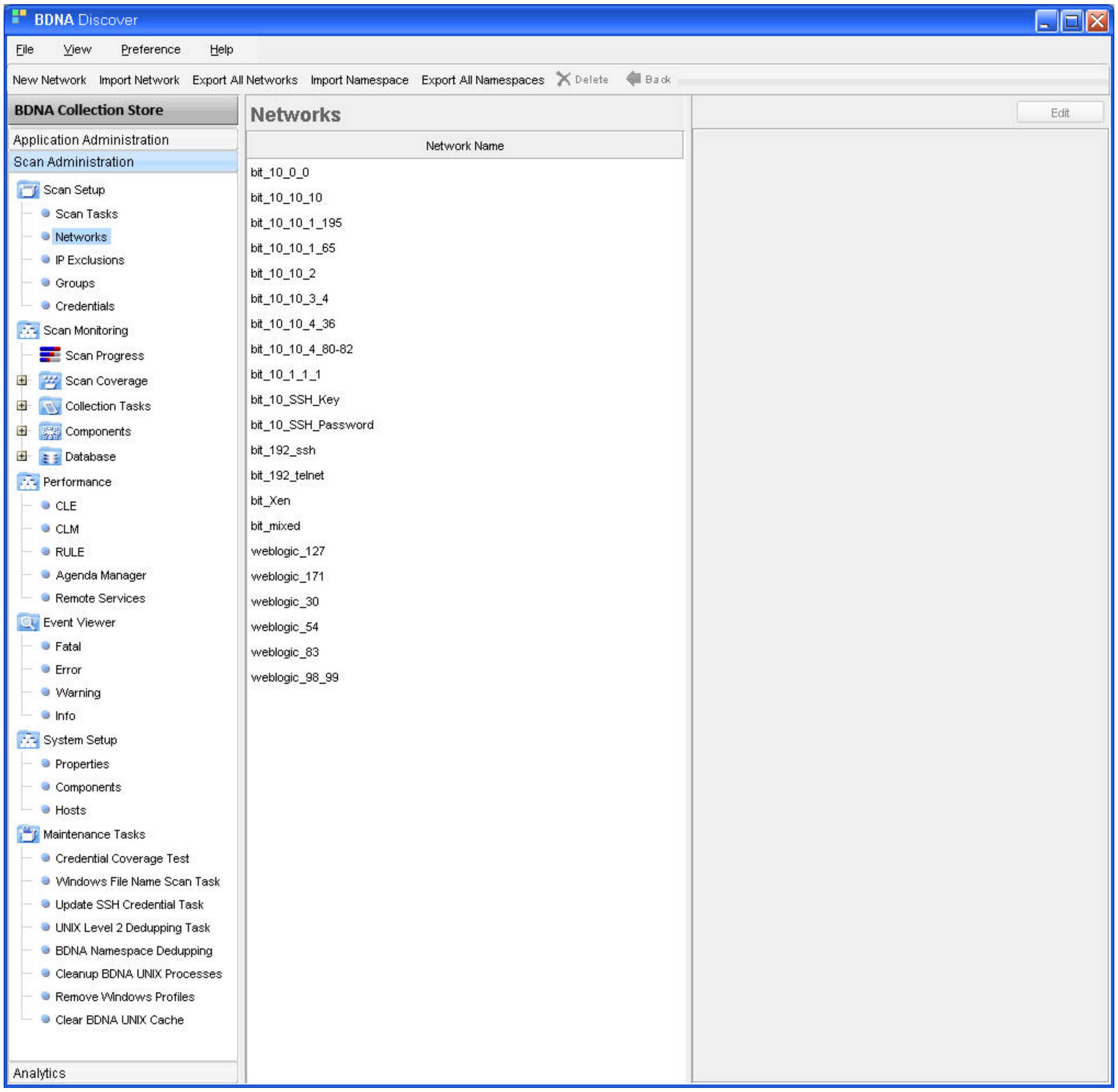
12. Optionally, define the DNS servers to use to resolve the hostname of the IP addresses.

    You can specify up to three DNS servers to use. When resolving the hostname, the tool uses the DNS servers specified in the order shown.

    – If the tool resolves the hostname, the DNS server used displays in the Networks to Scan screen.

    – If no DNS servers are specified or after trying all the DNS servers it does not find the hostname, the tools uses the PerlCS DNS server. This is the Linux server running PerlCS.
13. When both pages have been satisfactorily completed, click Done in the bottom right corner of the Wizard.
14. The new network appears in the Networks listing, which is listed alphabetically by Network Name.

Figure 17: Create New Network Entry



Network Name
bit_10_0_0
bit_10_10_10
bit_10_10_1_195
bit_10_10_1_65
bit_10_10_2
bit_10_10_3_4
bit_10_10_4_36
bit_10_10_4_80-82
bit_10_1_1_1
bit_10_SSH_Key
bit_10_SSH_Password
bit_192_ssh
bit_192_telnet
bit_Xen
bit_mixed
weblogic_127
weblogic_171
weblogic_30
weblogic_54
weblogic_83
weblogic_98_99

Modifying Networks

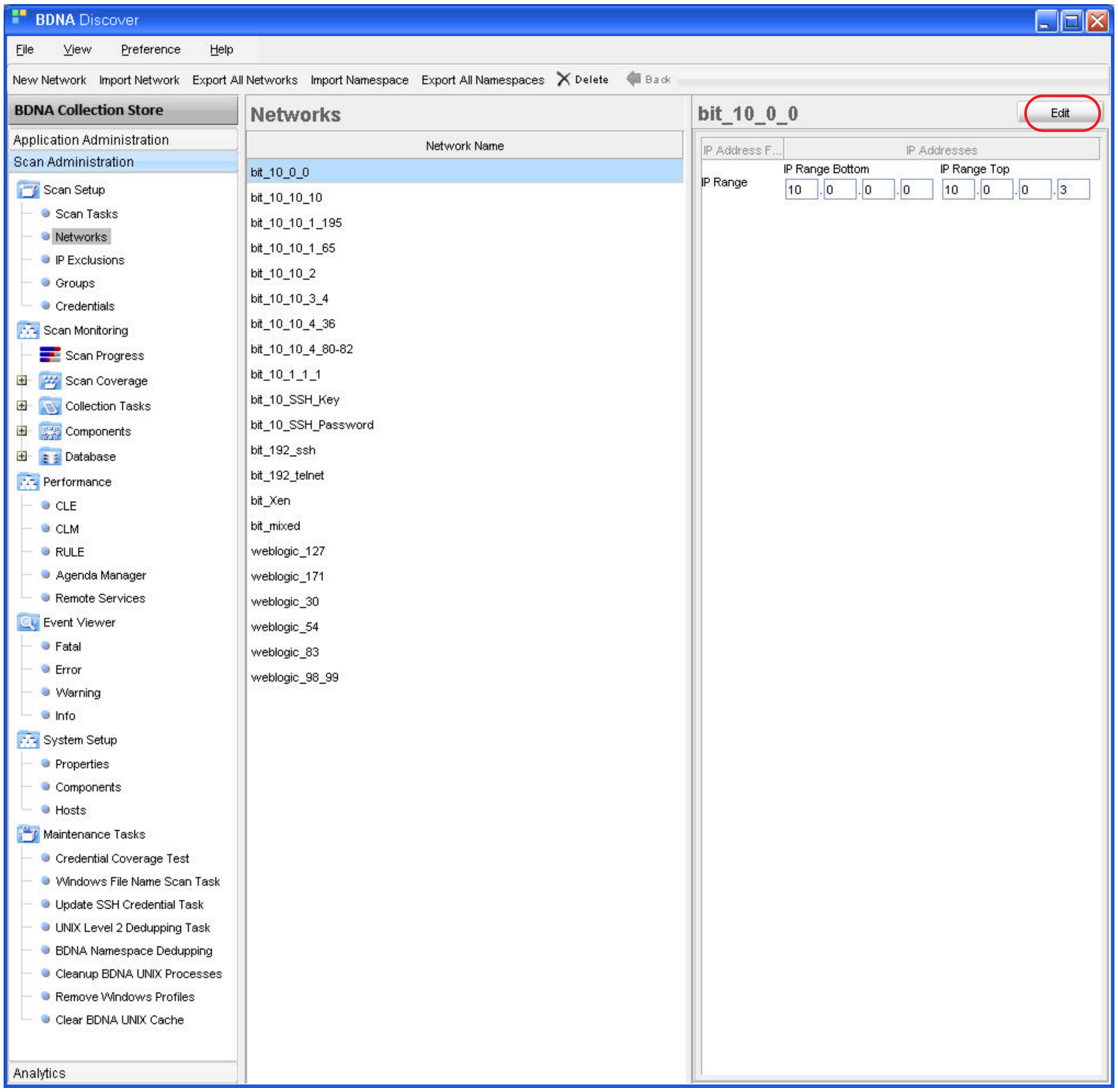
The following steps may be used to modify an existing Network that was created through the New Network wizard, through Network Bulk Import, or through scan task creation. No matter how a Network was created the following steps may always be used to edit an existing Network definition.

1. Log in to BDNA Discover and select Scan Administration.
2. Under Scan Setup, click Networks.
3. Highlight the network to be modified in the Networks list by left-clicking on it.

   The details of the selected network appear in the right-hand side of the window.

4. Click Edit at the top right of the details pane.

Figure 18: Modify Network Entry



Network Name
bit_10_0_0
bit_10_10_10
bit_10_10_1_195
bit_10_10_1_65
bit_10_10_2
bit_10_10_3_4
bit_10_10_4_36
bit_10_10_4_80-82
bit_10_1_1_1
bit_10_SSH_Key
bit_10_SSH_Password
bit_192_ssh
bit_192_telnet
bit_Xen
bit_mixed
weblogic_127
weblogic_171
weblogic_30
weblogic_54
weblogic_83
weblogic_98_99

IP Address F...	IP Addresses	
	IP Range Bottom	IP Range Top
IP Range	10 .0 .0 .0	10 .0 .0 .3

5. Add, modify and delete subnets using the Network wizard as detailed in the Creating a New Network Using the New Network wizard section above.

6. When satisfied with the changes, click Done on either page of the Wizard.

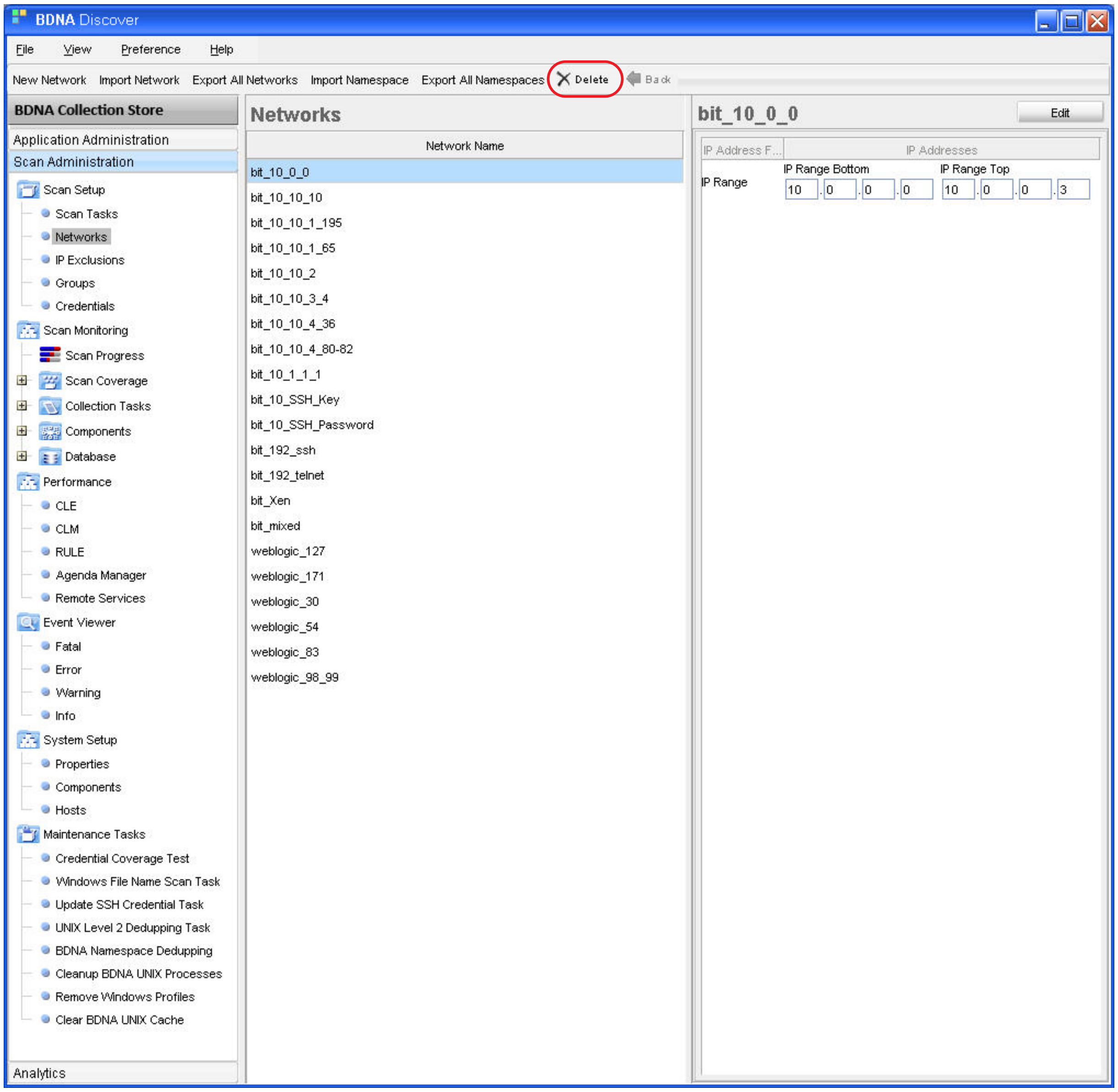
Deleting Networks

The following steps may be used to delete a Network. A Network's IP ranges may also be deleted via Bulk import; however the network, itself, must be deleted through the procedure, below.

Caution: Removing a network or modifying its IP ranges after discovery could result in invalid, incomplete or inconsistent information. Remove or modify networks with extreme caution in cases where the target Network has been used in a scan task that has already been run or is in the process of being run.

1. Log in to BDNA Discover and select Scan Administration.
2. Under Scan Setup, click Networks.
3. To highlight the network to delete in the Networks, left-click on that network.
4. In the application menu, click Delete (X).

Figure 19: Delete Network Entry



Scan Administration displays a request for confirmation.

5. To delete the network, click Yes.

The network disappears from the Networks list.

Creating IP Exclusions (Optional)

In some cases it may be desirable to prevent certain IP addresses and ranges from being scanned at any level. BDNA Discover supports such exclusions through the use of IP Exclusion definitions under the Scan Setup section. IP Exclusion is defined in much the same manner as Networks are defined, with the main difference that IP Exclusion definition is performed with a Namespace as the starting point for the IP Exclusion list.

Note: The system ignores Excluded IP Ranges, even if those ranges are explicitly included within scan ranges targeted for collection.

Note: Bulk importing an exclusion list will eliminate the previous exclusion ranges.

To exclude an IP address from scan operations for an existing Namespace:

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup, click IP Exclusions.
3. If no new Namespaces have been defined, the default Namespace (Global Namespace) will be automatically selected; if multiple Namespaces are listed, select the desired Namespace from the Namespace column.
4. Click Edit, in the upper right corner of the page.

   Scan Administration displays the Edit Namespace dialog box.
5. Use the Select Format drop-down list to select the desired IP Address Format.
6. Enter data appropriate to the type of format selected, as shown below; when finished, click Done.

   Any errors in input are immediately flagged in the Error column on the right side of the Wizard. A list of possible errors follows:

   – IP Subnet

     Requires a Subnet Address (e.g. 192.168.1.0) and a Subnet Mask (e.g. 255.255.255.0). Some errors in input are auto-corrected; for example, an input of 192.168.1.0 with a subnet mask of 255.255.0.0, which is invalid, would cause the Subnet Address to be auto-corrected to 192.168.0.0. If the Subnet Mask does not begin with 255.255, an error message will be generated as subnets larger than a class B cannot be input.

   – IP Range

     Requires an IP Range Bottom (e.g. 192.168.2.20) and IP Range Top (e.g. 192.168.2.45). If the bottom is larger than the top, this will generate an error message.

   – IP List

     Accepts multiple individual IP addresses, separated by a comma (e.g. 192.168.3.1, 192.168.3.10). If the list is empty or badly formatted the IP Exclusions Wizard will generate an error message.
7. Keep selecting input types from the Select Format drop-down list until all the IP information required for this Namespace exclusion list has been added; to remove a subnet definition from the exclusion list, click the X next to that definition.
8. When finished, click Done; if Scan Administration detects any errors, correct them and click Done again.

Troubleshooting Level 1 Scanning: Firewalls

Level 1 Scanning of Oracle VM and XenServer Systems

By default, Oracle VM and XenServer systems have a firewall installed and running. This firewall causes Level 1 discovery operations to run slowly or fail to detect the host at all. To remedy this issue, configure the firewall on these systems to allow all traffic from BDNA Discover Collection Servers. Oracle VM and XenServer both use `iptables`, so to address this issue, insert the following rule:

```
iptables -I INPUT -s <csserver> -j ACCEPT
```

where `<csserver>` is the IP address of the machine running BDNA Discover PerlCS.

Refer to "Requirements for Oracle VM Discovery" on page 62 for Level 2 Oracle VM scanning considerations.

Level 1 Scanning of Windows 7 Systems

In order to type Windows 7 at Level 1, the Printer and File Sharing feature must be enabled. On Windows 7, by default, Windows firewall is enabled but the Printer and File Sharing feature is *not* enabled.

To enable the Printer and File Sharing feature in Windows 7:

1. Open the Control Panel.
2. Under the Network and Internet section, click View network status and tasks.
3. In the left pane, click Windows Firewall.
4. In the left pane, click Allow a program or feature through Windows Firewall.
5. In Allowed Programs dialog, click Change settings button.
6. Select the File and Printer Sharing option from the list.

Depending on how you configured the network location, you may check either Home/Work (Private), Public, or both.

Configuring and Running Level 2 Discovery 5

About this Chapter

This chapter provides detailed information about Level 2 discovery. Level 2 discovery provides detailed system and software information. Specifically, BDNA Discover captures:

- System configuration (number of CPUs, memory size, machine type, etc.)
- Serial numbers when available
- Local attached storage
- File system layout
- Operating system version
- Operating system patch level
- Installed patches, and more

BDNA Discover also verifies attributes collected during Level 1 Discovery and provide additional detail where appropriate. BDNA Discover also collects information about installed software packages during this phase of discovery. BDNA Corporation develops and maintains software fingerprints for finding and collecting attributes about key installed software packages.

Note: Due to their sensitive nature, BDNA Discover's default behavior is to hide Microsoft Operating System and Microsoft Office application license keys discovered during a Level 2 scan. For information about how to enable display of Microsoft Office and Microsoft Operating System license key information, refer to "Enabling the Display of Microsoft Office/Microsoft OS License Keys".

Routable Access Requirements

BDNA Discover must have routable access to every machine on which Level 2 discovery is to be performed. Network routers and firewalls must allow BDNA Discover to connect to the target machines via the appropriate protocol. The target machines themselves must also allow BDNA Discover to connect. Note that because different protocols are used during Level 2 as opposed to Level 1, it is possible to have Level 1 discovery working while Level 2 discovery is failing.

SNMP Level 2 Access Requirements

The network's enterprise routers must allow SNMP traffic from the BDNA Discover collection servers. Sometimes routers restrict SNMP traffic originating from a specific set of IP addresses or a range of IP addresses. The IP addresses assigned to the BDNA Discover collection servers must be in the allowed range.

UNIX Level 2 Access Requirements

The target host must be running a BDNA Discover-supported network remote login service, either Telnet or SSH.

The remote login service must be the same protocol as specified by the type of UNIX Level 2 credential to be created. For instance, it would be futile for BDNA Discover to try UNIX Level 2 discovery with only a Telnet credential, if the vast majority of the organization's UNIX systems are only running SSH and no longer running Telnet.

Tip: Although BDNA Discover can employ Telnet for Level 2 connections, for security reasons it is generally not used.

Target hosts running TCP wrappers or employing some other firewall-like mechanism must allow connections from BDNA Discover collection servers.

Windows Level 2 Access Requirements

- The target host must be running the Windows DCOM and WMI software services.
- Target hosts running personal or server-local firewalls must allow connections from BDNA Discover collection servers.

If the firewall is configured to make the Windows system sufficiently difficult to access, BDNA Discover will not be able to find the system during Level 1 discovery, and Level 2 discovery then cannot occur. Such systems, however, are very difficult to manage at many other levels on a corporate network as well. The BDNA Discover Administrator, in cooperation with the System Administrator(s), should perform testing to determine the most secure way to allow BDNA Discover access to the machines to be scanned.

One possibility is to make the necessary configuration changes at the domain level for the Windows XP/Windows 2003 Windows Firewall by creating a Windows domain policy. Such a domain policy would be applied to the Windows XP/Windows 2003 systems which are joined to that corporate domain. The domain policy would be created in order to enable remote administration functionality.

It is unknown whether a user with local administrator privileges would be able to override such a domain policy setting. It is unknown whether analogous domain-level policies exist for configuring corporate editions of other Windows firewall products, such as Symantec Norton Firewall or McAfee Firewall. The System Administrator(s) can help the BDNA Discover Administrator make these determinations for the organization's IT environment and choose the best available solution.

Windows XP SP2 is a slightly special case in that unlike Windows XP RTM or Windows XP SP1, the Windows Firewall is enabled during installation by default. Previously, the Windows Firewall was disabled by default during installation. However, the default setting during installation may not matter too much if the organization has processes in place (e.g., imaging) for standardizing how they install Windows XP SP2 systems (i.e. processes that turn the firewall back off anyway).

A user with local administrator privileges could conceivably install arbitrary personal firewall software that does not allow overriding the local configuration setting at the Windows domain level.

Caution: WMI (used for Windows Level 2) uses dynamically allocated ports above 1024 after the initial handshake. It is non-trivial to configure firewalls to use port-based filtering and still allow WMI traffic correctly. Instead, it is recommended that firewall rule sets allow all by source IP addresses. Network engineers at BDNA Discover installations who have been uncomfortable making such a change to their firewalls have either routed BDNA Discover's traffic physically through another path, bypassing the firewall altogether, or have virtually bypassed the firewall by tunneling through it with some form of tunneling protocol.

Working with Credentials

In order to perform Level 2 collection, BDNA Discover requires one or more credentials for the devices to be discovered. A credential is typically an operating system username and password for a server or desktop system (UNIX or Windows), but could also be a non-public SNMP community string for a switch or router.

For BDNA Discover to perform Level 2 discovery, it needs to be able to connect, authenticate, and have proper authorization. In other words, for each targeted host, BDNA Discover collection servers must:

- Be able to connect to it using the protocols appropriate to that host
- Provide the applicable credential
- Have the authorization to undertake activity necessary for Level 2 discovery

General Credential Requirements

- Password Change: For any username/password based credential type, the BDNA Discover user must not be forced to change the password on first login.
- Account Expiry: Account should not expire during a discovery window.
- Activation Propagation: If it is desirable to activate/deactivate the credential (e.g., for a domain or other centrally managed credential type), activation propagation time must be factored into scan scheduling.

Mac OS Credential Requirements

The Mac OS credential should be for an account on the target hosts. The account does not need to have administrative privileges. To address Level 2 connection requirements, the target host must have Remote Login enabled to allow SSH connections with the credential.

SNMP Credential Requirements

A community string other than *public* will satisfy the SNMP credential requirements. Because the *public* community string is tried automatically, you do not need to specify it.

UNIX Credential Requirements

- The UNIX credentials supplied for Level 2 access must authorize BDNA Discover to log in and get a normal interactive user account with a UNIX command line shell.
- The user account must be able to run standard UNIX commands such as `find`, `uname`, `ps`, `ifconfig`, `hostname`, etc.

  BDNA Discover will use these commands to query the host during Level 2 discovery.
- The account should also have the privilege to traverse the directories of target machines where BDNA Discover is performing collection.
- The account should allow multiple simultaneous logins on the same target system.

  If not, BDNA Discover can be configured to support serialized logins.

SSH public/private key authentication is the preferred authentication mechanism for UNIX Level 2. For public/private key authentication, a private key/public key pair is generated, and the public key is distributed to the UNIX Level 2 targeted hosts. The BDNA Discover collection servers can prove to the SSH server running on the targeted hosts that they possess the matching private key and therefore should be allowed to log in, because both the BDNA Discover collection server and the targeted host can encrypt and decrypt information sent between them.

The alternative is username/password based authentication. The problem with username/password authentication is that a rogue machine could theoretically be on the network running a compromised version of the SSH service. When BDNA Discover presents its username/password, the rogue machine could simply capture the username and password and make them available to whoever controls it. With public/private keys, there is nothing for the rogue machine to capture that compromises the login account's security. Although rogue machine may have the public key, it would never be sent the matching private key by the BDNA Discover collection server.

The main drawback of the public/private key mechanism is that the public key must be distributed to all targeted UNIX hosts. BDNA Discover can help with this task. The list of hosts running SSH discovered at Level 1 could be exported and used to generate support request tickets to install the public key; and with iterative scanning, BDNA Discover can track the progress of the public key rollout. But it can still be a daunting task to roll out SSH public keys to a large number of hosts. Many customers stay with SSH username/password authentication because of this issue.

VMware ESX Credential Requirements

ESX credentials are required for Level 2 discovery on VMware ESX systems. As with other Level 2 credentials, this feature requires minimum level of access (read only), though collection will work fine with high-level access as well.

ESX Level 2 discovery uses the web services call on target ESX systems. The ESX Level 2 credential requires a user with read-only permission on the ESX Server, the port on which the web services are listening, and protocol configured on the ESX for the web service.

The protocol value by default is set to https (recommended by VMware) and the port by default is set to 443, the default port for the https application.

Windows Credential Requirements

BDNA Discover uses WMI (which in turn uses DCOM) to access Windows machines targeted for Level 2 discovery. Therefore, the Windows credential user account must have sufficient WMI and DCOM permission to access the target machines remotely. The account must also have read permission on the registry key HKEY_LOCAL_MACHINE\SOFTWARE of the target machines.

By default, only Windows users with Administrator privileges (either Local Administrators or Domain Administrators) have sufficient privileges for a BDNA Discover Windows Level 2 scan. However, non-Administrator users can be granted the appropriate privileges on a machine-by-machine basis.

Customers who already maintain standard desktop and server Windows OS images will find that they have more flexibility in how they deploy credentialed access for BDNA Discover Windows Level 2 discovery, because they can make the necessary configuration changes to the Windows systems from which they are preparing their standard desktop and server Windows OS images.

Windows Local Administrators

Credentials that grant BDNA Discover Local Administrator privileges on the targeted Windows system will immediately grant the necessary authorization to use DCOM, execute WMI methods, and access the Windows registry. Therefore, any Local Administrator account would be sufficient for BDNA Discover Windows Level 2 scanning.

Local Administrator privileges can be granted through indirection, such as when the Domain Administrators group is automatically added to the Local Administrators group when the Windows machine joins the domain (see below).

Tip: Active Directory GPO (Group Policy Objects) allows the Active Directory administrator to easily add groups or users to the Local Administrators group, in a scalable fashion for even very large Active Directory domains.

Windows Domain Administrators

Windows Domain Administrators are indirectly granted Local Administrator privileges on a machine when it joins the domain (see immediately above). Therefore, any Domain Administrator account would be sufficient for BDNA Discover Windows Level 2 discovery.

In practice, Domain Administrator accounts are the easiest to implement from BDNA Discover's perspective. However, network administrators often balk at the security implications of granting BDNA Discover Domain Administrator rights. While this is a classic case of balancing security versus convenience, the following can be done to mitigate security risks:

- Rather than using an existing account belonging to the Domain Administrators group, a new dedicated domain account is created for BDNA Discover.

  This provides improved auditing and the ability to enable/disable the account without affecting existing non-Discover users.
- The default state is for the domain account to be disabled when there is no discovery activity scheduled.
- Taking into account the typical domain server propagation delays of a small number of hours, the domain account is enabled with a new password just before the scheduled discovery activity window.

  Subsequent to the scheduled discovery activity window, the domain account is again disabled.

Windows non-Administrator Accounts

As stated above, Local Administrator accounts by default grant the appropriate privileges for BDNA Discover Level 2 discovery. It is also possible to grant the appropriate privileges to a non-Administrator account. However, such privileges must be granted on a machine-by-machine basis, and this might preclude non-Administrator accounts from being used for large-scale BDNA Discover Windows Level 2 discovery.

Note: Due to the underlying technical requirements to collect such additional data, some fingerprints require Administrator credentials.

Caution: The necessary access cannot be granted by any local group that has a level of privileges less than the local Administrators group. (This includes the Backup Operators group.) In fact, it is not possible to grant the appropriate privileges at the Domain level. Instead, they must be granted on a machine-by-machine basis.

Additional Client-Configuration Requirements for Windows 7, Vista, and 2008

With Windows 7, Vista, and Windows Server 2008, the User Account Control (UAC) feature prevents remote access to Windows Management Instrumentation (WMI). When UAC is enabled, all processes that require administrative privileges must be manually elevated through a dialog box, regardless of whether they were started by a standard or administrative account. Discover collection scripts require elevated privileges to access WMI, but since they are run remotely, there is no dialog box to click, which causes the scripts to fail.

Any one of the following three options will address this issue:

- Adjust the security on the account to access WMI without requiring elevation

  For details on this step, refer to http://msdn.microsoft.com/en-us/library/aa826699%28VS.85%29.aspx.

- Disable UAC for all remote connections

  You can disable remote UAC through the registry key `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\system\LocalAccountTokenFilterPolicy`. Refer to http://msdn.microsoft.com/en-us/library/aa826699%28VS.85%29.aspx for more information.

- Disable UAC completely

The BDNA Discover Windows Rollout tools can be used to change these settings.

Requirements for Oracle VM Discovery

Identification of virtual machines (guests) from an Oracle VM Server is currently possible by reading the configuration files on the host machine. You can read configuration files in one of two ways:

- By reading the `vm.cfg` files present in the Oracle VM Server installation directory structure (`/OVS/running_pool/`)
- By using the `xm` command, if the standard configuration file is not found in the installation directory path

  The `xm` command is the main command-line interface for managing Xen-based hypervisor's guest domains. You must be a privileged user in order to execute `xm` commands.

Refer to "Level 1 Scanning of Oracle VM and XenServer Systems" on page 56 for additional Oracle VM scanning considerations.

Discover Rollout Tools for Windows and Mac OS X Level 2 Data Collection

This section outlines the procedures on how to enable WMI, registry, and DCOM permissions to allow Discover Windows Collection Server (WinCS) to collect Windows system data and application data using a Windows domain or local user account. This section also outlines the procedure for using the rollout script to enable remote login on Mac OS X systems (refer to "BDNA Discover Rollout Tools for Mac OS X" on page 77 for details).

Note: You must perform Task 1 through Task 6 on all systems running Windows XP, Windows Server 2003, and Windows Vista.

Task 1: Enabling Classic Sharing and Security Model

For Windows XP and Windows Server 2003 users:

1. Select Start > Settings > Control Panel > Administrative Tools.
2. Double click Local Security Policy.
3. In Local Security Settings console, expand Security Settings > Local Policies > Security Options.
4. Select “Network access: Sharing and security model for local accounts” option from the right pane.
5. Make sure the option is set to “Classic – local users authenticate as themselves”.

or

1. Open Windows Explorer
2. Select Tools > Folder Options
3. Switch to View tab.
4. Under Advanced settings, uncheck Use simple file sharing (Recommended) option.

For Windows Vista users:

1. Open Control Panel.
2. Switch to the Classic View.
3. Double click the Administrative Tools icon.
4. Double click Local Security Policy.
5. In Local Security Settings console, expand Security Settings > Local Policies > Security Options.
6. Select “Network access: Sharing and security model for local accounts” option from the right pane.
7. Make sure the option is set to “Classic – local users authenticate as themselves”.

Task 2: Allowing Incoming Network Connections on TCP Ports 135 and 445

Note: This step is required only if Windows Firewall is enabled.

For Windows XP and Windows Server 2003 users:

1. Right click My Network Places from the Windows Desktop and then select Properties from the pop-up menu.
2. In Network Connections window, right click the Local Area Network Connection and then select Properties from the pop-up menu.
3. In the LAN Properties dialog, click the Advanced tab to view the Windows Firewall settings.
4. Click the Settings... button.
5. In the Windows Firewall dialog, click the Exceptions tab.
6. Add TCP port 135 to the exception list. To add a TCP port to the exception list, click the Add Port... button, then enter a name and a port number. Be sure to select TCP option. Click OK when done.

7. Add TCP port 445 to the exception list. (Note: If Windows complains port 445 already exists in the Exception list, just ignore it).

For Windows Vista users

1. Open Control Panel.
2. Switch to Classic View.
3. Double click the Network and Sharing Center icon.
4. In Network and Sharing Center dialog, click Windows Firewall located at the lower left corner.
5. Click Allow a program through Windows Firewall. If Windows Vista prompts for permission to continue, click Continue.
6. In Windows Firewall Settings dialog, click Add Port.
7. Add TCP port 135 to the exception list. To add a TCP port to the exception list, then enter a name and a port number. Be sure to select TCP option. Click OK when done.
8. Add TCP port 445 to the exception list.

On Windows Vista, the ICMP echo request is disabled by default if Windows Firewall is enabled. ICMP echo request is required for Discover Level 1 collection to work.

To enable ICMP echo request on Windows Vista OS:

1. Open Control Panel.
2. Switch to Classic View.
3. Double click Administrative Tools.
4. Double click Windows Firewall with Advanced Security.
5. Click on Inbound Rules then right click on it and choose New Rule...
6. Click on the Custom radio button and click Next.
7. Click on All programs and click Next.
8. Set the Protocol Type to ICMPv4 and click the Customize... button
9. Select Specific ICMP types, enable Echo Request and click on the OK button
10. Click on the Next button to go to the Scope page.
11. Click on the Next button again to go to the Action page. Verify if the Allow the connection radio button is selected and click Next to go to the Profile page.
12. Verify that all three rules (Domain, Private, and Public) are all checked and click Next.
13. Specify a Name and Description and click Finish.

Task 3: Allowing Remote Administration Using Administrative Tools Such as WMI

1. Click the Start menu, select Run...
2. Type gpedit.msc and then click OK.

3. In Group Policy console, expand Computer configuration, Administrative Templates, Network, Network Connections, Windows Firewall, Domain Profile. (Note that if the machine is not on a domain, you will need to select Windows Firewall, Standard Profile instead).
4. Double click "Windows Firewall: Allow remote administration exception". (On Windows Vista, it's labeled as "Windows Firewall: Allow incoming remote administration exception"). In the Properties dialog, select Enabled, and then in the Allow unsolicited incoming messages from field, enter *, then click OK.
5. Close the Group Policy console.

Task 4: Enabling Remote DCOM Access

Note: For more information about `dcomsd.exe`, refer to "Rollout Tools Information and Syntax" on page 66. `dcomsd.exe` is required for systems running Windows XP Service Pack 2 or higher, Windows Server 2003 R2, and Windows Vista.

1. Copy `dcomsd.exe` onto the target box, or it can be a remote server if you prefer to run it from a remote network location.
2. Open Command Prompt window.
3. Run `dcomsd.exe` to grant remote DCOM access to a domain user where it will be used to collect Windows Level 2 data. For example, `c:\dcomsd.exe` <*domain*>`\`<*username*>. Note that if the machine to which you are applying `dcomsd.exe` does not belong to a Windows domain, you should run `dcomsd` with only the user name specified. For example, `c:\dcomsd.exe` <*username*>.

For Windows Vista users

Discover Rollout Tools such as `wmisd.exe`, `regsd.exe`, and `dcomsd.exe` must be executed under Command Prompt with "Run as administrator" option.

1. Click Start menu.
2. Select All Programs > Accessories.
3. Right click Command Prompt icon.
4. Select "Run as administrator" from the popup menu.
5. In Command Prompt, run the rollout tools.

Task 5: Granting WMI Privileges and Network Access to a User Account

Note: For more details about the command usage of the rollout tools, refer to "Rollout Tools Information and Syntax" on page 66.

1. Copy `wmisd.exe` onto the target box, or it can be a remote server if you prefer to run it from a remote network location.
2. Open Command Prompt window.

3. Run `wmisd.exe` to grant WMI access and network access to a domain user where it will be used to collect Windows Level 2 data. For example, `c:\wmisd.exe /n /a` <*domain*>`\`<*username*>. Note that if the machine to which you are applying `wmisd.exe` does not belong to a Windows domain, you should run `c:\wmisd.exe /n /a` <*username*>.

Task 6: Granting Registry Read Permissions to a User Account

This step is optional and is required *only* if the HKLM\Software and its subkeys have restricted security permissions configured. In this case, you will need to run `regsd.exe` to grant registry read permissions to the account where WinCS will use to collect Windows Level data. Refer to "Rollout Tools Information and Syntax" for more information on `regsd.exe`.

Run `regsd.exe` to grant registry read permissions to a domain user where it will be used to collect Windows Level 2 data.

For example,

```
c:\regsd.exe <domain>\<username> hklm software /tkas
```

If the machine to which you apply `regsd.exe` does not belong to a Windows domain, you should use the following syntax:

```
c:\regsd.exe <username> hklm software /tkas
```

Rollout Tools Information and Syntax

Discover provides three rollout tools for Windows Level 2 discovery:

- `wmisd.exe`
- `regsd.exe`
- `dcomsd.exe`

These tools are included in
`$BDNA_HOME/pso/scripts/Windows/BDNA_Rollout_Tools_x.x.x_####.zip`

where:

x.x.x is the product version number

#### is the product build number

After unzipping this file, you can find the rollout tools in the Windows subdirectory of the extracted contents.

wmisd.exe

`wmisd.exe` is a tool that uses to grant WMI permissions to a specific user. After running `wmisd.exe`, the user will have the following permissions on CIMV2 and DEFAULT WMI namespaces:

```
ROOT\CIMV2 -> Execute Methods = Allow
ROOT\CIMV2 -> Enable Account = Allow
ROOT\CIMV2 -> Remote Enable = Allow
ROOT\DEFAULT -> Execute Methods = Allow
ROOT\DEFAULT -> Enable Account = Allow
ROOT\DEFAULT -> Remote Enable = Allow
```

In addition, `wmisd.exe` adds a domain account to the following local security policy: Local Policies > User Rights Assignment > Access this computer from the network

`wmisd.exe` also removes a domain account from the following local security policy: Local Policies > User Rights Assignment > Deny access to this computer from the network.

`wmisd.exe` supports the following platforms:

- Windows NT 4.0 Professional and Server
- Windows 2000 Professional and Server
- Windows XP Professional
- Windows 2003 Server (all editions)
- Windows Vista (all editions)
- Windows Server 2008
- Windows Server 2012

The following is the syntax and usage of `wmisd.exe`:

```
WMISD: Grant/Remove WMI Read/Write access.
Version 7.7.2 Build 4004
WMISD: Grant/Remove WMI Read access and Network Access.
 Usage:
     grant the WMI access  : wmisd [domain\]username.
     remove the WMI access : wmisd /r [domain\]username.
     grant the WMI access and network access: wmisd /n [domain\]username.
     remove the WMI access and network access: wmisd /r /n [domain\]username.
   when using SID instead of username just add /s switch.
   to enable local administrator access on Windows Vista add /a switch.
     to disabel local administrator access on Windows Vista add /d switch.
     the local administrator switches can be used standalone, e.g. wmisd /a or
wmisd /d.
```

The following is a list of return codes for `wmisd.exe`:

```
0 - success
13 - failed to allow network access
14 - rollout aborted
26 - failed to add user to WMI default namespace
39 - failed to add user to WMI cimv2 namespace
```

Applying WMI and network access permissions manually (without using `wmisd.exe`): For some reasons, if you want to apply the WMI and network access permissions manually, and do not want to use `wmisd.exe`, you may do the following:

1. On the target box, open the Computer Management console. You may do so by selecting Start > Settings > Control Panel > Administrative Tools > Computer Management.
2. Expand Services and Applications. Right click WMI Control and then select Properties.
3. In the WMI Control Properties dialog, click the Security tab.
4. Expand Root. Select the CIMV2 namespace, then click the Security button.
5. In the Security for ROOT\CIMV2 dialog, click the Add button. Add a domain user you wish to grant the permission. Click OK when done.
6. Now you are back to the Security for ROOT\CIMV2 dialog, select the domain user that you just added. Make sure the Execute Methods, Enable Account, and Remote Enable permission checkboxes on the Allow column are checked. Make sure all other checkboxes are unchecked. Click OK when done.
7. Select the DEFAULT namespace, then click the Security button.
8. In the Security for ROOT\CIMV2 dialog, click the Add button. Add a domain user you wish to grant the permission. Click OK when done.
9. Now you are back to the Security for ROOT\CIMV2 dialog, select the domain user that you just added. Make sure the Execute Methods, Enable Account, and Remote Enable permission checkboxes on the Allow column are checked. Make sure all other checkboxes are unchecked. Click OK when done.
10. Click OK again to close the WMI Control Properties dialog. Close the Computer Management console.
11. Open the Local Security Policy. You may do so by selecting Start > Settings > Control Panel > Administrative Tools > Local Security Policy.
12. Expand Local Policies, then select User Rights Assignment.
13. In the Policy list view, double click “Access this computer from the network”. Add a domain user to this policy where you want to grant the permission to. Click OK when done.
14. In the Policy list view, double click “Deny access to this computer from the network”. If the domain user where you wish to use for Windows data collection exists in this policy, you need to remove it from the list. Click OK when done.
15. Close the Local Policy Settings console.

regsd.exe

`regsd.exe` is a tool that uses to grant registry read permissions on a specific registry key(s) to a specific user. After running `regsd.exe`, the user will have the following permissions on the selected registry keys:

Registry key > Special Permissions > Advanced > User > Read > Query Value = Allow

Registry key > Special Permissions > Advanced > User > Read > Enumerate Subkeys = Allow

`regsd.exe` supports the following platforms:

- Windows NT 4.0 Professional and Server
- Windows 2000 Professional and Server
- Windows XP Professional
- Windows Server 2003 (any editions)
- Windows Server 2008
- Windows Server 2012

The following is the syntax and usage of `regsd.exe`.

```
REGSD: Grant/Remove Registry Read access.
Version 7.7.2 Build 4004
Usage:
  Grant read registry access: regsd [domain\]username hive key [/tkas |
/tko:n].
                           or regsd user_account_sid hive key /s [/tkas |
/tko:n].
  Remove read registry access: regsd [domain\]username hive key /r [/tkas |
/tko:n].
                            or regsd user_account_sid hive key /s /r [/tkas |
/tko:n].
hive parameter should be the following:
 HKCR          -       HKEY_CLASSES_ROOT
 HKCU          -       HKEY_CURRENT_USER
 HKLM          -       HKEY_LOCAL_MACHINE
 HKU           -       HKEY_USERS
 HKCC          -       HKEY_CURRENT_CONFIG
If the key name includes spaces, enclose the key in double quotes.
/tkas         -       Apply to this key and all subkeys through inheritance.
/tko:n        -       Apply to this key and the next n levels of subkeys.
When neither /tkas nor /tko:n is supplied, /tkas is used as default.
The following is a list of return codes for regsd.exe:
0 - success
13 - failed
14 - rollout aborted
```

Applying registry permissions manually (without using `regsd.exe`): For some reasons, if you want to apply the registry permissions manually, and do not want to use `regsd.exe`, you may do the following:

Note: The following example is to demonstrate how to grant registry permissions under the HKLM\SOFTWARE key and all subkeys).

1. On the target box, open the Registry Editor. You may do so by selecting Start Run…. In the Run dialog, enter `regedit` and then click OK.
2. Expand the HKEY_LOCAL_MACHINE hive.
3. Right click the SOFTWARE key, then select Permissions.
4. In the Permissions for SOFTWARE dialog, click the Add button.
5. Add a domain user where you wish to grant the registry permission to. Click OK when done.

6. Now you are back to the Permissions for SOFTWARE dialog. Select the domain user that you just added. Click the Advanced button. This should bring up the Advanced Security Settings for SOFTWARE dialog.
7. Highlight the domain user from the Permission entries list, then click the Edit… button. This should bring up the Permission Entry for SOFTWARE dialog.
8. In the Apply onto: drop down box, select “This key and subkeys”. In the Permissions checkboxes, check Query Value and Enumerate Subkeys on the Allow column. Ensure all other checkboxes are unchecked. The “Apply these permissions to objects and/or containers within this contain only” option is also unchecked. Click OK when done.
9. Now you are back to Advanced Security Settings for SOFTWARE dialog. Ensure both options “Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here” and “Replace permission entries on all child objects with entries shown here that apply to child objects” are unchecked. Click OK when done.
10. Click OK to close the Permissions for SOFTWARE dialog.
11. Close the Registry Editor.

dcomsd.exe

`dcomsd.exe` is a tool that uses to grant remote DCOM access to a specific user account. Systems running Windows XP Professional Service Pack 2 are required to run `dcomsd.exe`. Basically `dcomsd.exe` grants the following permissions to a specific user account:

dcomcnfg > My Computer > Default COM Security > Launch Permissions > Edit Default > User > Remote Launch = Allow

dcomcnfg > My Computer > Default COM Security > Launch Permissions > Edit Default > User > Remote Activation = Allow

`dcomsd.exe` supports the following platforms:

- Windows XP Professional Service Pack 2 or higher
- Windows Server 2003 Service Pack 1 or higher
- Windows Vista
- Windows Server 2008
- Windows Server 2012

```
dcomsd.exe syntax:
DCOMSD: Grant/Remove remote DCOM access.
Version 7.7.2 Build 4004
Usage:
  Grant  remote dcom access: dcomsd [domain\]username.
                          or dcomsd user_account_sid /s
  Remove remote dcom access: dcomsd [domain\]username /r.
                          or dcomsd user_account_sid /s /r
The following is a list of return codes for dcomsd.exe.
0 - success
```

```
13 - failed
14 - rollout aborted
15 - rollout is not required on target OS
```

Applying Remote DCOM Access Manually

For some reasons, if you want to apply the remote DCOM access manually, and do not want to use `dcomsd.exe`, you may do the following:

1. Click the Start menu, select Run...
2. Type dcomcnfg and then click OK.
3. In Component Services console, double click Component Services, then expand Component Services, expand Computers, right click My Computer and then select Properties from the pop-up menu.
4. Check "Enable Distributed COM on this computer" option.
5. In the My Computer Properties dialog, click the COM Security tab, then click the Edit Limits... button under the Launch and Activation Permissions section.
6. In the Launch Permission dialog, Click the Add button, then add a domain user where you would like to use to collect Windows L-2 data.
7. After adding a domain user, select the domain user, then check the Allow option for both Remote Launch and Remote Activation options, and leave other checkboxes unchecked.
8. Click OK when done.
9. Click OK again to close the My Computer Properties dialog.
10. Close the Component Services console.

Required Windows Services

The following Windows Services must be running on the target Windows host in order for Windows Level 2 data collection to work.

```
Windows 2000:
Event Log
Plug and Play
Remote Procedure Call (RPC)
Security Accounts Manager
Windows Management Instrumentation
Windows XP:
COM+ Event System
DCOM Server Process Launcher
Event Log
Network Connections
Network Location Awareness (NLA)
Plug and Play
```

```
Remote Procedure Call (RPC)
Security Accounts Manager
SSDP Discovery Service
System Event Notification
Windows Management Instrumentation
Windows Server 2003:
Application Experience Lookup Service
COM+ Event System
DCOM Server Process Launcher
Event Log
Network Connections
Plug and Play
Remote Procedure Call (RPC)
Security Accounts Manager
System Event Notification
Windows Management Instrumentation
Windows Vista:
Application Information
COM+ Event System
DCOM Server Process Launcher
Group Policy Client
Network Connections
Network Store Interface Service
Plug and Play
Remote Procedure Call (RPC)
Security Accounts Manager
Task Scheduler
User Profile Service
Windows Event Log
Windows Management Instrumentation
```

Known Issues

- The SID feature in rollout tools is not supported on systems running Windows NT 4.0.
- TFTP Client is not installed on Windows Server 2003 R2. We haven't found a away to install TFTP Client on Windows Server 2003 R2, other than manually copy TFTP.exe from another server running Windows Server 2003 (Non-R2) onto the <WINSYS32DIR>\ directory, where <WINSYS32DIR> usually is C:\Windows\System32\”.
- TFTP Client is installed but not enabled on Windows Vista or higher, by default. Therefore collections that required by the TFTP Client will not work until the user manually enabled the TFTP Client option.

- On Windows Vista, the power plan under the Power Options settings is set to Balance by default. This means that the system is set to auto sleep after 1 hour. When the system is put to sleep, none of the Discover collections would work against the system. To avoid this problem, the system should be adjusted so that it won't be put in sleep mode.
- The Last Login User attribute does not collect correctly on Windows Vista OS. The attribute always collects as "\".
- The Installed Hotfixes attribute does not collect correctly on Windows Vista OS. The attribute may collect empty list of installed hotfixes and patches.

Running the Rollout Tools

As stated above, `wmisd.exe`, `regsd.exe`, and `dcomsd.exe` need to be executed on every machine where BDNA Discover is to perform Level 2 discovery when non-Administrator accounts are to be used. Usually, this precludes non-Administrator accounts from being used to perform BDNA Discover Level 2 discovery, however the following methods could make this approach more feasible on a large scale:

- Login Scripts

  In many Domain environments, when a user logs in to the Domain, a centrally managed script is executed on the user's machine. The three rollout tool commands could be added to this script. The obvious limitation to this approach is that the system will not be ready for discovery until a user has logged in to the Domain from that machine at least once after the script was modified. Therefore, this method will work most effectively for desktop machines from which users can be expected to log in to the domain on a regular basis. This approach also requires that typical logon users would have local administrator privileges. Many BDNA Discover customers wish to deny local administrator privileges to typical users so as to reduce their ability to make unauthorized configuration changes to their corporate Windows desktop or laptop systems.

- Remote Execution

  It is possible to remotely execute `wmisd.exe`, `regsd.exe`, and `dcomsd.exe` with a tool such as PsExec, which is available as part of the PsTools portion of the Sysinternals suite. (Sysinternals was previously freeware, but was acquired by Microsoft in July 2006.) Such an approach would start with a list of target Windows machines generated from a BDNA Discover Level 1 scan, or with other lists, such as all computers that have ever been registered in Active Directory.

- Configuration Management Software

  In environments where Microsoft's SMS or Tivoli's Configuration Manager already has a presence, these tools can be used to centrally deploy `wmisd.exe`, `regsd.exe`, and `dcomsd.exe`.

Windows Level 2 Credential Rollout Methodologies Compared

Table 5 illustrates the strengths and weaknesses of each of the various methods for Windows Level 2 credential rollout:

Table 5: Methods for Windows Level 2 Credential Rollout

	Account Type			
	Domain Administrator	**Domain User**	**Local Administrator**	**Local User**
Account Details	Use domain account created for BDNA Discover (or existing account) added to the Domain Administrators group. (Or added to an equivalent group that has been added to the Administrators group locally on each Windows system.)	Use domain account created for BDNA Discover (or existing account) that *has not* been added to any special group to give the account administrator privileges.	Use local account created for BDNA Discover (or existing account) that has been added to the Administrators group locally on each Windows system.	Use local account created for BDNA Discover (or existing account) that *has not* been added to any special group granting administrator privileges.
Additional Grant Required?	No	Yes	No	Yes
Must Touch Every Windows System Where Level 2 Needs to Work?	No	Yes	NO: if using an existing account. YES: if using a new BDNA Discover account (touch to create the new account).	Yes
Scope	Windows systems joined to a central Windows domain or to a few such domains; or to a domain which trusts the domain for the account in the BDNA Discover Windows Level 2 credential.	Windows systems joined to a central Windows domain; or to a few such domains; or to a domain which trusts the domain for the account in the BDNA Discover Windows Level 2 credential.	Any Windows system, regardless of whether it is joined to any domain.	Any Windows system, regardless of whether it is joined to any domain.

Table 5: Methods for Windows Level 2 Credential Rollout (Continued)

	Account Type			
	Domain Administrator	**Domain User**	**Local Administrator**	**Local User**
Security Concerns	MEDIUM: compromised credential would allow attacker to do harm, but credentials remain under centralized control.	LOW: compromised credential would not allow attackers to do much harm, and credentials remain under centralized control.	HIGH: compromised credentials would allow attacker to do harm, and credentials are not under centralized control.	MEDIUM: compromised credentials would not allow attacker to do much harm, but credentials are not under centralized control.
Known Usage by BDNA Discover Customers	Many	Some	None Note: sometimes used during small pilots.	None
Recommended by BDNA Discover for Customer Consideration	Yes	Yes	No	No
Additional Comments	Allow for extremely fast credential rollout, and often extremely high credential coverage rates.	NA	NA	NA

SNMP Credential Requirements

For SNMP, BDNA Discover needs a community string that grants it read-only access to relevant MIBs.

The following SNMP OIDs are queried for Cisco Routers and Switches:

- chassisId
- chassisModel
- chassisSerialNumber
- chassisType
- ciscoFlashDeviceSize
- ciscoMemoryPoolFree
- ciscoMemoryPoolUsed
- cpmCPUTotal1min
- cpmCPUTotal5min
- cpmCPUTotal5sec
- entPhysicalDescr
- flashSize
- ifDescr
- ifNumber
- ifPhysAddress
- ifSpeed
- ifType
- ipAdEntIfIndex
- ipAdEntNetMask
- locIfInBitsSec
- locIfOutBitsSec
- sysDescr
- sysObjectID
- sysUpTime

The following SNMP OIDs are queried for NetApp Filer:

- cpuCount
- ifDescr
- ifPhysAddress
- ifType
- netcache.ncOptions
- nfscache.nfsCacheOptions
- OIDs under cifs.cifsOptions like cifsIsEnabled, cifsDomainName,
- cifsHostName, etc.
- OIDs under filesys.dfTable.dfEntry like dfFileSys, dfHighTotalKBytes,
- dfLowTotalKBytes, dfLowUsedKBytes, dfPlexCount etc.
- OIDs under netapp.netapp1.filesys like dfNumber, volNumber etc.
- OIds under raid.diskSummary like diskTotalCount, diskActiveCount,
- diskSpareCount, diskFailedCount
- OIDs under raid.raidVTable like raidVDiskName, raidVStatus, raidVDiskId,
- raidVTotalMb, raidVUsedMb, etc.
- productFirmwareVersion
- productId
- productModel
- productSerialNum
- productType
- productVendor
- productVersion
- sysName
- sysUpTime

Common OIDs from FCFABRIC-ELEMENT-MIB and FIBRE-CHANNEL-FE-MIB are queried across various Fibre Channel (FC) Switches:

- FcFeElementName
- FcFeFabricName
- fcFeModuleCapacity
- fcFeModuleDescr
- fcFeModuleFxPortCapacity, etc.

For Brocade Fibre Channel Switch, BDNA Discover queries:

- swFabric.swDomainID
- swFCport.swFCPortCapacity
- swSystem.swFirmwareVersion

For McDATA Fibre Channel Switch, BDNA Discover queries:

- ef6000SysConfigSpeed
- SysFirmwareVersion
- SysOemSerialNum for specific switch model

For Cisco Fibre Channel Switch, BDNA Discover queries:

- dmDomainId
- vsanMediaType
- vsanName
- vsanNumber

As BDNA Discover continues to enhance its products, discovery may access additional OIDs from additional device-specific SNMP MIBs in the future.

BDNA Discover Rollout Tools for Mac OS X

Discovery for Mac OS X requires an SSH connection to the target machine. This can be enabled through the Remote Login option in Mac OS X. Each computer to be scanned will need to have Remote Login enabled. Note that by default, Remote Login is disabled on Mac OS X installations. Remote Login can be enabled through the GUI or the rollout script.

GUI

Run the System Preferences application and go to the Sharing pane. Remote Login can be enabled under the Services tab. Administrator credentials will be needed to modify this option.

Rollout Script

The rollout script provided will enable Remote Login. The script is given as uncompiled AppleScript in the `Enable Remote Login.scpt` file. It can simply be run as is through the AppleScript Editor, or the script can be compiled before being copied to the target machines for execution.

Before running, the script will ask for administrator credentials. It is possible to place the credentials directly into the script; however, this method is insecure, even if the script is then compiled into a run-only application. To do so, open the script in the editor and modify the last line from:

```
do shell script "sh -c " & quoted form of shell_command with administrator
privileges
```

To:

```
do shell script "sh -c " & quoted form of shell_command user name "<name>"
password "<password>" with administrator privileges
```

where <name> and <password> are the name and password of an administrator account, respectively.

The rollout script is provided in plain text as well as the file `Enable Remote Login.scpt` if necessary.

The rollout script simply is an AppleScript wrapper that executes UNIX shell commands to enable Remote Login and/or the SSH server. The exact commands vary with the version of Mac OS X being run. In later versions, the file `/System/Library/LaunchDaemons/ssh.plist` controls this option. The utility `/bin/launchctl` can be used to modify it.

Older versions may use the `/sbin/service` utility, or configuration files such as `/private/etc/xinetd.d/ssh` or `/etc/hostconfig`.

Automation

Administrative software can be used to enable Remote Login on target machines. For example, Apple Remote Desktop 3 can be used to run the rollout script on any administrated machines, or it can be used to run UNIX commands to enable Remote Login directly. Automator tasks or various third-party software can be used as well. Refer to the Apple Remote Desktop 3 documentation for details.

Testing Credentials

The most thorough and meaningful way to test BDNA Discover Level 2 credentials is to perform a Level 2 scan and see where BDNA Discover is able to obtain Level 2 access and where it is denied. However, given that a non-trivial size scan will take at least a few hours to complete, and could take multiple days, after the scan is completed is not the ideal time to learn that a Level 2 credential did not work. Therefore, one should always test credentials before scanning.

Tip: Always continue to test credentials periodically, even if the credential worked successfully in the past. It is always possible that the account was disabled, has expired, or its password has timed out or been changed.

Individual credentials may be tested using the Scan Administration application at the point of credential creation. This procedure is covered in "Creating New UNIX Credentials for Level 2 Discovery" on page 80 and in "Creating New Windows Credentials for Level 2 Discovery" on page 83. Be aware that credential testing during credential creation only allows for testing of single IP address at a time. For broader credential testing, one of the other methodologies below is needed.

BDNA Discover Credential Checker Overview

To simplify the process of checking credentials, BDNA Discover has developed a Credential Checker utility. This Windows utility can check Windows, UNIX, and SNMP credentials over a range of IP addresses. In addition to checking to see if an account's password is valid, in the case of Windows credentials, it will also verify that the account has the appropriate privilege. For example, suppose the Windows credential is a non-Administrator account but `wmisd.exe` was never executed on the target machines. The Credential Checker would flag the fact that while the account allowed BDNA Discover to log in, it did not grant sufficient privileges to perform a BDNA Discover Level 2 scan. Note that while the Credential Checker is not as scalable as the BDNA Discover product, it was designed for near-real-time interactive usage.

Note: For more information about the Credential Checker, refer to "Installing and Running the BDNA Discover Credential Checker".

Other Methods for Testing Credentials

Aside from the BDNA Discover Credential Checker, the following are other methods for checking credentials outside of performing a scan.

- UNIX: UNIX Level 2 credentials can easily be tested using the appropriate command-line utility for the type of connection being used, i.e., `telnet` or `ssh`.
- Windows: Windows Level 2 credentials can be tested using the net use command to connect to the `\\HOSTNAME` pseudo-share, specifying just the hostname without a specific share name.
- SNMP: SNMP Level 2 Credentials can be tested using the `snmpwalk` and `snmpget` commands.

The aforementioned methods have an advantage over the BDNA Discover Credential Checker in that they are native methods; that is, they are not BDNA Discover tools, so they are readily available to system administrators. It would presumably be easier to convince a system administrator that there is a credential problem rather than a BDNA Discover product problem by using one of the above than by using the BDNA Discover Credential Checker. However, they do not scale well enough to test a range of machines. The preferred methodology is to test a range of IP addresses with the BDNA Discover Credential Checker and then to spot-check the results using the above "native" methods.

