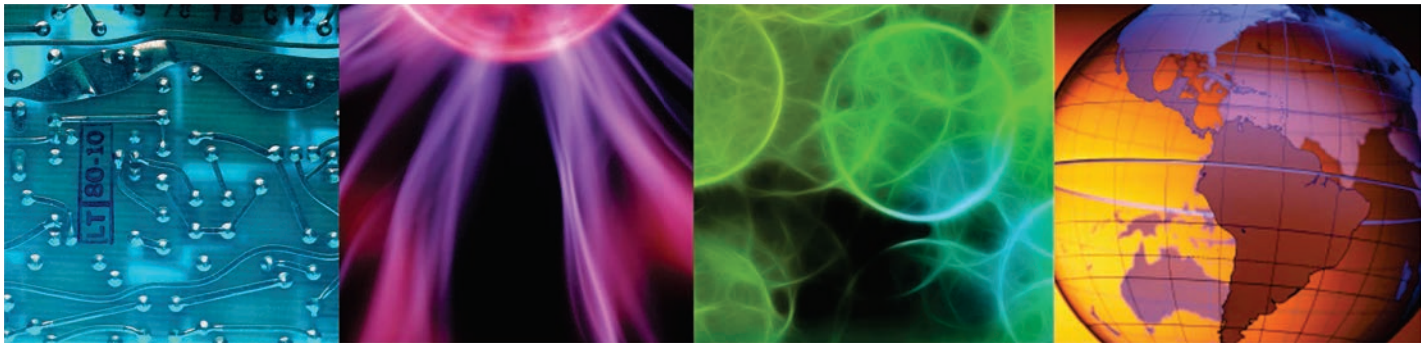
IBM Training

IBM Tivoli Netcool Configuration Manager 6.4.2: Operations and Configuration

Course Guide

Course code TOD44 ERC 1.0

June 2016



ibm.com/training

Authorized IBM Training

All files and material for this course are IBM copyright property covered by the following copyright notice.

© Copyright IBM Corp. 2016. All Rights Reserved.

US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM’s sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results.

Contents

About this course . . . xviii
About the student . . . xix
Learning objectives . . . xx
Course agenda . . . xxiii

1 IBM Tivoli Netcool Configuration Manager solution overview. . . . 1
Objectives . . . 2
Lesson 1 Business value . . . 3
Business value description . . . 4
Network configuration and change control manager . . . 5
Device configuration database . . . 6
Implementing change control . . . 7
Audit trail of device interactions . . . 8
Standardized change templates . . . 9
Analyze configurations to ensure standards . . . 10
Network security layer . . . 11
Integration with other systems . . . 12
Support a network of multiple vendors . . . 13
Lesson 2 Architecture . . . 14
Architecture description . . . 15
Deployment architecture . . . 16
Logical server layout . . . 17
Presentation server tasks . . . 18
Presentation server API portal . . . 19
Worker server tasks . . . 20
Evaluation server tasks . . . 21
Relational database tasks . . . 22
Typical architectural layout . . . 23
Multiple servers with virtualization . . . 24
Scalability . . . 25
Lesson 3 Terminology . . . 26
Terminology descriptions . . . 27
Student exercises . . . 32
Summary . . . 33

2 Configuration manager interface . . . 34
Objectives . . . 35
Lesson 1 Accessing the application . . . 36
Client computer prerequisites . . . 37

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Java Web Start . . . 38
Configuration manager GUI . . . 40
Advanced searching . . . 42
Lesson 2 The queue manager . . . 43
Queue manager overview . . . 44
Workflow states . . . 45
Execution status . . . 47
Lesson 3 The resource browser . . . 48
Resource browser overview . . . 49
Realm structure . . . 50
Lesson 4 User preferences and help . . . 51
User preferences . . . 52
Online help . . . 53
Student exercises . . . 54
Summary . . . 55

3 Single change configuration management . . . 56
Objectives . . . 57
Lesson 1 Discovering network devices . . . 58
Discovery . . . 59
Discovery process . . . 60
Discovery methods . . . 61
Starting the GUI discovery process . . . 62
BulkLoad discovery . . . 63
Discovery in Tivoli Network Manager . . . 64
Enhancing discovery . . . 65
Stand-alone discovery . . . 66
Lesson 2 Navigating the configuration and changes . . . 67
SmartModels and standard support . . . 68
SmartModel configuration . . . 69
Intelligent management: version restoration . . . 70
Viewing configurations . . . 71
Viewing the CLI of the device . . . 72
Configurations have a state . . . 73
Modifying configurations . . . 74
Submitting the change to a device . . . 75
Unit of work wizard . . . 77
Lesson 3 Making direct connections to devices . . . 78
VTY connections over predictable paths . . . 79
Device terminals . . . 80
Accessing a device terminal . . . 81
Lesson 4 Device synchronization . . . 82
Change management defaults . . . 83
Synchronizing the network . . . 84
Options of synchronization . . . 86
Only changed configurations are saved . . . 87

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Instructor demonstration 88
Student exercises 89
Summary 90

4 Mass change configuration management 91
Objectives 92
Lesson 1 Command sets 93
Change templates 94
Vendor-specific commands 95
Units of work 96
Parameters 97
Types of command sets 98
Modeled command sets 100
Native command sets 101
Interrogation command sets 102
Lesson 2 Building simple command sets 103
Creating command sets 104
Instructor demonstration 105
Create trivial command sets 107
Lesson 3 Processing command sets 108
Overview 109
Executing report only command sets 110
Using comma-separated value (CSV) files 111
CSV file format 112
Apply a CSV file 113
Student exercises 114
Summary 115

5 Administrative tasks 116
Objectives 117
Lesson 1 Approving changes 118
Units of work and tickets 119
Group policy 120
Approving units of work 121
Rejecting units of work 122
Ticketing behavior 123
Lesson 2 Stopping and restarting units of work 124
Stopping a unit of work 125
Stopping a processing unit of work 126
Restarting a unit of work 127
Filtering resources from a requeued UOW 128
Lesson 3 Searching and comparing units of work or resources 129
Searching for units of work or tickets 130
Searching for resources 132
Customizing the queue manager 134
Building custom searches 135
Saving custom searches 136
Adding custom searches 137
Comparing device configuration states 138

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Comparing two devices . . . 139
Comparing two configurations from two devices . . . 140
Comparing native or modeled configurations . . . 141
Lesson 4 Restoring a versioned configuration . . . 143
Restoring configurations . . . 144
Using models to compare configurations . . . 145
Restoring an IOS device . . . 146
Restoring a configuration on a device . . . 147
Restoring configurations by using other methods . . . 148
Standard device driver support . . . 149
Lesson 5 Reporting . . . 150
Running reports . . . 151
Security rules . . . 152
Accessing the reports in Tivoli Integrated Portal . . . 153
Tivoli Common Reporting generates the report . . . 154
Student exercises . . . 155
Summary . . . 156

6 Netcool integrations . . . 157
Objectives . . . 158
Lesson 1 Integration with other Netcool products . . . 159
Integration data flow . . . 160
Current integrations . . . 162
Importing devices from network manager . . . 163
Configuring device synchronization . . . 164
Traps are sent to monitoring software . . . 165
Trap types . . . 166
Enriching topology and events . . . 167
Embedded wizards to initiate actions . . . 168
Activity viewer . . . 169
Student exercises . . . 170
Summary . . . 171

7 Authentication and authorization model . . . 172
Objectives . . . 173
Lesson 1 Enabling access control . . . 174
Gateway to network devices . . . 175
User authentication . . . 176
Defining access control . . . 177
Lesson 2 Realms . . . 178
Realm structure . . . 179
Realm structure by customer . . . 180
Realm structure examples . . . 181
Domains and realms . . . 182
Tivoli Network Manager domains . . . 183
Tivoli Netcool Configuration Manager realms . . . 184
Lesson 3 Groups and users . . . 185
Determining how groups interact with realms . . . 186
Group structure . . . 187

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Group permissions basics . . . 188
Permission settings . . . 189
Realm security and file permissions . . . 190
Creating a group . . . 191
Creating a group name . . . 192
Implementing group activities . . . 193
Adding users to a group . . . 196
Defining an approval policy . . . 197
Define realm rights . . . 198
Define resource rights . . . 199
Defining content rights . . . 200
Group modification policies . . . 201
Creating users . . . 202
Associating a user to one or more groups . . . 203
Associating a user to multiple groups . . . 204
Discovering which groups you belong to . . . 205
Lesson 4 FIPS 140-2 security . . . 206
FIPS 140-2 compliance . . . 207
Scope of FIPS compliance . . . 208
Client access modes for FIPS . . . 209
Configuring FIPS on server . . . 210
Configuring FIPS on client side . . . 211
Student exercises . . . 212
Summary . . . 213

8 Device management . . . 214
Objectives . . . 215
Lesson 1 Modeling and drivers . . . 216
Advantages of modeling . . . 217
Modeling meta-language: XML . . . 218
Vendor syntax . . . 219
Setting the banner on Juniper M-series router . . . 220
Setting the banner on Cisco CatOS switch . . . 221
Setting the banner on Cisco IOS router . . . 222
Setting the banner on Extreme Summit switch . . . 223
Converting raw CLI to XML . . . 224
Using VTMOS schema to convert to XML . . . 225
Modeled device types . . . 226
Using drivers to model devices . . . 227
Device driver architecture . . . 228
Lesson 2 Security sets . . . 229
Defining security at a command level . . . 230
Mapping a native configuration to XML hierarchy . . . 231
Layering security sets onto a hierarchy . . . 232
Creating security sets . . . 233
Formatting security sets . . . 234
Applying a security set . . . 236
Results of applying a security set . . . 237

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Device interaction . . . 238
Overview . . . 239
Device interaction example . . . 240
Challenges to device interaction . . . 242
Lesson 4 Customizing device interaction . . . 243
Customizing the solution for a customer network . . . 244
Customization at a realm level . . . 245
Realm-level generalized resources . . . 246
Generalized resources that are specific to VTMOS . . . 247
Generalized resources resolution example . . . 248
Generalized resource location . . . 250
UOW logs . . . 251
Lesson 5 Device authentication . . . 252
Overview . . . 253
Finding device credentials for VTY access . . . 254
Authentication resource . . . 255
Successful credentials . . . 256
Encryption of authentication resource . . . 257
Lesson 6 Access resource . . . 258
Communication methods . . . 259
Modifying default access methods . . . 260
Modifying individual device properties . . . 261
Modifying realm-level resource access . . . 262
Access types and transport options . . . 263
SSH options . . . 266
Configuration . . . 267
Lesson 7 File transfer resource . . . 270
Moving data between workers and devices . . . 271
Moving a configuration change to a device . . . 272
Localizing all services on a worker server . . . 273
Resolving the local FTP host name . . . 274
Initializing the file transfer resource . . . 275
Defining data for ftpInfo . . . 276
Student exercises . . . 277
Summary . . . 278

9 Using workflow and scheduling . . . 279
Objectives . . . 280
Lesson 1 Controlling the flow of tickets . . . 281
Workflow overview . . . 282
Controlling the approval tickets . . . 283
Selecting the workflow realm . . . 284
Approval ticket generation example . . . 285
Approval ticket generation example: East devices . . . 286
Approval ticket generation example: West devices . . . 287
Approval ticket generation example: All devices . . . 288
Approval groups . . . 289
Ticket approval policy . . . 290

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Checking for approvers 291
Lesson 2 Scheduling units of work 292
Scheduling overview 293
Scheduling window 294
Estimating UOW execution time 295
Recurring units of work 297
Lesson 3 Understanding server time 298
Overview 299
When time zones are relevant 300
Storing UOW time stamps in GMT 301
Time stamps in local time zone 302
Executing units of work based on local time zone 303
Student exercises 304
Summary 305

10 UOW management 306
Objectives 307
Lesson 1 Units of work and tasks 308
Overview 309
Units of work are decomposed 310
Concurrent and nonconcurrent tasks 311
How a UOW is run 312
Task execution order 313
Setting priority 314
Impacts of priority 315
Lesson 2 Task processing 316
Designating work servers 317
Worker server resource 318
Configuring worker server resources 319
Worker server resource resolution example 320
Simultaneous worker server tasks 321
Workers and tasks 322
Worker server process scenario 323
Threads process two nonconcurrent units of work 324
Device is locked by a thread 325
Thread looks for other work to do 326
Thread returns to the previous UOW 327
Improving performance for nonconcurrent units of work 328
Splitting a unit of work 329
Reasons for splitting a UOW 330
Unit of work performance 331
Lesson 3 Change management quality 332
Configuration change management 333
Changing a configuration 334
Scheduling alert 335
Preparing for a device change 337
Ensuring that no errors are returned 338
Restoring to a previous state 339

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Retrieving the configuration . . . 340
Stale configuration state . . . 341
Student exercises . . . 342
Summary . . . 343

11 IBM device terminal . . . 344
Objectives . . . 345
Lesson 1 Device terminal overview . . . 346
IBM Device Terminal (IDT) . . . 347
IDT architecture . . . 348
Main IDT daemon server . . . 349
IDT controlled access . . . 350
IDT synchronized changes . . . 351
IDT keystroke logging . . . 352
Access to a device terminal . . . 353
IDT logging . . . 354
Lesson 2 Device terminal command filtering . . . 355
Command filtering . . . 356
Configuring a command filter . . . 357
Defining the commands to permit or deny . . . 358
Testing command filters . . . 359
Applying command filters to groups . . . 360
Results of applying command filter . . . 361
Lesson 3 Device terminal administration . . . 362
Providing group access to IDT . . . 363
Viewing logs . . . 364
View active sessions . . . 365
Configuring other IDT options . . . 366
Student exercises . . . 368
Summary . . . 369

12 Advanced command sets . . . 370
Objectives . . . 371
Lesson 1 Command set types . . . 372
Command sets . . . 373
Native and modeled command sets . . . 375
Command set comparison . . . 376
Lesson 2 Native command sets . . . 377
Overview . . . 378
Security settings . . . 379
Interrogation mode . . . 380
Interrogation mode output . . . 381
Configuration change . . . 382
Parameters . . . 383
Line-by-line mode . . . 384
JavaScript support . . . 385
Lesson 3 Modeled command sets . . . 387
Creating a modeled command set . . . 388
Defining markup in a command set . . . 390

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Adding commands to a device 391
Deleting commands from a device 393
Importing data into a command set 395
Testing a modeled command set 397
Instructor demonstration 399
Modifying commands on a device 400
Using wildcards 402
Using parameters 404
Instructor demonstration 406
Matching on sibling commands 407
Parent-child-sibling relationships 408
Lesson 4 Applying command sets 410
Overview 411
Execution mode and failure options 413
Report only command set UOW logs 415
Command set rollback options 417
Entering values into parameters 418
Modeled command set application rules 420
Lesson 5 Command set groups 421
Command sets are vendor-specific 422
Command set groups are vendor-neutral 423
Add community string across network 424
Command set creation 425
Applying command set group 426
Student exercises 427
Summary 428

13 Device OS upgrade manager. 429
Objectives 430
Lesson 1 OS upgrade overview 431
Upgrade manager 432
Upgrade process flow diagram 433
System preparation 435
Required resources 436
Lesson 2 OS registry 437
OS registry resource 438
Configuring the OS registry 439
Connecting to the image server 440
Loading the selected image 441
Lesson 3 OS specification 443
OS specification resource 444
Specifying name and VTM 445
Selecting the OS version and files 446
Selecting a boot command set 447
Updating boot parameters 448
Selecting a parser and describing the work 449
Custom parsing 450
Lesson 4 Optional resources 452

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Optional resources . . . 453
Using a file transfer resource . . . 454
Defining altFtpInfo data . . . 455
OS upgrade device script . . . 456
OS upgrade device script example . . . 457
Lesson 5 Launching an OS upgrade . . . 458
Upgrading devices . . . 459
Performing an OS upgrade . . . 460
Selecting an OS specification . . . 461
Selecting the device . . . 462
Configuring the execution options . . . 463
Selecting the workflow options . . . 464
Monitoring execution . . . 465
Completed upgrade . . . 466
Student exercises . . . 467
Summary . . . 468

14 Out-of-band change (OOBC) daemon. . . . 469
Objectives . . . 470
Lesson 1 Overview . . . 471
Reasons for out-of-band change . . . 472
Out-of-band change . . . 473
OOBC process . . . 475
Data flow diagram . . . 476
Reasons for consolidating events . . . 478
Event consolidation algorithms . . . 479
Synchronizer actions . . . 480
Lesson 2 Installing OOBC . . . 481
Installation prerequisites . . . 482
Installing the OOBC software . . . 483
Installation questions . . . 484
Postinstallation . . . 486
Lesson 3 Configuring OOBC . . . 488
Files in the execution directory . . . 489
Configuring OOBC . . . 491
Addressing multiple worker servers . . . 493
Addressing multiple authorized users . . . 494
Adding pattern matching rules . . . 495
Startup and shutdown . . . 497
User name or password . . . 498
Student exercises . . . 499
Summary . . . 500

15 Compliance manager interface . . . 501
Objectives . . . 502
Lesson 1 Accessing the application . . . 503
Initial access through web browser . . . 504
The Compliance user interface . . . 505
Lesson 2 The Policy Definitions tab . . . 506

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the Policy Definitions tab . . . 507
The Policy Definitions tab GUI . . . 508
Creating objects in the Policy Definitions tab . . . 509
Testing and policy versions . . . 510
Lesson 3 The Devices tab . . . 511
Key points of the Devices tab . . . 512
The Devices tab GUI . . . 513
View SmartModel configuration . . . 515
View by VTMOS . . . 516
Lesson 4 The Execution tab . . . 517
Key points of the Execution tab . . . 518
The Execution tab GUI . . . 519
Lesson 5 The Results tab . . . 520
Key points of the Results tab . . . 521
Key points of the validation view . . . 522
The validation queue GUI . . . 523
Drill into the compliance results . . . 525
Interpreting the details . . . 526
Key points of the remedial queue . . . 527
The Remedial Queue GUI . . . 528
Lesson 6 Integration with configuration management . . . 529
Compliance results are available . . . 530
Compliance details are also available . . . 531
Student exercises . . . 532
Summary . . . 533

16 Compliance reports . . . 534
Objectives . . . 535
Use Dashboard Application Services Hub to access reports . . . 536
Security settings affect reports . . . 537
Report types . . . 538
Report filters . . . 543
Processes generate compliance score . . . 544
Compliance score calculation . . . 545
You can automate and distribute reports . . . 546
Student exercises . . . 547
Summary . . . 548

17 Build and run a policy . . . 549
Objectives . . . 550
Lesson 1 Policy overview . . . 551
Policies . . . 552
Policies run optional actions . . . 553
Policies evaluate rules to determine compliance . . . 554
Rules evaluate one or more definitions . . . 555
Rules can run remedial actions . . . 556
Processes evaluate policies against devices . . . 557
Building policies . . . 558
Center for Internet Security example . . . 559

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policy realms . . . 560
Base policy realms support security . . . 561
Lesson 2 Building a definition . . . 562
Creating a definition . . . 563
Definition overview . . . 564
Creating the definition . . . 565
Differences between CLI and modeled . . . 566
Creating a definition . . . 568
Enter the search terms . . . 569
Notes about definitions . . . 570
Lesson 3 Building a rule . . . 571
Creating a rule . . . 572
Rules determine noncompliant devices . . . 573
Rules are vendor-specific . . . 574
Rules can be OS-specific . . . 575
Advanced VTMOS selection . . . 577
Rules use logic gates . . . 578
Create a rule . . . 579
Create the compliance logic . . . 580
Lesson 4 Email action . . . 581
Policies start optional email actions . . . 582
Email action overview . . . 583
Associate email action to policy . . . 584
Email example . . . 585
Emails are only created by a process . . . 586
Lesson 5 Build and test a policy . . . 587
Creating a policy . . . 588
Policies define a vendor-neutral condition . . . 589
Policies are a collection of vendor-specific rules . . . 590
Severity and weight . . . 591
Policies can create side effects . . . 592
Policies can be tested . . . 593
Create a policy . . . 594
Lesson 6 Exempting a device from a policy . . . 595
Reasons to exempt devices from a policy . . . 596
Exempting devices . . . 597
Lesson 7 Build and run a process . . . 598
Creating a process . . . 599
Processes evaluate policies against devices . . . 600
Automatic validations . . . 601
Select the policies to include in the process . . . 602
Preemptive compliance . . . 603
Select realms or devices . . . 604
Device coverage . . . 605
Setting local parameter values . . . 606
Scheduling process . . . 607

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Running processes 608
Student exercises 609
Summary 610

18 Remediation 611
Objectives 612
Lesson 1 Remediation actions 613
Creating a remediation action 614
Key points of the remediation action 615
Verify that the command set is defined first 616
Build a remediation action 617
Link a remediation action to command set 618
Optional remediation steps 619
Associating the remediation action to a rule 620
Lesson 2 Parameterization 621
Remediation notes parameter values 622
Parameter values can pass to command sets 623
Lesson 3 Approvals 624
Remediation actions can be approved 625
Each group can have different settings 626
Student exercises 627
Summary 628

19 Advanced definitions 629
Objectives 630
Lesson 1 General concepts 631
Definitions search through a BLOB of data 632
Most searches use regular expressions 633
Definitions use regular expressions 634
Definitions search for presence or absence of data 635
Definitions are made up of one or more evaluations 636
Evaluations can use one or more lines 637
Optional evaluations in a definition 638
Optional evaluations in a definition example 639
Not assessed prevents false positive results 640
Lesson 2 CLI definitions 641
Evaluate the raw configuration 642
Creating a CLI definition 644
Lesson 3 Show command definitions 645
Evaluate data that is retrieved from device 646
Devices already cache show data 647
Using cached commands is faster 648
Creating a native command definition 649
Lesson 4 SmartModel definitions 650
Evaluate the modeled configuration 651
SmartModel definitions search XML 652
UNIX-like syntax for finding nodes 653
Modeled definitions allow contextual XPath 654
Modeled definitions provide additional data 655

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a modeled definition . . . 656
Select a SmartModel to build XPath . . . 657
Select commands that are retrieved and searched . . . 658
Select the command to analyze . . . 660
Set the analysis on the nodes of interest . . . 661
Select the test condition for XPath . . . 662
Completed evaluation . . . 663
Manual override . . . 664
Show optional data about context . . . 665
Lesson 5 Using parameters in definitions . . . 666
Parameters make definitions reusable . . . 667
Parameter types . . . 668
Local parameters . . . 669
Global parameter . . . 670
Creating global parameters . . . 671
Group parameter . . . 672
Group parameters use evaluation criteria . . . 673
Difference between matching criteria . . . 674
Match exactly is a useful tool . . . 675
Extractions . . . 676
Extraction object in GUI . . . 677
Retrieve TACACS server example . . . 678
Example of an extraction in an evaluation example . . . 679
Policy realm implications . . . 680
Parameter administration . . . 682
Lesson 6 Using JavaScript . . . 683
JavaScript allows additional flexibility . . . 684
JavaScript parameter . . . 685
JavaScript definition . . . 686
Student exercises . . . 687
Summary . . . 688

20 Preemptive compliance . . . 689
Objectives . . . 690
Lesson 1 Overview . . . 691
Preemptive compliance overview . . . 692
Preemptive compliance is part of UOW wizards . . . 693
Preemptive compliance affects units of work . . . 694
Preemption starts after UOW submission . . . 695
Preemption affects approvals . . . 696
Interpreting the unit of work log . . . 698
Lesson 2 Requirements and configuration . . . 699
Preemptive compliance reliance on SmartModels . . . 700
Steps to enable preemption features . . . 701
Define policies as preemptive . . . 702
Process must declare if it has preemptive policies . . . 703
Process must include devices to apply preemption . . . 704
Add preemptive activities to appropriate group . . . 705

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

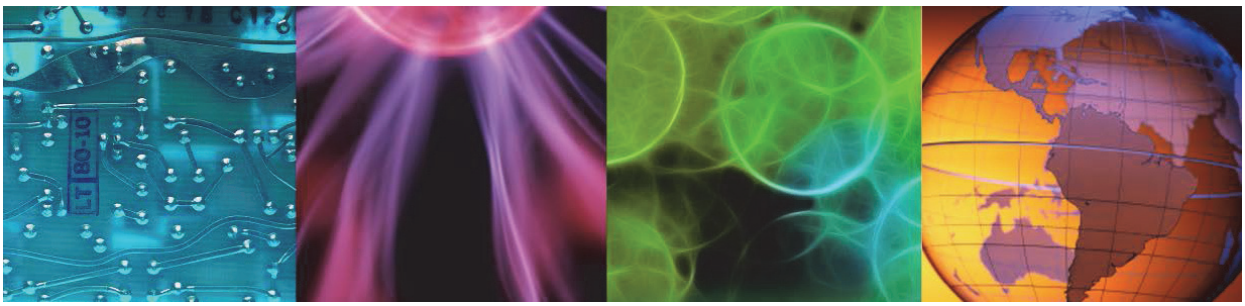
Groups who can disable preemptive compliance 707
Viewing preemptive policies for a device 708
Student exercises 709
Summary 710

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

About this course

IBM Tivoli Netcool Configuration Manager 6.4.2 Operations and Configuration



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This 40-hour self-paced course provides a comprehensive set of skills needed to operate, and configure the IBM Tivoli Netcool Configuration Manager solution for managing, modifying, and ensuring the compliance of network device configurations. This course provides a structured set of topics that cover many aspects of the solution and how it supports the network configuration management environment. This training provides access to virtualized servers, network devices, and extensive exercises to demonstrate how the solution manages network devices and the creating of compliance policies to ensure correct configuration settings. Class work includes the initial configuration of the software so that you can run network discovery. Discussions describing the methods that the solution uses for device management and its customization are included. Also included in the course ware are details about device modeling technology.

You work with the integrations between IBM Tivoli Netcool Configuration Manager, IBM Tivoli Network Manager, and IBM Tivoli Netcool/OMNIbus to gain experience with how these solutions work together to help manage the network.

The lab environment for this course uses the Linux platform.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

For information about other related courses, visit the Cloud & Smarter Infrastructure education training paths website:

ibm.com/software/software/tivoli/education/

	Details
Delivery method	Self-paced (SPVC)
Course level	ERC 1.0
	This course is a new course.
Product and version	IBM Tivoli Netcool Configuration Manager V6.4.2
Duration	5 days
Skill level	Basic

About the student

Partners, services, support, and technical sales are the intended audience for this course.

Before taking this course, make sure that you have the following skills:

- Basic UNIX skills are helpful but not mandatory

Before taking this course make sure that you have taken the following courses

- None required

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Learning objectives

Objectives

In this course, you learn to perform the following tasks:

- Describe the solution architecture
- Discover and import network devices into the solution
- View and modify a configuration with the configuration editor
- Modify one or more devices with a command set
- Approve, stop and restart units of work
- Import devices that are discovered by IBM Tivoli Network Manager
- Implement view, add, modify, and delete rights throughout the application
- Create schedules for units of work and implement recurring units of work
- Optimize execution of tasks and steps to increase throughput
- Use the IBM device terminal to connect to a device and make changes

© Copyright IBM Corporation 2016

Objectives (continued)

- Build and test modeled command sets that have advanced features
- Use the *upgrade manager*
- Make out-of-band changes with the out-of-band change daemon and synchronize them
- Create and test simple compliance policies
- Use the existing compliance reports
- Create automated actions to remediate device compliance
- Set up preemptive compliance and observe its effects on unit of work execution

© Copyright IBM Corporation 2016

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM open badge program

Communicating your credentials in today's online marketplace can be challenging. Now you may easily and quickly share verified proof of your achievement wherever and whenever you choose.

Benefits of IBM open badges:

- Combine your credentials to form a complete overview of your skills
- Display your accomplishment in your e-mail signature
- Display your qualifications on social and professional networking sites
- Provide employers with easy, valid verification of your credentials
- See the IBM Open Badge Program for more information

This course is the prerequisite to obtaining an Explorer badge.
Complete all units in this course and receive 80% or higher on the final test to claim your badge.

© Copyright IBM Corporation 2016

This course is a prerequisite to obtaining the Explorer badge. For more information regarding the IBM badge program, please refer to IBM Open Badge Program web site at the following URL:

`https://www.youracclaim.com/org/ibm`

IBM open badge program: Test

Take the test to earn an Explorer badge. You must receive a score of 80% or higher to earn the badge.

Launch the test

© Copyright IBM Corporation 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

After you complete the course, take the exam to earn an Explorer badge at the following URL:

`http://www.ibm.com/developerworks/middleware/services/badges/badge-tos47.html`

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Course agenda

The course contains the following units:

1. IBM Tivoli Netcool Configuration Manager solution overview

   This unit provides an overview of the features that the IBM® Tivoli® Netcool® Configuration Manager solution can provide. It also includes an overview of its architecture and scalability. Included is the terminology that you see throughout the course.

   In this unit, you start the lab images.

2. Configuration manager interface

   In this unit, you learn how to access the configuration manager interface and download it to your desktop with Java Web Start. You learn how to log in to the application and browse through the main portions of the interface.

   In this unit, you start the Netcool Configuration Manager user interface and view some of the available features.

3. Single change configuration management

   This unit describes how to import devices into the solution. After you import devices and put them under management, you learn how to manipulate their configurations and submit units of work. You learn how to connect to the command-line interfaces of specific devices. Finally, you learn about device synchronization so that the network and the database can have a common view of the device's state.

   In this unit, you discover devices and import their configuration files. You learn how to modify the device configurations.

4. Mass change configuration management

   This unit, teaches you about the template technology of command sets. You learn about the types that are available and the ways that you can use these templates to make mass changes to the network. You are also shown a method to use a CSV file to define a mass change with customized data for each line in the file.

   In the exercises for this unit, you view and apply command sets to lab devices.

5. Administrative tasks

   This unit covers topics that are necessary to operate Netcool Configuration Manager. These topics include approving units of work (UOWs), stopping, and restarting units of work, processing complex searches, comparing configurations, restoring previous configurations, and viewing reports.

   In this unit, you submit two changes that require approval. You also stop (dequeue) and start (requeue) a unit of work.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

6. Netcool integrations

   This unit covers the integration between Netcool Configuration Manager, Netcool/OMNIbus, and Network Manager. It also displays useful information in Dashboard Application Services Hub with the use of a configuration management event viewer.

   In this unit, you learn about the integration between Netcool Configuration Manager, Netcool/OMNIbus, and Network Manager.

7. Authentication and authorization model

   This unit provides lessons about the Netcool Configuration Manager security model. You learn how security is implemented, based on realms and groups. You also learn the authentication and authorization structures and how Tivoli Network Manager domains align with Netcool Configuration Manager realms. You learn how view, add, modify, and delete capabilities are implemented in the application.

   The exercises in this unit demonstrate how to create users, groups and assign security attributes.

8. Device management

   This unit provides information on modeling and how it can be extended to include security. It also introduces you to how Netcool Configuration Manager interacts with devices and how those interactions can be customized for a customer's network.

   The exercises in this unit demonstrate how to configure some of the Configuration Manager objects that you use for managing devices.

9. Using workflow and scheduling

   This unit teaches you about ticketing and scheduling. Netcool Configuration Manager directs tickets to the appropriate groups for approval. You learn how to estimate how long a UOW might take. You also learn about scheduling units of work and setting time zones.

   In this unit you learn how to schedule units of work, and create recurring units of work.

10. UOW management

    This unit teaches how worker servers break down units of work into tasks and processes them. You learn how worker servers select and process tasks. Some tasks are processed in parallel with each other for efficient completion. Others are processed one after the other. This unit presents the quality checks that are part of the change management philosophy that the application supports.

    In this unit, you learn how Configuration Manager processes units of work.

11. IBM device terminal

    This unit presents the IBM Device Terminal (IDT), which an authorized user can make direct VTY connections to the devices under management. You can filter the input for these connections to intercept restricted commands and collect keystroke logs to record all input.

    The exercises in this unit demonstrate how to use the IBM Device Terminal to make changes to devices.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

12. Advanced command sets

    This unit teaches you how to build command sets. You learn the advantages of choosing modeled or native command set types. You build and test modeled command sets by using their advanced search and replace features. You also learn about command set groups.

    The exercises in this unit demonstrate how to create, test, and apply modeled command sets.

13. Device OS upgrade manager

    This unit is about using Netcool Configuration Manager to upgrade the operating systems on network devices. You learn the upgrade process and the objects you need to create to ensure a successful upgrade.

    In the exercises for this unit, you work with two Netcool Configuration Manager resources. The operating system upgrade process requires these resources. The first resource is a catalog of IOS images. The second resource is the upgrade template for the upgrade of a device to a specific IOS image.

14. Out-of-band change (OOBC) daemon

    This unit teaches you how to use the out-of-band change (OOBC) daemon. OOBC is an external application that monitors a syslog file and parses it for events that warrant synchronizing the device. You learn how to install and configure this utility to monitor any flat file that receives syslog messages from the network.

    The exercises in this unit demonstrate how Configuration Manager identifies out-of-band device changes.

15. Compliance manager interface

    This unit is an introduction to the client graphical user interface (GUI) for the compliance manager. It provides an overview of the client technology and the way the software downloads to the client computer. You learn the main aspects of the compliance user interface.

    The exercise in this unit introduce the studentyou to the compliance manager user interface.

16. Compliance reports

    This unit presents the user with an overview of the compliance reports that are available in Tivoli Common Reporting. It shows how to access these reports and provides examples. You learn how to calculate a score to measure the compliance of your network.

    The exercises in this unit demonstrate some of the available reports for compliance analysis.

17. Build and run a policy

    This unit provides an overview of the objects that make up a compliance policy and the role of objects in the compliance process. This unit shows how to create and test a simple policy. The unit also teaches the student how to exempt devices from a policy and how to include a policy into a production process for scheduled execution.

    The exercises in this unit demonstrate how to create compliance policies.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

18. Remediation

    Remediation is an important feature of the compliance manager. This unit describes how remediation works and how you configure it. It explains how remedial actions are approved and how to use parameterization to ensure that the changes sent to remediate devices have the correct values. You learn how to implement approvals for remedial actions.

    The exercises in this unit build on the previous unit by expanding compliance policies to include remediation actions.

19. Advanced definitions

    This unit teaches you about the more advanced features to determine whether a configuration is compliant. This unit describes the different data structures that you can analyze and the different methods for analyzing them. It describes parameterization of definitions in detail. Finally, it introduces the use of JavaScript.

    In this unit, youthe student learns how to use various advanced features in policies, including XPath to search for text, and the extract function to retrieve text from a device.

20. Preemptive compliance

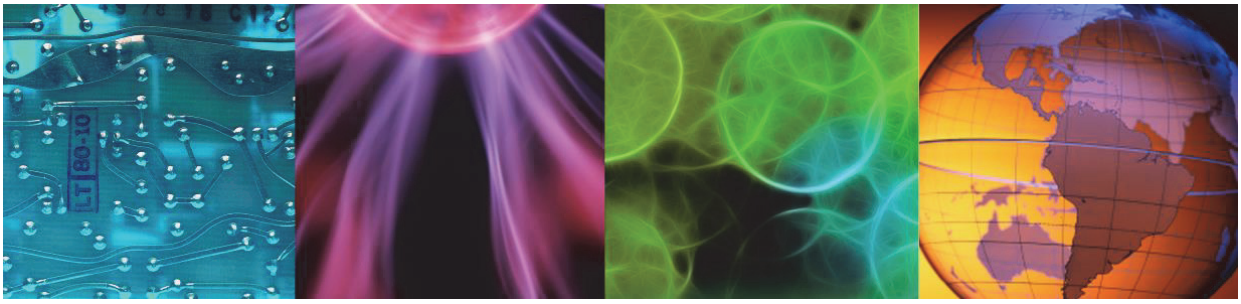
    This unit teaches how you can use compliance to inspect changes that you make in the configuration manager and notify users when changes take a device out of compliance. This unit teaches how to configure the application to enable preemptive compliance and how it affects users.

    The exercise in this unit demonstrate how to implement preemptive compliance to prevent a change from making a compliant device no longer compliant.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

1 IBM Tivoli Netcool Configuration Manager solution overview

IBM Tivoli Netcool Configuration Manager solution overview



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit provides an overview of the features that the IBM® Tivoli® Netcool® Configuration Manager solution can provide. It also includes an overview of its architecture and scalability. Included is the terminology that you see throughout the course.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Describe the key benefits of IBM® Tivoli® Netcool® Configuration Manager
- Explain the architecture and scalability
- Use specific terminology to describe the solution

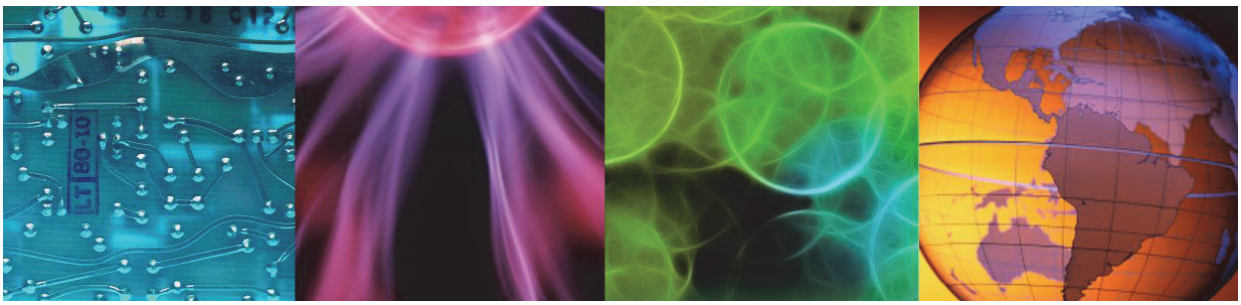
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Business value

Lesson 1 Business value



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to describe the four major areas of business value.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Business value description

Network configuration management and change control

Network security layer enforcing user access rights

Integration with other support systems

Support for a multiple-vendor network

© Copyright IBM Corporation 2016

Business value description

The business value of the IBM Tivoli Netcool Configuration Management (ITNCM) solution encompasses four strategic concepts.

- First, is the idea of implementing network configuration management and change control methodologies in your organization.
- Second, Configuration Manager provides many security tools to enforce user access privileges into the network and its associated data.
- Third, the solution provides several application programming interfaces (APIs) to allow northbound applications to gather network data and implement changes.
- Finally, the application is designed to address a multi-vendor network where communication protocols, syntax, and change implementation methods are vendor-specific.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Network configuration and change control manager

Maintain current configurations on all network devices

Implement network changes in a controlled and orderly fashion

Provide an audit trail of all network device access

Create standard configuration change templates

- Standardize change operations
- Provide procedural uniformity

Analyze configurations to ensure corporate standards

© Copyright IBM Corporation 2016

Network configuration and change control manager

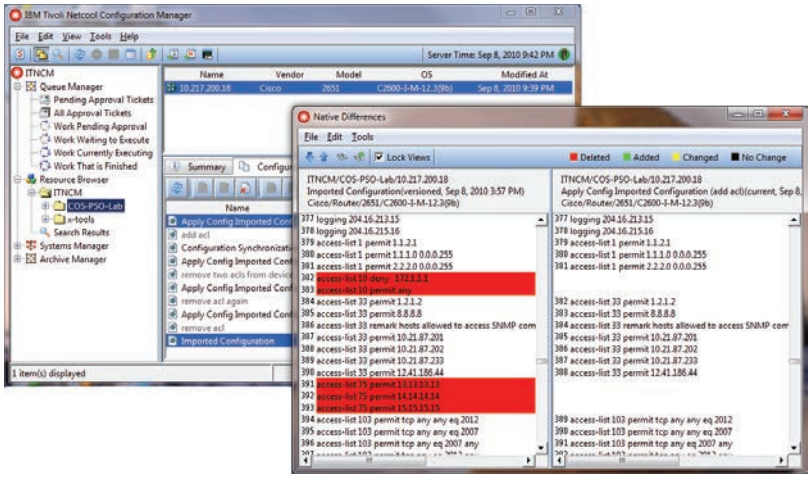
The value of Netcool Configuration Manager is based in its configuration management and change control philosophy. You can use change management systems for document control, computer-aided design, and source code management. You can use it to maintain current copies of information, including network device configurations. You can track and audit state changes for future needs. In similar configuration management systems, you might need rigorous steps to approve changes and move them into production. You might also be constrained to user-defined templates to apply standard changes to the objects that are managed. Finally, you can use the data that is captured about the device and analyze it to ensure that it meets the corporate standards for documentation, security, and any other industry best practices.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Device configuration database

View complete history of device changes

Compare differences between device configurations



© Copyright IBM Corporation 2016

Device configuration database

Tivoli Netcool Configuration Manager contains the history of all the changes that are made to a configuration over its lifecycle. It can contain as many configurations as the customer needs.

You can compare these configurations with each other to determine what changed between two points in time. These comparisons are not limited to single devices. You can compare configurations between two separate devices when they are the same vendor and type.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Implementing change control

Define workflow settings to ensure approvals are part of change process

Schedule changes for automatic application

Ensure changes are validated when applied to device

Implement nondisruptive restoration of device if change fails

© Copyright IBM Corporation 2016

Implementing change control

The product has tools both at the group level and at the device level to ensure that changes are applied correctly. At a group level, you can configure the work flow engine so that changes must be approved before they are applied.

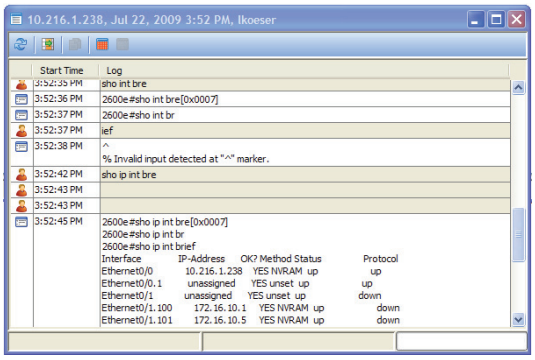
At a lower level in the application, users schedule changes to occur in execution windows. These users can align their tasks so that they follow the change control processes that are established for the particular device or devices.

At the lowest levels, as changes are applied to the device, listeners ensure that the changes are successfully accepted. If problems occur with the application of the changes, the device is restored to its previous state without service interruption.

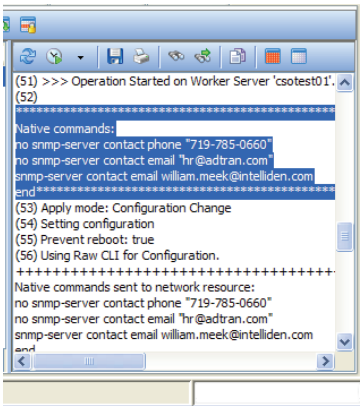
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Audit trail of device interactions

Execution and keystroke logs capture all interaction with the device



keystroke logs



execution logs

© Copyright IBM Corporation 2016

Audit trail of device interactions

Understanding how something moved from point A to point B is a key feature of any change control architecture. People can replay a series of events to fully understand the reason that something happened. This attempt to understand why events occurred is typically the result of a user-generated trouble ticket. Understanding the history of an event leads to process improvement.

With this product, you can track all the interactions with a device. These interactions include user interactive sessions with the device, or keystroke logs, and the automated change activities that are part of solution, the execution logs.

You can perform more auditing by the Out of Band Change (OOBC) solution. This tool synchronizes devices when users interact with devices outside of the product architecture.

Another method for understanding the change history of a device is to use the Activity Viewer, which is included in Dashboard Application Services Hub. This feature is integrated into both Tivoli Network Manager and Tivoli Netcool/OMNIbus and allows users to see change history as the result of a menu click in either of these applications.

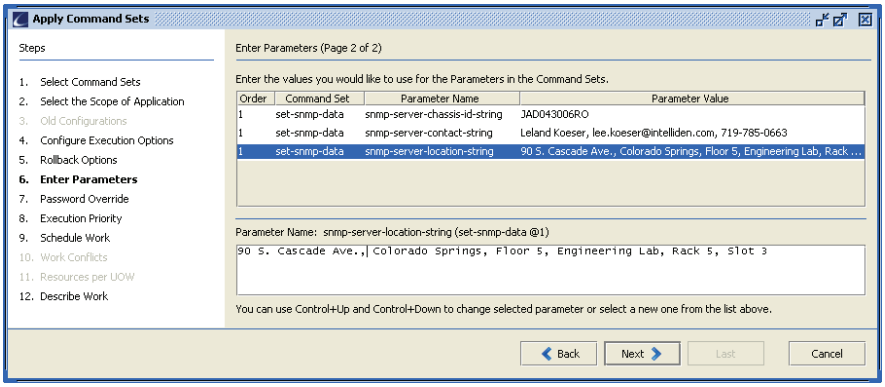
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Standardized change templates

Create templates with parameters that standardize changes

- You are prompted for values at execution time
- You can apply changes to one or more devices

Use match and replace in configuration searches



© Copyright IBM Corporation 2016

Standardized change templates

You can create standardized change templates that can be parameterized for reuse. With these templates, you can prepare changes to one or more devices. When prompted, you can enter the values that you want to change on the device.

With templates, less experienced users can prepare and implement changes to the network. Templates reduce the risk of giving users full access to the configuration mode.

These change templates can use advanced MATCH and REPLACE capabilities. These templates can make complex changes that require searching through configurations before a change is defined.

These templates are accessible to users of the application and to external applications that use the product application programming interface (API). An example is a provisioning application that creates, deletes, and modifies subinterfaces on an aggregation router.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Analyze configurations to ensure standards

Users implement vendor-neutral policy structures

Determines compliant and noncompliant devices

Command sets remediate problems

Automated checking of devices as their state changes on network

©

Analyze configurations to ensure standards

With all the network device information gathered and stored in the application database, Configuration Manager uses the Compliance Manager to analyze and ensure that industry and corporate best practices are implemented in configurations. You can implement vendor-neutral policy structures that determine whether a device is compliant or not. If a device is found not compliant, you can remediate the issue by using command sets. The solution can also automatically check for compliance as changes are implemented in the network.

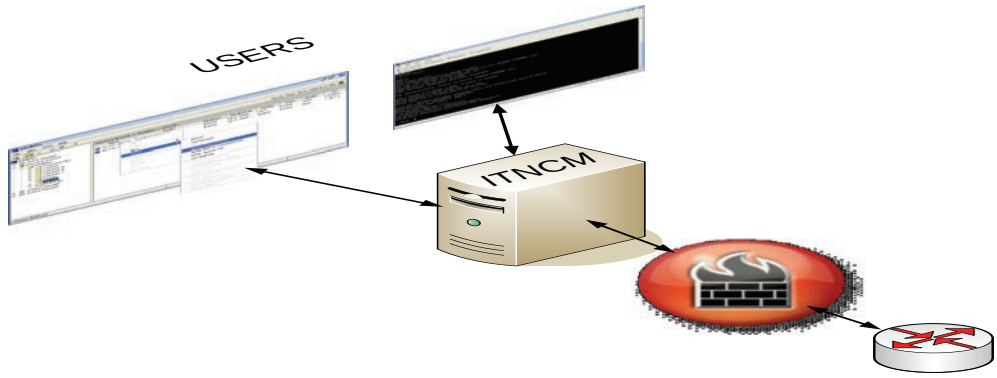
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Network security layer

Monitors and allows only authorized users to view, add, modify, and delete network devices

Protects the configuration of the network device from unauthorized changes

Ensures that only authorized commands are submitted to the device



© Copyright IBM Corporation 2016

Network security layer

You can use this product as the entry point for anyone who accesses the network hardware. Possible interactions include making configuration changes, accessing the command-line interface, or viewing device information, like hardware data or configuration files. You can divide the network so that user authorizations vary within different domains.

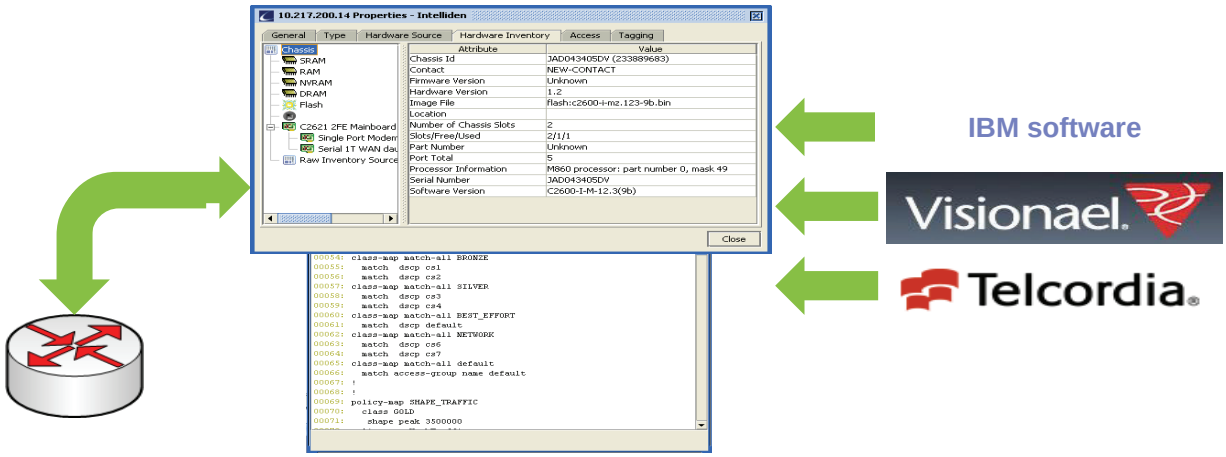
For example, a customer might have the rights to view the configurations of devices under their domain. However, they cannot view any other device under management. You can strictly control access to the command-line interface so that only experienced users can use it. You can also filter the commands that are submitted to the device through the command-line interface. For example, users can be prevented from submitting potentially harmful commands, like ***reload***.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Integration with other systems

Application programming interfaces (API) allows push and pull actions

- Java API
- REST API



© Copyright IBM Corporation 2016

Integration with other systems

You can choose from an extensive set of Java API commands to retrieve data from the server and submit a task to it. Most operations that you see demonstrated in this training can be coded into an API application and used by an external system.

You can use the product for inventory applications to update databases with near real-time information about network hardware. Provisioning applications can use the SmartModel technology to ensure that changes do not overwrite existing data in the configuration.

The Network Services Manager is a representational state transfer (REST) API that allows the creation and deletion of network services by using service templates that users create. This API has fewer functions than the Java API but it creates a means of tracking the application of services on a device to make their removal trivial.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Support a network of multiple vendors

Use SmartModel technology to support command-line interface (CLI) and SNMP managed devices

- Vendors with supported devices include Adtran, Alcatel, Arris, Cisco, Extreme, Foundry, Juniper, Lucent, Motorola, Nortel, Tellabs, WWP, Acme Packet, Harmonic, Huawei, and Nokia Siemens
- Support for other SNMP managed devices is expanding

Use standard device support for managing devices with no SmartModel technology

- Develop device support for text-based configurations by using standard processes and tools

© Copyright IBM Corporation 2016

Support a network of multiple vendors

This product has an architecture that is designed for a network with multiple vendors. Most element management systems (EMS) that you use to manage, monitor, and change devices are vendor-specific. These EMS solutions rarely support other vendor devices.

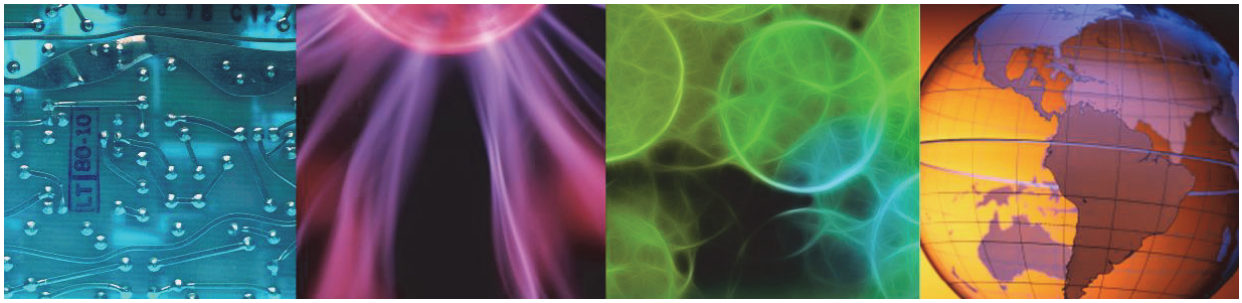
With the SmartModel technology, device configurations are normalized to a standard format, eXtensible Markup Language, or XML. Making this transition to an XML representation provides a common interface, standard toolsets, and similar change techniques. For example, little difference exists between building a change template for a CLI device and an SNMP device.

SmartModels are being developed on an ongoing basis as customers seek to use modeling as a means to define device configurations. If a SmartModel is not available, you can use standard device support to capture text-based or binary configurations for storage. Standard device support includes the use of product templates and supports most of the auditing and change management features associated with SmartModels.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Architecture

Lesson 2 Architecture



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the types of servers used in the solutions
- Describe the functions of each server in the application
- Describe how to make the solution highly available

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Architecture description

Built on industry standards

- Java EE server: IBM® Embedded WebSphere® Application Server 7.0
- Relational database: DB2/Oracle
- Reports: Cognos
- Java application programming interface (API)

Scalable through distribution of application servers and database

- Support for overlapping address spaces
- Multiple servers to address work loads
- DB2/Oracle data replication models

© Copyright IBM Corporation 2016

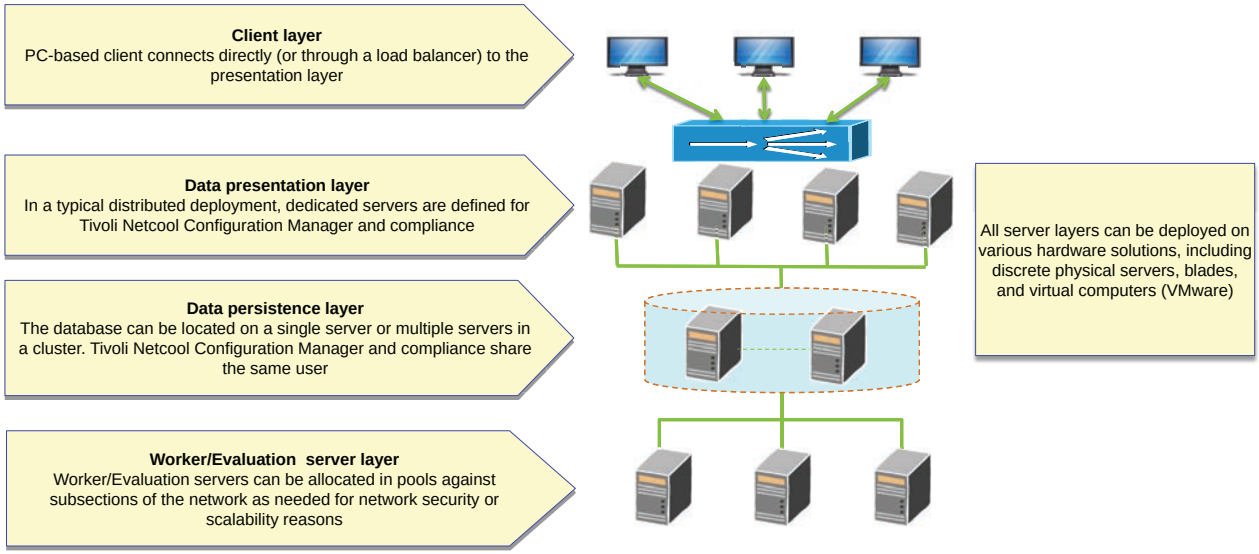
Architecture description

The solution is built to industry standards by using an open architecture. It also means having easy access to expertise on WebSphere, Oracle/DB2, and Cognos that are applications with a large user base. Integrations that are also straightforward with access to an experienced Java developer.

Scalability means that you can add more resources to the architecture. You might add resources to address network design issues like private address spaces. The application can scale to address larger work loads through expanding network coverage. Up-to-date database technologies include data storage that supports scalable features, like replication and clustering. The customer determines how critical the application is and scales the data storage services as necessary.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Deployment architecture



© Copyright IBM Corporation 2016

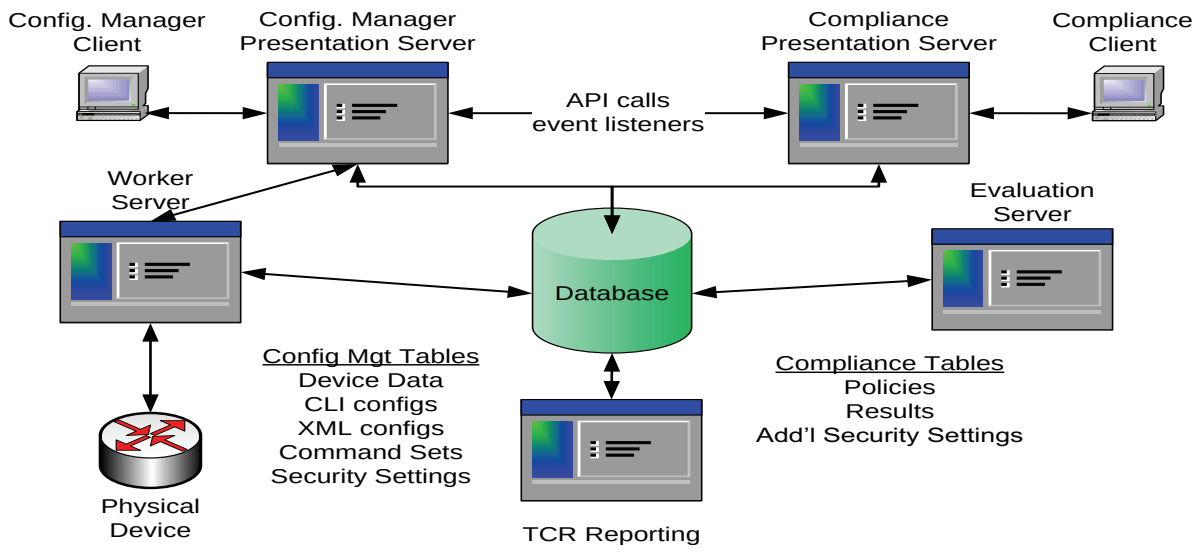
Deployment architecture

The deployment architecture consists of the following layers:

- The client layer uses a centrally managed thick client that supports automatic updates to the desktop software. Based on Java Web Start technology, the client layer is supported on any desktop that is operating the correct Java runtime environment.
- You can distribute the data presentation layer across many servers to support both user interaction and integration needs. This layer enforces the security architecture and ensures that the client does not view any unauthorized data.
- The data persistence layer store all the unique data about the implementation. You can design this layer to meet whatever data scalability and resiliency that the customer requires. Currently, support exists for Oracle and DB2 database servers.
- The worker/evaluation server layer is where most of the processing of the application occurs. You can scale this pool of processing capacity up and down, depending on the needs for network change and compliance. The evaluation server is designed for compliance analysis tasks. The worker server directly accesses network devices, converts native languages into XML, and applies changes.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Logical server layout



© Copyright IBM Corporation 2016

Logical server layout

The Configuration Manager solution is made of two separate applications that are hosted within the same WebSphere instance. These applications share a common database. The configuration management solution hosts device data, configurations, templates, and security settings. The compliance solution stores policies, the results of those policies against devices, and more security settings. Logically speaking, the applications are separate entities that can all exist on the same physical server if necessary. However, you might distribute all the servers, noted in the slide, to separate servers if necessary. The configuration management and compliance presentation servers always coexist within the same logical WebSphere container. However, you might distribute the compliance client to connect to one server and the configuration manager client to another. These servers are aware of each other's actions through event listeners and API calls. The worker server is the entity that connects to and communicates with the devices on the network. The worker server looks to the database regularly for tasks that it can process. In general, little connectivity exists between the configuration manager presentation server and the worker. However, when a user wants to directly connect to device with a virtual terminal, it must connect through the presentation server and then to a worker server. An evaluation server is similar to a worker server in that it does much of the processing for the compliance application. It looks to the database for policies it can process against devices. It never connects to physical devices. If that is needed, the compliance manager request the configuration manager to make a connection and retrieve the data. Finally, Tivoli Common Reporting is used to generate reports for both applications by directly accessing the database and building its reports.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Presentation server tasks

Serves as a security layer between the user and database of devices and configuration

Displays units of work, devices, and configurations with appropriate security setting, compliance policies and results

Sets up, schedules, and approves tasks

Processes XML to command-line interface (CLI) for GUI

Processes differences between configurations for GUI

Sends SNMP traps

© Copyright IBM Corporation 2016

Presentation server tasks

You use the presentation server for several tasks:

- It is the main security layer for defining what users can see and do with the GUI application. It applies all the view, add, modify, and delete rights that are assigned to the group a user belongs to.
- Using the security layer as a guide, it shows all the *units of work* and devices that a user can see within the GUI. It is where *units of work* are created, scheduled, and approved for execution by the worker servers. It converts XML-based configurations back into CLI for users to view. More security features can limit what the user sees in a configuration file. It can compare two configurations and show the differences between them.
- You can configure the presentation server to send SNMP traps to appropriate servers when various changes occur inside the application. For instance, a trap can be sent when a unit of work (UOW) begins on a device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Presentation server API portal

All API sessions connect to presentation server

An API session is the same as a GUI session

- Has the same security model
- API developers can view most functions
- Manipulation of XML is possible, although not preferred

An API cannot directly access a worker/evaluation server

© Copyright IBM Corporation 2016

Presentation server API portal

The presentation server also acts as the connection point for any API application. The presentation server cannot tell the difference between an API program that accesses the system and a standard user. The presentation server uses the same security model for the API login as for typical GUI usage. This security model ensures that the API program is limited in what it can do and what resources it can access in the application.

During an API session, you can create, modify, and delete devices within the application. The only limitation is that the API has no tools to directly manipulate device configurations. The ability to manipulate the XML and its associated markup requires a level of understanding that few customers want to invest in. Instead, the API uses prebuilt command sets to effect changes to devices. The API cannot directly access a worker server because no communications port is defined.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Worker server tasks

Converts CLI to XML and XML to CLI

Decomposes units of work into tasks for optimal processing

Processes tasks based on security settings, defined by worker server resources

Communicates with network resource

Sends FTP files to appropriate locations

Holds temporary files for configurations and changes

© Copyright IBM Corporation 2016

Worker server tasks

The worker server has the following main tasks:

- The worker server is designed to take device drivers and use them to convert raw configuration text (CLI) and convert it into an XML representation.
- The worker server takes *units of work* that users submit to the database from the presentation server and decompose them into tasks. A task is a major operation against an individual device. A worker eventually receives tasks and processes the requests against devices.
- In an implementation with multiple worker servers, the workers are assigned which devices they can process. Workers can be restricted to processing only the devices they access because of the network design.
- Workers communicate directly with network devices, typically through Telnet or SSH sessions. Workers can also employ SNMP when that is the only protocol that the device supports. This communication supports task execution or to provide the device when users are establishing interactive sessions.
- The workers host and FTP change files to the appropriate locations to support the network change mechanisms.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Evaluation server tasks

Similar to worker server

Evaluates compliance policies against devices

- Review device configuration
- Cached hardware data
- Requests configuration manager to gather live data

Determines if device is compliant or not

© Copyright IBM Corporation 2016

Evaluation server tasks

An evaluation server has tasks similar to a worker server. It is designed to enforce compliance. Its main purpose is to evaluate compliance policies against devices. Using either device configurations, cached hardware information, or live data that is retrieved from the device, it determines whether the device meets a specific compliance criteria or not.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Relational database tasks

Supports both Oracle and DB2 databases

Holds all the important items about the solution

Is used by all other servers

Is a single point of failure

Notes:

Oracle or DB2 installation and licensing is the responsibility of the customer

You can rebuild workers, presentation, and reporting servers

© Copyright IBM Corporation 2016

Relational database tasks

Configuration Manager supports both Oracle and DB2 database solutions. The database saves all important data for the solution. That includes the following information:

- Users and groups
- Devices and their configurations
- Generalized resources that are used for device interaction and security
- *Units of work* and their logs
- Keystroke logs for device interactions
- Scratch or tmp space to process *units of work*

The database is a single point of failure for the entire solution, and its implementation needs serious consideration from an architectural standpoint. You must consider how long the database can be down before seriously impacting operations. To help this decision, you can determine what tools are used in the application and other integration points. For example, if the device terminal solution is the main point of access into the network, its unavailability is a serious problem for users.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

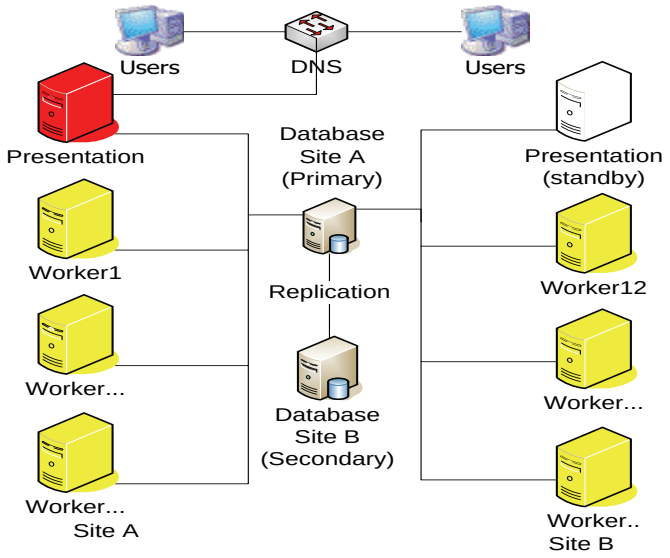
Typical architectural layout

Servers are typically defined as presentation or worker servers

Users access a single domain name server (DNS) host name

Failing over to secondary database or presentation server occurs outside the application

- Manual failover processes
- Automation through external monitoring tools



© Copyright IBM Corporation 2016

Typical architectural layout

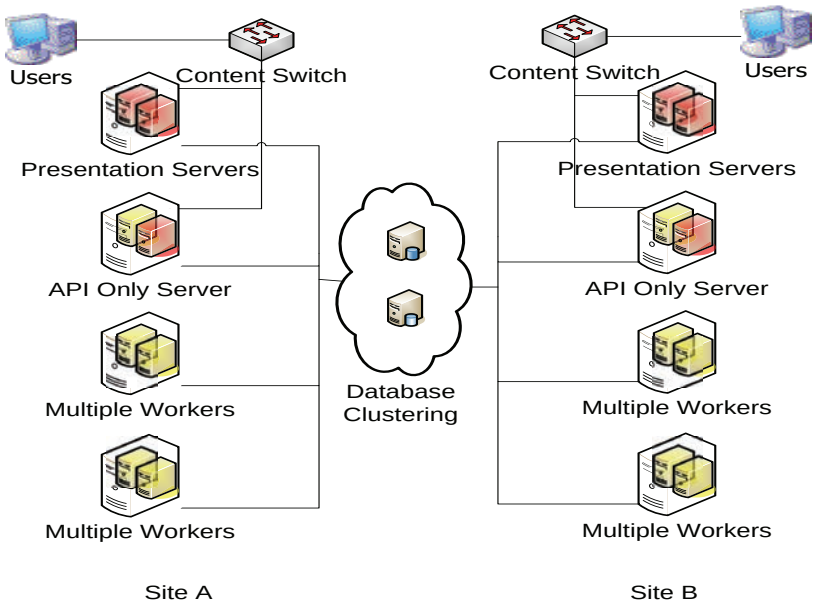
Typical layouts with device counts over 1000 have physical servers that typically perform one role: presentation or worker. A single active database server has a warm standby. A main presentation server is one that all users can access, and a secondary presentation server might be available on standby.

Currently, clustering technology is not employed with presentation servers. To implement multiple presentation servers, users must be manually directed to specific servers to ensure load sharing. Fail over to secondary presentation servers and databases is a manual process that you can automate with external monitoring software.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Multiple servers with virtualization

- Improved GUI response
- Virtual servers running on different ports
- Content switch for load sharing of presentation servers
- Servers at multiple sites for disaster scenarios
- Database clustering to ensure resiliency



© Copyright IBM Corporation 2016

Multiple servers with virtualization

With the advent of virtualization and content switching, multiple servers can exist on a single server. Each virtual server typically serves one role. You can accomplish load sharing with content switching. Designers can dedicate certain servers to specific tasks, like API integrations, to distribute workloads.

Virtualization supports easy scaling of both presentation and worker servers to adjust to different needs. Database clustering, such as DB2 high availability technology, can ensure that the architecture is scalable and resilient. Having presentation and worker servers at multiple sites ensures that the application is available during disaster scenarios with data centers.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Scalability

Manage small or large networks

- As small as 300 devices to as large as 100,000 devices

Expandable with additional servers

- Saves no critical network data on servers
- Bases current licensing on the total number of devices

Database technology allows for scaling of data store

- Clustering
- Network Attached Storage (NAS)
- Storage area network (SAN)

© Copyright IBM Corporation 2016

Scalability

Implementations range from as small as 300 devices to as large as 100,000 devices. No practical limits exist to the total number of devices that can be managed. Database space and read and write processing define limitations.

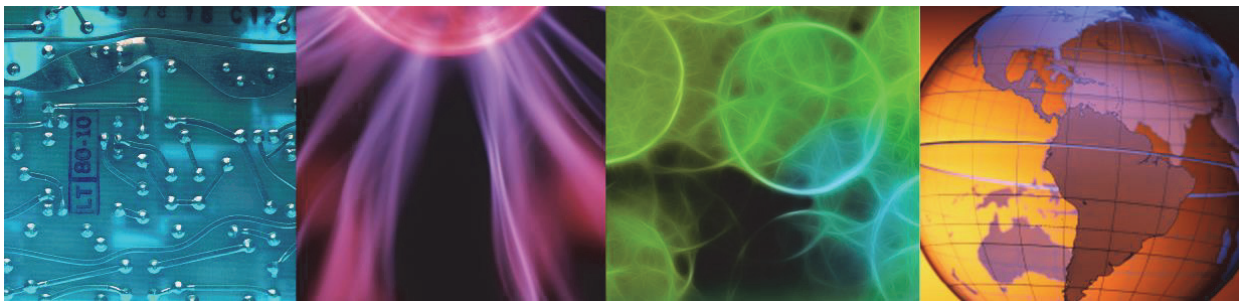
Adding presentation and worker servers is unimportant because no critical customer-specific information is saved on these servers. The only customer data is in the temporary change files that are used to modify configurations. These configurations are typically deleted after a week.

Licensing is based on the total number of devices under management. Adding servers to an implementation is not limited by legal issues, but by available computing resources. Options are available to use scaling technology, like clustering, and robust storage solutions, like Network Accessible Storage (NAS) and Storage Access Network (SAN).

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Terminology

Lesson 3 Terminology



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to describe the meaning of various terms that are used in the class.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Terminology descriptions

VTMOS

- Term used to describe a device
  - Vendor
  - Type
  - Model
  - Operating system

Network resource

- Routers
- Switches
- Firewalls
- Any other network devices

© Copyright IBM Corporation 2016

Terminology descriptions

VTMOS refers to vendor, type, model, and operating system. This term is used throughout the entire course to define a device. You use the VTMOS concept to select SmartModels from the drivers database that is installed on devices. You also use VTMOS as a filtering mechanism for items, such as custom scripts for one vendor, so that they are used only for that vendor.

A *network resource* is like a physical or virtual device, for example, a card in a chassis in a rack somewhere inside the network. The resource typically has a configuration that you are managing.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Terminology descriptions (continued)

Realm

- Logical container of resources
- A file or a folder directory for files
- An item that allows permissions, similar to UNIX

Generalized resource

- VTMOS or realm-specific files that are found by using hierarchical search mechanisms
- Files for these tasks:
  - Customized authentication
  - File transfer
  - Access scripting
  - Finding files needed by devices to process configuration changes

© Copyright IBM Corporation 2016

Terminology descriptions (continued)

A *realm* is a folder for organizing your network into logical areas. Realms align with the business view of the network and the people who work within it. Most realms have either network devices in them or other realms to further define the network hierarchy. You can also use realms for storing templates, security documents like passwords, and other items.

A *generalized resource* is a file in a realm. It is used for customizing interaction with network resources at or below that realm location. An example is having different passwords in one part of the network as compared to another part of the network, such as core devices compared to edge devices. Often, these generalized resources are specific to VTMOS. Some customers can have one policy for one type of device, and a different one for others. These resources are part of a resolution strategy that devices use to gather information about how they interact with the worker servers.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Terminology descriptions (continued)

Command set

- Use to modify one or more device configurations
- Allows parameterization for reusability

Security set

- Use for managing configuration content
- Controls what users can do inside a configuration file

Command filter

- Use for managing user device interaction
- Controls which commands users can submit in an device terminal session

© Copyright IBM Corporation 2016

Terminology descriptions (continued)

A *command set* is a reference to the template technology. Command sets support the creation of specific change documents that you apply to one or more devices to automate change. These templates can be parameterized for reuse. Command sets can use SmartModel technology to support complex search and replace scenarios. These command sets include checks to prevent overwriting existing configuration data.

Security sets support the implementation of view, add, modify, and delete capabilities to commands inside the configuration. Portions of the configuration can be read-only when users view it in the user interface.

Command filters are similar to security sets in that they control what users can do with devices. Instead of controlling the configuration editor, these filters define what commands can or cannot be used inside a terminal session. They are like access control lists that are used on routers.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Terminology descriptions (continued)

Unit of work (UOW)

- Work item associated with user-defined (or API) interaction with network devices
- The UOW is the object that moves through the workflow process that the application approves, schedules, and runs

Approval ticket

- One or more tickets are associated with a UOW
- Bases approvals on business policy

Task

- Units of work consist of one or more tasks
- Application of one major action against a device
- Task can run sequentially or in parallel

© Copyright IBM Corporation 2016

Terminology descriptions (continued)

A unit of work (UOW) is a user-defined or API-defined work item that interacts with one or more devices in the network. Work items include importing, synchronizing, or changing configurations on devices. Also included are operating system upgrades on supported devices. A unit of work is the envelope that defines one or more tasks that proceed through the workflow engine. The workflow engine is what creates and determines approvals, scheduling, execution, and status of work items.

A unit of work can have one or more *approval tickets* associated with it. When a user submits a task, the user might need one or more approvals to release the unit of work, based on their group workflow settings. If it requires approvals, it generates tickets for the groups who have the rights to approve the unit of work. After its approval, the unit of work moves from the *pending approval* queue to the *work waiting to execute* queue.

Tasks are a subset of a unit of work. A UOW has one or more tasks associated with it. A worker server receives a task and processes it as a discrete set of operations. These operations include imports, synchronization, or modification of a device configuration. *A worker server decomposes units of work* after their submission into the workflow engine. Decomposition provides the ability for multiple worker servers to compete for and process these tasks. Tasks might be processed concurrently to improve execution performance or nonconcurrently to improve controllability.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Terminology descriptions (continued)

Policy

- A structure used to determine if device settings are compliant or non-compliant

Remediation

- A process that links non-compliant devices to change templates that can make the device compliant through change management

© Copyright IBM Corporation 2016

Terminology descriptions (continued)

A *policy* is a vendor-neutral means of analyzing a device to determine whether its settings are compliant or noncompliant with organizational or industry standard best practices. A policy is made up of other objects that allow it to analyze different vendor syntax.

Remediation is a process that links policies with change templates in the configuration manager. When a device is found to be noncompliant, a remediation is automatically prepared to repair the device configuration. This repair process might be configured to run automatically or through a user approval.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Describe the key benefits of IBM Tivoli Netcool Configuration Manager
- Explain the architecture and scalability
- Use specific terminology to describe the solution

© Copyright IBM Corporation 2016

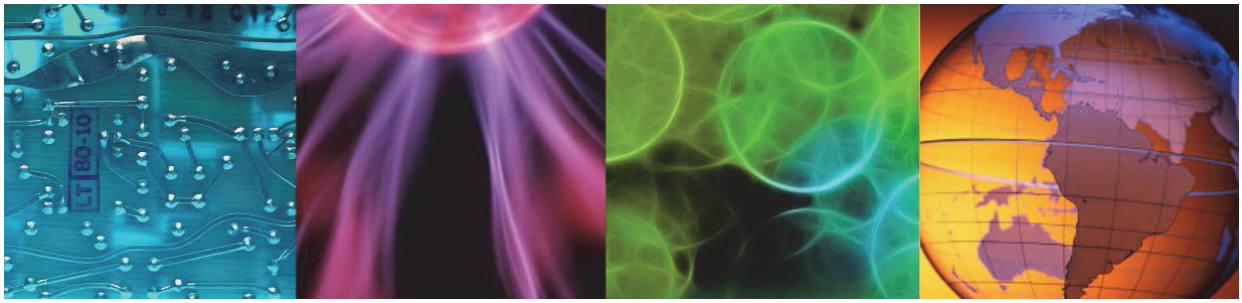
Summary

This unit provided an overview of the features of IBM® Tivoli® Netcool® Configuration Manager. It included an overview of its architecture and scalability. You also learned some of the terminology that is seen throughout the course.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

2 Configuration manager interface

Configuration manager interface



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this unit, you learn how to access the configuration manager interface and download it to your desktop with Java Web Start. You learn how to log in to the application and browse through the main portions of the interface.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Load the server virtual machine in VMware workstation
- Access and download the application
- Log in to the graphical user interface
- Navigate through the queue manager and resource browser

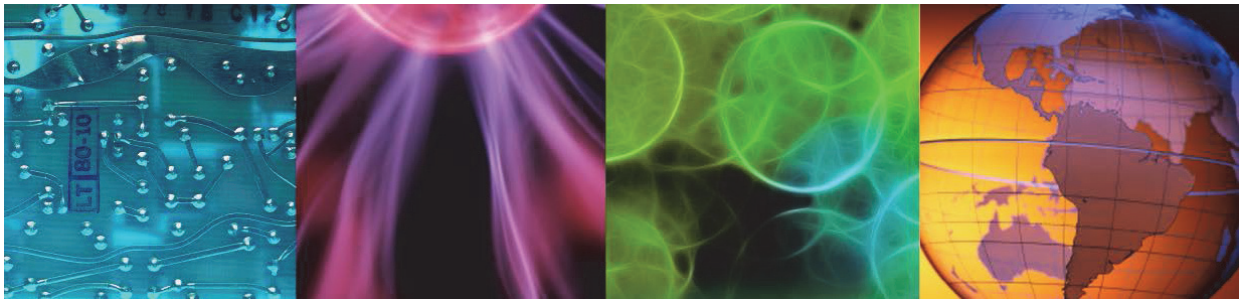
© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Accessing the application

Lesson 1 Accessing the application



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Log in to the client application
- Describe major features of the client application

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Client computer prerequisites

Configuration manager and compliance manager GUIs have the same requirements

Java runtime environment 1.6 or higher

Minimum of 192 MB of free memory

Access to a presentation server

Internet browser

© Copyright IBM Corporation 2016

Client computer prerequisites

The client software runs on Java Runtime Environment™ (JRE) 1.6 or higher. Any API integrations must use the IBM release of the JRE. For typical user access to the GUI, any JRE 1.6 is acceptable. You can obtain the current version of the JRE from this location:

http://www.java.com/en/

The client GUI requires a minimum of 192 MB of free memory. This setting is a default for the client memory allocation of Netcool Configuration Manager. You must increase this value to view large configurations from the client GUI. Juniper configurations typically require more memory. You can increase this value by accessing the system properties for Netcool Configuration Manager.

The client computer must be able to access the presentation server to download the client software. The default presentation port that the client must access is 16310.

An Internet browser is necessary for the initial download of the client software. You can use most versions of Internet Explorer or Mozilla Firefox for downloading the client and user account management.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Java Web Start

Is a centrally managed thick client, updates to the client are automatic

Protects the user from underlying XML complexities

Uses a Java Network Launching Protocol (JNLP) file to launch the application

You can find more information about Java Web Start at this website:

http://en.wikipedia.org/wiki/Java_Web_Start

© Copyright IBM Corporation 2016

Java Web Start

The client application is based on Java Web Start, which is a centrally managed thick client. The presentation server processes any upgrades that the client software needs. When you start the client GUI, it accesses the presentation server and requests the download and installation of any new software to the local cache.

The Web Start GUI protects the user from the underlying XML complexities that are associated with device modeling. XML is used extensively throughout the application, but users see little of it in practice.

Web Start uses a Java Network Launching Protocol file to download the application. This file is initially downloaded to the client, where you use it to find and download the correct Java files that the application needs to start.