Applying Credentials Selectively

Assuming valid credentials, the credentials must then be made available to BDNA Discover before the system can perform successful Level 2 discovery. It is important to understand that credentials are associated with Scan Tasks and not applied globally. This is because if BDNA Discover is given many different credentials, and asked to try all of the credentials against all the systems, it can create a discovery throughput problem where BDNA Discover must try many credentials and wait for most of them (probably all but one) to fail on each individual targeted system.

Instead, the system enables association of credentials with Scan Tasks, allowing the administrator to assign the credentials where they are most likely to work. For instance, a customer may be planning to use two different Windows credentials for the Americas (based on the dominant Windows domains there), two different Windows credentials for Europe (based on the dominant Windows domains there), and three different Windows credential for everywhere else, (although even this is probably more credentials than would be optimal). Then a separate Scan Task would be configured for each geographic region (one for the Americas, one for Europe, one for everywhere else).

Note: When it comes to the number of credentials, less is more. In addition to generating a lot of administrative overhead (entering, maintaining, and deleting credentials), having numerous credentials creates discovery throughput problems. Moreover, the more credentials are stored, the more difficult it is to validate them.

Managing Credentials

Credentials, similarly to networks, may be created before a Scan Task is created, or during the process of creating a new Scan Task. Credentials created either way will be available for use in all future Scan Tasks within the current schema. Whether created before or during Scan Task creation, the process is the same.

Level 2 credentials can be created for the following platforms:

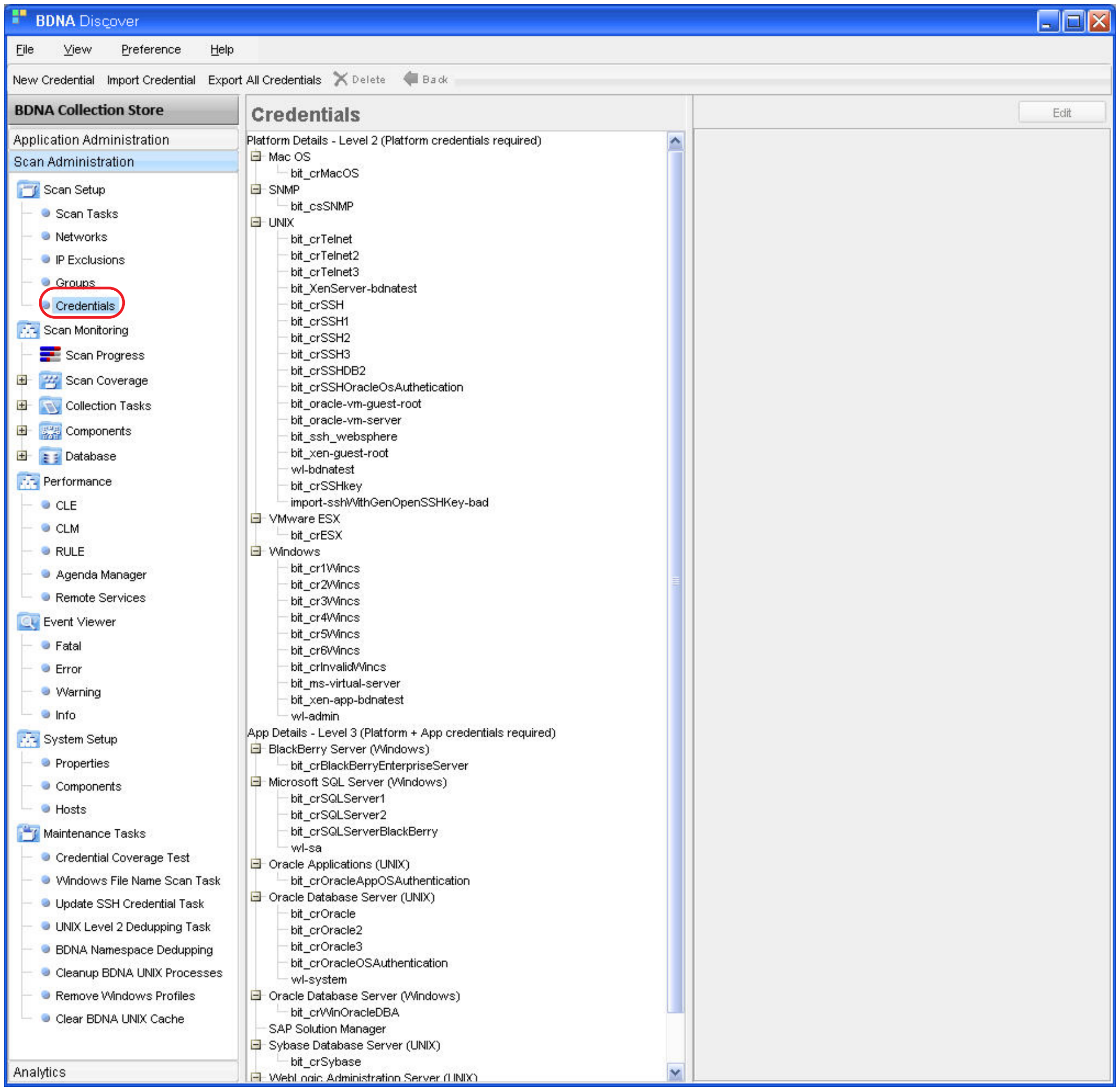
- UNIX
- Windows
- SNMPv3
- SMI-S
- MacOS
- VMware ESX

Creating New UNIX Credentials for Level 2 Discovery

To create a UNIX Credential in the Scan Setup section of Scan Administration:

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup in the left navigation pane, click Credentials.

Figure 20: Create New Credentials


BDNA Discover
File View Preference Help
New Credential Import Credential Export All Credentials Delete Back
Edit

BDNA Collection Store
Application Administration
Scan Administration

- Scan Setup
  - Scan Tasks
  - Networks
  - IP Exclusions
  - Groups
  - Credentials
- Scan Monitoring
  - Scan Progress
  - Scan Coverage
  - Collection Tasks
  - Components
  - Database
- Performance
  - CLE
  - CLM
  - RULE
  - Agenda Manager
  - Remote Services
- Event Viewer
  - Fatal
  - Error
  - Warning
  - Info
- System Setup
  - Properties
  - Components
  - Hosts
- Maintenance Tasks
  - Credential Coverage Test
  - Windows File Name Scan Task
  - Update SSH Credential Task
  - UNIX Level 2 Dedupping Task
  - BDNA Namespace Dedupping
  - Cleanup BDNA UNIX Processes
  - Remove Windows Profiles
  - Clear BDNA UNIX Cache

Analytics

Credentials

Platform Details - Level 2 (Platform credentials required)
- Mac OS
  - bit_crMacOS
- SNMP
  - bit_csSNMP
- UNIX
  - bit_crTelnet
  - bit_crTelnet2
  - bit_crTelnet3
  - bit_XenServer-bdnatest
  - bit_crSSH
  - bit_crSSH1
  - bit_crSSH2
  - bit_crSSH3
  - bit_crSSHDB2
  - bit_crSSHOracleOsAuthetication
  - bit_oracle-vm-guest-root
  - bit_oracle-vm-server
  - bit_ssh_websphere
  - bit_xen-guest-root
  - wl-bdnatest
  - bit_crSSHkey
  - import-sshWithGenOpenSSHKey-bad
- VMware ESX
  - bit_crESX
- Windows
  - bit_cr1Wincs
  - bit_cr2Wincs
  - bit_cr3Wincs
  - bit_cr4Wincs
  - bit_cr5Wincs
  - bit_cr6Wincs
  - bit_crInvalidWincs
  - bit_ms-virtual-server
  - bit_xen-app-bdnatest
  - wl-admin

App Details - Level 3 (Platform + App credentials required)
- BlackBerry Server (Windows)
  - bit_crBlackBerryEnterpriseServer
- Microsoft SQL Server (Windows)
  - bit_crSQLServer1
  - bit_crSQLServer2
  - bit_crSQLServerBlackBerry
  - wl-sa
- Oracle Applications (UNIX)
  - bit_crOracleAppOSAuthentication
- Oracle Database Server (UNIX)
  - bit_crOracle
  - bit_crOracle2
  - bit_crOracle3
  - bit_crOracleOSAuthentication
  - wl-system
- Oracle Database Server (Windows)
  - bit_crWinOracleDBA
- SAP Solution Manager
- Sybase Database Server (UNIX)
  - bit_crSybase
- WebLogic Administration Server (UNIX)

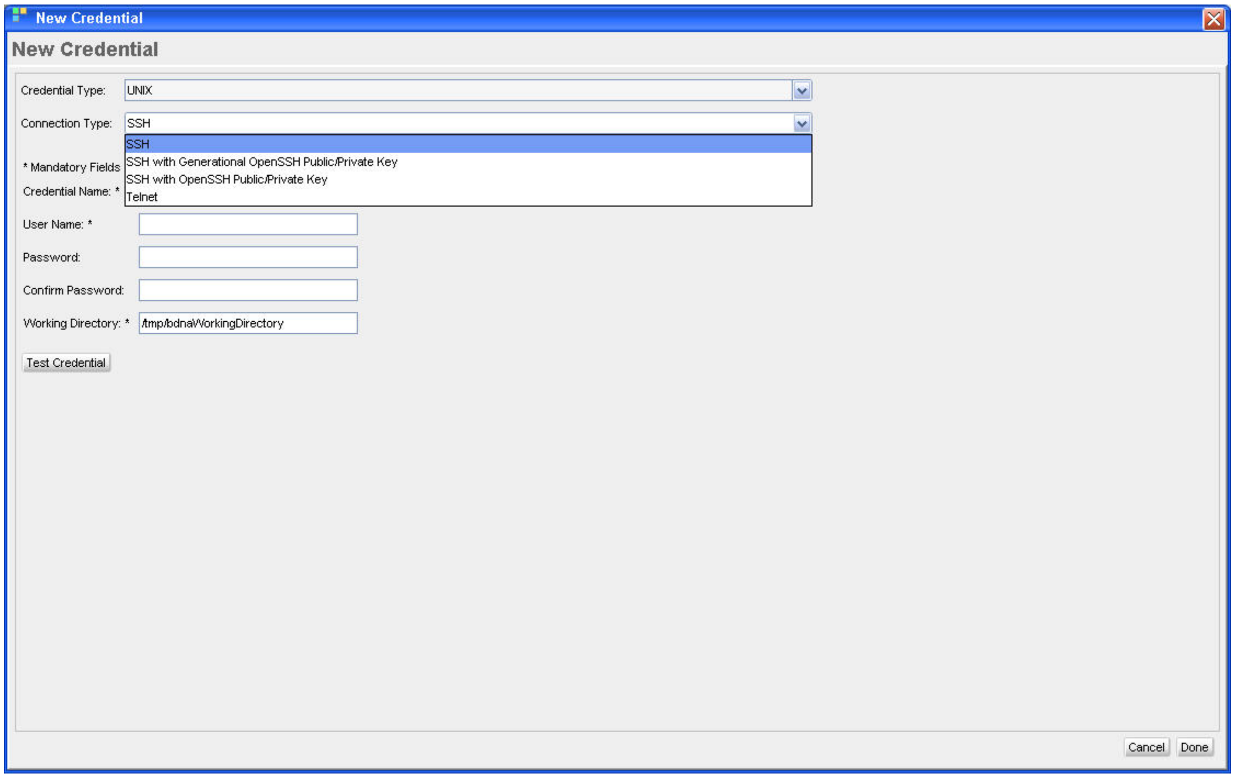
3. In the toolbar under the application menu at the top, click New Credential.

   Scan Administration displays the New Credential wizard.

4. For this example, in the Credential Type drop-down list, highlight UNIX and click Select.

   The New Credential wizard displays fields specific to your selection.

Figure 21: Create New Credentials: UNIX



New Credential

New Credential

Credential Type: UNIX

Connection Type: SSH

SSH
SSH with Generational OpenSSH Public/Private Key
SSH with OpenSSH Public/Private Key
Telnet

* Mandatory Fields

Credential Name: *

User Name: *

Password:

Confirm Password:

Working Directory: * /tmp/bdnaWorkingDirectory

The following choices appear in the Connection Type drop-down list:

- SSH
- SSH with Generational OpenSSH Public/Private Key
- SSH with OpenSSH Public/Private Key
- Telnet

5. For this example, choose SSH.

6. Complete the fields as applicable:

- Credential Name should be a short, descriptive name.
- User Name is the UNIX account name.
- Password and Confirm Password are for the account password (optional fields).
- Working Directory is a temporary directory that will be used during discovery.

  The default should be used unless there are specific reasons not to use it.

7. If desired, test the credential by clicking on Test Credential at the bottom of the form.

Note: Selecting Test Credential requires a connection method of type SSH or Telnet only; SSH with Public/Private Key will not work.

8. When prompted, enter an IP address and click Test.

9. When the form is completed, click Done to save the credential.

The credential appears in the credentials list, under UNIX.

Creating New Windows Credentials for Level 2 Discovery

To create a Windows credential in the Scan Setup section of Scan Administration:

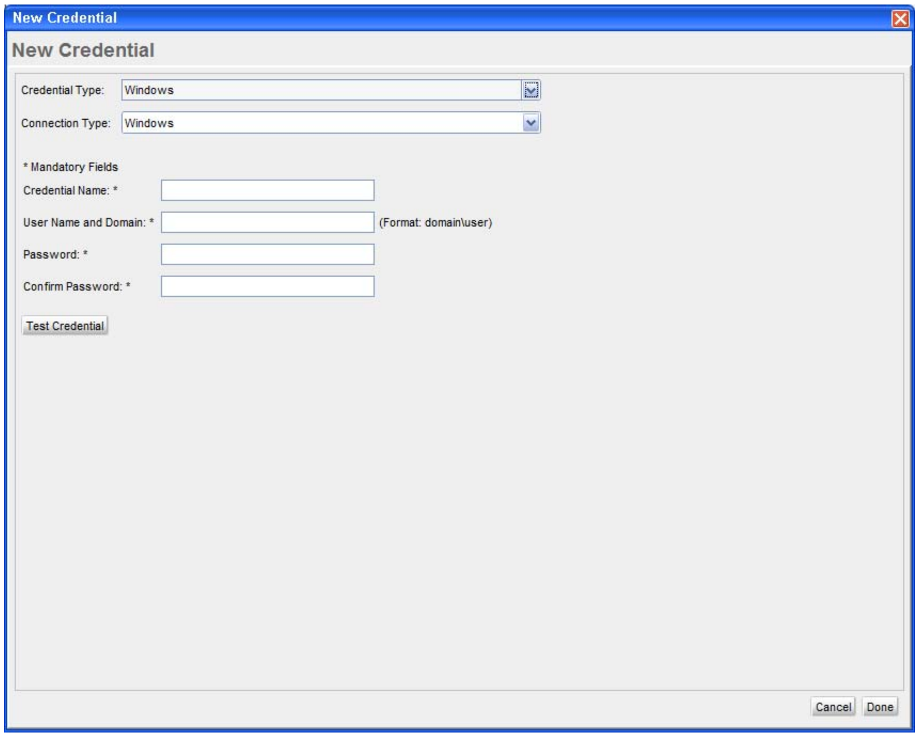
1. Log in to BDNA Discover and select Scan Administration.

2. Under Scan Setup, click Credentials.

3. In the toolbar under the application menu at the top, click New Credential.

The New Credential wizard appears.

Figure 22: New Credential Wizard (Windows)


New Credential

New Credential

Credential Type: Windows

Connection Type: Windows

* Mandatory Fields

Credential Name: *

User Name and Domain: * (Format: domain\user)

Password: *

Confirm Password: *

4. For this example, in the Credential Type drop-down list, choose Windows.

For Connection Type, only Windows is available.

5. Fill in the form as applicable:

5.1. Credential Name should be a short, descriptive name.

5.2. User Name and Domain is the Windows account name along with the domain name.

It uses the format DOMAIN\account_name. Note that for local accounts, a period (.) should be used for the domain (e.g., .\Administrator)

5.3. For Password and Confirm Password, enter the account password.

6. If desired, test the credential by clicking on Test Credential at the bottom of the form.

   Note that the Test Credential button will be grayed out if there is no WinCS component currently connected to the system. When prompted, enter an IP address and click Test.

7. When the form is completed, click Done to save the credential.
8. The credential appears in the Credentials list, under Windows.

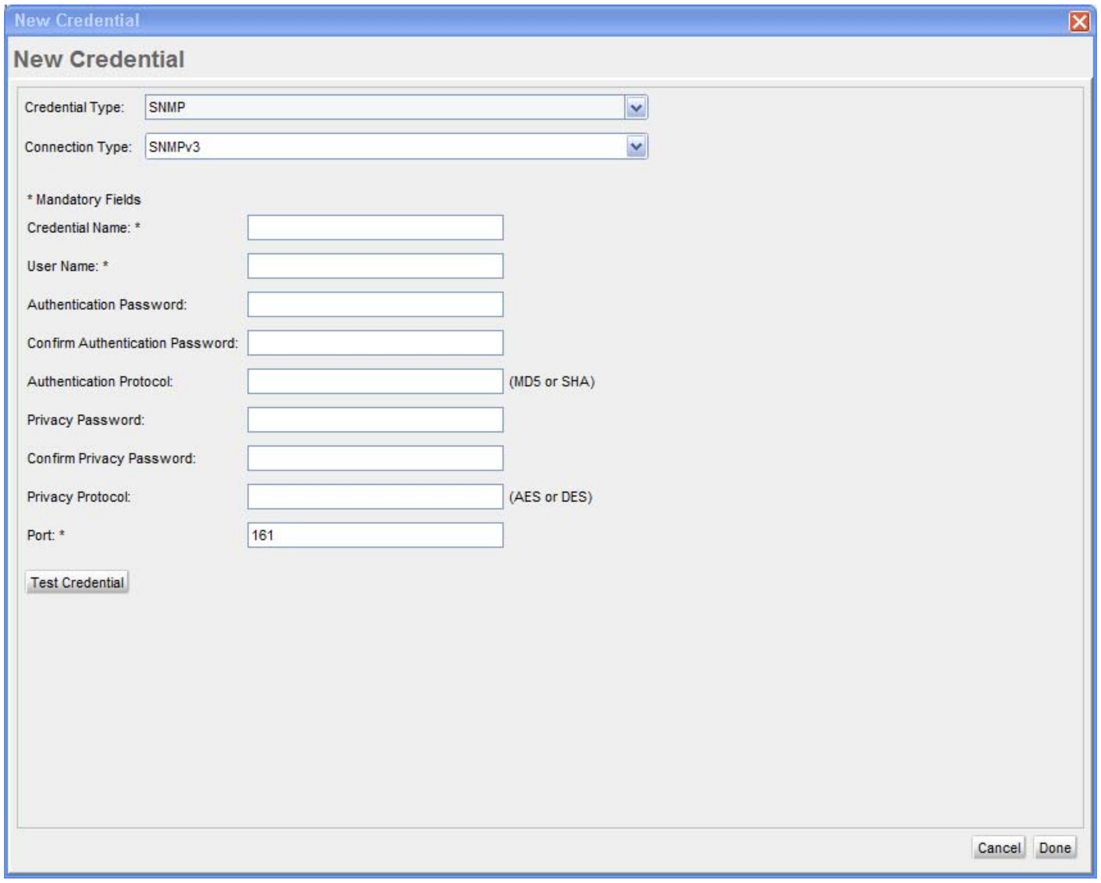
Creating New SNMP Credentials for Level 2 Discovery

To create an SNMP credential in the Scan Setup section of Scan Administration:

1. Log in to BDNA Discover and select Scan Administration.
2. Under Scan Setup, click Credentials.
3. In the toolbar under the application menu at the top, click New Credential.

   The New Credential wizard appears.

Figure 23: New Credential Wizard (SNMP)



New Credential

New Credential

Credential Type: SNMP

Connection Type: SNMPv3

* Mandatory Fields

Credential Name: *

User Name: *

Authentication Password:

Confirm Authentication Password:

Authentication Protocol: (MD5 or SHA)

Privacy Password:

Confirm Privacy Password:

Privacy Protocol: (AES or DES)

Port: * 161

4. For this example, in the Credential Type drop-down list, choose SNMP.

   For this example, in the Connection Type drop-down list, choose SNMPv3.

5. Fill in the form as applicable (asterisks indicate required fields):
   - 5.1. Credential Name should be a short, descriptive name.
   - 5.2. User Name is the SNMP security name configured on the target system.
   - 5.3. If you want to use SNMPv3 authentication, provide the password in the Authentication Password field.
   - 5.4. Confirm the authentication password in the Confirm Authentication Password field.
   - 5.5. In the Authentication Protocol field, enter the same value as the one configured on the target system.
   - 5.6. If you want to use the SNMPv3 privacy feature, enter the privacy password in the Privacy Password field.

Note: SNMPv3 privacy requires the use of authentication. To use the SNMPv3 privacy feature, you must provide values in the authentication fields as well as the privacy fields.

   - 5.7. Confirm the privacy password in the Confirm Privacy Password field.
   - 5.8. Enter the privacy protocol in the Privacy Protocol field.
   - 5.9. The default SNMP port value is provided in the Port field. If you have configured it on a different port, enter that value here.

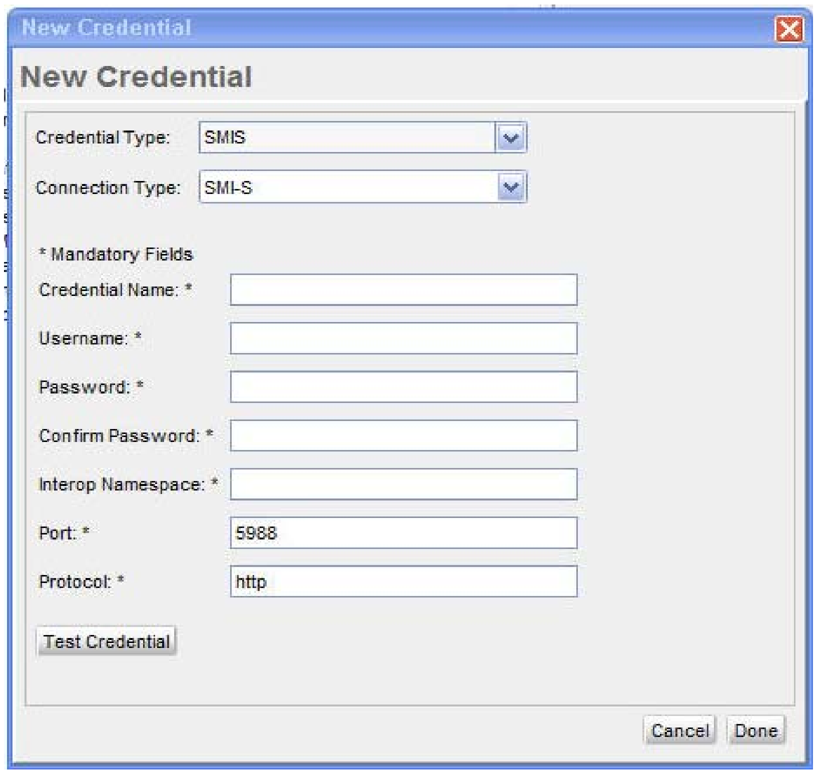
6. If desired, test the credential by clicking on Test Credential at the bottom of the form.
7. When the form is completed, click Done to save the credential.
8. The credential appears in the Credentials list, under SNMP.

Creating New SMIS Credentials for Level 2 Discovery

To create an SMI-S credential in the Scan Setup section of Scan Administration:

1. Log in to BDNA Discover and select Scan Administration.
2. Under Scan Setup, click Credentials.
3. In the toolbar under the application menu at the top, click New Credential.
4. The New Credential wizard appears.

Figure 24: New Credential Wizard (SMIS)



Credential Type:	SMIS
Connection Type:	SMI-S

* Mandatory Fields

Credential Name: *	
Username: *	
Password: *	
Confirm Password: *	
Interop Namespace: *	
Port: *	5988
Protocol: *	http

5. For this example, in the Credential Type drop-down list, choose SMIS.

6. For this example, in the Connection Type drop-down list, choose SMI-S.

7. Fill in the form as applicable (asterisks indicate required fields):

7.1. Credential Name should be a short, descriptive name.

7.2. Username is the SMIS security name configured on the target system.

7.3. Password is the password for the SMIS username.

7.4. Confirm the password in the Confirm Password field.

7.5. Specify the interop namespace for the CIM agent in the Interop Namespace field.

7.6. The default SMIS port value is provided in the Port field. If you have configured it on a different port, enter that value here.

7.7. The default protocol is provided in the Protocol field. If you have configured a different protocol, enter that value here.

7.8. If desired, test the credential by clicking on Test Credential at the bottom of the form.

7.9. When the form is completed, click Done to save the credential.

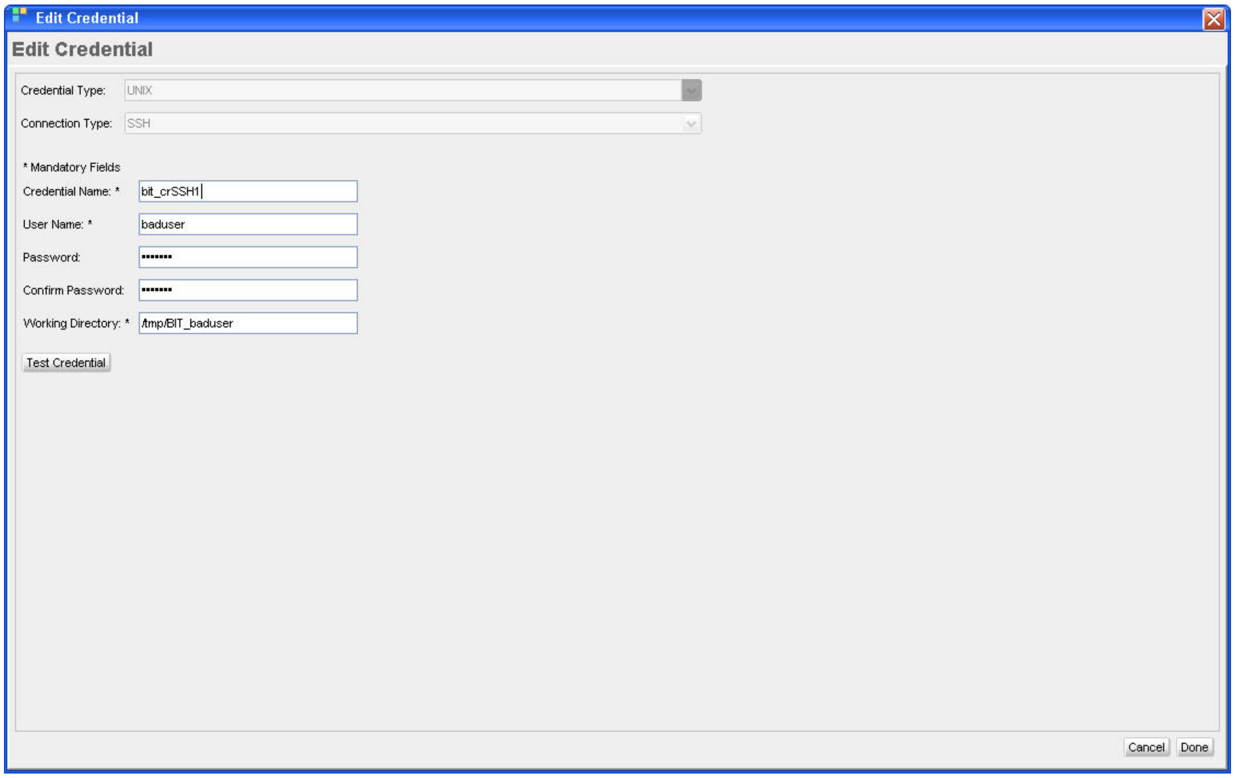
7.10. The credential appears in the Credentials list, under SMIS.

Modifying Existing Credentials

To modify a credential using the Scan Setup section of Scan Administration:

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup, click Credentials.
3. In the Credentials listing in the middle panel, locate the credential to be modified, and highlight it by clicking on it.
4. In the top right corner, click the Edit button to bring up the Edit Credential wizard.

Figure 25: Edit Credential Wizard



Credential Type: UNIX
Connection Type: SSH
* Mandatory Fields
Credential Name: * bit_crSSH1
User Name: * baduser
Password: •••••••
Confirm Password: •••••••
Working Directory: * /tmp/BIT_baduser

5. The Edit Credential wizard operates identically to the New Credential wizard.

   Refer to "Managing Credentials" on page 80 for more information.

Credential Precedence

It is quite possible that there will be multiple Credentials that could be used for a given host. For example, there may be two or more Windows Level 2 credentials for a Scan Task, only one of which is valid for any host. Which credential will BDNA Discover try first?

The short answer is that there is no way to determine which credential BDNA Discover will try first. BDNA Discover will arbitrarily try one of the credentials. If that credential fails, it will try the next credential assigned to the Scan Task, and so on until one of the credentials succeeds or all of the credentials have failed. Also note that the next time BDNA Discover has to access the same machine in a different scan created in a different BDNA Discover database repository, it will not remember which credential worked the last time so it will repeat this process.

For these reasons, all efforts should be made to have as few credentials as possible. Instead of having 20 credentials attached to one huge Scan Task it is much better to use one or two credentials for each of many small targeted Scan Tasks, as appropriate.

Creating Level 2 Scan Tasks

Configuring a Level 2 scan is very similar to configuring a Level 1 scan as described in "Configuring and Running Level 1 Discovery" on page 35. However, one thing to consider when planning for Level 2 discovery is whether a new Level 1 discovery task will need to be performed in conjunction with the new Level 2 discovery task.

You must complete a Level 1 scan task on a given set of assets before you can perform Level 2 discovery on them. This is because BDNA Discover needs to know some details about a machine (what OS it is running, etc.), before it tries to access it at Level 2. Level 1 discovery is how BDNA Discover obtains this information.

It is possible to run the Levels 1 and 2 scans within the same task. However, BDNA Discover also allows a BDNA Discover Administrator to separate the Level 1 and Level 2 scans into two different Scan Tasks. A Level 1 task could be run in the morning on a particular Network, and then a corresponding Level 2 task would be run in the afternoon. Reasons for doing so are beyond the scope of this document, but it may be helpful to know that it is possible. It is even more important is to understand when it would not be a good practice to rely on an earlier Level 1 scan before a Level 2 scan that depends upon its data.

A Level 1 scan captures the state of assets at a given moment in time. It provides no guarantee that the assets discovered will not change in the future. For this reason, any Level 2 scan relying on a previous Level 1 scan should be sufficiently close in time to the Level 1 scan to guarantee that the Level 1 data is still valid at the time of the Level 2 scan.

How long the Level 1 data will be valid, in terms of usability for a Level 2 scan, depends on the types of assets being scanned, as well as on the network environment in which those assets live. The following are some reasons why Level 1 data can become too stale to use for a Level 2 scan:

- DHCP Environments

  In environments that use DHCP, IP addresses for given machines may change arbitrarily at any time. This could be problematic when a Level 2 scan is performed well after the Level 1 scan, as a machine could have changed IP addresses in the intervening time. BDNA Discover would try a Level 2 discovery against the original IP address, which would not produce the desired result.

- Desktop Users Shutting Down

  If the environment being scanned contains desktop users, a Level 1 scan performed at 10:00am may have very different results than a Level 1 scan performed at 10:00pm. Users, especially those who use laptops as their primary Windows system, may shut down or disconnect their machines when they go home.

Undoubtedly there are many other reasons why Level 1 data can become too stale to use for a Level 2 scan. Because of this, it is often a better practice to run the Levels 1 and 2 scans within the same task.

To create a Level 2 scan task:

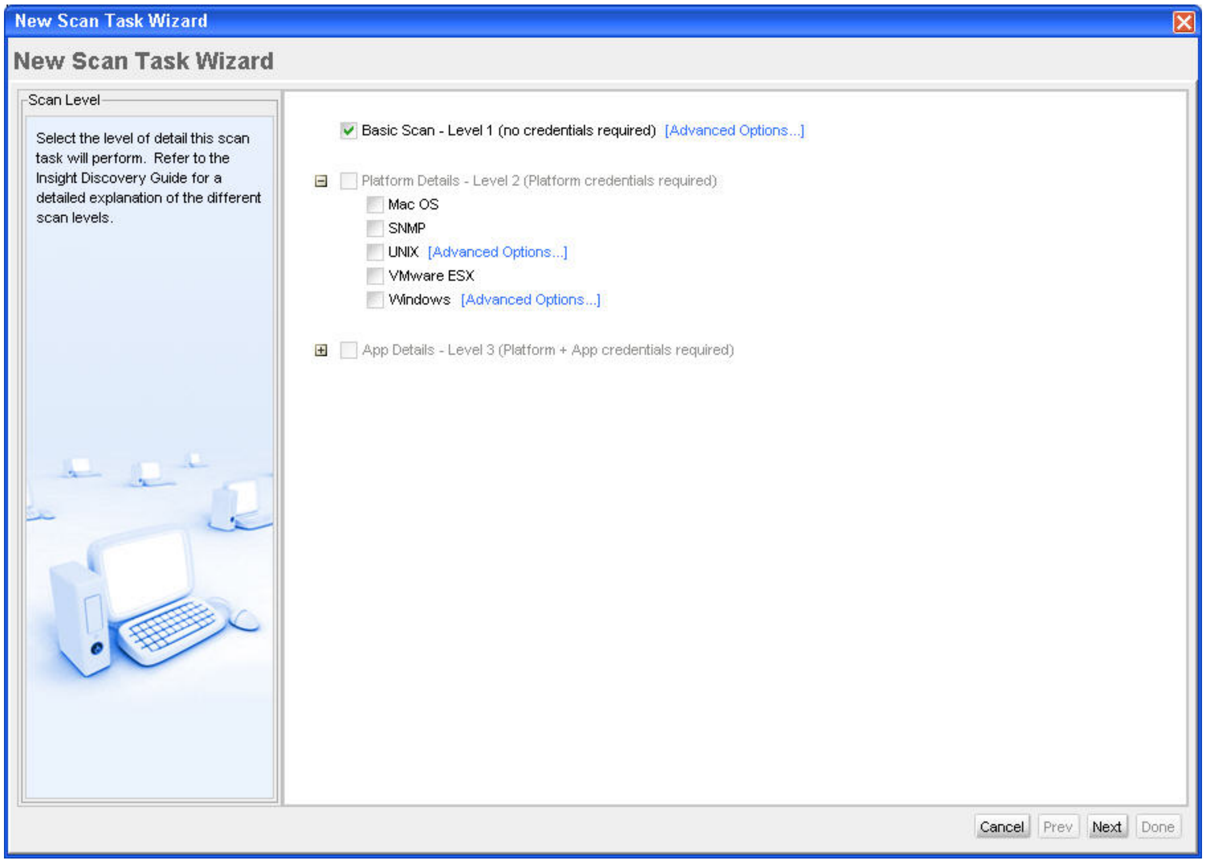
Note: The following procedure assumes that a Level 2 credential has previously been created. However, credentials can also be defined during scan task creation.

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup, click Scan Tasks.

3. In the toolbar underneath the application menu, click New Task.

Scan Administration displays the New Scan Task wizard.

Figure 26: New Scan Task Wizard: Platform Details - Level 2, Scan Level


New Scan Task Wizard

New Scan Task Wizard

Scan Level

Select the level of detail this scan task will perform. Refer to the Insight Discovery Guide for a detailed explanation of the different scan levels.

☑ Basic Scan - Level 1 (no credentials required) [Advanced Options...]

☐ Platform Details - Level 2 (Platform credentials required)
- ☐ Mac OS
- ☐ SNMP
- ☐ UNIX [Advanced Options...]
- ☐ VMware ESX
- ☐ Windows [Advanced Options...]

☐ App Details - Level 3 (Platform + App credentials required)

The first page of the New Scan Task wizard specifies what to collect. The default is Basic Scan – Level 1.

4. For simultaneous Level 1 and Level 2 scanning, leave Basic Scan – Level 1 checked; otherwise, uncheck it.

5. Specify the appropriate Level 2 platform details (refer to "Routable Access Requirements" on page 57 for the Level 2 credential requirements for the supported platforms):

For Mac OS, SNMP, and/or VMware ESX, select the checkbox(es) adjacent to the appropriate platform(s) and proceed to step 6 on page 90.

or

For UNIX and/or Windows, you have the option of scanning for both system and applications information (the default), or scanning for system information only. BDNA Discover recommends the default of Level 2 scanning for both system and applications information in most cases. However, in cases such as when you are collecting BDNA Discover inventory for a specific project where applications information is known not to be relevant, scanning for system information only is significantly faster.

To scan for both system and applications information on UNIX and/or Windows systems, select the checkbox(es) adjacent to one or both platforms and proceed to step 6 on page 90.

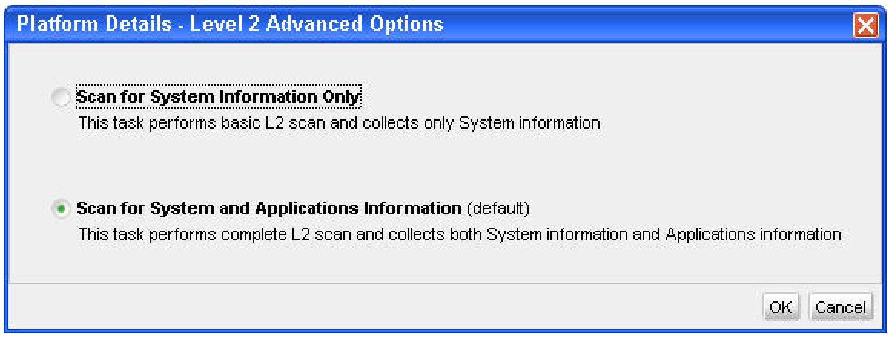
or

To scan for system information only:

5.1. Click Advanced Options adjacent to UNIX or Windows.

The Platform Details - Level 2 Advanced Options dialog box displays.

Figure 27: Platform Details - Level 2 Advanced Options



Platform Details - Level 2 Advanced Options

○ **Scan for System Information Only**
This task performs basic L2 scan and collects only System information

● **Scan for System and Applications Information** (default)
This task performs complete L2 scan and collects both System information and Applications information

5.2. Check the radio button adjacent to Scan for System Information Only.

5.3. Click OK.

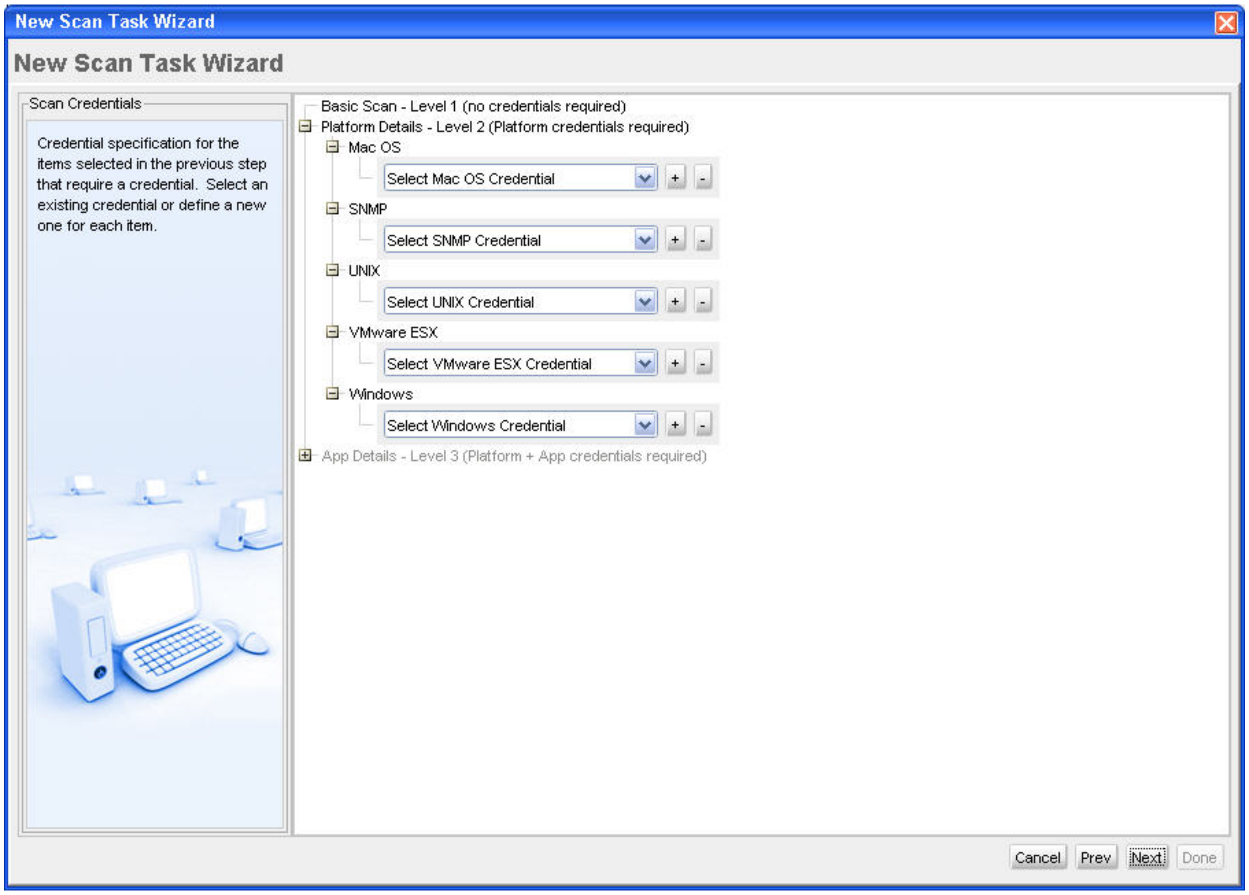
The New Scan Task wizard displays your selection.

5.4. If applicable, repeat this process for the OS that you didn't select in step 5.1; otherwise, proceed to step 6.

6. Click Next.

The next page of the New Scan Task wizard specifies what credentials to use for the scan. The wizard displays one drop-down list for each credential type specified on the previous screen.

Figure 28: New Scan Task Wizard: Platform Details - Level 2, Scan Credentials


New Scan Task Wizard
New Scan Task Wizard
Scan Credentials
Basic Scan - Level 1 (no credentials required)
Platform Details - Level 2 (Platform credentials required)
Mac OS
Select Mac OS Credential
SNMP
Select SNMP Credential
UNIX
Select UNIX Credential
VMware ESX
Select VMware ESX Credential
Windows
Select Windows Credential
App Details - Level 3 (Platform + App credentials required)
Cancel
Prev
Next
Done

Credential specification for the items selected in the previous step that require a credential. Select an existing credential or define a new one for each item.

7. Specify the scan credential(s):

To use an existing credential, select the applicable credential from the drop-down list adjacent to the platform specified in the previous screen; repeat for remaining platforms, as applicable.

or

To create a new credential, select Create [*platform*] Credential from the drop-down list(s) adjacent to the platform(s) specified previously; complete the fields in the New Credential dialog box that displays and click Done when you are finished. Refer to "Managing Credentials" on page 80 for details.

8. Use the "+" next to the drop-down list to add a second credential to the scan, if applicable; use the "-" to remove a credential from the Scan Task.

9. When you are finished specifying credentials, click Next.

The next page of the New Scan Task wizard specifies where collection will occur.

Any networks or groups already defined appear listed on this page. Note that there are also buttons on the bottom of this page to create and import both networks and groups.

Note: For more information on groups, refer to the *BDNA Discover Administration Tutorial, Module 1*.

10. Click the checkbox next to Select (located at the top of the column) to select all networks and groups, or select entries individually.
11. Click Next.

    The next page of the New Scan Task wizard specifies when collection will occur.
12. On the Scan Schedule page, define when you want the scan task to run.

    Also note the Advanced Scheduling button. Refer to step 8 on page 43 for details.

Note: Be sure that you *always* specify a scan end time by specifying Advanced Scheduling and clicking the Edit button.

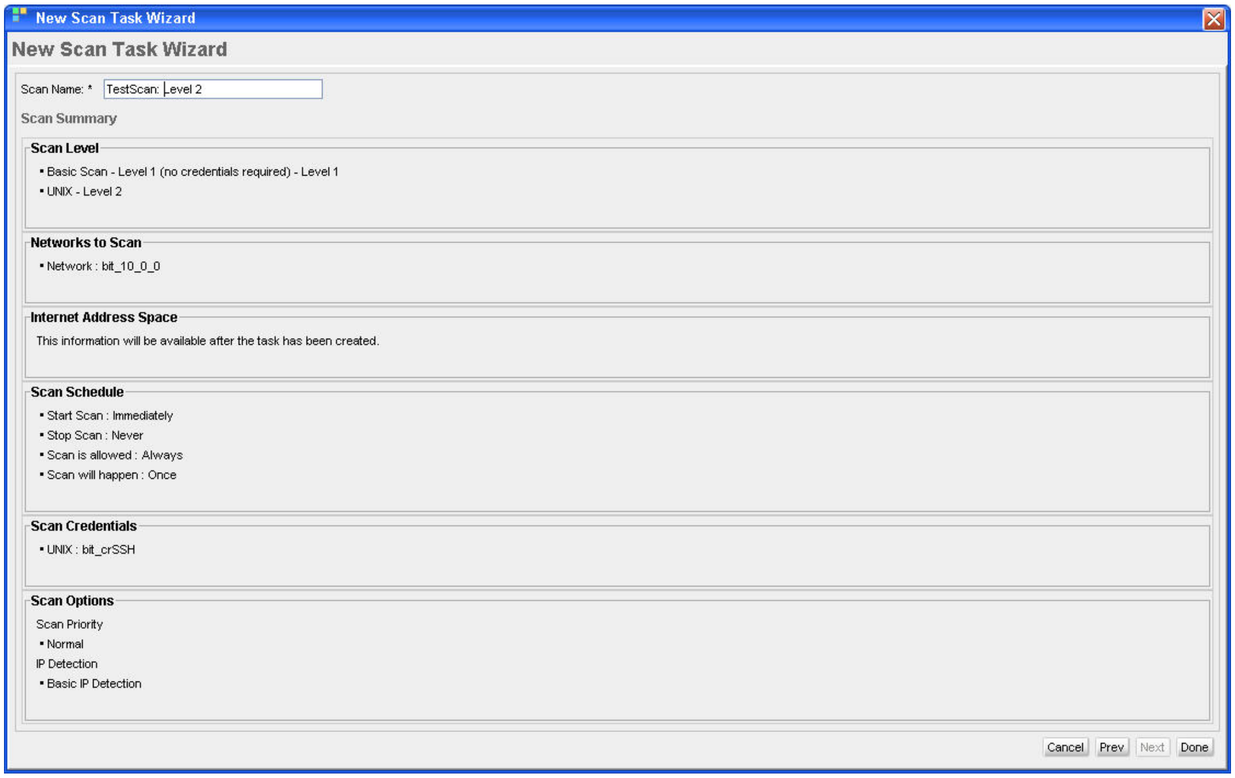
13. When you have finished specifying the scan schedule, click Next.

    Scan Administration displays the Scan Option page of the New Scan Task wizard.
14. Specify a priority for the scan:

Normal:	Collection happens in the order in which the task was queued, but behind any scheduled high-priority task.
High:	Collection happens in the order in which the high-priority task was queued, ahead of normal-priority task.
15. Click Next.

    The New Scan Task wizard displays a summary of your previous selections.

Figure 29: New Scan Task Wizard: Scan Summary Page—Level 2



New Scan Task Wizard

Scan Name: * TestScan: Level 2

Scan Summary

Scan Level
- Basic Scan - Level 1 (no credentials required) - Level 1
- UNIX - Level 2

Networks to Scan
- Network : bit_10_0_0

Internet Address Space

This information will be available after the task has been created.

Scan Schedule
- Start Scan : Immediately
- Stop Scan : Never
- Scan is allowed : Always
- Scan will happen : Once

Scan Credentials
- UNIX : bit_crSSH

Scan Options

Scan Priority
- Normal

IP Detection
- Basic IP Detection

16. On the Scan Summary page, assign a unique and descriptive name for the new scan task.

17. Examine the overview of all of the configuration settings chosen on the previous pages of the Wizard, and make sure they are as intended; to return to a previous page and reset a configuration item, click Prev.

or

When you are satisfied with the configuration settings for the new scan task, click Done to save the scan task and execute the scan.

The new scan task appears in the Scan Tasks list along with any previously defined scan tasks. To monitor the progress of your scan follow the instructions provided in Chapter 11, “Monitoring Discovery” .

Configuring and Running Level 3 Discovery 6

About this Chapter

This chapter provides detailed information about Level 3 discovery. BDNA Discover performs Level 3 discovery on credentialed applications to provide detailed application level data. For example, performing Level 3 discovery on an Oracle database server can provide information on Oracle users, tablespaces, high-water mark utilization, and configuration parameters.

Note: For a summary of what details Level 3 discovery provides and how those details are collected, review "Level 3".

Since credentials and authentication options vary widely by application, Level 3 discovery and its operation are different for each application. This section will focus on performing Level 3 discovery on an Oracle database server because of that application's high technical and financial value.

Overview of Level 3 Discovery Operations

Performing Oracle Level 3 discovery requires routable access to an Oracle database. The general procedure includes:

- Successfully completing Level 1 and Level 2 discovery on the Oracle database server(s)
- Creating a valid Oracle credential
- Configuring BDNA Discover to perform the Oracle Level 3 discovery

Credentials Requirements

Oracle databases, and many other enterprise applications, contain an internal credential system that is used for authenticating against the application. Therefore, it is necessary to create an application credential for BDNA Discover to use for authentication and discovery.

Database Administrators (DBAs) should grant SELECT access to the relevant views and tables for a BDNA Discover credential by running the SQL script (see below) provided by BDNA Discover. When executed by the DBA, this script creates a new Oracle credential and grants it the appropriate privileges. The SQL script also allows the Oracle credential to use OS authentication. The script is located here:

```
$BDNA_HOME/scripts/Administrator/Oracle/create.sql
```

Before use, this script must be edited to supply the proper os_user. Edit the script and change the following to the correct os_user:

```
--define os_user=SCHEMA_NAME_GOES_HERE
```

```
define os_user=bdnaadm
```

Because the script is in plain text, the DBA may view it to see exactly what it will do when run. The details and reasons for the script's actions are discussed below:

Oracle Level 3 discovery does not require DBA access or credentials. The Oracle database credential requires read-only access to various v$ and dba_ views and tables.

The specific views and tables used for Oracle Level 3 discovery include the following:

- `v_$parameter`
- `v_$session`
- `v_$open_cursor`
- `v_$license`
- `v_$instance`
- `v_$option`
- `dba_data_files`
- `dba_extents`
- `dba_tablespaces`
- `dba_tab_partitions`
- `dba_users`

Level 3 credentials for Oracle require two sets of credentials—one for the underlying OS on which the target application is installed, and one for the application itself. This is because BDNA Discover's Oracle Level 3 fingerprints for both UNIX and Windows involve first logging in to the underlying OS using OS-level remote connectivity, and then executing Oracle utilities. So, for example, in order to set up Oracle Level 3 discovery for an Oracle instance running on UNIX, a UNIX Level 2 credential would also be required.

Note: For more information on Level 2 credentials, review "Configuring and Running Level 2 Discovery" on page 57.

Oracle User vs. OS User Database Authentication

BDNA Discover supports two methods of authentication for Oracle databases:

- Direct authentication via an Oracle user (schema) name and password

  or

- Indirect authentication via an appropriately configured OS user

OS user authentication relies on environment variables being configured for the OS user, such that no password is needed by that user to connect to the target Oracle database as a maintenance account such as `sysdba`.

Managing Level 3 Credentials

Level 3 credentials, as with Level 2 credentials and networks, can be created before Scan Tasks are created or during the creation of a Scan Task. Level 3 credentials created either way will be available for use in all future Scan Tasks within the current schema. Whether created before or during Scan Task creation, the process is the same.

Creating a New Credential for Level 3 Discovery

To create a credential using the Scan Setup portion of Scan Administration:

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup in the left navigation pane, click Credentials.
3. In the toolbar under the application menu at the top, click New Credential.

Scan Administration displays the New Credential wizard.

4. For this example, in the Credential Type drop-down list, highlight Oracle Database Server (UNIX) and click Select.

   The New Credential wizard displays fields specific to your selection. For this credential type, Oracle is the only available connection type.

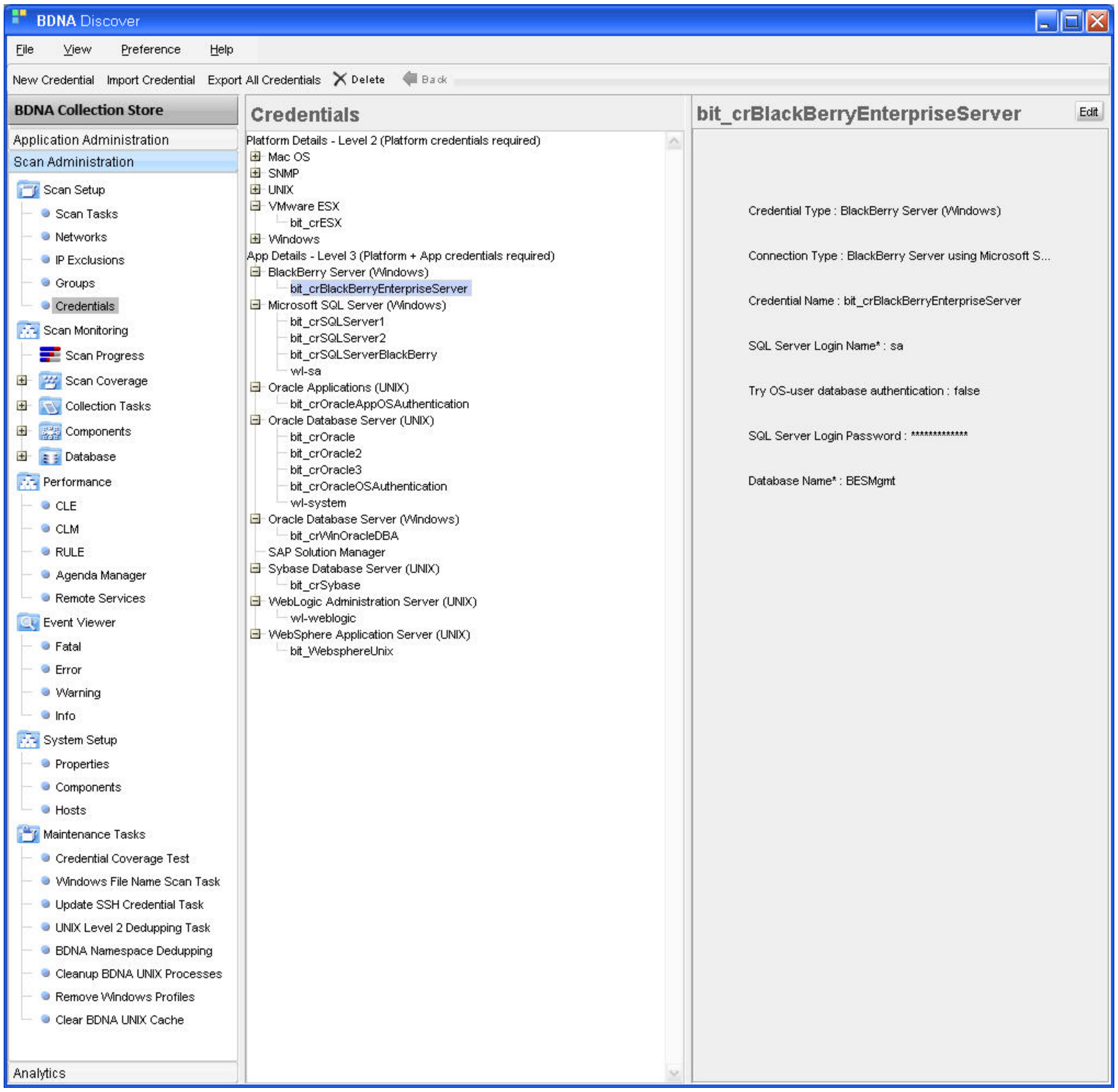
5. Complete the fields as appropriate:

- Credential Name should be a short, descriptive name.
- If Oracle database user authentication is being used, Schema name should be the Oracle user, such as `system` or `sysdba`.
- If Oracle database user authentication is being used, Schema Password and Confirm Schema Password should be the account password.
- If OS-user database authentication is being used, select the checkbox next to Try OS-user database authentication.

6. When the form is completed, click Done to save the credential.

   The credential appears in the Credentials list, under Oracle Database Server—UNIX.

Figure 30: Level 3: Credential Entry


BDNA Discover
File View Preference Help
New Credential Import Credential Export All Credentials Delete Back
BDNA Collection Store
Application Administration
Scan Administration
Scan Setup
Scan Tasks
Networks
IP Exclusions
Groups
Credentials
Scan Monitoring
Scan Progress
Scan Coverage
Collection Tasks
Components
Database
Performance
CLE
CLM
RULE
Agenda Manager
Remote Services
Event Viewer
Fatal
Error
Warning
Info
System Setup
Properties
Components
Hosts
Maintenance Tasks
Credential Coverage Test
Windows File Name Scan Task
Update SSH Credential Task
UNIX Level 2 Dedupping Task
BDNA Namespace Dedupping
Cleanup BDNA UNIX Processes
Remove Windows Profiles
Clear BDNA UNIX Cache
Analytics
Credentials
Platform Details - Level 2 (Platform credentials required)
Mac OS
SNMP
UNIX
VMware ESX
bit_crESX
Windows
App Details - Level 3 (Platform + App credentials required)
BlackBerry Server (Windows)
bit_crBlackBerryEnterpriseServer
Microsoft SQL Server (Windows)
bit_crSQLServer1
bit_crSQLServer2
bit_crSQLServerBlackBerry
wl-sa
Oracle Applications (UNIX)
bit_crOracleAppOSAuthentication
Oracle Database Server (UNIX)
bit_crOracle
bit_crOracle2
bit_crOracle3
bit_crOracleOSAuthentication
wl-system
Oracle Database Server (Windows)
bit_crWinOracleDBA
SAP Solution Manager
Sybase Database Server (UNIX)
bit_crSybase
WebLogic Administration Server (UNIX)
wl-weblogic
WebSphere Application Server (UNIX)
bit_WebsphereUnix
bit_crBlackBerryEnterpriseServer
Edit

Credential Type : BlackBerry Server (Windows)

Connection Type : BlackBerry Server using Microsoft S...

Credential Name : bit_crBlackBerryEnterpriseServer

SQL Server Login Name* : sa

Try OS-user database authentication : false

SQL Server Login Password : *************

Database Name* : BESMgmt

Modifying an Existing Credential for Level 3 Discovery

To modify a credential using the Scan Setup section of Scan Administration:

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup, click Credentials.
3. In the Credentials listing in the middle panel, locate the Level 3 credential to be modified and highlight it by clicking on it.
4. In the top right corner, click the Edit button to bring up the Edit Credential wizard.
5. The Edit Credential wizard performs identically to the New Credential wizard.

Refer to "Creating a New Credential for Level 3 Discovery" on page 96 for more information.

Creating Level 3 Scan Tasks

Configuring a Level 3 scan should be straightforward, since the procedure also uses the New Scan Task wizard discussed in "Configuring and Running Level 1 Discovery" on page 35 and in "Configuring and Running Level 2 Discovery" on page 57. However, the BDNA Discover Administrator should consider whether a new Level 1 and/or Level 2 discovery task will need to be performed in conjunction with the Level 3 discovery task.

You must complete both Level 1 and Level 2 scan tasks on a given set of assets before you can perform Level 3 discovery on them. This is similar to the requirement that Level 1 discovery be executed on a given set of assets before a Level 2 discovery can be performed on them. Refer to "Creating Level 2 Scan Tasks" on page 88 for more information.

In "Configuring and Running Level 2 Discovery" on page 57, the reasons that Level 2 should be performed soon after the Level 1 scan were discussed. Although some of the reasons discussed are less relevant for the class of machines where one would expect to find Oracle (i.e. those servers are not usually DHCP clients and they are less likely to be turned off), it is still possible that the Level 1 and 2 data will become stale prior to the Level 3 scan. Because of this, it is desirable to perform Level 3 discovery reasonably soon after the prerequisite Level 1 and Level 2 scans.