To obtain more information about Java Web Start, refer to the website noted on the slide.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

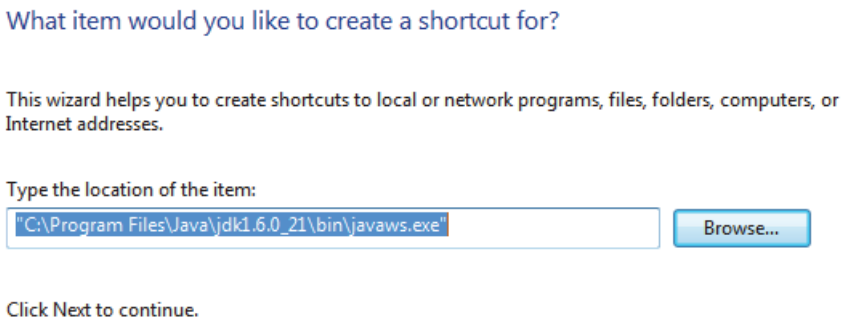
In previous releases of Netcool Configuration Manager, an icon was placed on the desktop of the user for future client launches. In the 6.4.2 release, this icon is not placed on the desktop. To create this icon manually, you can do the following steps:

- **Linux desktop systems**: Create a desktop launcher and add the following command:

  ```
  <location of 1.6 JRE>/javaws http://<ITNCM server
  location>:16310/soap/profiles/admin.jnlp
  ```

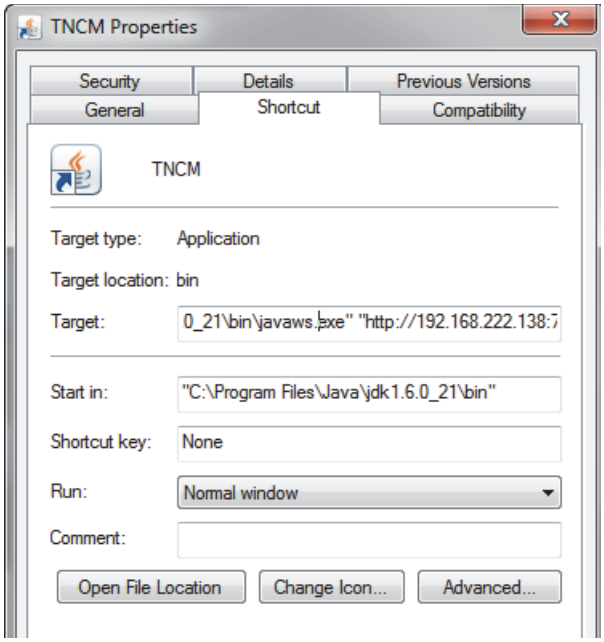
- **Windows desktop systems**: Create a shortcut by using the following command:

  ```
  “c:\<location of 1.6 JRE>\javaws.exe”
  ```



Edit the shortcut properties and add the following string:

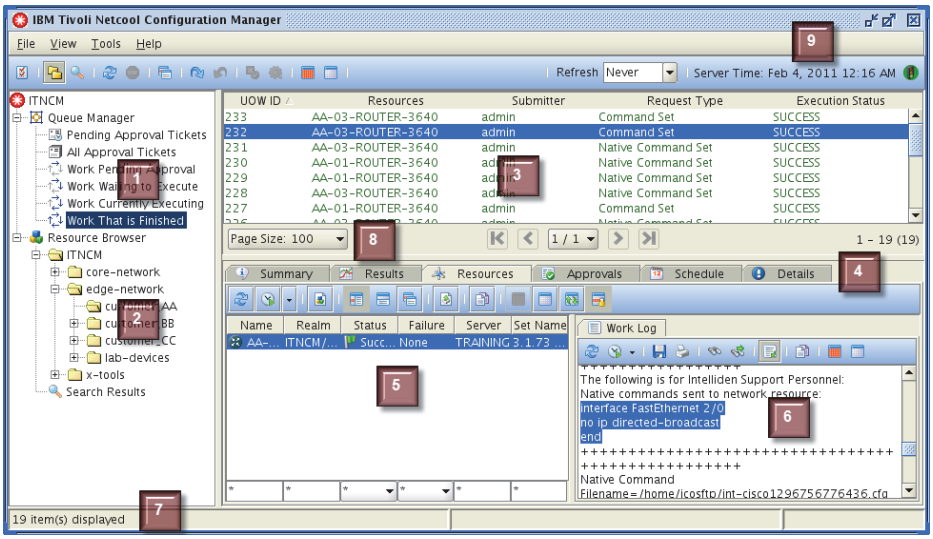
```
“http://<ITNCM server location>:16310/soap/profiles/admin.jnlp”
```



© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuration manager GUI



© Copyright IBM Corporation 2016

Configuration manager GUI

The following sections are part of the Netcool Configuration Management user interface:

1. The queue manager shows the tasks that are in various states in the workflow. The tasks that are visible are the only ones that the user is authorized to see. These tasks are the ones that are awaiting approval, waiting to execute, currently executing, or complete.

2. The resource browser is a customer-defined layout of the network devices that are under management. The users can peruse the devices that they are authorized to see. They can see the configuration history of the device and the keystroke logs for any of their terminal interactions.

3. In this window, you can see the items that are selected from either the queue manager or resource browser. You can order the items by selecting column heads and can group them for easier viewing.

4. By selecting an item in the display window, you can inspect it for more details. These details depend on whether you are viewing a unit of work or a network resource.

5. In this detailed example, the user is looking at the network devices that are part of a unit of work.

6. By selecting the device in the unit of work, the UOW log is presented, and details of what happened are available. In this example, a configuration change was applied that modified the **Fast Ethernet 2/0** interface and disabled the command **ip direct-broadcast**.

7. Anytime a search is performed in the queue manager or resource browser, the total number of items are noted. If the search is a large one, it is limited to the page size set in item 8.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

8. The page size value helps with GUI performance, some searches return many results. To the right of the page size is the total search number, in this case, **1-19 (19)**. In larger searches, you might have a number like **1-100 (3230)**. In this case, you have a total search result of 3230 items, of which, items 1 - 100 are represented.

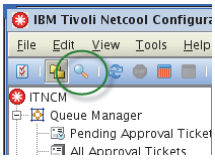
9. The server time is important because this time defines when items move from the *work waiting to execute* queue to the *work currently executing* queue. You can set the time to any time zone.

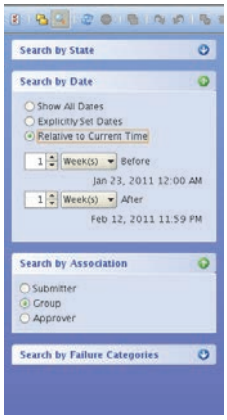
© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Advanced searching

The advanced search feature depends on the context





Queue manager

Resource browser

© Copyright IBM Corporation 2016

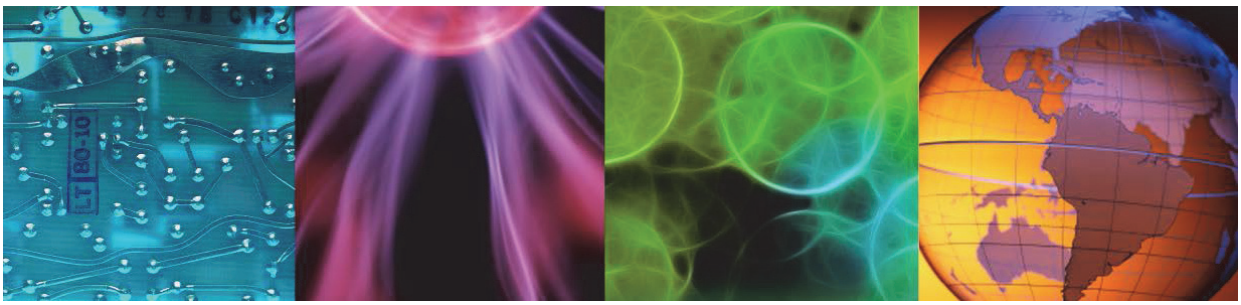
Advanced searching

To use the advanced search function, you select the appropriate icon to instantiate a search pane. This pane depends on what is represented in GUI. If you are looking at devices in the resource browser, clicking the advanced search icon presents browser-specific resource filters. You also use this behavior when viewing *units of work* in the queue manager.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 The queue manager

Lesson 2 The queue manager



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the queue manager
- Describe workflow states

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Queue manager overview

Use for viewing what has or what will happen

Use for viewing the units of work (UOW)

- By default it can only see groups work

Shows the details of a selected UOW

- Approvals
- Network devices touched
- Execution details

Searchable by time, state, association, failure code

Enabled for right-clicking items

© Copyright IBM Corporation 2016

Queue manager overview

You use the queue manager to obtain a global view of what is happening on the network. You can see the changes that are planned and the ones that already occurred. By default, a user can see only the work that the user's group submits. *Units of work* are the objects that the queue manager tracks. UOWs are the objects that users select to obtain more information about an operation that someone submitted. *Units of work* have detailed information, which includes these items:

- If approvals are needed, who approves the UOW and when it occurs
- The network devices that the UOW operates against
- Details about the execution, including its success or failure and a detailed log of the interaction with the device

You can search UOWs by time, their execution state, and failure codes. You can right-click the items in the GUI to access context-specific menus.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Workflow states

Work pending approval: Items that require management approval

Work waiting to execute: Approved tasks waiting for scheduled window or worker server resources

Work currently executing: Tasks worker servers are actively processing

Work that is finished: Units of work that have been processed or cancelled for various reasons

© Copyright IBM Corporation 2016

Workflow states

Units of work are the objects that move through the workflow. A unit of work is always in 1 of four states, as shown on the slide. A unit of work can move forward only to the next workflow state, it cannot move back.

When a group requires approvals, the UOWs submitted by that group begin their lifecycle in the *work pending approval* queue. Here the UOW waits for either approval or rejection by a user with the appropriate rights. Eventually, a UOW expires and is moved to *work that is finished* when it is not approved or rejected.

If approvals are not required for a user who submits a UOW, then it moves directly into the *work waiting to execute* queue. *Units of work* in the *work waiting to execute* queue are waiting for two conditions:

- The UOW has a window start time. It must wait in this queue until the window start time is equal to or after the server time.
- After the window start is reached, the UOW must wait for a worker server to pick it up and begin processing it. Moving the work to the *work currently executing* queue depends on how busy the worker servers are and the priority of the UOW.

Units of work in the *work currently executing* queue represent worker servers that are processing the tasks. These UOWs are most often directly accessing network devices.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The *work that is finished* queue has the UOW that is finished processing or is prevented from running for the following reasons:

- Scheduling issues
- Rejections by approvers
- Dequeuing by authorized persons

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Execution status

Finished executed states

SUCCESS, PARTIAL_SUCCESS, FAILURE

Finished unexecuted states

REJECTED, DEQUEUED, expired

UOW ID	Type	Resources	Submitter	Request Type	Execution Status
237	UOW	AA-02-ROUTER-3640	engineer	Command Set	expired
236	UOW	Multiple Devices	operator	Command Set	DEQUEUED
235	UOW	AA-02-ROUTER-3640	operator	Command Set	REJECTED
234	UOW	Multiple Devices	engineer	Configuration Synchronization ...	PARTIAL_SUCCESS
233	UOW	AA-03-ROUTER-3640	admin	Command Set	SUCCESS
232	UOW	AA-03-ROUTER-3640	admin	Command Set	SUCCESS
231	UOW	AA-03-ROUTER-3640	admin	Native Command Set	SUCCESS
230	UOW	AA-01-ROUTER-3640	admin	Native Command Set	SUCCESS
229	UOW	AA-01-ROUTER-3640	admin	Native Command Set	SUCCESS

© Copyright IBM Corporation 2016

Execution status

Units of work that reach the *work that is finished* queue are in one of the following execution status states. These states imply that the UOW began running its tasks.

- SUCCESS means that all tasks in the UOW completed successfully.
- PARTIAL_SUCCESS means at least one of the tasks in the UOW had a failure and at least one of the tasks completed successfully.
- FAILURE means that all tasks in the UOW failed.

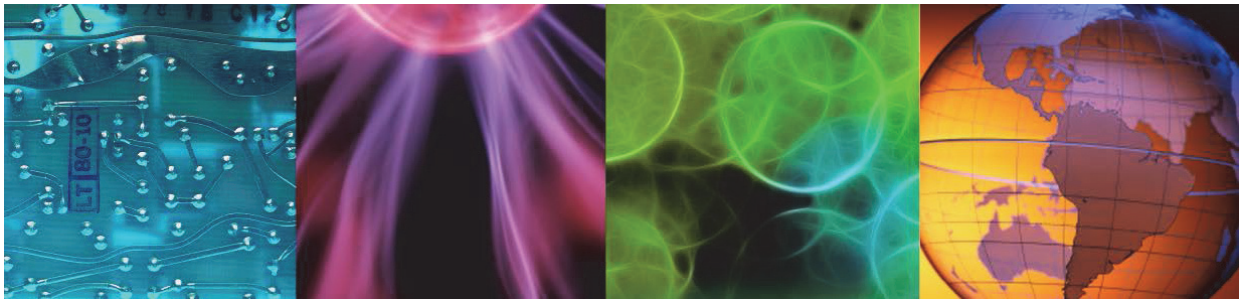
Some *units of work* do not reach the *work that is executing* queue:

- REJECTED UOWs were rejected by an approver. These *units of work* move directly from the *work pending approval* to the *work that is finished* queue.
- DEQUEUED UOWs were dequeued before they reached the *work that is executing* queue. Only authorized users can dequeue work. Rare cases exist where a *running* UOW was dequeued, and any unstarted tasks inside it are *canceled*.
- EXPIRED UOWs have a *window start* and *window end*. If a UOW does not begin processing by the *window end*, it becomes an expired UOW.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 The resource browser

Lesson 3 The resource browser



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the resource browser
- Describe the concept of a realm

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Resource browser overview

Use to view a set of managed objects

Use to view the authorized network resources under management

Shows details of the history of a network device

- Current and historical configurations
- Keystroke logs of terminal sessions
- UOWs that have processed against device
- Compliance information

Similar to Windows File Explorer

You can search resources by resource type, realm location, name, VTMOS, configuration contents

You can right-click items

© Copyright IBM Corporation 2016

Resource browser overview

The resource browser is where the objects, usually network devices, are managed inside Netcool Configuration Manager. You can use the resource browser to view the network objects that you are authorized to see. You can see the current and historical states of a device, and you can see these types of data:

- The current and historical configurations that are on the device
- The keystroke logs of interaction with the device by users
- A listing of the *units of work* that processed against the device
- Compliance information about the configuration state of the device

The resource browser uses a Windows File Explorer metaphor for its interaction. You can search for devices by realm location, VTMOS criteria, name, and configuration contents. You can right-click a device to select it and use the associated menu actions.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Realm structure

The key feature of the resource browser

A customer-defined file system

- Organize the network to align with organizational needs

A way to define security settings for viewing the network

A structure that contains realms:

- Network resources, devices
- Generalized resources, data on how to process devices
- Command set, templates
- Subrealms

© Copyright IBM Corporation 2016

Realm structure

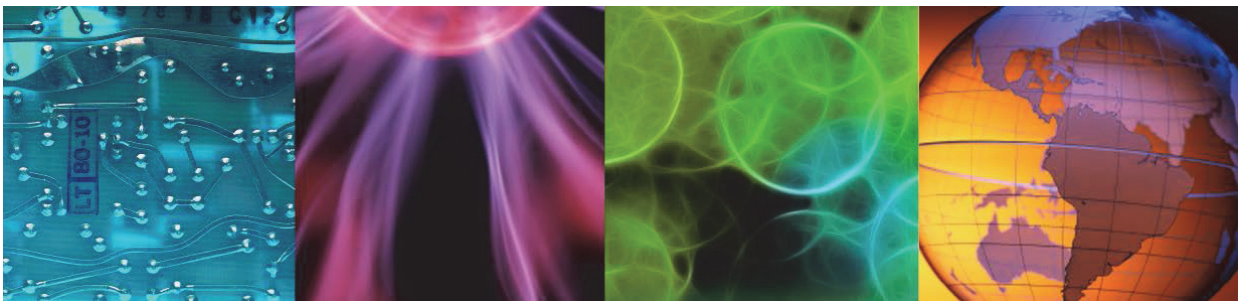
The key feature of the resource browser is the realm structure. It is a customer-defined structure that reflects how the customer views the network that is under management. It is a key piece of the security that controls access.

The realm structure is like a shared file system on a network drive. The realm is a series of folders for organizing the network devices into a hierarchical structure. This structure gives users an easy means of finding specific devices. Realms are used to organize other objects, like command sets, generalized resources, and other realms.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 User preferences and help

Lesson 4 User preferences and help



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe user preferences
- Describe online help

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

User preferences

Determine when refreshes of the GUI data occurs, affects GUI performance

Filter objects in the GUI

Disable wizard panels, disable unnecessary questions asked in UOW submissions

Enable email notifications

Update user and time zone information

© Copyright IBM Corporation 2016

User preferences

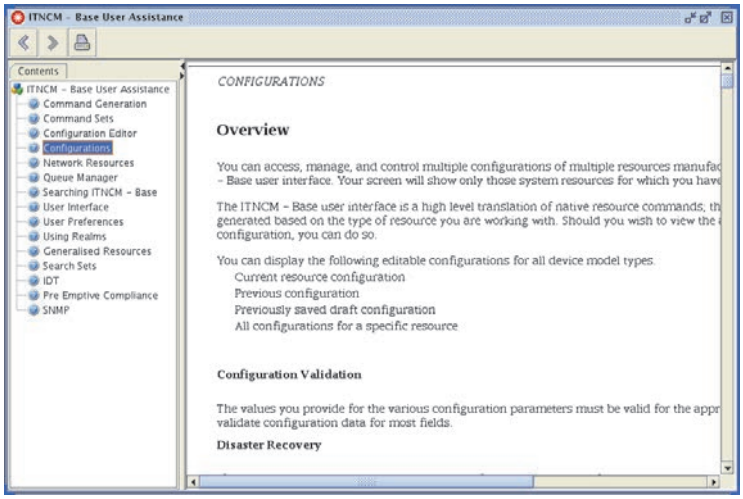
The user preferences define many of the features and interactions you have with the GUI. Some of the interactions include:

- Defining when the screen should refresh its data
- What objects should be hidden by default
- Disabling some of the questions that are asked when setting up *units of work*
- Enabling the receipt of email as UOWs move through the workflow
- Updating user information and the time zone you want the GUI to use

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Online help

Accessible through menus or the F1 key



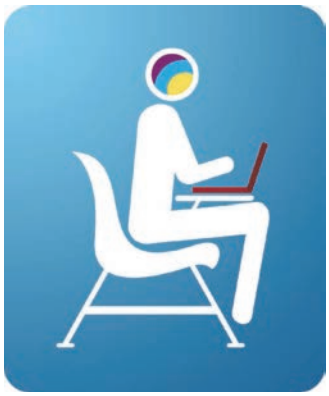
© Copyright IBM Corporation 2016

Online help

The application also has online help available by using the top-level menus or pressing the F1 key. The help is context-sensitive so it attempts select a help topic based on what is selected on the screen.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises



© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Load the server virtual machine in VMware workstation
- Access and download the application
- Log in to the graphical user interface
- Navigate through the queue manager and resource browser

© Copyright IBM Corporation 2016

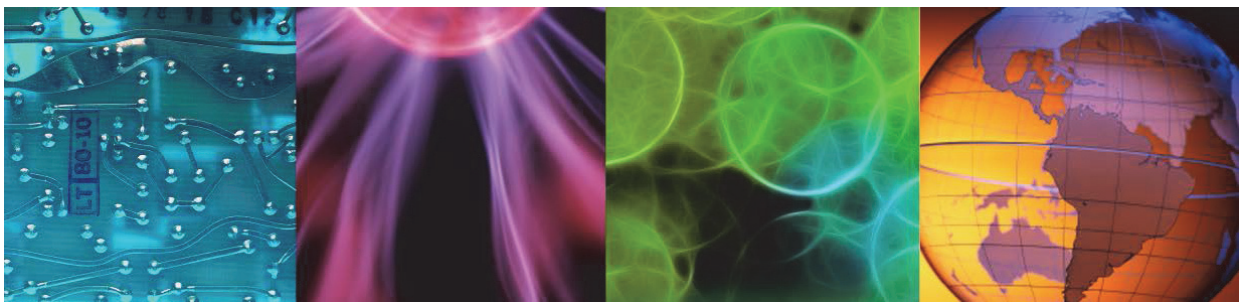
Summary

Now that you completed this unit, you understand how to access the configuration manager interface and download it to your desktop by using Java Web Start. You can log in to the application and know how to browse through the main portions of the interface.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

3 Single change configuration management

Single change configuration management



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit describes how to import devices into the solution. After you import devices and put them under management, you learn how to manipulate their configurations and submit *units of work*. You learn how to connect to the command-line interfaces of specific devices. Finally, you learn about device synchronization so that the network and the database can have a common view of the device's state.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Discover and import network devices into database
- View a configuration with the configuration editor
- Modify a configuration with the configuration editor
- Make direct device connections by using the IBM Device Terminal (IDT) to open VTY sessions
- Describe device synchronization between Tivoli Netcool Configuration Manager and device configurations

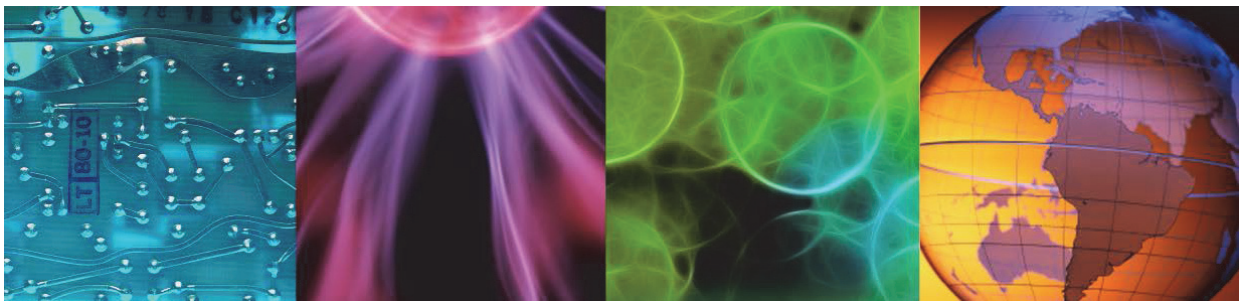
© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Discovering network devices

Lesson 1 Discovering network devices



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe what is meant by discovery
- Describe how a device is discovered
- Describe the methods available to discover a device

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

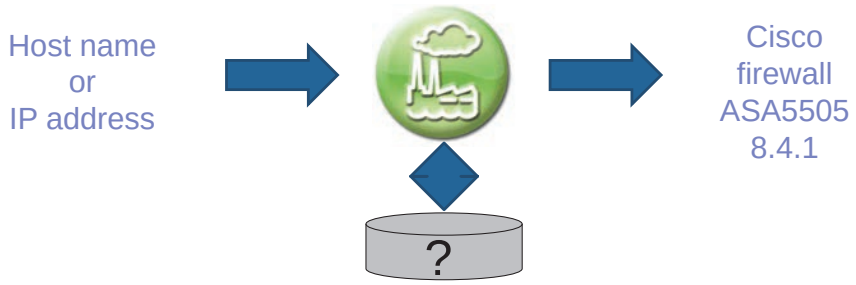
Discovery

What is discovered:

- The vendor, type, model, and operating system (VTMOS)
- Use VTY session to parse directly from command line

What is not discovered:

- The existence of network devices



© Copyright IBM Corporation 2016

Discovery

When you run Netcool Configuration Manager, you discover the vendor, type, model, and operating system (VTMOS) of a host name or IP address. Using this information, you can decide how to model the configuration of the device and access it.

With Netcool Configuration Manager, you do not search through the network for devices that are attached to it.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Discovery process

Discover the VTMOS of a host name or IP address

Use the credentials defined in the realm structure to attempt logging in to devices

- Having devices already pointing to the TACACS server is helpful

Worker servers need access to devices

Create two units of work

- Run Autodiscovery to log in to device and determine VTMOS
- Import Configuration to import the device, based on the discovered VTMOS

© Copyright IBM Corporation 2016

Discovery process

Discovery requires a resolvable host name or IP address to access a device through Telnet or SSH. You use the credentials that are defined in the realm structure to access devices. It is useful if the devices to discover are already in TACACS. Creating a specific account for Netcool Configuration Manager is also a good practice. Before discovery begins, it is appropriate to define an authentication resource in the base realm for the implementation. This authentication resource has the user name and passwords that are needed for device access.

Worker servers are the agents that go to each device, authenticate with the appropriate credentials, and then determine what device it is. Worker servers must be able to route to the device with the host name or IP address.

The successful discovery process creates two units of work. The first unit of work is designed to determine the VTMOS of the device in question. The worker server hosts a series of flat files that are used to determine this VTMOS. If successfully discovered and the correct device drivers are installed, it creates a second unit of work to import the device into the database. If the first UOW fails, it does not create the second UOW.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Discovery methods

Discovering devices by using the GUI

- Simple
- Not practical with many devices

Discovering devices by using the BulkLoader

- Uses a comma delimited text file with realm and device host names
- Use to complete a brute force search of a subnet
- Can have unknown (undiscoverable) devices imported into Tivoli Netcool Configuration Manager

© Copyright IBM Corporation 2016

Discovery methods

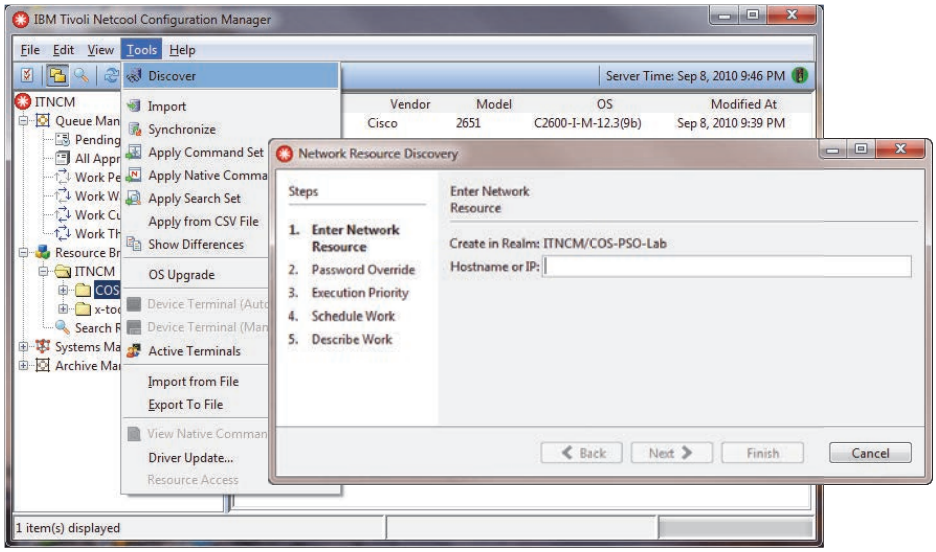
You can use two methods to import devices. The simple method is to use the GUI to start the discovery of a single host name or IP address. This method starts a UOW wizard that gathers the necessary information to start the unit of work. While simple, it is cumbersome when you have many devices.

The second method uses a tool that is built with the application programming interface (API). This method uses a comma-delimited file of host names and realm locations to import many devices. No known limitation exists to the total number of devices that you can define in this file.

The BulkLoader can start a brute force discovery of a subnet by attempting to access and discover each IP address. The BulkLoader allows the creation of unknown devices. In this context, an unknown device is a host name or IP address that the BulkLoader attempts to discover but cannot determine its VTMOS. This list of unknown devices is useful in an initial discovery of the network. You can export the list from the application and use it to help troubleshoot why they were not discovered.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Starting the GUI discovery process



© Copyright IBM Corporation 2016

Starting the GUI discovery process

You can start the GUI-based discovery process by selecting **Tools** > **Discover**. To run the discovery tool, you must have ***Add*** rights for resources for the realms that devices are imported to. You also need the ***Import*** activity that is associated with your group.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

BulkLoad discovery

BulkLoad uses the Tivoli Netcool Configuration Manager API to set up autodiscover units of work.

- Typically run from a worker server as a command-line utility
- Can start from an external server as API program

Uses a comma-delimited file

```
#realm,hostname,vendor,type,model,OS
ITNCM/edge-network/customer_CC,CC-01-ROUTER-3640,*,*,*,*
ITNCM/edge-network/customer_CC,CC-02-ROUTER-3640,*,*,*,*
ITNCM/edge-network/customer_CC,CC-03-ROUTER-3640,*,*,*,*
ITNCM/edge-network/customer_BB,BB-03-SWITCH-2960,*,*,*,*
```

Is helpful in building a realm structure

You can set individual parameters on each device

© Copyright IBM Corporation 2016

BulkLoad discovery

The BulkLoader uses the Netcool Configuration Manager API. You can run the software from any server that has the correct Java runtime environment and Java archive files installed. However, the worker server has all the necessary files to run the utility. Because a presentation server includes a worker server, it is most convenient to start the utility there. The BulkLoader sets up *units of work*. When the UOWs are created, standard Netcool Configuration Manager procedures run the tasks.

The input file has the following required format: realm, vendor, type, model, OS. To start the discovery process, replace the VTMOS parameters with asterisks. The BulkLoader is configured with VTMOS information, when needed. Netcool Configuration Manager must import devices as those specific types when those values are defined.

The BulkLoader is useful in developing the realm structure of your implementation. It can be run so that any realms that currently do not exist are created as devices are imported in the application. Creating realms with the BulkLoader is not a requirement but saves time for the administrator.

Keep capitalization consistent. The BulkLoader creates a **CORE** realm beside a **core** realm.

You can use the BulkLoader to set various hardcoded parameters on each individual device. This process is outside the scope of this course.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Discovery in Tivoli Network Manager

IBM Tivoli Network Manager creates a seed file

- Currently only a list of host names

Seed file is requested by configuration manager for import

- Tivoli Netcool Configuration Manager uses autodiscovery technology to determine VTMOS

Imports devices into a default realm that is based on a domain name

- Move devices to appropriate realms

© Copyright IBM Corporation 2016

Discovery in Tivoli Network Manager

Discovery of devices is where some overlap between IBM Tivoli Network Manager and Netcool Configuration Manager exists. With IBM Tivoli Network Manager, you discover the existence of a device on the network. With Netcool Configuration Manager, you discover the VTMOS. Currently, you can configure IBM Tivoli Network Manager to provide Netcool Configuration Manager a seed file of devices by domain, which loosely equates to a higher-level realm. This seed file contains a list of host names of devices that Netcool Configuration Manager does not manage. Netcool Configuration Manager must still discover the VTMOS of the device.

The devices are imported into a default realm and need to move to a location based on the realm structure. This behavior is acceptable when IBM Tivoli Network Manager attempts partial discoveries, that is, only a few devices need to be moved to final realm locations. However, if you need a full import of the network, the better solution is to use the BulkLoader.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Enhancing discovery

Worker servers discover devices

- Using a series of flat files to determine VTMOS

Files include the following information:

- Commands that generate model and OS information about devices:
  - show version, operation version, show system, about, get system
- Regular expressions to determine vendor
- Regular expressions to determine model and OS
- Mapping files that define type, based on model information

You can make some common changes

- Updating mapping file for model numbers
- Adding new devices that belong to a vendor

Changes to one file require synchronization across workers

© Copyright IBM Corporation 2016

Enhancing discovery

The worker server is the agent that discovers the VTMOS of a device. It has a series of flat files that it uses to accomplish this task. These flat files support the following functions:

- One file contains a number of device commands. These commands generate device responses that provide model and operating system information. It is a multi-vendor list of commands. It submits all the commands to the device. Most commands create command errors but one or more produce searchable results.
- Another file uses regular expressions to search through the results of the device commands to determine the device vendor.
- A third file takes this vendor information and uses regular expressions to determine the model and operating system of the device.
- The final flat file is used to map the device, model, and operating system data to a device type, for example, is the device a router or a switch.

You can enhance or extend the discovery process to cover new devices that are not typically discoverable by the default solution. You can modify these flat files to account for a new vendor or model. If changes are made to these files on one server, you must manually replicate the changes to all the other servers in the solution.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Stand-alone discovery

Can run the discovery process can outside of the solution

- Executable files are located on worker servers

Can run as a Java program

Takes a flat file as input

- Login credentials
- Host names, IP addresses, and subnets

Does not import device, but logs in and submits show commands

Outputs BulkLoad file with VTMOS details

```
#realm,hostname,vendor,type,model,OS
ITNCM/edge-network,CC-01-ROUTER-3640,Cisco,Router,3640,*12.2*
ITNCM/edge-network,CC-02-ROUTER-3640,Cisco,Router,3640,*12.3*
ITNCM/edge-network,CC-03-ROUTER-3640,Cisco,Router,3745,*12.4*
ITNCM/edge-network,BB-03-SWITCH-2960,Cisco,Switch,2960,*12.2*CC
```

© Copyright IBM Corporation 2016

Stand-alone discovery

The device VTMOS discovery can run as a process that is independent of the architecture. It can start from a worker server with the libraries in the $ITNCM_INSTALL_HOME/autodiscovery file system. It uses the JRE in the $ITNCM_INSTALL_HOME/jre to run its code. If necessary, you can move the necessary files to a separate server and run them from there. A Java runtime environment must be installed on this server.

This utility takes a flat file as input. It includes sets of credentials and a list of host names or IP addresses in which to discover the VTMOS.

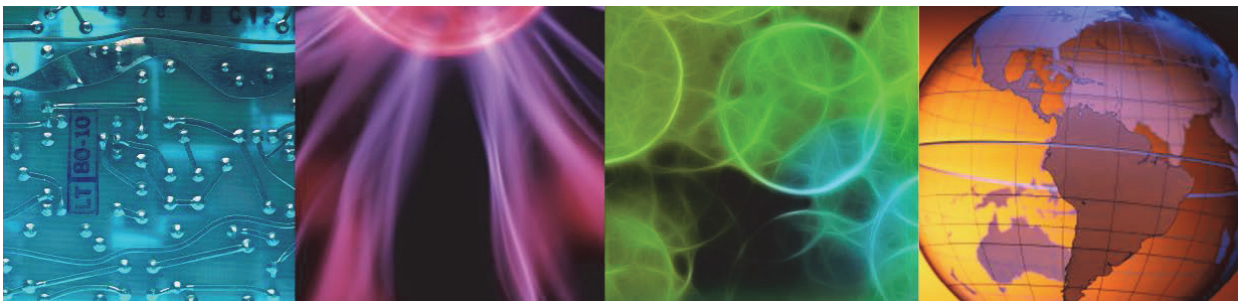
Unlike normal discovery that creates *units of work*, this discovery process just adds the discovered devices to a BulkLoad file that defines VTMOS information. This behavior differs from the discovery BulkLoad file that used *,*,*,*.

This tool helps people learn ahead of time of any inconsistencies between what they think their inventory is and what it is. In the example on the slide, the CC-03-ROUTER-3640 name implies it is a Cisco 3640 when it is a Cisco 3745.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Navigating the configuration and changes

Lesson 2 Navigating the configuration and changes



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe a SmartModel
- View a configuration in the GUI
- Apply a change to a device

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

SmartModels and standard support

Three levels of support for devices

SmartModel configurations

- Text-based configurations converted into XML documents

SmartModel SNMP

- SNMP MIBs converted into SmartModel configurations

Standard support

- Text-based configurations managed as raw text
- Binary configurations backed up as binary files

© Copyright IBM Corporation 2016

SmartModels and standard support

Netcool Configuration Manager uses a concept that is called SmartModels to convert the native language of a device into XML. With this technology, you convert device configurations from their original syntax into a vendor-neutral format. With vendor-neutral formats, you can create standardized tools and features for all vendors. However, modeling is a task that is done after vendors release their hardware and operating systems. To be as flexible as possible, you can support devices that are modeled or not modeled.

Netcool Configuration Manager supports two types of SmartModels: CLI and SNMP.

SmartModel configurations use a text-based configuration file as the basis for an XML representation. Translators process the data between these two formats to create configuration changes.

SmartModel SNMP supports builds on simple network management protocol and a device-specific management information base (MIB.) It polls customer-specific data from the device and stores it in XML. SmartModel SNMP supports both GETS and SETS and can be managed like a CLI configuration.

Standard support, which does not use any modeling technology, is also available. It retrieves CLI configurations from devices and manages them as a simple text file. The device is still modifiable with configuration changes and restorations, but many functions are not available. You can also use standard support to retrieve and restore binary configurations.

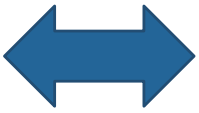
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

SmartModel configuration

Translates native data to XML representation

Data can be raw configurations or SNMP MIB data

```
interface FastEthernet0/0
 ip address 10.191.101.73 255.255.255.0
 speed 100
 full-duplex
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
```



```
<interface>
   <FastEthernet>
     <ARG.001>0/0</ARG.001>
     <ip>
       <address>
         <ARG.001>10.191.101.73</ARG.001>
         <ARG.002>255.255.255.0</ARG.002>
       </address>
     </ip>
     <speed>
       <_3100/>
     </speed>
     <full-duplex/>
   </FastEthernet>
   <FastEthernet>
     <ARG.001>1/0</ARG.001>
     <shutdown/>
     <duplex>
       <auto/>
     </duplex>
     <speed>
       <auto/>
     </speed>
   </FastEthernet>
```

© Copyright IBM Corporation 2016

SmartModel configuration

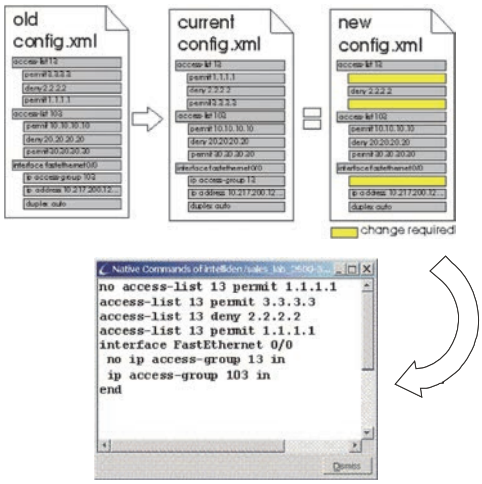
SmartModel technology is designed to take a text-based configuration language and convert it into an XML representation. SmartModels use a series of translators, which convert CLI or SNMP formats into XML and vice versa.

This conversion creates a fully tagged representation of the configuration. The additional metadata in this representation supports easier processing when compared to raw CLI or SNMP data.

The XML representation is a useful format for computers to process, but it requires configuration and greatly increases the size of the file. An architectural suggestion is to have worker servers as close to the database as possible. The data between a worker and the device is always in CLI or SNMP format. However, the data between worker and database is in XML. You can expect an increase of three times the amount of space for the XML representation and the need for more bandwidth.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Intelligent management: version restoration



Each versioned configuration has full XML representation

XML is useful for determining differences

You can convert XML differences into CLI differences

Changes are merged into the configuration

No reboot is needed

© Copyright IBM Corporation 2016

Intelligent management: version restoration

SmartModels support intelligent configuration management. The schema that is defined to model the configuration can include knowledge about how to format, order, and submit the configuration.

For example, Cisco devices merge configurations into the running memory of a device. Merging commands supports only adding and modifying commands, not deleting. Restoring a configuration requires the engineer to know what was deleted, modified, and most importantly, added to the configuration over time. Unless that is known, true restoration of a Cisco configuration requires pushing the configuration into nonvolatile memory and rebooting the device.

SmartModel technology supports the following capabilities:

- Configuration comparisons with exacting precision
- Ordering of commands
- Vendor-specific methods for adding, modifying, and deleting commands

With this programmatic capability, you can convert differences into CLI. This snippet of CLI includes the commands that must be added, modified, and most importantly, removed from the device to restore it to its previous state. Thus, using SmartModels preempts the requirement to restart the device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Viewing configurations

Tivoli Netcool Configuration Manager uses its own configuration editor GUI

Tivoli Netcool Configuration Manager enables graphical viewing of XML data

Configuration views are similar between vendor devices

You can always view the data in a native format

- Use security settings to limit user to partial configuration viewing

© Copyright IBM Corporation 2016

Viewing configurations

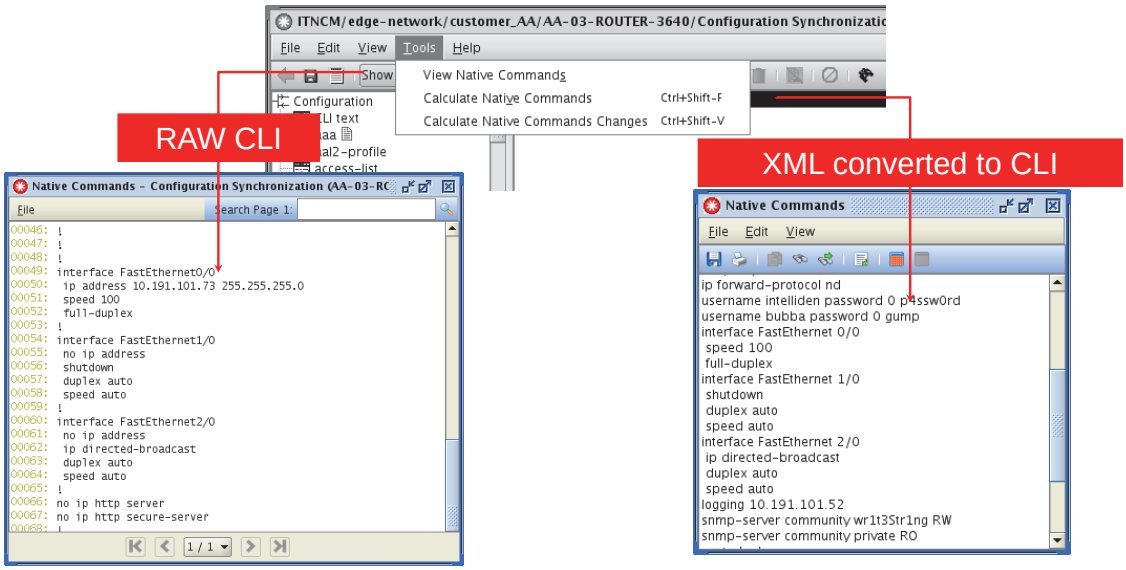
While XML is a great format for computers to process text data, XML can be overwhelming to view. The XML configurations are readable, when you know what you are looking at. However, much of the XML is hidden from view and visible only by the Netcool Configuration Manager editor.