Tip: To improve performance and avoid failed login attempts, it is recommended that credentials be applied to the smallest location possible. Typically, creating groups (i.e. projects) for Level 3 discovery provides the most flexibility and accuracy.

To create a Level 3 scan task:

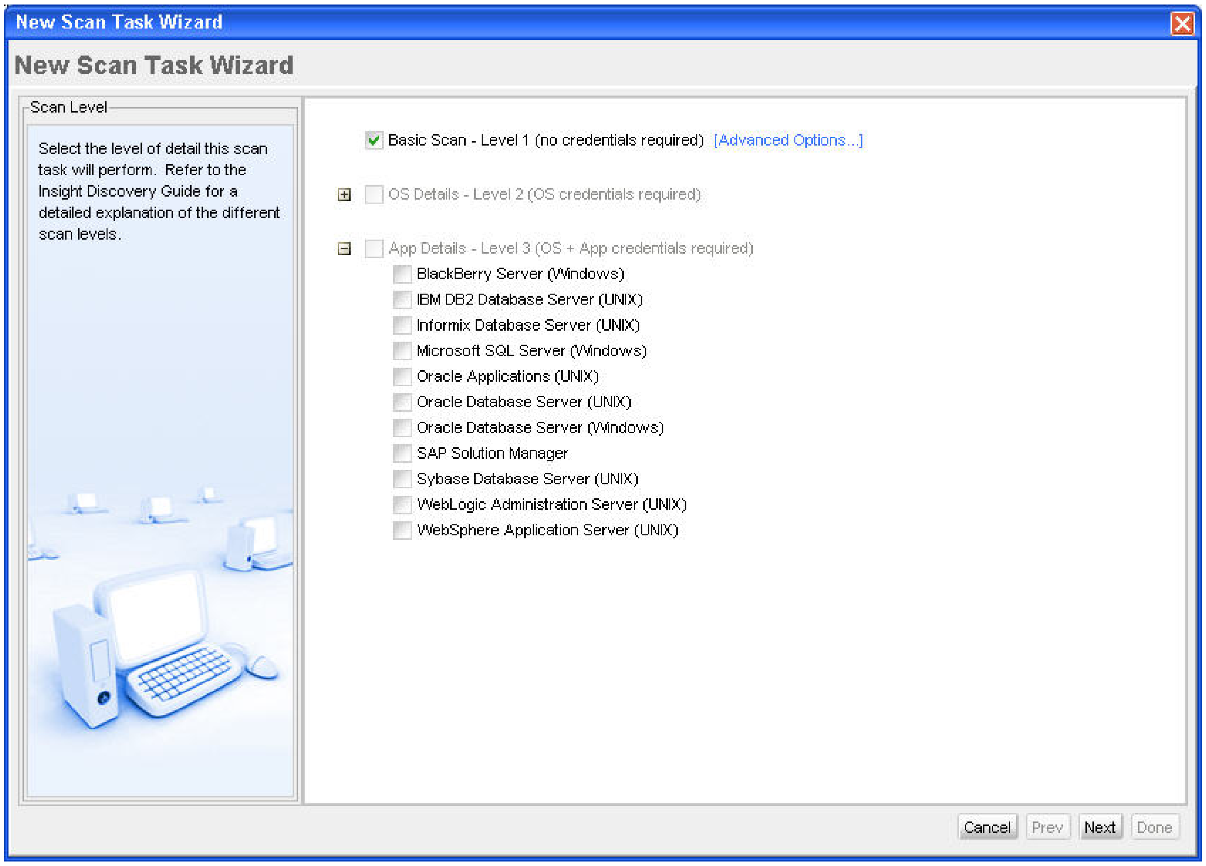
Note: The following procedure assumes that both a Level 2 and a Level 3 credential have previously been created. Note that a credential could also be defined during scan task creation.

---

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup, click Scan Tasks.
3. In the toolbar underneath the application menu, click New Task.

   Scan Administration displays the New Scan Task wizard.

Figure 31: New Scan Task Wizard: App Details - Level 3, Scan Level



Select the level of detail this scan task will perform. Refer to the Insight Discovery Guide for a detailed explanation of the different scan levels.

- [✓] Basic Scan - Level 1 (no credentials required)
- [ ] OS Details - Level 2 (OS credentials required)
- [ ] App Details - Level 3 (OS + App credentials required)
  - [ ] BlackBerry Server (Windows)
  - [ ] IBM DB2 Database Server (UNIX)
  - [ ] Informix Database Server (UNIX)
  - [ ] Microsoft SQL Server (Windows)
  - [ ] Oracle Applications (UNIX)
  - [ ] Oracle Database Server (UNIX)
  - [ ] Oracle Database Server (Windows)
  - [ ] SAP Solution Manager
  - [ ] Sybase Database Server (UNIX)
  - [ ] WebLogic Administration Server (UNIX)
  - [ ] WebSphere Application Server (UNIX)

4. For simultaneous Level 1 and Level 3 scanning, leave Basic Scan – Level 1 checked; otherwise, uncheck it.

5. For simultaneous Level 2 and Level 3 scanning, under OS Details – Level 2, select the checkbox(es) adjacent to the desired platform(s); otherwise, leave all Level 2 boxes unchecked.

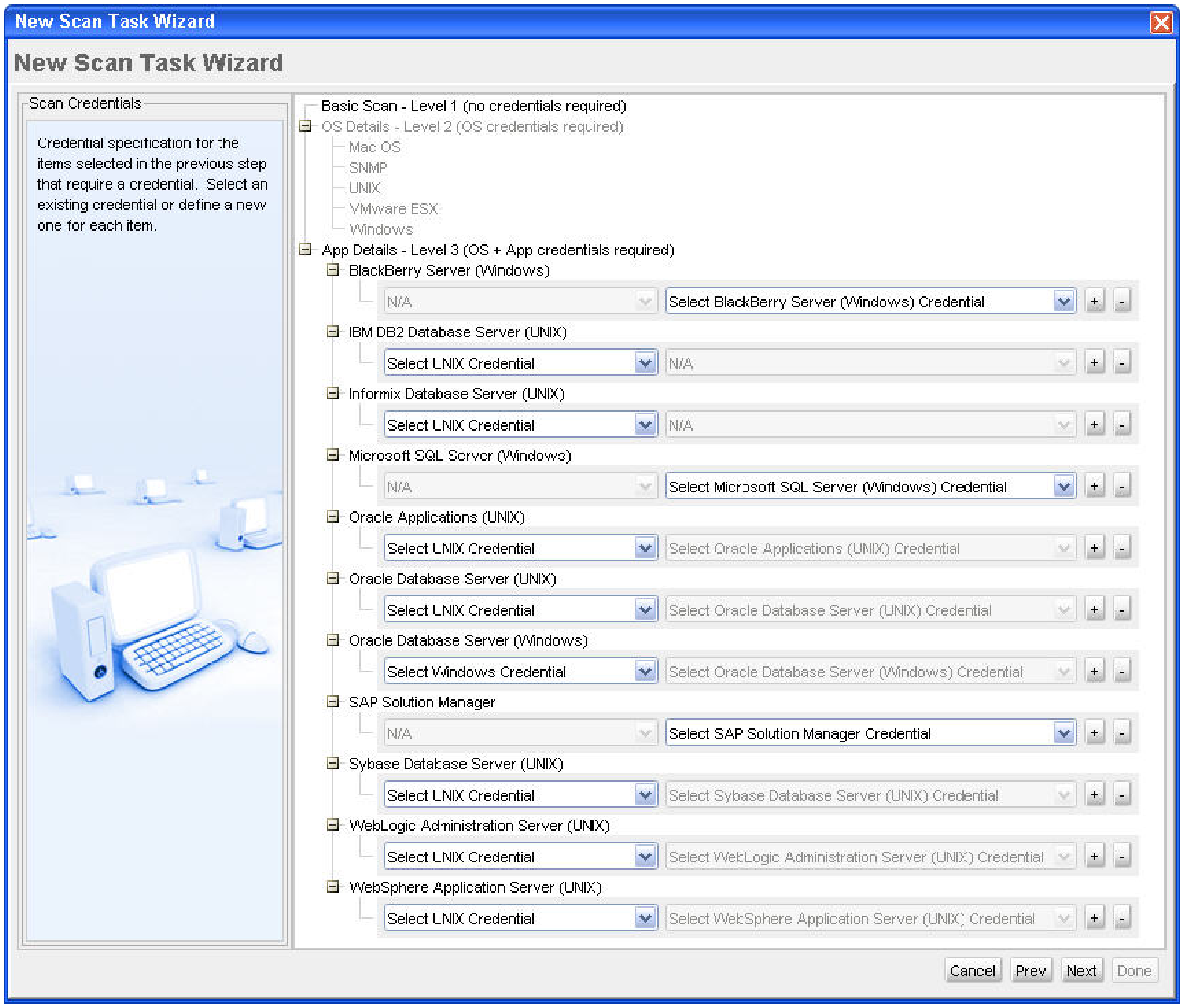
6. Select the checkbox(es) adjacent to the desired platform(s); to select all entries, select the checkbox adjacent to App Details - Level 3.

Refer to "Credentials Requirements" on page 95 for the Level 3 credential requirements for the supported platforms.

7. Click Next.

The next page of the New Scan Task wizard specifies what credentials to use for the scan. The wizard displays one drop-down list for each Level 2 credential type specified on the previous screen (assuming that you specified simultaneous Level 2 and Level 3 scanning in step 5), and two drop-down lists for each Level 3 credential type specified.

Figure 32: New Scan Task Wizard: App Details - Level 3, Scan Credentials



Credential specification for the items selected in the previous step that require a credential. Select an existing credential or define a new one for each item.

- Basic Scan - Level 1 (no credentials required)
- OS Details - Level 2 (OS credentials required)
  - Mac OS
  - SNMP
  - UNIX
  - VMware ESX
  - Windows
- App Details - Level 3 (OS + App credentials required)
  - BlackBerry Server (Windows): N/A | Select BlackBerry Server (Windows) Credential
  - IBM DB2 Database Server (UNIX): Select UNIX Credential | N/A
  - Informix Database Server (UNIX): Select UNIX Credential | N/A
  - Microsoft SQL Server (Windows): N/A | Select Microsoft SQL Server (Windows) Credential
  - Oracle Applications (UNIX): Select UNIX Credential | Select Oracle Applications (UNIX) Credential
  - Oracle Database Server (UNIX): Select UNIX Credential | Select Oracle Database Server (UNIX) Credential
  - Oracle Database Server (Windows): Select Windows Credential | Select Oracle Database Server (Windows) Credential
  - SAP Solution Manager: N/A | Select SAP Solution Manager Credential
  - Sybase Database Server (UNIX): Select UNIX Credential | Select Sybase Database Server (UNIX) Credential
  - WebLogic Administration Server (UNIX): Select UNIX Credential | Select WebLogic Administration Server (UNIX) Credential
  - WebSphere Application Server (UNIX): Select UNIX Credential | Select WebSphere Application Server (UNIX) Credential

8. For each platform specified previously, select the appropriate Level 3 credential from the adjacent drop-down list.

Note: It is also possible to create new credentials at this point. To do so, select Create [*platform*] Credential from the drop-down list.

9. Depending on the platform, your selection of a Level 3 credential in step 8 may activate the drop-down list in the adjacent column; if so, select the appropriate entry from the newly activated list.

10. Use the “+” next to the drop-down list to add a second Level 3 credential to the scan, if desired; use the “-” to remove a credential from a scan (the credential will not be deleted from the system).

11. When you are finished specifying credentials, click Next.

The next page of the New Scan Task wizard specifies the network(s) where collection will occur.

Any networks or groups already defined appears listed on this page. Note that there are also buttons on the bottom of this page to create and import both networks and groups.

Note: For a discussion about how to create and use groups, refer to the *BDNA Discover Administration Tutorial, Module 1*.

---

12. Click the checkbox next to Select to select all networks and groups, or select entries individually.

13. When satisfied, click Next.

The next page of the New Scan Task wizard specifies times when collection will occur.

14. For this example, use the default selection (Start Scan Immediately).

The scan will not actually start until the final page in the wizard.

15. Click Next.

Scan Administration displays the Scan Option page of the New Scan Task wizard.

16. Specify a priority for the scan:

Normal:	Collection happens in the order in which the task was queued, but behind any scheduled high-priority task.
High:	Collection happens in the order in which the high-priority task was queued, ahead of normal-priority task.

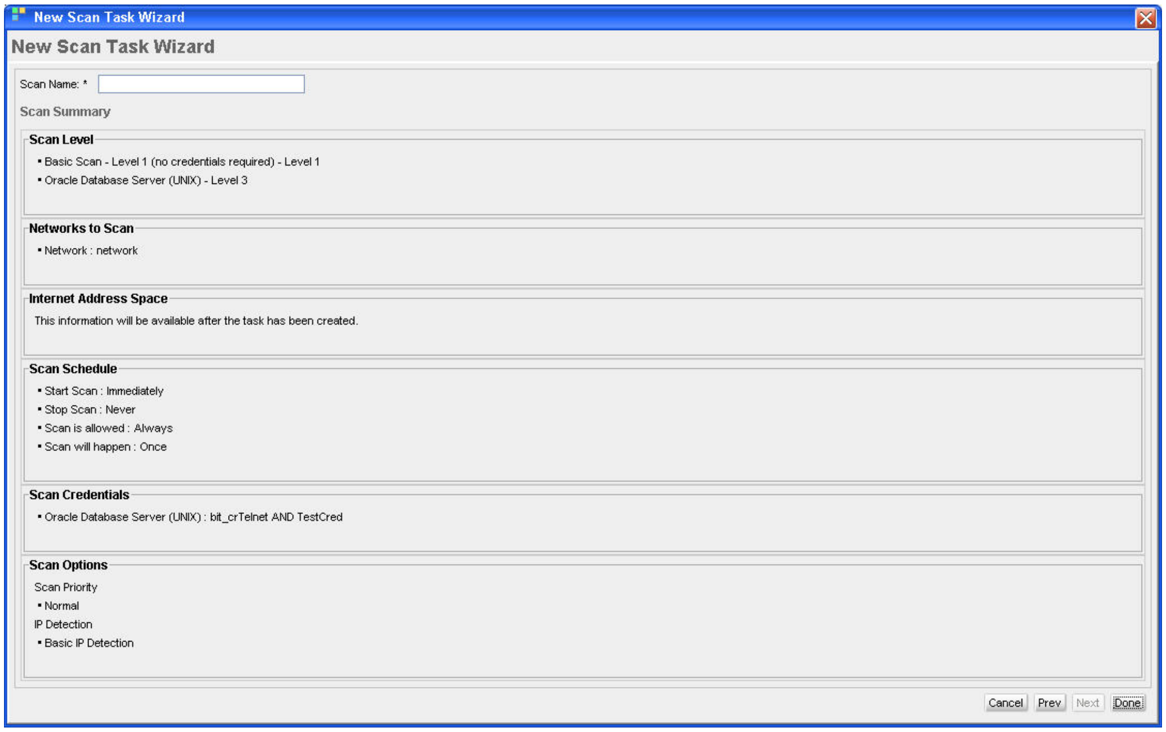
17. Specify IP detection for the scan:

Basic:	Detect active IP addresses using ICMP only.
Rigorous:	Detect active IP addresses using ICMP and TCP SYN.

18. Click Next.

The New Scan Task wizard displays a summary of your previous selections.

Figure 33: New Scan Task Wizard: Scan Summary Page—Level 3



New Scan Task Wizard

Scan Name: *

Scan Summary

Scan Level
- Basic Scan - Level 1 (no credentials required) - Level 1
- Oracle Database Server (UNIX) - Level 3

Networks to Scan
- Network : network

Internet Address Space
This information will be available after the task has been created.

Scan Schedule
- Start Scan : Immediately
- Stop Scan : Never
- Scan is allowed : Always
- Scan will happen : Once

Scan Credentials
- Oracle Database Server (UNIX) : bit_crTelnet AND TestCred

Scan Options
Scan Priority
- Normal

IP Detection
- Basic IP Detection

19. On the Scan Summary page, assign a unique and descriptive name for the new scan task.
20. Examine the overview of all of the configuration settings chosen on the previous pages of the Wizard, and make sure they are as intended; to return to a previous page and reset a configuration item, click Prev.

    or

    When you are satisfied with the configuration settings for the new scan task, click Done to save the scan task and execute the scan.

    The new scan task appears in the Scan Tasks list along with any previously defined scan tasks. To monitor the progress of your scan follow the instructions provided in Chapter 11, “Monitoring Discovery” .

Windows SQL Server Level 3 Discovery Requirements

Microsoft SQL Server level 3 discovery requirements:

- The SQL server instance running on the target host can be configured to use Windows Authentication Only, SQL Server Login, or Mixed Mode Authentication.

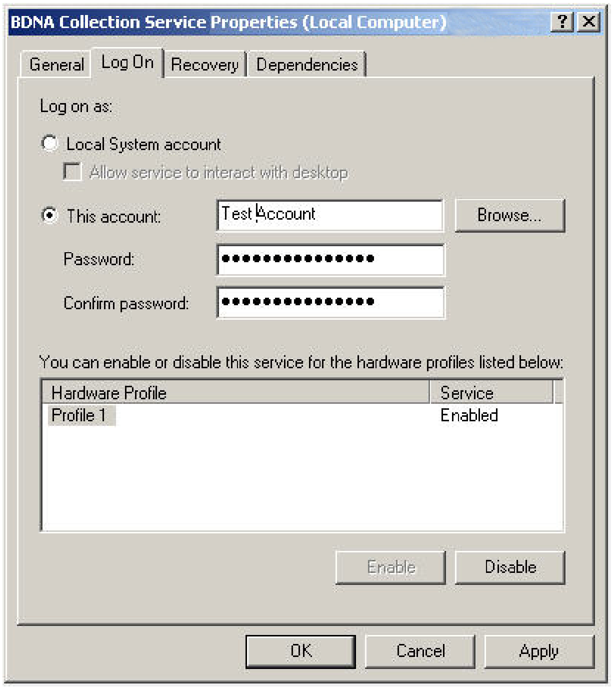
  If Windows Authentication Mode or Mixed Mode is used, BDNA Discover can accept only one Windows user account.
- In BDNA Discover Scan Administration the following two credentials must be created and included in the Scan Task:
  - Level 2 > Windows: This credential may either be an administrative account or a non-administrative account if the BDNA Discover rollout scripts have been applied to the target host.
  - Level 3 > Microsoft SQL Server: You can specify a SQL server login name, OS Authentication, or both.

If you specify OS Authentication, you need to set up an identical account with the following steps:

1. Create a local user account in the WinCS machine with identical name and password.

   For example, if domain user account BDNACORP\Test is used for scanning, create a local account Test from the WinCS machine with identical password.

Figure 34: BDNA Discover Collection Service Properties Dialog Box



Log on as:

Local System account

Allow service to interact with desktop

This account: Test Account

Password: •••••••••••••••

Confirm password: •••••••••••••••

You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

2. Grant this new, local, user account with local administrator privileges from the WinCS machine.
3. Register WinCS to be start up using this local account, and then restart WinCS services

   Now it is ready to scan.

- When creating a MS-SQL Server native user on the SQL Server instance, the user login account must have the following SQL Server privileges:
  – SQL user can be assigned to any default database
  – SQL user can be assigned to any default language
  – SQL user does not need to be part of any SQL Server roles

Troubleshooting Discovery 7

About this Chapter

Occasionally, things can go wrong. What to do if they do—and how to know that they have in the first place—is discussed in this chapter.

Troubleshooting for Level 1

Because BDNA Discover is composed of many moving parts, it is not always easy to troubleshoot if something goes wrong. Consider the following:

- BDNA Discover is distributed.

  In large installations, BDNA Discover may be distributed across 2, 5 or perhaps even 20 different machines. If something goes wrong, the BDNA Discover Administrator must have a method for determining where to look first.

- BDNA Discover uses Oracle.

  BDNA Discover uses Oracle for its database repository. It is possible that a problem may be due to an Oracle failure or misconfiguration.

- BDNA Discover runs on Linux.

  BDNA Discover component machines run Linux. BDNA Discover often pushes Linux to the limit, so some problems may be traceable back to the operating system.

- BDNA Discover uses Windows and WMI.

  The Windows Collection component runs on Windows to perform WMI-based discovery. Windows may have problems which can make troubleshooting very difficult for applications running on Windows.

- BDNA Discover is a network tool.

  BDNA Discover will most likely be touching every device on the enterprise, because of its very nature. Furthermore, BDNA Discover needs almost complete access to these machines. A misconfigured router, switch, or firewall could manifest in BDNA Discover by giving bad results. It can be difficult to trace suspect results to the individual device that caused them.

- BDNA Discover has security implications.

  Since BDNA Discover needs nearly complete access to machines, there are many security implications. Also, BDNA Discover scanning activity can appear very similar to an attack. It becomes important to notify the organization's security and network administrators before doing a scan, to avoid a panic and shutdown of the BDNA Discover discovery activities by people who might otherwise suspect a malicious attack.

- BDNA Discover is highly configurable.

  BDNA Discover can be configured in many different ways. Because of this, errors in BDNA Discover configuration can cause many different types of problems, which can be difficult to root out.

- Consider other factors.

It is a common syndrome for a new enterprise software rollout, especially one with the scope of BDNA Discover, to be regarded as the source of every problem on a network. This scenario typically occurs during and immediately after a scan. Carefully evaluate the array of factors that can impact network performance before concluding that the issue is attributable to BDNA Discover software.

While any single person cannot be expected to be an expert on all of the above, even when everything goes correctly, the BDNA Discover Administrator will probably still need to understand at least a little about every third party technology mentioned above.

No Substitute for Experience

After gaining some experience using BDNA Discover, troubleshooting will become much easier. In the beginning, it is not always easy to know what success looks like versus failure. Even after becoming somewhat experienced with BDNA Discover, it is not always possible to know what to expect the first time scanning a particular network. This is because there will be no baseline against which to compare the results.

This guide attempts to reduce the learning curve of BDNA Discover. But simply put, in terms of troubleshooting BDNA Discover, there is no substitute for experience.

BDNA Discover Support Can Help

The BDNA Discover Knowledge Base contains much useful information. It may already have the answer to many issues and questions. It can be accessed from https://bdna.service-now.com/support/kb_find.do. Our support professionals have years of experience with the product and they are available via the support portal at https://bdna.service-now.com/support/, email, or phone, in case the knowledge base cannot answer a particular question.

General Tools for Troubleshooting

BDNA Discover provides many ways to gain information about what is going on or what has happened during a scan. The following are tools BDNA Discover provides to help troubleshooting efforts:

- Inventory Application

  Assuming something didn’t go catastrophically wrong and halt a scan before it completed, the first place a problem may become visible is from BDNA Discover. After a scan completes, always login to BDNA Discover and look around to see if everything looks normal.

- Scan Administration

  Many problems can be diagnosed via Scan Administration, both during and after a scan. Perhaps the most import tool it provides is the Event Viewer, which enables quick viewing of errors, including fatal errors that BDNA Discover has experienced. The BDNA Discover Administrator should always look at the Event Viewer after a scan has completed.

- Log Files

  Every machine running a BDNA Discover component has a directory where BDNA Discover logs practically every event that occurs during a scan. It is recommended that the BDNA Discover Administrator look at these log files during a scan, but also after the scan has been completed, to see if anything has gone wrong.

- Interactive Shell

  The script `bdna.sh` provides an interactive shell which can be used to diagnose certain problems.

The following are some third party tools which can provide insight into a problem with a BDNA Discover scan:

- Linux Log Files

  Occasionally, Linux problems will occur. Look for more about these in the syslogs in `/var/logs/`.

- SQL*Plus and Oracle Enterprise Manager

  If there is a problem with Oracle, these Oracle tools can be very helpful in diagnosing it. Oracle log files are also useful for investigating Oracle related issues.

- Windows Event Viewer

  If there is a problem with the Windows Collection component, errors may show up in the Windows Event Viewer.

- Various Command-line Tools

  SSH, Telnet, Nmap and other command-line tools can all help in diagnosing all sorts of problems related to networks generally and BDNA Discover scanning in particular. BDNA Discover also provides diagnostic tools such as credential and IP range checkers.

Note: These BDNA Discover diagnostic tools are covered in detail in Chapter 13, “BDNA Discover Utility Reference” .

Troubleshooting for Level 2

Solving Level 2 Issues

It may seem counterintuitive, but solving problems experienced with Level 2 scanning is considerably easier than solving Level 1-related issues, because Level 1 touches many more hosts and ports and uses significantly more protocols to do so. Simply put, fewer things can go wrong at Level 2. And when they do, there are fewer places to look for the cause.

Three types of things can go wrong at Level 2:

- Rarely, a Level 2 scan will cause a negative impact on a target host.
- A more common problem is that Level 2 access will fail for some or all target hosts.
- Possibly the most difficult Level 2 problem to discover is when Level 2 only partially completes for some hosts.

Each of these is discussed in detail below.

Level 2 Negative Impact on Target Host Issues

BDNA Discover is a mature product, which has been extensively tested, both in the lab and in real enterprises. It is extremely rare that BDNA Discover would cause any adverse impact on machines being scanned at Level 2. However, in rare cases BDNA Discover has been the cause of a problem on a host it has scanned. More often, BDNA Discover exposes an otherwise unknown problem with another piece of software.

The following is a case study of a specific instance where a Level 2 scan adversely impacted a machine, how the problem was diagnosed, and the solution.

- UNIX file find overloads certain hosts

UNIX Level 2 discovery uses the `find` command to search through file systems, looking for files of specific patterns. For a machine that is already underpowered and stretched to its limit, this can push it over the edge. In this instance the system administrator of the machine in question is best positioned to diagnose the problem.

The system administrator should capture the output of commands such as `top` and `ps` in order to verify that BDNA Discover's file find is causing the problem. While it may be tempting for the BDNA Discover Administrator to use the credentials which have been supplied for Level 2 access and SSH or Telnet into the machine to diagnose the problem independently, in no case should this be done without the explicit permission of the machine's system administrator. Those credentials were provided for the purpose of performing a BDNA Discover Level 2 scan, not to do system administration. In this way, the BDNA Discover Administrator may stay on good terms with an organization's system administrators, in spite of the rare problem.

Solution: Exclude this machine from future scans by putting the IP address for the machine in an Exclusion List.

Note: To learn more about Exclusion Lists, refer to "Creating IP Exclusions (Optional)" on page 55.

Level 2 Access Failures

When troubleshooting Level 2 access problems, it should be noted that BDNA Discover will only attempt a Level 2 scan on target machines that can be typed and are specified for Level 2 access. Level 1 scans should be viewed to ensure that the machine in question was typed successfully before other fact finding is undertaken, in the event of a suspected Level 2 access failure. The BDNA Discover Administrator should, of course, check that the machine had indeed been specified for Level 2 access in that scan, as well.

If Level 2 access fails, the causes generally fall into one of the following categories:

- Invalid Level 2 credentials

  The BDNA Discover Credential Checker can be useful in diagnosing credentials-related issues. Additionally, the Analytics application in BDNA Discover contains a Credential Usage History (located in the Administrative folder) that lets you see which credentials worked and which ones didn't.

- Problems with network access to the target systems

  For example, a firewall is between the BDNA Discover system and a target system that blocks a network protocol required for collection, such as SSH or WMI.

- Network and/or target system latency exceeding the BDNA Discover collection timeout period
- A known product limitation, such as SSH, Telnet, and RSH credentials defined in the same credential set

  This is a known limitation that can cause a Level 2 collection access failure.

Note: For more on using more than one network protocol to scan a single IP, including a workaround for this issue, refer to "Credential Precedence" on page 87.

- Lock out problems on the target machine

  Some machines are set up to lock out a particular user after a number of failed login attempts. This could cause a failure in Level 2 access if BDNA Discover is attempting to use multiple credentials in one scan.

Discovering Level 2 Access Failures

In order to solve Level 2 access problems, the BDNA Discover Administrator must first notice that a problem has occurred. While this can be as easy as seeing that this month there is only a 10% success rate for Windows Level 2 discoveries where last month there was a 90% success rate, it is not always that obvious. For example, what about a small decrease in success rate, such as from 90% to 85% success?

Tip: Saving the Inventory Export files is the best way to keep statistics about previous scans outside of the BDNA Discover System.

It is much harder to notice a problem when the BDNA Discover Administrator has no prior experience with the environment being scanned. But after gaining more experience with more environments, it will become much easier to identify a bad success rate for Level 2 access.

Although aiming for as complete coverage as is possible, it might not be the BDNA Discover Administrator's intention to have 100% coverage. Whether complete coverage is a firm goal will be a function of the scan's original objective. Is the objective to consolidate servers and reduce the licensing cost of applications such as Oracle running on UNIX? Then perhaps a 10% success rate for Windows XP is perfectly acceptable. Is the objective to discover how up-to-date and complete the coverage is for anti-virus software? Then a 10% success rate for Windows XP is totally unacceptable.

Tip: Do not fall into the trap of seeking success for its own sake. Troubleshooting costs time and money. Keep in mind the overall objective, and spend the effort where it can be expected to reap the most rewards.

Finding Patterns in a Complete Access Failure

The first step in finding the cause of any failures is to look for patterns in the failures. Fortunately, finding patterns in discovered data is exactly what BDNA Discover was designed to do. A state-of-the-art tool that can help is readily available—BDNA Discover.

When performing a Level 2 scan, checking the Level 1/Level 2 summary reports inside the "Scan Summary" folder in the Inventory Consolidation application should be part of the standard procedure. These will provide the overall percentages for both UNIX and Windows Level 2 coverage.

Troubleshooting a Level 2 complete access failure problem:

Assume that the "Windows L1/L2 Summary" report inside the "Scan Summary" folder has been checked, and that a discovery was made of only a 60% success rate for Windows Level 2.

1. The next thing to do is map the percentages by the type of Windows machine:

  1.1. Go to the "Operating Systems" folder in the Inventory Consolidation application.

  1.2. Double-click the "Windows Operating Systems" report.

  1.3. Put the report into table mode by clicking the "Table" button in the menu bar at the top of the application.

  1.4. Obtain a baseline of all possible Windows Level 2 targets, by OS Type.

  To get this, right-click anywhere on the "OS Type" column, and choose "Rollup column" from the menu that appears. A two-column report appears: the first column is the OS Type and the second column is a count.

2. Open a second instance of the Inventory Consolidation application, leaving open the original window with the rolled up report.

Note: While it is possible to export this rolled up report in order to create the next report and compare the two, this approach is simpler.

---

3. In the second instance of BDNA Discover, again go to the “Operating Systems” folder and double-click the “Windows Operating Systems” report, then put it into table mode.
4. This time, however, the goal is to look only at machines where Level 2 was successful.

   To do this:

   4.1. Find the column titled “Level 2?” at the far right of the report and right-click any “Y” value from this column.

   4.2. In the menu that appears, choose “Filter by this value”.

   4.3. Right-click in the “OS Type” column, and choose “Rollup Column” from the menu that appears.

   Another two-column report, this time with smaller numbers in the second column, appears.
5. Now compare the first rolled-up report with the second and look for any obvious patterns.

   Look for a complete or near-complete failure for any Windows type, such as no Level 2 for any Windows 2008 Servers, or a 5% success rate for Windows 2012.

Working with the Credential Usage History Report

Note: This report does not work on blended schemas or FactBase schemas because they do not contain Agenda Manager transactional tables.

Located in the Administrative folder of the Analytics application for Collection Store, this report has a row for every IP address where a credential was defined. If the IP address has multiple credentials defined, is part of multiple scan tasks, or the credentials were tried repeatedly, all of the details are captured in the report. This is very useful when multiple credentials are defined for an IP address in a scan task, Level 2 succeeded, and the administrator wants to know:

- Which credentials worked for Level 2 collection?
- Which credentials didn’t work for Level 2 collection?
- Which credentials were never attempted for level 2 collection? (This happens if multiple credentials are defined for an IP address, but one succeeded before BDNA Discover had a chance to try the rest.)

The report also shows the scan task as well as the collection task ID for each row. The latter is useful to dig deeper into the CLE log files by correlating the collection task ID.

Finding Patterns in a Partial Access Failure

Obviously, tracking down complete failures of any type is the easy case. Partial failures are more difficult to track down. But here again, BDNA Discover’s Inventory Consolidation application can be an invaluable tool.

In the event of a partial failure, one approach would be to try something similar to what was just described.

Troubleshooting a Level 2 partial access failure problem:

Assume that there was a 10% success rate for Windows XP in a Level 2 scan, as determined by viewing the “Windows L1/L2 Summary” report.

1. Go to the “Windows Operating Systems” report, and filter on “OS Type” “Windows XP”.
2. Perform a rollup on “Patch Level/Service Pack”.

   This will be the baseline report.
3. Go to a second instance of BDNA Discover and go to the “Windows Operating Systems” report, filtering first on “OS Type” equals “Windows XP”, then on “Level 2?” equals “Y”.
4. Perform the rollup on the “Patch Level/Service Pack” column.
5. Compare the results of the two rolled-up reports.

   Look for patterns such as complete or near-complete failures for any Service Pack. For instance, if there was 95% coverage for Windows XP with Service Pack 1, but 0% coverage for Windows XP with Service Pack 2, then the problem has probably been found, or at the very least, some basic knowledge about the problem has been gained.

   Assuming that the Patch Level/Service Pack breakdown does not provide any answers, another way to analyze the failures is to look at how the failures break down by network:

   5.1. Roll up the “Windows Operating Systems” report on “Networks” to establish the baseline report.

   This assumes there are no troubles with Windows credentials.

   5.2. Filter “Windows Operating Systems” on “Level 2?” equals “Y”.

   5.3. Roll up on “Networks” and compare the results to the baseline.

   5.4. If there are too many networks to compare the reports in the BDNA Discover system, export each report and compare them using Excel.

   Look for patterns—places with complete or near-complete failure of Level 2 access.

Remedying Failures

Assuming some kind of pattern to the failures exists, what happens next? Probably the most the BDNA Discover Administrator can do is manually test the credentials against those networks or OS sub-types that showed patterns of failure. If no obvious credentials problems are detected, then follow up with the system administrators for the machines in question to attempt to diagnose and correct the problem.

Note: Refer to “Creating IP Exclusions (Optional)” on page 55 for more information.

For information on how to manually test the credentials supplied to BDNA Discover, refer to “Testing Credentials” on page 78.

As for reporting the results, a good approach is to present the rolled-up reports that were used to discover the patterns. Then ask questions like, “We got 98% coverage on this network for Windows XP, but 0% coverage on this network. What’s the difference between these two networks?”

Incomplete Level 2 Data

As stated above, possibly the most difficult problem to discover is when it appears the scan's overall Windows Level 2 coverage was quite good, but when examined more closely, complete Level 2 data is not present for some of the machines. This is one type of problem that will probably not be detected unless it is addressed proactively.

To understand how BDNA Discover can have incomplete Level 2 data, it is important to understand how BDNA Discover performs discovery. BDNA Discover makes multiple connections to each machine to gather all of the Level 2 data. For example, for Windows Level 2, the initial connection will discover the hardware details of the scan and capture the Add/Remove entries. BDNA Discover will then go back at a later time and individually query applications for which it has a fingerprint. With an application like Internet Explorer, for example, BDNA Discover will make a second connection to a target machine to fingerprint IE. This second connection could occur hours after the initial connection.

Success on the initial connection will cause the "Level 2?" column in the summary reports to read "Y", but there will be no obvious indication that the second connection failed. So how would this information be obtained? Again, BDNA Discover's Inventory Consolidation application is perfectly suited for this type of fact finding.

Discovering Incomplete Level 2 Data

The following technique for discovering where Windows Level 2 resulted in incomplete Level 2 data takes advantage of the fact that all or nearly all Windows machines have Internet Explorer installed on them. (Unfortunately there appears to be no equivalent procedure for UNIX machines, although in practice incomplete Level 2 data will occur more for Windows hosts than for UNIX hosts.)

Since all Windows machines have Internet Explorer, all that is necessary is to count the total number of Windows Level 2 successes and compare it to the total count of Internet Explorer installations discovered. Counting Windows Level 2 successes is covered above. Counting Internet Explorer installations is just as easy.

To count the number of IE installations:

1. In the Inventory Consolidation application, go to the "Applications" folder, double-click the "Fingerprinted Applications on all OS" report and put it into table mode.
2. Roll up on the "Application Name" column or filter where "Application Name" equals "Internet Explorer".

   Either method will provide a count to compare against the baseline of the total number of Windows machines where BDNA Discover reports Level 2 success.

Tip: The BDNA Discover Application Administration component is also a good place to look for failed Windows Level 2 tasks.

Causes of Incomplete Level 2 Data

Assuming it is now known that there is incomplete Level 2 data, the cause of the problem and whether or not this is worth worrying about must be determined.

In the example, since the second connection to discover Internet Explorer occurs well after the initial connection, the most likely cause of the failure is that the target machine was no longer listening at the same IP address as at the time of the first connection. In other words, the machine was either not on the network any longer, or it uses DHCP, its lease expired, and it was issued an IP address different than the one it had been issued previously.

Machines can go off the network for a couple of reasons. Perhaps it is a laptop, and the user takes it home at the end of the day. Assume the scan started at noon but completed at 10 PM. By the time the system fired the Internet Explorer fingerprint task for that laptop, it was already miles from the network. Or perhaps, with the same scan schedule, users in the environment have a tendency to shut their machines down at the end of the day. (The reason that this problem occurs less frequently with UNIX hosts should be obvious now.)

Another possible cause of the second connection failing is that the scan was prematurely stopped, before it had a chance to complete. This is almost always due to operator error. Care should be taken to let scans complete before shutting down in order to avoid this problem. But if the problem should occur, this is how it might be determined that the machine was shut down before the scan completed.

Another possible reason for incomplete data collection could be that a different system on the network has taken the same IP address as the target system (via DHCP) in the middle of a Level 2 scan on the target system.

Preventing Incomplete Level 2 Data

To prevent the problem of machines going off the network at the end of the day, it is helpful to construct the scan schedules such that all discovery occurs during a single workday. Of course, what constitutes working hours will be different from environment to environment. Keep in mind that even when taking this precaution, it is still possible to have machines go off the network at odd hours, but at least an attempt will have been made to minimize this type of Level 2 failure.

Also, remember that care should be taken to let scans complete before shutting the system down in order to avoid that type of incomplete Level 2 scan.

Configuring BDNA for Secure Operations 8

About this Chapter

This chapter reviews the various security requirements of the Scan Administration application and discusses configuration of Scan Administration for secure operations.

Level 1 Network Security Requirements

During Level 1 discovery, the Scan Administration application uses network scanning techniques similar to those used by vulnerability assessment tools. This makes the Scan Administration application appear very much like one of these tools. However, since the Scan Administration application typically scans far fewer ports than a vulnerability assessment tool, any risk posed would generally be lower.

Port Scanning

During Level 1 discovery, the Scan Administration application first performs an ICMP ping sweep of a network space to determine which devices are alive. If ICMP is not enabled, BDNA may also attempt to send TCP SYN packets to port 23. Once a device is determined to be alive, BDNA will port scan the device to get an initial list of ports that may be of interest for typing the device. The current list of ports and services that may be scanned is contained in the file `$BDNA_HOME/nih/Nmap/src/nmap-services`.

Potential Risks of Port Scanning

Port scanning is a common technique in network management. The security risk posed by port scanning is limited. If an attacker were to intercept a port scan data stream, they would be able to find out which ports on a given machine were open. This information could provide some benefit to an attacker. However, this scenario would generally occur only where an attacker does not have direct access to the system the data concerns.

For example, consider a situation where the Scan Administration application is behind a firewall scanning across a circuit to a target system protected by another firewall. The firewall on the target network would be configured to allow BDNA traffic to pass through it. In this situation, if an attacker intercepted the stream, they would be able to determine the open ports and yet would not have direct access to that system.

In situations where the attacker has direct access to the target machine, an attacker could use the BDNA data stream to eliminate the need to perform their own port scans. However, if the attacker has direct access to the system, the difficulty of intercepting the BDNA data stream would be considerably higher than simply running their own tools. Therefore, the data stream would have little if any value to an attacker in this scenario.

The risk associated with the Scan Administration application's performance of port scanning is mitigated by the presence of a firewall on the target network.

Port Fingerprinting

Subsequent to the port scan, BDNA will proceed to type what it deems are ports of interest. Specifically, it will only pursue ports that it knows to be open, based upon the scan above. The current ports of interest to BDNA, with descriptions of its activities upon them, are listed in Table 6 on page 116.

Table 6: Ports of Interest to BDNA Scan Administration

Service	Port(s)	Notes
ftp	21	BDNA attempts to open an FTP session to inspect the response. The session is subsequently closed.
ssh	22	BDNA attempts to open a SSH session to inspect the response. The session is subsequently closed.
smtp	25	BDNA attempts to initialize a SMTP session to inspect the response. The session is subsequently closed.
pop2, pop3	109, 110	BDNA attempts to open a POP session to inspect the response. The session is subsequently closed.
imap2, imap3	143, 220	BDNA attempts to open an IMAP session to inspect the response. The session is subsequently closed.
http, http-proxy	80, 8000, 8008, 8080	BDNA sends a HTTP 1.1 GET request and inspects the response.
ldap	389	BDNA attempts to bind to the LDAP server, request to the root DSE of the LDAP server and inspects the response.
https	443, 8443	BDNA sends a HTTPS 1.1 GET request and inspects the response.
ms-sql-s	1433	BDNA sends an MS SQL Server pre-connection UDP query and inspects the results.
oracle	1521, 1526	BDNA uses the Oracle utility TNSPING to attempt to open a TNS connection (SQL*Net) to an Oracle listener. The response is examined.
tivoli-node, tivoli-endpoint	9000, 9490–9500	Tivoli agents running on these ports respond like HTTP servers. BDNA sends a HTTP 1.1 GET request and inspects the response.

Potential Risks of Port Fingerprinting

Like port scanning, port fingerprinting is a common technology. If an attacker were to intercept all of the data streams during this process, they could have a fairly comprehensive picture of the services running on the target system from which the Scan Administration application is gathering information. As with port scanning, the risk is that an attacker may gain access to a data stream from a target system to which the attacker would not have access. If the attacker did have direct access, the effort of intercepting BDNA’s data stream would not be worth the time required to do so. Moreover, since the Scan Administration application fingerprints a small number of ports, the amount of information that an attacker might capture is well bounded and well understood beforehand.

Perhaps the most important consideration with respect to port fingerprinting risk involves access. Specifically, the Scan Administration application neither requires nor requests special access via agents or ports during Level 1 discovery. Aside from routable access to the device, there is no information BDNA captures that could not be easily captured by someone with network access to the target system using off-the-shelf and/or freeware tools.

Level 2 Network Security Risks

The Scan Administration application uses operating system credentials during Level 2 discovery to extract data from target systems. The network security risks of Level 2 are driven far more by the method used to connect to the target system than by the Scan Administration application. The potential risks at Level 2 are well known and easily managed. The network security risks are best understood by examining UNIX and Windows separately.

UNIX Level 2 Network Security

The Scan Administration application uses standard UNIX operating system user accounts to scan target systems at Level 2. BDNA supports several different connection methods for connecting to remote UNIX systems. The following connection methods are recommended, in order of highest to lowest security:

- SSH with public/private keys
- SSH with username and password
- telnet with username and password

Note that telnet is not a secure connection method. All usernames, passwords, commands and responses are passed in clear text when using telnet.

All commands executed by the Scan Administration application, and all results returned by the target system, traverse the connection method chosen by the user. Therefore, if an insecure method is chosen, the data stream would be available for interception. Since this data could provide far more detailed information about a given system than Level 1 data, BDNA strongly recommends using secure connection methods.

Potential Risks of Level 2 on UNIX

Level 2 data provide a very in-depth picture of the target system. The information captured includes configuration information, patch levels, peripherals, installed software, serial numbers, file system layout and more. Therefore, these connections should be protected carefully.

If an insecure protocol such as telnet were used, there is some risk of an attacker intercepting the data stream. However, an attacker sophisticated enough to intercept a data stream would most likely not be interested in the BDNA data stream. Rather, the attacker would be interested in harvesting the username and password from the data stream and connecting directly to the system to compromise it on their own. It should be noted that the user account for BDNA does not require “root” access; therefore, any subsequent attack would be limited to the privileges of the BDNA user account.

However, on UNIX systems, the various secure protocols, such as SSH, are well known and widely distributed. They provide a significant number of strong encryption algorithms to protect data streams. The risks associated with BDNA with respect to Level 2 collection are therefore no higher than the risk associated with the general use of secure protocols such as SSH.

Windows Level 2 Network Security

BDNA uses the Windows Management Instrumentation (WMI) protocol during Windows Level 2 collection. WMI supports several levels of security on the network connections, including encryption of all communications.

All commands executed by the Scan Administration application, and all results returned by the target system, traverse the connection method chosen by the user. Therefore, if an insecure method is chosen, the data stream would be available for interception. Since this data could provide far more detailed information about a given system than Level 1 data, BDNA strongly recommends using secure connection methods.

Potential Risks of Level 2 on Windows

Level 2 data provide a very in-depth picture of the target system. The information captured includes configuration information, patch levels, peripherals, installed software, serial numbers, file system layout and more. Therefore, these connections should be protected carefully.

If an insecure WMI authentication level such as Connect or Call were chosen, there is some risk of an attacker intercepting the data stream. However, an attacker sophisticated enough to intercept a data stream would most likely not be interested in the BDNA data stream. Rather, the attacker would be interested in harvesting the username and password from the data stream and connecting directly to the system to compromise the system on their own. It should be noted that the user account for BDNA does not require “Administrator” access; therefore any subsequent attack would be limited to the privileges of the BDNA user account. This provides a limited amount of additional defense.

Additional risks of using WMI relate to the protocol itself. WMI is a Windows-only protocol which is not used outside the Windows environment. Therefore, as with other well known Microsoft products, WMI may be found at some future time to be vulnerable to attack.

However, the existing WMI protocol provides sufficient levels of authentication and encryption to ensure adequate security.

Configuring the BDNA Discover Interface for SSL

By default, the BDNA user interface runs over the HTTP protocol. The direct URL for accessing the BDNA user interface is:

```
http://hostName:8080/bdna/AppConsole.jnlp
```

This URL is often formed by redirection from a base URL, such as `http://hostname` or similar. However, organizations often prefer to run the BDNA user interface over a secure protocol, such as HTTPS. To configure the BDNA user interface to run over HTTPS, the underlying Apache Tomcat server which provides the servlet support to the BDNA user interface must be reconfigured.

To enable SSL in Tomcat, follow the directions provided by Apache at `http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html` to generate a keystore file.

To Enable SSL in BDNA

The following is a step-by-step process for enabling SSL in BDNA:

1. Log on to the BDNA system as the BDNA owner.
2. Be sure that BDNA is shut down and the Application Agent is halted.
3. Generate a self-signed keystore using the keytool executable:
   - **3.1.** `$BDNA_HOME/nih/jdk/jre/bin/keytool -genkey -alias bdnakey -keyalg RSA`
   - **3.2.** Enter a keystore password.

     Tomcat recommends the password to be “changeit” in its configuration.
   - **3.3.** Enter a contact name.
   - **3.4.** Enter a contact organizational unit.
   - **3.5.** Enter a contact organization.
   - **3.6.** Enter a location city.

3.7. Enter a location state.

3.8. Enter a location country.

3.9. Enter yes to confirm previous entries.

3.10. Enter a key password.

Tomcat recommends the key password to be the same as the keystore password.

4. Copy the generated keystore file to the `$BDNA_HOME/conf/bdnakeystore` location:

```
cp ~/.keystore $BDNA_HOME/conf/bdnakeystore
```

5. Edit `$BDNA_HOME/nih/Tomcat/jakarta-tomcat-6.0/conf/server.xml`:

5.1. Comment out the non-SSL Coyote HTTP Connector.

5.2. Remove the comments surrounding the SSL Coyote HTTP Connector.

6. Restart the Application Agent and the BDNA Shell.

7. From the BDNA Shell, issue the following commands:

```
bdna>property –m bdna.rs.SSLEnabled true
```

```
bdna>property –m bdna.rs.sslPort 8443
```

```
bdna>property -m bdna.rs.jvmdownload
"https://<servername>:8443/bdna/jre.exe"
```

8. Start BDNA:

```
bdna>startup RS
```

9. After restarting RS, the BDNA user interface will be enabled for SSL operations and available at the following URL:

```
https://hostName:8443/bdna/AppConsole.jnlp
```

10. Modify any URL redirection as necessary, to point to the new BDNA URL above.

BDNA provides initial application introductory screens, so that certain checks can be performed. For example, when accessing `http://hostname`, where the RS component is running, the client's Java version is verified before proceeding to the Java application on port 8443. These introductory pages are not encrypted and, by their nature, need to be open to various platforms before forwarding to the application itself. That means that the BDNA UI can be configured to use a single port for all of its traffic when the application's address is accessed directly. However, having the additional ports open would also allow some of the UI checks to take place.

Managing Groups 9

About this Chapter

This chapter introduces and explains the concept of *groups* as it pertains to BDNA Discover. BDNA has created the concept of *groups* to help the BDNA Administrator organize and simplify administration of scans, define roles for access control, and provide a business context when doing analytics. In a Collection Store schema, the BDNA Administrator can create groups of items using Scan Administration. In a FactBase schema, on the other hand, the administrator can create groups by using the `loadgroupset` command from the BDNA shell (refer to the *Discover FactBase User Guide* for more information). Groups provide an easy mechanism for managing logically related elements within BDNA. Many different element types are available for grouping, including:

- Individual discovered IP addresses
- Discovery zones (subnets such as a class C 192.168.10.0)
- Networks
- Software elements (such as Oracle installations)
- Hardware elements (such as routers)
- Operating system elements (such as Windows XP machines)
- Some groups can also contain other groups

BDNA groups are versatile. In Collection Stores, they can be used for scanning and analytics. A common example of groups used for scanning involves creating projects (one of the many group types), each of which contains many networks or discovery zones (subnets) and each represents a different time zone. Then, a scan task can be created to scan all of the Asia project (which comprises many networks and locations) during business hours in Asia. Similarly, a scan task can be created to scan the North America project during local business hours. This provides a convenient mechanism to organize how scans are done.

In a FactBase, groups are invaluable for analytics. Groups are created globally in the FactBase and then mapped to individual inventories. For example, a production environment group can be defined globally in the FactBase that contains networks from each inventory with machines that host applications and databases used on an assembly line.

Groups are also used for defining access control roles. For example, suppose the BDNA Administrator creates region groups Asia, North America, and Europe. The administrator also creates division groups Engineering, Finance, and Marketing. If the Analytics application contains a custom report that only Finance personnel in Europe should see, an access control role containing these two groups can be defined. For more information on groups and access control, refer to Chapter , “User Management and Access Control,” on page 131.

BDNA provides numerous preconfigured group types. Examples of group types include geographic regions, organizational divisions, and some generic types that can be configured by the administrator.

Each group type can contain different element types, both discovered and user-supplied metadata. In general, most group types can contain the following attributes:

- Discovered IP addresses (such as those found during Level 1)
- Subnets (supplied when defining networks such as a class C 192.168.10.0)
- Networks (supplied in order to define a scan task; usually contain several subnets)
- Operating systems (discovered OSs such as UNIX, Windows, routers, printers, storage, and so on)

As a best practice, BDNA recommends creating groups to encompass high-level metadata containers (as opposed to discovered data such as IP addresses and OSs) such as networks and subnets, because it is easier to manage and does not rely on the results of a particular scan. In special cases, especially when doing targeted scanning in a Collection Store, it is desirable to create groups that contain specific IP addresses or OSs.

Although BDNA has many group types—regions, projects, divisions, organization, building, environments, and so on—out of the box, they don't always align with business needs. For example, you may want to have a group (and, by extension, a column in an Analytics application report) called *sector*, but BDNA does not have this group type by default. To accommodate this, BDNA has created 20 flex-groups named *Group01 – Group20*. You can view these groups in the Scan Administration application when creating new groups and you can specify them as group types in the group bulkload file. The display label for these groups can be arbitrarily set. Whatever label is chosen will appear in the Analytics application. For information on how to set labels for Group01 – Group20, refer to the knowledge base article *How to change custom group display labels (Group01 – Group20) in Analytics reports*.

Creating, Modifying, and Removing Groups

The Scan Administration application is used to create, modify, and remove groups in a Collection Store schema. In a FactBase schema, groups are created by using the `loadgroupset` command from the BDNA shell (refer to the *Discover FactBase User Guide* for details). Groups cannot be empty, so groups can only be created after group type elements have been created in the system. In most situations, therefore, groups should only be created after networks have been created.

The following task assumes that at least one network has been created. For more on creating networks, refer to "Configuring and Running Level 1 Discovery" on page 25.

Caution: Whenever a group is created, modified, or removed, discovery data must be snapshotted and the Inventory Reports mush be refreshed.

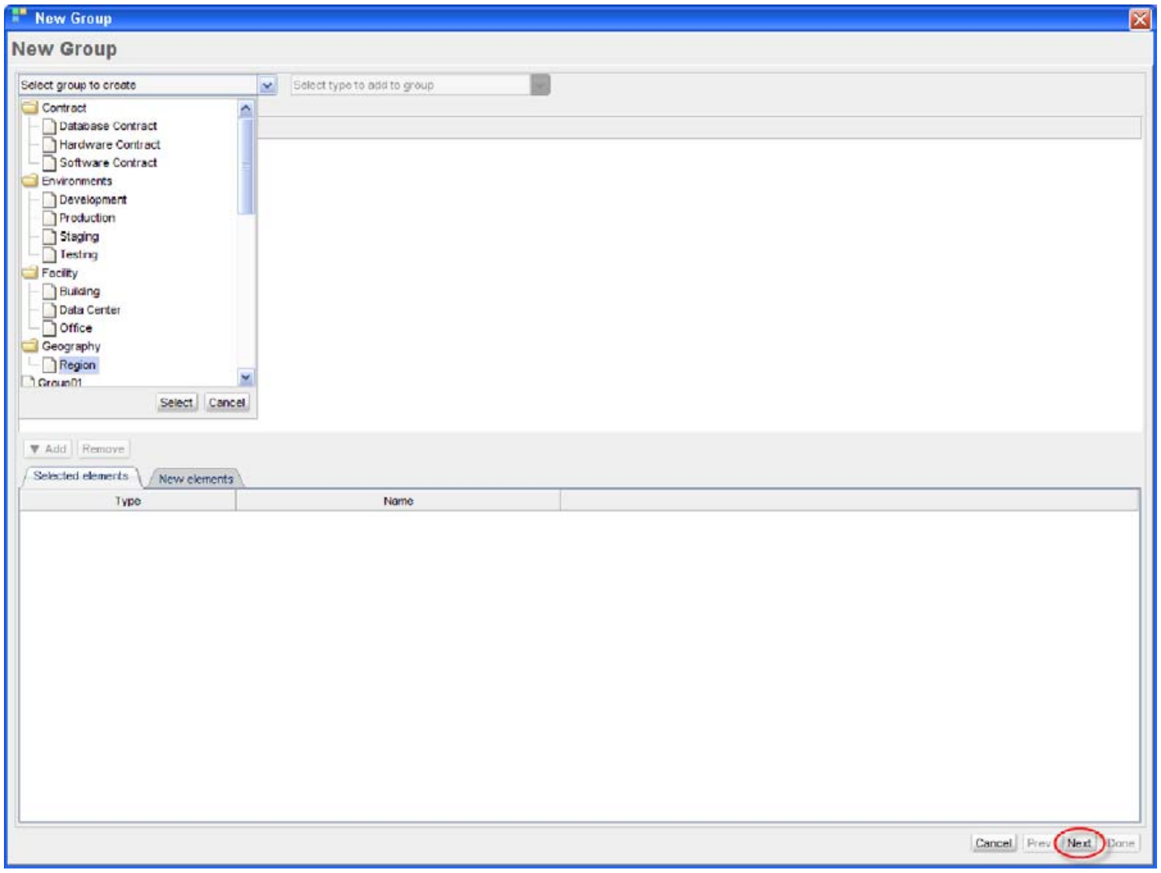
Creating a New Group

The following is a step-by-step procedure for creating a new group using Scan Administration:

1. Log in to BDNA Discover and open Scan Administration.
2. Click "Scan Setup > Groups".

   The middle pane will list previously defined groups. In the case of a clean schema, this pane will be empty.
3. In the toolbar under the application menu at the top, click "New Group". On the first page of the New Group wizard, there is a drop-down list that says: "Select group to create".
4. For this example, select "Geography > Region" from the list.

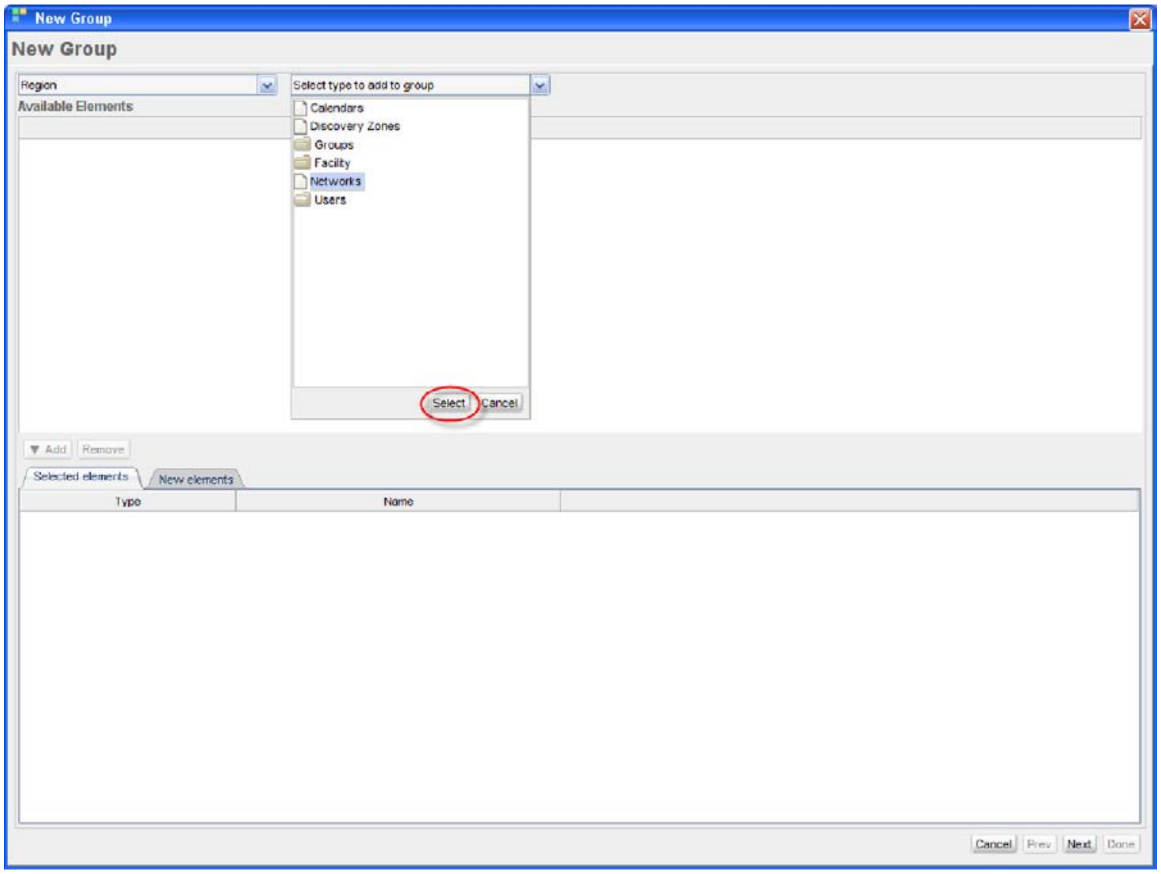
Figure 35: Create New Group: Select Group to Create



The drop-down list that says: “Select type to add to group” will now be active.