The configuration editor is a graphical means of manipulating the configuration. Configurations are altered by adding, modifying, and deleting data through an editor. This editor supports similar manipulation across vendor types. For example, the ability to add a subinterface between a Cisco device and Juniper device is done with the same graphical icon. This similar manipulation approach supports the management of a multi-vendor network.

With the configuration editor, you can view the configuration in its original format. The original format can either be the raw configuration that you retrieve from the device or a conversion of its XML representation. The XML representation can have more security settings such that only portions of the configuration are viewable.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Viewing the CLI of the device



© Copyright IBM Corporation 2016

Viewing the CLI of the device

An XML version and a native version of each configuration exist in the database. With the appropriate security settings, you can view the native configuration. You can retrieve this raw configuration by selecting **Tools** > **View Native Commands**.

You can convert the XML configuration back into the native configuration by clicking **Tools** > **Calculate Native Commands**.

An example is shown on the slide. Note the differences between the Cisco router configurations:

- The CLI to XML translator removes the exclamation points to save space.
- Commands can be reordered in the XML representation. Order is not important for that portion of the configuration.
- The calculated native configuration supports cutting and pasting onto a device without issue.

Customers can implement the calculated native configuration to include view, add, modify, and delete rights on the XML configuration. These settings are defined at a group level. When users with these security settings view a configuration, they are restricted to viewing only a portion of the data.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configurations have a state

Current
- The current configuration

Draft
- A configuration that someone modifies but does not submit
- A configuration that someone modifies and attempts to apply but fails

Locked
- A draft configuration that is submitted to the workflow
- A draft configuration that is successfully applied

Versioned
- A configuration that is on the device

© Copyright IBM Corporation 2016

Configurations have a state

Every configuration in the database has a state. These states change over time as the configuration changes. A configuration starts as a current configuration or as a draft.

Initially, importing a device creates a new ***current*** configuration. From there, the list of configurations grows as the state of the device changes. A current configuration changes only into a versioned configuration.

A ***draft*** configuration is when someone edits a configuration, usually the current one, and saves it. If the user submits that change, the draft configuration changes its state to *locked*.

You cannot modify a ***locked*** configuration. This configuration either reverts to a draft configuration when a problem with its application occurs, or it remains locked after being successfully applied. If it is successfully applied, it creates a new current configuration in the database and the configuration remains locked.

A ***versioned*** configuration is a configuration that was, at one point, a current configuration.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modifying configurations

The configuration editor in Tivoli Netcool Configuration Manager uses graphical manipulation to prepare changes

The syntax for the commands is predefined

Rules for adding, modifying, and deleting are the same for all vendors

- Rules handle the correct formatting of the CLI
- Rules completely remove commands when deleting
- Rules place commands in the correct order, when relevant

You can review the actual CLI change that is submitted to a device

You can save draft configurations for review or to use later

- You must use good naming conventions
- You can have multiple drafts

© Copyright IBM Corporation 2016

Modifying configurations

The configuration editor creates on-demand changes to a device. The editor uses graphical manipulation of the configuration to prepare changes. The configuration editor has the syntax for the commands predefined in the SmartModel schema. The arguments that you input are checked for validity.

The configuration editor uses similar graphic manipulations, regardless of the vendors, with SmartModel technology. As commands are added, modified, and deleted, rules build a set of CLI commands to send to the device. The rules have these criteria:

- Correct formatting of commands to include more modifiers like **no** or **delete**
- Interactive commands that it uses when necessary. For example, the deletion of a VLAN often is followed by a confirmation question, like; Are you sure?
- Sequencing the commands so that it places them on the device in the correct order
- Deletion of a high-level command, which is preceded by the deletion of all the subcommands

The manipulations in the graphical editor are converted to the CLI so that users can double-check their modifications. Finally, you can save these changes to the configuration as a draft document that others can review. Using good naming conventions for each draft is essential for keeping changes in order.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Submitting the change to a device

Submitting a draft configuration creates a unit of work

- Can be applied immediately or scheduled

Submitting the draft locks the configuration

- You can not modify a locked configuration
- Completing the change application can be successful or unsuccessful
- Successful completion creates a configuration and leaves the draft locked
- Unsuccessful completion, for whatever reason, means that the configuration reverts to a draft

© Copyright IBM Corporation 2016

Submitting the change to a device

After you create a draft device configuration through the configuration editor, you can submit the changes. Submitting a draft configuration to a device creates a unit of work. You can submit the task to process immediately against the device or schedule it for a later date.

After you submit the draft, it becomes a locked configuration. Users can view locked configurations but cannot change them. Copies of these configurations are made by selecting **File > Save As**.

The configuration change is attempted. If it is successfully applied, that is, it generates no device error codes, a new configuration is created in the database and the draft remains locked.

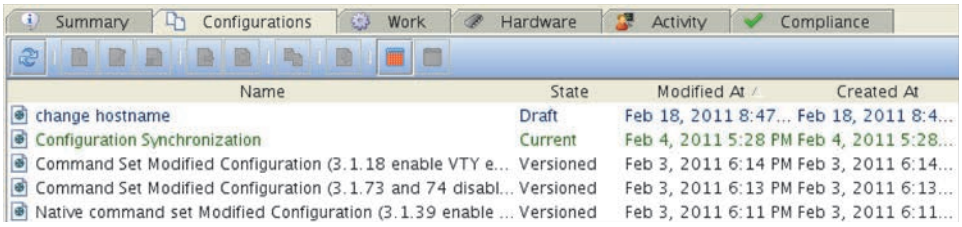
If a failure in the application process occurs, the locked configuration reverts to a draft. A failure can occur for several reasons:

- DNS issues
- Not able to connect or route to the device
- Authentication failed
- Bad syntax in the draft configuration

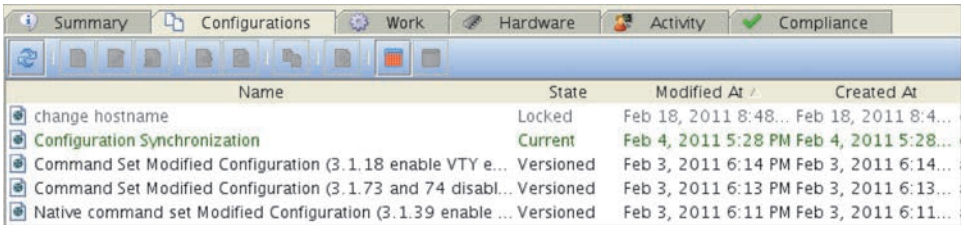
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Note: The following description is an example of a configuration change that goes through a series of state changes.

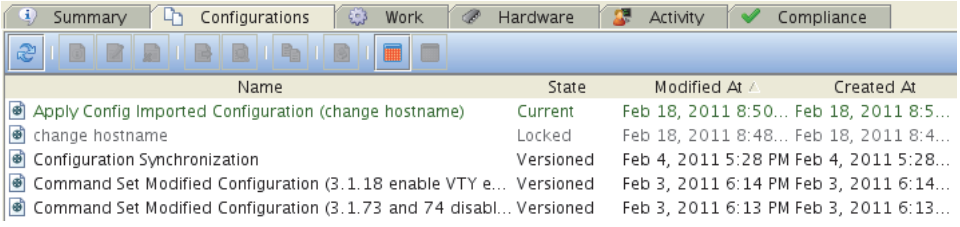
1. A draft configuration, called **change hostname**, is created from the current configuration. The name implies a device modification. The current configuration is created when a configuration synchronization notes a change to a device and imports the configuration to the database.



2. The draft configuration **change hostname** is submitted to the workflow engine and becomes a locked configuration.



3. The UOW runs to completion and successfully applies the changes in the **change hostname** draft. These configuration changes are retrieved by collecting the new configuration and placing it into a new current configuration called **Apply Config Imported Configuration (change hostname)**. Notice that the locked **change hostname** draft configuration remains locked.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Unit of work wizard

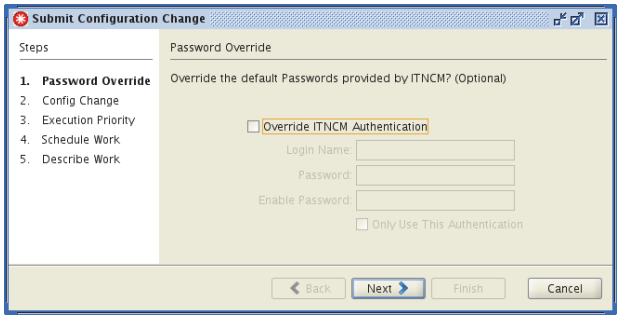
Submitting a unit of work starts a wizard

You use the wizard to help you prepare the unit of work

- Questions vary, based on the type of UOW

All wizards have these steps:

- Password override
- Execution priority
- Schedule work
- Describe work



© Copyright IBM Corporation 2016

Unit of work wizard

In this unit, you use the UOW wizard to submit your first unit of work. Using the UOW wizard ensures that you provide all the pertinent information that is needed to run the UOW. The steps in the UOW wizard vary depending on the UOW type. All units of work have these steps:

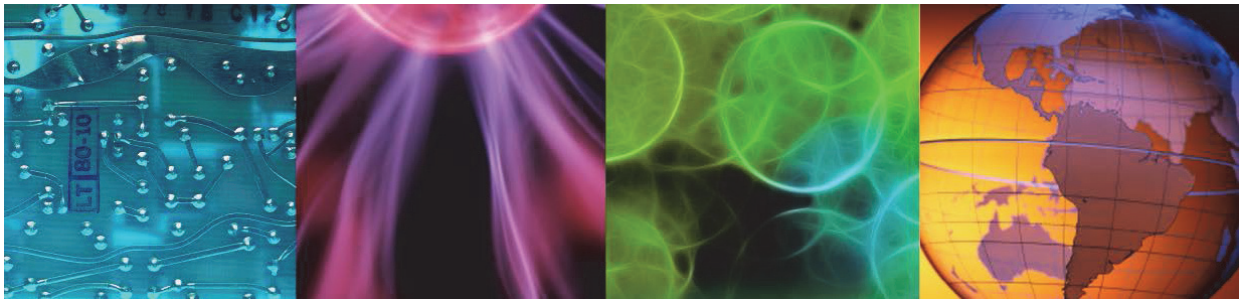
- Password Override to specify nonstandard credentials for device access.
- Execution Priority to set the priority of the UOW when you group with other tasks.
- Schedule Work to define when the UOW can begin processing.
- Describe Work to enter a description of the task.

You can set the user preferences to hide these steps. If you never use Password Override, you can configure the user preferences to hide that step.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Making direct connections to devices

Lesson 3 Making direct connections to devices



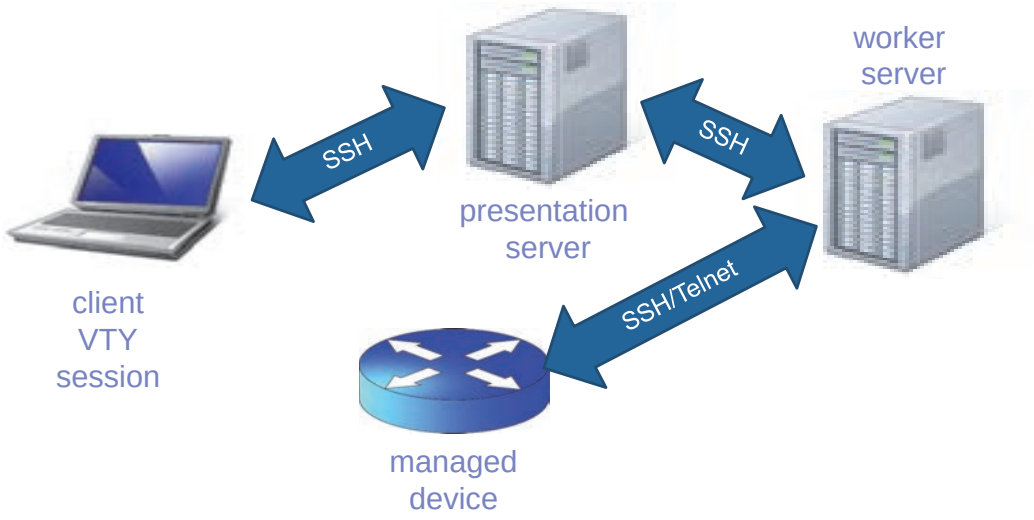
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Connect to a device with a device terminal
- Describe the keystroke logs

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

VTY connections over predictable paths



© Copyright IBM Corporation 2016

VTY connections over predictable paths

You can make direct connections to the devices whose configurations are under management. Customers can use the solution without changing any of their current device management practices.

The device terminal makes secure connections to devices through the Netcool Configuration Manager infrastructure to route to devices. Connections to devices always start from the client computer with a secure connection to the presentation server. It then creates another secure connection between the presentation server and a worker server. The worker connects to the device with either an SSH or Telnet connection, depending on what the device supports.

The device terminal has predictable paths to the device. With these connections, you can incorporate more security with firewalls and access lists.

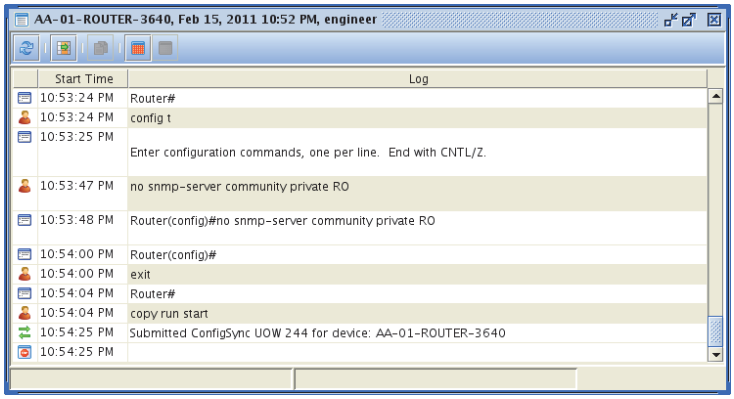
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Device terminals

Incorporate the Tivoli Netcool Configuration Manager security model

Enable automatic synchronization

Use full key stroke logging of a session



© Copyright IBM Corporation 2016

Device terminals

The device terminal has three main advantages over a simple terminal emulator.

1. The device terminal is integrated into the GUI and is employing the security model. Using this model means that if the user cannot see the device in the GUI they cannot connect to it.
2. You can configure the device terminal to synchronize devices when you use it to connect to them. You can configure this function to synchronize devices with different methodologies.
   a. Every time the device terminal connects to device
   b. When the device terminal detects that certain keywords are input by the user, for example **config t**.
3. The device terminal captures in a session log not only the keystrokes that the user enters, but also the response from the device. It stores the keystroke log in the database through the presentation server.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

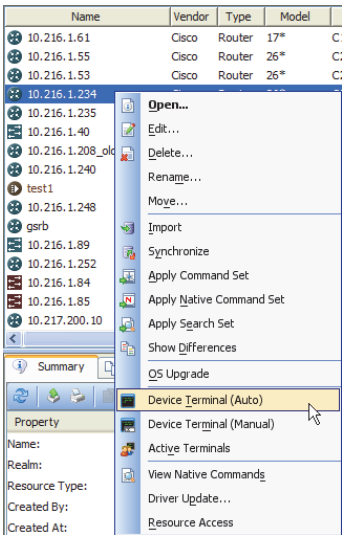
Accessing a device terminal

Device terminal access is granted to a group

- Option might be unavailable

You select and right-click a device

- Device Terminal (Auto) uses Tivoli Netcool Configuration Manager credentials to log on
- Device Terminal Manual requires that you supply logon credentials



© Copyright IBM Corporation 2016

Accessing a device terminal

To access a device in the device terminal, you select and right-click the device. You can access a device terminal that is assigned to your group. If your group does not have permission for a device, the option is not available.

Two options are available:

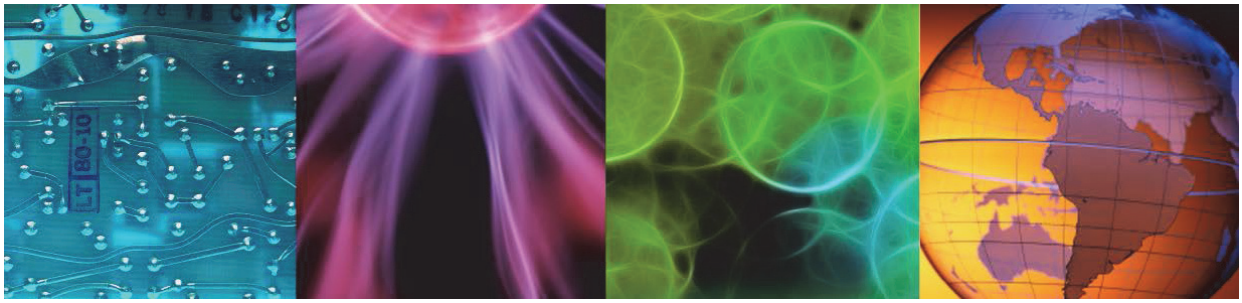
- **Device Terminal (Auto)**: You use the Netcool Configuration Manager credentials to access devices and automatically log in to the device. This device terminal option can be configured so that users are in *user* mode. Be sure to consider how TACACS affects automatic login. The Netcool Configuration Manager credentials can be set to *privilege* mode.
- **Device Terminal (Manual)**: You provide login credentials.

Note: The ***user*** mode is an initial, nonprivileged mode. Users can access basic functions on the router, but they cannot change the configuration state of the router or modify the operating system.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 Device synchronization

Lesson 4 Device synchronization



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the problems with out-of-date configurations
- Describe the methods to keep configurations synchronized with database.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Change management defaults

You can change only to devices that are synchronized

A synchronized device means that the configuration in a database matches the current configuration of a device

Some units of work fail on unsynchronized devices

- Command sets
- Configuration changes

UOW ID	Resources	Request Type	Failure Categories	Submitter
245	AA-02-ROUTER-3640	Configuration Change	Device Out of Synch	engineer
244	AA-01-ROUTER-3640	Configuration Synchronization (Device to ITNC...		engineer
243	AA-01-ROUTER-3640	Configuration Synchronization (Device to ITNC...		engineer
242	AA-02-ROUTER-3640	Command Set		pbcm-rem
241	AA-03-ROUTER-3640	Native Command Set	Host Unreachable	pbcm-rem

Page Size: 100 1 / 1 1 - 5 (5)

© Copyright IBM Corporation 2016

Change management defaults

The default behavior is to prevent changes on a device that is ***Out of Synch*** with the database. This situation means that differences exist between the running configuration on the device and the current configuration in the database.

The most vulnerable UOWs to the out of synch situation are configuration changes and command sets. The out of synch error occurs only at execution time. Such timing can become a frustrating situation for operators.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Synchronizing the network

Tivoli Netcool Configuration Manager does not process all changes

Other tools make scheduled changes

- Emergency break-fix scenarios occur
- Business policies are not enforced

Synchronizing the network occurs in three modes:

- Report differences only, what has changed?
- Resource to Tivoli Netcool Configuration Manager, update the data store
- Tivoli Netcool Configuration Manager to Resource, update the configuration of the device

© Copyright IBM Corporation 2016

Synchronizing the network

Having unsynchronized resources usually occurs when customers do not employ Netcool Configuration Manager for all changes to network devices. When Netcool Configuration Manager processes a change to the network, it collects the changed configuration and stores it in the database. This standard procedure implies that all changes are synchronized. However, other situations occur to create unsynchronized devices. These situations include:

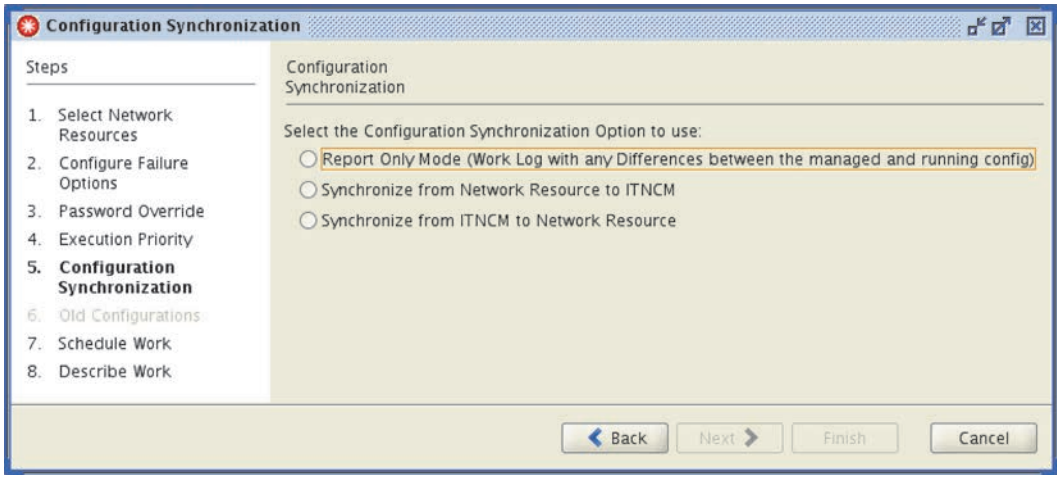
- Customers have other change tools that modify the network.
- An emergency occurs that requires immediate network changes that are outside planned change management activities.
- Management is not enforcing change management strategies, and users are employing nonstandard practices.

Netcool Configuration Manager has three modes to investigate differences between the database and the network:

- **Report Only Mode**: This mode notes the differences only between the running configuration on the device and the current configuration in the database. Differences are noted in the UOW log.
- **Synchronize from Network Resource to ITNCM**: This mode reports any differences between the running configuration on the device and the current configuration in the database. If it finds differences, it creates a configuration in the database. This option is what most users choose.
- **Synchronize from ITNCM to Network Resource**: This mode reports any differences between the running configuration on the device and the current configuration in the database. If it finds

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

differences, it modifies the device configuration to match that which is in the database. This selection is rarely used.



© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

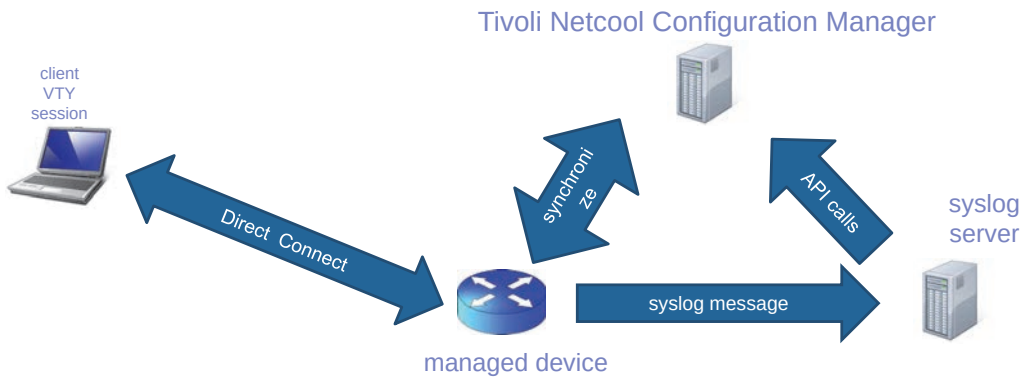
Options of synchronization

Recurring unit of work to regularly synchronize network

- Can be completed by device, groups of devices, and a realm

Out of Band Change synchronization (OOBC)

- Triggers synchronization, based on syslog messages



© Copyright IBM Corporation 2016

Options of synchronization

Keeping the database synchronized with the network is important. Synchronized devices ensure that Netcool Configuration Manager processes the changes successfully. Two methods exist to keep the database synchronized:

1. Some customers use a brute force approach, where a recurring UOW is set up to run on a nightly basis. This UOW communicates with every device in a domain and determines whether it needs a new configuration in the database. This synchronization is effective, but it generates a large amount of traffic.

2. Some customers use the *out of band change* (OOBC) daemon that operates with a syslog system. This daemon runs on the syslog server and inspects the syslog messages that network devices send. After the daemon detects a message that implies a change to the configuration, it uses the API to request that Netcool Configuration Manager synchronize that device. This synchronization is a more targeted approach and provides information about who is not using Netcool Configuration Manager. However, it depends on syslog messages, which are UDP-based, meaning they can get lost. Also, the network device must be correctly configured.

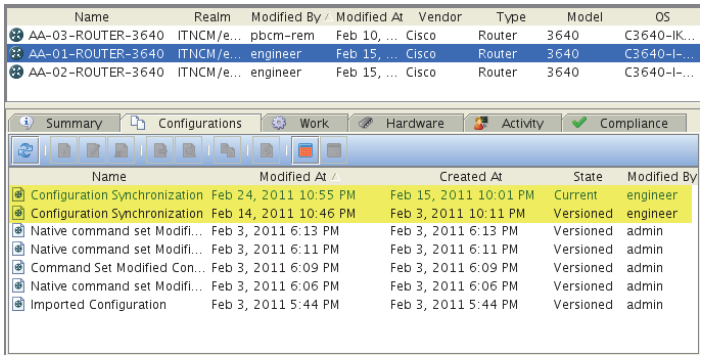
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Only changed configurations are saved

Saves new configurations only when differences are detected

- Otherwise, only the time stamp is updated

You can compare the difference between the *created at* and *modified at* timestamps



© Copyright IBM Corporation 2016

Only changed configurations are saved

Synchronization creates new configurations only when changes between the current configuration in the database and that which is on the device are different. If no differences exist between the two, the Modified At time stamp on the current configuration is updated. Comparing when a configuration is created and when it was last modified, you can, in general, determine how long a configuration remained the same.

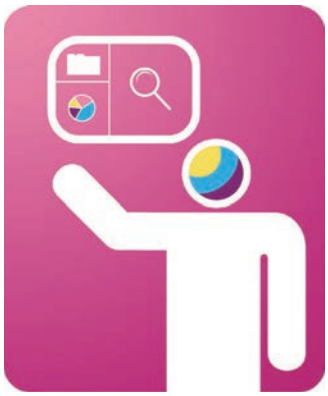
Note: Review the three configurations on the following device. A nightly synchronization UOW reviews all devices for changes around 10:00 PM

The following statements can be deduced from this example:

- Between Feb 3, 2011 6:13 PM, when the *native command set* configuration is created, and Feb 3, 2011 10:11 PM that evening, a change is implemented in the configuration. The cause of this change must be a user who accesses the devices outside of Netcool Configuration Manager (*out-of-band*).
- The configuration remains static until sometime between February 14 at 10:46 PM and February 15 at 10:01 PM, when another *out-of-band* change occurs.
- The device has the same configuration from February 15 through February 24 when the nightly synchronization UOW last checked it.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Instructor demonstration



© Copyright IBM Corporation 2016

Instructor demonstration

To help in the following exercises, the instructor discovers and modifies a device through the configuration editor and the device terminal by performing these tasks:

- Discovers a virtual Cisco router
- Views the imported device in a configuration editor
- Modifies the device by using a configuration editor
- Connects to the command-line interface and makes a change

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Discover and import network devices into database
- View a configuration with the configuration editor
- Modify a configuration with the configuration editor
- Make direct device connections by using the IBM Device Terminal (IDT) to open VTY sessions
- Describe device synchronization between Tivoli Netcool Configuration Manager and device configurations

© Copyright IBM Corporation 2016

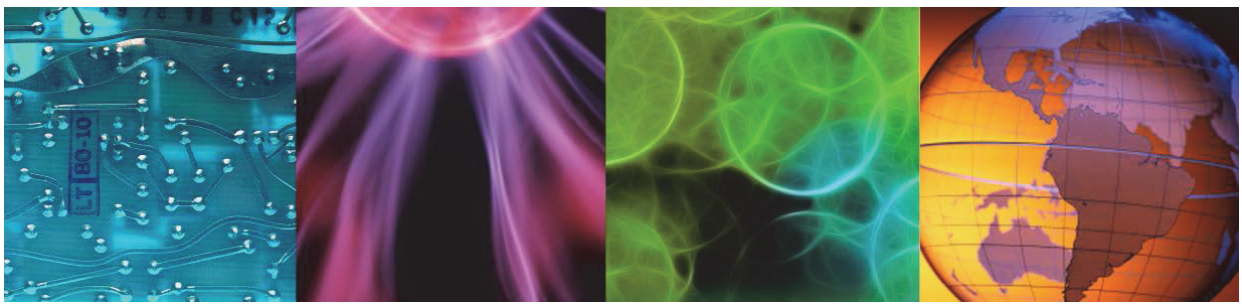
Summary

This unit described how devices are imported into the solution. After devices are imported and under management, you learned how to manipulate their configurations and submit units of work. You learned about how to connect to the command-line interfaces of specific devices. Finally, you learned about device synchronization so that the network and the database can have a common view of its state.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

4 Mass change configuration management

Mass change configuration management



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit, teaches you about the template technology of command sets. You learn about the types that are available and the ways that you can use these templates to make mass changes to the network. You are also shown a method to use a CSV file to define a mass change with customized data for each line in the file.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Describe what a command set is
- Describe the difference between command set types
- Explain how to create a command set
- Modify one or more devices with a command set
- Modify devices by using command sets and a CSV input file

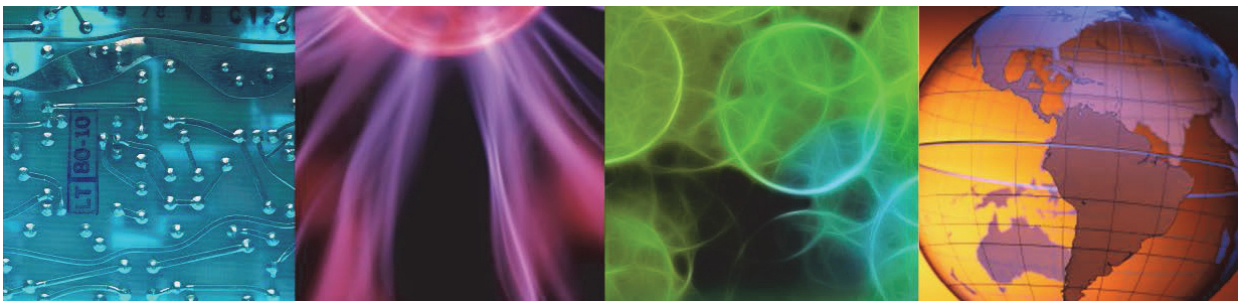
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Command sets

Lesson 1 Command sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe a command set and how they are run
- Describe the types of command sets and the advantages of each

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Change templates

Command sets are change templates

- One or more commands are needed to change a configuration

Use command sets for one purpose, like these examples:

- Create an interface
- Modify community strings
- Delete a user name

Apply one or more command sets to one or more devices

© Copyright IBM Corporation 2016

Change templates

You can build change templates that apply standardized changes to the network. These templates typically apply one or more commands to a configuration to start or stop a service that the network device is supplying.

Command sets are designed to accomplish one objective to add, modify, or delete one or more commands. Keeping the command set simple affords easier management. You can apply one or more command sets to a series of devices in a unit of work.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Vendor-specific commands

Command sets contain vendor-specific commands

Software filters for vendors

You can apply command sets across models and operating systems

- Banner commands are similar on all Cisco IOS devices
- Advanced features might be specific to an individual device

© Copyright IBM Corporation 2016

Vendor-specific commands

Command sets are vendor-specific. Command sets are designed to modify commands on vendor devices and are associated with the specific syntax of that device to enable, modify, or disable commands. For instance, command sets that are designed for a Juniper device do not function on a Cisco device. The command set does not function because the syntax uses a **set** command to enable a feature whereas a Cisco device begins with the name of the feature.

Command sets rarely work between vendors. With Netcool Configuration Manager, you can automatically filter the command sets. For example, selecting a Cisco command set forces the execution wizard to select Cisco devices exclusively.

However, within vendors, command sets can apply to various models and operating systems. For example, the creation of a banner for a Cisco device is syntactically the same since the earliest versions of the Cisco IOS. A command set for setting a banner works on all Cisco IOS devices, regardless of model or operating system. This assumption breaks down as you delve into the IOS command structure that you attempt to change. Special features such as VPNs, quality of service, and traffic-shaping are ideas that change between models and operating systems. Assuming similarities in these areas can cause problems when running command sets.

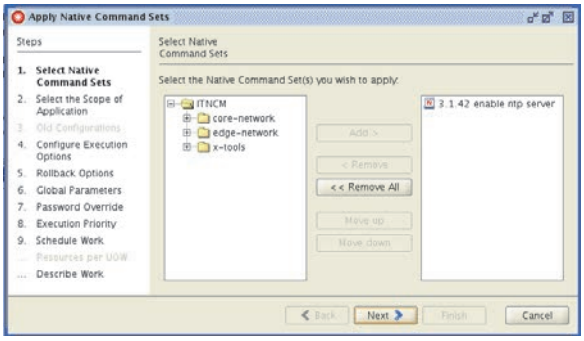
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Units of work

Use the UOW wizard

- Select command sets and devices
- Set schedule and priority
- Select execution options

Obtain approval, based on group security, if required



© Copyright IBM Corporation 2016

Units of work

Command sets apply to devices as a unit of work. Like other units of work, you select the items that are necessary for its execution. Unique to command set execution is the selection of command sets and their application order.

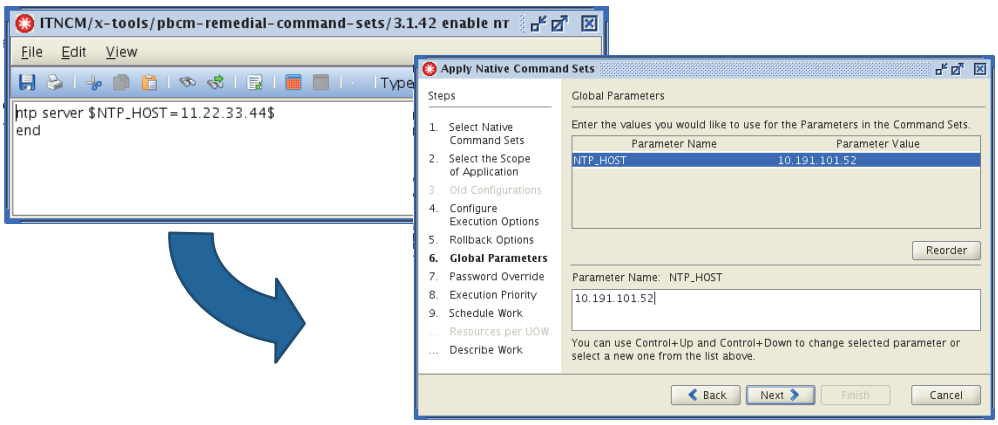
You can schedule and prioritize the command set execution. You can also select options that determine whether and when to stop the UOW from processing. As with other units of work, you might need approval for the unit of work to begin processing.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Parameters

Parameters provide reuse

Parameter values are defined when you create a unit of work



© Copyright IBM Corporation 2016

Parameters

You can add parameters to command sets to make them reusable. When you process these command sets, you are prompted to enter their values. If command sets are applied through an external API program, that utility can replace the parameters with values the higher-level code supplies.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Types of command sets

Modeled command set

- SmartModel
- Syntax checking
- Search and replace
- Defined rules for application to ensure expected results
- Can be tested
- Changes to configuration only

Native command set

- Cutting and pasting of an existing CLI snippet is permitted
- Interrogation mode allows use of show commands
- No syntax checking
- No search or replace function
- CLI-based (user-supplied intelligence)

© Copyright IBM Corporation 2016

Types of command sets

The solution provides two types of command sets: modeled and native. Modeled command sets have these features:

- They use the SmartModel to support features like syntax checking.
- They support search and replace features to allow dynamic changes to device configurations.
- They provide a set of defined rules for ensuring that data is not accidentally overwritten.
- You can test them to predict the actual CLI that is sent to one or more devices.

Note: Modeled command sets support only configuration changes on devices.

Native command sets have these features:

- They support the cutting and pasting of existing CLI templates to an editor, like Notepad. Cutting and pasting existing change templates can migrate quickly to Netcool Configuration Manager.
- You can use them in an interrogation mode where commands are sent to the device, and it saves their results in the unit of work log.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Note: Native command sets do not support syntax checking or search and replace functions. They require that the user supplies all the intelligence to ensure that the commands are correctly applied.

© Copyright IBM Corp. 2016

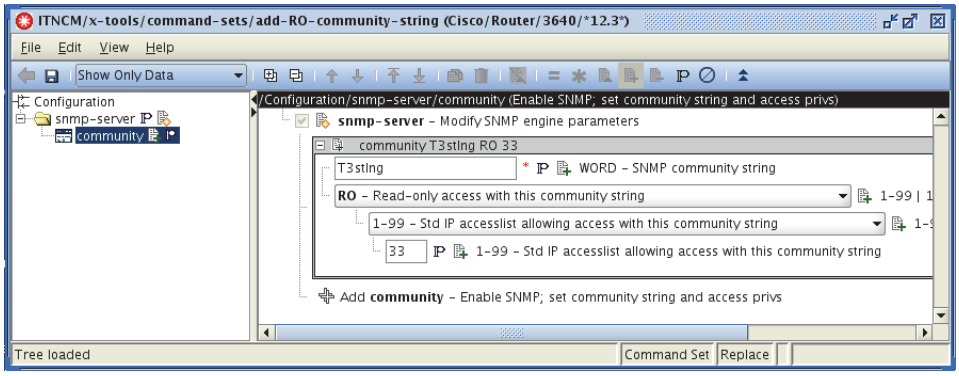
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modeled command sets

Are referred to as a command set in GUI

Use a command set editor and markup

Have CLI that is generated by SmartModel translators



© Copyright IBM Corporation 2016

Modeled command sets

The slide has an example of a modeled command set. It uses a modified version of the configuration editor. This editor has extra icons along the top of the screen that you use to apply markup to the commands. The markup determines whether commands are added, modified, or deleted from the configuration when it is applied.

Command sets are layered on top of a configuration at execution time. The markup attempts to line up with the commands that are currently configured on the device. If everything aligns, the markup in the command set becomes the markup in the configuration. From there, the SmartModel converts marked-up commands into the correct CLI commands, which are applied to the device.

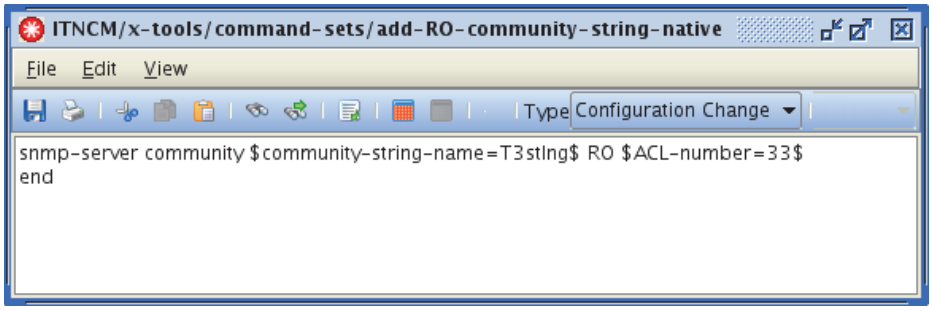
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Native command sets

Are referred to as a native command set in GUI

Have commands that are directly applied to the device, no translation

Require user group authorization



© Copyright IBM Corporation 2016

Native command sets

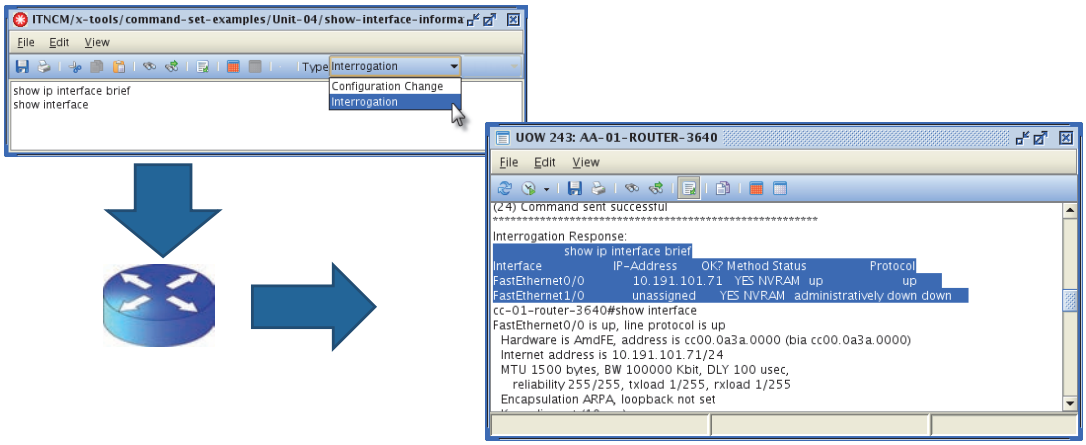
To create native command sets, you use a simple text editor, like Notepad. This editor supports cutting and pasting and direct character input. You apply these commands directly to the device. Native commands do not require conversion. These commands can overwrite an existing command when you do not apply them carefully. Because SmartModeling is not associated with this command set type, The user must supply all the intelligence associated with the change. This intelligence includes deleting any necessary commands, correct syntax, and correct ordering of the commands. The group that applies them must have the activity **Apply Native Command Sets**.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Interrogation command sets

Use these sets to interrogate devices and return data

- Enter show commands on the command line
- View the results in UOW logs



© Copyright IBM Corporation 2016

Interrogation command sets

An extension to the native command set is the interrogation command set. By selecting the type as **Interrogation** instead of **Configuration Change**, the system applies the commands in a nonconfiguration mode. The commands are submitted to the command line, and the device outputs any response to the unit of work log.

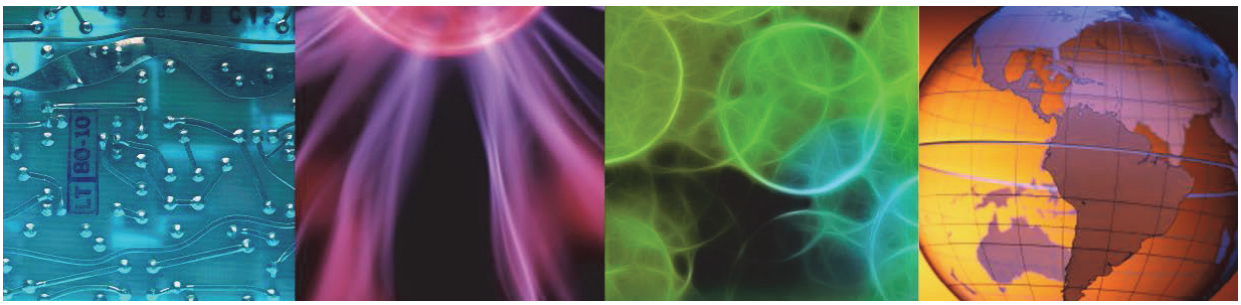
The commands that you typically use in these command sets are **show** commands. These commands respond with information about the current state of the device.

The interrogation mode submits only certain commands to the device. It filters commands like **reload** and **config t** and it cannot save them in the command set.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Building simple command sets

Lesson 2 Building simple command sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to build a simple command set.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating command sets

Understand device-specific syntax

Define the VTMOS of command set

- Modeled command sets needs a base schema to build commands
- Native commands sets do not use schema, only need VT criteria

Modeled command sets require markup to define change

- Determine whether the change is adding commands or modifying existing ones

Native command sets require correctly formatted CLI

- Test before applying

© Copyright IBM Corporation 2016

Creating command sets

To build a command set, you must have an understanding of the command syntax that the device uses. You must also know the intent of the change the command set is to apply. This latter idea is the most important aspect of command set development. Without understanding the intent, you can develop a command set that has unintended consequences for the device.

You must define a vendor and type for the command set so that the software filter mechanism ensures that it is applied to the correct devices. For modeled command sets, you must select a model and operating system. This detailed information selects a schema from the drivers database, which provides the command syntax to build the command set.

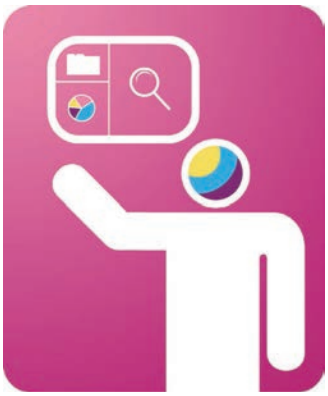
To build modeled command sets, you must perform a two-step development process:

1. Define commands to add, modify, or delete.
2. Mark up those commands with appropriate icons to tell the system when it is adding, modifying, or deleting the commands.

For native commands, you must correctly define the CLI syntax and verify that the commands are in the correct order. These command sets require more testing because no **Report Only** mode exists like there is with modeled command sets.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Instructor demonstration



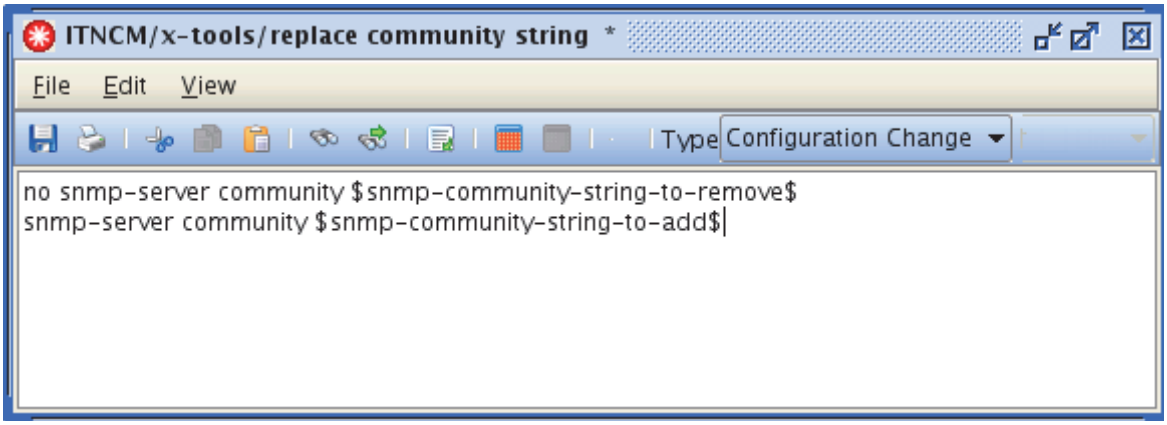
© Copyright IBM Corporation 2016

Instructor demonstration

In this demonstration, the instructor creates two command sets: a modeled set and a native set. These command sets must be of the same change to show the difference. Also, the command sets must have parameters to show how to make them generic.

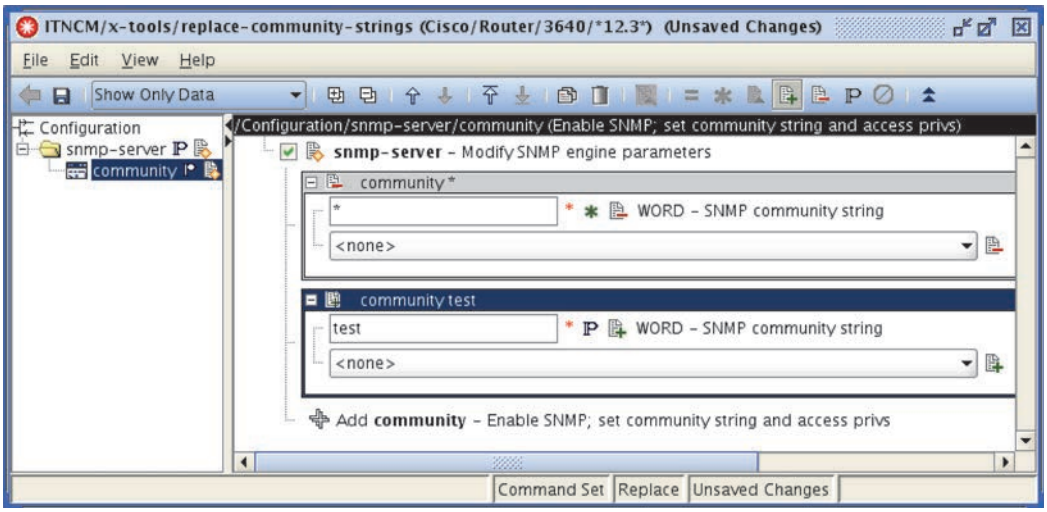
In this demonstration, the instructor performs these tasks:

- Create a native command set to delete and add a community string.



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

- Create an XML command set to delete all community strings and add one.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Create trivial command sets

Create an XML command set that:

- Adds an additional network time protocol (NTP) server command to a Cisco router
- Use a parameter to define the address of the server

Create a native command set that:

- Adds the same NTP server to the Cisco router
- Use a parameter to define the address of the server

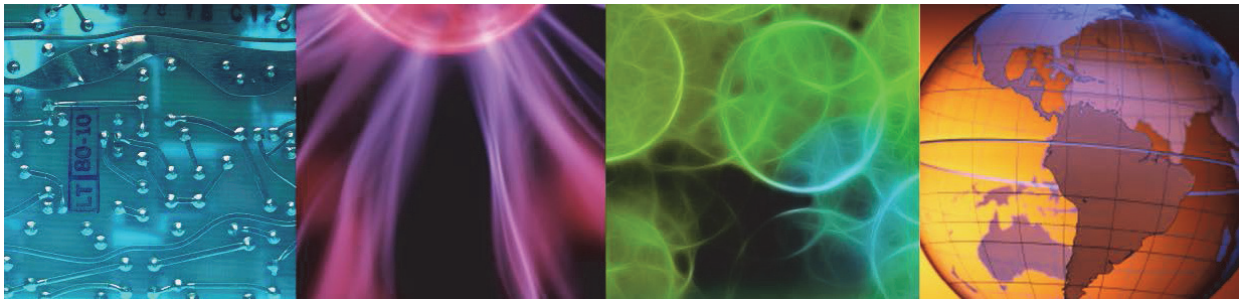
© Copyright IBM Corporation 2016

Create trivial command sets

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Processing command sets

Lesson 3 Processing command sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Run a command set against a device
- Describe where and when using a CSV file is advantageous

This lesson provides an introduction into processing command sets and options you have available to you

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Overview

You can run multiple command sets against multiple devices as one UOW or as multiple units of work (multitasking)

- Defined inside of the Execution wizard

You can set failure criteria

- Setting when to fail the UOW for when its command sets start to fail

You have multiple rollback options if failures occur

Processing issues

- You cannot mix and match vendors
- You cannot mix and match modeled command sets and native command sets
- The modeled command set might not change due to markup rules

© Copyright IBM Corporation 2016

Overview

When applying command sets to devices, you can use the execution wizard to select multiple command sets to apply to multiple devices. The command sets process in a specific order. Some changes require that certain commands are applied before others for the operating system of the device to accept the changes. Using the execution wizard, you can split the operation into separate *units of work* for multitasking.

When defining the unit of work, you set the failure criteria so that the UOW stops processing if problems occur when applying the commands. You use the command set execution wizard to set rollback options on an individual device, when command sets fail.

The current release of Netcool Configuration Manager has some execution issues.

- You cannot select multiple vendors. After you select a command set or device, a filter prevents other vendors in the UOW.
- You cannot combine native command sets with modeled command sets.
- Some modeled command sets do not modify devices because of the rules enforced with the markup. For example, a command set does not add a command if it exists in the configuration. The task log for this device reports **Success - no changes made.**

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Executing report only command sets

Only modeled command sets support *report only*

You can safely apply a command set to devices in this mode, runs commands only on the server

This command set determines what commands, if any, are sent to a device

Report only mode is effective for applying only one command set

© Copyright IBM Corporation 2016

Executing report only command sets

The *report only* function that is available for modeled command sets is a useful tool. It helps both in the development of command sets and their application in a production environment.

This mode ensures the safe application of changes to devices as the server applies the marked up command set against the XML configurations in the database. The physical network device is never touched in this mode. All activity is on the server side.

The XML configurations apply the markups in the command sets and then converted into the CLI changes sent to the device. This method is useful because the command set does not change the configuration based on what the command set finds.

The *report only* function provides useful information when testing one command set. If you apply more than one command set, they apply only to the current configuration in the database. In actual practice, the first command set is applied to the current configuration of the device. The second command set is applied to the configuration that results from the first application.

Important: Because no intermediate configuration is created in the report-only mode, applying two command sets that depend on each other is impractical.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using comma-separated value (CSV) files

Command set executions apply the same commands

- Example: Changing Simple Network Management Protocol (SNMP) community strings

CSV execution allows each command set to have unique values

- Example: Setting router host name to match the domain naming service (DNS) name

The CSV file is hosted on the client computer

© Copyright IBM Corporation 2016

Using comma-separated value (CSV) files

You can import a comma-separated value file that allows multiple command sets to apply to multiple devices. In a typical unit of work for command set execution, you apply the exact same commands to all the devices in the command set. Even with parameters, the values that are submitted to the unit of work apply to every device. In many cases, this behavior is preferred when applying changes to standardize various settings, such as SNMP community strings, NTP servers, and authentication settings.

However, network devices with individual settings require each command set to have a different parameter value. The CSV execution supports the use of parameters for each command set that are different for each device in the file. For example, the host name of each router is unique. Resetting these values with the normal command set executions is impractical.

Users must build these CSV files on the client desktops and then import them into the GUI to analyze and run.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

CSV file format

work-type: CommandSet or NativeCommandSet

work-name: Command set name

work-realm: Command set location

work-device: Device name, can be a shortcut

device-realm: Realm location of the work-device

parameter columns: Each column represents a command set parameter

column names must be in CSV file

heading names must match parameter names

	A	B	C	D	E	F	G
1	work-type	work-name	work-realm	work-device	device-realm	hostname	SNMP-contact
2	CommandSet	modify-hostname	ITNCM/command-sets	CC-01-ROUTER-3640	ITNCM/CC	cc-01-router-3640	
3	CommandSet	modify-SNMP-contact	ITNCM/command-sets	CC-01-ROUTER-3640	ITNCM/CC		Lee Koeser 555-212-7673
4	CommandSet	modify-hostname	ITNCM/command-sets	CC-02-ROUTER-3640	ITNCM/CC	cc-02-router-3640	
5	CommandSet	modify-SNMP-contact	ITNCM/command-sets	CC-02-ROUTER-3640	ITNCM/CC		Lee Koeser 555-212-7673
6	CommandSet	modify-hostname	ITNCM/command-sets	CC-03-ROUTER-3640	ITNCM/CC	cc-03-router-3640	
7	CommandSet	modify-SNMP-contact	ITNCM/command-sets	CC-03-ROUTER-3640	ITNCM/CC		Lee Koeser 555-212-7673

© Copyright IBM Corporation 2016

CSV file format

The format of the CSV file has a few requirements that you must maintain:

- You cannot combine native and modeled command sets.
- The first line of the CSV file must define the column headers.

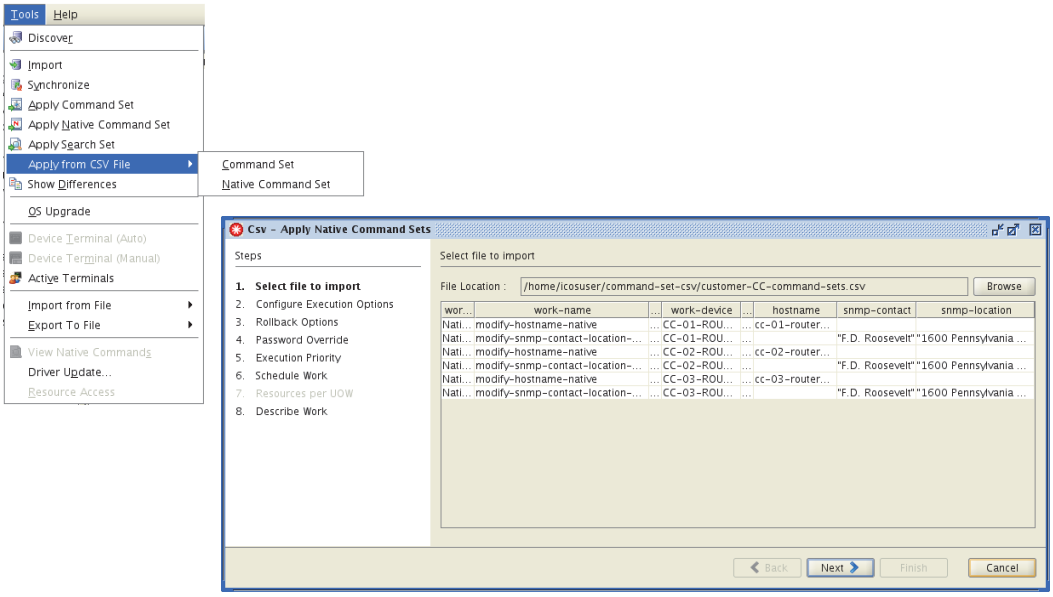
The description of each column is noted in the slide. The CSV file can mix and match a series of command sets that apply different commands. In the example in the slide, two command sets are applied to each device. One command set modifies the host name of the device, while the other sets the SNMP contact name.

Each line in the CSV file is a combination of one command set and one device. For each command set with parameters, the correct columns specify the parameter values.

The parameter columns can confuse you. When you build a list of the parameters for the CSV files that are defined in those command sets, each parameter name must have a column name that aligns with the command set. You must define the command set and device name on one line of the file so that the parameter value aligns to the correct column. Using a spreadsheet program such as Microsoft Excel is helpful. Be sure to save the file in a CSV format.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Apply a CSV file



© Copyright IBM Corporation 2016

Apply a CSV file

You can apply the CSV file from the GUI. Selecting **Tools > Apply from a CSV file** starts an execution wizard. You must know the type of command sets used in the CSV file: native command set or command set, which means modeled command sets.

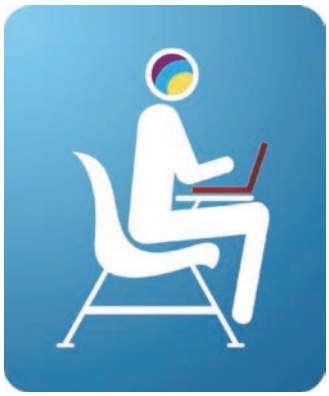
From the application wizard, you can select the CSV file from the client file system. When the file is imported into the wizard, it performs these checks:

- Check for the existence of the command sets in the specified realm locations
- Checks for the existence of the devices in the specified realm locations

If any inconsistencies are detected in these checks, you are notified. After importing the command set CSV file, the remaining options are similar to other UOW wizards. Multiple *units of work* are used for each device. If multiple command sets per device exist, each UOW has multiple command sets per device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises



© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Describe what a command set is
- Describe the difference between command set types
- Explain how to create a command set
- Modify one or more devices with a command set
- Modify devices by using command sets and a CSV input file

© Copyright IBM Corporation 2016

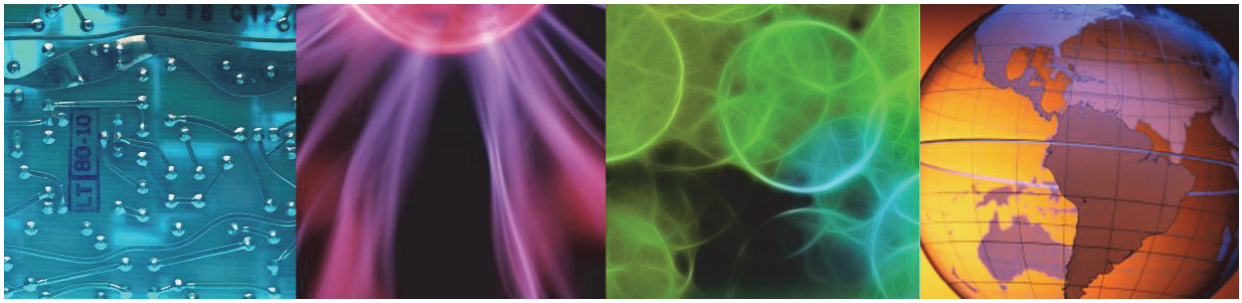
Summary

In this unit, you learned about the template technology of command sets. You learned about the types that are available and the ways that you can use these templates to make mass changes to the network. You were also shown a method to use a CSV file to define mass change with customized data for each line in the file.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

5 Administrative tasks

Administrative tasks



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit covers topics that are necessary to operate Netcool Configuration Manager. These topics include approving units of work (UOWs), stopping, and restarting units of work, processing complex searches, comparing configurations, restoring previous configurations, and viewing reports.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Describe the difference between a ticket and a UOW
- Approve units of work
- Stop and restart units of work
- Search for data in the database
- Determine the differences between configurations
- Restore versioned configurations
- Review reports

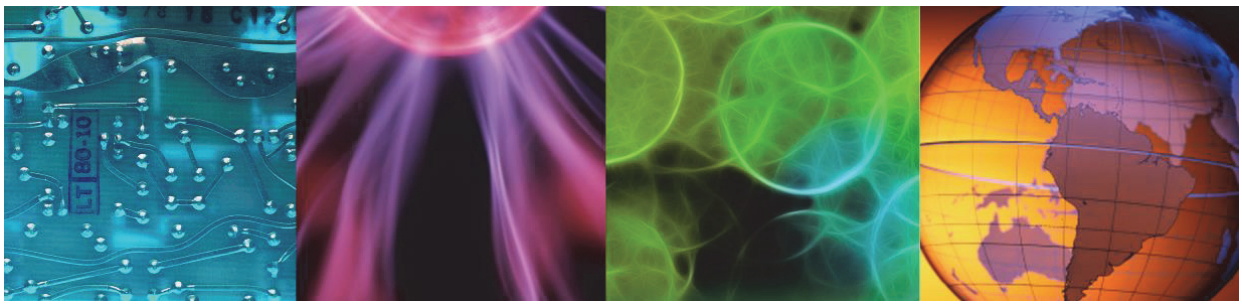
© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Approving changes

Lesson 1 Approving changes



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Approve units of work
- Enable approvals for groups

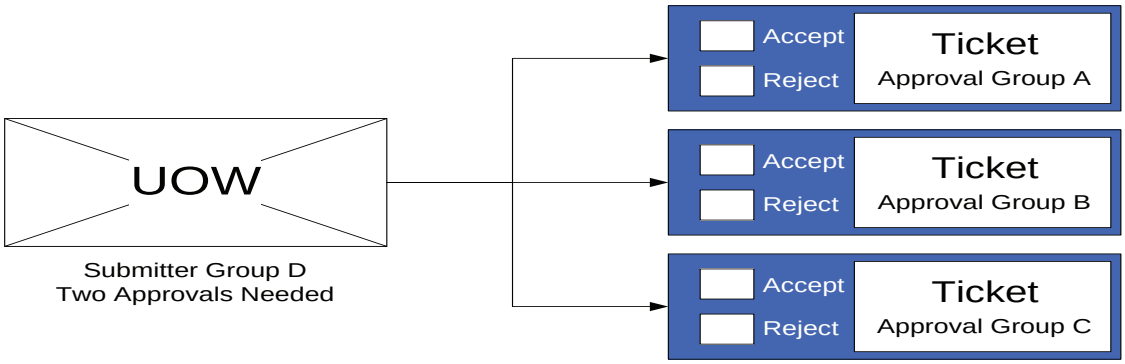
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Units of work and tickets

A unit of work (UOW) is an envelope that moves through a workflow

Tickets are the tools that allow the UOW to run

Not every UOW generates tickets



© Copyright IBM Corporation 2016

Units of work and tickets

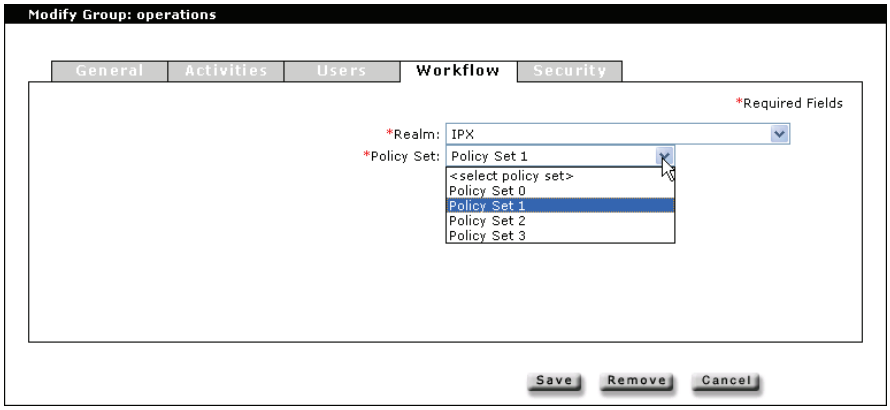
As mentioned in a previous lesson, the UOW is the object that most people monitor in the execution queues. It is the envelope in which devices and the associated operations are placed and sent through the Netcool Configuration Manager workflow.

Depending on who submits the UOW, it might require approvals. If approvals are needed, tickets are generated. People with the rights to approve *units of work* can approve or reject these tickets.

Not every unit of work generates tickets, only those requiring approval. If it requires tickets, one ticket is generated for each group that has the rights to approve the UOW.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Group policy



© Copyright IBM Corporation 2016

Group policy

When a group is created, you select a policy set for the group. A policy set defines how many approvals are required for various units of work types. In general, a policy set of one means that most units of work require one approval. The *report-only* units of work are the exception to this rule, which needs no approval. For example, a policy set of two means that it requires two approvals, except for a report only command set unit of work.

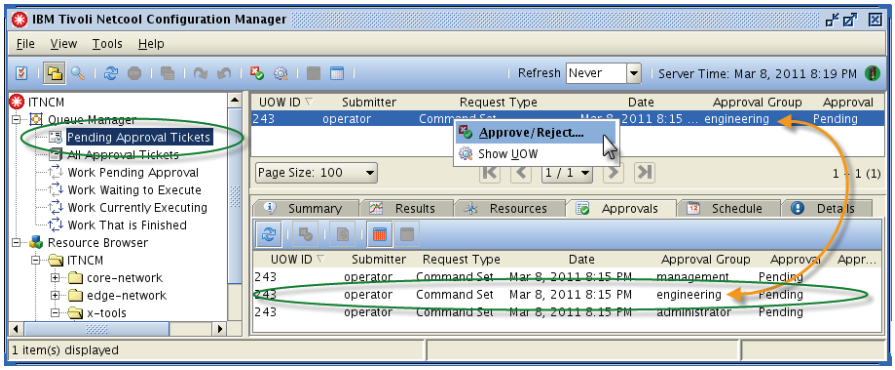
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Approving units of work

The UOW stays in the *pending approval* queue until the required number of approvals is reached

An approver approves tickets, not units of work

Use the **Pending Approval Tickets** search



© Copyright IBM Corporation 2016

Approving units of work

When a group that needs approval submits the unit of work, it is in the *work pending approval* queue until one of these actions occur:

- It is approved and moves to the *work waiting to execute* queue.
- It is rejected and moves to the *work that is finished* queue.
- The submitter or approver removes the unit of work from the queue, and it is moved to the *work that is finished* queue.
- It is expired and moved to the *work that is finished* queue when the scheduled end is reached and none of the previous actions occurred.

You can approve or reject tickets in the *pending approval tickets* queue. When you select the ticket, you see all the other group tickets that are associated with the unit of work. In the example, the ticket in the upper window is selected for approval. The approver can view all the other groups who can approve the ticket also. The ticket requires approval before the UOW can begin processing.

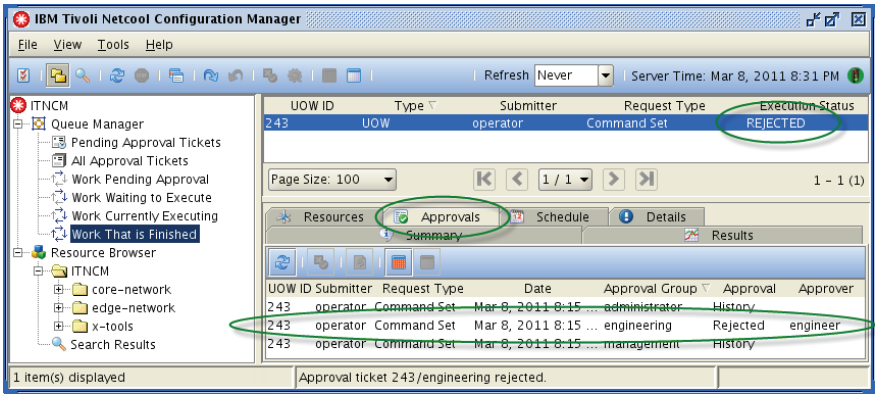
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Rejecting units of work

Anyone from an approval group can reject the ticket

Only one rejection is required to fail the UOW

Comments are attached to the ticket



© Copyright IBM Corporation 2016

Rejecting units of work

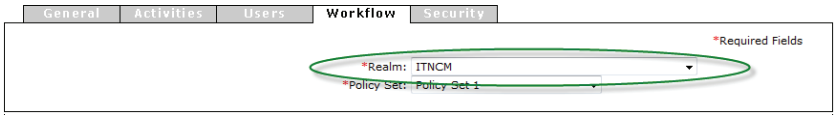
Rejecting a ticket is the same process as approving. Any user in one of the three approval groups in the slide can reject the ticket. It takes only one rejection to fail the unit of work. Whoever rejects the ticket is noted in the unit of work. By selecting the unit of work and looking at the **Approvals** tab, you can see who rejected the ticket and when it happened. For future auditing purposes, the user who rejects the ticket must submit a comment to explain the rejection.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Ticketing behavior

Approval groups see tickets simultaneously

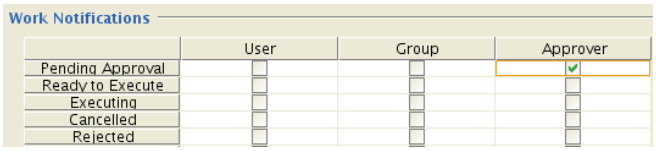
Associations between tickets and groups are defined in Workflow tab



Some units of work do not generate tickets

Report only units of work run without approval

Use work notifications to notify approvers of tickets



© Copyright IBM Corporation 2016

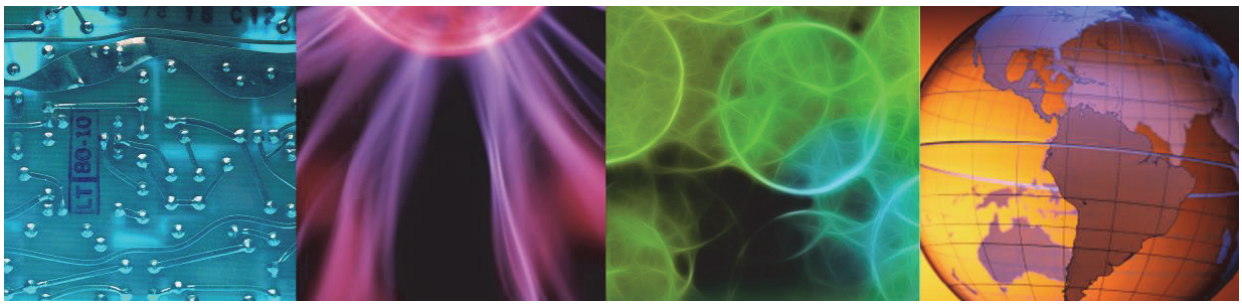
Ticketing behavior

Tickets are sent to all approvers simultaneously. For example, in a two-approval hierarchy, the engineers and the managers see the tickets at the same time. The simultaneous arrival of tickets can cause confusion when a manager approves a ticket that an engineer rejects. The application provides some control over who receives tickets, but not when they receive them. Approvers can use work notifications that are part of their user preferences to receive email that is generated when work is available for approval.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Stopping and restarting units of work

Lesson 2 Stopping and restarting units of work



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Stop the execution of a UOW
- Restart the execution of a finished UOW

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

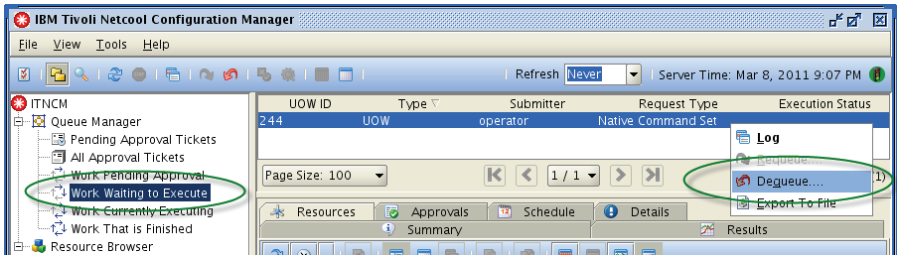
Stopping a unit of work

Units of work can be stopped (dequeued) by specific users

- User who submits the task
- User who can approve the task

Units of work can be dequeued in the following states

- Work pending approval
- Work waiting to execute
- Work currently executing



© Copyright IBM Corporation 2016

Stopping a unit of work

You can stop or dequeue units of work. However, only the user who submits the unit of work or users who can approve the unit of work can remove them from the queue, or *dequeue* them. You can dequeue units of work when they are in the *work pending approval* or *work waiting for execution* queues. Because these UOWs are not processing, the entire work object moves to the *work that is finished* queue with the state of DEQUEUED. If you dequeue a processing UOW, you use a different process to stop the UOW.

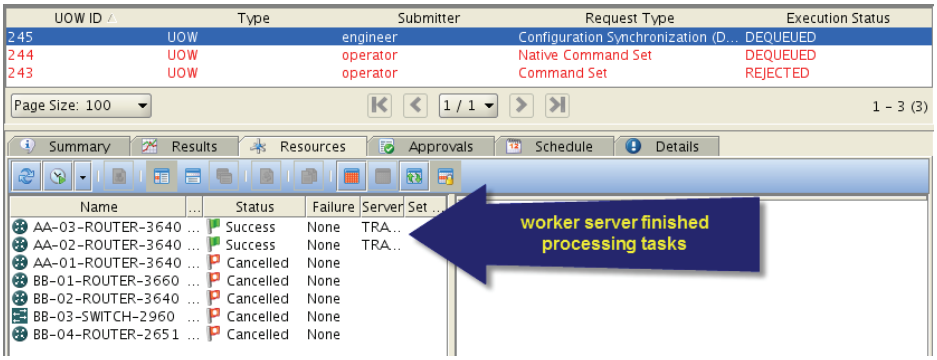
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Stopping a processing unit of work

Worker servers finish processing tasks

Worker servers do not start new tasks in a UOW

It dequeues units of work that have multiple tasks



© Copyright IBM Corporation 2016

Stopping a processing unit of work

Stopping a processing unit of work does not instantaneously stop the threads that are interacting with a device. Any UOW tasks that are processing are completed, and all unprocessed tasks are ignored. For example, in the slide the worker servers complete the two synchronizations and cancel the remaining ones. The worker server always completes the task that it is processing. That being said, it makes sense to dequeue only units of work with multiple tasks. The tasks that are not processed in the UOW that is dequeued have a status of canceled.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Restarting a unit of work

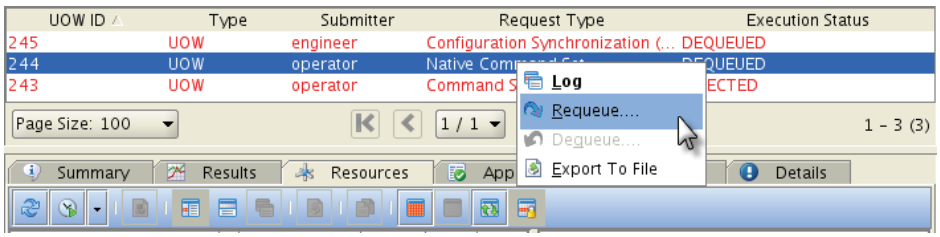
You can restart or requeue a unit of work

If you see a finished UOW, you can restart it

An approval policy might apply

The same change applies when the UOW is requeued

Changes to a command set are not reflected in a requeued UOW



© Copyright IBM Corporation 2016

Restarting a unit of work

Like dequeuing a UOW, you can restart or requeue a UOW to repeat its execution. If you see a UOW in the *work that is finished* queue, you can requeue it. The approval policy that is associated with a specific user is still applicable, and can generate tickets.

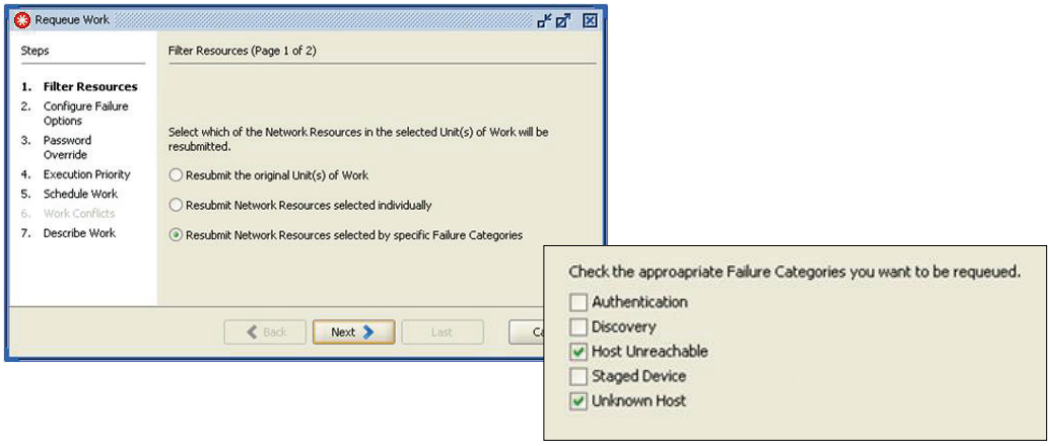
The requeued UOW has the same properties and changes of the original UOW. For instance, a command set UOW caches the command sets in its data structure. Even if the original command sets are modified in the database, The UOW saves the original command set and uses it for execution.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Filtering resources from a requeued UOW

Resubmit only the failed tasks in a UOW

Only on units of work with multiple resources



© Copyright IBM Corporation 2016

Filtering resources from a requeued UOW

During the requeuing of the UOW, you must enter the same information that you do for a UOW of the same type. However, the requeued UOW has an extra feature that can save processing time.

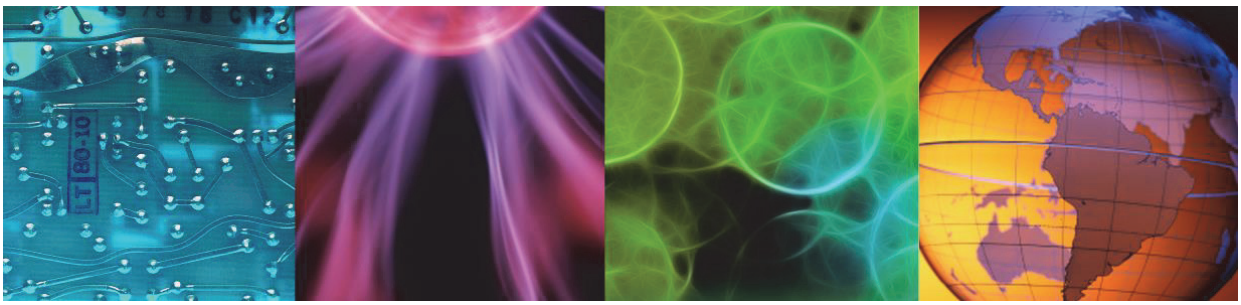
When the UOW processes and has failures, the UOW can be requeued to include only the devices that failed. To do so, you select **Resubmit Network Resources selected by specific Failure Categories**.

Selecting this option prompts for the failure categories that are unique to this UOW, as opposed to all failure categories that are encoded in Netcool Configuration Manager. If the UOW failed for issues that are associated with DNS and routing problems, the UOW can be requeued for only those devices when the problems are resolved.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Searching and comparing units of work or resources

Lesson 3 Searching and comparing units of work or resources



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Perform searches
- Compare configurations
- Create custom searches

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

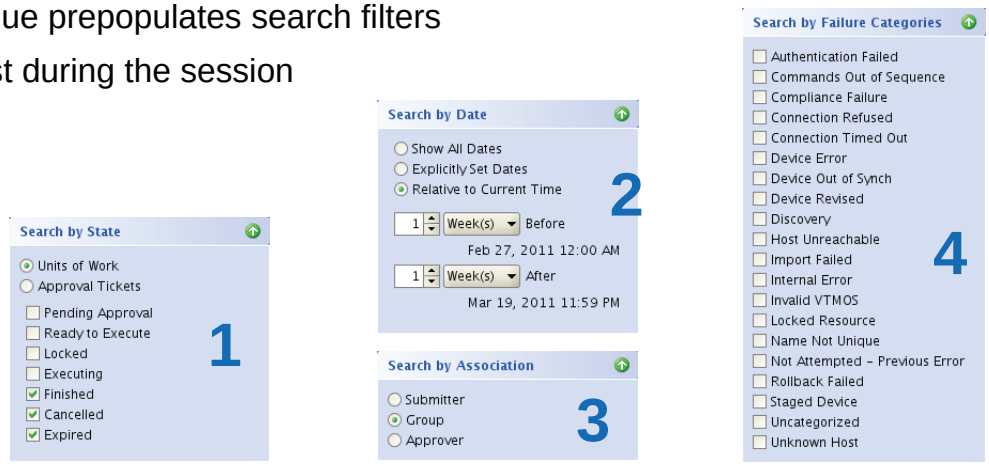
Searching for units of work or tickets

Security defines what units of work are searchable

You see work that is associated with your group

Selecting a queue prepopulates search filters

Changes persist during the session



© Copyright IBM Corporation 2016

Searching for units of work or tickets

You can use the advanced search capabilities for both *units of work* and their tickets, and resources in the resource browser. What you currently view in the GUI determines which search pane is presented when you select the advanced search options.