5. From this list, select “Networks”.

Figure 36: Create New Group: Select Type to Add

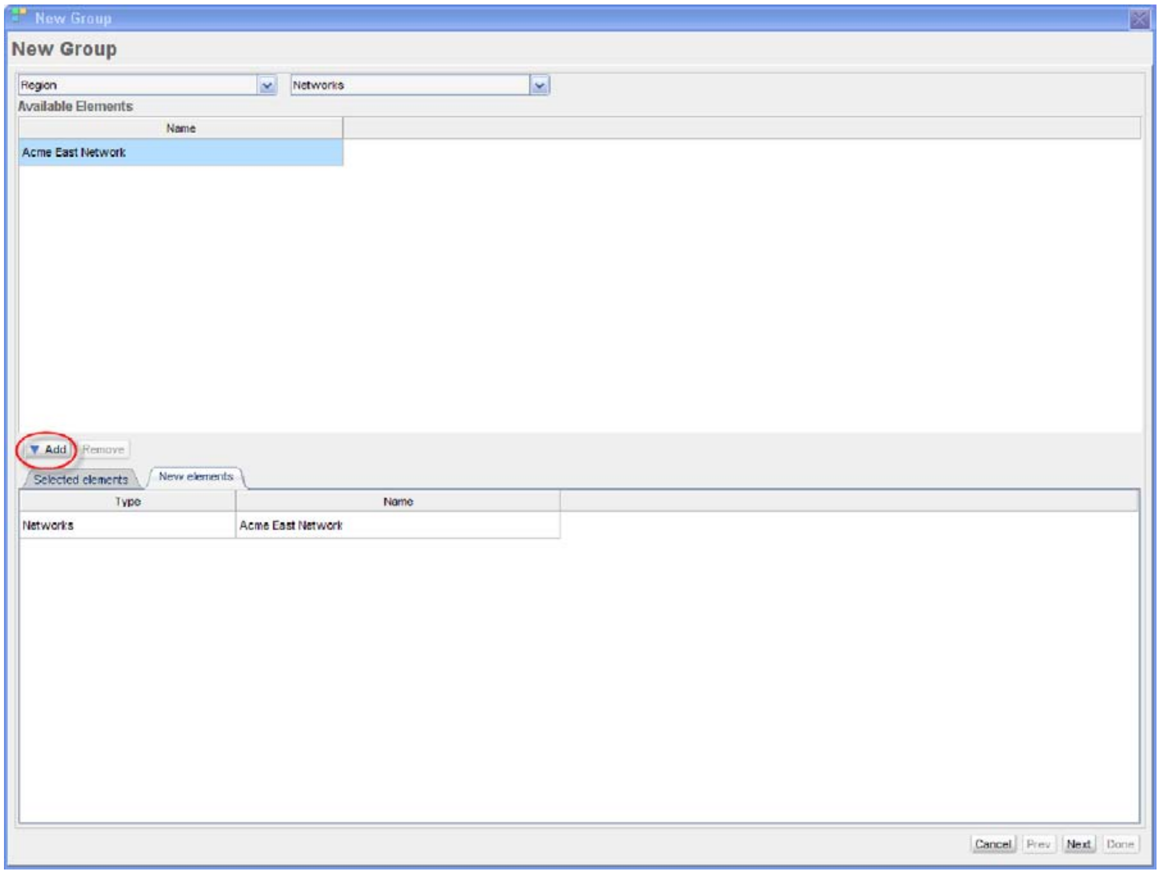


Previously created networks will now be listed under “Available Elements”.

6. If no networks appear, click “Cancel” in the bottom right corner and create some networks before proceeding.
7. Highlight the networks to be added to the group by clicking them.

   Contiguously listed networks can be selected at the same time by using the “Shift” key while clicking. Non-contiguously listed networks can be selected at the same time by using the “Ctrl” key while clicking.
8. When the correct networks have been selected, click “Add” directly below the network listing.

Figure 37: Create New Group: Add Element

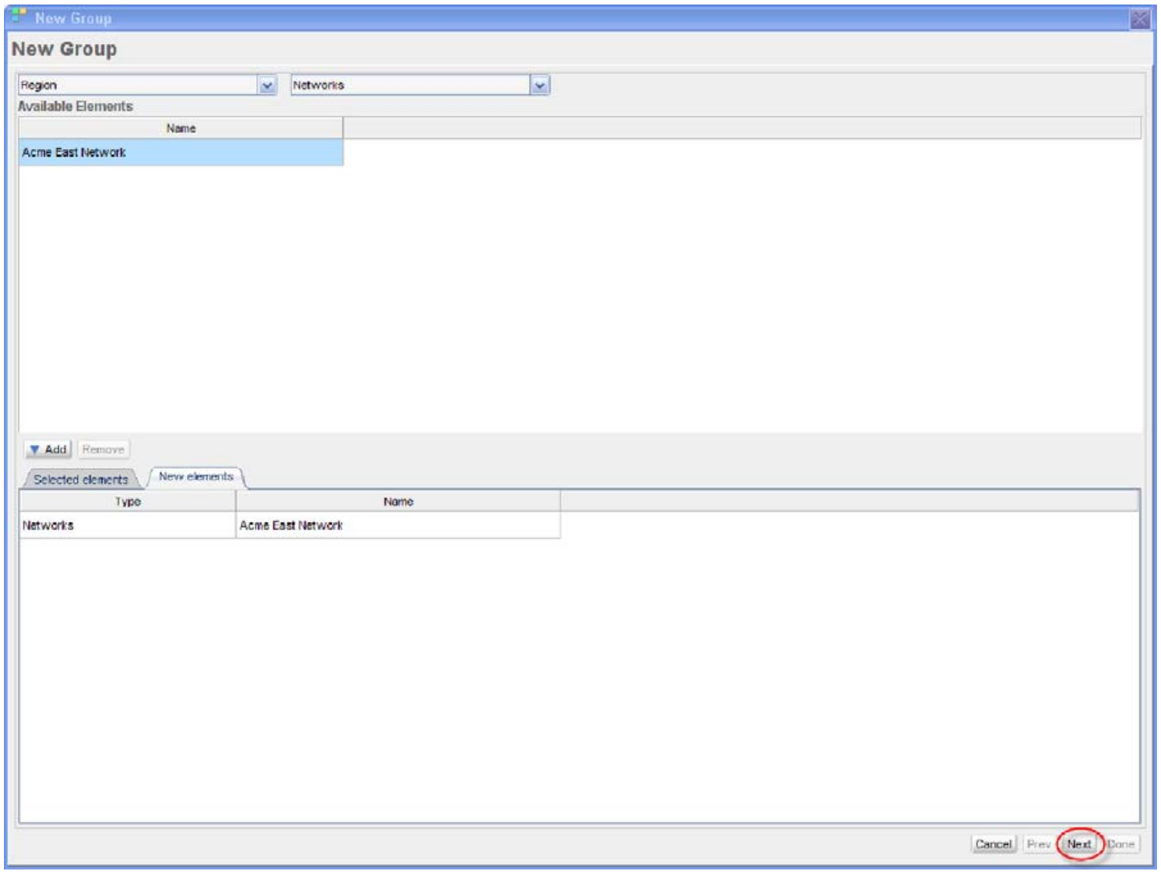


The network(s) added will now appear in the “New Elements” pane at the bottom of the wizard.

9. To remove networks, highlight them in the “New Elements” pane and click “Remove”.

10. When satisfied with the networks for this group, click “Next” at the bottom right of the wizard.

Figure 38: Create New Group: Add Element Complete

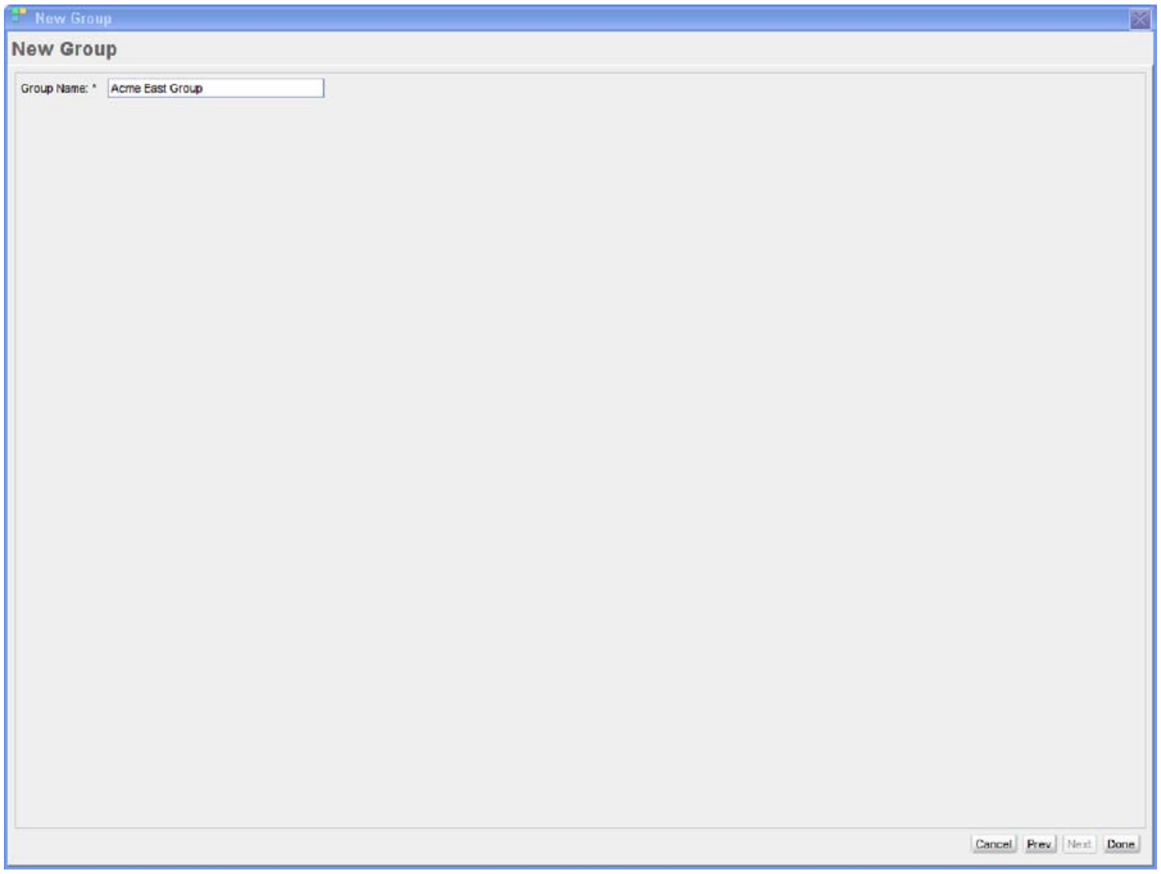


Name
Acme East Network

Type	Name
Networks	Acme East Network

11. On the last page of the New Group wizard, give the group a name.

Figure 39: Create New Group: Group Name

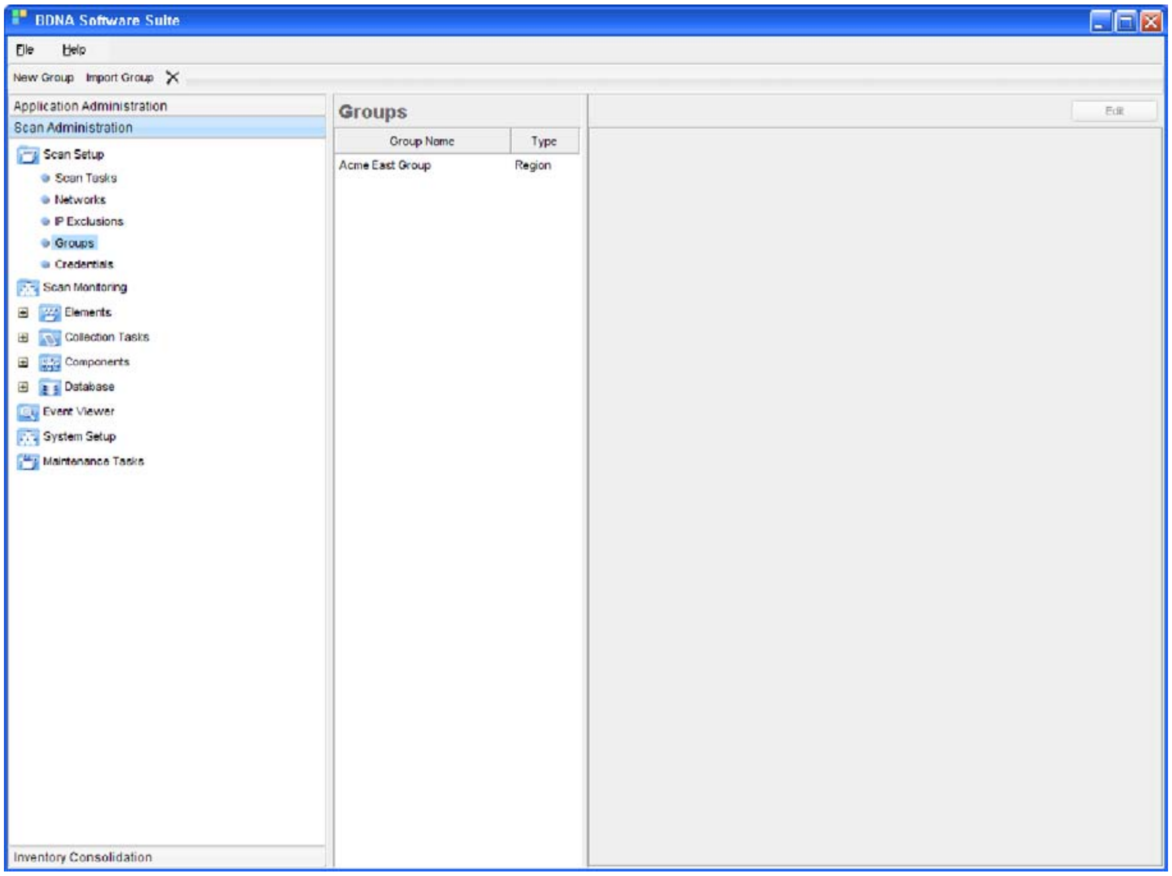


Group Name: * Acme East Group

12. Use the “Prev” button to go back to the previous page and verify entries.

13. When finished, click “Done”.

Figure 40: Create New Group: Entry



Group Name	Type
Acme East Group	Region

The new group appears in the “Groups” listing in the middle pane of the application.

Modifying an Existing Group

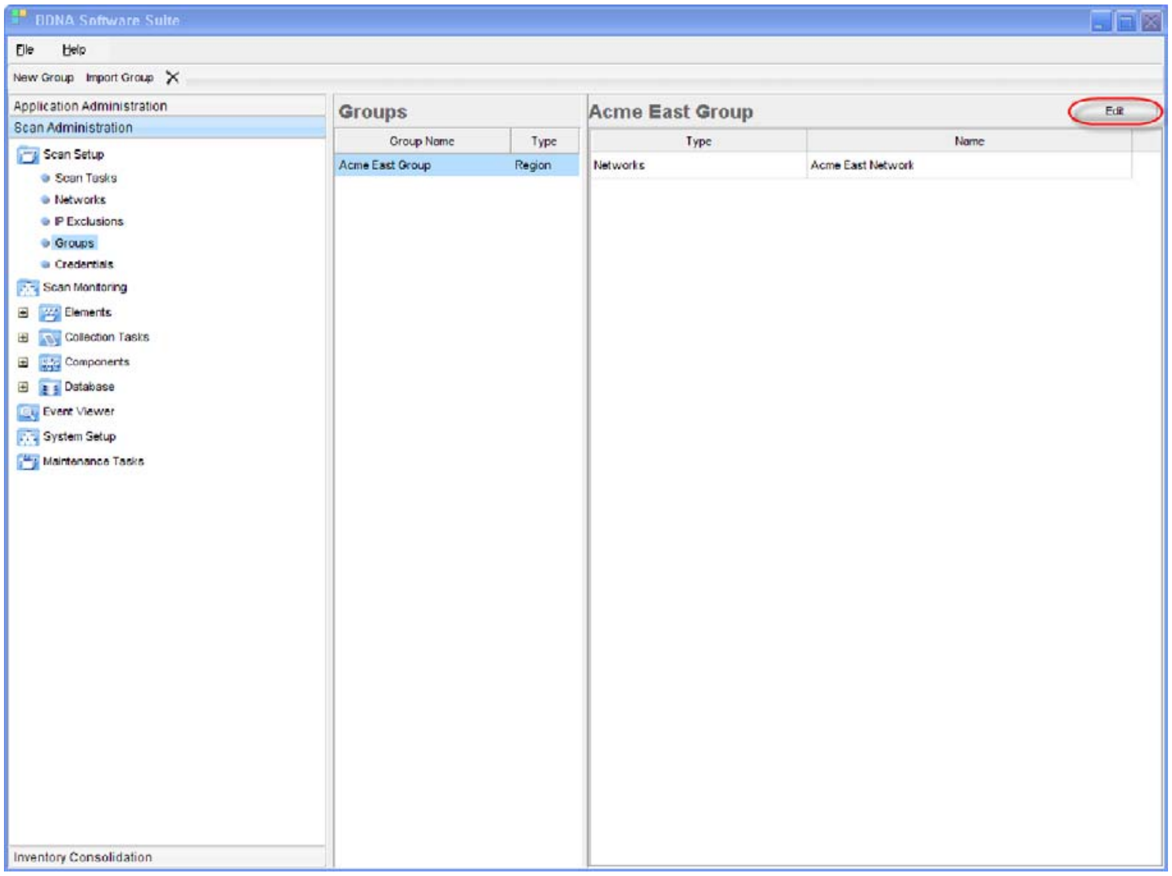
The following is a step-by-step procedure for modifying a group using Scan Administration:

1. Log in to BDNA Discover and open Scan Administration.
2. Click “Scan Setup > Groups”.

   The middle pane will list previously defined groups.
3. Locate the group to be modified in the listing and select it by clicking on it.

   Details about the group appear in the right pane of the application.
4. Click the “Edit” button at the right above the details pane.

Figure 41: Modify Group



Group Name	Type
Acme East Group	Region

Type	Name
Networks	Acme East Network

The Edit Group wizard is identical to the New Group wizard, except that previously added elements appear in the "Selected elements" tab at the bottom of the first page. This tab is empty when creating a new group.

5. Elements can be removed from this tab in a manner identical to the "New elements" tab.

   Refer to the procedure for "Creating a New Group" on page 122, starting with step 4 on page 122.

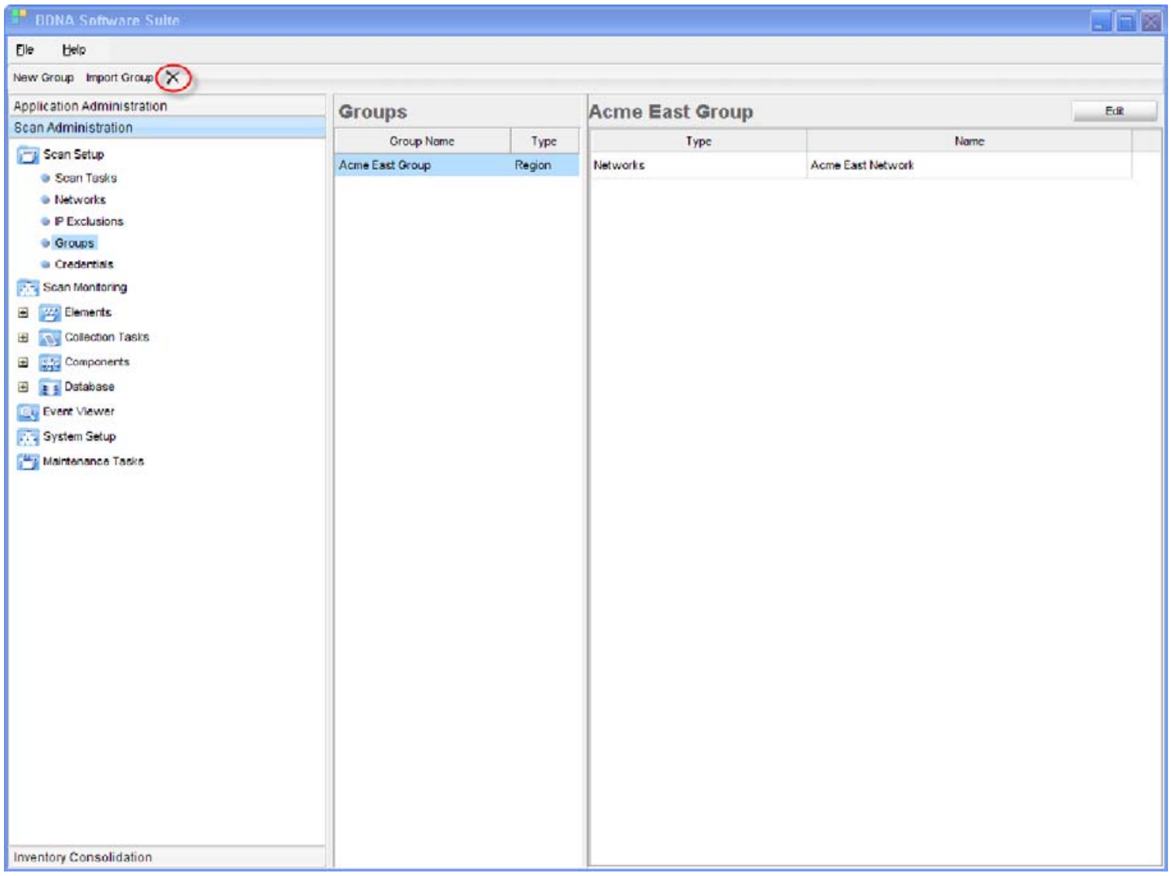
Removing a Group

The following is a step-by-step procedure for removing a group using Scan Administration:

1. Log in to BDNA Discover and open Scan Administration.
2. Click "Scan Setup > Groups".

   The middle pane will list previously defined groups.
3. Locate the group to be removed in the listing and select it by clicking on it.
4. In the toolbar under the application menu at the top, click the "X" next to "Import Group".

Figure 42: Remove Group



Group Name	Type
Acme East Group	Region

Type	Name
Networks	Acme East Network

5. Click “Yes” in the confirmation dialog box that appears.

The group will no longer appear in the listing in the middle pane.

Caution: Removing a group could invalidate associated tasks which use it for their target during a scan. Also, removing a group associated with access control could invalidate access control. Remove groups with extreme caution.

User Management and Access Control 10

About this Chapter

The BDNA Administrator can create roles to limit what data, such as rows in a report, users can see in applications such as Analytics.

This chapter describes how to manage the two basic types of users:

- BDNA Native Users

  Created and maintained using BDNA Discover, these users are internal to BDNA and are used only to access the BDNA system. When configured to use BDNA Native Users, BDNA Discover handles user authentication and role assignment.

- LDAP Users

  These users are created and managed outside of BDNA using any of the many available standard LDAP providers such as Microsoft Active Directory or Sun Java System Directory Server. When BDNA is configured to use LDAP Users the configured LDAP provider handles user authentication and group assignment.

This chapter also describes how to set up user authentication for LDAP.

Role Types

BDNA has two types of roles:

Data Roles	Data roles are defined by the administrator. In a typical access control scenario, users will have both object and data roles that control what the user can do and what data they can see. Data roles determine which rows, for example, in an Analytics report the user can see. This is sometimes known as row level access control. Data roles are defined by associating BDNA groups with each role.
Object Roles	Object roles cannot be configured by the administrator. Object roles determine what capabilities a user has or what objects a user can see. For example, BDNA Admin is an object role that allows users (who are assigned to that role) to create scan tasks whereas the BDNA Analyst object role can not create scan tasks. Object roles are also used to determine which reports a user can or cannot see. For example, a user can create a custom report in the Analytics application and save it under My Reports. The user can then share the entire report object with another user. Even if the user has object privileges on the report, whether or not the other user can see the actual rows in that report will be determined by that user's data roles.

Managing Native Users

BDNA comes with two predefined users: *root* and *analyst*. (The BDNA root user is separate and distinct from the Linux root user, although they are similar in that the BDNA root user is the super-user for BDNA.) The root user has the power to do anything in the system, including creating and initiating scan tasks, as well as viewing all reports and data in the Analytics application. The analyst user, however, is limited in what it can do because of its object role.

BDNA has two predefined object roles: *BDNA Admin* and *BDNA Analyst* (not to be confused with the predefined root and analyst users). In order to log in to BDNA Discover and use the applications, users must be assigned at least one of these object roles.

Users in the BDNA Admin role are able to do anything in the system. The root user is a default user that has been assigned the BDNA Admin role.

Users in the BDNA Analyst role can log in to BDNA and view the various reports. The data (that is, rows) that is available in these reports depends on whether row-level access control has been enabled (see below). The *analyst* user is the default user that has been granted the BDNA Analyst role.

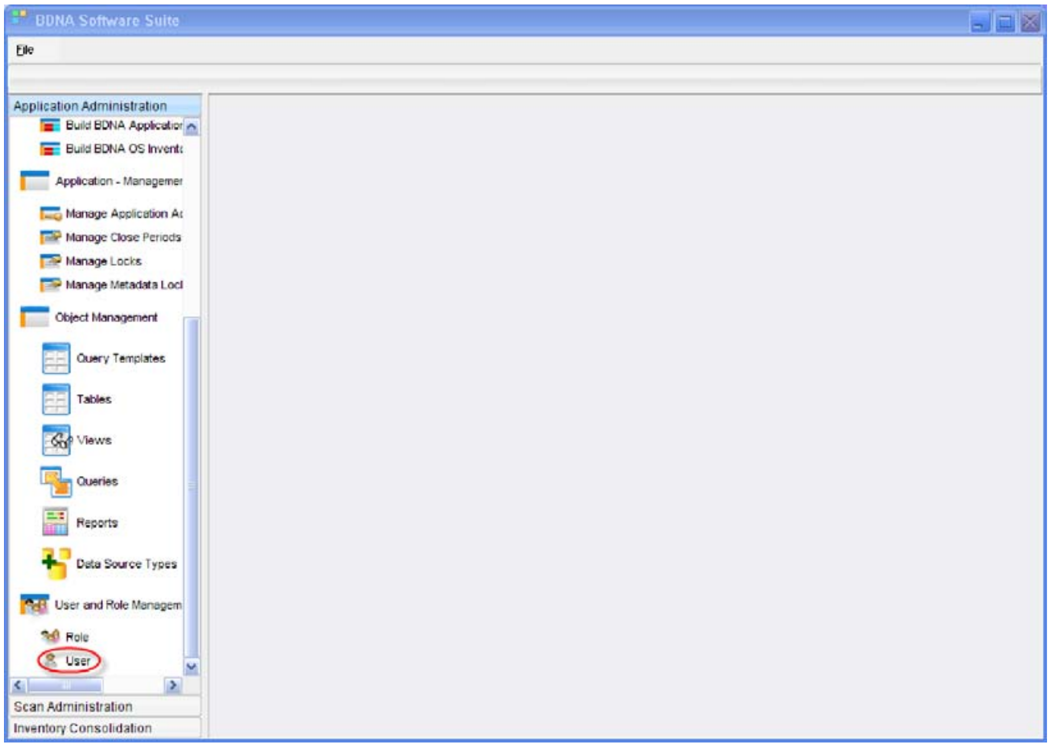
Creating a Native BDNA Analyst User

To create a Native BDNA Analyst user:

1. Log in to BDNA Discover with a BDNA admin username and password and open Application Administration.
2. In the left navigation pane, click “Application Administration > User and Role Management > User”.

   Action buttons display under the application menu. A listing of users already defined in the system appears in a pane to the right of the navigation pane.

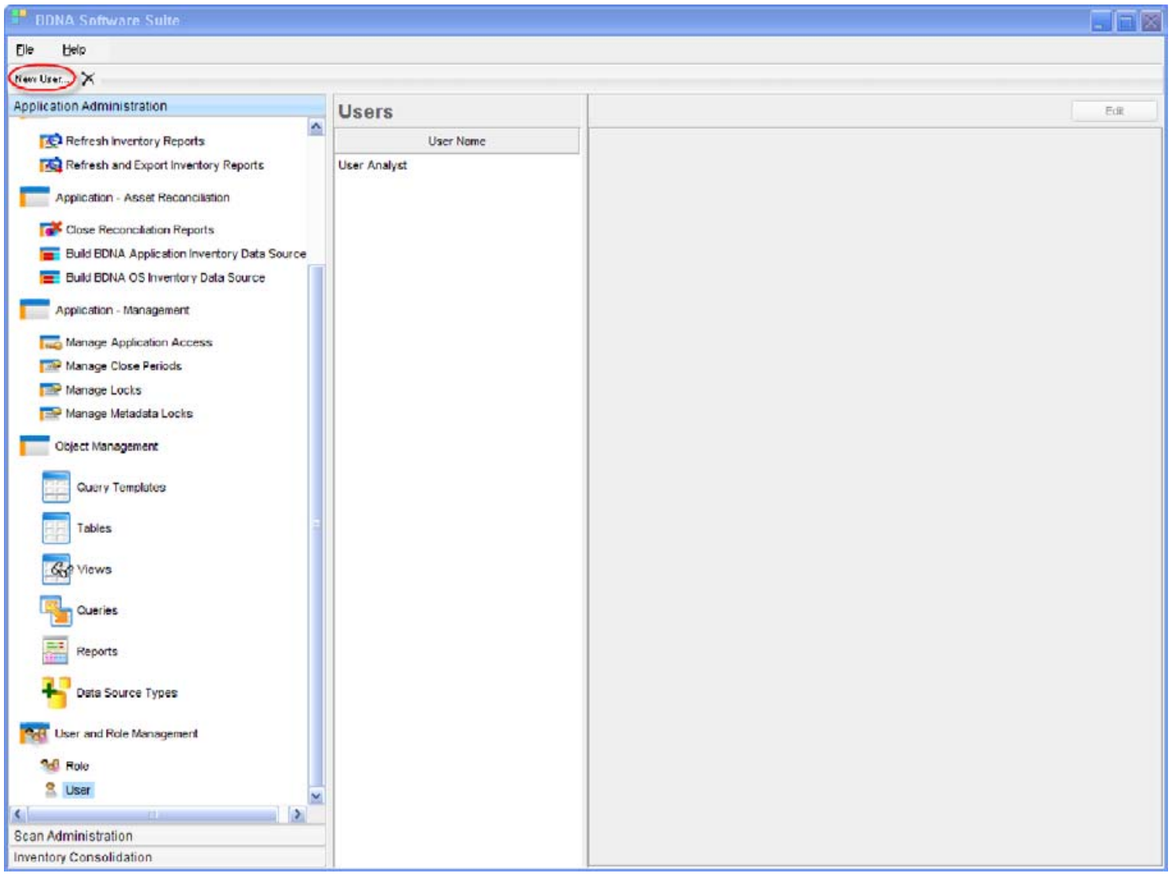
Figure 43: Application Administration: User and Role Management



3. At the top of the main content pane, click “New User”.

   The New User Wizard appears.

Figure 44: Application Administration: Create New User

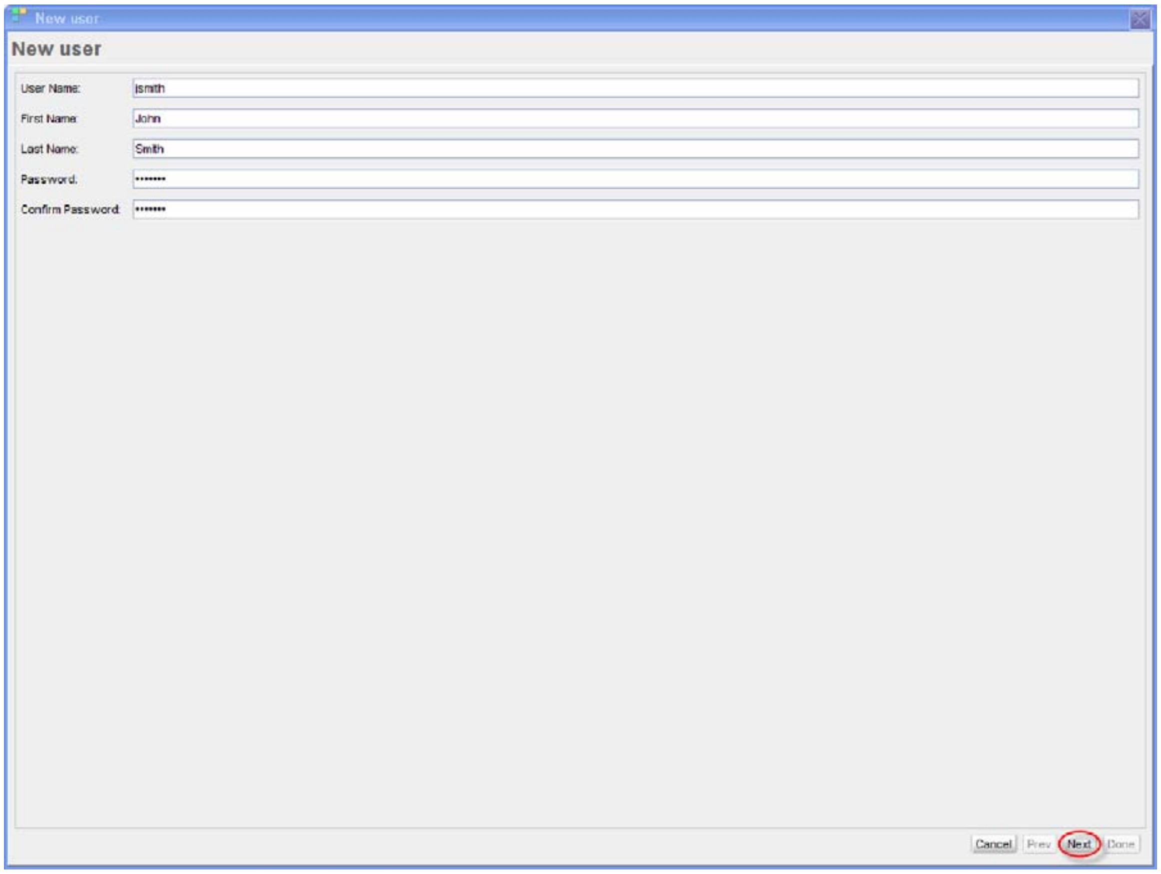


4. In the New User wizard, fill in the fields as appropriate.

“User Name”, “Password”, and “Confirm Password” are the only mandatory fields.

5. Click “Next”.

Figure 45: Application Administration: Create New User—Details



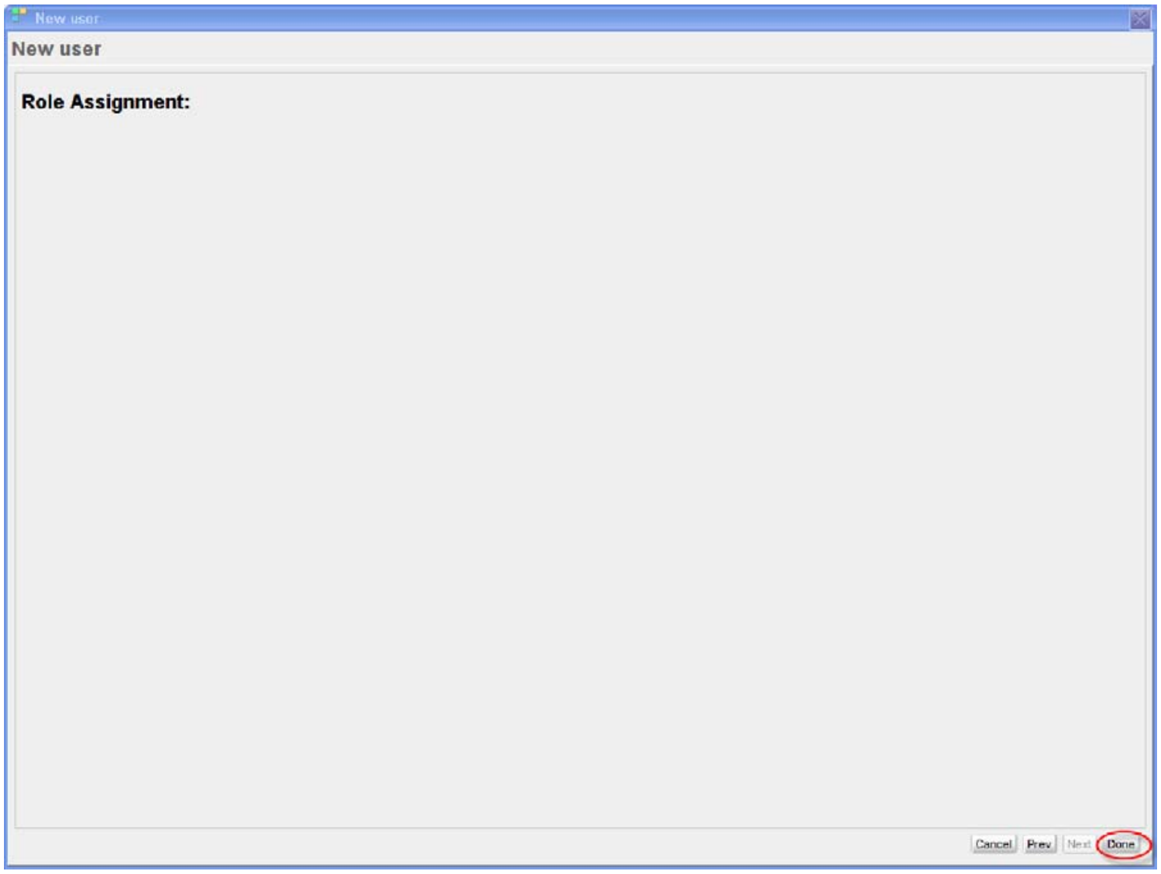
New user

User Name:	jsmith
First Name:	John
Last Name:	Smith
Password:	•••••••
Confirm Password:	•••••••

On the second page of the New User wizard there is a list of available data roles. Object roles are not shown here because they cannot be explicitly assigned to a user. By default, the user is granted the object role BDNA Analyst. If Access Control has not been enabled, no roles appear here.

6. Click “Done”.

Figure 46: Application Administration: Create New User—Role Assignment



7. Test the user by logging in to BDNA Discover.

Modifying a Native BDNA Analyst User

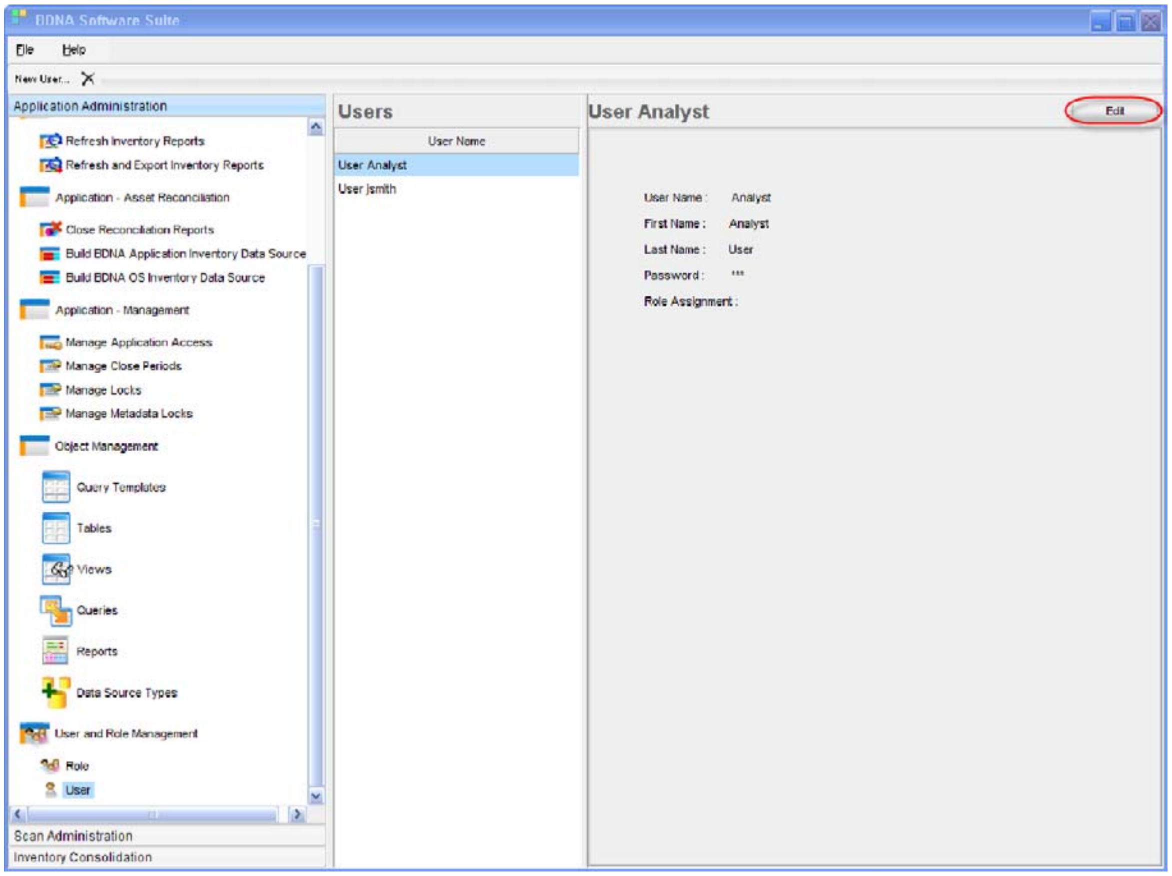
The following procedure can be used to modify a Native BDNA Analyst user, a user password, or to add roles (see Access Control, below).

1. Log in to BDNA Discover with the BDNA username and password for an Administrator user.
2. In the left navigation pane, click “Application Administration > User and Role Management > User”.
3. In the list of users in the middle, find the user to be modified and highlight it by clicking on it.

   Details about the user appear in the pane to the right.
4. Click the button at the top right labeled “Edit”.

   The Edit User Wizard appears.

Figure 47: Application Administration: Modify User



User Name :	Analyst
First Name :	Analyst
Last Name :	User
Password :	***
Role Assignment :	

The Edit User Wizard works exactly like the New User Wizard described above, except that “User Name” cannot be edited. Refer to step 4 on page 133 through step 7 on page 135 for more information on how to use the Edit User Wizard.

About Access Control

BDNA provides support for row-level access control of data shown through the Analytics application. This level of access control is configured explicitly. By default, row-level access control is disabled in BDNA. In this configuration, the default user Analyst can see all data in all Analytics reports. Once an access control role is created for the first time, the application prompts the user to enable row-level access control for the entire system (refer to “Enabling Access Control” on page 140). When access control is enabled, the default user Analyst sees no data in each report. This is intentional and is due to the fact that the default user Analyst has no data roles (for row-level access control) assigned to it. Thus, once BDNA roles are created, it is expected that users (either native or LDAP) will be created and assigned these new data roles. These users will then be able to see data appropriate to their role assignment.

Roles and Access Control

Each role grants access to subsets of the total available data (or possibly all available data). In BDNA, the total available data is defined by the total number of active operating systems. Therefore, each role will grant rights to a subset of the total available OSs. BDNA groups are typically defined in terms of networks, discovery zones (subnets), IP addresses, or operating systems. Regardless of how they are defined, each BDNA group will ultimately contain different operating systems. These groups form the basis of role-based grants for access control.

Using Groups for Access Control

Data roles are defined in terms of BDNA groups. "Managing Groups" on page 121 discusses the use of BDNA groups to logically manage groupings of active IPs, networks, subnets, and the like. Consider the example where two groups are defined: MV and DC, each containing dozens of networks. With row-level access control enabled (refer to "Enabling Access Control" on page 140), a user belonging only to the "BDNA Analyst" object role would not be able to see any data in the Analytics reports (that is, the reports would be empty), because they are not assigned any data roles.

But suppose the BDNA Administrator has created a data role, called "MV Analyst". This role grants access to data in the MV group. The admin then adds a user to this role. When this user logs into the Analytics application, the reports will contain only data pertaining to the MV group. Likewise, a second data role could be created for a "DC Analyst", granting access to data discovered on only the DC group.

Now, suppose there is a third group, MV2, and data from this group should be available only to an "MV Analyst" user. The BDNA Administrator can simply extend (edit) the "MV Analyst" role by including the MV2 group in the role's grant definition. BDNA takes the union of the two groups when determining what assets an "MV Analyst" user has rights to view.

Finally, suppose that there should be another, higher level of user. This user should have access to all data. To accomplish this, the BDNA admin would simply create a new role that includes all groups—in this example, MV, MV2, and DC—in its grant definition.

Mixing Group Types for Access Control

Thus far, only the simple case of roles containing unions of same-type groups has been covered. It is often the case that a single role includes different group types in its grant definition. For example, assume again that an "MV Division" and a "DC Division" exist, with "MC Analyst" and "DC Analyst" users. This time, however, there exists another group, for example of type "Workgroup". The name of this new group is called "Windows Workgroup", and it contains all of the Windows machines in all of the networks.

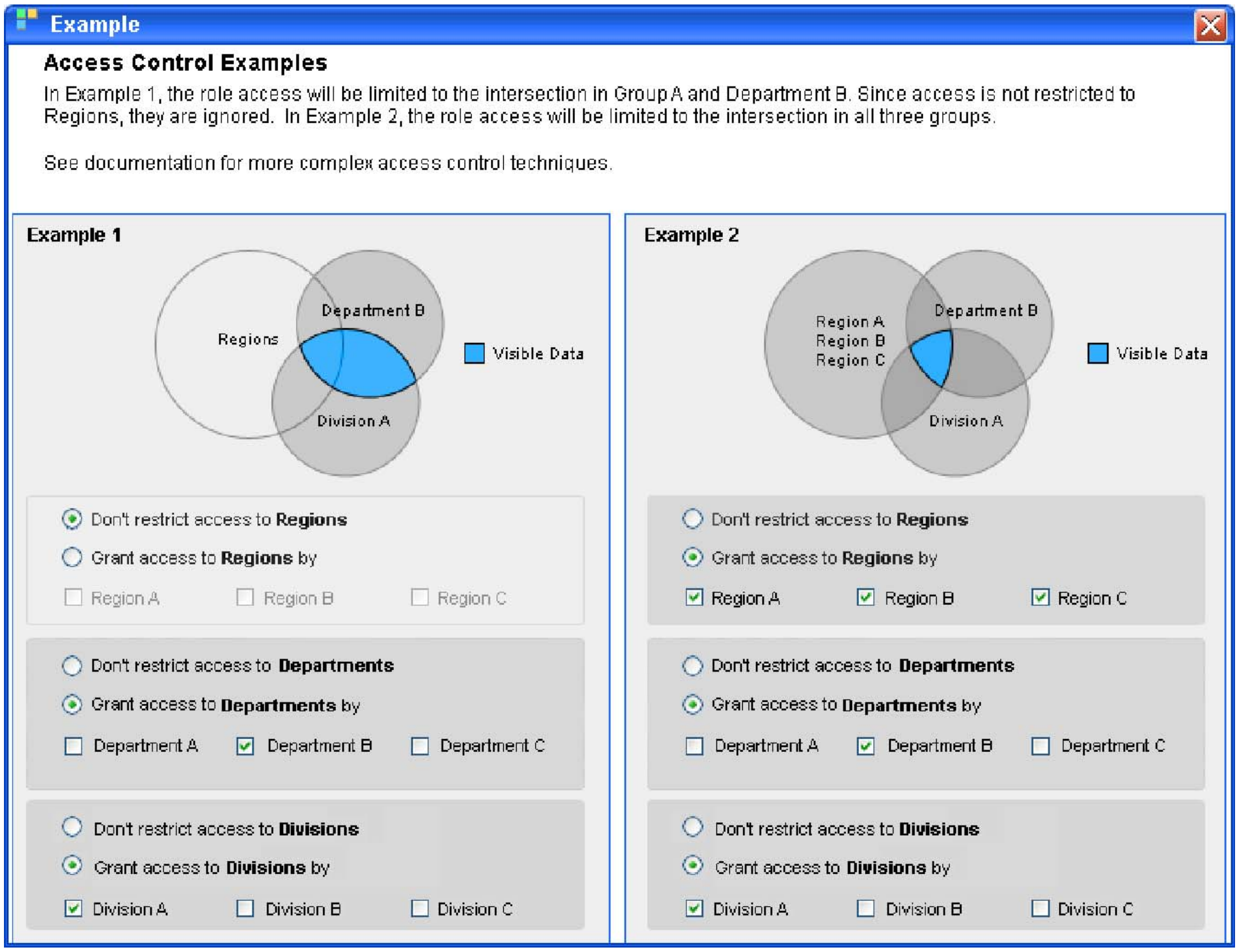
Suppose the BDNA Administrator creates a new data role. The grant definition for this new role contains both the "MV Division" and "Windows Workgroup" groups. In this case, instead of the union of these two groups, BDNA uses the intersection of these groups for granting access to this new role. The BDNA Administrator could call this new role "MV Windows Analysts", and users in this role would only have access to discovery data pertaining to Windows machines in the "MV Division".

It is very important to understand that when mixing group types within the same role, the result is an intersection of the described assets, while the union is taken when group types are the same. This allows for the creation of complex data access control definitions.

Access Control Example Illustration

The following illustration, which is also available inside the Application Administration application, graphically shows the results of unions and intersections for access control:

Figure 48: Access Control Examples



Access Control Examples

In Example 1, the role access will be limited to the intersection in Group A and Department B. Since access is not restricted to Regions, they are ignored. In Example 2, the role access will be limited to the intersection in all three groups.

See documentation for more complex access control techniques.

Example 1

[✓ Don't restrict access to **Regions**]
Grant access to **Regions** by
☐ Region A ☐ Region B ☐ Region C

Don't restrict access to **Departments**
[✓ Grant access to **Departments** by]
☐ Department A [✓ Department B] ☐ Department C

Don't restrict access to **Divisions**
[✓ Grant access to **Divisions** by]
[✓ Division A] ☐ Division B ☐ Division C

Example 2

Don't restrict access to **Regions**
[✓ Grant access to **Regions** by]
[✓ Region A] [✓ Region B] [✓ Region C]

Don't restrict access to **Departments**
[✓ Grant access to **Departments** by]
☐ Department A [✓ Department B] ☐ Department C

Don't restrict access to **Divisions**
[✓ Grant access to **Divisions** by]
[✓ Division A] ☐ Division B ☐ Division C

Users and Multiple Roles

Users can also have multiple roles. When a user has multiple roles, the user is granted the union of the rights granted by each of the roles. Hence, in the above examples, a “super user” could be created by granting a user both the “MV Analyst” and “DC Analyst” roles.

Only Groups with Active IPs are Available for Grants

For a group to be available for access control, the group must contain active IPs. This requires that at minimum, Level 1 discovery of those IPs has been performed, and a snapshot has been made by running “Data Preparation > Snapshot Discovery Data”. As a practical matter, access control should be applied after all scanning is completed. It cannot be applied to a blank schema.

Workflow for Configuring Access Control

To summarize, the following steps must be completed in order to configure access control on a set of scan data:

1. Create networks
2. Create groups prior to scanning (optional)
3. Scan
4. Create groups after scanning (optional)
5. Snapshot discovery data
6. Configure access control
7. Refresh Analytics reports

Creating Roles

The following is a step-by-step procedure for creating roles in BDNA Application Administration:

1. Log in to BDNA Discover with a BDNA administrator username and password.
2. In the left navigation pane, click “Application Administration > User and Role Management > Role”.
3. Click “New Role…” under the application menu in the upper left corner.

   If “Snapshot Discovery Data” has never been performed on this schema there will be an error to indicate this.
4. If the error displays, run “Snapshot Discovery Data” before proceeding.

   If a scan has never been performed with this schema, there will be an error indicating that no networks or groups exist. Refer to “Only Groups with Active IPs are Available for Grants” on page 138 for more information.
5. On the first page of the New Role wizard, assign a name for the role.

   To simplify integration with LDAP, role names should only include alphanumeric characters and spaces, and spaces should not be at the end of the role name. When done, click “Next”.
6. On the second page of the New Role wizard, configure the networks and groups that will be visible to this role.

   Refer to “Using Groups for Access Control” on page 137 for more information on configuring visibility based on networks and groups.
7. When done, click “Next”.

   (Click See Example to view the illustration in Figure 48 on page 138.)
8. On the third page of the New Role wizard, choose “BDNA User Assignment” to assign the role to native BDNA users.

   or

   Choose “LDAP User Assignment” if only LDAP users will be assigned this role.

   or

   If both native BDNA users and LDAP users will be assigned this role, choose “BDNA User Assignment” and assign this role to native BDNA users; this will not preclude assigning this role to the LDAP users afterwards.

   If access control has previously been enabled, the “Done” button will be active in the bottom right corner of the wizard.

9. If the “Done” button is active, when satisfied, click “Done” and skip step 10 and step 11.

   If access control has not previously been enabled, the “Done” button will still be grayed out.

10. Click “Next” when ready to continue.

    The final page of the wizard asks whether access control should be turned on now.

11. Leave the default “Yes” checked and click “Done”.

    The newly created role will now appear in the “Roles” listing in the middle of the application window.

Enabling Access Control

There are two ways of enabling row-level access control within BDNA. As in the previous example, “Creating Roles” on page 139, the first method is to create a new role. The last page of the Create Role wizard will ask if access control is to be enabled. Selecting “Yes” will enable access control.

The second method of enabling access control is to manually set the BDNA property `bdna.pl.app.enableACL` to “true”; in Collection Store, use “Scan Administration > System Setup > Properties” to do so. In FactBase, the property can be set using the BDNA shell command “`property -m bdna.pl.app.enableACL true`. The two possible values for this property are “true” and “false”, with “false” being the default.

Once access control has been enabled by either method, the value of `bdna.pl.app.enableACL` will be “true”.

Disabling Access Control

To disable row-level access control, manually set the `bdna.pl.app.enableACL` property to “false”; in Collection Store, use “Scan Administration > System Setup > Properties” to do so. In FactBase, the property can be set using the BDNA shell command `property -m bdna.pl.app.enableACL false`.

Modifying Roles

1. Log in to BDNA Discover with a BDNA admin username and password.
2. In the left navigation pane, click “Application Administration > User and Role Management > Role”.
3. In the list of roles in the middle pane, locate the role to be modified and highlight it by clicking on it.

   Details for the role appear in the right pane.
4. Click the “Edit” button above the right details pane to open the Edit Role wizard.

   The Edit Role wizard is identical to the New Role wizard except that “Role Name” on the first page of the wizard cannot be edited. For more information on how to use this wizard, refer to step 6 through step 11 in “Creating Roles” on page 139.

Exporting and Importing Users and Roles

Once users and roles are created in a FactBase, they persist. However, if FactBase is not being used and users are logging in to a different Collection Store on a regular basis, you may want to export and import access control definitions across schemas. To accomplish this, BDNA has provided an export and import capability in Application Administration.

To export and import users and roles:

1. After defining users and roles, click on either Role or User under User and Role Management.
2. Click the Export button on the toolbar.
3. Choose a location and file name and save the file.

   This exports roles and users, regardless if you clicked Role or User.
4. Once a new Collection Store has been initialized and the groups have been created in it, click the Import button, select the file, and the users and roles will be imported.

Using BDNA with LDAP

A directory is like a telephone directory—it is used to look up users and groups. Microsoft’s Active Directory is an implementation of *Lightweight Directory Access Protocol* (LDAP). BDNA can use LDAP for user authentication. The BDNA Administrator should choose whether to use LDAP for user authentication, or to use native BDNA users, based on their own situation.

Even in situations where no pre-existing LDAP user schema exists, it still may be beneficial to configure a standalone LDAP server just to administer and maintain BDNA users. Native BDNA users are not persistent across Collection Store schemas; using LDAP will make it unnecessary to recreate those users every time a new schema is initialized. Configuring a stand-alone LDAP server solely for use with BDNA is beyond the scope of this document. The following assumes that an LDAP server exists to which BDNA can connect for the purposes of user authentication.

LDAP Users

Any user that BDNA uses for authentication must reside in the LDAP schema. For Microsoft's Active Directory, any user that BDNA uses for authentication must be an Active Directory User.

Distinguished Name

The Distinguished Name (DN) of an object in LDAP is the name that uniquely identifies an entry in the LDAP directory. Any object in LDAP is usually referenced by its DN, typically stored as an attribute of the object in LDAP. For Microsoft's Active Directory, the DN attribute of an LDAP object is called *distinguishedName*

Make note of the location of the DN in the LDAP schema where the BDNA LDAP users are stored:

An example of a distinguished name using the format above is “CN=Users,DC=bdnacorp,DC=com”. The distinguished name is case sensitive.

Working with LDAP Groups

Every BDNA role to be employed by LDAP users must map to a corresponding LDAP group; you must have one group for each role. The LDAP group should have exactly the same name as the BDNA role that it maps to. Once the LDAP group has been mapped to the BDNA role and LDAP users have been added, the LDAP group will be granted the mapped BDNA role. The LDAP Administrator must then add all users who require access to BDNA scan data to the appropriate BDNA LDAP group.

BDNA has two predefined object roles: *BDNA Admin* and *BDNA Analyst*. If you want to use these roles, create two LDAP groups called *BDNA Admin* and *BDNA Analyst*. Make sure that your users are members of one of these two LDAP groups.

If you want to use roles other than the two predefined object roles (*BDNA Admin* and *BDNA Analyst*), you will need to create roles in BDNA that map to the LDAP groups you created.

Note: If you create two new roles in BDNA, *the roles must have exactly the same name as the LDAP groups you created*. Bear in mind that name values are *case sensitive*.

The mapping from BDNA to LDAP is done through the role name. Make certain that one of the roles is a BDNA Administrator, as is the case with the predefined BDNA Admin role.

Points to consider about LDAP groups:

- Distinguished name

  Make note of the DN of the location where these groups are stored. Note also that it is possible to create it in the same location as the users. The distinguished name is case sensitive.

- Attribute Name

  The LDAP Attribute Name of the group holds references to the members of the group. The Attribute Name might be something like “member” or “uniqueMember”.

Overview of BDNA Authentication Using LDAP

The BDNA Discover Client implements LDAP authentication using a double-bind authentication scheme. This scheme is typically used with LDAP configurations that do not permit users to bind to LDAP with just a username and password. In a double-bind LDAP authentication scheme, the client must first find the DN of the user in LDAP and bind to LDAP with that user DN and password.

Initially, the client binds to LDAP using a special rootuser/rootpassword. This rootuser has permission to look up information about other users in the LDAP directory. When a user logs into Discover Client, the Discover Client will first bind to LDAP as rootuser/rootpassword. The Discover Client will then look up the login user in the LDAP directory (while it is logged in to the LDAP directory as rootuser), and retrieve the DN of the login user. The Discover Client then does a second bind to LDAP with this login user DN and password. If the bind succeeds, the Discover Client has successfully authenticated the login user credentials.

Connecting BDNA to LDAP

It is necessary to configure BDNA so that it uses the LDAP server for authentication. To do so, properties relevant to LDAP must first be set. The RS component must then be restarted.

The following information is needed to set up BDNA for LDAP:

- LDAP Server Hostname and Port

  Make note of the server hostname and port where the LDAP server will be listening. The standard port for LDAP is 389.

- LDAP Root Distinguished Name

  A “distinguished name” is the unique means of identifying an individual record in an LDAP database. Make note of the LDAP root distinguished name for the LDAP schema which contains BDNA users. One example might be “DC=yourcompany,DC=com”. The LDAP administrator can supply this information. Also, make note of the root password.

Properties can be set via two methods—either through BDNA Discover Client or through the BDNA Shell.