If you are viewing *units of work* and tickets, advanced searches are only allowed on the UOWs that you have access to. Currently, you either see all *units of work* and their tickets or only the UOWs associated with your group.

When you select a default search in the queue manager and then select advanced search, the search criteria are inherited from the default search. Changing the default values, for example the Search by Date pane, persists until it is reset or the GUI is closed.

1. **Search by State**: The items that you select in this pane default to whatever search was most recently completed. If you look at the *work that is finished* queue, the check boxes for Finished, Canceled, and Expired are selected. You can either retrieve *units of work* or tickets, but not both. As a general statement, most users are interested in the *units of work* that are completed and the UOWs that are *ready to run*.

2. **Search by Date**: The default for all UOW searches is to look one week into the future and one week into the past, which is based on the current server time. While setting explicit dates for searches is useful for auditing purposes, most searches use *relative to current time*. Many of these searches are for UOWs that finished in the recent past, within either hours or days.

3. **Search by Association**: By default, all searches are for *units of work* that the group instantiates. However, this search can be limited by looking only for UOWs that you submit. If your group has approval rights, you can see *units of work* you can approve.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

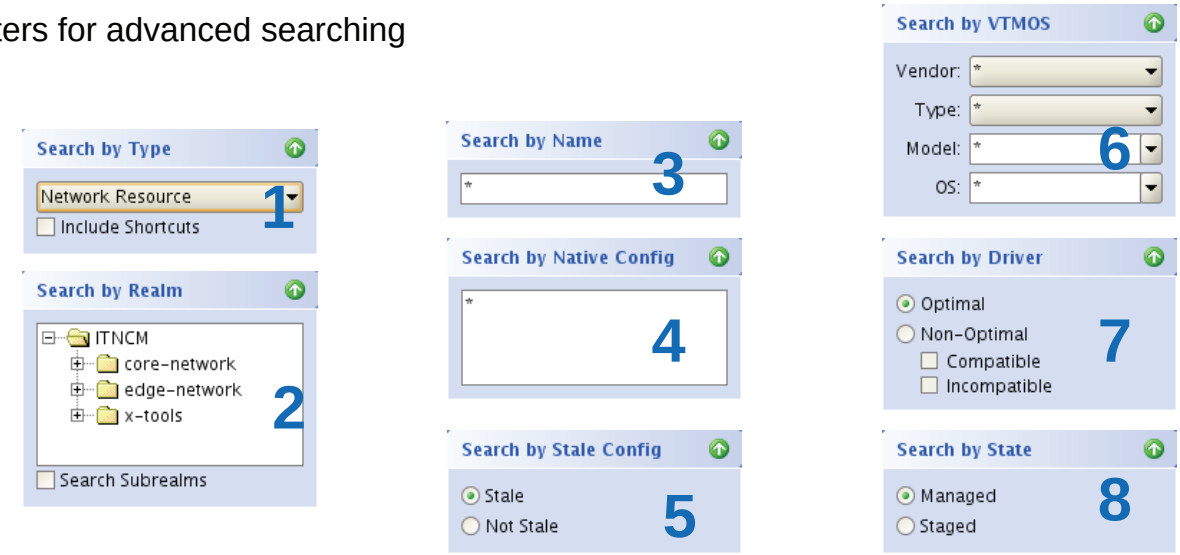
4. **Search by Failure Categories**: This list of *failure states* filters *units of work* in the *work that is finished* queue. These UOWs failed for known reasons. This search is useful for gathering failures that are caused by network routing or DNS issues that you must fix outside the application.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Searching for resources

Filters for advanced searching



© Copyright IBM Corporation 2016

Searching for resources

When you search in the resource browser, you typically search for network devices. However, other resources might also interest you, for example, authentication resources or resource access documents. Using the assumption that a network resource is the main search object, you can apply these filters. **Search by Realm** is a useful tool as it helps limit the scope of the search. Reducing the scope of your search is especially important when using **Search by Native Config,** which is computationally expensive for the database.

1. **Search by Type**: Select the resource type that you are interested in searching, most commonly, network resources. Most resources can have a logical link or shortcut. Selecting a type of resource can potentially remove search filters that do not apply.

2. **Search by Realm**: Reducing the size of the search by looking only in specific realms is a useful choice to reduce the data returned. If you select a realm, select the option to search subrealms also.

3. **Search by Name**: If resources, such as network resources, have a standardized naming convention, you can use a regular expression to search for resources by name. Typical searches have the format of ***<some value>***.

4. **Search by Native Config**: Use this option only if Search by Type is set to either Configuration or Network Resource. You can use a regular expression to search through the native configuration of a device. It returns any device that matches the expression. Perform this search carefully and have the Search by Realm filter set to reduce the potential devices searched. Also, character strings in this field are vendor-specific. Setting the Search by VTMOS

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

criteria can reduce processing that the search requires. This search is resource-intensive and can cause problems with the database, if you do not limit it.

5. **Search by Stale Config**: You can configure devices not to retrieve configurations after a change is made. In this situation, a device is referred to as *stale*. You must synchronize stale devices to allow future changes. This search retrieves devices that are noted as stale.

6. **Search by VTMOS**: This filter returns only resources that match the VTMOS selected. You can select one or more fields as a filter. For instance, you can search for network resources that have an OS of *12.3* and it returns a list of devices, regardless of vendor, type, or model.

7. **Search by Driver**: This search is associated with network resources. This search retrieves devices that require updated drivers. This issue occurs after a new or updated driver is added to the implementation. Drivers might also need updating when you change the operating system.

8. **Search by State**: This filter returns devices that are either *managed* or *staged*. A staged device is one that is created in the database but does not have a configuration from a device on the network. It is a placeholder.

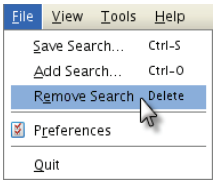
© Copyright IBM Corp. 2016

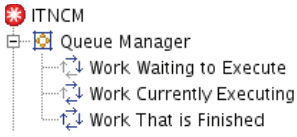
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Customizing the queue manager

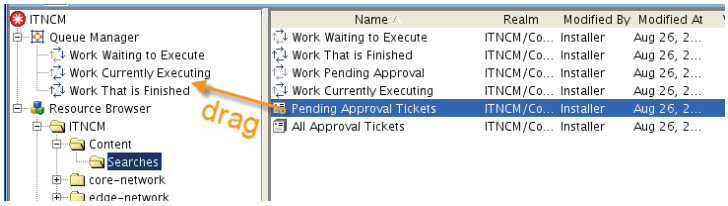
You can customize your queue manager view

- Click **File > Remove Search** to delete default searches





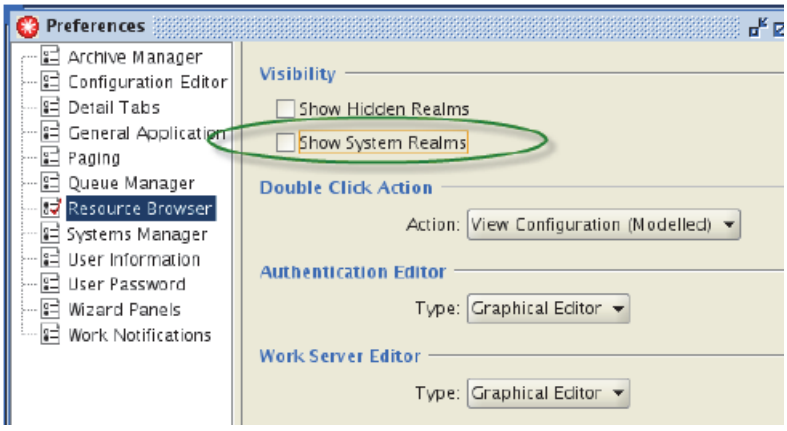
- Drag searches from the **Content > Searches** realm to restore default searches



© Copyright IBM Corporation 2016

Customizing the queue manager

You can remove the default searches in the *queue manager* by selecting them and using **File > Remove Search**. Deleting searches can be useful for users who never approve work. You can restore the search by selecting the **Content > Searches** realm. This realm might not be visible because of default resource browser settings in the user preferences. The content realm is considered a system realm, and is not visible by default.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

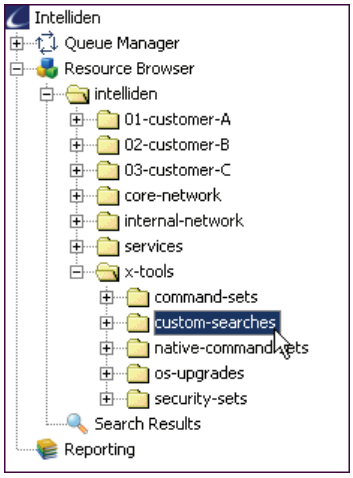
Building custom searches

You can implement custom searches in the queue manager only

You can create, use, and share custom searches

- They are a resource, like a command set
- Common practice is to save all custom searches to a single realm

The search becomes part of the user profile, which is on the local client



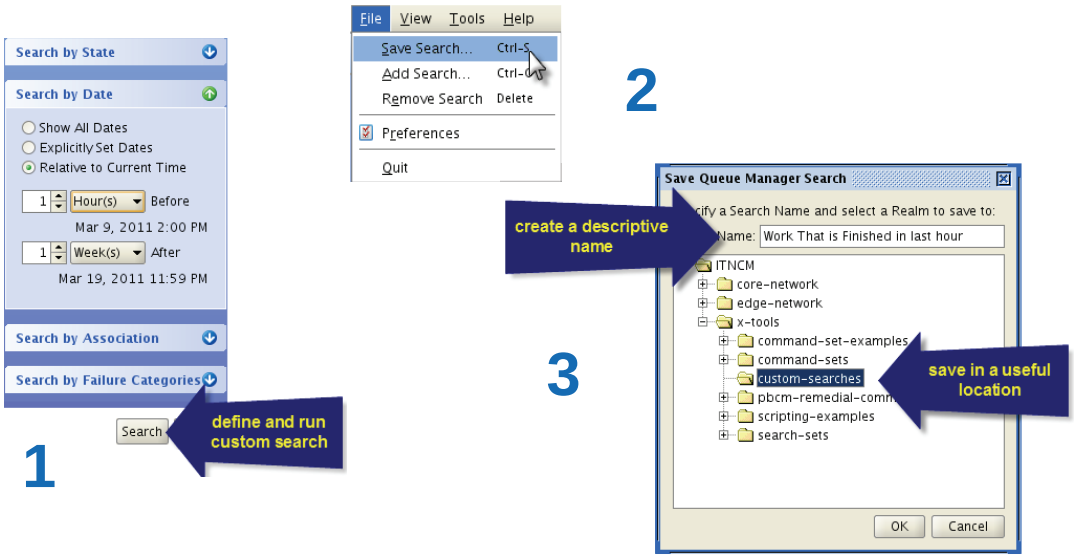
© Copyright IBM Corporation 2016

Building custom searches

You can build custom searches for the queue manager in Netcool Configuration Manager. When you create these searches, you create a resource that you save in the resource browser. This search resource is available to any user who can view the location where it is saved. It is a common practice to create a specific realm to save these searches. You can add these searches to the ones shown in your GUI, which become part of your profile and are available after subsequent GUI logins.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Saving custom searches



© Copyright IBM Corporation 2016

Saving custom searches

You can use the queue manager to create a custom search.

1. By selecting a default search, for instance, Work That is Finished, you can modify the default search by selecting the filters that are available in the queue manager. In this example, the default setting is modified from one week to 1 hour. The *units of work* that finished in the last 60 minutes are retrieved.

2. You can save the search like other resources inside the resource browser. Click **File > Save Search** and specify a descriptive name for the search.

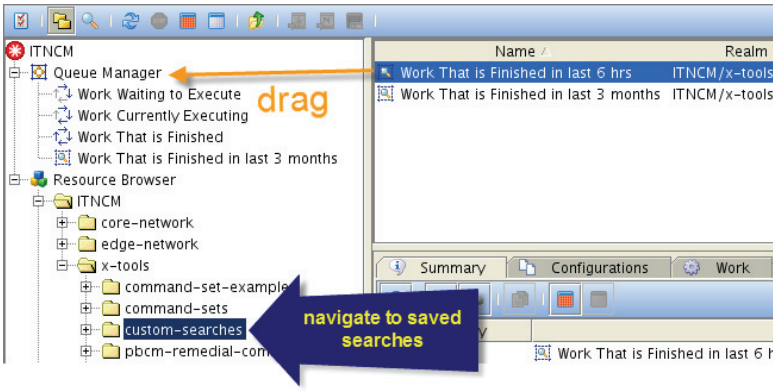
3. When it prompts you for a location to save the search, save it in a standard location in the realm structure so that others can use it.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Adding custom searches

Navigate to saved searches

- Double-click a search to run it immediately
- Drag it to the queue manager to save it in a profile for later use



© Copyright IBM Corporation 2016

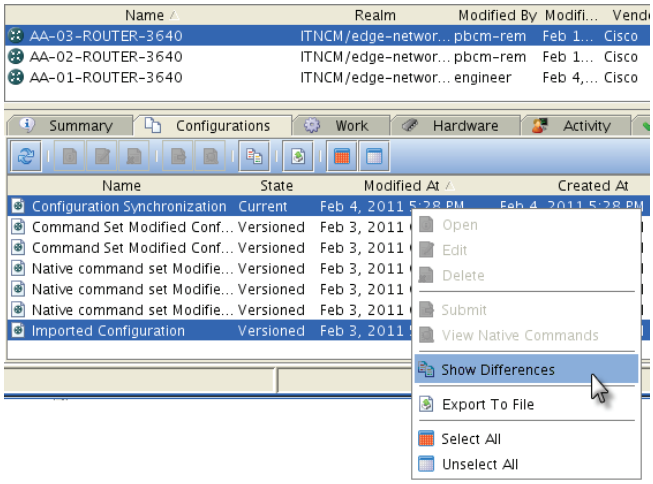
Adding custom searches

You can use these saved searches directly from their saved location in the queue manager. Double-clicking the saved search processes it immediately. If you use the search often, you can add it to the existing list of searches in the queue manager. Select the search and drag it to the queue manager to add it. It is saved in your profile and is available in future logins. You can relocate the search by selecting it and using the **Edit > Move Search Up** or **Edit > Move Search Down** menu selection.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Comparing device configuration states

Select any two configurations for a single device



© Copyright IBM Corporation 2016

Comparing device configuration states

In the following slides, you use Netcool Configuration Manager to compare configurations.

You can compare configuration states on the same device. Select a device and select the **Configurations** tab. Select the two configurations of interest, right-click, and select **Show Differences**.

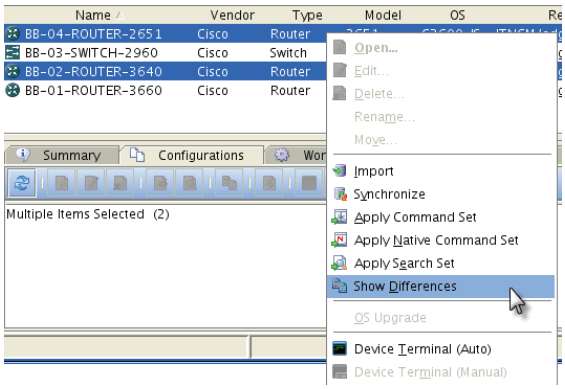
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Comparing two devices

Devices must be the same vendor and type

For example, devices must be Cisco routers

Devices are compared for the current configuration of each device



© Copyright IBM Corporation 2016

Comparing two devices

You can also compare two devices. The two devices must have the same vendor and type. Select two devices, right-click, and select **Show Differences**. You are comparing the current configurations of each device.

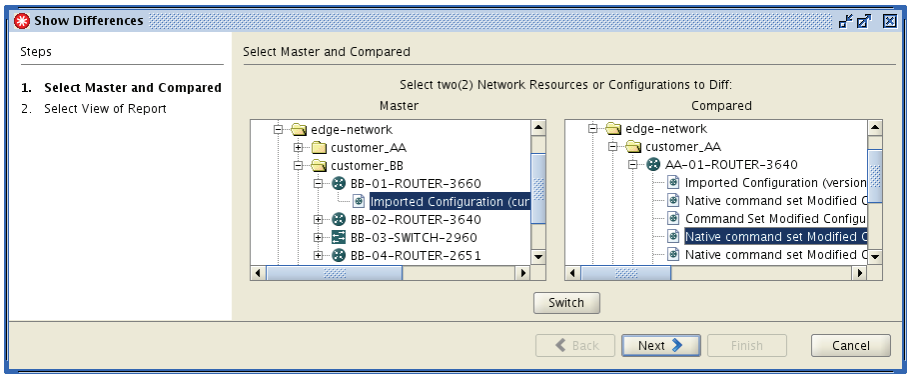
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Comparing two configurations from two devices

Select the first device and select **Show Differences**

The results are filtered for vendor and type

Select specific configurations to compare



© Copyright IBM Corporation 2016

Comparing two configurations from two devices

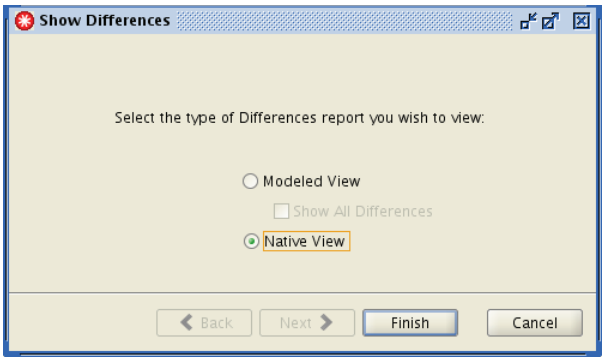
A final option is to compare two specific configurations from two devices. Select a device, right-click, and select **Show Differences**. The GUI filters the compare list for devices that are the same vendor and type as the master device. You can then browse through devices and select the specific configurations that you want to compare.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Comparing native or modeled configurations

Compare differences between raw or modeled (XML) versions of configurations

Native view is the most common type of comparison

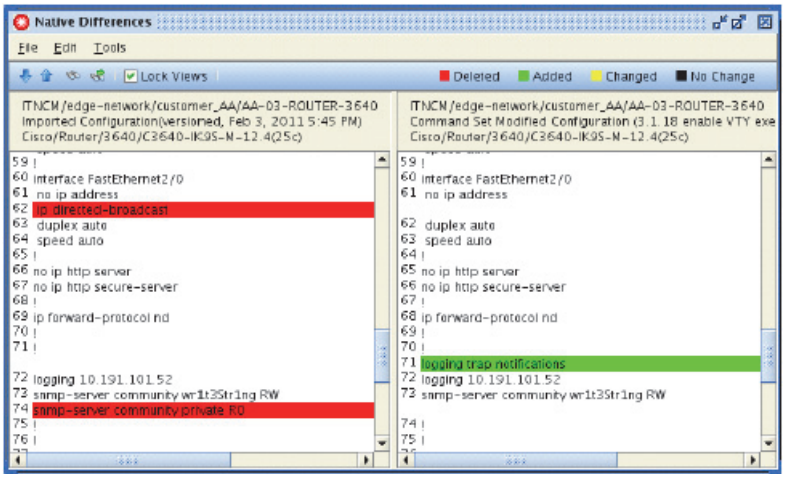


© Copyright IBM Corporation 2016

Comparing native or modeled configurations

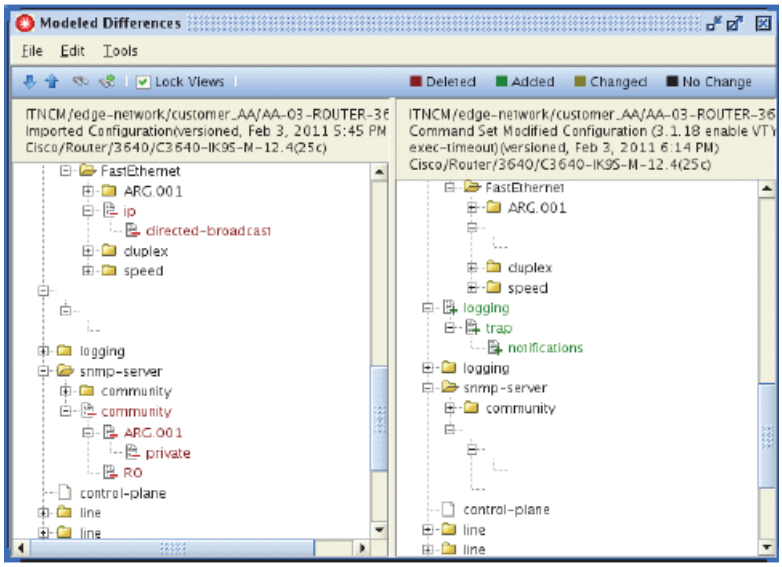
When the **Select the type of Differences report you want to view** dialog box is shown, select which configuration type you want to compare: the modeled or native configuration. The native configuration is what most people choose when comparing two configurations.

Two native configurations are compared in the following example.



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In the next example, you see a comparison of two modeled configurations.

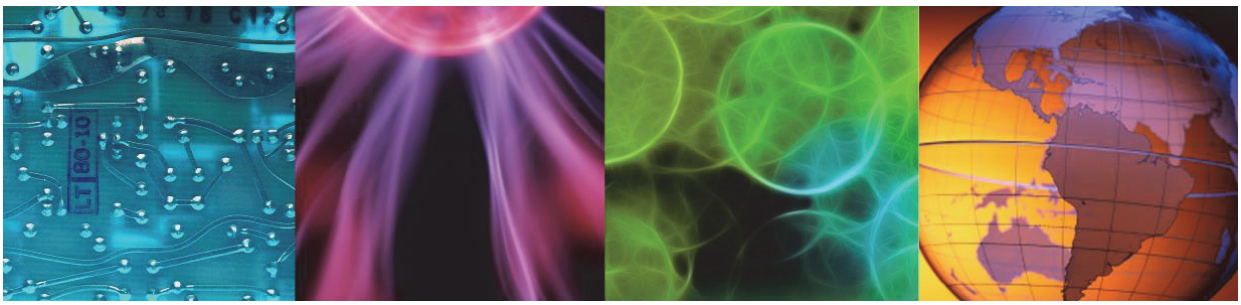


A Juniper Junos device retrieves a native configuration that is already in an XML format. Viewing that configuration in the native view might produce unexpected results.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 Restoring a versioned configuration

Lesson 4 Restoring a versioned configuration



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe how modeling supports better restoration
- Restore a version configuration

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Restoring configurations

Issues can occur when restoring traditional configurations

- Some devices, for example, Cisco IOS devices, allow only merging of configurations into memory
- Merging adds and modifies commands on a device, but does not remove them
- Accurate restoration requires service interruption, copy data to a stored configuration and reload the device

© Copyright IBM Corporation 2016

Restoring configurations

In some devices, for example, the Cisco 12.x line of IOS code, a previous configuration can be merged only into the running configuration of the device. The running configuration is what currently exists in memory, and merging the configuration into it adds or modifies commands only. No commands are deleted from the running configuration.

To return a Cisco IOS device to its previous state, you must restore the previous configuration to the startup configuration. This configuration is stored in nonvolatile memory. You must then reload the device so that all configuration data is erased from memory, and the data from the startup configuration can be loaded. This action causes a service disruption while the router reloads its configuration, rebuilds its routing tables, and initializes.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using models to compare configurations

Use XML and underlying schema to compare configurations

- Ensures that command order is important
- Includes processing information about how to create device commands

Uses third-party tools to compare XML documents

- Marks up document with what is added, modified, or removed

© Copyright IBM Corporation 2016

Using models to compare configurations

When you use XML and its schema, the model includes what is important in a configuration file. For instance, the command order in an access list is much more important than a list of community strings. Understanding this difference ensures that permit statements are treated with a higher priority for differences than community strings. Schemas are also useful for creating the commands that are needed to restore the device. For example, the schema for a Cisco IOS **no** command to delete data from a configuration is differentiated from a **clear** command that another vendor uses.

You can compare XML documents and determine what commands are added, modified, and deleted between two configuration states. This comparison process leaves a document *marked up* with the appropriate changes required to return to the previous state.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Restoring an IOS device

OLD CONFIGURATION
```
access-list 12
      permit 3.3.3.3
      permit 2.2.2.2
      permit 1.1.1.1
      deny any
access-list 102
      permit 10.10.10.10
      permit 20.20.20.20
      deny any
interface fastethernet 0/0
      ip access-group 102 in
      ip address 10.217.200.18
      duplex full
```

CURRENT CONFIGURATION
```
access-list 12
      permit 5.5.5.5
      permit 2.2.2.2
      permit 6.6.6.6
      deny any
access-list 102
      permit 10.10.10.10
      permit 20.20.20.20
      deny any
interface fastethernet 0/0
      ip access-group 12 in
      ip address 10.217.200.18
      duplex auto
```

PLANNED CONFIGURATION
```
access-list 12
      permit 2.2.2.2
      deny any
access-list 102
      permit 10.10.10.10
      permit 20.20.20.20
      deny any
interface fastethernet 0/0
      ip access-group
      ip address 10.217.200.18
      duplex
```

Required changes

COMMANDS TO RESTORE OLD CONFIG
```
no access-list 12
access-list 12 permit 3.3.3.3
access-list 12 permit 2.2.2.2
access-list 12 permit 1.1.1.1
access-list 12 deny any
interface fastethernet 0/0
      no ip access-group 12 in
      ip access-group 102 in
      no duplex auto
      duplex full
end
```

© Copyright IBM Corporation 2016

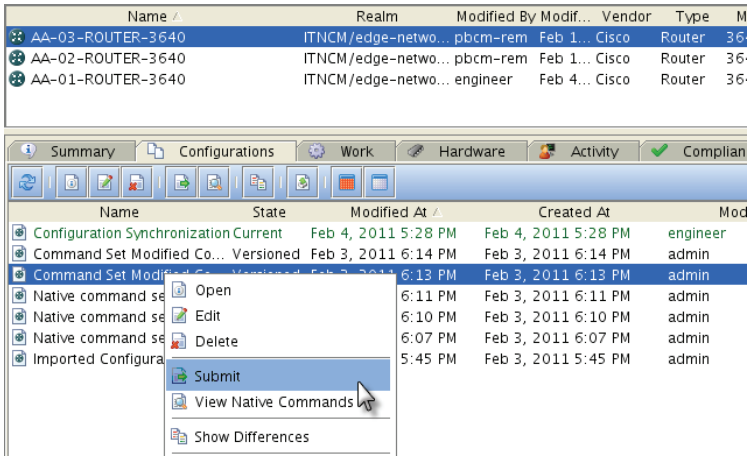
Restoring an IOS device

You can compare the old configuration with the current configuration by using intelligent restoration. With SmartModel technology, the access list and FastEthernet interface are modified automatically. The schema includes the information that is needed. In this case to alter an access list, it must first delete it and then rebuild it with the correct details. You must delete commands before they are re-created to eliminate the possibility of data left in memory. The importance of the model is that you are sending only the commands down to the device to return it to the previous configuration state. Using SmartModels prevents you from having to reload the device to get it back to its original state.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Restoring a configuration on a device

Select a versioned configuration, and submit it to workflow



© Copyright IBM Corporation 2016

Restoring a configuration on a device

You can restore a previous configuration by selecting the original configuration and submitting it to the workflow engine. In processing the task, a comparison is made and only the commands that are needed to restore the device to a previous state are sent to the device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

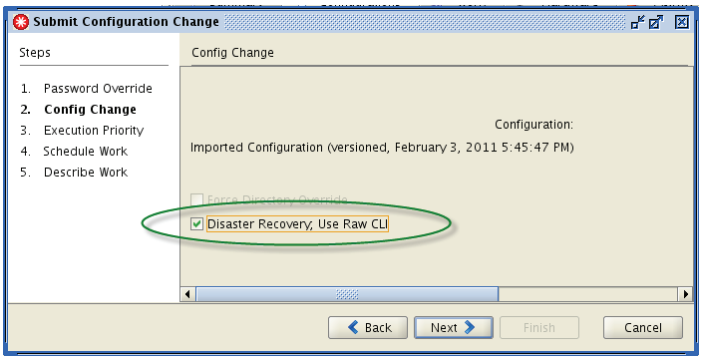
Restoring configurations by using other methods

Tivoli Netcool Configuration Manager stores raw CLI for every configuration

You can restore the entire configuration in an emergency

Restoration methods depend on vendor device

This method is referred to as disaster recovery



© Copyright IBM Corporation 2016

Restoring configurations by using other methods

Sometimes, you must push the entire configuration to the device. Netcool Configuration Manager has a modeled version of the configuration and the original configuration that was retrieved from the device. You can use this *raw* configuration to push the entire configuration to the device and then reload it.

This restoration method is useful in these situations:

- The hardware is defective, and a new device is placed in the rack. The device is given minimal network connectivity, and the engineer has the entire configuration to push to the device and restart.
- The device has only standard support. This method is the only means for restoring a previous configuration.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Standard device driver support

Standard driver support does not have modeled configurations

Restoration of versioned configurations is accomplished only with disaster recovery

Restoration can include text configurations or binary configurations

© Copyright IBM Corporation 2016

Standard device driver support

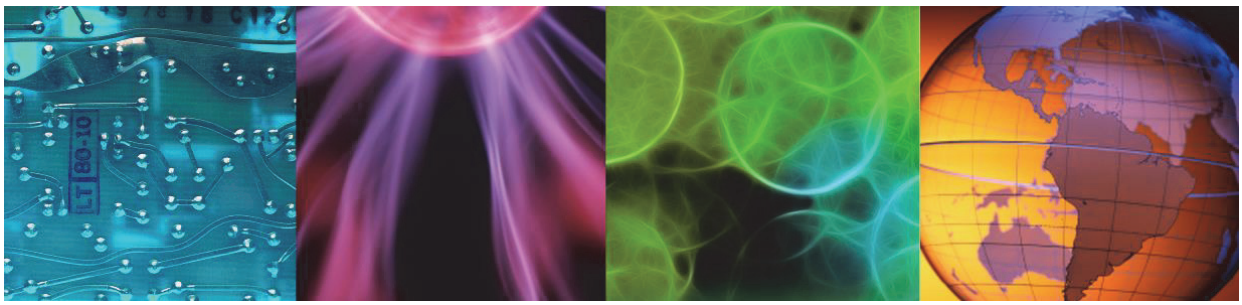
Standard drivers manage only the native or raw configuration of the device. The standard drivers do not support modeled configurations because the raw configuration is text only. Without modeling, You cannot use the concepts of intelligent configuration management. In devices that allow only the merging of configurations, reloading of the device is necessary.

You must accomplish restoration of configurations through disaster recovery or the cutting and pasting of the configuration into the device terminal. Standard device support allows the restoration of a proprietary image file, when that is all that the device supports. This proprietary image file support requires that you use another application for configuration changes. Netcool Configuration Manager, in this instance, supports only the capture and restoration of this proprietary image file.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 5 Reporting

Lesson 5 Reporting



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the available reports
- Create reports in Tivoli Common Reporting

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Running reports

Reports are installed in Dashboard Application Services Hub in Common Reporting

- Separate installation process

Includes both configuration management and compliance reports

- 6 configuration management reports
- 9 compliance reports

Tivoli Common Reporting supports the following

- Saving reports in multiple formats
- Running reports on a regular schedule
- Delivering reports by email

Reports use the same database as the application

- Reports are up-to-date

Note: Support for Dashboard Application Services Hub is new in V6.4.2

© Copyright IBM Corporation 2016

Running reports

Reports are installed in Dashboard Application Services Hub. You install the reports into Dashboard Application Services Hub in a separate installation process from installing the Netcool Configuration Manager.

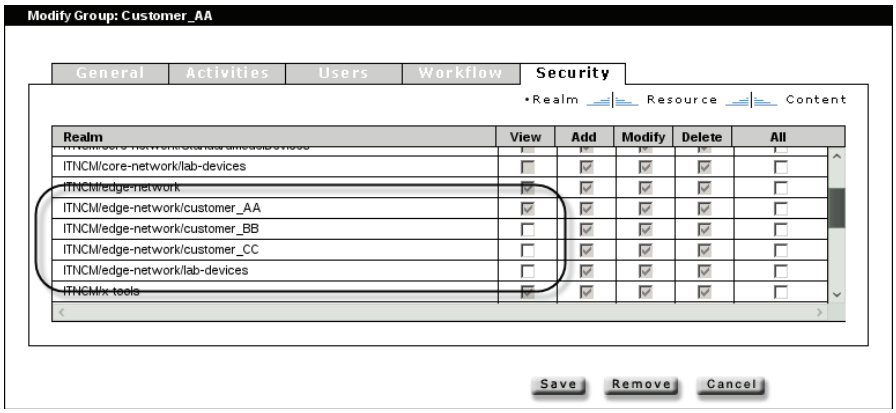
During the reporting installation, both configuration management and compliance reports are installed into Tivoli Common Reporting. Six reports exist that support various configuration management functions. Tivoli Common Reporting supports the creation and deploying of reports in various formats by using various methods. Reporting makes SQL calls to the same database that the presentation and worker servers are using, which provides up-to-date reports.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Security rules

Reports reflect the *view* rights of a group

Some reports, for example, user and group reports, require the *manage accounts* activity



© Copyright IBM Corporation 2016

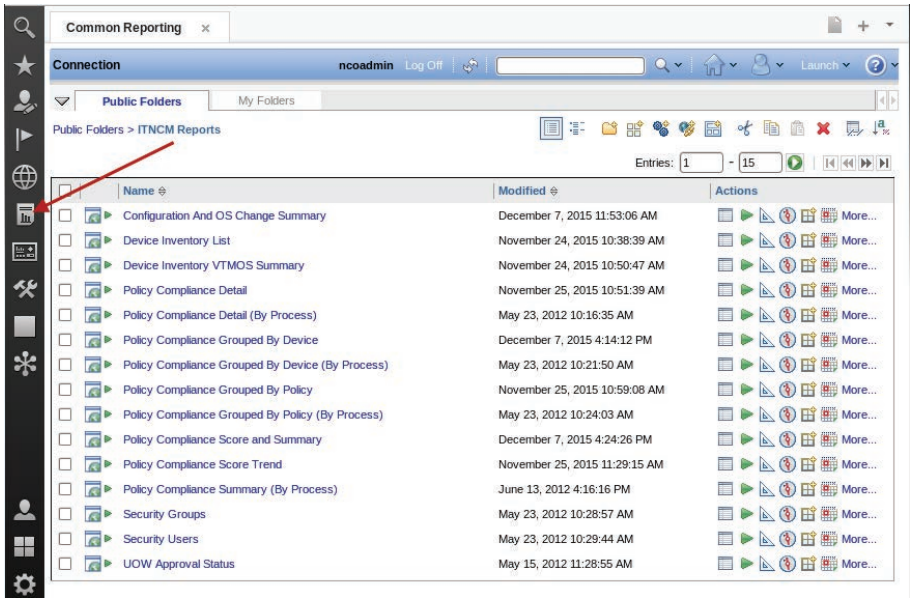
Security rules

Reporting uses the same security model as the Tivoli Netcool Configuration Manager GUI. If a user cannot see devices in the GUI, they do not see the devices in a report. These view settings are defined on the group administration page where view rights on realms are defined.

Some reports require *manage accounts* activities to see the full details of the report. This limitation is reflected in the *security groups* and *security users* reports. Without *manage accounts* activities that are associated with your group, the report displays only information about your user and your group.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

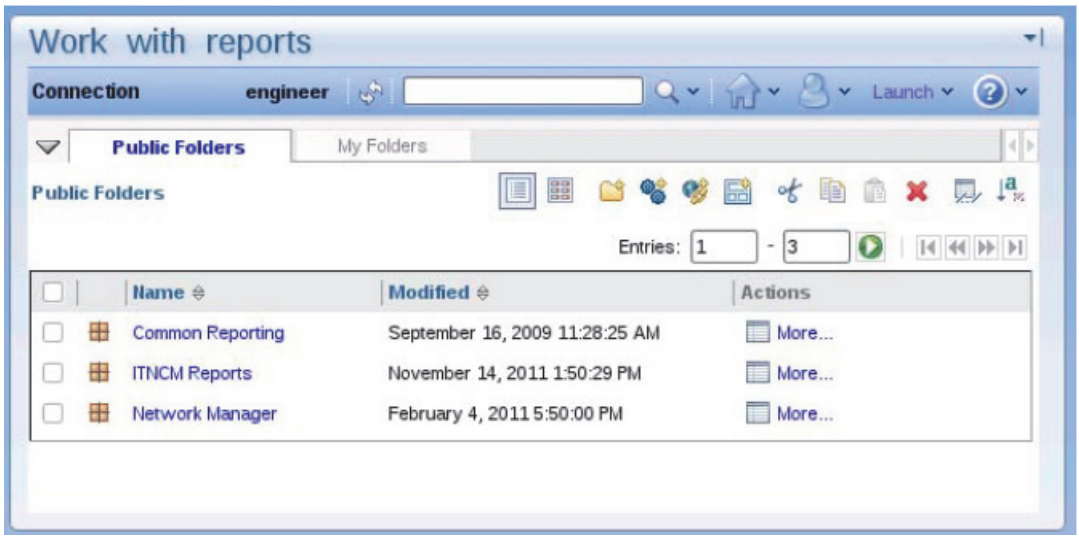
Accessing the reports in Dashboard Application Services Hub



© Copyright IBM Corporation 2016

Accessing the reports in Tivoli Integrated Portal

You can access reports through the *Common Reporting* feature of Dashboard Application Services Hub. You select the **ITNCM Reports** section that is shown in the **Public Folders**. You need to know where your Dashboard Application Services Hub instance is located and log in to it with your configuration manager credentials. If you configured single-sign-on for your solution, you used Dashboard Application Services Hub to start the configuration manager GUI and already have access to reports.



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Tivoli Common Reporting generates the report

Tivoli. software IBM.

Device Inventory VTMOS Summary

Selected Realms	ITNCM/edge-network/ customer_CC	Model	*
Vendor	*	OS	*
Type	*	Name	*

Vendor	Device Type	Model	OS
Cisco	Router	3640	C3640-I-M-12.3(5b)
			C3640-I-M-12.4(25c)
			C3640-IK9S-M-12.4(25c)
		Device Count For OS ->	3
	Device Count For Model,OS ->	1	3
Device Count For Type,Mode,OS ->	1	1	3
Total Device Count For Vendor ->			3

Total Vendors 1
Total Device Inventory 3

© Copyright IBM Corporation 2016

Tivoli Common Reporting generates the report

After selecting a report and defining the appropriate filters, Tivoli Common Reporting and Cognos are the underlying reporting generation tools.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Describe the difference between a ticket and a UOW
- Approve units of work
- Stop and restart units of work
- Search for data in the database
- Determine the differences between configurations
- Restore versioned configurations
- Review reports

© Copyright IBM Corporation 2016

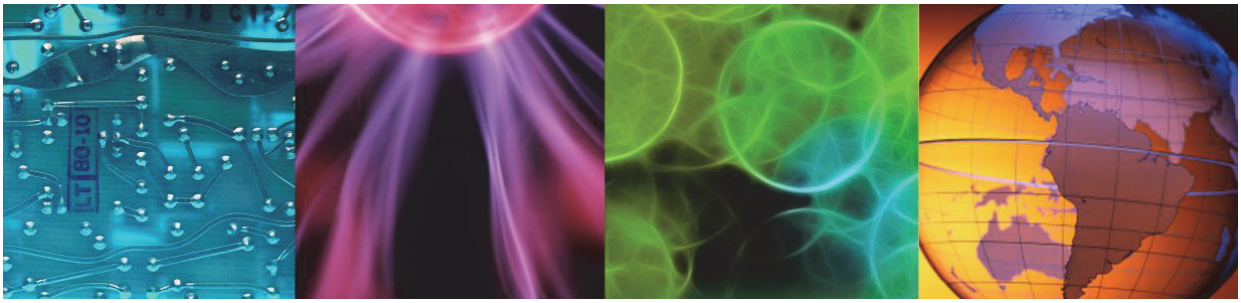
Summary

In this unit, you learned to operate Netcool Configuration Manager. The topics include approving *units of work*, stopping, and restarting *units of work*, processing complex searches, comparing configurations, restoring previous configurations, and viewing reports.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

6 Netcool integrations

Netcool integrations



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit covers the integration between Netcool Configuration Manager, Netcool/OMNIbus, and Network Manager. It also displays useful information in Dashboard Application Services Hub with the use of a configuration management event viewer.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Identify the integration points with Tivoli Network Manager
- Describe the event enrichment that is available for Netcool/OMNIbus
- Discover devices with Network Manager
- Delete customer CC devices
- Verify integration with Configuration Manager
- Apply a policy to devices from Tivoli Network Manager
- Apply a change to a device by using the device terminal
- View device history in Activity Viewer

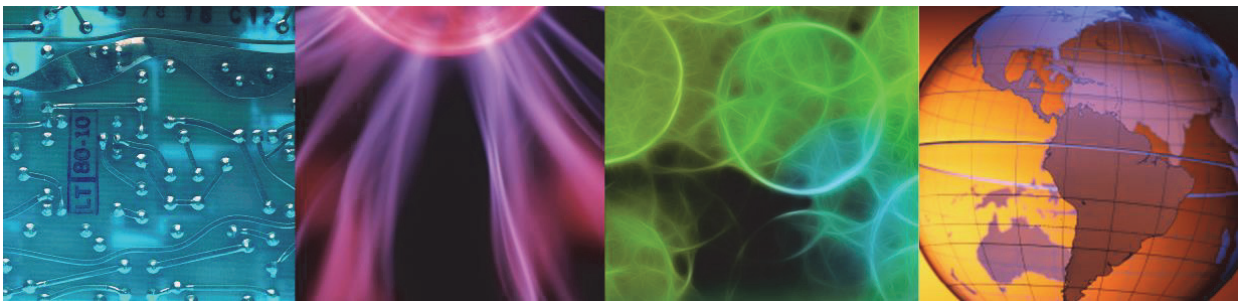
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Integration with other Netcool products

Lesson 1 Integration with other Netcool products



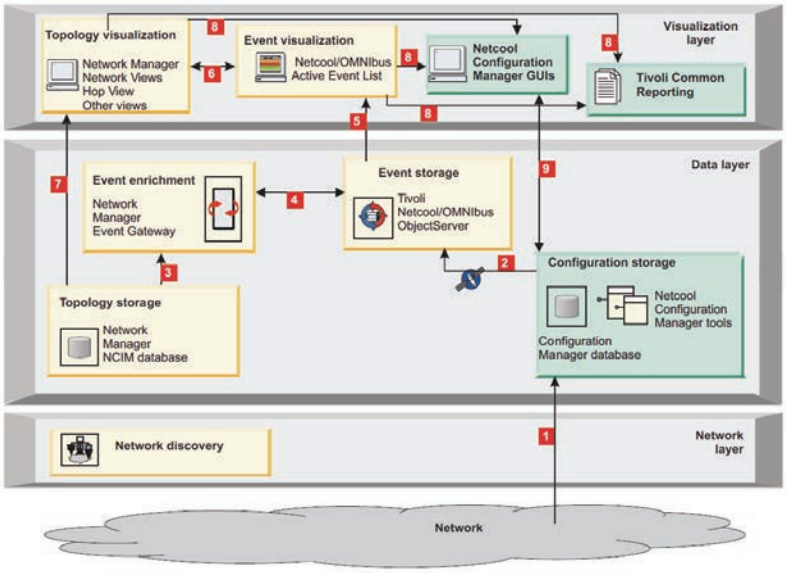
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe data flows between products
- Describe the SNMP traps created
- Describe the actions that can be taken from other Netcool products

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Integration data flow



© Copyright IBM Corporation 2016

Integration data flow

When Netcool Configuration Manager is part of the Netcool Network Management deployment, which includes Netcool/OMNIbus and Network Manager, data flows occur in the following manner.

1. When a configuration change or policy violation occurs in a network resource, information is sent to the configuration manager.
2. Netcool Configuration Manager logs the data and sends a corresponding event to the Netcool/OMNIbus ObjectServer with the SNMP probe.
3. Network topology data is stored in the Network Manager Network Connectivity and Inventory Model (NCIM) database and is made available to the event gateway.
4. Netcool Configuration Manager events are passed to the event gateway, where they are enriched with network topology data and then returned to the ObjectServer.
5. The Event Viewer in Netcool/OMNIbus shows that the configuration change and policy violation events are stored in the ObjectServer.
6. This event information is shared with the Event Viewer and Network Manager views.
7. Network Manager GUIs display network topology data that is stored in the NCIM database. This data is enriched with configuration change and policy event information from the ObjectServer.
8. Configuration changes and policy violations are represented in the Network Manager and Netcool/OMNIbus GUIs. By using the right-click menus, operators can open Tivoli Common Reporting and the Netcool Configuration Manager GUIs for extra analysis.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9. The full Netcool Configuration Manager operations remain available through the main configuration manager GUIs.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Current integrations

Importing devices that are found by network manager discovery

Sending SNMP traps to Netcool/OMNIbus

Enriching traps and topologies with IBM Tivoli Netcool Configuration Manager actions and reports

Activity viewer

© Copyright IBM Corporation 2016

Current integrations

The integration with the other Netcool products provides these items:

- Bulk importing from the Network Manager discovery results to Netcool Configuration Manager
- Sending SNMP traps to Netcool/OMNIbus or other trap collector, which includes traps for changes of device states, UOW executions, successes, and failures
- Enriching traps, topologies with in-context reports, and various Netcool Configuration Manager actions
- An activity viewer to show the change history of a device and recent compliance policy data that is in the Netcool Configuration Manager compliance tool

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Importing devices from network manager

You can discover devices in IBM Tivoli Network Manager and import them into IBM Tivoli Netcool Configuration Manager on regular basis

Network manager discovers elements on the network

Network Configuration Manager retrieves and manages configuration

© Copyright IBM Corporation 2016

Importing devices from network manager

Using the integration that is developed between Network Manager and Netcool Configuration Manager, you use the Network Manager discovery functions to find devices on the network. You can configure Netcool Configuration Manager to regularly import the results of the discovery functions and manage the configurations of the new devices.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring device synchronization

Netcool Configuration Manager automatically imports devices that are discovered by Network Manager

The installation configures this synchronization process

By default, the synchronization runs once every day

The following steps demonstrate how to modify that frequency

1. Edit the following file:

```
/opt/IBM/ncm/config/properties/rseries.properties
```

2. Change the following property:

```
#
# Label: 1440 is a Daily (in minutes) delay. Update time in minutes.
#
NMEntityMappingComponent/period=1440
```

Note: this change is optional

© Copyright IBM Corporation 2016

Configuring device synchronization

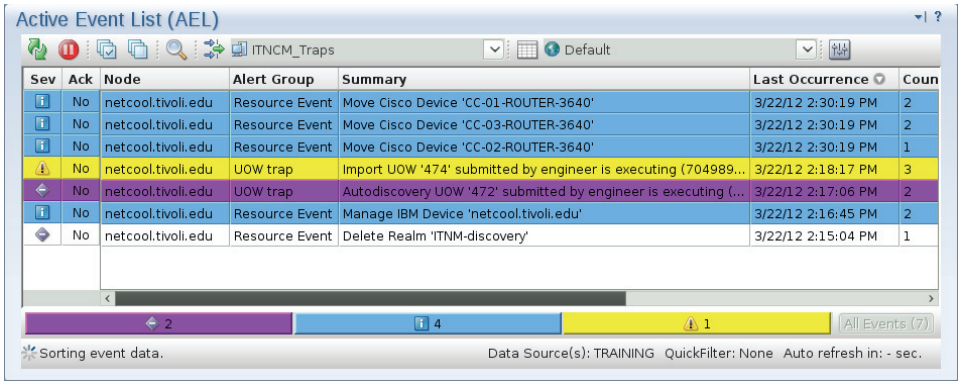
Device synchronization between Network Manager and Configuration Manager is automatic. Configuration Manager queries the Network Manager topology database periodically and looks for devices that Network Manager discovered. When it finds new devices, Configuration Manager retrieves the device configuration files.

The synchronization configuration is accomplished during the core installation process. No further configuration is required. However, the automatic frequency that the process uses is configured for one time each day. You can modify a property file and change the frequency.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Traps are sent to monitoring software

Provides configuration management status



© Copyright IBM Corporation 2016

Traps are sent to monitoring software

You can configure Netcool Configuration Manager to send traps to monitoring software, for example, Netcool/OMNIbus. You can monitor network change events and use them for troubleshooting issues that require root cause analysis.

These traps support other applications that are integrated with Netcool Configuration Manager. You can use traps either as notifications that their actions are complete or to cause applications to react to changes to the configuration database. For example, if a trap is received and indicates that a new device is imported, it causes an inventory system to obtain specific information about a device to update its database.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Trap types

Policy
- Device compliant or noncompliant notifications
- Device exempt or not assessed notifications

Unit of work
- Upgrades
- Synchs
- Changes

Task
- Device communication or authentication failures
- Device autodiscovery or model change failures
- Task failure or success notification

Resources
- Realm changes
- Resource changes

© Copyright IBM Corporation 2016

Trap types

Netcool Configuration Manager sends traps on the following actions that occur in the software. Policy traps can be sent on per-device basis when a policy is applied to a device and it is compliant, noncompliant, exempted, or not assessed. UOW traps are created when a *unit of work* begins processing and when it completes. *Units of work* are made up of tasks. Task traps are sent on a per device basis when tasks are processed against them. These traps notify whether a task is a success or failure and provide details about the failure, when one is detected. Finally, when resources in the system are modified, traps are sent notifying of these changes

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

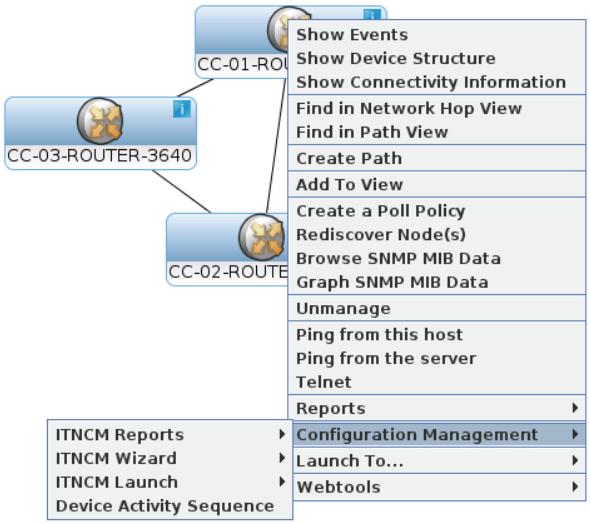
Enriching topology and events

Context-specific reports

Context-specific wizards

Thick-client launch

Device activity viewer



© Copyright IBM Corporation 2016

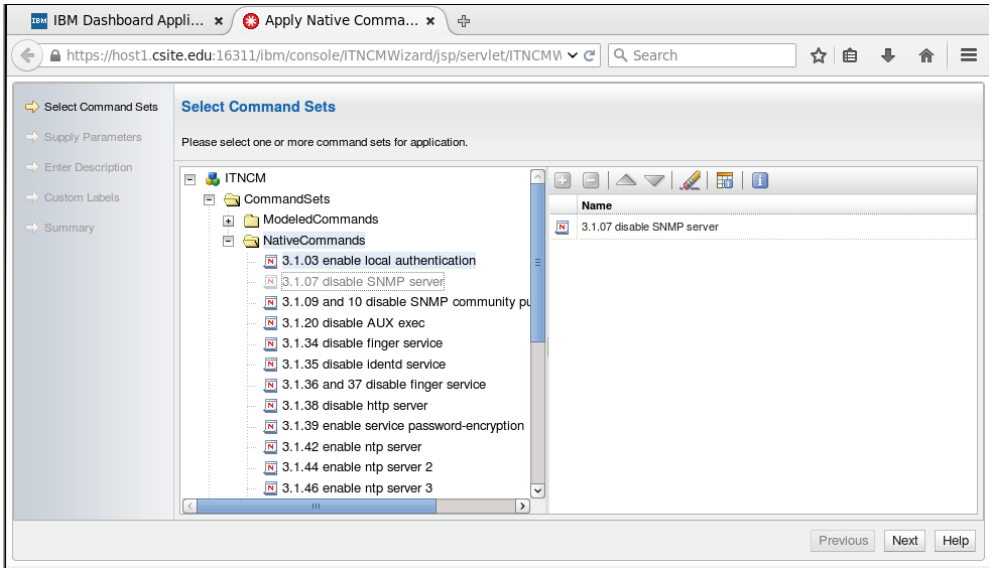
Enriching topology and events

From within Network Manager views and the Netcool/OMNIbus Event Viewer, you can access several Netcool Configuration Manager functions:

- You can run context-specific **ITNCM Reports** about the selected device in Dashboard Application Services Hub.
- You can use the **ITNCM Wizard** to process configuration management functions and apply compliance policies, command sets, and device synchronizations to a specific device.
- You can run a context-specific **ITNCM Launch** against a device and open the thick client tools to support more detailed operations. These actions include a device terminal session and other configuration manager and compliance tools.
- You can use the **Device Activity Sequence** to show in Dashboard Application Services Hub a graphical history of what configuration management actions occurred on a device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Embedded wizards to initiate actions



© Copyright IBM Corporation 2016

Embedded wizards to initiate actions

Wizards for configuration management operations are available in Dashboard Application Services Hub. You can set up the following actions within Dashboard Application Services Hub:

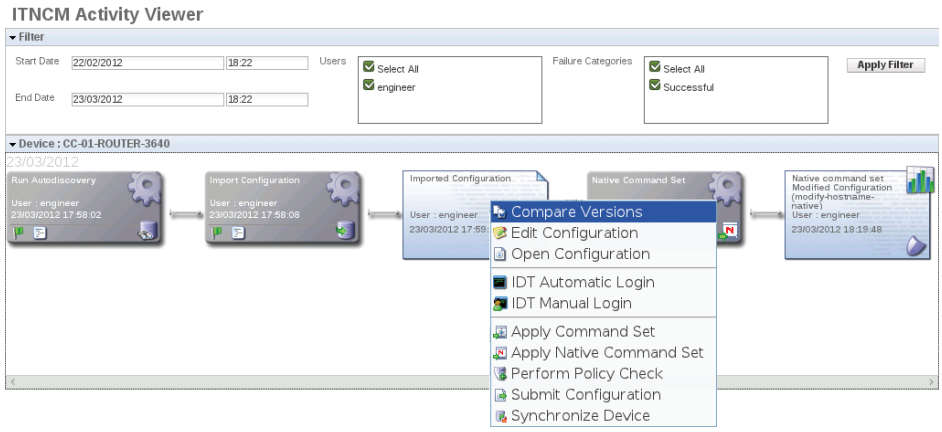
- You can apply a policy against the selected device to determine its compliance.
- You can apply either native or modeled command sets against the selected device.
- You can synchronize the configuration of the selected device. When differences exist between the configuration on the device and the configuration in the database, the device is modified to match the database configuration.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Activity viewer

View the state transitions of a device and current compliance

Access IBM Tivoli Netcool Configuration Manager utilities



© Copyright IBM Corporation 2016

Activity viewer

The *activity viewer* (or Device Activity Sequence) is a graphical representation of the *units of work* and terminal sessions that occur on a device. You can view the history of a device from a configuration management perspective and access Netcool Configuration Manager functions. These functions include accessing the configuration editor, initiating command sets, starting a device terminal, and running compliance checking utilities.

The activity viewer allows a user to determine the possible causes of a device that has either become unstable or out-of-compliance.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and follow the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Identify the integration points with Tivoli Network Manager
- Describe the event enrichment that is available for Netcool/OMNIbus
- Discover devices with Network Manager
- Delete customer CC devices
- Verify integration with Configuration Manager
- Apply a policy to devices from Tivoli Network Manager
- Apply a change to a device by using the device terminal
- View device history in Activity Viewer

© Copyright IBM Corporation 2016

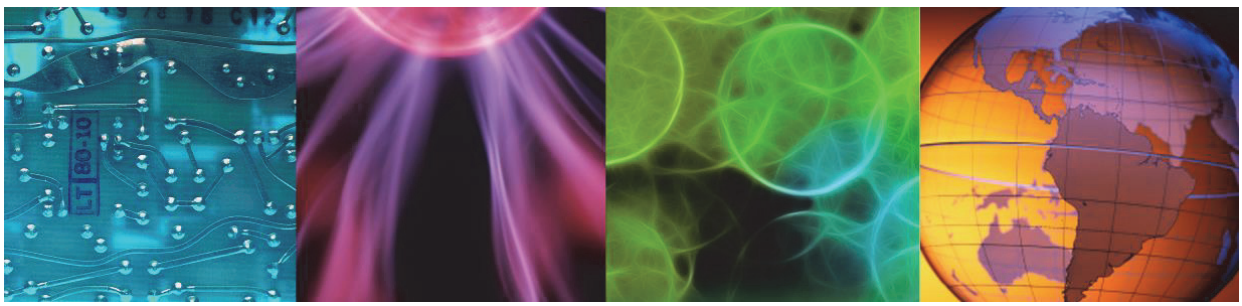
Summary

This unit taught the integration between Netcool Configuration Manager, Netcool/OMNIbus, and Network Manager. It also shows useful information in Dashboard Application Services Hub with the use of the configuration management event viewer.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

7 Authentication and authorization model

Authentication and authorization model



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit provides lessons about the Netcool Configuration Manager security model. You learn how security is implemented, based on realms and groups. You also learn the authentication and authorization structures and how Tivoli Network Manager domains align with Netcool Configuration Manager realms. You learn how *view*, *add*, *modify*, and *delete* capabilities are implemented in the application.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Create and verify groups and users
- Describe the group security that users are associated with
- Explain how realms control access to a network
- Assign security attributes and describe the authentication mechanisms to access the application
- Describe differences between domains and realms
- Implement view, add, modify, and delete rights throughout the application

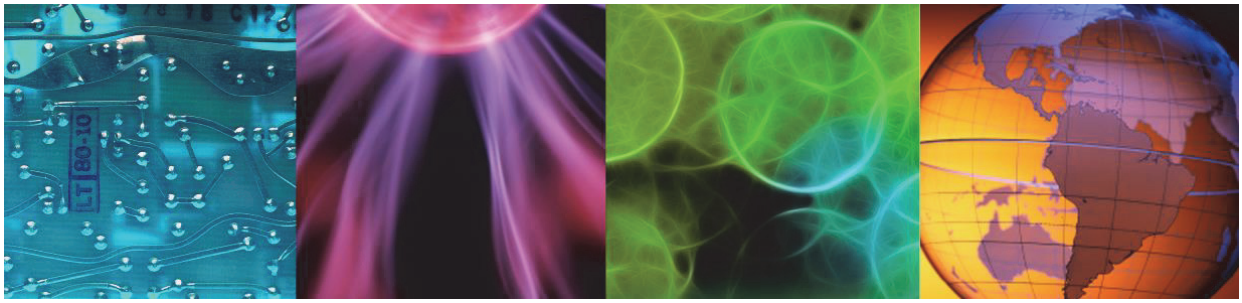
© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Enabling access control

Lesson 1 Enabling access control



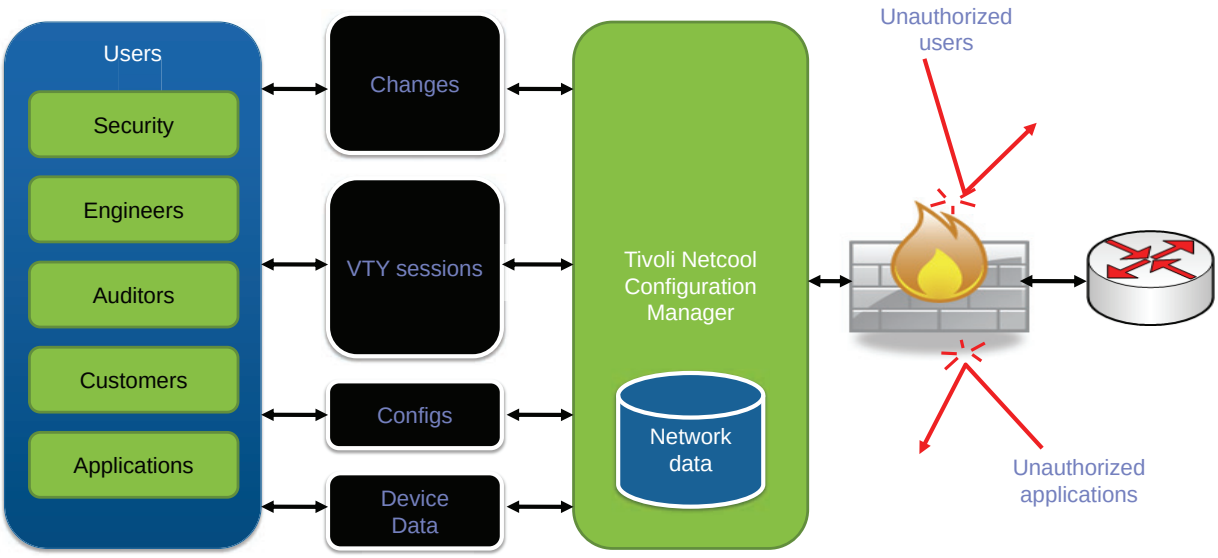
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe user authentication methods
- Describe a process for defining your security structure

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Gateway to network devices



© Copyright IBM Corporation 2016

Gateway to network devices

Netcool Configuration Manager is like a gateway to access the physical device to retrieve data that is not available through protocols like SNMP. Many stakeholders must gain access to the command-line interface of the device and possibly implement changes and retrieve important data. These stakeholders can be people who support roles in the security, engineering, and auditing organizations. They can be customers whose hardware you are responsible for managing. Finally, applications might require device data to support their operations. All these stakeholders can gain access to these services though either the GUI or API. Because Netcool Configuration Manager uses defined pathways, you can configure the filtering mechanisms that the network uses, access lists and firewalls to prevent unauthorized access.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

User authentication

Logging in to the application, not the network device

Internal authentication

External authentication

- Storing only user names in a database
- Passing credentials to vendor authentication systems
- Supports single sign on with Dashboard Application Services Hub

Authentication issues and precautions

- No fallback when the authentication server is down
- Some internal authentication user names should be kept

© Copyright IBM Corporation 2016

User authentication

User authentication, in this instance, is how people or applications authenticate to the Netcool Configuration Manager application and not the network devices. Both internal and external authentication are available. Internal authentication is when the Netcool Configuration Manager database stores user names and passwords. External authentication is when the user name and password are on an external server to verify that the credentials supplied are the correct ones. In an external authentication, the user name is all that is saved in the database. Credentials are passed to different authentication servers, including RADIUS or TACACS servers. Authentication can also be integrated with Dashboard Application Services Hub to allow single sign-on capabilities.

Currently, users are either externally or internally authenticated, not both. In other words, if the external server is not accessible, those users cannot log on to the application.

Important: Set up some internally authenticated users in case issues occur with the external authentication mechanism.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining access control

Know the tools for defining and enforcing policy

Define the structure as early as possible

Define the realm structure for organizing devices

Define groups, group roles, and group responsibilities

Determine the intersection between groups and realms

Expect this process to be iterative

© Copyright IBM Corporation 2016

Defining access control

To develop the security model for a customer, you must understand the tools for enforcing security. Understanding these tools and laying out a security plan before you implement it prevents security holes.

Defining the security plan early through whiteboard exercises with the stakeholders who access network devices is critical. You might want to develop a realm structure that reflects how the network is organized. Developing this structure first makes creating groups easier, especially when realm limitations are imposed on specific groups.

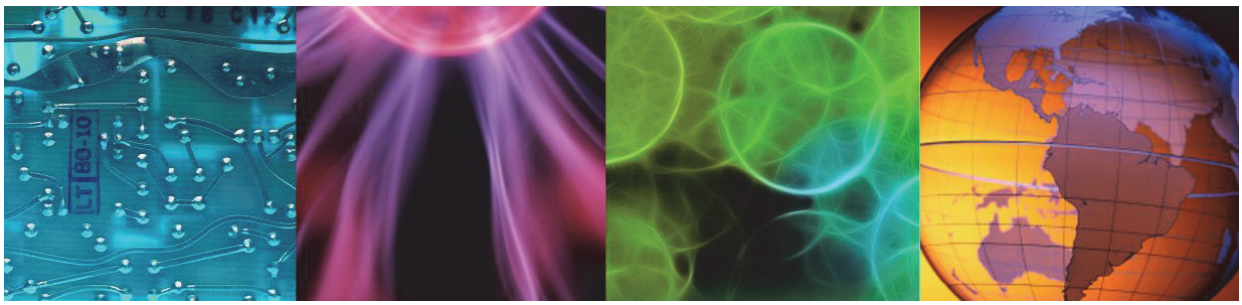
Develop the groups and determine the responsibilities for each group. Then, determine the intersection between realms and groups to ensure that users see only the devices that they are authorized to see and change.

Assume that this process is iterative. Stakeholders might not be available or understand the effect of the decisions that are made in these early steps for defining security.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Realms

Lesson 2 Realms



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe realms and their use
- Describe differences between domains and realms

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

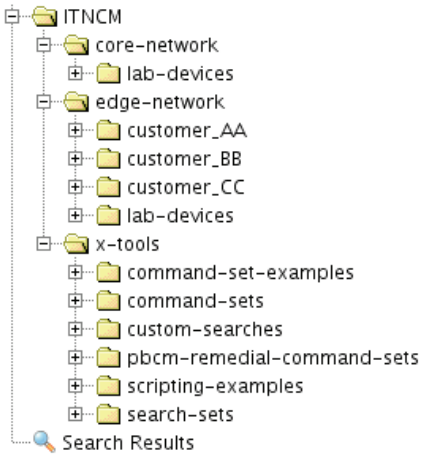
Realm structure

Basis of the security structure

Same concept as an NT or Linux file system, read, write, and run

Container for resources

- Network devices
- Passwords
- Other realms



© Copyright IBM Corporation 2016

Realm structure

Realms are the basic security structure that all other items are built on. The realm structure is like a Windows or Linux file system. You can define read, write, and execution privileges for different groups or users.

The realm structure is a set of containers for organizing network devices, passwords, and other realms. You can divide the network into the most logical areas for a customer or your organization.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Realm structure by customer

As simple or as complex as required

A virtual file system

- Hosted in a relational database
- Shared structure
- Hierarchical
- Resources can be bilocated with use of shortcuts

Aligned by geography, customer, service, device role

© Copyright IBM Corporation 2016

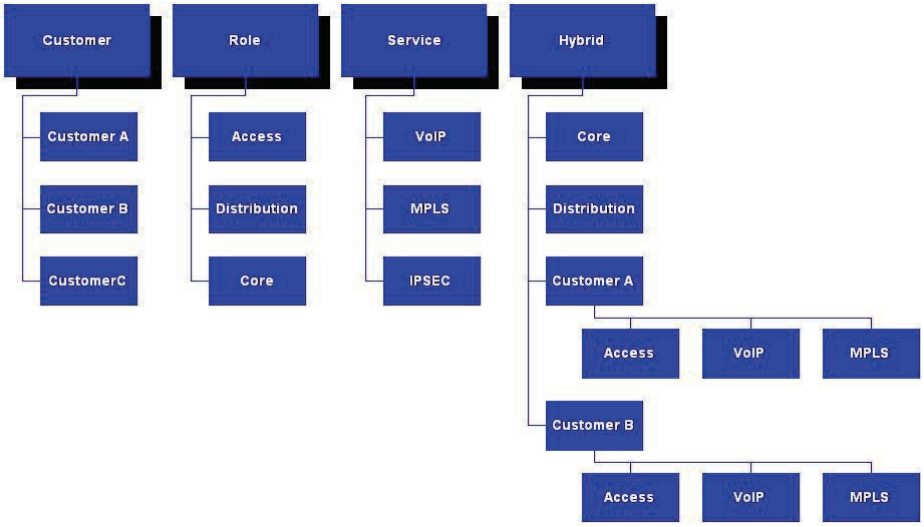
Realm structure by customer

The realm structure is based on the customer. Other than the base realm name, all other folders are created to organize network devices that the customer wants to manage. It can be as simple or as complex as required to reflect the nature of the network. Managing 300 devices versus 30,000 devices requires different realm structures to manage the differences of scale.

The realm structure is a virtual file system. All the realms and resources that are managed within it are contained within the database. It is a shared structure that people can access, but it is not a private user space in this system. It is a hierarchical structure that creates parent-child relationships. With shortcuts, you can have resources in different locations. Although the customer develops the realm structure, it is typically created for geographical, organizational, customer, or service type roles.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Realm structure examples



© Copyright IBM Corporation 2016

Realm structure examples

This slide illustrates some examples of possible realm structures that a customer can build to organize the devices that they are responsible for managing. You can break up your network by customers and device roles. Often times when working with large telecommunication companies, they use a hybrid structure that breaks up their network by core devices and then customer-related network devices.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Domains and realms

Tivoli Network Manager has domains

- Domain is a boundary by which information is discovered in a network
- Domain is a physical location of a network and its nodes

Tivoli Network Configuration Manager has realms

- Realm is a boundary by which user security is employed
- Realm is a logical location of a network, by role, customer, or other criteria

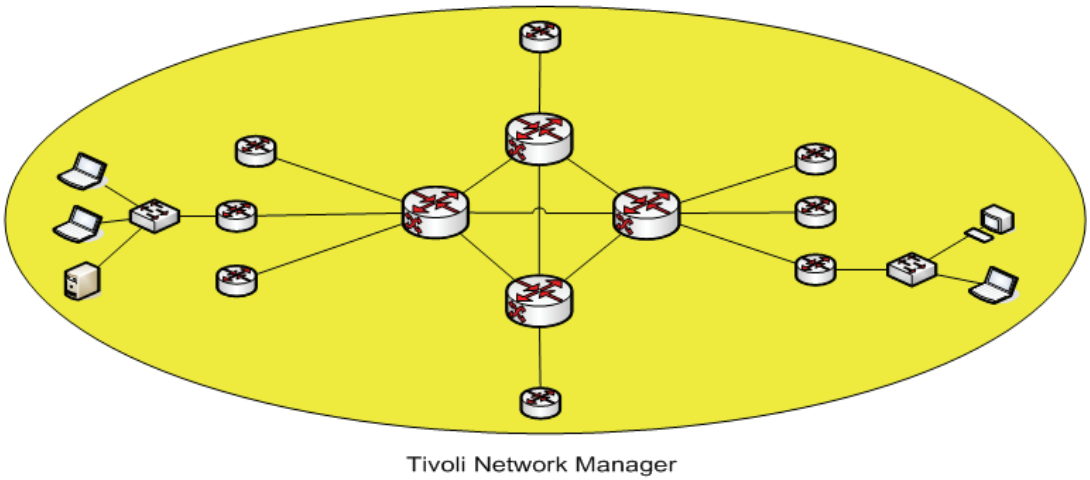
© Copyright IBM Corporation 2016

Domains and realms

When working with the Netcool Network Manager application, which includes both Tivoli Network Manager and Netcool Configuration Manager, it is important to know the difference between domains and realms. Tivoli Network Manager is used for device discovery and network topology. It uses the concept of domains. Domains refer to a boundary within which devices are discovered and their physical topology, which is logged within the application. A domain is a much broader concept than a realm. Realms are logical boundaries which security employs to ensure that authorized people see and interact only with authorized network data.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Tivoli Network Manager domains



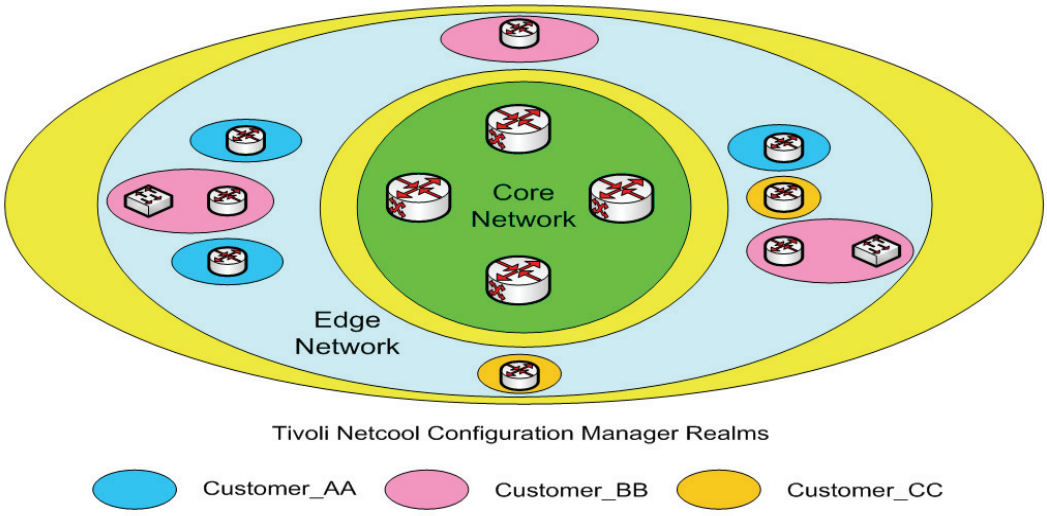
© Copyright IBM Corporation 2016

Tivoli Network Manager domains

For example, a domain defines a large area in the network where devices are discovered and linkages are mapped to reflect their physical connectivity.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Tivoli Netcool Configuration Manager realms



© Copyright IBM Corporation 2016

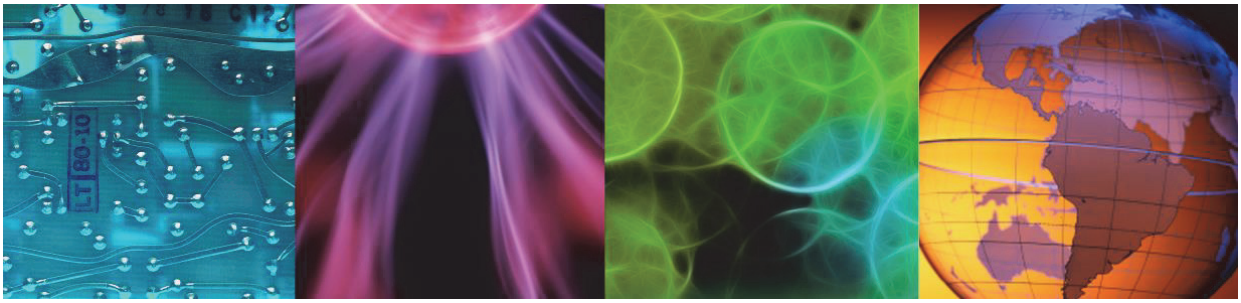
Tivoli Netcool Configuration Manager realms

Unlike domains, realms are a much more granular concept. Devices are not physically connected in realms. They are groupings that reflect the organizations that have responsibility for the devices. They can also be used to distinguish one set of devices from another, for example, Customer_AA devices versus Customer_BB devices.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Groups and users

Lesson 3 Groups and users



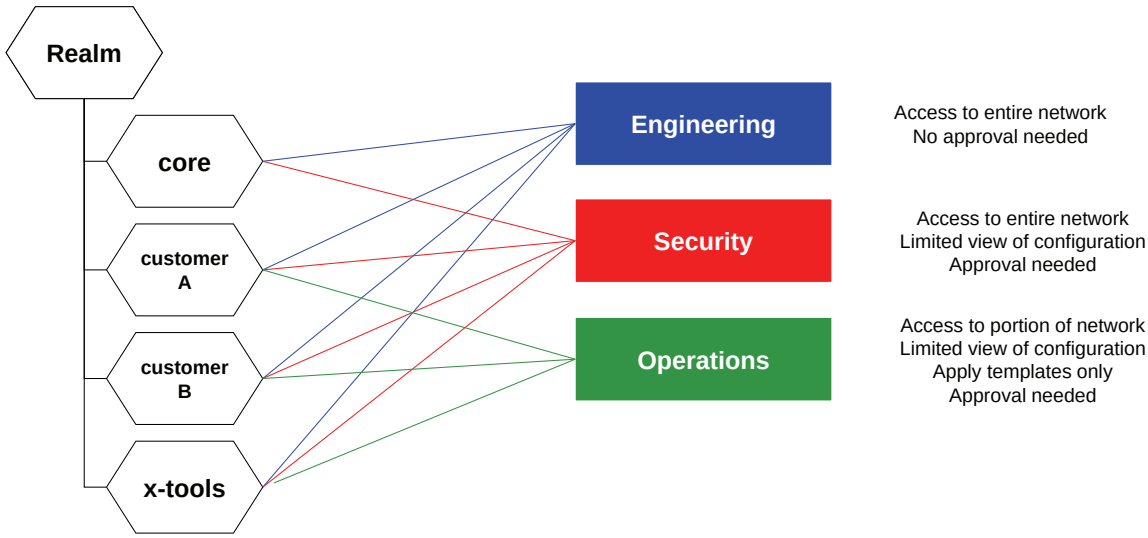
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Create groups.
- Create users.
- Associate users with multiple groups.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Determining how groups interact with realms



© Copyright IBM Corporation 2016

Determining how groups interact with realms

As mentioned earlier, defining a realm structure and groups is a process that should be done in a group meeting of network stakeholders. In this example, three organizations require access to network data to change it. This realm structure has core network assets and customer network assets. Notice each group and the organizational policy for each one.

The engineering group is a group of technical experts who understand how the network is configured and how their changes affect the network. Their expertise and capabilities authorize them to access all the network assets. They need no approvals for changes.

The security group is responsible for all security policies. They can access the entire network, but only a portion of the configuration is shown to them. They are only authorized to see what they must see for configuration settings. They are responsible for ensuring that security protocols are enforced. They must change configurations and obtain approval for their network changes. Typically, the engineering group approves the changes.

The operations group has the junior technical staff. Their view of the network is limited to the customers. They have limited access. If these operators submit changes, they can use templates that the engineering group provides. Like the security group, their views of configuration data are limited, and any changes they make must be approved.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Group structure

Groups implement a *control plane* concept

Groups define roles and responsibilities, implements business policy

Groups define the view and modify rights inside the realm structure

© Copyright IBM Corporation 2016

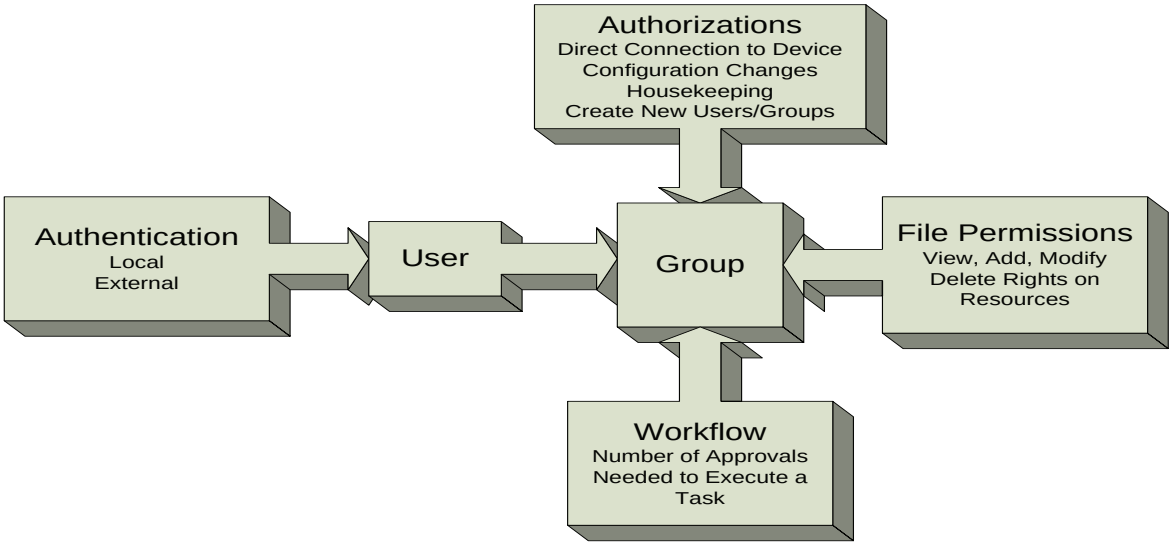
Group structure

Where realms are used to organize the network and create the framework for security, it is the group that creates a control plane. When you log in to the application, the group settings control what you can do within the application.

The group determines the roles and responsibilities of its members. Essentially, the group is where business policy is applied inside the application. Each user belongs to a certain group and has a job description. Using the job description as a guide, the job's responsibilities must translate into a set of software permissions. Groups also determine the rights that users have inside the realm structure.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Group permissions basics



© Copyright IBM Corporation 2016

Group permissions basics

Most security settings are applied to the group. The one exception is how you authenticate: remotely or locally.