Setting BDNA Properties for LDAP through BDNA Discover

The following is a step-by-step procedure for setting properties using the BDNA Discover Client:

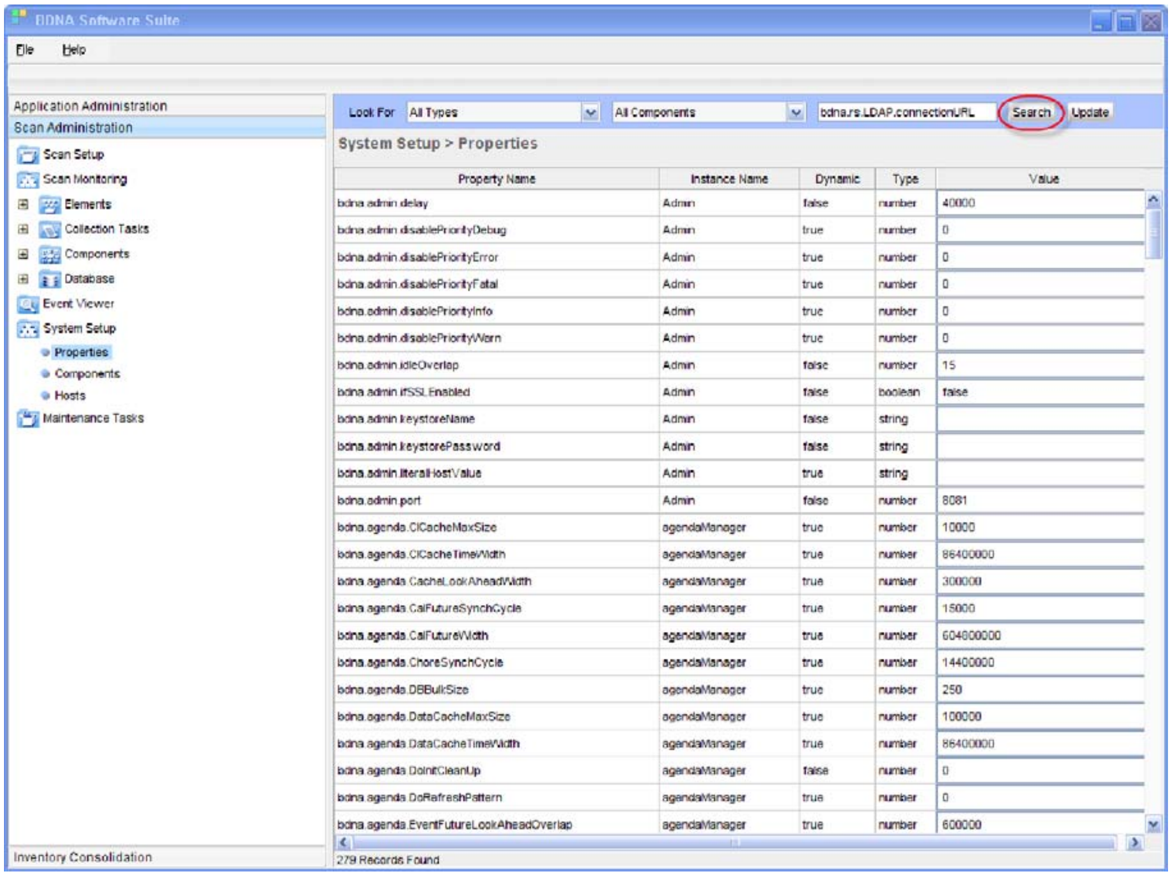
1. Log in to the BDNA Discover Client with a BDNA Admin username and password.
2. In the left navigation pane, click “Scan Administration > System Setup > Properties”.
3. In the text box to the left of “Search” (upper right corner of Application Suite window), enter the property to be edited.

   For example:

   `bdna.rs.LDAP.connectionURL`
4. Click “Search”.

   The property to be edited appears on the panel below, with an entry for Value. If there is a current value for this property, that value appears. If there is no current value for this property, the box is blank.

Figure 49: Setting BDNA Properties for LDAP: Search



All Types | All Components | bdna.rs.LDAP.connectionURL

System Setup > Properties

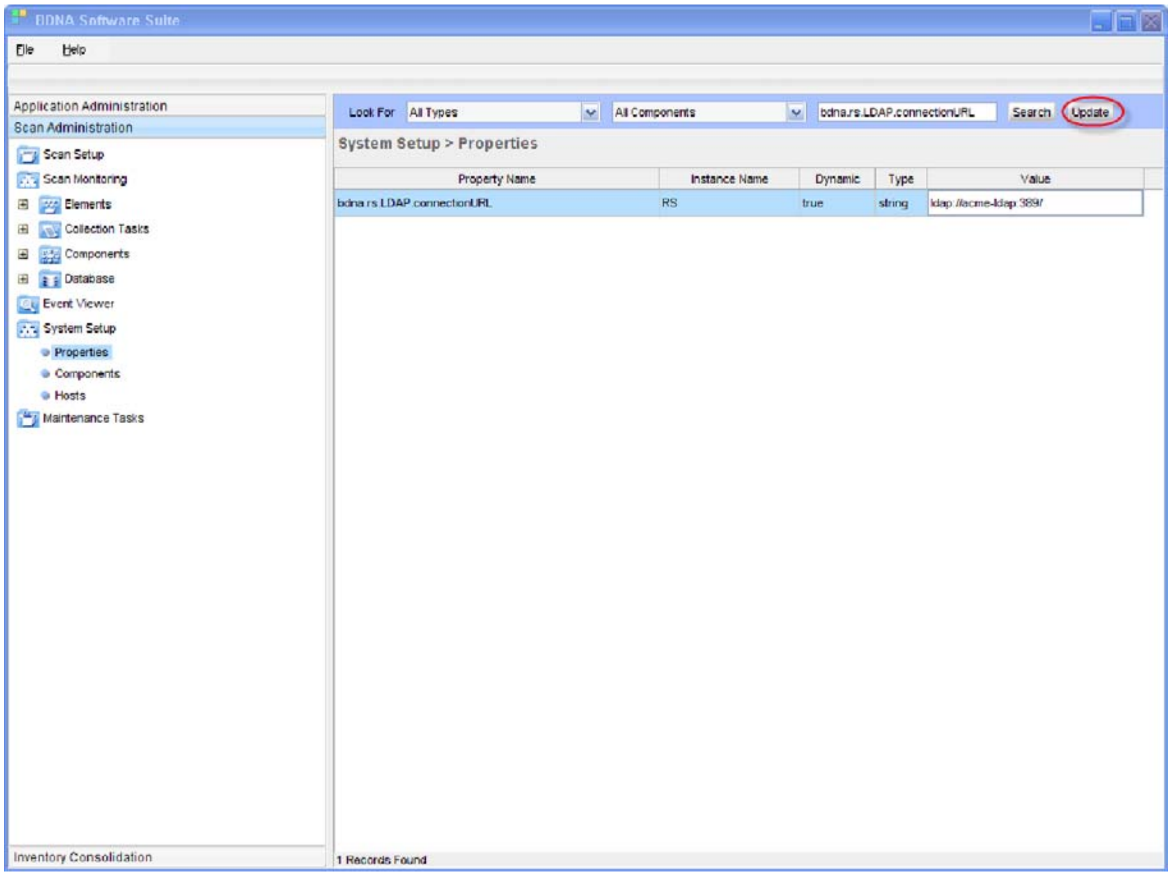
Property Name	Instance Name	Dynamic	Type	Value
bdna.admin.delay	Admin	false	number	40000
bdna.admin.disablePriorityDebug	Admin	true	number	0
bdna.admin.disablePriorityError	Admin	true	number	0
bdna.admin.disablePriorityFatal	Admin	true	number	0
bdna.admin.disablePriorityInfo	Admin	true	number	0
bdna.admin.disablePriorityWarn	Admin	true	number	0
bdna.admin.idleOverlap	Admin	false	number	15
bdna.admin.ifSSLEnabled	Admin	false	boolean	false
bdna.admin.keystoreName	Admin	false	string	
bdna.admin.keystorePassword	Admin	false	string	
bdna.admin.literalHostValue	Admin	true	string	
bdna.admin.port	Admin	false	number	8081
bdna.agenda.CICacheMaxSize	agendaManager	true	number	10000
bdna.agenda.CICacheTimeWidth	agendaManager	true	number	86400000
bdna.agenda.CacheLookAheadWidth	agendaManager	true	number	300000
bdna.agenda.CalFutureSynchCycle	agendaManager	true	number	15000
bdna.agenda.CalFutureWidth	agendaManager	true	number	604800000
bdna.agenda.ChoreSynchCycle	agendaManager	true	number	14400000
bdna.agenda.DBBulkSize	agendaManager	true	number	250
bdna.agenda.DataCacheMaxSize	agendaManager	true	number	100000
bdna.agenda.DataCacheTimeWidth	agendaManager	true	number	86400000
bdna.agenda.DoInitCleanUp	agendaManager	false	number	0
bdna.agenda.DoRefreshPattern	agendaManager	true	number	0
bdna.agenda.EventFutureLookAheadOverlap	agendaManager	true	number	600000

279 Records Found

5. Enter the new value in this text box, and click “Update” (upper right corner of application window).

   A confirmation dialog box appears with the proposed update(s).

Figure 50: Setting BDNA Properties for LDAP Update



Property Name	Instance Name	Dynamic	Type	Value
bdna.rs.LDAP.connectionURL	RS	true	string	ldap://acme-ldap:389/

6. Click “OK” to update the property.

Setting BDNA Properties for LDAP through the Command Line

In order to ensure proper authentication of the BDNA analyst and BDNA admin, LDAP must be set up in two places: the BDNA server and the LDAP server; for example, to list a property value:

```
property -l <property>
property -l bdna.rs.LDAP.connectionURL
```

And to set a property:

```
property -m <property> <value>
property -m bdna.rs.LDAP.connectionURL ldap://...
```

At the BDNA Server:

Set the following parameters at the command line:

Note: All but one of the properties listed in Table 7 are of the format `bdna.rs.LDAP.`*<property>*; for example, `bdna.rs.LDAP.connectionURL`. The lone exception is `allowAllLDAPUser`, the format of which is `bdna.rs.access.allowAllLDAPUser`.

Table 7: LDAP Properties for BDNA

Property	Description	Examples
`bdna.rs.LDAP.connectionURL`	Connection URL for the LDAP server. Use the following format: `ldap://<server>:<port>`	`ldap://ldap.bdnacorp.com:389`
`bdna.rs.LDAP.alternateURL`	URL of an alternate LDAP server.	`ldap://localhost:389`
`bdna.rs.LDAP.roleBase`	Distinguished Name (DN) of the top-level LDAP entry; it is used when searching for the role entry. If not specified, the top-level element in the directory context is used.	`CN=Users,DC=bdncorp,DC=com`
`bdna.rs.LDAP.roleName`	Name of the LDAP attribute (typically `cn`) that contains the LDAP group name in the LDAP directory entries found by a role search. You can use the `userRoleName` property to specify the name of an LDAP attribute in a user entry containing additional role names. If `roleName` is not specified, a role search does not take place and roles are taken only from the user entry.	`cn`

Table 7: LDAP Properties for BDNA (Continued)

Property	Description	Examples

`bdna.rs.LDAP.roleSearch`	LDAP filter expression used for conducting role searches.A role search is conducted on the LDAP directory object represented by the DN in `bdna.rs.LDAP.rolebase`.
	A {0} in the filter indicates where to substitute for the DN of the user; a {1} indicates where to substitute the username. You can use either or both values.
	If not specified, a role search does not occur and roles are taken only from the attribute in the user entry specified by the `userRoleName` property.
		`bdna.rs.LDAP.roleSearch=(&(uniqueMember={0})(description=manager)`
		`bdna.rs.LDAP.roleSearch=member={0}`

`bdna.rs.LDAP.roleSubtree`	Boolean that controls whether BDNA searches under the LDAP node specified by roleBase. Values: true or false. If the roles are in an LDAP sub-group, value must be true.	`true`
`bdna.rs.LDAP.authentication`	Type of authentication to be used in an LDAP bind operation. This is based on your LDAP server implementation. (Binding allows the client access to the LDAP server.) If this property is not set, the default authentication mechanism is "simple".	**sasl_mech**: Space-separated list of SASL mechanism names. Use one of the SASL mechanisms listed. For example: "CRAM-MD5" uses the CRAM-MD5 SASL mechanism, as described in RFC 2195. **none**: Use no authentication (anonymous) **simple**: Use weak authentication (clear-text password) For further info, see: http://docs.oracle.com/javase/jndi/tutorial/ldap/security/auth.html

Table 7: LDAP Properties for BDNA (Continued)

Property	Description	Examples
`bdna.rs.LDAP.rootdn`	Rootuser the Discover Client uses to bind to the LDAP directory; user must have read access so it can look up information in the LDAP directory. See detailed information here: http://docs.oracle.com/javase/jndi/tutorial/ldap/security/ldap.html This value depends upon the type of authentication specified in `bdna.rs.LDAP.authentication`. For Microsoft Active Directory, this value can take a number of forms. See detailed information here: https://msdn.microsoft.com/en-us/library/cc223499.aspx	`bdnacorp.com\bdnauser` `CN=bdnauser,CN=Users,DC=bdnacorp,DC=com`
`bdna.rs.LDAP.rootpw`	The password for the user specified by `bdna.rs.LDAP.rootdn`.	bdna
`bdna.rs.LDAP.useSSL`	Determines whether or not the LDAP server uses SSL. False by default, unless the site uses LDAP with SSL (LDAPS). In that case, `useSSL` is always true.	false
`bdna.rs.LDAP.userBase`	Distinguished name of the LDAP directory top-level entry to use when searching for the user entry. If not specified, uses the top-level element in the directory context. When `userdn` is specified, the `userBase` is appended to the value to form the user DN used for authentication.	`CN=Users,DC=bdnacorp,DC=com`

Table 7: LDAP Properties for BDNA (Continued)

Property	Description	Examples
`bdna.rs.LDAP.userdn`	This optional parameter is the LDAP attribute that identifies a user in the LDAP user directory (e.g., cn, uid). If set, the value is used to determine the DN of the user who logs into the Discover Client UI instead of searching for the DN in the LDAP directory. If not set, the DN of the user who logs into the Discover Client UI is found by searching LDAP on the basis of `bdna.rs.LDAP.userBase`, using the search filter `bdna.rs.LDAP.userSearch`. When this property is specified, the system constructs the user DN in the format `userdn=username{,}userBase.` For example, using the following settings and parameters: - userBase: CN=Users,DC=bdnacorp,DC=com - userDN: CN - username: John Smith - password: goodday the system will attempt to bind to LDAP with the user entry “CN=John Smith,CN=Users,DC=bdnacorp,DC=com” using the password “goodday”. The user is considered authenticated if the binding is successful. This property supersedes the userSearch and userSubtree parameters (i.e. the latter two are not be used when this is specified).	`CN` `uid`

Table 7: LDAP Properties for BDNA (Continued)

Property	Description	Examples
bdna.rs.LDAP.userRoleName	Name of an LDAP attribute in the user LDAP directory entry containing zero or more values for the names of roles assigned to this user. Can also be used to specify the name of an attribute to be retrieved from individual role entries found by searching the directory. If `userRoleName` is not specified, all the roles for a user derive from the role search.	`memberof`

bdna.rs.LDAP.userSearch	Use this LDAP filter expression when searching for a user directory entry on the search base of `bdna.rs.LDAP.userBase`
	{0} indicates where the actual username should be inserted. Use this property (along with the `userBase` and `userSubtree` properties) to search the directory for the user entry.
		`sAMAccountName={0}`
		`(&(uid={0})(description=Manager))`

bdna.rs.LDAP.userSubtree	Boolean controls whether BDNA searches under the node specified by `userBase`. If the users are in a subgroup, this must be true.	true

Setting Authentication Mode

The property `bdna.rs.AuthenticationFacility` sets the authentication mode: BDNA (the default) or LDAP. After setting the properties listed in Table 7 on page 145, set the value of `bdna.rs.AuthenticationFacility` to `ldap`. Whenever you change this property, you must restart RS for the change to take effect.

If running LDAPS (LDAP with SSL):

If you decide to enable SSL, `useSSL` must be set to true, and the port number must point to the SSL server.

To set LDAPS on Windows Server 2012:

1. If the Active Directory Certificate Services is not already installed on the Active Directory server, do so now. We have provided a summary of the procedure here. For more detailed information, see: http://careexchange.in/how-to-install-certificate-authority-on-windows-server-2012.
   - 1.1. Bring up the Server Manager.
   - 1.2. Select Add roles and features.
   - 1.3. On the "Before you begin" page, click Next
   - 1.4. On the "Select installation type" page, select "Role-based or feature-based installation" and click Next.

- **1.5.** On the "Select installation type" page select "Select a server from the server pool". Make certain that the server on which you want to install CA is selected. Click Next.
- **1.6.** On the "Select server roles" page, under "Active Directory Certificate Services" select "Certification Authority", select "Certificate Authority Web Enrollment". Click Next.
- **1.7.** On the Select features page, click Next.
- **1.8.** On the Confirm installation selections page, click Install.
- **1.9.** Configure the Active Directory Certificate Services according to your site-specific requirements.

It is also necessary to import the SSL server key, enabling Microsoft Active Directory for SSL access. Perform the following procedure:

2. Verify that SSL has been enabled on the Active Directory server:

- **2.1.** Ensure that Windows Support Tools is installed on the Active Directory machine. The `suptools.msi` setup program is located in the `\Support\Tools` directory on the Windows installation CD.
- **2.2.** Select “Start > All Programs > Windows Support Tools > Command Prompt”.
- **2.3.** To start the `ldp` tool, go to the command prompt and enter `ldp`.
- **2.4.** From the `ldp` window, select “Connection > Connect”.
- **2.5.** Supply the host name and port number (636).
- **2.6.** Select the SSL check box.

Note: Be certain that the Active Directory domain server name has been entered correctly.

If the connection is successful, a window appears listing information related to the Active Directory SSL connection.

If the connection is unsuccessful, restart the system and repeat this procedure.

To Export the CA certificate from the Active Directory server:

1. Log on as a Domain Administrator to the Active Directory domain server.

2. Export the certificate from the Active Directory server to a file:

- **2.1.** Open the CA Microsoft Management Console (MMC).
- **2.2.** Highlight the CA machine and right-click to select Properties for the CA.
- **2.3.** From the General menu, click “View Certificate”.
- **2.4.** Select the “Details” view.
- **2.5.** Click Copy To File to save the CA certificate in a file.

Note: You can save the CA certificate in either DER Encoded Binary X-509 format or Based-64 Encoded X-509 format.

---

3. Import the CA certificate to the BDNA server setup:

3.1. Log in to bdna server install machine.

3.2. At the command prompt, enter:

```
cd $BDNA_HOME/nih/jdk/jre/ keytool -import -file servert_cert.cer
-keystore keystore
```

4. Set up BDNA properties, as previously noted in "Setting BDNA Properties for LDAP through BDNA Discover."

5. Start the BDNA server and initiate a connection.

Connecting BDNA to LDAP

You can use the following script to set up a BDNA Collection Store for LDAP. Be sure to change all of the property values to those that are appropriate for your environment.

To change individual properties, follow the procedure in either "Setting BDNA Properties for LDAP through BDNA Discover" on page 143 or in "Setting BDNA Properties for LDAP through the Command Line" on page 145.

Note: You can also use the following script to set up a BDNA FactBase for LDAP. However, before doing so, be sure to change `initdb` to `initfb`. To learn more about the `initdb` and `initfb` commands, refer to "BDNA Shell Command Reference."

```
sh bdna.sh << EOF
initdb -y
property -m bdna.rs.AuthenticationFacility ldap
property -m bdna.rs.LDAP.connectionURL ldap://vm-nm01:389
property -m bdna.rs.LDAP.alternateURL ldap://vm-nm02:389
property -m bdna.rs.LDAP.roleBase "CN=Users,DC=bdnacorp,DC=com"
property -m bdna.rs.LDAP.roleName cn
property -m bdna.rs.LDAP.roleSearch "member={0}"
property -m bdna.rs.LDAP.roleSubtree true
property -m bdna.rs.LDAP.rootdn "CN=bdnauser,CN=Users,DC=bdnacorp,
DC=com"
property -m bdna.rs.LDAP.rootpw bdna
property -m bdna.rs.LDAP.useSSL false
property -m bdna.rs.LDAP.userBase "CN=Users,DC=bdnacorp,DC=com"
property -m bdna.rs.LDAP.userSearch "sAMAccountName={0}"
property -m bdna.rs.LDAP.userSubtree true
property -m bdna.rs.access.allowAllLDAPUser true
list -a
startup
list -a
exit
EOF
```

After running the above script, BDNA should be configured to use LDAP for authentication. To test that the LDAP configuration is working, attempt to log in to the BDNA Discover Client. For more information on BDNA Shell commands relating to LDAP and their usage, refer to "BDNA Inventory Component Reference" on page 62.

Monitoring Discovery 11

About this Chapter

This chapter discusses the various ways to monitor scans:

- Scan Status and Scan Progress
- Scan Monitoring Reports

What to Look For from a Scan

Discovery—also called a *scan*—is an iterative process. Discovery begins with a few baseline query/response sessions to all of the accessible devices on the network. The remainder of the discovery process involves queries that are based on the responses received, effectively creating a short-term feedback loop.

For example, BDNA Discover begins with an ICMP ping scan. For every live IP address discovered, it then goes back and queries that IP address to see if any ports from a specified list of ports are open. In this way, discovery returns results that differ depending upon the state of each individual asset at the time of the discovery. The overall duration and the amount of work to be completed depend on the number, type, and configuration of the devices on the network.

It is important for the BDNA Discover Administrator to make note of trends that occur during a discovery. These may indicate opportunities for optimization of the data collection and performance of future scans. Some important trends to monitor include:

- Discovery Zones being broken up into smaller ranges (class C or less) for discovery (also called *network componentizing*)
- Active device discovery via Nmapping scans
- Level 1 discovery activity such as OS typing and port fingerprinting
- Level 2 discovery activity such as base OS querying and application fingerprinting.

BDNA Discover makes use of two types of tasks:

- *Scan tasks*

  Scan tasks are what the user creates in the Scan Administration application. They define what networks will be scanned, when they'll be scanned, and using what credentials.

- *Collection tasks*

  Collection tasks represent the actual work done to collect information from target machines on the network. A single scan task generates thousands of collection tasks. Each collection task has a different purpose (for instance, to find if an IP is active; to collect the number of CPUs; or to collect tablespace information from an Oracle database) against a different IP address. Collection tasks return collection results that get processed by the rules engine and ultimately become the data viewed in the Analytics application.

Scan Status and Scan Progress

The Scan Administration application provides a variety of functions to help Administrators set up and monitor scan tasks. (For information about how to use Scan Administration to prepare a system for scanning, and set up scan tasks, refer to "Configuring and Running Level 1 Discovery" on page 35.) Scan Progress provides a useful means of determining the current progress of a running scan within a Collection Store (this functionality is not available in FactBase). Scan Progress provides dynamic, real-time status of a running scan.

One of the functions within Scan Progress is the availability of Scan Status information for every Scan Task that has been created. To view the status of a scan, open the Scan Administration application and select Scan Progress under Scan Monitoring. Scan Status will be displayed under the column labeled Status for each configured scan task.

Scan Status

Scan Status displays the status of all the scan tasks running at a given point in time. Each scan task will be listed with the current status under the Status column and the time at which that status reported will be listed under the Status Last Updated column.

Scan Status Messages

Scan Status returns a number of messages, including:

- Active (in shift): The scan task is currently in progress and in shift; it has started at the time it was scheduled to start.
- Active (out of shift): The scan task is currently outside a valid shift window; there is still more work left for the task. Once the shift starts again, work will resume.
- Complete: The scan task has been queued, processed, started, and completed.
- Incomplete (ran out of time): The scan task is beyond the calendar end date; for example, the scan task was configured to end on June 20 and today's date is June 21. When the end date occurred, the system determined that additional work still needed to be done, thus, the scan task is considered incomplete. This may result in partial data collection, which is visible in the Analytics application.
- Not Processed: The scan task has recently been created and the agenda manager has not yet had a chance to process it. Need to wait for agenda manager polling (every seven minutes) to process the scan task.
- Not Started: The agenda manager has processed the scan task and the current time is before the calendar start time for the scan task; for example, the scan task is scheduled to start on June 21, but today is June 19.
- Pending Completion: The scan task is beyond the calendar end date; for example, the scan task was configured to end on June 20 and today's date is June 21. The system is still processing results from the scan task and has already determined that no additional collection is necessary. Once the system is done processing, the scan task will enter the Complete state.
- Processing: The scan task is beyond the calendar end date; for example, the scan task was configured to end on June 20 and today's date is June 21. The system is still processing results from the scan task and determining if the scan task was actually finished when the calendar end date occurred. If the system determines the scan task finished its work, the scan task will go into the Complete state. If the system determines the scan task was unable to complete its work when the calendar end date occurred, it will go into the Incomplete (ran out of time) state.
- Processing (in shift): The scan task is currently inside a valid shift window. The scan task is not currently doing collection but the system is processing results from the scan task and determining if more work needs to be done. If so, the state will change to Active (in shift) and scanning will continue shortly; if not, the scan task will enter the Complete state.

- Processing (out of shift): The scan task is currently outside a valid shift window. The system is still processing results from the scan task and determining if more work needs to be done. If so, the state will change to Active (out of shift) and scanning will continue once the shift starts again; if not, the scan task will enter the Complete state.
- Stalled: The scan task was in progress and in shift, but the system is now idle and there are still incomplete tasks pending. (Agenda Manager, CLE, PerlCS, or CLM may be busy or temporarily offline. Examine the logs to ascertain further status.)
- Unknown: The status could not be determined.

Shifts

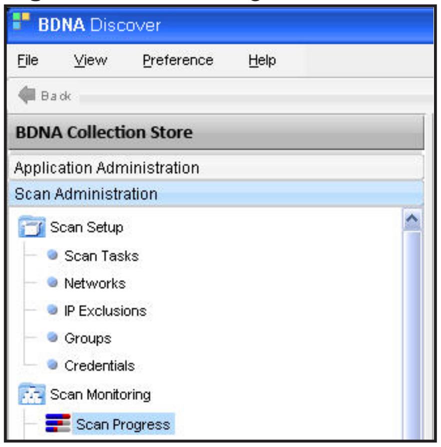
Just as a human may work the night shift, day shift, or swing shift, a scan task that is scheduled to start during a particular time period is considered to have its own shift. When the Scan Administration application reports on the status of a scan task, it includes observations about its execution with respect to the time it was scheduled to begin or end. In shift and out of shift are defined as follows:

- In shift:
  - The scan task is inside a valid calendar window; for example, the scan task is allowed to run between June 10 and June 20 and today is June 12.
  - The scan task is inside a valid shift window; for example, the scan task is configured to run Mon, Wed, Fri from 9 AM – 5 PM and today is Monday and the current time is 10 AM.
- Out of shift
  - The scan task is outside a valid calendar window; for example, the scan task is allowed to run between June 10 and June 20 and today is June 22.
  - The scan task is outside a valid shift window; for example, the scan task is configured to run Mon, Wed, Fri from 9 AM – 5 PM and today is Tuesday and the current time is 10 AM.

Monitoring the Progress of a Scan

Access the Scan Progress feature from the Scan Administration application, as shown in Figure 51 on page 155.

Figure 51: Scan Progress: Menu Item

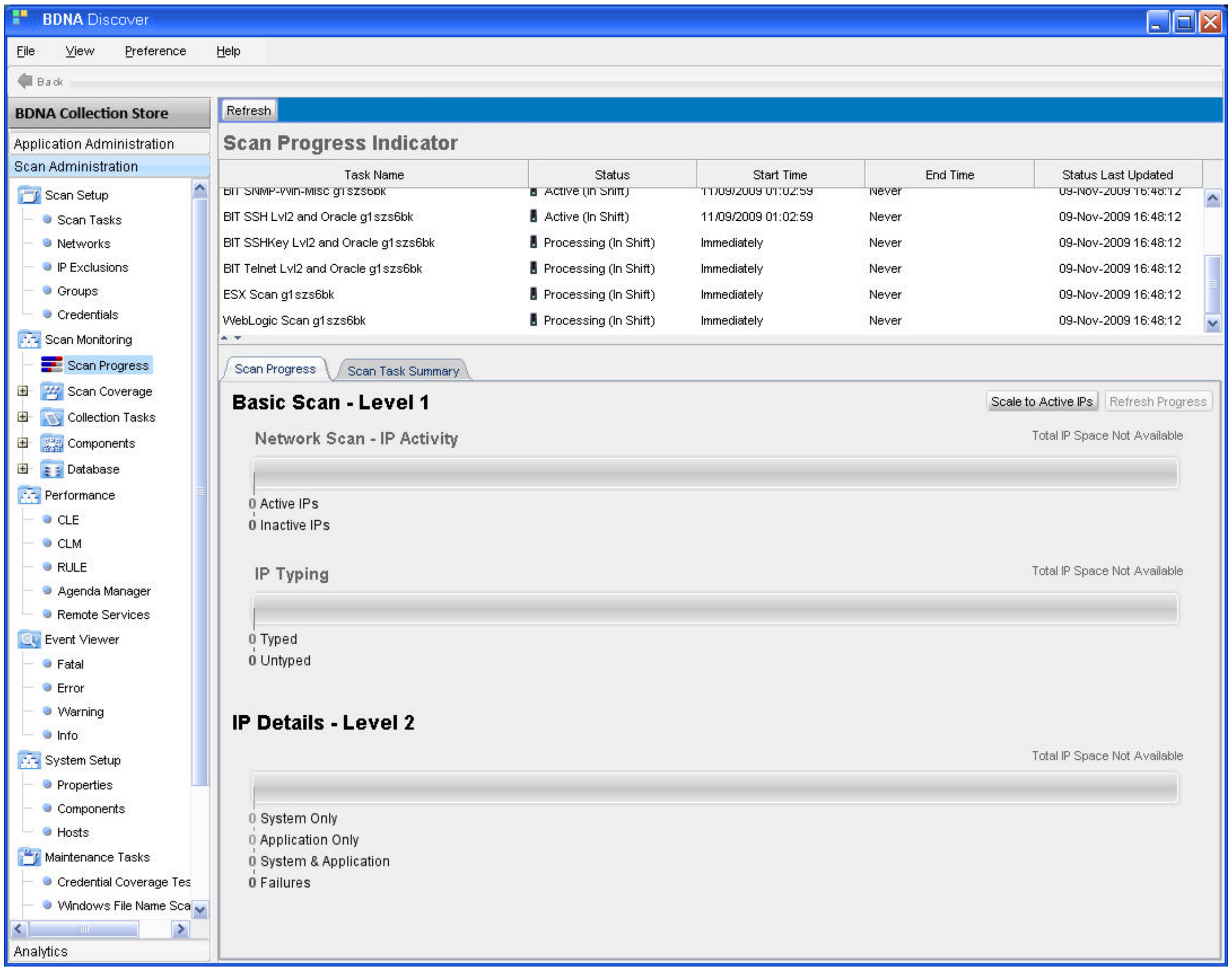


To monitor the progress of an existing scan:

1. From the Scan Administration application, select Scan Progress.

   The Scan Progress Indicator displays, as shown in Figure 52 on page 156.

Figure 52: Scan Progress Indicator: Main Page



Scan Progress Indicator

Task Name	Status	Start Time	End Time	Status Last Updated
BIT SNMP-Win-Misc g1szs6bk	Active (In Shift)	11/09/2009 01:02:59	Never	09-Nov-2009 16:48:12
BIT SSH Lvl2 and Oracle g1szs6bk	Active (In Shift)	11/09/2009 01:02:59	Never	09-Nov-2009 16:48:12
BIT SSHKey Lvl2 and Oracle g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:48:12
BIT Telnet Lvl2 and Oracle g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:48:12
ESX Scan g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:48:12
WebLogic Scan g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:48:12

Basic Scan - Level 1

Network Scan - IP Activity

Total IP Space Not Available

0 Active IPs
0 Inactive IPs

IP Typing

Total IP Space Not Available

0 Typed
0 Untyped

IP Details - Level 2

Total IP Space Not Available

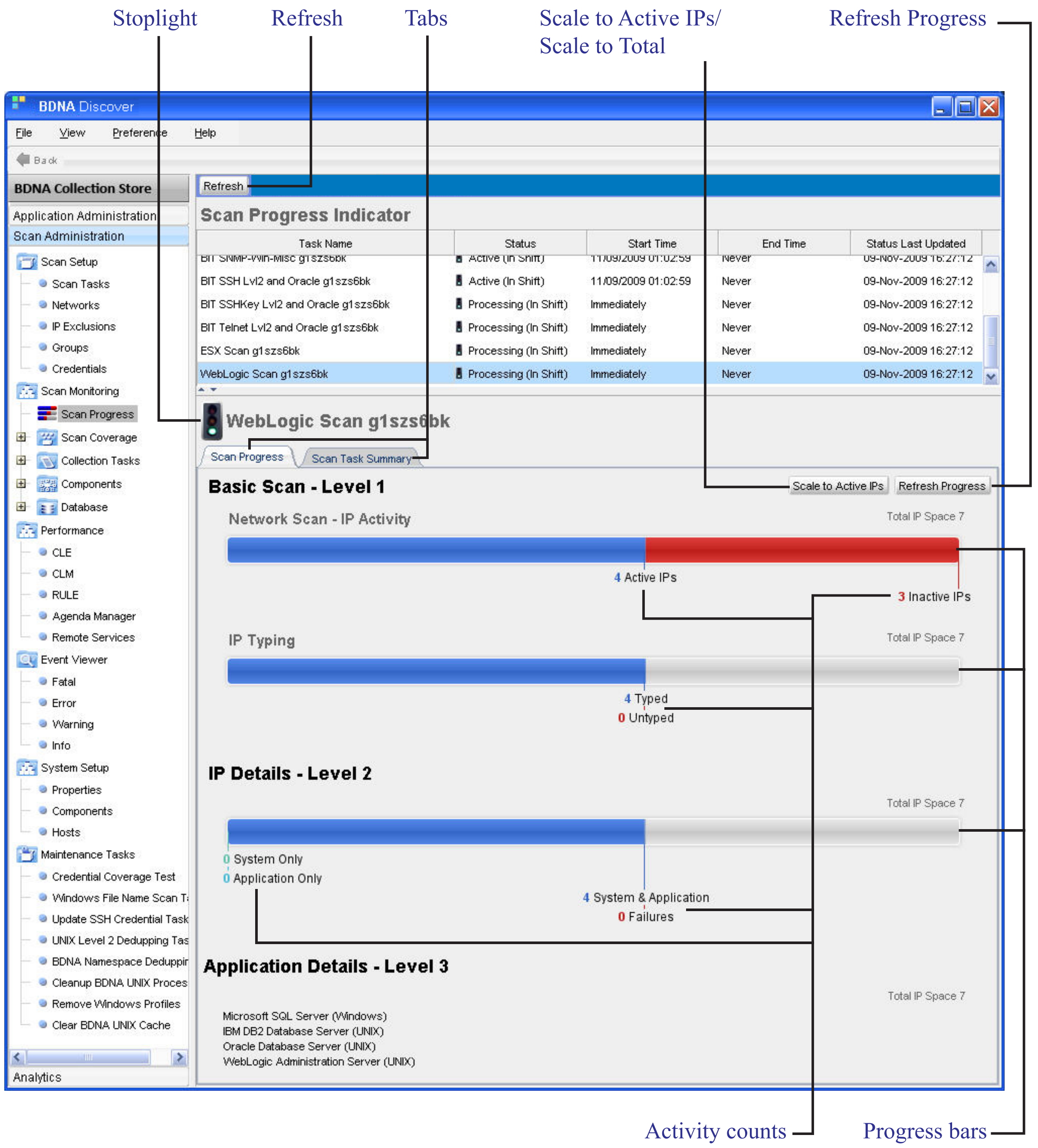
0 System Only
0 Application Only
0 System & Application
0 Failures

2. Click on an entry from the list of tasks at the top of the screen.

   It may take a few moments for the application to refresh. The Scan Progress Indicator displays the progress of the scan you selected, as shown in Figure 53 on page 157.

Tip: The callouts in Figure 53 on page 157 are hypertext links to the various Scan Progress features; click on a callout to view the description of that functionality.

Figure 53: Scan Progress Indicator: Main Page, with Results



Scan Progress Indicator

Task Name	Status	Start Time	End Time	Status Last Updated
BIT SNMP-Win-Misc g1szs6bk	Active (In Shift)	11/09/2009 01:02:59	Never	09-Nov-2009 16:27:12
BIT SSH Lvl2 and Oracle g1szs6bk	Active (In Shift)	11/09/2009 01:02:59	Never	09-Nov-2009 16:27:12
BIT SSHKey Lvl2 and Oracle g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:27:12
BIT Telnet Lvl2 and Oracle g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:27:12
ESX Scan g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:27:12
WebLogic Scan g1szs6bk	Processing (In Shift)	Immediately	Never	09-Nov-2009 16:27:12

WebLogic Scan g1szs6bk

Basic Scan - Level 1

Network Scan - IP Activity — Total IP Space 7

4 Active IPs
3 Inactive IPs

IP Typing — Total IP Space 7

4 Typed
0 Untyped

IP Details - Level 2 — Total IP Space 7

0 System Only
0 Application Only
4 System & Application
0 Failures

Application Details - Level 3 — Total IP Space 7

Microsoft SQL Server (Windows)
IBM DB2 Database Server (UNIX)
Oracle Database Server (UNIX)
WebLogic Administration Server (UNIX)

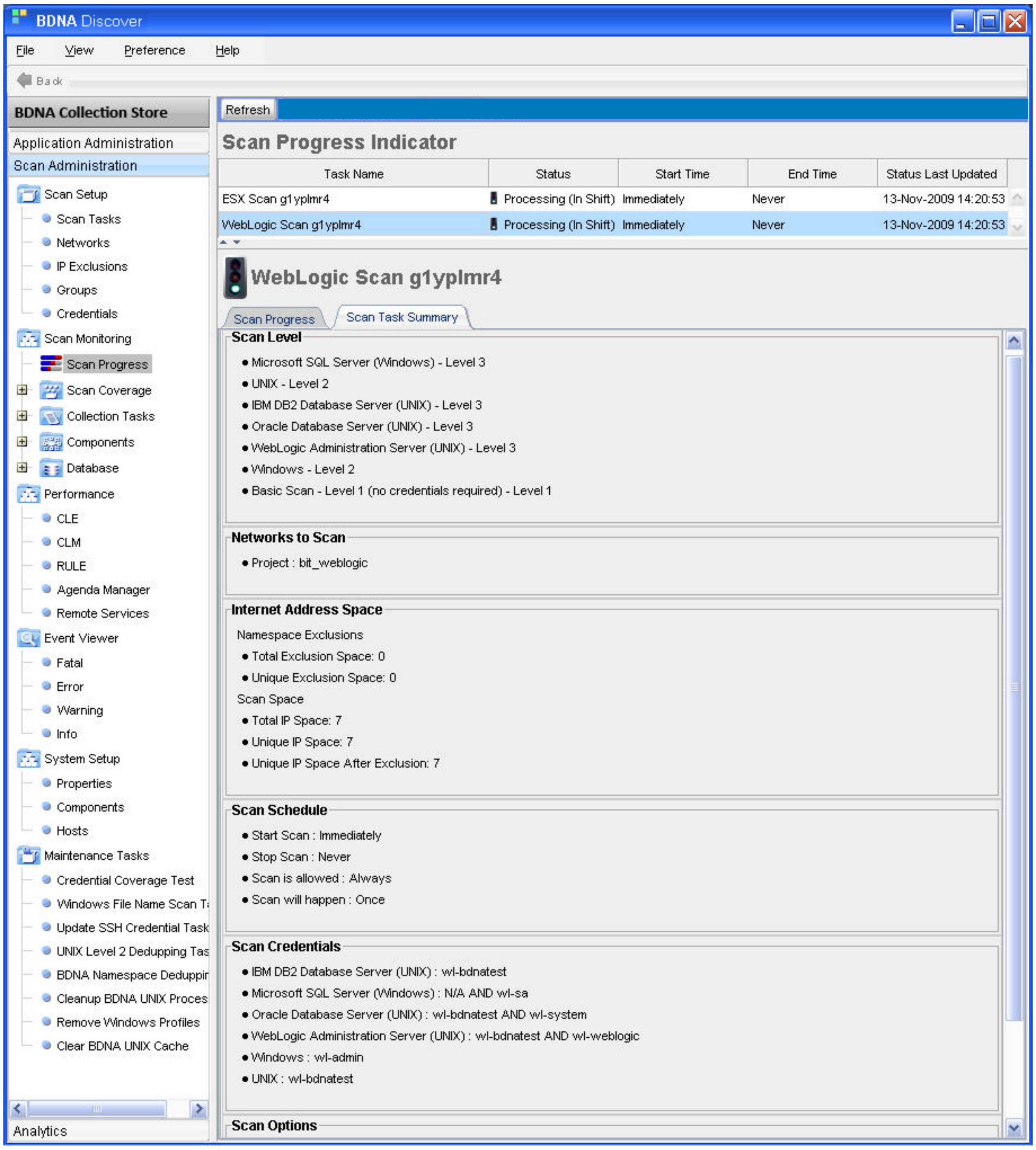
The various fields embedded within the BDNA Discover Scan Progress functionality are defined on page 158.

Activity counts	Numeric display of current scan activity. Click on any value to view detailed scan information for each entity. Refer to "Scan Status" on page 154 for descriptions of the various scan states. Back
Progress bars	Visual display of current scan activity. Each color (blue, red, teal) denotes a different status; refer to the labels adjacent to each bar to ascertain the context associated with a given color. Colored portions of the bar are hyperlinks to the scan data represented in the region that was clicked. The Total IP Space label at the right of the progress bars corresponds to the number of IPs that were defined in the New Scan Task wizard (see Figure 6 on page 40). Refer to "Scan Status" on page 154 for descriptions of the various scan states. Back
Refresh	Click this button to refresh not only the status for *all* of the current scan tasks (listed at the top of the screen), but also the progress information for the currently selected task. Because it is also refreshing the progress of the selected task, this functionality can impact overall scan performance and thus should be used judiciously. Also keep in mind that, depending on system activity, refreshes may take some time to complete. The status value of a scan task is updated on a seven-minute cycle that is not tied to the functionality of this button. The latest refresh time is listed under the Status Last Updated column. Back
Refresh Progress	Click this button to refresh the progress of the *currently selected* scan task. As is the case with the Refresh button, this functionality impacts overall network performance and should be used judiciously. Back
Scale to Active IPs/ Scale to Total IP Space	Akin to a zoom button, this functionality toggles the scale of the progress bars between active IPs (typed versus untyped) and all IPs in the range. Back
Stoplight	An icon that summarizes the status of a scan. • Red light: complete, incomplete, and unknown • Yellow light: stalled, processing (out of shift) • Green light: active (in shift and out of shift), pending completion, processing/processing (in shift) • No stoplight: not started, pending, initializing, and not processed Refer to "Scan Status" on page 154 for descriptions of the various scan states. Back
Tabs	Tabs to toggle between Scan Progress and Scan Task Summary views. Back

3. To view the summary of the selected task, click the Scan Task Summary tab.

The scan task summary screen displays, as shown in Figure 54.

Figure 54: Scan Progress Indicator: Scan Task Summary Tab



Scan Progress Indicator

Task Name	Status	Start Time	End Time	Status Last Updated
ESX Scan g1yplmr4	Processing (In Shift)	Immediately	Never	13-Nov-2009 14:20:53
WebLogic Scan g1yplmr4	Processing (In Shift)	Immediately	Never	13-Nov-2009 14:20:53

WebLogic Scan g1yplmr4

Scan Level
- Microsoft SQL Server (Windows) - Level 3
- UNIX - Level 2
- IBM DB2 Database Server (UNIX) - Level 3
- Oracle Database Server (UNIX) - Level 3
- WebLogic Administration Server (UNIX) - Level 3
- Windows - Level 2
- Basic Scan - Level 1 (no credentials required) - Level 1

Networks to Scan
- Project : bit_weblogic

Internet Address Space

Namespace Exclusions
- Total Exclusion Space: 0
- Unique Exclusion Space: 0

Scan Space
- Total IP Space: 7
- Unique IP Space: 7
- Unique IP Space After Exclusion: 7

Scan Schedule
- Start Scan : Immediately
- Stop Scan : Never
- Scan is allowed : Always
- Scan will happen : Once

Scan Credentials
- IBM DB2 Database Server (UNIX) : wl-bdnatest
- Microsoft SQL Server (Windows) : N/A AND wl-sa
- Oracle Database Server (UNIX) : wl-bdnatest AND wl-system
- WebLogic Administration Server (UNIX) : wl-bdnatest AND wl-weblogic
- Windows : wl-admin
- UNIX : wl-bdnatest

Scan Options

4. Optional: Edit or delete an existing scan task (as applicable).

To edit a scan task:

4.1. In the Scan Administration application, select Scan Tasks under Scan Setup.

4.2. Identify the appropriate task from the list that displays.

4.3. Either double-click the entry in the Scan Tasks column or highlight it and click the Edit button

4.4. Make your changes in the Edit Scan Task wizard.

or

To delete a scan task, follow the instructions in "Deleting Scan Tasks" on page 45.

Scan Monitoring Reports

The Scan Administration application contains reports useful for monitoring a BDNA Discover scan in progress. Unlike Analytics reports, which are static once built, Scan Monitoring reports are refreshed each time you view them. Consequently, in large scans, it may take a few moments for some reports to appear.

Some of the Scan Monitoring reports are more relevant at the early stages of a scan, while some reports are more relevant during the latter stages of a scan. All of the available Scan Monitoring reports contain features to aid in data navigation, to enable focusing on certain aspects of the returned data. Near real-time monitoring is also possible through the Scan Monitoring reports.

The Scan Monitoring reports interface has the following features:

- Query and Filter Saving

  Where applicable, reports that have query capability (such as time-based filtering, component selection, number of maximum results, and the like) can have the query or filter saved. The saved query or filter can then be exported to a file and imported into another Collection Store. This makes it possible to configure custom and commonly used queries or filters and reuse them. For example, if the user creates a complex expression in the Event Viewer to look for fatal errors in the past five hours that contain the strings *timeout* or *connection*, this can then be saved and reused later in the same Collection Store and even in an entirely new one.

- Scan Task Filtering

  Where applicable, reports may be filtered by Scan Task. If this feature is available for a particular report, a drop-down list appears at the top of the report next to the text "Task:". Inside the drop-down list, all, some, or one of the existing Scan Tasks may be selected as the filter criteria. This feature is especially useful when multiple Scan Tasks are running concurrently – the filter will allow only the data of interest to be shown.

- Time-based Filtering

  Where applicable, reports may be filtered based on the timestamp of the event being reported on. If this feature is available for a particular report, a drop-down list appears at the top of the report with the words "In the Last" for indicating the period by which to constrain the report data. Next to this drop-down list, a text box displaying the default numeric time value appears, followed by a drop-down list that indicates the default time increment. Possible values for the time increment may include minute(s), hour(s), day(s) and month(s). The first drop-down list for the time period may be changed to "In Between" instead of "In the Last". When "In Between" is selected, additional fields appear allowing a start date and time and an end date and time to be entered.

- Maximum Results

  Some of the reports allow for the maximum number of results returned to be specified. If this feature is available for a report, the words "Maximum Results" appear at the top of the report, followed by a text box that defaults to "500". It is important to note that this does not necessarily represent the maximum number of rows in the final report. Some rows may contain aggregations of more than one result. As such, this number represents the maximum number of results returned prior to aggregation. By specifying a small number of maximum results, the report can be made to run faster. Use this feature in conjunction with column sorting to find extremes, e.g. "show the 10 most recently created elements" or "show the top 50 longest running collection tasks".

- Window Group

Some reports are designed to show trends, such as task creation, over time. If this feature is available for a particular report, the words Window Group appear at the top of the report followed by a drop-down list. Currently, two supported time windows are supported: hourly and every five minutes. Changing this setting will change how the report groups its results. For example, for long-running scans (more than a day) it is desirable to see a high-level overview of how fast tasks were generated and executed. For this purpose, an hourly grouping on the report Task Flow Over Time should be used. Another use case is to diagnose if the system is currently working or has stalled. In this case, a window group of every five minutes in combination with a time filter *in the last* one hour would show what the system has done most recently.

- Automatic Refresh

  Many of the available reports support automatic refresh of the report data based on a fixed time interval. If this feature is available for a particular report, there will be a checkbox at the top of the report followed by the words “Automatic Refresh in” followed by a text box that defaults to “5”, followed by “Minutes”. It is not possible to specify a refresh time in an increment other than minutes. However, decimal values may be entered to specify refresh times shorter than one minute. It is not recommended to enter a value lower than .1 in this field. Check the box next to “Automatic Refresh” in order to enable automatic refresh for the selected report.

- Back Button

  Reports that support drilldown into the result set also enable the user to go back to the previous report view. This is accomplished by clicking on the back arrow (left facing arrow) in the report’s title/description bar.

- Column Sorting

  Each of the column headers sorts the results of the report by the values in that column, when clicked. One click will sort in ascending order (a–z, 0–9), which is represented by an upward pointing triangle. A second click will sort in descending order, which is represented by a downward pointing triangle. A third click will return the report to the default sort order and clear sorting settings from the selected column. Column sorting always applies to the entire result set. For example, if Maximum Results is set to 10 and the report has 100 rows, it’s likely that different rows will be returned for an ascending versus descending sort.

Report Drilldown

Report Drilldown is available in the Scan Monitoring reports. Many of the reports let you drilldown into a selected data point in order to see the underlying data. If this feature is available for a particular report, column values appear in blue text instead of black text. Clicking on the blue text in a data cell will open the drilldown report for the selected data.

Depending on the report, clicking on a drilldown can yield three types of detail reports:

- Element Details

  Lists all of the elements represented in the referring cell by name, along with modification and creation times. The element name in the element detail may also be drilled down, resulting in the Task Details report for that element.

- Task Details

  Lists all of the collection tasks associated with the referring cell, including the task type, task status, queue time, dequeue time, time allowed, deadline for completion, time completed, time to complete, CLE host, result code, and IP address. The task type, task status, and IP address columns all support drilldown and yield a Task Details report for the selected item.

- Task Details (Grouped)

Lists all of the collection tasks associated with the referring cell, grouped by task type, and including the completion code and the total number of tasks of this type with the same completion code. The total number of tasks may be drilled down resulting in the Task Details report for the grouped tasks.

Scan Coverage Reports

The Scan Coverage component of the Scan Administration application includes the following reports:

- Scan Summary by Network
- Level 1 Scan Summary by Network
- Level 2 Scan Summary by Network
- IP/Host Elements
- Level 2 Coverage
- Most Recently Modified Elements
- Element Creation and Modification Over Time

Details of these reports are documented in the following sections.

Scan Summary by Network

Scan Summary by Network provides summary values, by network, for the following elements:

Table 8: Scan Summary by Network

Element Name Value	Displays	Relevance
Networks/Groups	Name of the network or logical set	NA
Total IP Space	Total number of IPs in the network or group	NA
Unique IP Space	Total number of unique IPs in the network or group	NA
# Inactive IPs	Number of IPs that cannot be reached (not active in the network)	Drills down to a report that contains the list of inactive IP addresses
# Active IPs	Number of IPs that can be reached (active in the network)	Drills down to a report that contains the list of active IP addresses
# Untyped IPs	Number of IPs for which BDNA Discover cannot find the OS type	Drills down to a report that contains IP addresses and host names of IPs that are not typed to any OS
# Typed IPs	Number of IPs that can be typed to an OS	Drills down to a report that contains OS types and the count of IP addresses typed to that OS type
# L2 Sys Attempted IPs	Number of IPs where BDNA Discover attempted to perform Level 2 system (hardware) collection	Drills down to a report that contains the IP addresses, host names, and OS type for the IPs where BDNA Discover attempted to perform Level 2 system collection

Table 8: Scan Summary by Network (Continued)

Element Name Value	Displays	Relevance
# L2 Sys Succeeded IPs	Number of IPs where BDNA Discover succeeded in performing Level 2 system (hardware) collection	Drills down to a report that contains IP addresses, host names, and OS type for the IPs where BDNA Discover succeeded in performing Level 2 system collection
# L2 Sys Failed IPs	Number of IPs where BDNA Discover failed in performing Level 2 system (hardware) collection	Drills down to a report that contains the IP addresses, host names, OS type, and reason for failure for IPs where BDNA Discover failed to perform Level 2 system collection
# L2 Apps Attempted IPs	Number of IPs where BDNA Discover attempted to perform Level 2 applications collection	Drills down to a report that contains the IP addresses, host names, and OS type for IPs where BDNA Discover attempted to perform Level 2 application collection
# L2 Apps Succeeded IPs	Number of IPs where BDNA Discover succeeded in performing Level 2 applications collection	Drills down to a report that contains IP addresses, host names, and OS type for IPs where BDNA Discover succeeded in performing Level 2 application collection
# L2 Apps Failed IPs	Number of IPs where BDNA Discover failed in performing Level 2 applications collection	Drills down to a report that contains the IP addresses, host names, OS type, and reason for failure for the IPs where BDNA Discover failed to perform Level 2 application collection

Level 1 Scan Summary by Network

The values for Level 1 Scan Summary by Network are a subset of those for Scan Summary by Network and are documented in Table 8 on page 162.

Level 2 Scan Summary by Network

The values for Level 2 Scan Summary by Network are a subset of those for Scan Summary by Network and are documented in Table 8 on page 162.

IP/Host Elements

The IP/Host Elements report shows the corresponding IP elements that have been created or modified by the Rule Engines. For every active IP discovered, the Rule Engines will create an IP element (or modify an existing one, in the case of a rescan), which will then appear in this report. As such, this report can be used to gauge how far behind the Rule Engines are in processing raw discovered active IP addresses.

For every IP element created (or modified in the case of a rescan), a Core Level 1 discovery task will be performed on the target IP (refer to “Task Summary by Type of Task” on page 167). Based on the results of these discovery tasks, the Rule Engines may create (or modify) a host element. Because of this, the number of host elements created will always be less than or equal to the number of IP elements created. (Unless the IP element creation occurred outside the time-based filter at the top of the report and the host element was created within that time.)

Given a sufficiently long time-frame, the number of elements created will always equal the number of elements modified.

Drilldown is available on the number of IP elements created and modified, as well as on the number of host elements created and modified. Drilling down will yield the element details (refer to “Report Drilldown” on page 161 for details).

Level 2 Coverage

The Level 2 Coverage report lists the following information:

- UNIX and Windows operating system types supported by BDNA Discover Level 2
- Level 1 count: The number of operating systems of this type found during a Level 1 scan
- Level 2 attempted: The number of operating systems where Level 2 collection was attempted
- Level 2 attempts as a percentage of Level 1 counts
- Level 2 successes: The number of operating systems where Level 2 collection was successful
- Level 2 successes as a percentage of Level 2 attempts
- Level 2 failures: The number of operating systems where Level 2 collection was attempted but failed entirely. In cases where multiple credentials were tried, none of the credentials worked.

Level 1 counts, Level 2 attempts, Level 2 successes, and Level 2 failures all support drilldown. Drilling down yields element details for each (refer to “Report Drilldown” on page 161).

It is important to note that Rule Engine processing is required for Level 2 successes to be registered in this report. For a more real-time indication of Level 2 success, open the Task Summary report, drill down on the tasks that have completed successfully, and look for tasks of type Win*Static, LinuxStatic, SolarisStatic, etc.

Most Recently Modified Elements

The Most Recently Modified Elements report lists the most recently modified elements in a discovery. For elements that were just created, their modification and creation time are equal. The report contains an element name column that provides a simple description of the element. Other columns display the element count and the most recent creation/modification time for the element type.

Drilldown is available on the counts column and gives the element details associated with the element name (refer to “Report Drilldown” on page 161).

Key element names and their relevance are listed in the following table:

Table 9: Most Recently Modified Elements Report Possible Values for Elements Listed

Element Name Value	Displays	Relevance
Internet Addresses	Total active IP addresses	Gives an idea of the size of the overall scan as well as the Rule Engine progress during a Level 1 scan.
Hosts	Total devices, typically equal to or less than Internet Addresses	Useful for comparing actual device counts against expected device counts.
[Operating System Names]	Various operating system names (e.g., Windows 2008, Windows 2012, Linux, etc.)	Useful for evaluating typing and OS distribution.
Operating Systems	Untyped devices, typically the count decreases as the scan progresses	Useful for comparing against Hosts for typing ratio.
Cisco Switch NIC	Network interface cards found on Cisco switches	Useful for confirming SNMP access to Cisco switches during Level 1 and Level 2 discovery.
CPUs	Device CPUs, typically equal to or greater than number of devices with Level 2 access	Useful for confirming Level 2 discovery.
[Application Names]	Various application names (e.g. Microsoft Word, Netbackup Client, etc.)	Useful for confirming Level 2 discovery and application distribution.
Oracle Tablespace [on OS]	Tablespaces for Oracle installations	Useful for confirming Level 3 discovery.

The Most Recently Modified Elements report is useful for analyzing discovery results in real time and post scan. It is helpful in the following ways:

- Displays number of devices, operating systems, applications, and their configuration
- Shows that the Rule Engines are working and healthy (for example, when the timestamps are recent it indicates recent Rule Engine activity)
- Displays the last time Rule Engine activity occurred
- Gives a general idea of what the system is working on
- Gives visibility into what tasks will be scheduled shortly (e.g., if an application element was created, a follow-up task may result.)

Element Creation and Modification Over Time

The Element Creation and Modification Over Time report shows element creation and modification broken down by time windows. The time windows can be either hourly or every 5 minutes and are set with the drop down list labeled Window Group. This is crucial for showing the health of the Rule Engines over time.

Collection Tasks Reports

The following reports are located under the Scan Monitoring Collection Tasks report group:

“Task Summary”

“Task Summary by Type of Task”

“Task Flow Over Time”

“Currently Queued Tasks Over Time”

“Currently Queued Tasks”

“Task Backlog”

“Long-Running Tasks”

“Pending Unscheduled Discovery Tasks”

The Task Summary report and the Task Summary by Type of Task reports display Collection Tasks that are in one of the three active queues. Those queues are titled Queued, Active and Completed. Collection tasks are entered into these queues under the following conditions:

- Queued: Collection tasks in this queue are available to be executed. This means that they are part of a valid Scan Task, and they are inside of a valid Calendar to be run. A valid Calendar means that the time (now) is between a valid Scan Start time and End time, and within a valid Shift Start and End time. Tasks that are in the Queued queue are valid to be executed now, and will be executed as soon as capacity is available on a Collection engine of the appropriate type.
- Active: Collection tasks in this queue are currently being executed by a Collection engine. When a Collection Task is in the Active queue it has been dispatched and it will either complete or be terminated within a specified time limit.
- Completed: This queue holds tasks that have completed execution. The outcome for the Collection Task is report under the Completion Code.

When a Collection task is placed into the Queued queue it means it is available to be run now. If, while waiting in the Queued queue the Scan Task reaches its Scan End time, or if the Shift End time is reached, the collection task will be removed from the Queued queue as it is no longer in a valid state to be executed. The Pending Unscheduled Tasks report shows you collection tasks that have been created but cannot currently be executed because they are not in a valid calendar window. If these Collection Tasks move into a valid calendar window they will be moved back into the Queued queue and they will be executed.

Both the Task Summary Reports and the Pending Unscheduled Reports can run collection tasks when a calender is multi-shift. For example, a multi-shift scan task would allow for scanning M-F only, between 8am and 5pm, i.e., the Calendar is M-F and the shift is 8-5.

Task Summary

The Task Summary report gives a high level overview of queued, active, and completed collection tasks. For completed tasks, a completion code is provided that provides details about the task execution. For example, a completion code may indicate a successfully completed task, a task that could not connect to the remote host, a timed out task, a task where the credentialed login failed, etc. This report is very useful for getting a quick status about collection activity on the system.

The column “Total Number of Tasks” is available for drilldown. Clicking on a number in this column will display a grouped Task Details report for all collection tasks with the corresponding status (queued, active, or completed) and completion code description.

Task Summary by Type of Task

The Task Summary by Type of Task report lists the various collection tasks associated with discovery and the status of those tasks. The report contains columns for the collection task type, an internal ID for the type, and the associated counts for total, active, queued, and completed collection. Drilldown is available on the total, active, queued, and completed columns; yielding task details for the associated column (see the section titled "Report Drilldown", above).

The typical order of operations in a discovery involves tasks being created, queued, activated, and finally becoming completed. Keep in mind that the total number of collections can, and almost always will, increase over the duration of a scan, since tasks will continue to be created and placed into the queue. It is important to monitor the progress of the various discovery tasks to confirm that the tasks are being processed and, if applicable, newer tasks are being created.

Key task types and their relevance include:

- Active IP Addresses

  Networks are potentially componentized into "IP Ranges", which are no larger than a class C network. (Networks smaller than a class C do not get componentized.) For each IP Range, an Active IP Addresses task will perform a ping scan to find Active IP Addresses in the IP Range. Active IP Address elements will then be created by the Rule Engines. For example, if a class B network is being scanned during Level 1, it will be broken (componentized) into 256 class C-sized IP Ranges and there will be 256 Active IP Address collection tasks generated.

- Core Level 1 Attributes Attribute Set

  For each Active IP Address Element created by the Rule Engines, a Core Level 1 Attributes Attribute Set task will be queued. This task performs the initial Level 1 query against an individual IP address, during which the relevant open ports are discovered, and the operating system of the device is typed. The Rule Engines will then create elements for each port discovered, as well as an Operating System element for the device.

- Fingerprint Port

  For each Port element created by the Rule Engine, a Fingerprint Port task is queued to further query that port. Note that only a small number of major ports (e.g. telnet, HTTP, SSH, Oracle, etc) are explicitly fingerprinted in this manner.

- [Operating system name] Static

  For each Operating System element created by the Rule Engines for which there is a matching credential (i.e. Level 2 is to be performed) an appropriately named task is created to perform the base Level 2 discovery for the device. Names of these tasks include WinXPStatic, LinuxStatic, etc. When these tasks are active, Level 2 discovery is being performed. The Rule Engines will then create Application elements based on the results of the <OS>Static tasks.

- [Application name] Footprint Static

  For each application element created, an appropriately named task will be queued to further query the application. Note that these tasks only occur for applications where associated fingerprints exist, and not for all applications installed on the device (for example, viewable via the Windows Add/Remove Program listing on the machine). Names of these tasks include MSOfficeFootprintStatic, InternetExplorerFootprintStatic, etc. These tasks will only be queued when there has been successful Level 2 credential access to the machine, so when these tasks appear, it verifies that the credentials supplied to BDNA Discover are valid.

- namespaceDuplicateInfoStatic

These are heartbeat collection tasks. By default, one of these collection tasks will execute every hour to do system maintenance (sometimes referred to as housekeeping) such as dedupping. Thus, if the filter is set to show "in the last 24 hours", it will show a count of 24. This indicates a baseline level of system health. If these tasks do not appear or the majority of them are not completed, it may indicate a stagnant/overloaded system or that certain components are not running.

The Task Summary by Type of Task report's configuration pane allows the display to be manually refreshed by selecting the "Run Report" option. The display can be refreshed automatically by selecting the "Automatic Refresh" option, entering a refresh interval in the "Minutes" field, and selecting the "Query" option. More elements can be displayed by entering a higher value in the "Row Count" field and selecting the "Run Report" option. Time based filtering options are also available.

The Task Summary by Type of Task report is useful for monitoring discovery in a granular and trend-based view. It provides a view of BDNA Discover's active and queued tasks that can be used to monitor the overall discovery timeline. For example, if all of the <OS>Static tasks have completed (WinXPStatic, LinuxStatic, etc), and the only remaining tasks to be performed are <Application>FootprintStatic tasks (MSOfficeFootprintStatic, InternetExplorerFootprintStatic, etc.), then it is likely that the Level 2 scan is nearing completion. It is important to keep in mind, however, that new tasks can be added to the queue at any time during the scan, so new <OS>Static tasks can be added, which will in turn create new <Application>FootprintStatic tasks.

Task Flow Over Time

The Task Flow Over Time report shows the number of tasks queued and completed, broken down by time windows. The time windows can be either hourly or every 5 minutes. Drilldown is available on both the Tasks Enqueued and Tasks Completed columns, giving task details for both (see the section titled "Report Drilldown", above).

This report can help predict how long the current scan will take to complete, because it shows overall discovery activity. It can also help in the scheduling of future scans, since it provides an idea of how much the system can scan per time period.

Note that different types of collection tasks take different amounts of time to complete. For example, it's not uncommon to see a fast task completion rate (tasks completed per hour) during Level 1 and a slower rate during Level 2 or Level 3. The latter types of tasks are more complex and take more time to execute.

Currently Queued Tasks Over Time

Shows collection tasks that are currently in the queue (waiting to be executed) and when they were put in the queue. If the queue contains a large number of tasks, it's useful to know if they were queued up recently or a long time ago and also how quickly they were put into the queue. In general, it is desirable to have the queue times be as recent as possible. When collection tasks sit in the queue for a long time (more than a day), they have a higher chance of not being able to contact the target machine because it may have gone offline.

Currently Queued Tasks

A shortcut view to show the details of collection tasks that are currently in the queue (waiting to be executed). This view is the same as drilling down on the Queued collection tasks in the report Task Summary.

Task Backlog

Shows active (running) collection tasks and buckets them into hourly or every five minute time windows. For each time window, it shows when the active collection tasks were queued (put in the queue) and dequeued (picked up from the queue for execution). Typically, active collection tasks will have been dequeued very recently, usually no more than a couple of hours. On the other hand, an active collection task may have been put in the queue hours or days ago. In the latter case, it may indicate small shift windows (not enough time to execute the tasks in the allotted window), insufficient PerlCS/WinCS throughput, or a slow network.

Long-Running Tasks

Shows active and completed collection tasks that are taking or have taken an exceptionally long time to complete. The length of time is configurable as a query parameter. Typically, collection tasks should be dequeued (picked up by a PerlCS or WinCS) and completed within a few minutes. Some collection tasks are known to take a long time, such as those that run a `find` command on UNIX systems during level 2. In the latter case, collection tasks can take 30 – 90 minutes and sometimes longer.

Pending Unscheduled Discovery Tasks

The Pending Unscheduled Discovery Tasks report displays discovery tasks that are pending but not yet scheduled. Not all collection tasks that have been processed by the Agenda Manager show up as "queued" in the various Task reports. There may be tasks which have yet to be scheduled. For example, if a calendar has been created for the scan such that it runs only between 9am and 5pm, at 5:01pm there may be thousands of tasks that will be put into the queue the next day at 9am. These tasks would appear in the Pending Unscheduled Discovery Tasks report. Since these tasks don't yet exist, no drilldown is available for this report.

Diagnostics Reports

Host-Based Analysis

The Host Based Diagnostics report is located in the Collection Store under Scan Administration -> Scan Monitoring -> Diagnostics -> Host Based Analysis. To utilize this report, enter in a single scanned IP address (or multiple scanned IP addresses separated by commas), and then click on "Run Report". This displays details for each collection task of the specified type, such as task status, time to complete, CLE host, task ID, and other data.

Viewing the Detailed CLE and WinCS Log Files from the Host-Based Analysis Report

The detailed CLE and WinCS log files for each collection task are not available in the Host-Based Analysis report by default. If you want to view the log files, use the following procedure:

1. Run $BDNA_HOME/bin/mountWincsLog.sh and enter the parameters for the WinCS server. This runs a test that mounts the directory where the wincs logs are stored, and then un-mounts the directory. The data in the wincs logs is stored in the database for a later step in the process.
2. Open a bdna shell:
   ```
   > sh bdna.sh
   ```
3. Execute the following command to allow the BDNA Discover server to find and load the CLE logs from the Linux server, mount the WinCS log directory based on the information provided in step 1, and load the log files into the Host-Based Analysis section:
   ```
   > bdna> processLog
   ```
   When the processLog command completes it will un-mount the directory on the WinCS server.

Now that the log files have been loaded into the Host Based Analysis section, each "Task Type" will have a clickable blue link to review the logs for that Task type.

Note: Users that are only interested in the CLE logs from the Linux server may skip step 1 and only complete steps 2 and 3.

The processLog command should only be run when the scan has completed. As noted above, the details for each collection task will be available during the scan via the report. However the log files should not be “processed” using the processLog command until the scan has completed.

The mountWincsLog.sh script may be run anytime. The password that is provided to the WincsLog.sh script is encrypted in the database. The information is stored in the wincs_log_host table. It is only accessible to the application during the processLog command.

Anytime an initdb command is executed, the mountWincsLog.sh must be run again.

The WinCS installation process automatically creates a Windows share named “wincs_logs” (read-only), on the WinCS server. However, the BDNA Discover instance will not access the WinCS logs for Host Based Diagnostics until the mountWincsLog.sh script is used to provide credentials for the shared folder.

Note: The processLog command processes the BDNA Discover scan logs and loads them into the database for use by the Host-Based Diagnostics report. BDNA Discover reads all CLE logs from all running servers on a distributed installation, and all WinCS servers found by the mountWincsLog.sh script. The processed data is stored in the schema in the log_host_state and collection_log tables.

Components Reports

The following reports are all located under the Scan Monitoring Components report group:

Rule Execution Status

The Rule Execution Status report requires a short explanation of how the Rule Engines divide up their work, in order to understand the information this report displays.

The BDNA Discover discovery system groups blocks of active IPs into what are called network groups. Worker Rule Engines (RULE1, RULE2, etc) work on one network group at a time. The default maximum network group size is 5,000 IPs. In the default configuration, BDNA Discover has one manager Rule Engine (RULE0) and one worker Rule Engine (RULE1). In this configuration there would be one network group for every 5,000 active IPs discovered. For example, if the system discovered 13,000 active IPs, there would be 3 network groups of sizes 5,000, 5,000 and 3,000. For systems with more than one worker Rule Engine, the system will divide discovered active IPs across Rule Engines until each Rule Engine has a full network group. At this point, the system will create a second network group for one of the Rule Engines. When that is filled up, the system will create another network group for one of the other Rule Engines. This process will continue as needed to account for all of the discovered IPs. Regardless of the number of worker Rule Engines, each network group will manage a maximum of 5,000 active IP addresses.

This report lists the various network groups, the component (Rule Engine) they are associated with, as well as other information about the network group, including when it was last scheduled and how long it has been idle.

- From the perspective of scan monitoring, the most important column is the “idle” column.

  All of the network groups must be idle in order for a scan to be considered complete.
- The “active” column indicates which network group is currently being processed.

  These groups are automatically rotated based on how much work needs to be done for the IP addresses managed by that group.
- The “splits” column usually reads zero.

If a network group runs out of memory, it will automatically split itself into more network groups. This may result in poor overall scan performance, especially if there is more than one split for any given network group. The most common cause of a split is that a worker Rule Engine's memory has been set too low. A value of 1.7 GB for worker Rule Engine memory is recommended for scans exceeding 5,000 active IP addresses. A split may also occur if a single network group has a large number of IPs being scanned at Level 2 and Level 3.

- The "rules fired" column indicates network group activity.

  If the value increases when the group is active, it indicates that the Rule Engine was working for the IP addresses in that group. If no rules were fired for that network group cycle, the group may be headed toward idleness (although more work may be generated as discovery progresses).

- The "priority" column shows which autogroup will be processed next.

  The "priority" value assigned to the autogroups is used by the Rule engine to decide which autogroup it should process next, provided there are multiple autogroups with new data collection to process.

  The Priority value will be updated to a different value depending on whether there is L1, L2, or L3 data to process. An autogroup with L1 data will get processed before an auto group with L2 or L3 data and an auto group with L2 data will be processed before one with L3 data.

  The Priority is set against autogroups that are not currently active. So the autogroup that is active will have a priority of zero. If all auto groups have a priority of zero, they will be processed in a round robin fashion.

- The "Last Scheduled Time" column shows how long it has been since an autogroup was scheduled for pickup.
- The "Last Scheduled Time" value shows how long it has been since the autogroup was scheduled for pickup, in the same way that the "idle" time shows how long the autogroup has been sitting idle.

Ideally, at the end of a scan you should see an idle time in hours, and all of the priorities showing as zero. Rule engines with a non-zero priority indicate there is data to be processed, and that you need to wait until it is completed before taking any system action such as exporting or shutting down.

How Current Is Agenda Manager

The How Current is Agenda Manager report shows the status of the Agenda Manager by time stamp. The Agenda Manager is responsible for scheduling tasks by placing them in the queue and removing them from the queue (as happens at the end of a Scan Task shift). If the Agenda Manager gets behind, it can appear as if the scan is stalled, since no new tasks are being placed in the queue. If it appears that the scan has stalled, the Agenda Manager can be ruled out as a problem when this report shows that the Agenda Manager is current.

The report shows two columns, "Status Object" and "Status Data". The most important Status Object is "Behind database changes by". The Status Data for this object uses the format "HH:MM:SS.SSS". A current Agenda Manager should be behind database changes only by a matter of seconds. An Agenda Manager that is behind by a matter of minutes or hours is not current and could be the cause of a scan stalling.

Note: If the system time of the Agenda Manager component machine is not the same as that of the Oracle database machine, the difference is reflected in the "Behind database changes by" column. Use NTPD to keep the system times for the servers in sync.

A current and idle Agenda Manager will also show "sleeping" in the "Status data" column. It is normal for some of these states to go into "working" during a scan. However, even with a current "Behind database changes by" time, if other statuses are stuck in a "working" state for a prolonged time, this indicates a very busy system. In such cases, scans may appear stalled or sluggish.

Database Reports

Free Tablespace

The Free Tablespace report is located under the Scan Monitoring Database report group. This report shows the default tablespace for the BDNA Discover database user and how much free space is available in that tablespace. Running out of tablespace during a BDNA Discover operation can result in partial data collection and often makes troubleshooting problems cumbersome. As such, free tablespace should be checked before a scan begins, periodically during a scan, and before report refresh to ensure adequate space is available.

Accessing the Event Viewer

The Event Viewer is a key tool in Scan Administration and is used to view events reported by all of the components running within the BDNA Discover installation. BDNA Discover logs major events and labels them depending upon their severity. The Event Viewer enables the BDNA Discover Administrator to search for major application events, filtered based upon severity, from a single easy to use report-like interface. In cases where the BDNA Discover Administrator suspects that a specific component is having trouble, it is possible to search the Event Viewer by component, to quickly evaluate if the component in question is in a desirable state.

The most important events to look for are those of level ERROR and FATAL. The following is a step-by-step procedure to search for these events using the Event Viewer:

1. Log in to BDNA Discover as an admin level user and select Scan Administration.
2. Select the Event Viewer in the left navigation pane.
3. Open the Status drop-down menu, and select the checkboxes next to ERROR and FATAL.
4. Click Search.

While there may be events of status ERROR that can be ignored, those of type FATAL are serious. They indicate catastrophic events from which BDNA Discover could not recover.

SanityCheck Command in the BDNA Shell

Most of the administrative and monitoring tasks for BDNA Discover should be performed using the Scan Administration application. However, a very important post-discovery command only available in the BDNA Shell is `sanityCheck`. The command runs through various tests to confirm that the internal state of the system is valid. Typically, all tests should pass. Any failed tests should be considered during the post-discovery procedure and accounted for in results analysis. Note that this command does not guarantee the quality of the discovered data. It only checks for catastrophic failures during discovery, and for consistency within each component.

Note: `sanityCheck` should be run *only after* all scanning activity is done. This means no active collection is happening, rule engines and agenda manager are idle, and so on. If `sanityCheck` is run during a scan, it may return false positives.

Note: `sanityCheck` should be run only against schemas that have performed a scan. If the schema was created through blending, the necessary transactional tables are not populated; thus, `sanityCheck` may report false positives.

For more information on blended repositories (combining discovery data using the blender command), refer to the *DNA Discover Administration Tutorial, Module 1*.

Other Monitoring and Administrative Commands in the BDNA Shell

The BDNA Shell contains several commands useful for monitoring and administration. Two examples of these commands are refreshanalytics and refreshpub.

Refreshanalytics

refreshanalytics: Refresh Analytic reports in the FactBase or Collection Store.

refreshanalytics is used to build or refresh the Analytics reports. For a Collection Store refreshanalytics will build or refresh all Analytics reports.When refreshanalytics is used with a FactBase it will refresh either a single inventory or all available inventories (-a). refreshanalytics builds only the reports that need to be built. For example, if refreshanalytics is run once, then the user creates custom reports (without building them); if the user runs refreshanalytics again, only the new reports are built.

Reports can only be built when a snapshot exists. For ease of use, the refreshanalytics command automatically builds the snapshot if necessary. Snapshot refresh is intelligent and incremental. If no snapshot exists one will be built. In a FactBase, if a snapshot exists but collections have been added or removed since the last snapshot, the snapshot will be incrementally built as necessary. In extreme cases, the user may want to override the snapshot and report refresh logic. This can be done by specifying the -f argument, which forces the snapshot and reports to be rebuilt from scratch. Because refreshanalytics is a time-consuming operation, it should be used with discretion.

USAGE: `refreshanalytics [ -f ] [ -y ] < -a | inventoryName >`

INPUTS:

-f (optional): Force the snapshot and reports to be rebuilt entirely from scratch.

-y (optional): Refresh Analytics reports without asking for confirmation.

-a: Refresh Analytics reports for all inventories in a factbase.

inventoryName: Refresh Analytics reports for a specified inventory in a factbase.

EXAMPLES:

FactBase only:

```
refreshanalytics QOneInventory
refreshanalytics -f -y "Q1 Inventory"
refreshanalytics -y -a
```

Collection Store only:

```
refreshanalytics -y
```

refreshpub

refreshpub: Publish discovery data to the BDNA Discover Publisher schema.

refreshpub is used to publish discovery data to the Publisher schema. Once data for an inventory is in the Publisher schema, it may be overwritten (refreshed) by running refreshpub after changes (such as adding or removing a collection in a FactBase or scanning additional Networks in a Collection Store) have been made.

The BDNA Discover Publisher schema can only be built when a snapshot exists. For ease of use, the refreshpub command automatically builds the snapshot if necessary. Snapshot refresh is intelligent and incremental. If no snapshot exists, it will be built. In a FactBase, if a snapshot exists but collections have been added or removed since the last snapshot, the snapshot will be incrementally built as necessary. In extreme cases, the user may want to override the snapshot and Publisher refresh logic. This can be done by specifying the -f argument, which forces the snapshot and Publisher schema to be rebuilt from scratch.

Because refreshpub is a time-consuming operation, it should be used with discretion.

USAGE: `refreshpub [ -f ]  [ -y ] < -a | inventoryName>`

INPUTS:

-f (optional): Force the snapshot and Publisher schema to be rebuilt

entirely from scratch.

-y (optional): Refresh Publisher schema without asking for confirmation.

-a: Refresh Publisher schema for all inventories in the FactBase.

inventoryName: Refresh Publisher schema for a specified FactBase inventory.

EXAMPLES:

FactBase only:

```
refreshpub QOneInventory
refreshpub -f -y "Q1 Inventory"
refreshpub -y -a
```

Collection Store only:

```
refreshpub -y
```

For more information on BDNA Shell commands, refer to "*BDNA Shell Command Reference*."

Key Log Files

The best way to monitor a BDNA Discover scan is through the Scan Monitoring reports and the Event Viewer, found under Scan Administration. However, viewing and analyzing BDNA Discover's log files can be useful for detailed, in-depth troubleshooting. BDNA Discover's UNIX component logs are stored in `$BDNA_HOME/logs/` on each machine running components. BDNA Discover's Windows components logs are stored in `%PROGRAMFILES%/BDNA/WinCS/logs/`. The logs are automatically rotated by appending numerical extensions to the log filenames. For example, `mbus.log` and `mbus.log.1` are both logs for the MBUS. `mbus.log.1` will contain older entries than `mbus.log`.

The following section describes the two key log files for the various BDNA Discover components and their relevance for administering and monitoring tasks.

cle.CLE*.log

The `cle.CLE*.log` contains logged information associated with the BDNA Discover Collection Engine components. Stored details include requests made to the associated Collection Manager, status messages containing memory utilization and component workload, and raw data results from discovery tasks.

The log provides a task-by-task and device-by-device account of the discovery operation. Data returned by PerlCS and WinCS is logged in the CLE logs.

dw.log

The `dw.log` contains logged information associated with building the BDNA Discover data warehouse (snapshot discovery data) and report refresh, including database details for building base tables and report views. The log can be monitored during the post-discovery procedure of building and possibly exporting BDNA Discover reports.

Maintenance Tasks

The Maintenance Tasks tool within the Scan Administration application contains several useful functions that are designed to maximize the effectiveness of your scans:

- Credential Coverage Test

  Performs a level 1 scan and a minimal level 2 scan. After running this test, check the scan results to see where credentials succeeded or failed. Saves time by enabling credentials verification before a complete Level 2 scan begins. Use this information to change credentials before the scan, if needed.

- Update SSH Credential Task

  Used to streamline security processes when changing the public/private key pair used by BDNA Discover to access UNIX servers using SSH. Updates SSH credentials (public key) on target machines by connecting to each machine and pushing out the new public key.

- UNIX Level 2 Dedupping Task

  Performs a minimal level 2 scan to obtain the information necessary for dedupping. When the actual scan is run following this test, a target machine with multiple IP addresses will be treated as a single target machine. This saves time and resources by avoiding unnecessary duplication of level 2 scanning for target machines with multiple IP addresses.

- BDNA Discover Namespace Dedupping

  Finds discovered operating systems in the BDNA Discover repository and represents them as a single entity. By default, dedupping is automatically run by BDNA Discover on a regular basis. Thus, new dedupping tasks should only be created under special circumstances, when advised by BDNA Discover support.

- Cleanup BDNA Discover UNIX Processes

  Special level 2 task that logs on to target machines and kills any remaining BDNA Discover processes. May be run after a level 2 scan. Frees up resources by cleaning up UNIX processes started by BDNA Discover that are no longer necessary.

- Remove Windows Profiles

  Removes the profile directory from specified Windows machines. This maintenance task should only be run after all scanning activity has been completed. Using the provided credentials, this task will remove the profile directory from the targeted Windows machines regardless if BDNA Discover created it or not.

Caution: Executing this task will result in the deletion of the entire profile directory, including all files and folders, on every machine specified. The deletion cannot be undone once completed. Do not proceed with this task unless you are absolutely certain it is safe to deleted the profiles specified by your selected credentials.

Profiles that have open files at the time of the attempted deletion or profiles that the Windows operating system is unable to unload cleanly after the user last logged out may not be deleted or may be only partially deleted.

- Clear BDNA Discover UNIX Cache

  Removes files created during level 2 discovery of UNIX machines. These files typically use minimal storage space. They act as a cache to improve level 2 performance as well as to facilitate asset tracking. Remove these files only after all scan activity has completed. Only UNIX machines where valid credentials are specified will have their cache cleared.

Caution: This task will delete all cache files generated by BDNA Discover UNIX discovery on selected machines. This procedure cannot be undone. If level 2 discovery is performed against these machines in the future, performance and tracking may be affected since the cache files will be recreated.

Finishing a Scan and Generating Reports 12

About this Chapter

After discovery completion, the BDNA Discover Administrator must prepare the system for users to view and analyze the collected data. This chapter reviews the steps that the BDNA Discover Administrator performs to accomplish this, including performing a discovery snapshot and rebuilding the baseline Analytics reports. It also includes a discussion of command-line functions for exporting the data to flat files for use in other systems.

An alternate workflow after the scan is finished is to export the collection and import it to a FactBase. For instructions on doing so, refer to the *BDNA Discover FactBase User Guide*. Nevertheless, a valid use case still exists for refreshing reports in the Collection Store, because it enables users to verify the results before importing the collection to FactBase. By refreshing the reports in a Collection Store, the user can look at the results in the Analytics application and see if they make sense before importing them to FactBase. Refer to Chapter 7, "Troubleshooting Discovery" for more information.

Performing a discovery snapshot and rebuilding the reports are accomplished using the Application Administration component in BDNA Discover.

Note: For information on how to launch BDNA Discover, refer to "Launching the BDNA Discover Client" on page 35.

Snapshot Discovery Data

The first task of post-discovery administration is creating a snapshot of the discovered data. Since an inventory cannot be performed instantaneously, snapshots are used to provide a point-in-time view of all data collected prior to the snapshot. If the BDNA Discover Administrator does not create a snapshot, users will be viewing data from the previous snapshot, even if data has been collected since then. If no snapshot has ever been taken, users will see no data.

To create a snapshot:

1. Log in to BDNA Discover and open Application Administration.
2. In the left navigation pane, under the Data Preparation heading, click Snapshot Discovery Data.

   The system displays a message stating that this could be a long-running operation.
3. Click Yes to confirm the operation.

   Upon successful completion of the snapshot, BDNA Discover will have summarized all discovered collected data up to the point of the snapshot.

Building Analytics Reports

The next task of post-discovery administration is to rebuild the Analytics reports. These reports comprise the baseline reports furnished by BDNA Discover to end users. These reports can be modified and subsequently saved by the user.

To build Analytics reports:

1. Log in to BDNA Discover and open Application Administration.
2. To generate the reports, click Refresh Reports under Application Administration > Analytics.

   The system displays a message stating that this could be a long-running operation and providing a snapshot discovery data option.
3. Optional: Select the checkbox if snapshot discovery data has never been run or if additional scanning has been done since the last snapshot.
4. Click Refresh.

   Once the operation completes, the application will report success.
5. To export the data, click Refresh and Export Reports.

   This will generate the reports and then export them to flat files.

   You will be given several options for export, including file type, filename and the inclusion of large column data. (Large column data includes dumps of data from BDNA Discover such as lists of installed packages and other large blocks of data. Many uses for export files do not require this level of detail, so it may often be excluded. Excluding these columns will decrease the time required to execute the export.)

   Reports that have been created by users and saved under the My Report folder will be exported as well.

BDNA Discover Utility Reference 13

About this Chapter

This chapter provides information about the Analytics application, which ships with three Microsoft Windows utilities that can help the BDNA Discover Administrator prepare for BDNA Discover system deployment. These utilities include:

- Installing and Running the BDNA Discover Credential Checker
- BDNA Discover Network Range Comparison Tool
- BDNA Discover Bulk Import Validation Tool

Note: Bulk Import is also available from the command line. For further information, refer to "Working with Bulk Load Files" on page 215.

Installing and Running the BDNA Discover Credential Checker

The BDNA Discover Credential Checker enables a BDNA Discover Administrator to validate a credential against a set of IP addresses, IP ranges, or subnets.

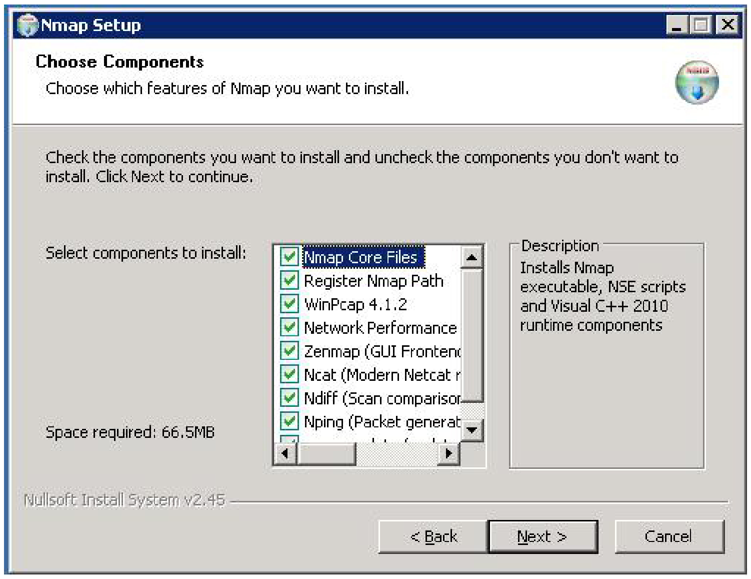
To install the BDNA Discover Credential Checker:

1. Install the Microsoft .NET Framework 2.0 SP1 Runtime, which may be downloaded from:

   `http://www.microsoft.com/en-us/download/details.aspx?id=16614`
2. Install NMAP 6.0 or higher, which can be downloaded from:

   `http://nmap.org/dist/nmap-6.00-setup.exe`
3. During NMAP installation, make sure to select all options.

Figure 55: Nmap Setup



Nmap Setup

Choose Components

Choose which features of Nmap you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ✓ Nmap Core Files
- ✓ Register Nmap Path
- ✓ WinPcap 4.1.2
- ✓ Network Performance
- ✓ Zenmap (GUI Frontenc
- ✓ Ncat (Modern Netcat r
- ✓ Ndiff (Scan compariso
- ✓ Nping (Packet generat

Description
Installs Nmap executable, NSE scripts and Visual C++ 2010 runtime components

Space required: 66.5MB

Nullsoft Install System v2.45

4. Copy the following file onto the Windows system where the BDNA Discover Credential Checker will be running:

   `$BDNA_HOME/pso/scripts/Windows/BDNA_Windows_Tools_`x.x.x_####`.zip`

   where:

   x.x.x is the product version number

   #### is the product build number

5. Unpack the following file:

   `BDNA_Windows_Tools_x.x.x_####.zip`

   where:

   x.x.x is the product version number

   #### is the product build number

6. Run `CredentialCheckerTool.msi`.
7. Continue to click “Next” during the installation until it finishes.

Launching BDNA Discover Credential Checker

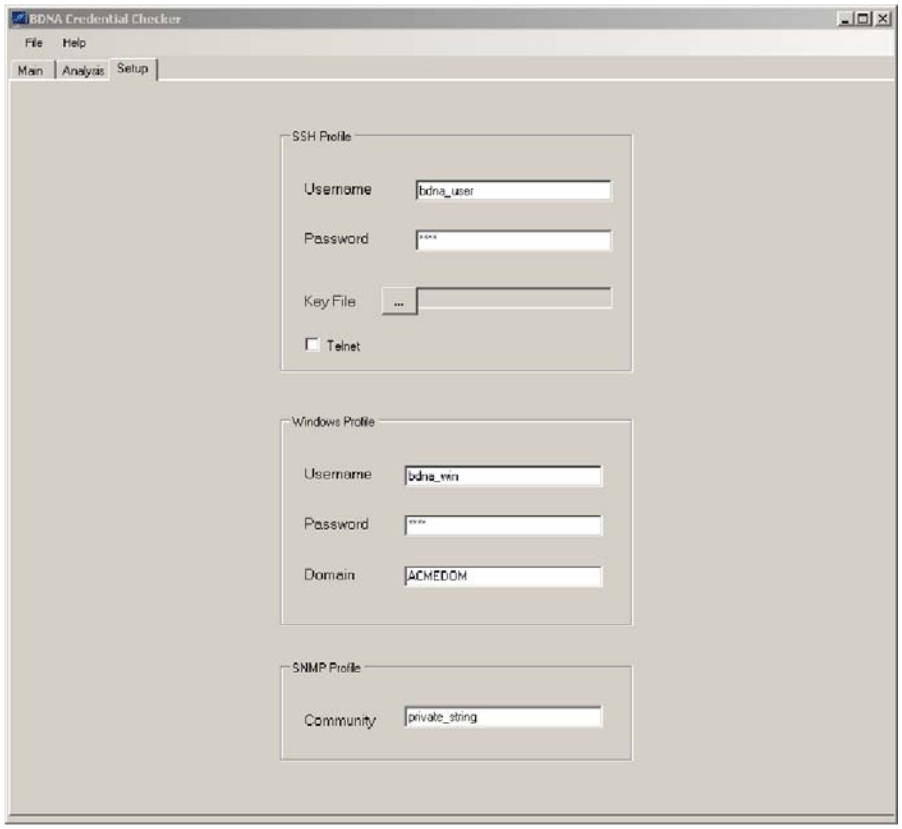
Launch BDNA Discover Credential Checker by clicking “Start > Programs > BDNA Tools > BDNA Credential Checker”.

Using BDNA Discover Credential Checker

To use the BDNA Discover Credential Checker:

1. Click the "Setup" tab and enter the credentials to be tested.

Figure 56: BDNA Discover Credential Checker Setup Screen



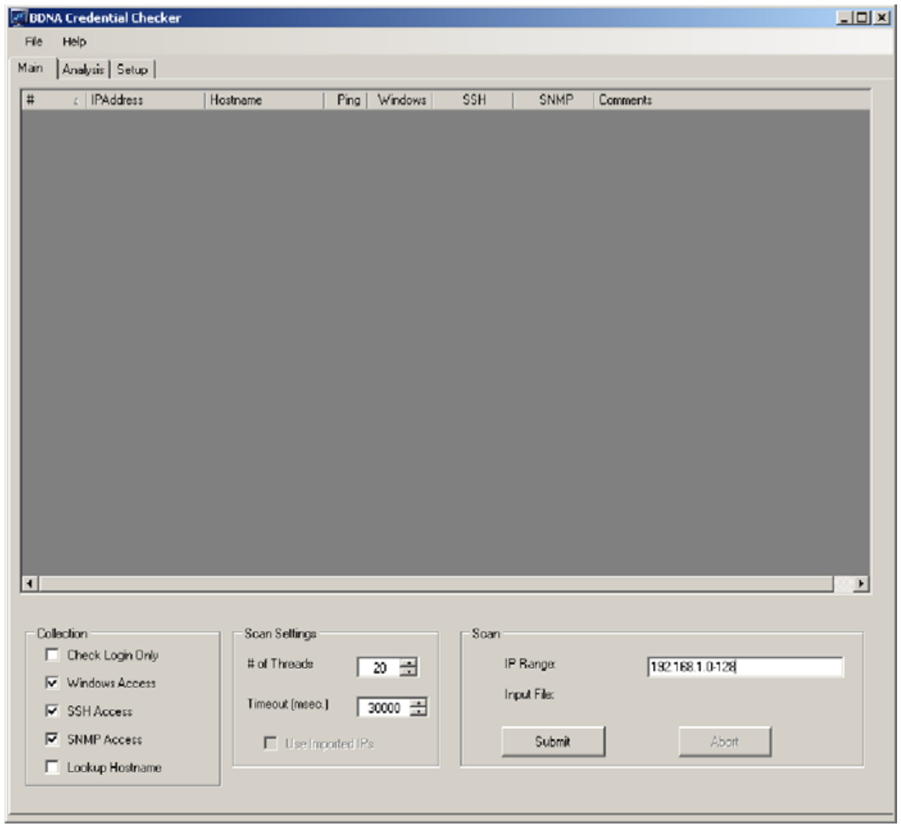
Username bdna_user
Password ****
Telnet
Username bdna_win
Password ****
Domain ACMEDOM
Community private_string

2. Click the "Main" tab and select the checkbox(es) for the credentials to be tested.
3. Under "Collection Settings", select "Check Login Only" to verify login credentials without verifying BDNA Discover-specific requirements.

   This will increase the scan speed, but will not validate for the required permissions of the credentials.
4. Select "Lookup hostname" to identify the hostname of the machine.

   This will decrease the overall scan performance.
5. Under "Scan", enter the IP Range (for example, 192.168.1.0-128).

Figure 57: BDNA Discover Credential Checker Main Settings



☐ Check Login Only
☑ Windows Access
☑ SSH Access
☑ SNMP Access
☐ Lookup Hostname

# of Threads: 20
Timeout (msec.): 30000
☐ Use Imported IPs

IP Range: 192.168.1.0-128
Input File:

6. Click “Submit”.
7. Click the “Analysis” tab to analyze the success/failure ratio of the tested credential(s).
8. To save the results to a file, select “File > Export > Credential Results”.
9. Select a filename for the export.

   The exported file will be in CSV format.

Importing

Use the importing feature to test multiple IP Ranges.

To import a file:

1. Select “File > Import > IP List”.
2. Find the file containing the list of IPs to be imported.
3. Click the “Use Imported IPs” checkbox under Scan Settings.

   Refer to “Using BDNA Discover Credential Checker” on page 181, step 6 through step 9, for information on how to perform a credential test.

Windows Error Messages

The most common errors that are reported by BDNA Discover Credential Checker on Windows credential validation are presented in Table 10 on page 183.

Table 10: Common Credential Checker on Windows Errors

Message	Description
Timeout/cscript timeout	Net use/WMI cscript command did not return a response before timeout.
The network connection could not be found	Most likely the machine is no longer at that IP address.
-2147417848: The object invoked has disconnected from its clients	Most likely the machine is no longer at that IP address.
System xxx error has occurred	Net use command failed because of a connectivity issue: 53: The network path was not found. Most likely a non-Windows machine. 67: The network name cannot be found. Most likely a non-Windows machine. Level 2 discovery may be possible. 1240: The account is not authorized to log in from this station. 1311: There are currently no logon servers available to service the logon request. 1368: Login/password valid. Unable to impersonate using a named pipe until data has been read from that pipe. Partial Level 2 discovery may be possible. 1385: Login/password valid. Logon failure: the user has not been granted the requested logon type at this computer. 1789: The trust relationship between this workstation and the primary domain failed. 1792: An attempt was made to logon, but the network logon service was not started. 2240: Login/password valid. The user is not allowed to log on from this workstation.
Bad Login/Password	The supplied credentials are not valid for the remote machine.
No DCOM connect	Login/password valid. No DCOM connect possible to remote machine. No Level 2 discovery possible.
No DCOM access	Login/password valid. DCOM access allowed. WMI protocol exists but there are no providers for the expected WMI namespaces. Likely a Windows NT machine with partial WMI software installed.
429:Class not registered	Login/password valid. DCOM access allowed. WMI protocol exists but there are no providers for the expected WMI namespaces. Likely a Windows NT machine with partial WMI software installed.
No WMI access to cimv2	Login/password valid. DCOM access allowed. WMI permissions have not been granted to user for root/default namespace. No Level 2 discovery possible.
No WMI access to default	Login/password valid. DCOM access allowed. WMI permissions have not been granted to user for root/default namespace. No Level 2 discovery possible.

Table 10: Common Credential Checker on Windows Errors (Continued)

Message	Description
No registry access	Login/password valid. DCOM access allowed. WMI permissions have been granted to user. Cannot access system registry. Partial Level 2 discovery possible.
No reg access to Uninstall	Login/password valid. DCOM access allowed. WMI permissions have been granted to user. No registry read access for the Uninstall path. Partial Level 2 discovery possible.
No reg access to Software	Login/password valid. DCOM access allowed. WMI permissions have been granted to user. No registry read access for the Software path. Partial Level 2 discovery possible.

SSH/Telnet Error Messages

The most common errors that are reported by BDNA Discover Credential Checker on SSH/Telnet credential validation are listed in Table 11 on page 184.

Table 11: Common Credential Checker on SSH/Telnet Errors

Message	Description
Timeout	SSH did not return a response before timeout.
Automatic telnet login	Automatic Telnet server that does not prompt for a login.
Connection refused Connection timed out	Most likely no SSH host on this machine.
Server unexpectedly closed network connection	Configuration issue with SSH host.
Bad Login/Login incorrect	The supplied credentials are not valid for the remote machine, or no users are allowed to login to machine (because /etc/nologin is set).
This account is currently not available	Login/password valid. Account is disabled (shell set to `/sbin/nologin`).
No shell allowed	Login/password valid. Account has no shell privileges (shell set to `/bin/false`).
Cannot access directory	Login/password valid. Account cannot view or create `/tmp/bdnaWorkingDirectory`.

Advanced Usage

Under “Scan Settings”, “# of Threads” can be adjusted according to the robustness of the system on which the Credential Checker is running. For a faster system (Pentium4 2GHz and 512MB RAM), 20-25 threads per CPU is a good number. For a slower system, 10-15 threads per CPU is a better number.

“Timeout (msec.)” is the maximum amount of time given for each IP address before Credential Checker gives up on a response. The timeout parameter is valid for SSH/Telnet, SNMP, and Login only Windows checks. For a faster scan this number can be lowered, but some machines may not be detected. For a particularly high-latency network, this number can be increased to ensure scan accuracy.

Known Limitations

If doing a Telnet scan, non-standard Telnet login prompts or non-standard automatic logins will cause threads to hang indefinitely. If this occurs, delete the `plink.exe` threads manually. `plink.exe` is a command line utility that is used by the Credential Checker to establish a secure or non-secure shell session.

The application has been tested with a maximum of 5000 active IPs. Care should be taken when exceeding this value.

BDNA Discover Network Range Comparison Tool

The BDNA Discover Network Range Comparison Tool allows a BDNA Discover Administrator to analyze two sets of IP addresses, IP ranges, or subnets and identify the unique IP addresses and the overlapping IP addresses between the two sets.

Installing BDNA Discover Network Range Comparison Tool

To install the BDNA Discover Network Range Comparison Tool:

1. Install the Microsoft .NET Framework 2.0 Runtime, which may be downloaded from:

   `www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en`

2. Copy the following file onto the Windows system where the BDNA Discover Network Range Comparison Tool will be running:

   `$BDNA_HOME/pso/scripts/Windows/BDNA_Windows_Tools_x.x.x_####.zip`

   where:

   x.x.x is the product version number

   #### is the product build number

3. Unpack the following file:

   `BDNA_Windows_Tools_x.x.x_####.zip`

   where:

   x.x.x is the product version number

   #### is the product build number

4. Run `OverlapToolInstall.msi`.
5. Continue to click Next during the installation until it finishes.

Launching BDNA Discover Network Range Comparison Tool

Launch BDNA Discover Network Range Comparison Tool by clicking “Start > Programs > BDNA Tools > BDNA Network Range Comparison Tool”.

Preparing Input Files

Before using the BDNA Discover Network Range Comparison Tool, the two text files that will be used to compute the overlaps must be prepared. The two text files must be defined with at least one column, which can be an IP addresses, IP ranges, or a subnet. A subnet mask can optionally be created in the second column.

Sample input file 1:

```
192.168.1.0/24
192.168.2.0/24
192.168.3.0/24
192.168.4.0/24
192.168.5.0/24
```

Sample input file 2:

```
192.168.1.14-192.168.1.32
192.168.2.5
192.168.4.56
```

Defining Input File Format

Once both input files are created, define an input file format. The following is a step-by-step procedure to define an input file format:

1. Select “Edit > Define Input File Format”.
2. Enter the file format for both input files.

   Both files must use the same format.
3. Select the file type.

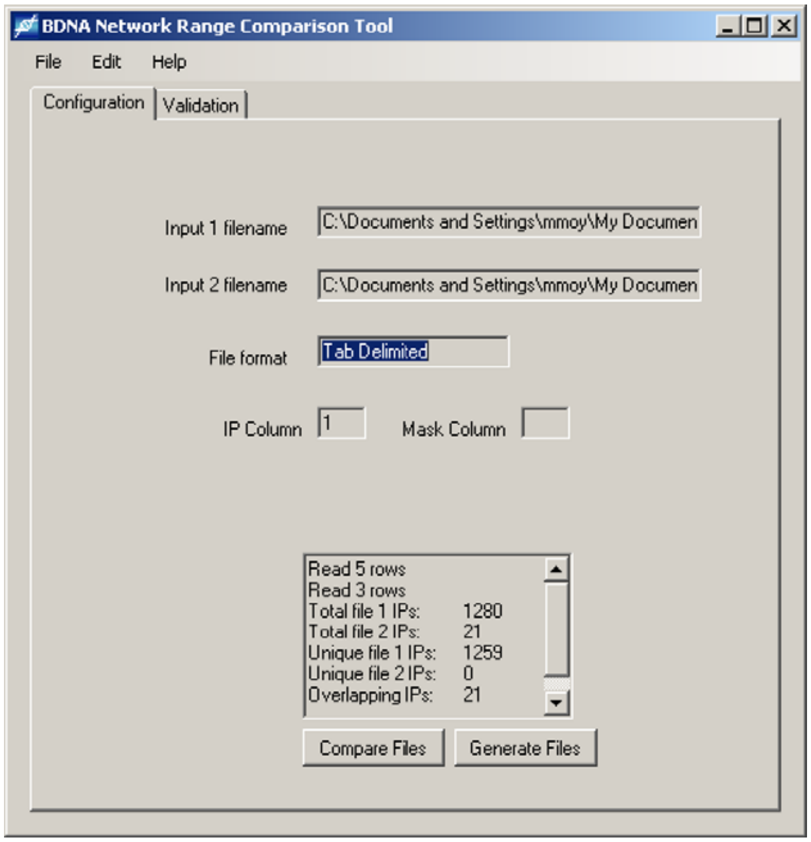
   Select either “Excel” or “Delimited Record”. For Excel files, specify the worksheet to read data from. For delimited files, specify the delimiter.
4. Click “Next”.
5. In the “IP Range column” field, select the column number where the IP range can be found.
6. If the subnet mask is in a separate column, specify the column number in the Mask column; otherwise, set the Mask column field to 0.
7. Click “Finish”.

Analyzing Input Files

To analyze the input files:

1. Select “File > Import File 1”.
2. Select the first file containing the network ranges to compare.
3. Select “File > Import File 2”.
4. Select the second file containing the network ranges to compare.
5. Click “Compare Files”.

Figure 58: BDNA Discover Network Range Comparison Tool



Input 1 filename: C:\Documents and Settings\mmoy\My Documen

Input 2 filename: C:\Documents and Settings\mmoy\My Documen

File format: Tab Delimited

IP Column: 1

Mask Column:

```
Read 5 rows
Read 3 rows
Total file 1 IPs:     1280
Total file 2 IPs:     21
Unique file 1 IPs:    1259
Unique file 2 IPs:    0
Overlapping IPs:      21
```

After the comparison completes, use the scrollbar to review the information in the text box at the bottom of the Configuration window. This shows the number of rows contained in each file, the total number of IPs in each file, and the number of unique IPs and overlapping IPs found.

6. Click the “Validation” tab to view the results.
7. Click “Generate Files” and specify a ZIP filename.

   The ZIP file will contain the unique network ranges of file 1, the unique network ranges of file 2, and the intersection of network ranges of both files.

Known Limitations

When working with large input files, for example more than 1,000 rows, the performance of the Network Range Comparison Tool may be slow.

BDNA Discover Bulk Import Validation Tool

The BDNA Discover Bulk Import Validation Tool allows a BDNA Discover Administrator to generate a set of bulk load files that can be imported into the BDNA Discover system.

Note: Bulk Import is also available from the command line. For further information, refer to “Working with Bulk Load Files” on page 215.

Installing BDNA Discover Bulk Import Validation Tool

To install the BDNA Discover Bulk Import Validation Tool:

1. Install the Microsoft .NET Framework 2.0 Runtime, which you can download from this web page:

   `www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en`

2. Copy the following file onto the Windows system where the BDNA Discover Bulk Import Validation Tool will be running:

   `$BDNA_HOME/pso/scripts/BDNA_Windows_Tools_x.x.x_####.zip`

   where:

   x.x.x is the product version number

   #### is the product build number

3. Unpack the following file:

   `BDNA_Windows_Tools_x.x.x_####.zip`

   where:

   x.x.x is the product version number

   #### is the product build number

4. Run `BulkImportValidationToolInstall.msi`.
5. Continue to click “Next” during the installation until it finishes.

Launching BDNA Discover Bulk Import Validation Tool

Launch the BDNA Discover Bulk Import Validation Tool by clicking “Start > Programs > BDNA Tools > BDNA Bulk Import Validation Tool”.

Preparing Input Files

Before using the BDNA Discover Bulk Import Validation Tool, a text file that can be used to generate bulk load files must be prepared. An input file contains several columns, depending on the data type. BDNA Discover Bulk Import Validation Tool supports four different data types:

- Network/Logical Sets (groups)
- Exclusions
- Credentials
- SSH Key Credentials

Table 12: Sample BDNA Discover Bulk Import Validation Tool Input File

IP Range	Network Name	Division	Region	Project
192.168.1.0/24	SF_192_168_1	Sales	San Francisco	Project_Test1
192.168.23.0/24	LA_192_168_23	Marketing	Los Angeles	Project_Test1
192.168.2.0/24	SF_192_168_2	Engineering	San Francisco	Project_Test1

Creating a Template

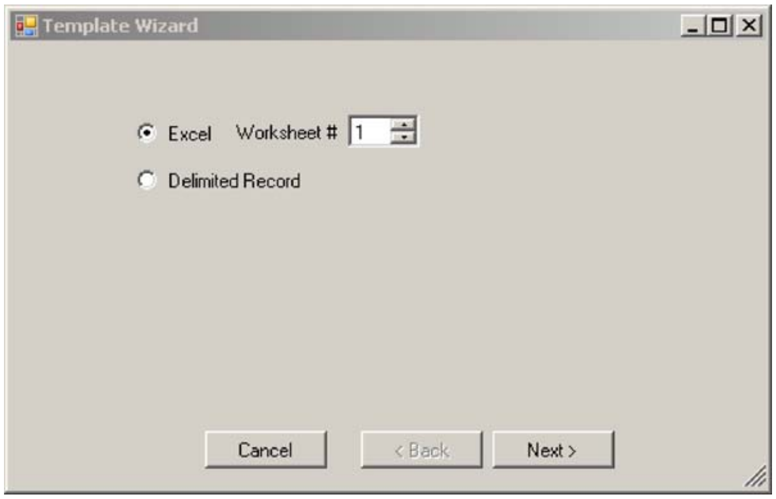
Once the input file is created, create a new template.

To create a new template:

1. Select “Edit > Create/Modify Template”.
2. Select file type.

   Select either “Excel” or “Delimited Record”. For Excel files, specify the worksheet to read data from. For delimited files, specify the delimiter.
3. Click “Next”.

Figure 59: Select File Type for BDNA Discover Bulk Import Validation Tool



4. In the Data Type drop-down, select the data type of the input file.

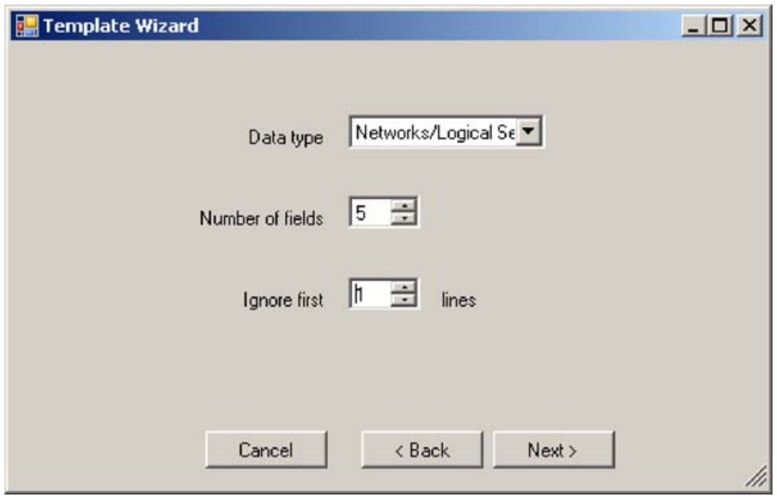
5. In the “Number of fields” field, enter the number of columns to use from the input file.

   Usually this value should match the total number of columns defined in the input file.

6. In the “Ignore first” field, enter 0 if the input file does not contain a column header in the first row; enter 1 if the input file contains a column header in the first row.

7. Click “Next”.

Figure 60: Select Data Type, Number of Fields, and Ignore Headers

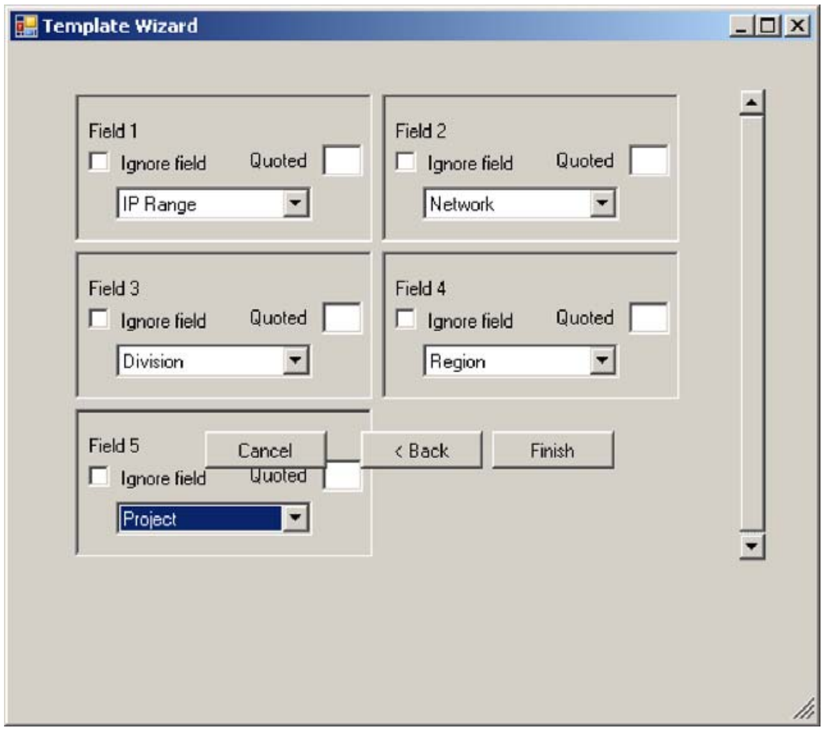


Data type: Networks/Logical Se
Number of fields: 5
Ignore first: 1 lines

8. Match the column type against each column defined in the input file.

   For example, if the first column defined in the input file is an IP range, then select IP Range from the drop down.

Figure 61: Defining Columns for BDNA Discover Bulk Import Validation Tool



9. Click “Finish”.

10. To save the template for future use, select “File > Save Template”.

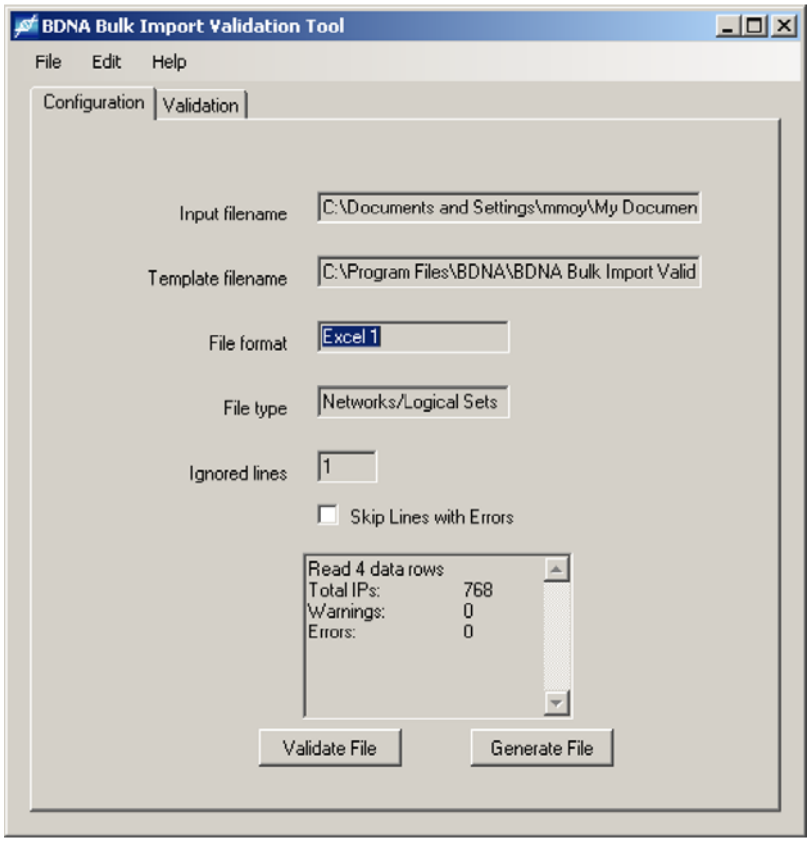
Validating Input Files

To validate the input file and generate bulk load files:

1. Select “File > Import File” to load the input file.

2. Click “Validate File” button.

Figure 62: Validating an Input File

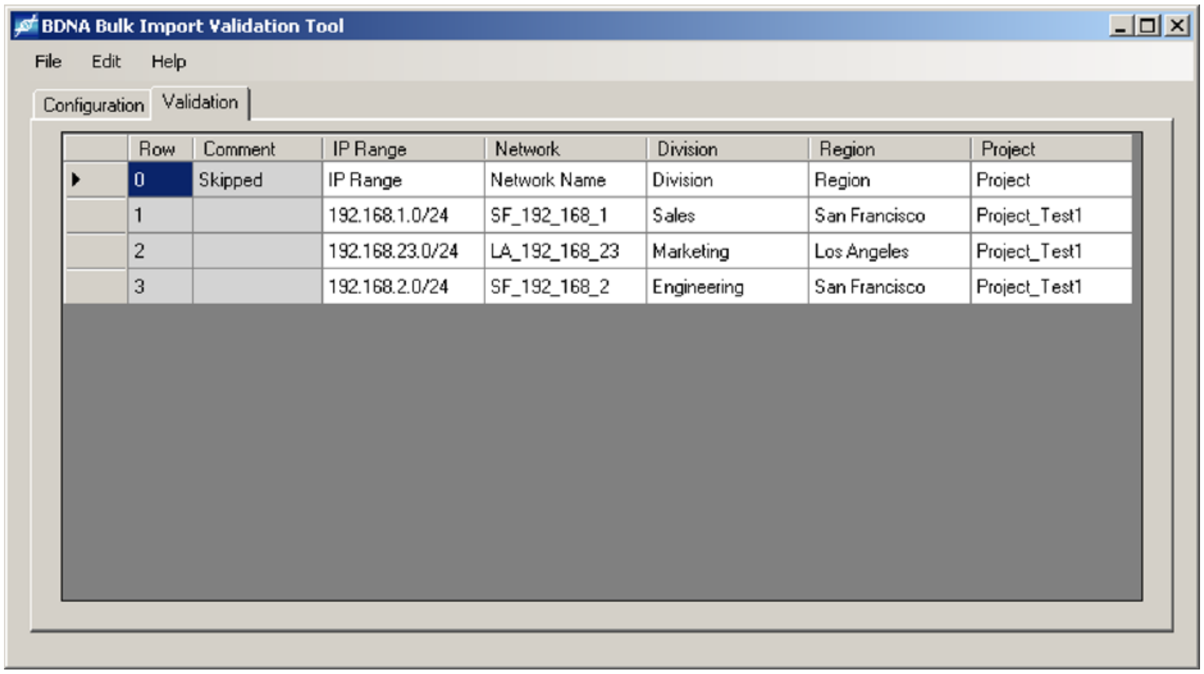


Input filename: C:\Documents and Settings\mmoy\My Documen
Template filename: C:\Program Files\BDNA\BDNA Bulk Import Valid
File format: Excel 1
File type: Networks/Logical Sets
Ignored lines: 1
[ ] Skip Lines with Errors

```
Read 4 data rows
Total IPs:      768
Warnings:       0
Errors:         0
```

3. Click the “Validation” tab to view the results, if necessary.

Figure 63: Viewing Results of a Bulk Import



Row	Comment	IP Range	Network	Division	Region	Project
0	Skipped	IP Range	Network Name	Division	Region	Project
1		192.168.1.0/24	SF_192_168_1	Sales	San Francisco	Project_Test1
2		192.168.23.0/24	LA_192_168_23	Marketing	Los Angeles	Project_Test1
3		192.168.2.0/24	SF_192_168_2	Engineering	San Francisco	Project_Test1

4. Click “Generate File” and specify a ZIP filename.

The ZIP file will contain the bulk load files that can be imported into the BDNA Discover system.

References

Networks/Logical Sets

Networks define the IP ranges that should be scanned by the Analytics application. Logical sets are used to segregate groups of IP ranges into distinct categories. The following bulk import files are generated from this data type:

- Network Definitions
- Logical Sets: One for each logical set field defined in the input file.

  The logical sets are associated with the IP Range field.

The fields that can be input to generate network and logical set bulk import files are presented in Table 13 on page 194.

Table 13: Fields for Network and Logical Set Bulk Import Files

Field	Validation	Definition
Network	No blank values	The name of the network associated with an IP range. If the field is not specified, the value will default to ‘Default Network’.
IP Range	Required field Must use the format: XXX.XXX.XXX.XXX or XXX.XXX.XXX.XXX/BB or XXX.XXX.XXX.XXX-YYY.YYY.YYY.YYY XXX can be any value between 0–255 inclusive. BB can be any value between 16–32 inclusive. YYY.YYY.YYY.YYY >= XXX.XXX.XXX.XXX Whitespaces will be ignored.	The IP range (or network IP address) to include for scanning.
Mask	Must use the format: BB /BB BB can be any value between 16 and 32 inclusive. If this field is specified, IP Range must use the format: XXX.XXX.XXX.XXX Whitespaces will be ignored.	The bitmask to associate with a network IP address, defined by the IP Range field.
Department	None	A logical set type
Division		
Workgroup		
Region		
Project		
Building		
Data Center		
Production		
Staging		
Testing		

Exclusions

Exclusions define the IP ranges that should not be scanned by the Analytics application. The bulk import exclusion file created by the tool will belong to the namespace 'Global Namespace'.

The fields that can be input to generate an exclusion bulk import file are listed in Table 14 on page 195.

Table 14: Fields for Network and Logical Set Exclusion Bulk Import Files

Field	Validation	Definition
IP Range	Required field Must use the format: XXX.XXX.XXX.XXX or XXX.XXX.XXX.XXX/BB or XXX.XXX.XXX.XXX-YYY.YYY.YYY.YYY XXX can be any value between 0-255 inclusive. BB can be any value between 16-32 inclusive. YYY.YYY.YYY.YYY >= XXX.XXX.XXX.XXX Whitespaces will be ignored.	The IP range (or network IP address) to exclude from scanning
Mask	Must use the format: BB /BB BB can be any value between 16 and 32 inclusive. If this field is specified, the IP Range must use the format: XXX.XXX.XXX.XXX Whitespaces will be ignored.	The bitmask to associate with the network IP address defined by the IP Range field

Credentials

Credentials define the login information required by the Analytics application for Level 2 access to a target machine. The bulk import credential file created by the tool will associate a single credential to all logical sets and networks specified in the same input file row and all rows below the credential up to the next defined credential or end of file. If no logical set or network field is specified, the credentials will be associated with "network_Default_Network".

The fields that can be input to generate a credential bulk import file are presented in Table 15 on page 196.

Table 15: Fields for Network and Logical Set Credential Bulk Import Files

Field	Validation	Definition
Username	Required field	
Password	If any of the fields Username, Password or Type is non-blank in a row, all of these fields must be non-blank.	The password of the account used to access the target machines.
Type	Required field If any of the fields Username, Password or Type is non-blank in a row, all of these fields must be non-blank. Must be one of these values (non-case sensitive): SSH Telnet Rsh Windows	The credential type. The value maps to a BDNA Discover type: SSH: `perl.shellConnection.ssh` Telnet: `perl.shellConnection.telnet` Rsh: `perl.shellConnection.rsh` Windows: `windows.wincs.defaultwincs`
Network	There must be at least one network or logical set defined for each credential row if any network or logical set field is specified.	The network against which the credential should be applied.
Department		
Division		
Workgroup		
Region		
Project		

SNMP Credentials

SNMP Credentials define the login information required by the Analytics application for Level 2 access to a target machine using SNMP. The bulk import credential file created by the tool will associate a single credential to all logical sets and networks specified in the same input file row and all rows below the credential up to the next defined credential or end of file. If no logical set or network field is specified, the credentials will be associated with 'network_Default_Network'. If no port is specified, default is 161.

The fields that can be input to generate a credential bulk import file are listed in Table 16 on page 197.

Table 16: Fields for SMTP Credential Bulk Import Files

Field	Validation	Definition
Community String	Required field No blanks allowed	The SNMP community string used to access the target machines.
Port	Port is a numeric integer value greater than 0.	The SNMP port used to access the target machines. Port 161 is the default SNMP port.
Network	There must be at least one network or logical set defined for each credential row if any network or logical set field is specified.	The network against which the credential should be applied.
Department		
Division		
Workgroup		
Region		
Project		

SSH Key Credentials

SSH Key credentials define the login information required by the Analytics application for Level 2 SSH access to a target machine using Public/Private Keys. The bulk import credential file created by the tool will associate a single credential to all logical sets and networks specified in the same input file row and all rows below the credential up to the next defined credential or end of file. If no logical set or network field is specified, the credentials will be associated with ‘network_Default_Network’.

The fields that can be input to generate a credential bulk import file are presented in Table 17 on page 198.

Table 17: Fields for SSH Key Credential Bulk Import Files

Field	Validation	Definition
Username	Required field If any of the fields Username, Key (the first row) or Type is non-blank in a row, all of these fields must be non-blank.	The username of the account used to access the target machines.
Password	None	The password of the account used to access the target machines.
Key	Required field If any of the fields Username, Key (the first row) or Type is non-blank in a row, all of these fields must be non-blank. Key must start with -----BEGIN DSA PRIVATE KEY----- or -----BEGIN RSA PRIVATE KEY----- Key must end with -----END DSA PRIVATE KEY----- or -----END RSA PRIVATE KEY----- Key must have 64 characters on each line, except for the last line.	The SSH Key of the account used to access the target machines.
Type	Required field If any of the fields Username, Key (the first row) or Type is non-blank in a row, all of these fields must be non-blank. Must be of value (non-case sensitive): Key GenerationalKey	The credential type. The value maps to a BDNA Discover type: Key: `sshWithKey` GenerationalKey: `sshWithKey.sshWithGenerationalOpenSSHKey`
Network	There must be at least one network or logical set defined for each credential row if any network or logical set field is specified.	The network against which the credential should be applied.
Department		
Division		
Workgroup		
Region		
Project		

Working with Collection Stores 14

About this Chapter

Prior to starting a scan with BDNA Discover, you must initialize the Collection Store. During the initialization process, the database tables that support BDNA Discover are created and the content is loaded into the database. This chapter provides detailed information about the initialization process.

Starting the Application Agent

Before you execute the command to initialize the BDNA Discover Oracle user schema, you must first start the BDNA Discover Application Agent. The Application Agent is responsible for managing BDNA Discover components on the Linux machine where the agent is running. When the Application Agent starts, it attempts to communicate with the BDNA Discover Message Bus. The BDNA Discover Message Bus provides a communication mechanism for all BDNA Discover components, and it must be started prior to performing any other operations. If the agent does not find a Message Bus when starting, it will start one. Therefore, in a single node installation, starting the Application Agent also starts the Message Bus.

To start the Application Agent, log in as the BDNA Discover user on the system where BDNA Discover is installed and issue the following command:

```
% sh startagent.sh &
```

Caution: Be sure to start the Application Agent as a background task by using the & modifier; otherwise, when the BDNA Discover user exits, the Agent will stop. If the & modifier is not used, typing Ctrl+z and then `bg` puts the Application Agent in the background.

The Application Agent should output something like the following:

```
BDNA version 7.7.2 GA, Build 4130(2008_08_28_22_06)
Starting Application Agent on host: BDNA1/192.168.2.100
Master agent trying to start application monitor on host: BDNA1/192.168.2.100
....Done!
```

At this point, the Application Agent has started and the Message Bus is ready to accept connections from other components.

Note: This description defines the processes for a single node installation of BDNA Discover. In more complex, distributed configurations, more steps are required. For further information about distributed installations, refer to the *BDNA Discover 7.7.2 Installation Guide* and the *BDNA Discover 7.7.2 Capacity Planning Guide*.

Starting the BDNA Shell

With the Application Agent running, the BDNA Discover Administrator can start the BDNA shell. The BDNA shell is a command-line interface to BDNA Discover. This interface provides the administrator with fine-grained control over starting, stopping, and managing the components of BDNA Discover.

Prior to starting the BDNA Shell, the Application Agent must be running. To start the BDNA Shell, log in to the system running BDNA Discover as the BDNA Discover user and issue the following commands:

```
% sh bdna.sh
```

The BDNA Shell should output something like the following:

```
BDNA version 7.7.2 Beta, Build 2092(2012_02_02_00_05)

Server: BDNA version 7.7.2, Build 2092(2012_02_02_00_05)
Database: svr115 as BDNA611_2092
Connected to an uninitialized schema
System state: Initial
Tablespace free: 21.1GB (last updated: 2012-02-03 13:32:11)
```

When the BDNA Shell starts, it attempts to connect to the Message Bus to verify it is running. Once successful, it connects via System Interface to the Oracle database. The database (BDNA1_DB) and the user it connects as (BDNA_USER) are defined by the `connection.properties` file. It extracts the system state from the database (Initial), which indicates that the system has not been started.

With the Application Agent, Message Bus and BDNA Shell running, the BDNA Discover Administrator is ready to initialize the Collection Store.

Initializing the Collection Store

Initializing the Collection Store creates the necessary tables in the Oracle database and populates the Collection Store with the initial content. This process may take from 5 to 20 minutes.

Note: The initialization process will drop any existing user objects prior to proceeding. Consequently, be certain that you are connected to an Oracle user that is either empty or that has content that is safe to delete. You cannot undo database initialization.

To initialize the Collection Store from the BDNA Shell, issue the following command:

```
bdna> initdb

This will permanently delete your existing database. Continue (y/n)? y
```

or

```
bdna> initdb [--yes] [--task]
```

If -y option is used, initdb creates a new Collection Store. All existing discovery-related data, scan tasks, and settings are lost. This option does not prompt for confirmation. Use the -y for scripting initializations to run without administrator intervention.

Note: Use the -y flag with caution. Once a Collection Store has been initialized, any data that was previously in that repository will be lost.

If -t option is used, initdb creates a new Collection Store. All existing discovery-related data and scan tasks are lost. All settings, including networks, credentials, groups, users, roles, and component layouts are retained. This option prompts for confirmation.

Note: The –y and –t options can be run separately.

Once the command completes successfully, the Collection Store is initialized and the system is ready to be started.

Advanced Topics 15

About this Chapter

This chapter provides detailed information about the following topics:

- "Directory Exclusion"
- "Enabling the Display of Microsoft Office/Microsoft OS License Keys"
- "Modular Collection Infrastructure (MCI) for UNIX Level 2"
- "Customer-Provided Attributes"

Directory Exclusion

The Directory Exclusion feature of BDNA Discover provides the functionality to specify directories to exclude during a UNIX Level 2 scan. This functionality is controlled through a configuration XML module imported into the BDNA Discover server. A sample module is provided at `$BDNA_HOME/pso/scan/modules/ExcludeDirectory.xml`.

This file should contain sets of XML tags similar to the following:

```
<element elementName="CustomerExcludeDirectory__$NUMBER__"
elementTypePath="root.types.moduleConfig.singleModuleConfig" isTemplate="false"
parentPath="root.$bdna.globalModuleConfig"> </element> <data
elementPath="root.$bdna.globalModuleConfig.CustomerExcludeDirectory__$NUMBER__"
attributeName="applicationFootprintTag">
<![CDATA[CustomerExcludeDirectory__$NUMBER__]]> </data> <data
elementPath="root.$bdna.globalModuleConfig.CustomerExcludeDirectory__$NUMBER__"
attributeName="filePatternList"> <![CDATA[ExcludeDirectory:$DIRECTORY]]> </data>
```

Each set of an element and two data tags will exclude a directory from UNIX Level 2 scanning. Each set must have a unique number represented by `$NUMBER` above; the directory is specified by `$DIRECTORY` in the example above. Directories must be in an absolute path format (start with a leading forward slash). The example `ExcludeDirectory.xml` contains a single exclusion for the `/prj` directory.

The special directory `ALL_AUTOFS` can be specified as well. This will cause BDNA Discover to dynamically determine which directories are being used as `autofs` mount points and exclude them. The `ALL_AUTOFS` option is not supported on HP-UX and Tru64 systems.

Once `ExcludeDirectory.xml` is configured, import it by using the BDNA shell:

```
bdna> module -i $YOUR_PATH/ExcludeDirectory.xml
```

Only one `ExcludeDirectory.xml` configuration module can be loaded in a BDNA Discover server at a time. If you are making changes to a configuration that you already imported, use the `-r` option instead:

```
bdna> module -r $YOUR_PATH/ExcludeDirectory.xml
```

As it is documented in this appendix, you will need to repeat this procedure every time you issue the `initdb` command. The `ExcludeDirectory.xml` file contains instructions on loading this module automatically when you run the `initdb` command.

Enabling the Display of Microsoft Office/Microsoft OS License Keys

The BDNA Discover application collects license keys for discovered assets running Microsoft Windows based Operating Systems and installed Microsoft Office Suite applications. However, due to the sensitive nature of this collected license key data the default behavior of the BDNA Discover reports is to not display the collected Microsoft license keys. The collected Microsoft license keys may only be made available in BDNA Discover Analytics reports if the BDNA Discover Administrator explicitly enables the inclusion and display of this data.

Caution: By enabling this function and choosing to display the Microsoft license key data in the BDNA Discover reports, the customer assumes responsibility for the use and control of the data (Microsoft license keys) made available. BDNA takes no responsibility for the use or misuse of this data.

If the decision to enable this data in the Analytics reports has been made, BDNA Discover highly recommends consideration of the use of BDNA Discover-provided security levels (both user access and ACL) in conjunction with these steps in order to control access to sensitive and valuable information.

The following steps should be executed from a BDNA Discover Linux Component server while logged in as the BDNA Discover user. Invoking the BDNA shell assumes that the path `$BDNA_HOME/bin` is part of the PATH variable. If this is not the case the command in step 1 will require the fully qualified path to the script `bdna.sh`.

To enable the display of Microsoft operating system and Microsoft Office license keys, do the following:

1. To confirm the precise location of the BDNA Discover home directory for use in step 3, enter the following command:

   ```
   $ echo $BDNA_HOME
   ```

2. To invoke the BDNA shell, enter the following command:

   ```
   $ sh bdna.sh
   ```

   The BDNA shell headers appear.

3. To import the module to display license key information (`EnableLicenseKey.xml`) from the `$BDNA_HOME/pso/scan/modules` directory, enter the following command:

   `bdna> module -i` *DIRECTORY*`/pso/scan/modules/EnableLicenseKey.xml`

   where *DIRECTORY* is the directory identified in step 1. BDNA Discover displays the reminder:

   Run `buildPresRls` to rebuild presentation layer containment rules after all module commands if there are <relationship> tags in the modules.

4. Enter the command to rebuild the presentation layer:

   ```
   bdna> buildPresRls
   ```

5. Exit the BDNA shell

   ```
   bdna> exit
   ```

   The change is now complete. Microsoft license key data for Microsoft Operating Systems and Microsoft Office applications is available immediately in the Analytics reports.

Note: If an exception is returned during execution of these steps this may indicate that the module has already been imported. Check the Analytics reports for the desired data. If the data is not available in the reports, capture the text of the exception and log the error into a case with BDNA Discover Customer Support.

Modular Collection Infrastructure (MCI) for UNIX Level 2

BDNA Discover fingerprinting relies on the existence of certain file paths on the target machine. BDNA Discover detects the existence of file patterns of interest for use during UNIX Level 2 discovery. As part of BDNA Discover Level 2 UNIX discovery, the find command is run on the target UNIX machines to collect asset information. Because UNIX servers can contain a large number of files, this process can be lengthy. It may not be suitable for certain times of the day, such as when the servers are already experiencing heavy workloads. The Modular Collection Infrastructure (MCI) allows this data to be gathered ahead of time by running the find script apart from the context of a scan, and placing the results into a file. During a normal Level 2 scan, BDNA Discover will search for and read this file, instead of running the `find` command.

The collection script also indicates whether the results employed a find command issued prior to the scan, and the timestamp of how old the data from the command was.

The Modular Collection Infrastructure consists of three major components:

- An XML file that tells BDNA Discover that there is an additional file to use during scanning and where the file is located.
- Shell scripts that must be run as the root user on every Linux machine that is to be scanned, for which modular collection is to be used.
- The file that was generated by running the shell scripts discussed above.

  This file with the additional modular collection data is stored on the target machine, and will be read during the scan.

Steps for Using MCI

The following is a step-by-step procedure for using MCI:

1. Import a module (module -i) that specifies the file path on the target machine where all the necessary files to invoke the `find` script will be deposited.

   A sample file can be found in

   `$BDNA_HOME/examples/mci/ModularCollectionForOutOfSystemFind.xml`

   The file path specified in this module is called out-of-system find file path. For the rest of this example, the path `/home/bdnauser/ModularCollFind` will be used.

Note: The `out-of-system find file path` must be on a local file system on the target machine. Unexpected behavior may result if this path is a network path (NFS).

The `out-of-system find file path` should be a path for which the BDNA Discover user has been granted the write permission. Refer to “UNIX Credential Requirements” on page 59 for details.

2. Go to the directory where the Perl script is located:

   ```
   cd $BDNA_HOME/scripts/ModularCollection/Perl
   ```

   On the BDNA Discover server, execute the generator Perl script `generateFindProvScript.pl`:

   ```
   perl generateFindProvScript.pl
   ```

   The command above creates a provisioning script called `ProvisionFindScript.sh` in the same directory from which the `generateFindProvScript.pl` was executed on the BDNA Discover server. This is the provisioning script that must be copied over to the target machine.

3. Copy `ProvisionFindScript.sh` over to the target machine in any directory for which the BDNA Discover user has read/write access, such as `/tmp`.

4. Run the provisioning script on the target machine.

   Assuming that the provisioning script was placed in the directory `/tmp` on the target UNIX machine, the commands would be:

   ```
   cd /tmp
   ```

   ```
   sh ProvisionFindScript.sh
   ```

   The above command will create the directory `/home/bdnauser/ModularCollFind/bdnaFind` on the target UNIX machine. Under this directory, all the files necessary for invoking the find script independent of a BDNA Discover scan will be deposited. If the target machine's hostname is foo.acme.com, then after this step, the following files would appear in `/home/bdnauser/ModularCollFind/bdnaFind/` on the target UNIX machine:

   ```
   FindFileList.PatternsConsolidated.foo.acme.com
   FindFileList.Patterns.foo.acme.com
   findscript.sh
   invokeFindScript.sh
   ```

   The script `invokeFindScript.sh` invokes the find script with all the necessary parameters on the target machine. In other words, `invokeFindScript.sh` is simply a wrapper. It is possible to either set up a cron job to run the invoker script or just to run it manually:

   ```
   sh invokeFindScript.sh
   ```

   As a result of running the above command, the file `FindFileList.foo.acme.com` would be created in the same directory. This file contains all the file paths found on the target machine that matched the patterns specified by the BDNA Discover fingerprints. This file will be automatically detected and read during a normal Level 2 scan.

Running the Level 2 Scan

During the Level 2 scan, all collection scripts that invoke `findscript` will now look to see if a directory name that matches the `out-of-system find file path` exists on the target machine. The path `/home/bdnauser/ModularCollFind` will be used as the `out-of-system find file path` for this example. The collection script does the following during the Level 2 scan:

1. Checks if the files `findscript.sh` and `invokeFindScript.sh` exist in the directory `/home/Discoveruser/ModularCollFind/bdnaFind/` on the target machine for which Level 2 collection is being preformed.
2. If the above is true, then it checks if a file named `FindFileList.foo.acme.com` exists in that directory, assuming that the target machine's hostname is `foo.acme.com`.
3. If the above is true, then the collection script reads the results from the file `FindFileList.foo.acme.com` and populates the BDNA Discover attribute `applicationFileFootprintTags`.

   It also sets the attribute `isFindRunOutOfSystem` to Y. It then grabs the timestamp of the file `FindFileList.foo.acme.com` and stores that into the attribute `outOfSystemFindFileTimeStamp`. It then auto-updates the following files on the target machine with those from the current installation:

   `FindFileList.PatternsConsolidated.foo.acme.com`

   `FindFileList.Patterns.foo.acme.com`

   `findscript.sh`

   `invokeFindScript.sh`
4. If step 1 above is true and step 2 above is false, then the script sets the attribute `isProvisionedAndFindNotRun` to Y.

   This implies that the target UNIX machine has been set up with the out-of-system find collection infrastructure, but the out of system find script was never actually invoked. The collection script also auto-updates the following files on the target machine with those from the current installation:

   `FindFileList.PatternsConsolidated.foo.acme.com`

   `FindFileList.Patterns.foo.acme.com`

   `findscript.sh`

   `invokeFindScript.sh`
5. If both steps 1 and 2 above are false, then the collection script runs the find script during the scan as is standard, and populates the attribute `applicationFileFootprintTags` in the usual manner.

Customer-Provided Attributes

BDNA Discover's extensive library of fingerprints enables it to gather information about a wide variety of hardware and software. For hardware and software that is currently beyond the scope of BDNA Discover's fingerprints, customers can provide their own attribute information. This broadens the scope of information about devices or software in the customer's network that BDNA Discover can discover. With this feature, you can make this information available for discovery by BDNA Discover; for example, information such as the administrator for a database, UNIX system, and so on can be supplied by customers as values to custom attributes that can then be discovered by BDNA Discover.

Such information is furnished by customers (and collected by BDNA Discover) in the form of *Customer-Provided Attributes* (CPAs).

CPAs are not intended to replace traditional discovery. In fact, for data that can be automatically discovered, traditional discovery is probably more scalable and reliable than is configuring discovery to use CPAs. (BDNA Discover can extend existing fingerprints and create additional fingerprints for information that already exists on an asset.)

CPA collection is implemented for both UNIX and Windows.

How CPA-Assisted Discovery Works

CPA data is collected during the BDNA Discover Level 2 scan. To configure BDNA Discover to recognize and make use of CPA data, several steps need to occur:

First, customers define the CPA attributes and import them into the scan schema. This will enable BDNA Discover to know where to look for the CPA attributes during a scan, and precisely what to look for.

Then the INI files are copied onto the scan target machines. Machines that are to be scanned by BDNA Discover can have a special INI file placed on the machine by the BDNA Discover Administrator. This INI file holds the CPA values specific to that machine. This file will be read by the BDNA Discover UNIX Level 2 scan. It tells BDNA Discover which additional CPA values it should scan for. The data obtained is then stored in CPAs in the BDNA Discover database repository for inclusion in the Analytics reports, just like any other properties.

The CPAs for the BDNA Discover installation are defined in a BDNA Discover XML driver file. The driver file defines where the discovery should look for the INI file and what attributes to expect inside the file. The attributes can be broken down as follows:

- UNIX: 20 attributes, distributed between the UNIX operating system and the Oracle database
- Windows: 30 attributes, distributed between the Windows operating system, the Oracle database, and Microsoft SQL Server

Note: For Windows, the credentials specified during Scan Setup should be those of a Local Administrator.

Each attribute is collected as a character string that may be up to 4,000 characters long. Each machine can have only one entry for each Operating System attribute, and one entry for each Oracle instance attribute (for as many as may run on that machine).

Caution: The INI file must be in the same location on all scanned machines and have sufficient permissions for the Level 2 scan credential to read the file. Note that the path will be different for UNIX than it is for Windows machines. For UNIX, 444 is a suggested safe setting; for Windows, Read Only is a suggested safe setting.

The CPAs are shown in the existing Analytics reports, in the details views.

CPA is an optional feature. If CPA attributes have been defined, and BDNA Discover performs Level 2 discovery on a machine where there is no INI file containing CPA data, discovery proceeds without error. The attributes that aren’t present are not collected, and the CPA attribute values for that machine are left blank in Analytics reports.

For further information about the presentation of attributes, and how this information may be added to a report view and saved for re-use, refer to the *BDNA Analytics Reports Reference Guide*.

Setting up Customer-Provided Attributes

To set up BDNA Discover to access CPA during Level 2 scans:

1. Identify the machines where BDNA Discover should process CPAs.
2. Create a CPA INI file for each affected machine, specifying the relevant attributes and their values.

   For a look at sample INI files for both UNIX and Windows environments, refer to “Sample Configuration Files” on page 210.

3. Place the INI file on each machine of interest, making sure the file is placed in the same location on every target machine (it should be the same location on every machine, though of course this will be different between UNIX and Windows machines).
4. Repeat the previous step for all UNIX and/or Windows machines of interest.
5. Hand-generate (based on sample) a CPA driver file for either UNIX or Windows.

   This driver file maps customer-defined attributes to BDNA Discover-defined attributes, along with label names. It also contains the location of the CPA driver file and the INI file. For a look at sample driver files for both UNIX and Windows, refer to "Sample Configuration Files" on page 210.
6. Import the CPA driver file into the BDNA Discover schema; for example:

   ```
   bdna> module -i <full path of the CPA driver file>
   ```
7. Set up UNIX/Windows Level 2 scan.

   During the scan, CPAs are collected with BDNA Discover-specified data. You can view CPA data in the Analytics reports.

Driver-File Schema

The driver file is an XML module file used to specify the configuration of the CPAs for the current scan schema. The BDNA Shell's module command is used to enable and configure the CPAs. This command must be used for each BDNA Discover database repository, but the configuration is active for any scans done into that database repository.

The syntax for the core of the driver file is described by the following XML Document Type Definition (DTD):

```
<!-- ................ BDNA CPA Mapping ................ -->

<!ELEMENT bdna-cpa-mapping (mapping-info*) >

<!ELEMENT mapping-info
(element-type,user-attr-name,bdna-attr-mapped-to,label-name) >
<!ELEMENT element-type (#PCDATA) >
<!ELEMENT user-attr-name (#PCDATA) >
<!ELEMENT bdna-attr-mapped-to (#PCDATA) >
<!ELEMENT label-name (#PCDATA) >
```

where:

`bdna-cpa-mapping`:	Root element containing mapping-info(zero or more) elements
`mapping-info`:	Element encapsulating customer-provided information.
`element-type`:	Type of BDNA Discover element to which the customer-defined attribute belongs.
`user-attr-name`:	Name of the customer-/user-provided attribute.
`bdna-attr-mapped-to`:	Name of BDNA Discover attribute(CPA01..CPA10) to which `user-attr-name` is mapped.
`label-name`:	Name of the display label against which this attribute is shown in the report.

INI Files

The INI file containing the attribute name-value pairs have the following format:

```
[BDNA_Element_Type::Element_Specific_Info]

<attribute-name>=<attribute-value>

     …

     …
```

where:

`BDNA_Element_Type`:	Name of the BDNA Discover element/resource to which this attribute name-value pair section belongs.
`Element_Specific_Info`:	Any information specific to the element. For example, this could be SID in case of Oracle.
`attribute-name`:	Name of the user-specified attribute.
`attribute-value`:	Value of the user-specified attribute

Sample Configuration Files

This section provides:

- Sample INI files that a customer would place in a predefined location, on all the machines of interest
- Sample CPA driver file that the BDNA Discover administrator would import on a machine where BDNA Discover is installed
- Sample code showing how to define multiple INI file paths for Windows and Unix driver files

UNIX

Sample INI File

```
[UNIXOperatingSystem]
Administrator="James Jones"

[OracleDatabase::ora92]
Site="San Jose"
SupportGroup="SOS/DCO"
CriticalityLevel="2"
Project="RAC-Pack"
Administrator="Jobs"
SecondaryAdministrator="Bob"
Department="corp_it"
```

Sample Driver Module

```
<module

    name="com.bdna.modules.examples.CPAInfo"

    displayLabel="CPA specific Module Configuration Information"

    requiredPlatformVersion="030301"

    version="1"

    lastCompatibleVersion="1">

<!-- See $BDNA_SRC/modules/com/bdna/modules/app/CPA.xml -->

<!--

    Example of how to specify CPA driver file :

<data attributeName="collectedCPADriverData"

elementPath="root.$bdna.globalModuleConfig.CPADriver"><![CDATA[<?xml
version="1.0" encoding="UTF-8"?>

<!DOCTYPE bdna-cpa-mapping SYSTEM "cpa-mapping.dtd">

<bdna-cpa-mapping>

  <mapping-info>

    <element-type>UNIXOperatingSystem</element-type>

    <user-attr-name>SysAdmin</user-attr-name>

    <bdna-attr-mapped-to>CPA01</bdna-attr-mapped-to>

    <label-name>System Administrator</label-name>

  </mapping-info>

 <mapping-info>

    <element-type>OracleDatabase</element-type>

    <user-attr-name>DbaAdmin</user-attr-name>

    <bdna-attr-mapped-to>CPA05</bdna-attr-mapped-to>

    <label-name>Database Administrator</label-name>

  </mapping-info>

</bdna-cpa-mapping>
]]></data>

-->

<!--

    Example of how to set the path where the CPA INI file is located:

<data attributeName="CPADiscovery::CPAINIFilePath"

elementPath="root.$bdna.globalModuleConfig.com_bdna_modules_app_CPA"><![CDATA[
/tmp/USData.ini]]></data>

-->

</module>
```

Windows

Sample INI File

```
[WindowsOperatingSystem]

Administrator="Joe Smith"

[MSSQLServer::SQLENT1@WINFP05]

DbAdministrator="Peter Stevens"

Department=”Human Resources”

[OracleDatabase::ora92]

GVRev="2.3.4"

Site="Mountain View"

SupportGroup="SOS/DCO"

CriticalityLevel="2"

RoomRow="P2"

Project="RAC-Pack"

Administrator="Jack"

SecondaryAdministrator="Bob"

Department="corp_it"
```

Sample Driver File

```
<module
```

```
    name="com.bdna.modules.examples.CPAInfo"
    displayLabel="CPA specific Module Configuration Information"
    requiredPlatformVersion="030301"
    version="1"
    lastCompatibleVersion="1">
<!-- See $BDNA_SRC/modules/com/bdna/modules/app/CPA.xml -->
<!--
    Example of how to specify CPA driver file :
<data attributeName="collectedCPADriverData"
elementPath="root.$bdna.globalModuleConfig.CPADriver"><![CDATA[<?xml
version="1.0" encoding="UTF-8"?>
<!DOCTYPE bdna-cpa-mapping SYSTEM "cpa-mapping.dtd">
<bdna-cpa-mapping>
  <mapping-info>
    <element-type>UNIXOperatingSystem</element-type>
    <user-attr-name>SysAdmin</user-attr-name>
    <bdna-attr-mapped-to>CPA01</bdna-attr-mapped-to>
    <label-name>System Administrator</label-name>
  </mapping-info>
 <mapping-info>
    <element-type>OracleDatabase</element-type>
    <user-attr-name>DbaAdmin</user-attr-name>
    <bdna-attr-mapped-to>CPA05</bdna-attr-mapped-to>
    <label-name>Database Administrator</label-name>
  </mapping-info>
</bdna-cpa-mapping>
]]></data>
-->
```

```
<!-- Example of how to set the path where the CPA INI file is located:

<data attributeName="CPADiscovery::CPAINIFilePath"

elementPath="root.$bdna.globalModuleConfig.com_bdna_modules_app_CPA"><![CDATA[
/tmp/USData.ini]]></data>

-->

</module>
```

Sample Code Showing How to Define Multiple INI File Paths in Driver File (Unix and Windows)

```
<data

attributeName="CPADiscovery::IniFilePath"

elementPath="root.$bdna.globalModuleConfig.com_bdna_modules_app_CPA">

<![CDATA[/tmp/OSInfo.ini,/tmp/IT/USData.ini,/tmp/IT/UKData.ini]]>


<data

attributeName="CPADiscovery::CPAWindowsIniFilePath"

elementPath="root.$bdna.globalModuleConfig.com_bdna_modules_app_Windows_CPA"><
![CDATA[C:\\CPADriver\\USData.ini,C:\\CPADriver\\UKData.ini]]></data>
```

Working with Bulk Load Files A

About this Appendix

This appendix provides information about creating and using bulk load files.

If you have many elements (networks, credentials, tasks, and so on) to create, the BDNA Discover user interface is not the most efficient method to create them; for large deployments, it can be tedious and error-prone. Instead, consider creating and using bulk load files. Bulk loading is available for the following types of elements:

- Scan tasks
- Networks
- IP exclusions
- Groups
- Credentials
- Namespaces

For each of these element types, the procedure is identical. However, the file formats for each are different.

Note: For information about the bulk load tool itself, refer to "BDNA Discover Utility Reference."

It is highly recommended to use the bulk load tool to automatically turn Excel-based Network Catalogs into bulk load files.

Command-Line Usage

Before creating/editing the input file, make sure that an installation configured to connect to a BDNA Discover database has been initialized and has the locations and credential templates you want to use in the scan task are already defined. As part of the parse process the generator will verify that the locations and credential selected already exists. This verification forces the following workflow to successful bulk import the data needed to setup the scan:

- `initdb`
- Import credential templates
- Import networks definitions
- Import bulk scan task

To import the scan task directly without an intermediate file:

```
sh $BDNA_HOME/bin/runjava.sh com.bdna.agenda.BulkCommon BulkTask4
<ScanTaskDefinitionsFile>
```

The file will be processed and imported directly to the database configured in $BDNA_HOME/conf/connection.properties if the content definitions have no errors.

General Format of Input File

Each input file consists of one or more lines. Each line can be blank or start with '#' symbol in which case they are ignored by the parser.

If the multi-column line matches the 'header' it is also ignored. For any other multi-column line the parser expects the specific column format described below in order. The format chosen makes it easy to work with the data in Excel for readability and then exported to a tab delimited text file. The parser only works on text files so make sure to export to a tab delimited file before running the parser. For some unknown reason, Excel wraps some of the text columns in double-quote and not others. After the parser splits the tabbed-columns it will also strip the outer set of double-quotes if present before processing the column.

Tips for creating input files:

- Use Excel

  It is a lot easier to line up columns and expand/compress columns when entering data.

- Treat columns as a literal in Excel

  This is done to prevent things such as date conversion and wrapping strings with double quotes) by starting the field with a single quote.

- Avoid exporting the Excel file as a comma-separated field

  Instead, select, cut, paste from Excel to Notepad or another text editor. This prevents undesired cell conversions in Excel.

Importing Bulk Load Files

To import bulk load files to BDNA Discover:

1. Log in to BDNA Discover and open Scan Administration.
2. Under Scan Setup, locate the element type to be imported and click it.

   For this example, choose Networks.

   Under the application menu, a button Import Network appears. (For other element types, a similar, appropriately named button appears in the same place. For example: Import IP Exclusions.)
3. Click Import Networks.
4. Use the Browse button at the far right side of the wizard to locate the file to be imported.
5. Select the file to import and click Next.

   The system will validate the file. This may take some time.

   The second page of the Import Networks wizard will display the results of the validation. Each row of the input file will be displayed, along with any errors to the right of each line.
6. If there are errors, correct them in the original file and save them.
7. Click the Prev button to go back until the first page of the Import Networks wizard is showing.

   There is no need to Browse for the same file twice. Clicking Next will reload the file for validation.

8. When satisfied that there are no errors in the file to be imported, click Import at the bottom right of the wizard.

The last page of the Import Networks wizard will show the progress of the import. When the last line begins "Finished importing…" the import has completed.

9. Click Close to close the Import Networks wizard.

Elements imported will now appear in the element listing in the middle pane. In this example, all networks imported will be listed in this pane, in alphabetical order.

Generating Bulk Load Files

Credential Bulk Load Files

The Scan Administration application allows an administrator to create credential templates. These are pieces of user defined credential information to be used when defining the credentials associated with a Scan Task. Each template for a specific type and level of access can then be select from the UI when defining the credentials to be used in a Scan Task.

Credential Bulk Load Input File Format

Each credential bulk load input file consists of one or more lines. Lines may be blank or start with a '#' symbol, in which case that line is ignored by the parser. The parser expects all other lines to have the following fixed format:

3 columns separated by 2 tab characters followed by a new line.

If the multi-column line matches the 'header', it is also ignored. For any other multi-column line, the parser expects the specific column format in "Using Bulk Load From The Command Line" on page 219, in order. This format makes it easy to work with the data in Excel for readability and then export it to a tab delimited text file. The parser only works with text files, so make sure to export to a tab delimited text file before running the parser. For an unknown reason, Excel wraps some of the text columns in double quotes, but not others. After the parser splits the tabbed columns, it will also strip the outer set of added double quotes, if present, before processing the column.

The parser uses the following characters during parsing:

'-', '\t', '=', ';', ' ', '\n'

If some of the data in the credential arguments section uses these characters, a slightly modified version of HTML encoding can be used to enter the data. Encodings are indicated in the following table.

Table 18: Credential Bulk Load Input File Character Encodings

Character	Escape Sequence
'-'	&#45
'\t'	&#09
'='	&#61
';'	&#59
' '	&#32

Proper HTML formatting for character encodings would end each escape sequence with a semicolon, but this is not possible because the semicolon character is being used as a separator in the credential bulk load input file.

Before submitting the script for parsing, replace the reserved characters in any data fields containing them with the HTML encoded equivalents from the table above. You can optionally use the following command to automatically translate the data:

```
sh runjava.sh com.bdna.agenda.BulkCred
EscapeData4 <sourceFileName> <destFileName>
```

Bulk Task Load Files

The Scan Administration application provides simple, easy to use functionality for creation and management of BDNA Discover Scan Tasks.

This section describes the format used to create bulk Scan Tasks that are compatible with Scan Tasks created using the UI. Whenever possible, the same columns and format used by the prior version bulk Scan Task has been used.

Bulk Scan Task Input File Format

Note: For sample bulk Scan Task input files and column descriptions, refer to “Using Bulk Load From The Command Line” on page 219.

Each bulk Scan Task input file consists of one or more lines. Lines may be blank or start with a ‘#’ symbol, in which case that line is ignored by the parser. The parser expects all other lines to have the following fixed format:

10 columns separated by 9 tab characters followed by a new line.

If the multicolumn line matches the ‘header’, it is also ignored. For any other multicolumn line, the parser expects the specific column format described in “Using Bulk Load From The Command Line” on page 219, in order. This format makes it easy to work with the data in Excel for readability and then export it to a tab delimited text file.

The parser only works with text files, so make sure to export to a tab delimited text file before running the parser. For an unknown reason, Excel wraps some of the text columns in double quotes, but not others. After the parser splits the tabbed columns, it will also strip the outer set of added double quotes, if present, before processing the column.

Exporting Bulk Load Files

Bulk loading applies only to Collection Store (not FactBase) implementations.

Sample commands for exporting . . .

```
. . . namespaces:
sh $BDNA_HOME/bin/runjava.sh com.bdna.pl.util.BulkLoader -e:1 -f:<export_file>
. . . networks:
sh $BDNA_HOME/bin/runjava.sh com.bdna.pl.util.BulkLoader -e:4 -f:<export_file>
. . . IP exclusions:
sh $BDNA_HOME/bin/runjava.sh com.bdna.pl.util.BulkLoader -e:5 -f:<export_file>
. . . groups:
sh $BDNA_HOME/bin/runjava.sh com.bdna.pl.util.BulkLoader -e:3 -f:<export_file>
```

```
. . . credentials:
sh $BDNA_HOME/bin/runjava.sh com.bdna.agenda.BulkCommon ExportCred4 <export_file>
. . . tasks:
sh $BDNA_HOME/bin/runjava.sh com.bdna.agenda.BulkCommon ExportTask4 <export_file>
```

Using Bulk Load From The Command Line

In addition to the GUI-based Bulk Load utility, BDNA Discover provides the means to bulk load elements, networks, containers, Level 1 data, or exclusion lists, from the command line. As with the GUI Bulk Import utility, you can choose to validate these elements, or validate and then import them.

There are two commands for this purpose: BulkLoader and BulkCommon.

BulkLoader Arguments

The syntax for the BulkLoader follows, along with details for each of the parameters.

```
bin/runjava.sh com.bdna.pl.util.BulkLoader
   -e:<export_type>
   -i:<import_type>
   -f:<file_path>
   -o<operation_action_type> [-t:<element_type> -l:<location>]
   -u:<user>
   -p:<password>
   -c:<compressed,0|1>
```

Where:

-i Import type. (Required) A single-digit code that specifies the type of import desired:

1 = Namespace
3 = Group
4 = Network
5 = Exclusion list

-f File path of the file to bulk-import. (Required)

-o Operation action type. (Optional) A single-digit code to specify the type of action to take with the file noted in –i:

1 = Default. Validate and import. If chosen, you must also specify element type (-t flag) and location (-l flag).
2 = Validate only

-t Element type. (Required for validate-and-import operations.) Full name of the element type.

-l Location. (Required for validate-and-import operations.) Full path for the elements' destination location.

-u User. (Optional) Specify only if the user is not already defined in $BDNA_HOME/connection.properties.

-p Password. (Optional) Specify only if the user is not already defined in $BDNA_HOME/connection.properties)

-c Compression. (Optional)

0 = Default. Not compressed
1 = Compressed

BulkCommon Arguments

The basic format of the BulkCommon command is as follows:

```
BulkCommon <type of task> <destination>
```

Import a credential bulk load file:

```
BulkCommon BulkCred4 <credentialFile>
```

Export a credential bulk load file:

```
BulkCommon ExportCred4 <credentialFile>
```

Process a scan task bulk load file:

```
BulkCommon BulkTask4 <tasksFile>
```

Export a credential bulk load file:

```
BulkCommon ExportTask4 <credentialFile>
```

Used for credential files that have characters used internally by BDNA Discover. The file must be escaped before using one of the credential import commands.

Examples: Bulk Import

```
sh $BDNA_HOME/bin/runjava.sh com.bdna.pl.util.BulkLoader -i:4
-f:/tmp/BIT02_BulkIplevel1.txt

sh $BDNA_HOME/bin/runjava.sh com.bdna.agenda.BulkCommon BulkCred4
/tmp/BIT03_BulkCred4.txt

sh $BDNA_HOME/bin/runjava.sh com.bdna.pl.util.BulkLoader -i:3
-f:/tmp/BIT04_BulkContainers.txt

sh $BDNA_HOME/bin/runjava.sh com.bdna.agenda.BulkCommon BulkTask4
/tmp/BIT05_BulkTask4.txt
```

Note: To obtain some sample import files, contact BDNA Discover Technical Support.

BDNA Discover supports the following bulk-load file types:

- Scan Tasks
- Networks
- IP Exclusions
- Groups

- Credentials
- Namespaces

The following tables and descriptions explain the file format of each bulk load file type:

Scan Tasks

Table 19: Bulk Scan Task Column Summary

Column Number	Column Name	Description
1	Task Type	One of the possible bulk Scan Task types. They correspond to the content definitions for package roots.
2	Packages	Given a package root, lists the associated packages that are part of the Scan Task or the special symbol '*', indicating all packages associated with the package root type.
3	Locations	Semicolon separated list of locations where the bulk Scan Task should apply.
4	Start_Scan	Calendar start date for the bulk Scan Task.
5	Stop_Scan	Calendar end date for the bulk Scan Task.
6	Shift	Definition of the shifts for the Scan Task.
7	Scan_interval	One of 'once', 'never_scanned', or <n> - the interval between collection times.
8	Credentials	Semicolon separated list of complete credentials to be used in the bulk Scan Task.
9	Scan_Name	Name of the Scan Task element and root name for other related elements associated with task. Note that this field cannot be longer than 119 characters.
10	Scan_Desc	User-defined description for the task.
11	priority	Optional column to indicate scan task priority.

Individual Description of Each Input Column

Following is a definition of the format for each individual column, in the order that the columns must be defined. If the column has more structure than simple strings, semicolons, minus signs, and commas are generally used in that order to separate internal structure.

Column 1: Task Type

The Task Type column lists the type of bulk Scan Task, as defined by content package root definitions. It should be one of:

- Inventory
- BDNA Discover Namespace Dedupping

- Cleanup BDNA Discover UNIX Processes
- Credential Coverage Task
- UNIX Level 2 Dedupping Task
- Update SSH Credential Task
- Remove Windows XP Profile
- Clear files from BDNA Discover UNIX cache (working directory)

This corresponds to either the inventory Scan Task in the UI or one of the defined maintenance tasks in the UI. Note that the BDNA Discover Namespace Dedupping task is not currently definable in the UI, but can be bulk imported.

Column 2: Packages

Either '*', indicating all packages associated with the package root type, or a <BDNA,> delimited list of packages for the root that applies for this Scan Task.

For an inventory Scan Task, the entry must be one or more of the names matching the package selection screen in the UI.

Level 1:

- Basic Scan - Level 1 (no credentials required)
- Basic Scan - Level 1 Light
- Basic Scan - Level 1 Rigorous

Level 2:

- Mac OS
- SNMP
- SNMP v3
- SMI-S
- UNIX
- UNIX - Scan for System Information Only
- VMware ESX
- Windows
- Windows - Scan for System Information Only

Level 3:

- BlackBerry Server (Windows)
- IBM DB2 Database Server (UNIX)
- Informix Database Server (UNIX)
- Microsoft SQL Server (Windows)
- Oracle Applications (UNIX)
- Oracle Database Server (UNIX)

- Oracle Database Server (Windows)
- SAP Solution Manager
- Sybase Database Server (UNIX)
- WebLogic Administration Server (UNIX)
- WebSphere Application Server (UNIX)

The choices for a maintenance-type package include the following:

- BDNA Discover Namespace Dedupping
- Cleanup BDNA Discover UNIX Processes
- Level 1<BDNA,>Level 2 Credential Detection
- UNIX Level 2 Dedupping Task
- Update SSH Credential Task
- Remove Windows XP Profile
- Clear files from BDNA Discover UNIX cache (working directory)

Column 3: Locations

A <BDNA,> separated list of element full names corresponding to the locations where the bulk Scan Task applies; for example:

```
root.$bdna.NS_GLOBAL.network_halley<BDNA,>root.$bdna.NS_GLOBAL.network_abe2
```

The example above states that two networks are the associated locations for this Scan Task. Other element full names can be used such as groups or projects if they have already been configured by the BDNA Discover Administrator. The BDNA Discover Namespace Dedupping type task should always use the `root.$bdna.folder_dedup` element full name since this folder is the container of all namespaces.

Column 4: Start_Scan

The date and time when the Scan Tasks should become active. It must be in the following format: ‘MM/DD/YYYY hh:mm:ss’. The value ‘immediately’ is also accepted to mean start as soon as the calendar is defined.

Column 5: Stop_Scan

Date and time the scan should stop and become inactive. Use the same data format as Start Scan or ‘never’ to indicate the Scan Task does not end.

Column 6: Shift

Defines the times during each day when collections are enabled for a particular Scan Task. It has the following syntax:

```
shifts ::= shiftDefinition{+shiftDefinition}*

shiftDefinition ::= always | shiftType;hh1:mm1;hh2:mm2;shiftPattern

shiftType ::= daily | nights | workHours | weekdays | weekend
```

```
shiftPattern ::= daily;<n> | weekly;<n>;day{,day}*

 day ::= mon | tue | wed | thu | fri | sat | sun
```

Table 20: Examples of Shift Definitions in a Bulk Load File

Example	Description
always	Allowed to collect 24x7 every day. equivalent to '00:00;24:00;daily;1'
daily;08:00;18:00;daily;2	Collect every other day during business hours 8am to 6pm
weekdays;00:00;08:00;weekly;1;fri+weekdays;18:00;24:00;weekly;1;sat,sun	Collection on weekends during non-business hours
weekdays;08:00;18:00;weekly;3;mon,tue,wed,thu,fri	Collect every three weeks, every workday, during business hours

Note: The Scan Administration application in BDNA Discover supports a limited subset of the shift definitions above. Editing a bulk task with the UI saves only the shift definitions supported by the UI.

Column 7: Scan_interval

One of ‘once’, ‘never_scanned’, or <n>. ‘Once’ means to do each collection successfully once regardless of what has happened in previous tasks. ‘Never scanned’ means only do the collection if previous tasks have not done it successfully already. <n> is used to repeat the Scan Task, with each collection repeating every <n> seconds. For example ‘3600’ is once an hour.

Column 8: Credentials

If the package selected requires no credentials, such as ‘Inventory/level1’, this column can be blank. Otherwise it is has the following format:

```
credentials ::= <empty>|credDefs

credDefs ::= credDef[<BDNA,>credDef]*

credDef ::=
credTypeName<BDNA,1>credTemplateLvl2Name[<BDNA,1>credTemplateLvl3Name]
```

where each credDefs represents a complete credential to be used by the bulk Scan Task. The credTemplateLvl3Name part is only required if the credTypeFullname is a Level 3 type connection method; for example:

```
snmp<BDNA,1>snmp_snmp1<BDNA,>ssh<BDNA,1>    \
ssh_ssh1<BDNA,>defaultwincs<BDNA,1>         \
defaultwincs_win1<BDNA,>OracleConnection    \
<BDNA,1>ssh_ssh1<BDNA,1>OracleConnection_unixOra1
```

Defines three new credential instances of types: snmp, wincs, and oracleConnection with the values from the credential templates specified: snmp_snmp1, ssh_ssh1, defaultwincs_win1, and OracleConnection_unixOra1.

Column 9: Scan_Name

Root name given to the Scan Task. The name is also used as a base for calendar, shifts, and intervalSets associated with this task. In order to avoid conflicts, the timestamp at the time of parsing is appended to each generated tasks from the input file.

Column 10: Scan_Desc

User defined description for the task.

Column 11: priority

Optional column to indicate scan task priority. It can be one of two string values: Priority or Normal. The default if optional parameter is not defined is Normal.

One interesting issue surfaces when doing two or more scan tasks that have different priorities and overlapping collections (two scan tasks for the same network for example). If each collection for the scan task happens to be queued simultaneously, then the max of all scan tasks should be the assign priority for the shared collection. For instance, if Scan Task 1 is assigned a Normal priority and Scan Task 2 is assigned a Higher priority and they both happen to share an IP range, then the underlying collection tasks of Scan Task 2 will be queued and executed first, since it has the Higher priority.

It is also important to note that since the two Scan Tasks share a common network in the above example, the shared collection tasks will only be executed once, provided they are in the same scan calendar.

Networks

Table 21: Bulk Networks Column Summary

Column Header	Required Field	Description	Example
Add/Delete	Yes	Allows the user to add or remove a row. The value must be set to A if a row is to be added. The value must be set to D if a row is to be removed. **Note:** Delete only removes elements from an object. It does not delete the object itself.	A
Network Name	Yes	The name of the network.	Network_192_168_1_0
Zone (IP)	Yes	The IP address, list of IPs, IP range, or subnet to be imported.	Single IP: 192.168.1.1 List of IPs: 192.168.1,192.168.1.2,192.168.1.3 Range of IPs: 192.168.1.1-192.168.1.50 Subnet: 192.168.1.0/16

Table 21: Bulk Networks Column Summary

Column Header	Required Field	Description	Example
CLE	No	The Collection Engine(s) name with which the network is associated. If this field is blank, then it will default to Default collection engine.	Leave this field blank.
NAMESPACE	Yes	The namespace name with which the IP address, IP range, or subnet is associated.	Global Namespace
DNS Server	No	A comma separated list of DNS servers used to determine the hostname of an IP address for that network. You can specify up to three DNS servers.	192.168.45.7,192.168.45.8,192.168.45.9
Max Concurrent WinCS Tasks	No	Maximum number of concurrent Windows tasks you can configure on a per network basis.	Any numeric value. A value of 0 in this column means “no restriction”. A value > 0, for e.g say 3 means that at any time, for this network, there can only be a maximum of three L2 Wincs concurrent tasks executing.

IP Exclusions

Table 22: Bulk IP Exclusions Column Summary

Column Header	Required Field	Description	Example
IP	No	The IP address, list of IPs, IP range, or subnet to be excluded from the scan.	Single IP: 192.168.1.1 List of IPs: 192.168.1,192.168.1.2,192.168.1.3 Range of IPs: 192.168.1.1-192.168.1.50 Subnet: 192.168.1.0/16
Networks	No	The name of the network to be excluded from the scan. If the value of IP column is not specified, then this column is required.	Network_192_168_1_0
Namespace	Yes	The namespace name with which the IP address, IP range, or subnet is associated.	Global Namespace

Groups

Table 23: Bulk Groups Column Summary

Column Header	Required Field	Description	Example
Parent Type	Yes	The full element type name of a group to be imported.	root.types.resource.logicalset.project root.types.resource.logicalset.organization.division root.types.resource.logicalset.geography.region
Parent Key Value	Yes	The name of the group to be imported.	Mountain View
Parent Key Column	No	The attribute name of the parent key to be imported.	Leave this field blank
Child Type	Yes	The full element type name of the collection zone.	root.types.collectionZone
Child Key Value	Yes	The attribute value of the child type to be imported. Usually this is the IP address, IP range, or the subnet to be attached to a group.	192.168.1.12 192.168.1.1–192.168.1.255 192.168.1.0/24
Child Key Column	Yes	The attribute name of the child key to be imported.	hostIPAddress
Add/Delete	Yes	Allows the user to add or remove a row. The value must be set to A if a row is to be added. The value must be set to D if a row is to be deleted.	A
IPSpace	No	The IPSpace associated with identified child element. If blank, use `Default_IPSpace`.	`NS_PROD` `NS_TEST`

Credentials

Table 24: Bulk Credentials Column Summary

Column Header	Required Field	Description	Example
credential_name	Yes	Name of the credential template. The bulk code automatically forms the name by merging `<credType>_<name>`. This is the name that should be used when referring to this template in bulk task import.	See below
credential_type	Yes	Credential type element name to which to apply the next column (credential arguments). It must match one of the existing connection methods in the system that requires user entered arguments.	See below
credential_args	Yes	A value pair set of arguments separated by the <BDNA,> tag: CredArgs::=credArg{<BDNA,>credArg}* credArg::=<atrName><BDNA,1><value>	See below

Namespaces

Table 25: Bulk Namespaces Column Summary

Column Header	Required Field	Description	Example
Add/Delete	Yes	Allows the user to add or remove a row. The value must be set to A if a row is to be added. The value must be set to D if a row is to be deleted.	A
Display Label	Yes	The value to be shown in the UI.	NS_BDNA
name	Yes	The value to be stored in the BDNA Discover database (not visible in the UI).	NS_BDNA

BDNA Shell Command Reference B

About this Appendix

This appendix provides a reference to the commands contained in the BDNA Shell. The BDNA Shell is a command shell utility that enables an administrator to manage the BDNA system.

To run the BDNA Shell utility, log in to the BDNA console as the bdna user and type the following command:

```
$ sh bdna.sh
```

Command Summary

The following conventions are used: Optional arguments appear between square brackets: []. Required arguments appear between angled brackets: <>.

The following is a list of supported BDNA Shell commands:

bounce

Syntax:

```
bounce [--kill, -k] <component name>
```

Description: This command restarts a component. Using the –kill option will force a component to shutdown.

Note: Do not use the kill option unless the component refused to stop.

Examples:

In the following example, the bounce command will restart the RS component:

```
bounce RS
```

In the following example, the `bounce` command will kill the RULE0 component and then will restart the component:

```
bounce ––kill RULE0
bounce –k RULE0
```

buildPresRls

Syntax:

```
buildPresRls
```

Description: This command rebuilds the presentation layer containment rules. Usually this command needs to be executed after the `module` command.

Examples:

In the following example, the `buildPresRls` command will rebuild the presentation layer containment rules:

```
buildPresRls
```

checkpoint

Syntax:

```
checkpoint <component>
```

Description: This command checks the Rule Engine to ensure the checkpoint is up-to-date. This command supports Rule Engines only.

Examples:

In the following example, the `checkpoint` command will check RULE0 to ensure the checkpoint is current:

```
checkpoint RULE0
```

cleanDeadHost

Syntax:

```
cleanDeadHost <hostName>
```

Description: This command changes a host, and the agent and all components running on the host, to the `Init` state.

Note: It is important to ensure that the host is really dead, before executing this command.

Example:

```
cleanDeadHost BDNA_TEST
```

connMethod

Syntax:

```
connMethod [--list, -l] <cle name> <connectionMethod>
connMethod [--enable, -e] <cle name> <connectionMethod>
connMethod [--disable, -d <cle name> <connectionMethod>
```

Description: This command lists, activates or deactivates the connection method for the specified CLE (Collection Engine). This is useful, for example, to restrict a CLE from running certain types of tasks (such as all WinCS related tasks).

Examples:

In the following example, the `connMethod` command lists the connection method "wincs" used by CLE1 component:

```
connMethod --list CLE1 wincs
```

In the following example, the `connMethod` command activates the connection method "wincs" on the CLE1 component:

```
connMethod --enable CLE1 wincs
```

In the following example, the `connMethod` command deactivates the connection method "wincs" on the CLE1 component:

```
connMethod --disable CLE1 wincs
```

dedup

Note: While this command is available in BDNA Discover, if you type in `help` from bdna.sh, it doesn't appear in the command list. Nevertheless, the command does work.

Syntax:

```
dedup
```

Description: This command schedules a deduping event(s) for the attribute set `namespaceDuplicateInfoStatic`. The command will generate the task only if the system is running. In particular, the Agenda Manager, all the Rules Engines and the Collection components should be running and healthy. There might be significant latency between the time the command is issued and the time when the task is actually executed.

Examples:

In the following example, the `dedup` command will schedule a deduping event:

```
dedup
```

exit

Syntax:

```
exit
```

Description: This command exits the BDNA Shell.

Examples:

In the following example, the `exit` command will quit the BDNA Shell and return to the UNIX shell:

```
exit
```

expcoll

Syntax:

```
expcoll [-x exclusionFileName] [-i ipSpaceName] <collectionStoreName>
<collectionStoreDescription> <outputFileName>
```

Description: The `expcoll` command exports the collection store data from a schema that has been initialized using `initdb`. Do not run the `expcoll` command against a FactBase (a schema that has been initialized using `initfb`). You must input a collection name, description, and output file name when using the `expcoll` command. You can change the name and description of the collection store at import time (refer to "impcoll" on page 233 for more information). Because the target output file is created by the `expcoll` command, the output file must not exist. If the target output file does exist, `expcoll` will return an error. If no extension is specified for the output file name, then a default extension of .bdnacs will be appended. You can specify the optional argument `-x` to exclude items from the dataset being exported.

Example:

```
expcoll -x /home/bdna/exclude.txt janScan1 "Level 1, 2, 3 scan of Mountain View
office, in January" /home/bdna/janScanDump.bdnacs
```

halt

Syntax:

```
halt [host]
```

Description: This command stops the BDNA Application Agent running on the host. If no host is specified, it shuts down the entire BDNA control system. Shut down the BDNA system cleanly before issuing this command. If the system is not in the “init” state when running `halt`, an error message is returned and the halt will not happen.

Examples:

In the following example, the `halt` command will shut down all Agents that are currently connected to the BDNA system:

```
halt
```

In the following example, the `halt` command will shut down only the Agent running on host1.domain.com:

```
halt host1.domain.com
```

help

Syntax:

```
help [command]
```

Description: This command displays a list of commands supported by the BDNA Shell. When the [command] parameter is specified, it displays the help message for that command. When no command is specified, a list of all available commands is displayed.

Examples:

In the following example, the `help` command will display a complete list of commands that are supported by BDNA Shell:

```
help
```

In the following example, the `help` command will display a help message for the `updcomp` command:

```
help updcomp
```

idle

Syntax:

```
idle <regular expression matching component names>
```

Description: This command checks if one or more components are idle. By using the proper regular expression, it can check if the system is globally idle.

Examples:

In the following example, the `idle` command will show the idle status of all Rule Engine components:

```
idle RULE.*
```

impcoll

Syntax:

```
impcoll [ -n collectionStoreName ] [ -d collectionStoreDescription ]
[ -i ipspace ] <dataFileName>
```

Description: `impcoll` imports collection store data into a FactBase from a file generated by the `expcoll` command. The command optionally allows the user to change the Collection Store name and/or the Collection Store description specified in the export file. The user can use the `-i` option to associate the collection store to an IPSpace. The IPSpace must already exist in the FactBase before importing. For information on how to create an IPSpace and an explanation of when to use an IPSpace type `help mkipspace`. If IPSpace is not specified, the Collection Store is imported into the default Global IPSpace (a default IPSpace called Global IPSpace is always available).

Examples:

```
impcoll janScanDump.exp
```

```
impcoll -n janScanNewName -i "MV 10net" janScanDump.bdnacs
```

```
impcoll -n "New January Scan Name" -d "Scan Executed Jan 1 for MV"
/home/bdna/janScanDump.bdnacs
```

info

Syntax:

```
   info
```

Description: This command shows the database and system state information.

Examples:

In the following example, the `info` command will display the currently connected database and the system state of the BDNA system:

```
   info
```

initdb

Syntax:

```
   initdb [-y] [-t]
```

Description: Initializes the Oracle schema as a BDNA Collection Store.

Examples:

initdb can be run without options. A prompt will appear requesting confirmation before initializing the BDNA repository:

```
   initdb
```

If the -y option is used, initdb creates a new Collection Store. All existing discovery-related data, scan tasks, and settings are lost. This option does not prompt for confirmation.

In the following examples, the `initdb` command will initialize the BDNA repository without prompting for a confirmation message. This can be useful for scripting:

```
   initdb –y
```

If the -t option is used, initdb creates a new Collection Store. All existing discovery-related data and scan tasks are lost. All settings, including networks, credentials, groups, users, roles, and component layouts are retained. This option prompts for confirmation.

```
initdb –t
```

Note: The -y and -t options can be run separately.

---

initfb

Syntax:

```
initfb [ -y ]
```

Description: `initfb` is similar to `initdb` but it is used only to initialize the FactBase schema. A FactBase schema is the aggregate repository for all collections and inventories. A FactBase schema may not be used for scanning.

Use `initdb` to create a schema for scanning purposes. The target schema to be initialized by `initfb` is specified in the connection.properties file. The schema owner must be created before executing the `initfb` command. If the schema being initialized has been previously initialized using either the `initdb` or the `initfb` command, you will be prompted with a warning message before you may proceed. Executing this command will result in the creation of a complete and empty FactBase; any data existing in the schema will be lost.

The `initfb` command asks for confirmation unless the -y option is specified.

Examples:

In the following example, `initfb` will ask for confirmation before executing the command

```
initfb
```

In the following example, `initfb` will automatically execute the command.

```
initfb -y
```

list

Syntax:

```
list [--all, -a] [--verbose, -v]
```

Description: This command displays a complete list of BDNA components, including component name, Agent host name, and component status for the BDNA system.

Examples:

In the following example, the `list` command will display a complete list of BDNA components, including the ApplicationMonitor, ApplicationMonitorAdmin, and Database Watcher:

```
list –a
```

In the following example, the `list` command will display a complete list of BDNA components. The ApplicationMonitor, ApplicationMonitorAdmin, and Database Watcher are excluded from the list:

```
list
```

The following example is similar to the above example, but it also displays the amount of time for which the component has been running:

```
list -v
```

The two arguments can also be combined:

```
list –av
```

lscomp

Syntax:

```
lscomp
```

Description: This command lists all components.

Examples:

In the following example, the `lscomp` command will produce a complete list of components:

```
lscomp
```

lshost

Syntax:

```
lshost
```

Description: This command lists all BDNA Agent hosts.

Examples:

In the following example, the `lshost` command will list all BDNA Agent hosts, whether the Agent is running or not:

```
lshost
```

mkcomp

Syntax:

```
mkcomp [-Dmx=max memory] [-Dlike=existing component name] \
[-Ddc=deployment categories] [-Dhost=host name] [--active] \
<componentType> <componentName> <args>*
```

`<componentType>` can be one of the following: [SYSTEM_SERVICE, CLE, RULE, REMOTE_SERVICE, REPLAY, ADMIN, EXTERNAL_CLA, AGENDA, PERLCS, FE_REPLAY]

Description: This command creates a new component instance.

Examples:

In the following example, the `mkcomp` command will create a new Rule Engine component called RULE2 to run on host1.domain.com, with the maxMemory property set to 1.7GB and the deployment category set to “discovery”:

```
mkcomp -Dmx=1.7G -Ddc=discovery -Dhost=host1.domain.com RULE RULE2
```

In this example, a second PerlCS is created on a separate host with the same settings as the PerlCS on the first host:

```
mkcomp -Dlike=PerlCS1 -Dhost=host2.domain.com PERLCS PerlCS2
```

mkhost

Syntax:

```
mkhost <hostname>
```

When creating a new host, ensure that the value of the `<hostname>` argument provided to the command matches the value of the property `bdna.mbus.uuid` in the `conf/mbus.properties` file located on the remote host Alternatively, it is possible to execute the following command on a remote system to determine the value to supply to the `mkhost` command:

```
sh runjava.sh com.bdna.control.agent.AgentIdentity
```

The output returned will be as follows:

```
BDNA: Host name is host1.domain.com
BDNA: Full name is host1.domain.com
```

Description: This command adds a new BDNA Agent host to the system.

Examples:

In the following example, the `mkhost` command will add host1.domain.com to the BDNA system:

```
mkhost hsost1.domain.com
```

mkipspace

Syntax:

```
mkipspace <ipSpaceName> <ipSpaceDescription>
```

Description: `mkipspace` is used to create a new IPSpace. An IPSpace is used to disambiguate machines that have the same IP address but are in different network segments. It is necessary to create additional IPSpaces when the FactBase contains multiple private networks that use the same IP addresses. For example, if two different collection stores use the subnet 10.0.0.0/24, but each collection store represents a different private network, each Collection Store should be placed in a separate IPSpace. Note that a default IPSpace called Global IPSpace is always available in the FactBase. If the user wants to create additional IPSpaces, they can use the `mkipspace` command.

Example:

```
mkipspace mvIPSpace "IP Space for Mountain View"
```

module

Syntax:

```
module [-i] [-r] <filename>
```

or

```
module -d <module name> <function> \ <componenttype> <endpoint> <pattern>*
```

Description: This command imports, re-imports, or replaces a module for the Rule Engine.

Examples:

In the following example, the `module` command will import a new module `/tmp/myModule.xml` into the BDNA system:

```
module –i /tmp/myModule.xml
```

In the following example, the `module` command will re-import module `/tmp/myModule.xml`:

```
module –r /tmp/myModule.xml
```

Note: The `–r` option will not work unless `myModule.xml` has already been imported.

partition

Syntax:

```
partition [-lp] [-la] [-c] [-r] [-a] [-d]
```

List all active partitions:

```
partition -lp
```

List all active attachments:

```
partition -la
```

Create a new partition:

```
partition -c <partition name> <root element full name>
```

Remove a partition:

```
partition -r <partition name>
```

Attach a partition to a network group:

```
partition -a <network group name> <partition name>
```

Detach a partition from a network group:

```
partition -d <network group name> <partition name>
```

Description: The `partition` command manages the partitions and network groups.

Caution: The `partition` command is primarily used for troubleshooting problems within the BDNA system and should only be used under the supervision of BDNA Customer Support.

Examples:

None. BDNA recommends that you use this command only when directed to do so by BDNA Customer Support.

property

Syntax:

```
property -l <property>
property -m <property> <value>
```

`<property>` is of the form `[instanceName:]pattern` where the `instanceName` is optional and is used to avoid ambiguity between multiple instances and `pattern` is any string which is used to do a substring match on the property name in the database.

Description: This command lists or modifies the component properties.

Examples:

In the following example, the `property` command will display the componentTimeout property of all components:

```
property -l componentTimeout
```

In the following example, the `property` command will update the LingerTimeout property value of CLE2 component to 1:

```
property -m CLE2:LingerTimeout 1
```

purge

Note: While this command is available in BDNA Discover, if you type in `help` from bdna.sh, it doesn’t appear in the command list. Nevertheless, the command does work.

Syntax:

```
purge
```

Description: This command purges the derived system states in preparation for export. Note that after the `purge` command is executed, the BDNA repository cannot be used for data discovery anymore.

Examples:

In the following example, the `purge` command will purge unnecessary data from the system:

```
purge
```

rebalance

Note: While this command is available in BDNA Discover, if you type in `help` from bdna.sh, it doesn't appear in the command list. Nevertheless, the command does work.

Syntax:

```
rebalance -DmaxHosts=<max hosts per component> <from component regex> \
<to component regex> <max run time in second> -y
```

Description: Rebalance the partition among a collection of components. This command can result in a large amount of back-end processing, which can take a long time.

Caution: The `rebalance` command should only be used under the supervision of BDNA Customer Support.

reset

Note: While this command is available in BDNA Discover, if you type in `help` from bdna.sh, it doesn't appear in the command list. Nevertheless, the command does work.

Syntax:

```
reset --all
```

```
reset --rule <rule name>
```

```
reset --group <network group name>
```

Description: This command resets the states of the Rule Engine(s) to clean up corrupted Rule Engine states. This command can result in a large amount of back-end processing which can take a long time.

Caution: The `reset` command should only be used under the supervision of BDNA Customer Support.

Examples:

In the following example, the `reset` command will force the Rule Engine to be reprocessed:

```
reset –-all
```

In the following example, the `reset` command will force only RULE1 to be reprocessed:

```
reset –-rule RULE1
```

In the following example, the `reset` command will force only group AUTOGRP26 to be reprocessed:

```
reset --group AUTOGRP26
```

rmcomp

Syntax:

```
rmcomp <componentName>
```

Description: This command removes a component from the BDNA system. Typically, only components which will never be used or which were created in error should be removed. Otherwise, the component should be disabled by stopping it and setting “active” to false (refer to “updcomp” on page 243).

Examples:

In the following example, the `rmcomp` command will remove the RULE2 component from the BDNA system:

```
rmcomp RULE2
```

Note that the component can only be removed if it is in the “Init” state (refer to “lscomp” on page 235).

rmhost

Syntax:

```
rmhost <agent>
```

Description: This command removes an existing BDNA Agent host from the BDNA system.

Examples:

In the following example, the `rmhost` command will remove host1.domain.com from the BDNA system:

```
rmhost host1.domain.com
```

Note: A host can only be removed if no components are assigned to it. Before removing a host, all components assigned to it must either be reassigned (refer to "updcomp" on page 243) to another host or removed (refer to "rmcomp" on page 239) from the system.

rules

Syntax:

```
rules
```

Description: This command shows the status of rule execution. The following columns are found in the `rules` command output:

Network Group: the name of the network group

Component: the Rule Engine instance name

Active: indicates whether the network group is currently active or inactive

Splits: if a Network Group has gotten too big for its assigned Rule Engine memory footprint, shows the number of times it has split to fit in the memory footprint

Priority: a dynamic number assigned to each network group that roughly indicates the order the owner Rule Engine will use to work its assigned network groups

Rules Fired: the total number of rules fired against the network group

Last Scheduled: indicates the last time that the network group was active

Idle: indicates how long the network group has been idle

Examples:

In the following example, the `rules` command will display the status of rule execution:

```
rules
```

sanityCheck

Syntax:

```
sanityCheck [--verbose, -v] [--input, -i \ <source file path>]
```

Description: This command checks the integrity of the BDNA repository.

Examples:

In the following example, the `sanityCheck` command will run integrity checking against the BDNA repository:

```
sanityCheck
```

In the following example, the `sanityCheck` command will run a custom check list file:

```
sanityCheck –i /tmp/myCheckList.xml
```

scanStatus

Syntax:

```
scanStatus
```

Description: This command shows the status of a scan. `ScanStatus` displays the time windows during which different parts of the Collection Engine are not processing any jobs. It is divided into three sections:

- Scan Task Status

  Deals with collections Scan Tasks. For each active Scan Task it shows Collection idle, start, and end times; Rule idle, start, and end times; and the 'Task Idle'—the intersection of Collection and Rule idles times.

- Background Activity Status

  Shows idle, start, and end times of background deduping activities.

- Overall Status

  The intersection of all the above times. The Collection Engine cannot be considered idle unless there is a non-empty intersection of all the above times reflected in Overall Status.

Examples:

In the following example, the `scanStatus` command displays the overall status of the scan:

```
scanStatus
```

shutdown

Syntax:

```
shutdown [--kill, -k] [-Dtimeout=<timeString>]
```

Description: This command stops all components and shuts down the BDNA system. The `Dtimeout` property can be used to destroy any components that have not exited within the time indicated. The kill option is equivalent to timeout=0.

Valid examples of the timeString (to indicate 5 minutes) are 5m, 300, and 300s.

Examples:

In the following example, the `shutdown` command will stop all components gracefully and then shut down the BDNA system:

```
shutdown
```

In the following example, the `shutdown` command will force all components to terminate and then shut down the BDNA system:

```
shutdown -–kill
```

In the following example, the `shutdown` command will attempt to shut down the BDNA system gracefully. The command will force the components to terminate if it fails to stop the components in five minutes:

```
shutdown –Dtimeout=5m
```

start

Syntax:

```
start <component>
```

Description: This command starts a component. This command can only be used when the system is already in the Running state (refer to “startup” on page 242).

Examples:

In the following example, the `start` command will start up the RS component:

```
start RS
```

In the following example, the `start` command will start up the RULE1 component:

```
start RULE1
```

startup

Syntax:

```
startup [--init, -i] [--yes, -y] <componentNames>
```

Description: This command starts up the BDNA system. If the `init` parameter is specified, it re-initializes the BDNA repository. If no components are specified, all the components where the active flag is set to true will be started. Use the `lscomp` command to check the active flag of components.

Examples:

In the following example, the `startup` command will start up all components:

```
startup
```

In the following examples, the `startup` command will initialize the BDNA repository and then start up the components:

```
startup –i
```

```
startup –i –y
```

In the following example, the `startup` command will start up only the RS and Admin components:

```
startup RS Admin
```

status

Syntax:

```
status <component>
```

Description: This command shows the status of a specified component.

Examples:

In the following example, the `status` command will return the status of the RS component:

```
status RS
```

In the following example, the `status` command will return the status of the Agenda Manager component:

```
status agendaManager
```

stop

Syntax:

```
stop [--kill, -k] <componentName>
```

Description: This command stops a component.

Examples:

In the following example, the `stop` command will stop the RULE0 component:

```
stop RULE0
```

In the following example, the `stop` command will attempt to stop the RULE0 component. If the `stop` command fails to stop the RULE0 component, then it will kill the component:

```
stop –k RULE0
```

Note that the `kill` (`--kill`, `-k`) option should only be used when all other attempts to stop the component have failed. Killing a component could result in a corrupt state for that component.

updcomp

Syntax:

```
updcomp -Dproperty=value <componenName>
```

`Dproperty` must be one of the following: [`debugPort`, `active`, `mx`, `host`]

Description: This command updates a component configuration in the BDNA system.

Examples:

In the following example, the `updcomp` command will enable debugging for a component on port 9020:

```
updcomp -DdebugPort=9020 CLE1
```

In the following example, the `updcomp` command will disable debugging for a component:

```
updcomp -DdebugPort=0 CLM1
```

In the following example, the `updcomp` command will assign 768 megabytes to agendaManager:

```
updcomp -Dmx=768M agendaManager
```

In the following example, the `updcomp` command will reassign RULE0 component to run on host1.domain.com:

```
updcomp -Dhost=host1.domain.com RULE0
```

updhost

Syntax:

```
updhost [-Dname=<new name>] [old Agent hostname
```

Description:

This command updates a host in the BDNA system. In the following example, the `updhost` command will rename host1.domain.com to host2.domain.com:

```
updhost -Dname=host2.domain.com host1.domain.com
```

The command requires that the host has no components assigned to it; otherwise, an exception will be raised.

```
updhost -Dname=host2.domain.com host1.domain.com
updhost -Dname=host2.domain.com host1.domain.com
Exception caught:
com.bdna.control.ComponentSet.cannotRemoveHostComponentDefined; PerlCS2,
```

BDNA Inventory Component Reference C

About this Appendix

This appendix discusses each component of the Analytics application from a high level. It also lists the most commonly used properties for each component.

Component Summary

The Analytics application consists of the following components:

- "Application Monitor" on page 245
- "Agent" on page 246
- "Message Bus" on page 246
- "Database Watcher" on page 246
- "Agenda Manager" on page 246
- "Collection Manager" on page 247
- "Collection Engine" on page 247
- "PerlCS" on page 249
- "bdnaPerlCS" on page 250
- "bdnaCLE" on page 250
- "WinCS" on page 250
- "Rules Engine" on page 251
- "" on page 251 System Service
- "Remote Services" on page 251

Application Monitor

The Application Monitor component is installed on the central host for the BDNA system. It is responsible for maintaining application startup, shutdown, and runtime. It monitors system processes and is responsible for the following tasks:

- Initializing all system processes
- Stopping all active system processes on system shutdown
- Controlling all processes associated with BDNA
- Identifying processes that are no longer responding or are absent, and then correcting the problem
- Cleaning up any system process that fails and releasing any resources used by that process
- Monitoring the database

When the Application Monitor detects that a component has stopped, it instructs the Agent running on that host to restart that component.

The Application Monitor is a dedicated process and does not stop running when the shutdown command is issued. It must be available at startup to start the other BDNA system components by means of the Agents.

Agent

Like the Application Monitor, an Agent is a dedicated process. An Agent is installed on all machines running BDNA components. Its function is to start and stop components when instructed to do so by the Application Monitor. Only one Agent may be run on each machine.

If the Application Monitor detects that a component on a remote machine has stopped, it instructs the Agent running on that machine to restart that component. Agents are the parent processes of all BDNA components, including the Application Monitor.

Message Bus

The Message Bus consists of a number of services published by the Application Monitor and serves as a communication channel. The Message Bus API provides a mechanism for BDNA components to communicate with one another. The Message Bus provides methods to register and unregister components and exchange real-time information between components. Each component, including the Agents and the Application Monitor, registers with the Message Bus on startup and then uses it to exchange control and status messages.

Database Watcher

The database watcher is a small process that runs in the Application Monitor and communicates with the BDNA repository. It also controls all of the dynamic properties.

Agenda Manager

The Agenda Manager (agendaManager) is a BDNA component that is responsible for scanning the database for resource attributes that need to be refreshed and then generating and scheduling data collection tasks. Tasks are generated based on predefined knowledge—for example, that operating systems are a type of software—as well as identification data obtained through the fingerprinting process. The tasks are placed in a task queue in the database for subsequent pickup by the Collection Manager, which in turn assigns the tasks to specified Collection Engines. Only one instance of Agenda Manager can be configured on the BDNA system.

Table 26: Most Commonly Used Properties for Agenda Manager

Property Name	Type	Default	Dynamic	Min.	Max.	Description
bdna.agenda.localHostMaxRsrc	number	150	true	N/A	N/A	Maximum number of localhost outbound connections for all defined PerlCSs. Set to 150 x the number of PerlCSs in the system.
bdna.control.hostName	string	N/A	true	N/A	N/A	The Agent host on which agendaManager is running.

Collection Manager

A Collection Manager (CLM) is a BDNA component that is part of the BDNA back end. A Collection Manager manages one or more Collection Engines. When polled by a Collection Engine, the Collection Manager places collection tasks, generated by the Agenda Manager, in a task queue. When tasks are completed, the Collection Engine passes the data to its assigned Collection Manager for post-processing. The data is then stored in the BDNA repository or other central database. A Collection Manager can manage multiple Collection Engines. Only one instance of Collection Manager can be configured on the BDNA system.

Table 27: Most Commonly Used Properties for Collection Manager

Property Name	Type	Default	Dynamic	Min.	Max.	Description
bdna.agenda.DBBulkSize	number	250	true	N/A	N/A	Number of bulk operations at a time against the database
bdna.control.hostName	string	N/A	true	N/A	N/A	The Agent host where the CLM is running
bdna.clm.port	number	8009	false	N/A	N/A	The port number configured for the CLM
bdna.clm.service	string	/axis/services/ CLMService	false	N/A	N/A	The SOAP service identification
bdna.clm.maxConnsToAHost	number	2	true	1	2147483647	The default maximum number of concurrent connections to a given remote host
bdna.clm.statusUpdateInterval	number	60	true	1	86400	The status update interval (in seconds)
bdna.clm.schemaLockTimeout	number	1200	true	1	2147483647	The maximum time to wait for a database lock (in seconds)
bdna.clm.logDBTaskCalls	boolean	false	true	N/A	N/A	The setting to enable DBTask logging
bdna.clm.logFullDataRows	boolean	false	true	N/A	N/A	The setting to log full data rows

Collection Engine

A Collection Engine (CLE) is a data collection process run by the assigned Collection Manager. Each Collection Engine is assigned to the Collection Manager. A Collection Manager can manage multiple Collection Engines. The Collection Engine gathers data from one or more discovery zones based on tasks received from the Collection

Manager. The Collection Engine passes the collected data back to its Collection Manager for post-processing. All information is then stored in the BDNA repository or other central database. One or more Collection Engines can be configured on the BDNA system.

Table 28: Most Commonly Used Properties for Collection Engine

Property Name	Type	Default	Dynamic	Min.	Max.	Description
bdna.cle.maxConcurrentTasks	number	150	true	1	2147483647	The maximum number of tasks to execute concurrently
bdna.cle.clmPollingInterval	number	10	true	1	86400	The number of seconds between calls to the CLM to report results and request new tasks
bdna.cle.clientTimeout	number	1200	true	1	86400	The timeout in seconds for all calls made to the CLM
bdna.cle.statusUpdateInterval	number	60	true	1	86400	The status update interval (in seconds)
bdna.cle.useProxy	boolean	false	false	N/A	N/A	The setting to enable HTTP proxy to connect to the CLM
bdna.cle.proxyHost	string	N/A	false	N/A	N/A	The HTTP proxy address
bdna.cle.proxyPort	number	8080	false	0	65535	The HTTP proxy port
bdna.cle.proxyUser	string	N/A	false	N/A	N/A	The HTTP proxy user name
bdna.cle.proxyPassword	string	N/A	false	N/A	N/A	The HTTP proxy password
bdna.cle.claServerPort	number	8027	true	0	65535	The listening port for CLA server (e.g. PerlCS/WinCS).
bdna.cle.claServerBindAddress	string	N/A	true	N/A	N/A	The CLA server bind address
bdna.cle.claAckTimeout	number	600	true	1	86400	The time to wait (in seconds) for a CLA to acknowledge a NON-TASK related message, e.g. IdentifyAck or KeepAliveAck

Table 28: Most Commonly Used Properties for Collection Engine (Continued)

Property Name	Type	Default	Dynamic	Min.	Max.	Description
bdna.cle.minStreamHandlerThreads	number	4	true	1	2147483647	The minimum number of stream handler worker threads to use
bdna.cle.maxStreamHandlerThreads	number	250	true	1	2147483647	The maximum number of stream handler worker threads - High water mark for stream handler thread pool
bdna.cle.streamHandlerKeepAlive	number	3600	true	N/A	N/A	The number of seconds a stream handler worker thread can sit idle before being killed
bdna.cla.traceCLAMessages	number	0	true	0	7	Trace CLA messages—controls logging of messages number on the connection between the CLE and CLA. Control word format: 0 = tracing disabled 1 = trace sent messages 2 = trace received messages 3 = trace both 5 = trace sent headers only 6 = trace receive headers only 7 = trace both headers only

PerlCS

PerlCS is a Collection Agent (CLA) component that performs all data discovery except Windows Levels 2 and 3. The PerlCS component connects to a CLE instance. One or more PerlCS components can be connected to a single CLE instance.

Table 29: Most Commonly Used Properties for PerlCS

Property Name	Type	Default	Dynamic	Min.	Max.	Description
bdna.perlcs.maxTasks	string	50	true	1	2147483647	The maximum concurrent tasks allowed for each PerlCS instance—for a large scale deployment, this value should be between 75 and 150

bdnaPerlCS

bdnaPerlCS is a CLA component that performs all deduping tasks.

bdnaCLE

bdnaCLE is a CLA component that manages all deduping tasks.

WinCS

Windows Collection Service (WinCS) performs only Windows Levels 2 and 3 data discovery. WinCS is a service that runs on a Windows-based server. WinCS connects to a CLE component. One or more WinCS services can be connected to a single CLE instance.

Usually WinCS is installed under `C:\Program Files\BDNA\WinCS`. The configuration file for WinCS is located at `C:\Program Files\BDNA\WinCS\WinCs.exe.config`.

Table 30: Most Commonly Used Properties for WinCS

Property Name	Type	Default	Dynamic	Min.	Max.	Description
cleHostName	string	N/A	false	N/A	N/A	The host name where CLE component is running.
maxConcurrentTasks	number	75	false	N/A	N/A	The maximum concurrent tasks allowed by WinCS service—If the WinCS server has 4GB of physical RAM, set this value to 75 for optimal performance. If the WinCS server has 2GB of RAM, set this value to 25. If the WinCS server has 1GB of RAM, set this value to 10.

The flag of tftpUNCPath is only used if UNC Path is not empty.

If tftpUNCPath is provided, any fingerprint that requires tftp will use the tftp executable from this UNC path. Access privilege is controlled by tftpPath_login and password. If tftpUNCPath option is used, a valid username and password must be provide. It is recommended to provide IP address as UNC path to save further effort of DNS lookup.

Example:

```
<add key="tftpUNCPath" value="\\10.10.13.61\unc"/>
<add key="tftpPath_login" value="normalizevm\administrator"/>
<add key="tftpPath_password" value="Simple.0"/>
```

Default value:

```
<add key="tftpUNCPath" value=""/>
<add key="tftpPath_login" value=""/>
<add key="tftpPath_password" value=""/>
```

Rules Engine

The Rules Engine (RULE) is responsible for identification of all resources on all hosts discovered on the system, including operating systems, applications, databases, and so forth. The Rules Engine analyzes collected data and matches it against sets of preconfigured templates known as fingerprints. A fingerprint is a set of rules stored in modules. When the BDNA system starts up, it loads the modules into the database. The Agenda Manager, as one of its predefined tasks, instructs the Rules Engine to scan the database for rules that need executing.

The Rules Engine comprises two parts:

- Truth Maintenance System (TMS)

  TMS monitors the repository for changes. When a change occurs that requires a new set of rules, TMS adjusts the required rules accordingly. For example, the rule might change to state that a resource should now be placed in a different location.

- Inference

  Inference runs rules according to the data collected. For example, if the following rule is in use:

  > IF there is a host named <variable>
  >
  > AND there is an open port whose number is 80
  >
  > AND the host and the port share the same address
  >
  > THEN declare the host “Apache” and place “Apache” under <location>

  and TMS sees that port 80 on a host is not firing, it will tell Inference to use another set of rules.

Table 31: Most Commonly Used Properties for Rules Engine

Property Name	Type	Default	Dynamic	Min.	Max.	Description
bdna.rule.maxNetworkGroupSize	number	4000	true	3000	N/A	The maximum number of IPs per rule network group

Remote Services

Remote Services is the BDNA component responsible for all interaction with interface components such as the Scan Administration and Inventory Analytic applications.

Table 32: Most Commonly Used Properties for Remote Services

Property Name	Type	Default	Dynamic	Description
bdna.rs.LDAP.alternateURL	string	N/A	true	The location of a secondary LDAP server, should the primary connection fail
bdna.rs.LDAP.authentication	string	N/A	true	The type of authentication to use
bdna.rs.LDAP.connectionURL	string	N/A	true	The location of the primary LDAP server. If the connection fails the system will attempt to connect using the alternate URL.
bdna.rs.LDAP.roleBase	string	N/A	true	The base directory entry for performing role searches. If not specified, the top-level element in the directory context will be used.
bdna.rs.LDAP.roleName	string	N/A	true	The name of the attribute that contains role names in the directory entries found by a role search. In addition the `userRoleName` property can be used to specify the name of an attribute, in the user's entry, containing additional role names. If roleName is not specified, a role search does not take place and roles are taken only from the user's entry.
bdna.rs.LDAP.roleSearch	string	N/A	true	The LDAP filter expression used for performing role searches, with {0} marking where to substitute the distinguished name (DN) of the user, and/or {1} to substitute the username. If not specified, a role search does not take place and roles are taken only from the attribute in the user's entry specified by the userRoleName property. Example: bdna.rs.LDAP.roleSearch=(&(uniqueMember ={0})(description=manager)
bdna.rs.LDAP.roleSubtree	boolean	false	true	Set to true to search the entire subtree of the element specified by the roleBase property for role entries associated with the user. The default value of false causes only the top level to be searched.
bdna.rs.LDAP.rootdn	string	N/A	true	A distinguished name that can access and look up information from the LDAP server. This is necessary when the LDAP server does not allow anonymous lookup and the user is not authenticated using the directory entry name.
bdna.rs.LDAP.rootpw	string	N/A	true	The credential associated with the root distinguished name (DN).

Table 32: Most Commonly Used Properties for Remote Services (Continued)

Property Name	Type	Default	Dynamic	Description
bdna.rs.LDAP.useSSL	boolean	false	true	Indicates whether to use SSL when making the connection. The user could also specify SSL connections via the connection URL (ldaps://server:636). The default value is false.
bdna.rs.LDAP.userBase	string	N/A	true	The distinguished name of the top-level entry to use when searching for the user entry. If not specified, the top-level element in the directory context will be used. When `userdn` is specified, the userBase will be appended to that value to form the user's Distinguished Name (DN) used for authentication.
bdna.rs.LDAP.userdn	string	N/A	true	The distinguished name (DN) attribute that identifies a user in the LDAP user directory (e.g., cn, uid, etc.). When this property is specified, the system will attempt to authenticate the user using the DN in the format userdn=username{,}userBase using the specified password. For example, with the following settings and parameters: - userBase: CN=Users,DC=bdnacorp,DC=com - userDN: CN - username: John Smith - password: goodday The system will attempt to bind to the user entry “CN=John Smith,CN=Users,DC=bdnacorp,DC=com” using the password “goodday”. The user is considered authenticated if the binding is successful. This property supersedes the userSearch and userSubtree parameters (i.e. the latter two will not be used when this is specified).

Table 32: Most Commonly Used Properties for Remote Services (Continued)

Property Name	Type	Default	Dynamic	Description
bdna.rs.LDAP.userRoleName	string	N/A	true	The name of an attribute in the user's directory entry containing zero or more values for the names of roles assigned to this user. In addition the roleName property can be used to specify the name of an attribute to be retrieved from individual role entries found by searching the directory. If userRoleName is not specified, all the roles for a user derive from the role search.
bdna.rs.LDAP.userSearch	string	N/A	true	The LDAP filter expression to use when searching for a user's directory entry, with {0} marking where the actual username should be inserted. Use this property (along with the userBase and userSubtree properties) to search the directory for the user's entry. Example: bdna.rs.LDAP.userSearch=(&(uid={0})(description=Manager))
bdna.rs.LDAP.userSubtree	boolean	false	true	Set to true to search the entire subtree of the element specified by the userBase property for the user's entry. The default value of false causes only the top level to be searched. Not used if using the userdn expression.

Completion-Code Reference D

About this Appendix

This appendix provides detailed information about BDNA Discover completion codes. Each collection task processed by BDNA has a completion code, indicating whether the task succeeded or failed. Table33, "BDNA Completion Codes," shows a complete list of completion codes:

Table 33: BDNA Completion Codes

Result Code	Completion Code Description	Explanation	Corrective Action
0	Task execution completed successfully	This completion code indicates that a connection or collection task was completely successful without any errors.	N/A
1	Invalid message size	This completion code indicates that a message exchanged between data collection components had an unexpected size.	Check the data collection component versions. Contact BDNA Technical Support.
2	Message type is not supported	This completion code indicates that a message exchanged between data collection components is obsolete.	Check the data collection component versions. Contact BDNA Technical Support.
3	Connection to remote host failed	This completion code indicates that a specific connection or collection script has failed to connect to the target system.	Check network connectivity to remote host by using Credential Checker.
4	Network read error	This completion code indicates that a network socket read error has occurred during the data collection.	Check network connectivity to remote host by using Credential Checker and retry collection.
5	Network write error	This completion code indicates that a network socket write error has occurred during the data collection.	Check network connectivity to remote host by using Credential Checker and retry collection.
6	Login to remote host failed	This completion code indicates that an access denied error was returned by the target host. This happens when given credential sets are invalid. This completion code applies to Windows collection only.	Check host credentials by using Credential Checker and retry collection.
7	Bad connection script	This completion code indicates that the system has failed to compile a given connection script.	Contact BDNA Technical Support.

Table 33: BDNA Completion Codes (Continued)

Result Code	Completion Code Description	Explanation	Corrective Action
8	Bad collection script	This completion code indicates that the system has failed to compile a given collection script.	Contact BDNA Technical Support.
9	Network read timeout	This completion code indicates that a data collection component has timed out waiting for a message.	Restart data collection component.
10	Process creation failed	This completion code indicates that a remote process creation has failed. Some collection scripts such as Oracle for Windows require an execution process to be created on the target system in order to collect data.	Check the credentials used. Make sure the credentials have administrative privileges to launch remote process.
11	Task execution exceeded the maximum allowed duration	This completion code indicates that a connection or collection task has not completed within a timeout period.	Contact BDNA Technical Support.
12	Script does not exist in Collection Agent cache	This completion code indicates that an internal script management error has occurred.	Restart CLE data collection component. Contact BDNA Technical Support if problem persists.
13	Parameter set does not exist in Collection Agent cache	This completion code indicates that an internal script management error has occurred.	Restart CLE data collection component. Contact BDNA Technical Support if problem persists.
14	Task execution canceled	This completion code indicates that a task was canceled by user action.	N/A
15	Connection to remote Collection Agent failed	This completion code indicates that the data collection component could not be contacted.	Check connectivity between data collection components.
16	Invalid script type code	This completion code indicates that an invalid component script has occurred.	Check data collection component versions. Contact BDNA Technical Support.
17	Invalid parameter set type code	This completion code indicates that an invalid collection parameter has occurred.	Check data collection component versions. Contact BDNA Technical Support.
18	Script name cannot be null	This completion code indicates that a collection script with invalid information has occurred.	Check BDNA installation and databases. Contact BDNA Technical Support.

Table 33: BDNA Completion Codes (Continued)

Result Code	Completion Code Description	Explanation	Corrective Action
19	Parameter set name cannot be null	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
20	Remote host connection name cannot be null	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
21	Task Id not found	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
22	Both collection script and parameter set names cannot be null	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
23	Both connection script and parameter set names cannot be null	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
24	Entity name cannot be null	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
25	Attribute set for task missing	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
26	Database timestamp cannot be zero	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
27	Local timestamp cannot be zero	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
28	Agent type name missing	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
29	Element type name missing	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
30	Element Id cannot be zero	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
31	Parameter set cannot be null	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.

Table 33: BDNA Completion Codes (Continued)

Result Code	Completion Code Description	Explanation	Corrective Action
32	Collection Engine Id cannot be zero	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
33	Task Id cannot be zero	This completion code indicates that a collection script with invalid information has occurred.	Contact BDNA Technical Support.
34	Maximum concurrent connection exceeded	This completion code indicates that a maximum concurrent connection has been exceeded.	Contact BDNA Technical Support.
35	Remote host connection does not exist	This completion code indicates that an internal CLA error has occurred.	Restart CLA/CLE.
36	Task processing resulted in an exception	This completion code indicates that a connection or collection task has failed with a known exception error.	Contact BDNA Technical Support.
37	Component is shutting down	This completion code indicates that active collections did not finish completely because of collection system or component shutting down during active collection. Collections will restart next time the system is started.	Contact BDNA Technical Support.
38	Internal processing error	This completion code indicates that a connection or collection task has failed due to a system level error. One example would be that the collection Agent ran out of system memory and suffered an unexpected crash.	Check BDNA system resources. Contact BDNA Technical Support.
39	Message type received does not match the expected message type	This completion code indicates that a data collection component received an unexpected message type.	Check data collection component versions. Contact BDNA Technical Support.
40	List or table entry count is not correct	This completion code indicates that an internal CLA error has occurred.	Contact BDNA Technical Support.
41	Entity name is not recognized	This completion code indicates that an internal CLA error has occurred.	Contact BDNA Technical Support.
42	Remote Collection Agent failed to initialize	This completion code indicates that a CLA data collection component failed to initialize.	Check BDNA environment and logs. Contact BDNA Technical Support.

Table 33: BDNA Completion Codes (Continued)

Result Code	Completion Code Description	Explanation	Corrective Action
43	File I/O error	This completion code indicates that a file I/O error occurred while processing data collection.	Check BDNA and remote host environment.
44	Operation timed out	Superseded by codes “11” (Task Expired) and “45” (Task exceeded deadline).	Refer to related codes “11” and “45”
45	Task execution could not be completed before the shift ended	This completion code indicates that the scheduled collection did not start before the Scan Task's shift ended.	No remedial action is needed. Collection will resume at next available shift.
46	A Collection Agent of the correct type is not registered and cannot be loaded on demand	This completion code indicates that the collection task requires a Collection Agent (CLA) of a type not currently configured in the BDNA system.	Check BDNA system configuration.
47	Current collection task state is invalid	This completion code indicates that the system found a scheduled collection for an inactive object. The collection will be skipped.	None required.
48	Unable to decrypt collection task arguments	This completion code indicates that the decryption keys to decode a data collection message could not be used.	Check BDNA encryption configuration. Contact BDNA Technical Support.
49	Deduping failed	This completion code indicates that a deduping task has failed.	Check bdnaCLE component and tasks.
50	Bulk insert of collected data failed	This completion code indicates that the data collected was not successfully inserted into the BDNA database.	Check BDNA database environment. Repeat collection tasks.
51	SOAP attachment processing failed	This completion code indicates that the collected data packet could not be processed.	Check BDNA data component environment. Contact BDNA Technical Support.
52	Exec process command line missing	This completion code indicates that a remote process command has failed due to invalid command line parameters.	Contact BDNA Technical Support.
53	A connection with the specified name already exists	This completion code indicates that more than one connection was created for the collection task.	Contact BDNA Technical Support.

Table 33: BDNA Completion Codes (Continued)

Result Code	Completion Code Description	Explanation	Corrective Action
54	The connection object passed to the collection script is cannot be null	This completion code indicates that the connection object sent to the collection script is null.	Contact BDNA Technical Support.
55	TFTP file transfer between the remote host and the Collection Server failed	This completion code indicates that a collection script has failed the remote file transfer command. Some collection scripts such as VMWare for Windows and Oracle for Windows require additional result sets to be transferred back from the target system. This completion code applies to Windows collection only.	In order to successfully perform Oracle Level 2 and Level 3 collection tasks (and other collection tasks that use TFTP protocol such as VMWare Virtualization collection) when Windows Firewall is enabled, UDP Port 69 must be opened (see Exceptions tab on Windows Firewall settings), or Windows Firewall should be disabled on the WinCS server.
56	WMI connection to the remote host has failed	This completion code indicates that a WMI connection could not be established between the CLA server and the target host.	An internal WMI error might have occurred. Contact BDNA Technical Support.
57	A required script parameter is not present in the script parameter set	This completion code indicates that the script parameter information required is missing.	Contact BDNA Technical Support.
58	A WMI query to the remote host timed out	This completion code indicates that a specific WMI query has timed out (default = 30 seconds) with a collection script. This completion code applies to Windows collection only.	Increase the wmiMethodTimeout property on the WinCS server and restart the data collection.
59	A database error has occurred while processing tasks. Check error logs.	DB error during credential test.	Check DB and try credential test again.
60	Task identifier parameter is not valid	Credential task test not found in database.	Try credential test again.

Configuration-File Reference E

About this Appendix

This appendix lists the individual BDNA configuration files, along with descriptions of their contents and notes.

Configuration File Directory

The majority of BDNA configuration files are stored under the `$BDNA_HOME/conf` directory. Additional configuration files are contained in sub-directories as explained in "Configuration File Subdirectories" on page 268.

adminCommand.properties

Description: This configuration file contains linkages between some of the administrative BDNA Shell commands and the actual Java commands.

Editable by user: N

Comments: Do not modify this file.

adminQuery.properties

Description: This configuration file is an internal file used by the BDNA Shell.

Editable by user: N

Comments: Do not modify this file.

agenda.properties

Description: This property file is used for standalone debugging for the Agenda Manager component.

Editable by user: N

Comments: Do not modify this file.

agent.properties

Description: This file gets generated when the Application Agent is being started on a BDNA host. The command startagent.sh reads information from `/proc/meminfo` such as physical memory, virtual memory, etc. of the host machine, and writes it to `agent.properties`. In the BDNA shell, the lshost command displays the values from this property file.

Editable by user: N

Comments: Do not modify this file.

app.dtd

Description: This is a document type definition file for the BDNA front end applications.

Editable by user: N

Comments: Do not modify this file.

appLogging.properties

Description: This is the logging configuration file for the Analytics application.

Editable by user: N

Comments: Do not modify this file.

archiver.properties

Description: This property file is used for standalone debugging for the Archiver component.

Editable by user: N

Comments: Do not modify this file.

bdna.dtd

Description: This is a document type definition file for the Analytics application.

Editable by user: N

Comments: Do not modify this file.

bdna.Linux.properties

Description: This is a system property file for the Analytics application. This file is used only when the Analytics application runs on a Linux server.

Editable by user: N

Comments: Do not modify this file.

cle.xml

Description: This is a XML file containing a SQL query to check the CLE status. This file is obsolete.

Editable by user: N

Comments: Do not modify this file.

clm.properties

Description: This property file is used for standalone debugging for CLM component.

Editable by user: N

Comments: Do not modify this file.

CLMServiceDeploy.wsdd

Description: This is the web service deployment file to deploy handlers, chains and services to the Axis engine.

Editable by user: N

Comments: Do not modify this file.

CLMServiceUndeploy.wsdd

Description: This is the web service deployment file to release handlers, chains and services from the Axis engine.

Editable by user: N

Comments: Do not modify this file.

component.properties

Description: This property file is used for standalone debugging for all component.

Editable by user: N

Comments: Do not modify this file.

ConfigureMessages.properties

Description: This property file contains the predefined message strings that will be used in the `$BDNA_HOME/bin/configure.sh utility`.

Editable by user: N

Comments: Do not modify this file.

connection.properties

Description: This property file contains the database connection parameters which are necessary for the BDNA components to establish a connection to the BDNA repository.

Editable by user: Y

Comments: An alternative for setting connection properties is the `resetpass.sh` script found in $BDNA_HOME/bin. The script sets the following properties: bdna.dbURL, bdna.dbServer, bdna.dbUser, and bdna.dbPassword.

Note: You must use `resetpass.sh` if you are using an encrypted password.

Modify the following four properties during the BDNA installation.

Table 34: Required Properties in connection.properties file to Enable BDNA to Connect to the Database

Property Name	Description
bdna.dbURL	This property is the full database connection string used by the Oracle thin driver, which is used extensively by the BDNA components. This property must use the following format: jdbc:oracle:thin:@HOST_NAME:ORACLE_LISTENER_PORT:ORACLE_SID Example: jdbc:oracle:thin:@pilotlin:1521:ora10g
bdna.dbServer	This property is the DB server TNS name. It is used for executing SQL*Plus. The entry specified in this field must have already been defined in `$ORACLE_HOME/network/admin/tnsnames.ora`.
bdna.dbUser	This property is the user name for the BDNA repository.
bdna.dbPassword	This property is the password for the BDNA repository. (If you are using password encryption, you must use the `resetpass.sh` script found in $BDNA_HOME/bin)

content.properties

Description: This property file is used for standalone debugging of the content modules.

Editable by user: N

Comments: Do not modify this file.

ExternalMessages.properties

Description: This property file contains the predefined message strings that will be used by the External Data Collection service.

Editable by user: N

Comments: Do not modify this file.

glue-config.xml

Description: This is the configuration file for the Glue engine.

Editable by user: N

Comments: Do not modify this file.

j4bundle.properties

Description: This property file is used for standalone debugging for the J4 fingerprint generator.

Editable by user: N

Comments: Do not modify this file.

.keystore

Description: This is a keystore file that enables the Analytics application to support SSL connections.

Editable by user: N

Comments: Do not modify this file.

label.properties

Description: This file contains the translation of element names which is required for multi-lingual platform support.

Editable by user: N

Comments: Do not modify this file.

license.bin

Description: This is the license key validation engine.

Editable by user: N

Comments: Do not modify or remove this file.

license.txt

Description: This file contains license keys that enable individual BDNA applications.

Editable by user: Y

Comments: Add license keys to this file to enable the Java Swing-based applications, such as Administration, Scan Administration, and Analytics.

logging.properties

Description: This file contains the file logging level for BDNA components. The log files are stored under the `$BDNA_HOME/logs` directory.

Editable by user: N

Comments: Note that this file doesn't control the logging level for the database. This file should not be modified, because enabling a higher level of debug logging may cause a negative impact on the overall BDNA application performance. The logging levels for a logger can be enabled or disabled at run time. This file contains the default logging level for different loggers.

mbus.properties

Description: This file stores information about the Message Bus location.

Editable by user: Y

Table 35: Properties in the mbus.properties File

Property Name	Description
bdna.mbus.url	This property is the URL of the Message Bus. If BDNA is installed using a single server configuration, then set this property to http://localhost:8006/glue. If BDNA is installed using a distributed server configuration, then set this property to http://<IP_address_of_central_ BDNA_host>:8006/glue.
bdna.mbus.urn	This property cannot be changed. bdna.mbus.urn must be set to urn:MBus.
bdna.mbus.uuid	When BDNA is forced to go through firewalls or across networks, use host UUID. Assign each host a unique UUID (any string) and ensure that the property bdna.mbus.uuid in `mbus.properties` on the host matches the UUID in the host table. The BDNA Shell supports host UUIDs with the `mkhost` command. Example: bdna> mkhost -Duuid=long_unique_string host1.domain.com
bdna.mbus.useProxy	Set this property to false if the Message Bus does not connect to a proxy server. Set this property to true if the Message Bus does connect to a proxy server. Note that when this property is set to true, then all of the following properties must be provided. Default is false.
bdna.mbus.proxyHost	This property is the name of the proxy server to which the Message Bus will connect. This property will be used only when bdna.mbus.useProxy is set to true. Default is blank.
bdna.mbus.proxyPort	This property is the port number of the proxy server to which the Message Bus will connect. This property will be used only when bdna.mbus.useProxy is set to true. Default is blank.
bdna.mbus.proxyUser	This property is the username which will be used to authenticate the proxy server. This property will be used only when bdna.mbus.useProxy is set to true. Default is blank.
bdna.mbus.proxyPassword	This property is a password that is used to authenticate the proxy server. This property will be used only when bdna.mbus.useProxy is set to true. Default is blank.

MessagesBundle.properties

Description: This file contains a set of pre-defined strings for the BDNA applications.

Editable by user: N

Comments: Do not modify this file. This is being used for internationalization purpose. The default MessagesBundle.properties is for country code US and language code EN.

nm.properties

Description: This property file is used for standalone debugging for the Notification Manager component.

Editable by user: N

Comments: Do not modify this file.

notifierLoadTest.properties

Description: This property file is used for standalone debugging for the Notifier Load Test component. This file is used for internal testing.

Editable by user: N

Comments: Do not modify this file.

precomputeQueries.xml

Description: This file is used to pre-compute the queries for the Query Engine. This file is obsolete.

Editable by user: N

Comments: Do not modify this file.

properties.xml

Description: This file contains all of the BDNA component properties. The properties defined in this file will be populated into the BDNA repository during the initialization process.

Editable by user: N

Comments: Do not modify this file.

replay.config

Description: This file is used for internal testing.

Editable by user: N

Comments: Do not modify this file.

ReplayScript.perl

Description: This file is used for internal testing.

Editable by user: N

Comments: Do not modify this file.

ReplayScript.wincs

Description: This file is used for internal testing.

Editable by user: N

Comments: Do not modify this file.

rs.properties

Description: This property file is used for standalone debugging for the RS component.

Editable by user: N

Comments: Do not modify this file.

safepw

Description: This file stores the username and password which are accessible to the safe mode of the Administration UI.

Editable by user: Y

Comments: Modify the username and password for the safe mode login.

sanity.xml

Description: This file is a set of SQL queries that validate the integrity of the BDNA repository. This file is used by the `sanityCheck` command under the BDNA Shell.

Editable by user: N

Comments: Do not modify this file.

scmap.conf

Description: This is the Spell Checker configuration file, which maps special characters and numbers to English letters, so that a spell check can work correctly.

Editable by user: N

Comments: Do not modify this file.

trimContainmentType.conf

Description: This file contains a list of BDNA element types that will be ignored during the containment build process. The containment build process is handled by the Query Engine. This file is obsolete.

Editable by user: N

Comments: Do not modify this file.

xsiCommand.properties

Description: This configuration file contains linkages between the XSI command and the actual Java command. XSI commands are used for managing CMDB data sets.

Editable by user: N

Comments: Do not modify this file.

Configuration File Subdirectories

There are also three subdirectories under the `$BDNA_HOME/conf` directory. The subdirectories are listed below, along with descriptions of their contents and notes:

$BDNA_HOME/conf/components

Description: This directory contains the JVM configuration template files for each component.

Editable by user: N

Comments: Do not modify any files under this directory.

$BDNA_HOME/conf/jiffy

Description: This directory contains the jiffy fingerprint template files.

Editable by user: N

Comments: Do not modify any files under this directory

$BDNA_HOME/conf/sql

Description: The files under this directory are obsolete.

Editable by user: N

Comments: Do not modify any files under this directory.

Viewing East-Asian Languages F

About this Appendix

This appendix provides information about viewing east-Asian Languages on Windows (Double-Byte Character Sets).

BDNA Discover supports data collection with double-byte character sets (DBCS) such as Chinese and Japanese. This functionality is enabled through the Regional and Language Options of the Windows Control Panel.

To view double-byte character sets on Windows systems:

Tip: You will need your Windows Installation CD to complete this task.

1. On the Windows client, click the Start button.
2. Expand the Settings menu option and select Control Panel.
3. Double-click Regional and Language Options.
4. Open the Languages tab.
5. Check the box next to Install files for East Asian languages.
6. Open the Advanced tab.
7. Under Code page conversion tables, check the box next to the appropriate language. Click OK.
8. Windows displays a prompt to insert a Windows Service Pack CD.
9. Insert the correct CD and follow the prompts.

For instructions on DBCS installations on BDNA Discover servers, refer to “Double-Byte Character Set Installation for East Asian Languages” on page 19.

Viewing Double-Byte Characters in Microsoft Excel

All double-byte characters exported from the Report Viewer in BDNA Discover are encoded in UTF-8. In order to view the double-byte characters in Microsoft Excel, you need to set the file encoding to UTF-8. Here is the procedure to accomplish that in Excel 2007:

1. Launch Microosft Excel.
2. Go to Data menu.
3. Select Get External Data -> From Text.
4. In Import Text File dialog, select the *.tab or *.csv file that you exported from the Report viewer in BDNA Discover.
5. In Text Import Wizard - Step 1 of 3 dialog, select Delimited. Set File origin to “65001: Unicode (UTF-8)”. Click Next.
6. In Text Import Wizard - Step 2 of 3 dialog, under Delimiters, check Tab if the file was saved in Tab format, or check Comma if the file was saved in CSV format. Click Next.

7. In Text Import Wizard - Step 3 of 3 dialog, accept the default settings. Click Finish.

Blackberry Server L3 Fingerprint Report Columns G

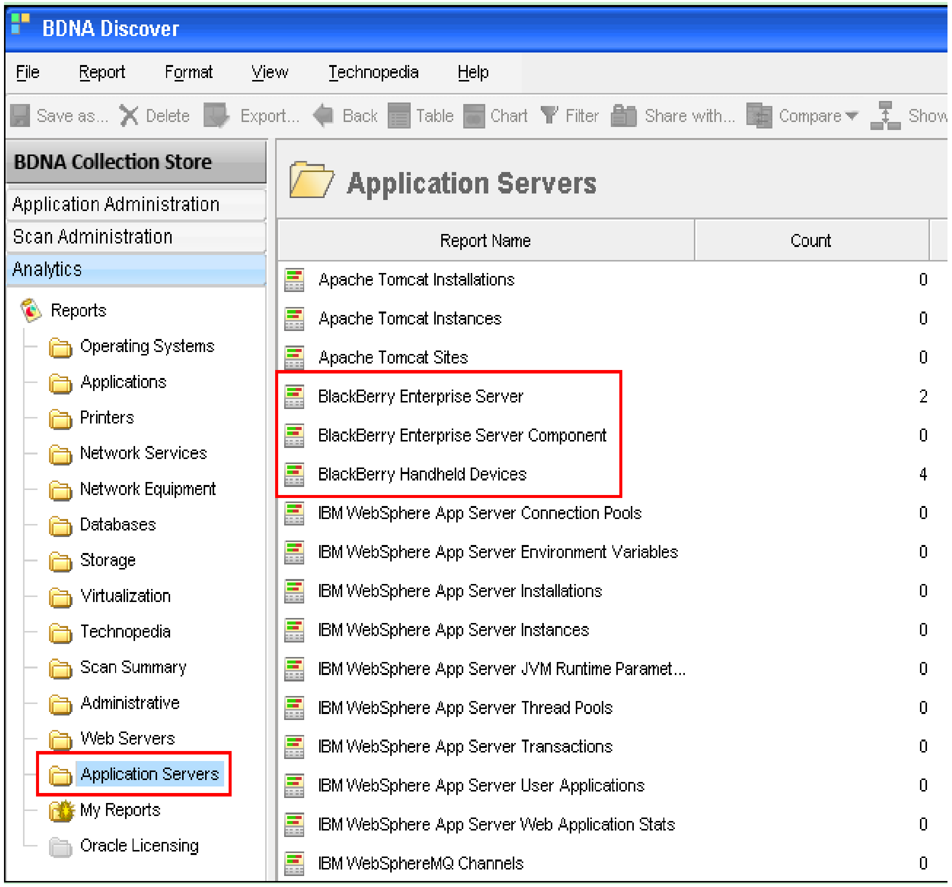
About this Appendix

This appendix provides information about the reports that BDNA Discover provides for the Blackberry Server L3 fingerprint, including:

- BlackBerry Enterprise Server
- BlackBerry Enterprise Server Component
- BlackBerry Handheld Devices

These reports are located in the Analytics section of BDNA Discover, under the Application Servers folder.

Figure 64: Blackberry Reports



Report Name	Count
Apache Tomcat Installations	0
Apache Tomcat Instances	0
Apache Tomcat Sites	0
BlackBerry Enterprise Server	2
BlackBerry Enterprise Server Component	0
BlackBerry Handheld Devices	4
IBM WebSphere App Server Connection Pools	0
IBM WebSphere App Server Environment Variables	0
IBM WebSphere App Server Installations	0
IBM WebSphere App Server Instances	0
IBM WebSphere App Server JVM Runtime Paramet...	0
IBM WebSphere App Server Thread Pools	0
IBM WebSphere App Server Transactions	0
IBM WebSphere App Server User Applications	0
IBM WebSphere App Server Web Application Stats	0
IBM WebSphereMQ Channels	0

Blackberry Enterprise Server Report Columns

The Blackberry Enterprise Server Report provides values in the following columns:

- Host name
- OS Type
- BlackBerry Enterprise Server Name
- # of BlackBerry Handheld Devices
- # of BlackBerry Enterprise Server Components
- Database Name
- Server Name
- Database Server Host Name
- TCP Port
- Data Directory
- Number of CPUs
- # of Networks
- Networks
- # of IPs (Scanned)
- IP(s) (Scanned)

Note: You can open Blackberry Server Components Report and Blackberry Handheld Devices Report from within the Blackberry Enterprise Server Report.

To Open the Blackberry Server Components Report: Click a value within the # of Blackberry Enterprise Server Components column.

To Open the Blackberry Handheld Devices Report: Click a value within the # of Blackberry Handheld Devices column.

Blackberry Enterprise Server Component Report Columns

The Blackberry Enterprise Server Component Report provides values in the following columns:

- Host name
- OS Type
- BlackBerry Enterprise Server Component Name
- Startup Mode
- State
- Number of CPUs
- # of Networks
- Networks
- # of IPs (Scanned)
- IP(s) (Scanned)

Blackberry Handheld Devices Report Columns

The Blackberry Handheld Devices Report provides values in the following columns:

- Host name
- OS Type
- BlackBerry Handheld Devices Name
- BlackBerry Version
- Network Carrier
- Model Name
- Platform Version
- Device Memory
- Device Serial Number (IMEI)
- Device Pin Number
- Phone Number
- User Name
- Mailbox Address
- IT Policy Name
- IT Policy Time
- Security Password
- Auto Signature
- Enable Calendar Mode
- Last Message Forward Time

- Last Message Sent Time
- Last Contact Time
- Last Update Time
- Number of Message(s) Forwarded
- Number of Message(s) Sent
- Number of Message(s) Pending
- Number of Message(s) Filtered
- Number of Message(s) Expired
- Number of Message(s) Failed
- Networks
- Number of CPUs
- # of Networks
- # of IPs (Scanned)
- IP(s) (Scanned)

Deploying Enterprise Best Practices H

About this Appendix

BDNA's architecture creates opportunities for graceful scaling, so that discovery and analysis can be performed using a single laptop or several server clusters. Of course, configuration parameters will vary based on the system scale. It is also the case that operational procedures should vary based on the system scale, since there are vast differences in complexity and manageability depending upon the system configuration being used.

When operating BDNA in an enterprise scale deployment, it is important to consider ways to optimize pre-discovery and post-discovery operations. This appendix documents some best practices for deploying BDNA for the enterprise.

Clearing Log Files and File Store

BDNA logs a good deal of information to log files during discovery. While these log files can contain valuable information for troubleshooting, the utility of these files is reduced when they are cluttered with the data from previous scans. For this reason, it is best to make sure BDNA's log directory is empty before beginning a scan.

To delete the BDNA log files, simply use the Linux "rm" command:

```
rm $BDNA_HOME/logs/*.*
```

BDNA also utilizes a file system store to optimize database performance. These files are not used for debugging, but nonetheless file stores from previous schemas should be deleted prior to starting a new scan.

The file store is saved inside the directory `$BDNA_HOME/file_store/<schema_name>`, where `<schema_name>` is the name of the Oracle schema used in the scan.

Before starting a new scan, use the Linux command rm for every directory under `$BDNA_HOME/file_store/`:

```
rm –Rf $BDNA_HOME/file_store/<schema_name>
```

Post-Discovery Operations

After performing a data discovery, it is important to back up the repository using the Oracle export command.

Exporting Schemas

A comprehensive backup procedure for BDNA includes using the Oracle export command to archive BDNA repositories.

The syntax for the Oracle export command is as follows:

```
exp <USERNAME>@<SERVER> file=<FILENAME> direct=y
```

The USERNAME and SERVER values can be obtained by executing the Linux command:

```
grep -i -e "^bdna.dbUser" -e "^bdna.dbServer" \
$BDNA_HOME/conf/connection.properties
```

The FILENAME value is user defined and can include paths on the file system.

Caution: BDNA repositories are often several gigabytes in size. It is important to confirm that there is adequate disk space available on the file system before exporting.

The resulting file contains the Oracle schema and it can be restored for subsequent discovery operations.

Tip: Oracle schemas compress well using the Linux standard bzip2 command. Type bzip2 <FILENAME> to compress the schema using bzip2. The command to decompress files that have been compressed using bzip2 is simply bunzip2.

Archiving Log Files and File Store

As discussed above, BDNA uses log files and file stores during discovery. Archiving the log files and the file stores should be set up to occur automatically post-discovery, after the archiving of the schema.

How long to keep these archive files should be determined by how long their schema will be in use. The file store archive can be safely deleted after it has been determined that no further scans using the associated repository or schema will occur.

Linux component server log files are stored in the directory `$BDNA_HOME/logs/`. Log files for Windows component servers are stored in the directory `%PROGRAMFILES%\BDNA\WinCS\logs`. The file store is located on each Linux component server in the `$BDNA_HOME/file_store/` directory.

Archiving the logs on a per-scan basis assists in tracking down any possible scan issues by creating distinct sets of logs associated with distinct discovery activities. This is of course much easier than having to differentiate between the activities in a set of logs from multiple scans. For Linux servers, log files can be archived using the archival software available in Linux. The following command will archive and then remove all Linux component log files into the home directory:

```
<DATE>-bdnalogs-<HOSTNAME>.tar.bz2
```

```
tar --remove-files -cvjf ~/`date "+%Y%m%d"`- \
bdnalogs-$HOSTNAME.tar.bz2 $BDNA_HOME/logs/
```

For Windows servers, the logs directory has to be moved and then archived manually.

Tip: Installing Windows ports of UNIX utilities will allow the UNIX tar command to work on Windows. A set of Windows ports are available for download at http://unxutils.sourceforge.net/.

Archiving the file store will enable faster recovery of discovery operations if there are serious failures. The file store is only created on Linux component servers. The following command can be used to archive Linux component file stores into the home directory:

```
<DATE>-bdnafilestore-<HOSTNAME>.tar.bz2
```

```
tar -cvjf ~/`date "+%Y%m%d"`-bdnafilestore- \
$HOSTNAME.tar.bz2 $BDNA_HOME/file_store/
```

Other Operations

Restoring Schemas

Oracle schemas that were saved either for backup or template reasons can be restored by using the Oracle import command and executing BDNA SQL packages. Restoring schemas requires a valid Oracle system credential.

The following command describes the syntax for the Oracle import command:

```
imp system@<SERVER> file=<FILENAME> \
fromuser=<EXPORT USERNAME> touser=<CURRENT USERNAME>
```

The SERVER value can be obtained by executing the Linux command:

```
grep -i -e "^bdna.dbServer" \ $BDNA_HOME/conf/connection.properties
```

The FILENAME value is the path to the input file, i.e. the Oracle export file. The EXPORT USERNAME value is the original Oracle schema name used to create the export file.

Tip: If unknown, the EXPORT USERNAME value can be obtained by executing the Linux command `head –n 10 <FILENAME>`.

The CURRENT USERNAME value is the name of the Oracle schema where data is to be restored. This Oracle user needs to be created prior to executing the Oracle import command.

Note: For more information on creating Oracle users for BDNA purposes, refer to "Configuring Oracle for BDNA" on page 11.

If the Oracle import completes successfully without any warnings, a BDNA SQL procedure needs to be executed against the newly created Oracle schema before it can be used. Use the following syntax for executing the SQL procedure against the newly created schema:

```
sqlplus <CURRENT USERNAME>@<SERVER> \
@$BDNA_HOME/sql/oracle/PLATFORM_PKG_ora9.sql
```