Three settings are associated with a group:

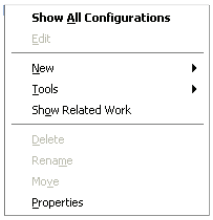
- Authorization or activity settings that define what actions you can access in the application
- Workflow settings that define how many approvals are needed to effect a change to the network
- Permission settings that define where within the realm structure the user can effect changes to or even see network devices

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Permission settings

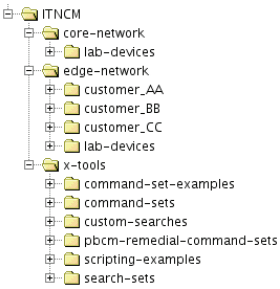
Menus

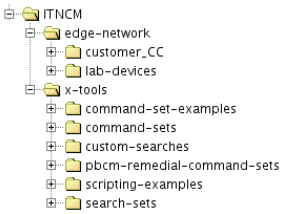
Some items are unavailable



Viewable realms

Some realms are hidden





© Copyright IBM Corporation 2016

Permission settings

Permission settings affect what menus are available in certain realms. Permissions also affect what realms are viewable. If a realm is not viewable, menus are irrelevant.

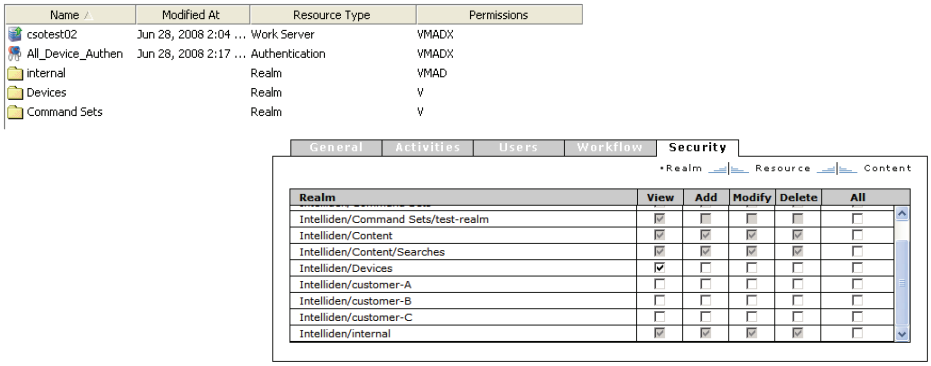
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Realm security and file permissions

Security is placed on both realms and the items within them

Administrators define view, add, modify, delete, and execute rights for the group

- Permissions are inherited by default
- Permissions show in the GUI as VAMD(X)



© Copyright IBM Corporation 2016

Realm security and file permissions

You apply security by group to both realms and the files or resources that are in a realm. The following options are available:

- You can **View** the realm or resource
- You can **Add** a new realm or resource
- You can **Modify** a realm by renaming it or edit non-network resources
- You can **Delete** a realm or a resource
- You can **Execute** a change on a network resource

Permissions are inherited when the administrator does not explicitly break the inheritance. You can view in the GUI what your permissions are on both the realms and the items within them.

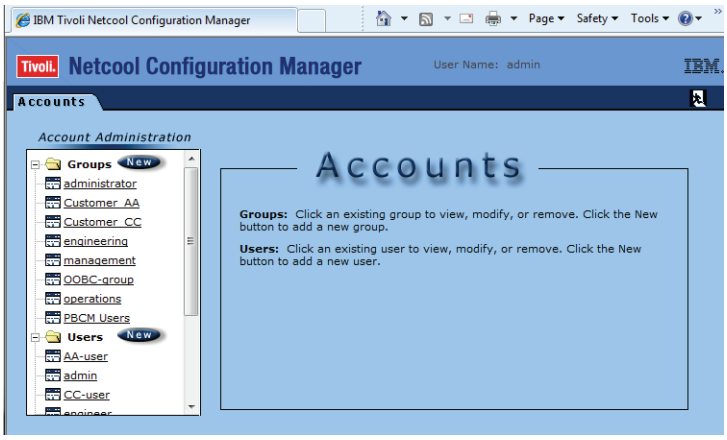
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a group

Use a internet browser

Use http://<itncm-server>:16310/intelliden.jsp (default port is typically 16310)

Have **Manage Account** privileges



© Copyright IBM Corporation 2016

Creating a group

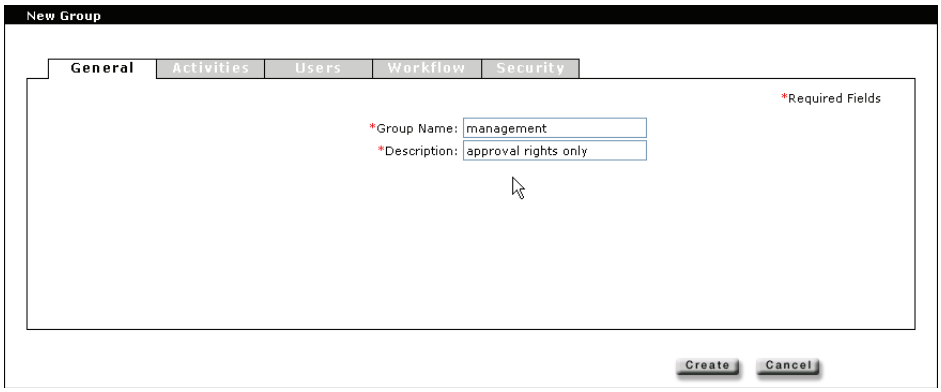
To create a group, you use an Internet browser, either Mozilla or Microsoft, to log in to the account manager. Go to **http://<itncm-server>:16310/intelliden.jsp** or use the profiles that are presented when you log in to the standard URL, **http://<itncm-server>:16310**. You must belong to a group with *managed account* privileges to access the account manager.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a group name

Define the group name and description

Fields with asterisks (*) are required items



© Copyright IBM Corporation 2016

Creating a group name

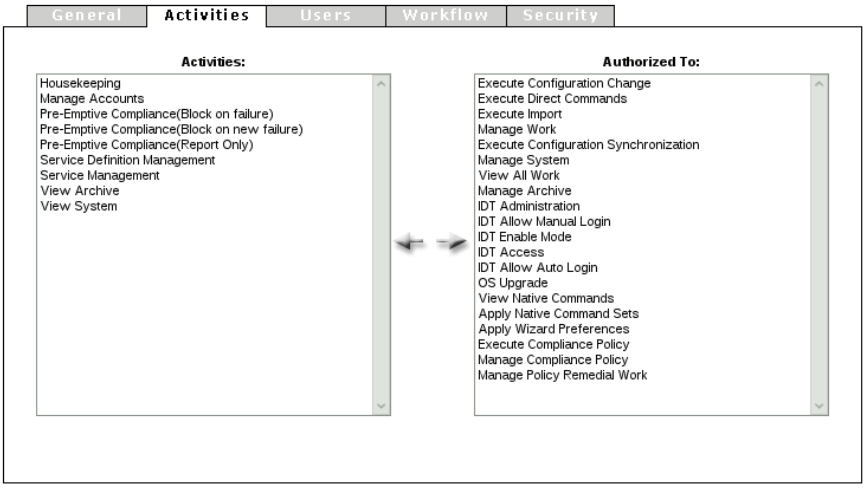
When you create a new group, you must provide a group name and a description. Anywhere you see red asterisks in the GUI, you must include required data. You cannot change the group name after you save it.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Implementing group activities

Activities are the authorizations that are associated with a group

A group does not need to have activities associated with it



© Copyright IBM Corporation 2016

Implementing group activities

An important aspect of creating a group is the authorizations or activities that you give it. These activities enable a number of capabilities. The activities allow changes to devices, and access device terminal sessions and compliance capabilities. You do not have to give a group any activities. However, the group is restricted to limited capabilities with the application.

As of the Netcool Configuration Manager 6.4.2 releases, you can apply theses activities to a group. The average group needs the activities that are highlighted with an asterisk (*).

- ***Apply Native Command Sets**: Users can apply native command sets. However, a user is authorized to apply a modeled command set without this activity.
- ***Apply Wizard Preferences**: Users can change the UOW Submission wizard settings in user preferences.
- ***Execute Compliance Policy**: Users can apply policies against authorized devices in the compliance manager
- ***Execute Configuration Change**: Users can apply a configuration change UOW.
- ***Execute Configuration Synchronization**: Users can perform configuration synchronizations on one or more resources.
- **Execute Direct Commands**: Users can call a function on irm\ResourceManagerSB to submit a command directly to a network resource. This feature is designed for API users
- ***Execute Imports**: Users can import new resources into ITNCM - Base.
- **Housekeeping**: Users can clean up completed *units of work* and previous configurations through the *work housekeeping* and *configuration housekeeping* utilities.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

- ***IDT Access**: Users can access IDT through the tools menu. Users can access login modes and view their logs.
- **IDT Administration**: Users can view active sessions and device logs for all other users.
- **IDT Allow Auto Login**: Users can use automatic login for the device. Must have IDT access permission also.
- ***IDT Allow Manual Login**: Users can use manual login to gain access to the device. Must have IDT access permission also.
- **IDT Enable Mode**: Users can authorize IDT to gain access to *enable mode* on a device, when a device script is used. IDT users must also have *allow auto login* mode.
- **Manage Accounts**: Users can add, modify, and delete users and groups from the system. Modifications can include changes to data scope and changes in permissions of any group.
- **Manage Archive**: Users can run the *archive* housekeeping utility in icosutil from the command line.
- **Manage Compliance Policy:** Users can create, edit, and delete policies in the compliance manager
- **Manage Policy Remedial Work:** Users can release remedial actions that modify devices from the compliance manager
- **Manage System**: Users can pause or restart ITNCM - Base. Users can also denote a realm as a system realm. Users are also able to change values in the Property Manager GUI.
- **Manage Work**: Users can approve, reject, or dequeue any UOWs submitted through the GUI or API. Users without manage work permissions can dequeue their own *units of wor*k. However, to dequeue the UOW of another user, *manage work* rights must be granted. If this permission is not enabled, the user cannot override the default *pre-emptive compliance* settings for the other three *pre-emptive* activities.
- **OS Upgrade**: Users can run OS upgrades.
- **Pre-Emptive Compliance (Block on Failure)**: This permission blocks an *apply command set* or *submit config* from completing if any *compliance* failure exists against the projected configuration.
- **Pre-Emptive Compliance (Block on new failure)**: This permission blocks an *apply command set* or *submit config* from completing if the change makes the device more noncompliant.
- **Pre-Emptive Compliance (Report Only)**: This permission reports the status of pre-compliance in the audit log, changes to the device are still made.
- **Service Template Management**: Users can create and delete service templates in the Network Service Management database.
- **Service Management**: Users can use the GET, POST, and DELETE methods for services on the Network Service Management Interface.
- **View All Work**: Users can generate reports, which display work that all users submit. If the user does not have this activity, they can view only the work submitted by their groups.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

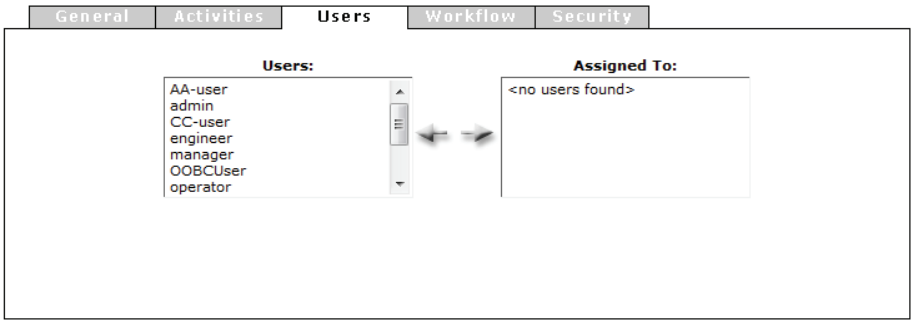
- **View Archive**: Users can view the archive manager in the user interface.
- **View Native Commands**: Users can view native commands on a network resource.
- **View System**: Users can view the systems manager in the user interface.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Adding users to a group

Associate existing users with a group, optional



© Copyright IBM Corporation 2016

Adding users to a group

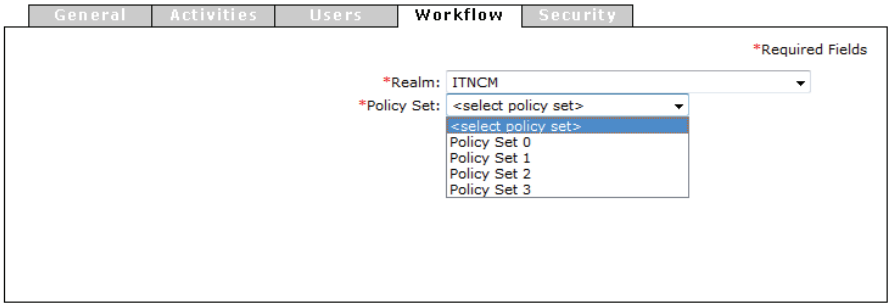
Ignore the **Users** tab in the group creation process, when users are not created yet. When LDAP is configured as a user repository, users are created and deleted within LDAP.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining an approval policy

Determine how many approvals a user needs, in this group, to run a unit of work

Determine, by realm, who sees the approval tickets, only relevant when group has **Manage Work** activities



© Copyright IBM Corporation 2016

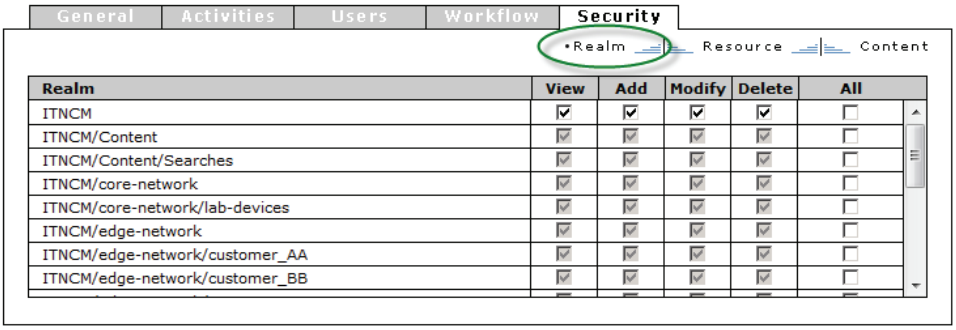
Defining an approval policy

Another important aspect of creating a group is deciding whether you want to implement approvals on the *units of work* the group submits. On the **Workflow** tab, you define a workflow *realm* and a *policy set*. The workflow *realm* applies only to groups with managed work activity. The policy set defines how many approvals *units of work* need that touch network devices. A **Policy Set 0** means that it needs no approvals to submit any unit of work. A **Policy Set 3** requires three approvals to submit most units of work. There must be at least three approval groups that are defined to approve tickets or it cannot submit a UOW.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Define realm rights

Define whether users can view, add, modify, or delete realms



© Copyright IBM Corporation 2016

Define realm rights

The final tab in the group creation process is the **Security** tab. Here you decide what portions of the realm structure are visible to the group and what can be modified within it. The **Security** tab is made up of three subpages that define various levels of view, add, modify, and delete rights for realms, resources, and content.

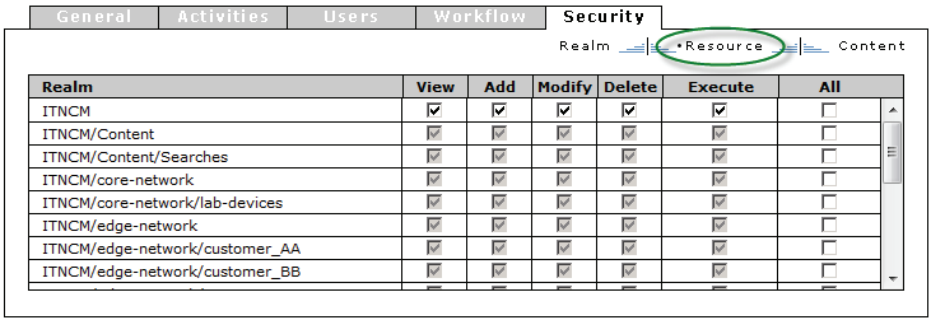
On the Realm page, you control what a user can do with the realm structure. If a check box has a gray check mark, permissions are inherited from the parent. You can implement the following rights:

- Giving the group *view* rights on realms is an important setting. Without this permission, you never see the underlying resources that are in the realm or its subsequent subrealms.
- Giving the group *add* rights allows them to add a subrealm. This permission plus *delete* is needed to move a realm.
- Giving the group *modify* rights on a realm allows the renaming of it.
- Giving the group *delete* rights allows a user to delete a realm. This permission plus *add* is needed to move a realm.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Define resource rights

Define whether users can view, add, modify, delete, or run resources in the realms



© Copyright IBM Corporation 2016

Define resource rights

The Resource page sets permissions to determine what users can do with non-realm resources in a particular realm. If a check box is not accessible, permissions are inherited from the parent.

Users with the appropriate permissions can use these options on the resource security page:

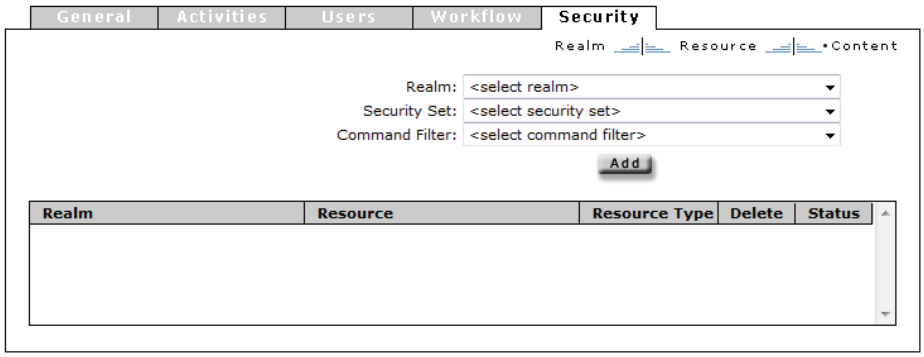
- Giving a group *view* right allows user to view items in a realm. For example, view a device's configuration.
- Giving a group *add* rights allows users to add a non-realm resource to the realm. For example, a new device or set of passwords.
- Giving a group *modify* rights allows a user to modify non-network resources in a realm. This security setting allows users to modify or view authorized data in resources that control access to network devices. Without this privilege, users cannot view the passwords in an authentication resource in the affected realm.
- Giving a group *delete* rights allows a user to delete a resource from the realm. *Delete* and *add* privileges are needed to move a nonrealm resource.
- Giving a group *execute* rights allows a user to apply configuration changes to a network resource in that realm.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining content rights

Determine what users can see in a modeled configuration

Determine what users can submit in a device terminal



© Copyright IBM Corporation 2016

Defining content rights

Content security applies security settings that affect two areas:

- Security sets are applied to the group and realm pairs to control what a user can do and see inside a modeled configuration. This action is essentially placing *view*, *add*, *modify*, and *delete* rights on commands inside the configuration.
- Command filters are applied to the group and realm pairs that control what commands can or cannot be sent over a device terminal session.

These topics are addressed later in this class.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Group modification policies

After you create a group, you cannot change its name

You cannot modify a group that you belong to

Removing a group does not delete the users who are associated with it

© Copyright IBM Corporation 2016

Group modification policies

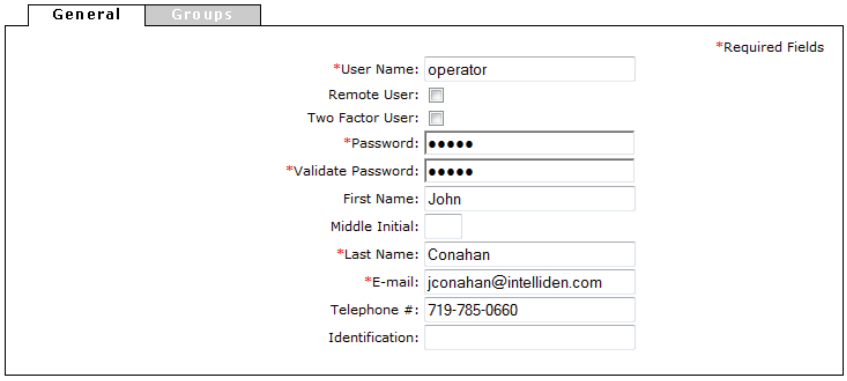
After you create a group and save it, you cannot change its name. You are also prevented from changing any details about a group you are a member of. Finally, if you delete a group, you do not delete the users who belong to that group. If those users do not have membership in another group, they are prevented from logging in to the application.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating users

Fields with asterisks (*) are required items

Remote User uses external authentication mechanisms



© Copyright IBM Corporation 2016

Creating users

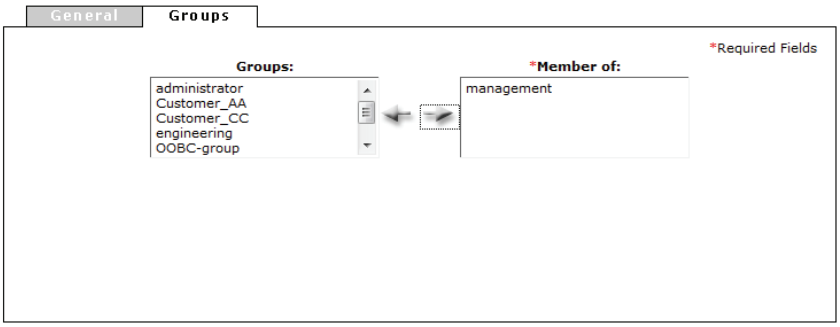
There is no requirement to create groups before users. However, a user that is not associated with a group cannot log in to the application. When creating a user, a number of the fields have red asterisks beside them denoting required data. By selecting the Remote User option, you are telling Netcool Configuration Manager to use the external authentication mechanism that is configured in WebSphere. The Two-Factor User is no longer supported and should not be checked.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Associating a user to one or more groups

User must belong to at least one group to log in

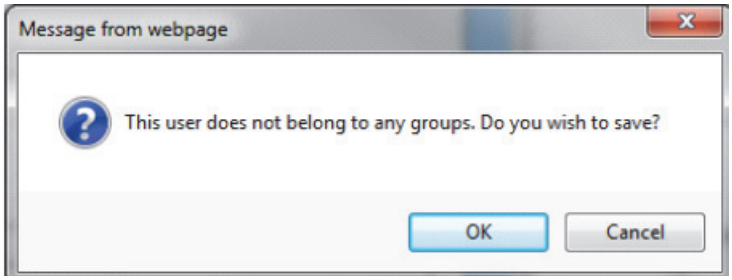
Generates a warning when no group is defined



© Copyright IBM Corporation 2016

Associating a user to one or more groups

A user must belong to at least one group. Sometimes, administrators create users and forget to associate them with a group. If that is the case, a warning is generated and shown.

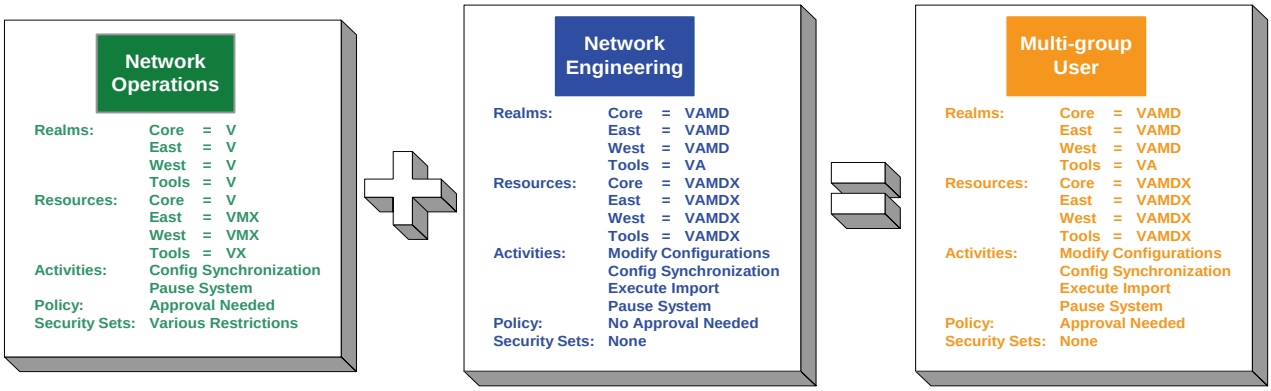


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Associating a user to multiple groups

Users can belong to multiple groups

User rights are the sum of the associated groups rights



© Copyright IBM Corporation 2016

Associating a user to multiple groups

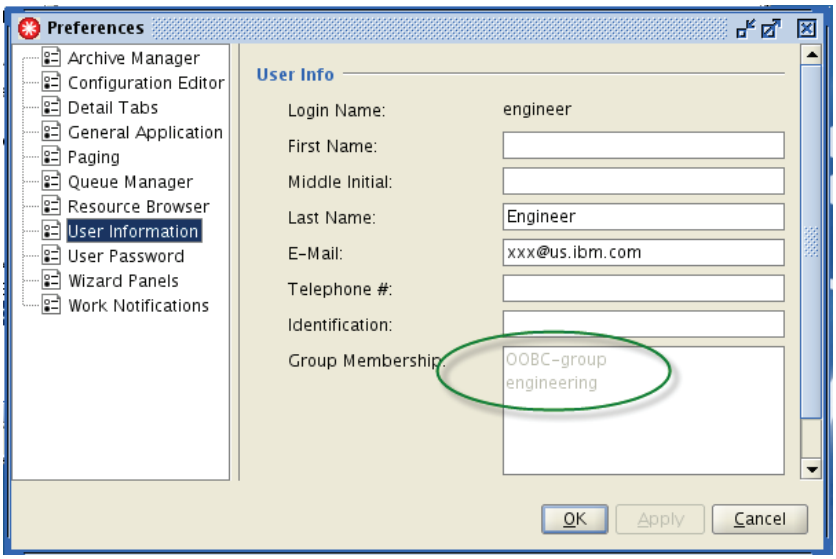
A user can be associated with more than one group. If that is the case, the user receives the union of the rights of each group. In most cases, this association means that the user gets more rights with the application.

The one area that is counterintuitive is that the user gets the more restrictive workflow policy. If one group needs approval and the other does not, a user who belongs to both groups must get approval.

If a user is associated with multiple groups, each with manage work activities, that user can approve two tickets, one for each of the groups that the user belongs to.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Discovering which groups you belong to



© Copyright IBM Corporation 2016

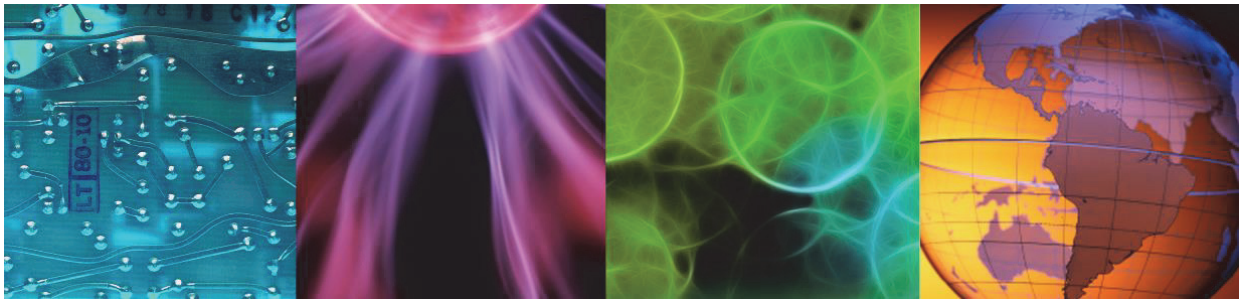
Discovering which groups you belong to

By accessing the User Preferences window, users can determine their group membership by looking at the **User Information** section.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 FIPS 140-2 security

Lesson 4 FIPS 140-2 security



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe FIPS 140-2 compliance
- Describe how Netcool Configuration Manager implements the standard
- Describe how it affects the client application

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

FIPS 140-2 compliance

Federal Information Processing Standard: NIST standard for federal government computer systems

Servers can be configured for FIPS 140-2 compliance

Force the use of FIPS certified cryptographic products: use of FIPS certified IBM Java runtime environment

© Copyright IBM Corporation 2016

FIPS 140-2 compliance

Certain customers require that applications meet requirements of the Federal Information Processing Standard (FIPS) 140-2. The National Institute of Standards and Technology developed the standard. The standard defines the security requirements for cryptographic modules. It defines levels of security for the cryptography that an application is going to use. It goes from a simple level of defining the algorithm that is used to encrypt data to more complex levels of security that determines whether cryptographic modules were tampered with.

You can configure the presentation servers in the Netcool Configuration Manager to meet Level 1 security for FIPS 140-2 compliance.

By choosing to configure the servers to use FIPS, you force the presentation server to use the certified FIPS security module included in the IBM Java runtime environment (JRE).

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

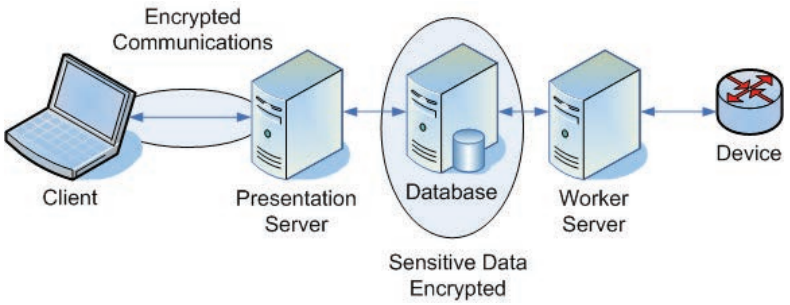
Scope of FIPS compliance

Areas where FIPS cipher is used

- HTTPS communication between clients and server
- Encryption of sensitive data, not all data in database

Areas where FIPS cipher is not used

- Communication between worker server and network devices
- Communication between servers and database



© Copyright IBM Corporation 2016

Scope of FIPS compliance

When FIPS is configured on the presentation server for Netcool Configuration Manager, it encrypts the following information:

- The communication between the client and presentation server
- Sensitive data that is contained in the database includes the passwords for authentication to the presentation server and to the network elements under management.

Attention: Not all data in the database is encrypted with the FIPS encryption modules. Most notable is the configuration that is retrieved from the device. The configuration is not encrypted in the database. If the passwords for local authentication are in clear text in the configuration, the database also stores them in clear text.

FIPS encryption algorithms are not in use between the presentation/worker servers and the database server. These communication channels can be encrypted but are not included in the currently supported FIPS compliance configuration. FIPS encryption modules are also not in use between worker server and network element communications. Most network elements do not have FIPS certified algorithms with their secure communication methods. Using secured communications with network elements is a function of the operating system on the device and beyond the capability of Netcool Configuration Manager to enforce.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Client access modes for FIPS

disabled
- Server does not guarantee the availability of a FIPS compliant cipher

warn
- Configures server to use IBMJCEFIPS for secure communication and data encryption.
- Permits IBM JRE clients without IBMJCEFIPS to make secure connections to the server.

strict
- Only IBM JRE clients configured with IBMJCEFIPS can make secure connections to the server

© Copyright IBM Corporation 2016

Client access modes for FIPS

The presentation servers can be configured to have one of three FIPS modes.

By default, the FIPS mode is *not enabled* for a presentation server. This mode does not ensure FIPS encryption even if the WebSphere server is using HTTPS connections.

If FIPS is configured on a server, you can select from *warn* or *strict* connections. When you configure for *warn* connections, the server is configured to use the IBMJCEFIPS plug-in for communications. However, if a client-side Java runtime environment is missing the IBMJCEFIPS, it still allows a secure connection to be made to the server. If you configure a *strict* connection protocol on the server, a client can connect to the presentation server only with the IBM Java environment with the IBMJCEFIPS plug-in.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring FIPS on server

Administrator enables or disables FIPS Mode on the NCM server

1. Enable configFIPSMode.sh for disabled|warn|strict
2. Edit TNCM Server JRE java.security file to use IBM cryptography module
3. Enable FIPS in WAS server console
4. Edit rseries.properties to use https
5. Restart TNCM servers to apply these changes

© Copyright IBM Corporation 2016

Configuring FIPS on server

Configuring FIPS on the presentation server is a fairly simple process. First, run the **$INSTALL_HOME$/bin/utils/configFIPSMode.sh** and set it for *disabled*, *warn*, or *strict*.

Next, configure the Java runtime environment the Netcool Configuration Manager presentation server is using. Set it to use the IBMJCEFIPS module by editing the **java.security** file in **$INSTALL_HOME/jre/lib/security/java.security**. Uncomment the provider and renumber the preference order so the IBMJCEFIPS is number one.

You then configure WebSphere to use the FIPS module. You need to access the WebSphere console and authenticate as the Netcool Configuration Manager superuser.

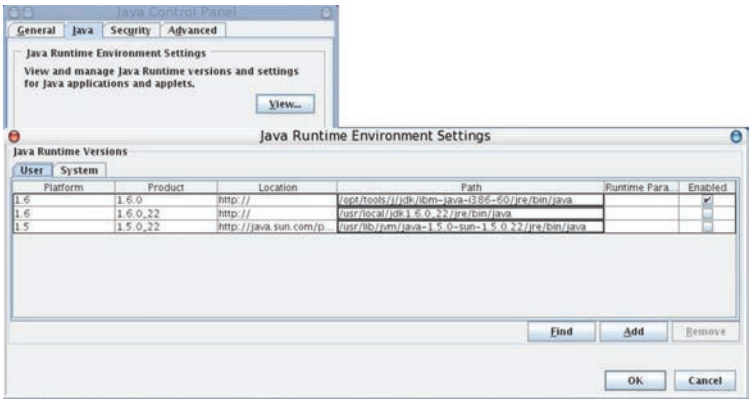
Details of this configuration process can be found at the IBM Knowledge Center website.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring FIPS on client side

Only IBM JRE is supported when FIPS enabled; FIPS Mode is Warn or Strict

You might have to configure client Java control panel to use correct JRE



© Copyright IBM Corporation 2016

Configuring FIPS on client side

You might need to configure the client to ensure that it uses the right Java runtime environment. If multiple Java instances are installed on the client computer, it is possible that the IBM JRE is not used for connections to the Netcool Configuration Manager presentation server. To enforce the use of the IBM JRE, you must go to the Java control panel and enable only the IBM JRE. If the presentation server is configured to use *strict* FIPS option, this additional step is required.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Create and verify groups and users
- Describe the group security that users are associated with
- Explain how realms control access to a network
- Assign security attributes and describe the authentication mechanisms to access the application
- Describe differences between domains and realms
- Implement view, add, modify, and delete rights throughout the application

© Copyright IBM Corporation 2016

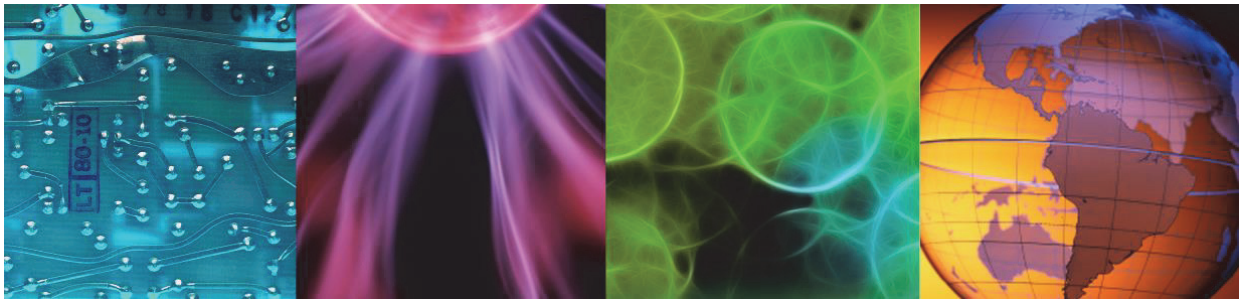
Summary

This unit provided lessons about the Netcool Configuration Manager security model. You learned how security is implemented, based on realms and groups. You also learned the authentication and authorization structures and how Tivoli Network Manager domains align with Netcool Configuration Manager realms. You were presented how *view*, *add*, *modify*, and *delete* capabilities are implemented in the application.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

8 Device management

Device management



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit provides information on modeling and how it can be extended to include security. It also introduces you to how Netcool Configuration Manager interacts with devices and how those interactions can be customized for a customer's network.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

- In this unit, you learn to perform the following tasks:
- Describe the advantages of modeling
- Explain how models support security
- Work with security sets
- Identify methods of device interaction with software
- Support customization with generalized resources
- Use an authentication resource
- Create a device script resource

© Copyright IBM Corporation 2016

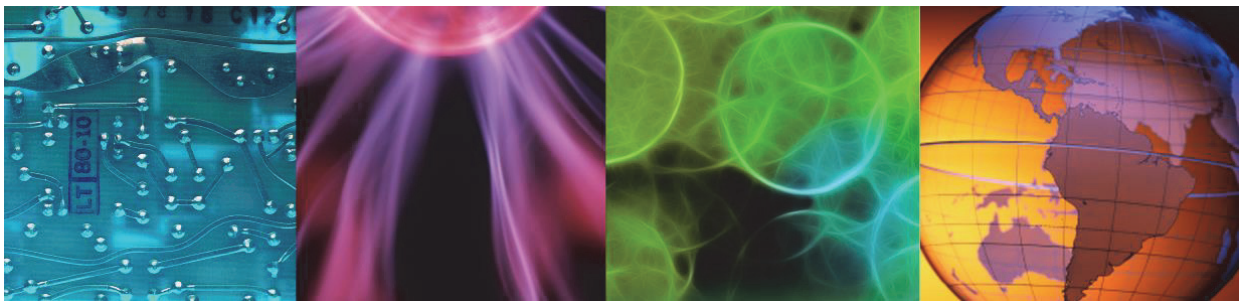
Objectives

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Modeling and drivers

Lesson 1 Modeling and drivers



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the advantages of modeling
- Describe how modeling normalizes device interaction
- Describe the components of a device model

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Advantages of modeling

Independence from unique vendor implementations

Capabilities that span a heterogeneous environment

Models can support generic IP concepts

© Copyright IBM Corporation 2016 4

Advantages of modeling

Modeling of device configurations provides some key differentiators to configuration management. With modeling, the application gains independence from the unique configuration management implementations that vendors select. Some vendors have different ways to configure their devices and communication protocols. Some devices have command-line interfaces, while others support only SNMP. Some apply changes directly to the configuration in memory, while others apply it to a draft configuration that is committed later. Each vendor selects different mechanisms for managing the hardware.

With modeling, you can abstract away from these unique implementations and create a standard procedure for creating and applying changes to these devices. An analogy that is useful is to think of the early days of computers and printing. Each vendor had their own print driver, which was installed and configured manually. Setting up a printer was not a trivial process. Now, operating systems have drivers that install themselves with little interaction with the user. This feature makes installing a printer less difficult.

Modeling provides a common interface for viewing a configuration and standard operations for adding, modifying, and deleting configuration data. This common interface extends to the tools that you use for managing configurations. All modeled devices can share tools, like command sets, search sets, and security sets that are based on modeling.

Modeling leads to more generic concepts that are vendor-neutral, such as border gateway protocol (BGP), virtual private networks (VPN), and quality of service (QoS).

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modeling meta-language: XML

Extensible markup language (XML) schema is used to translate between the device configuration and its modeled representation

XML is a fully tagged representation of actual configuration

- Searchable: Specific data can be found in documents
- Modifiable: Add, modify, delete **markup** is added to a document
- Extensible: Security sets are used

XML schema is specific to vendor, type, model, and operating system (VTMOS)

© Copyright IBM Corporation 2016

Modeling meta-language: XML

Modeling uses XML technologies to take the native configuration data of a device and convert it into a fully tagged XML representation. The data is contained within the device or its mechanism of retrieval, for example, SNMP, CLI, or TL-1. The neutral format for presentation and manipulation is the only consideration.

This XML representation of configuration data then becomes easy to search, modify, and extend through a series of external tools. These tools are industry-standard utilities that are unaware of the vendor specifics that underlay the configuration data that it analyzes or changes.

The challenge of this modeling is that the XML schema development is often specific to vendor, model, and operating system. The XML schema represents command structure. This command structure is the variable that changes from device to device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Vendor syntax

Configuration syntax variations exist, even within vendors

- Cisco: CatOS, IOS
- Juniper: M-Series, E-Series, Netscreen-Series

Example: Using vendor syntax to configure a banner

```
set system login message "Login Message - Juniper M-Series”
```

```
banner login ^CLogin Message - Cisco 7200 Router^C
```

```
set banner motd ^CLogin Message - Cisco 6509 CatOS Switch^C
```

```
configure banner
Login Message - Extreme Summit Switch
```

© Copyright IBM Corporation 2016

Vendor syntax

In this example, you have four different devices from three different vendors. You are configuring a simple banner to show when the user attempts to access the device. Notice the variation between each. Some vendors use an initial verb to define their commands, like **set** or **configure**, while other vendors choose a noun, like **banner**. From there, each command syntax is different and unique to those devices. Even within the same vendor this variation occurs. Cisco offers devices that use two distinctly different operating systems that use different configuration languages.

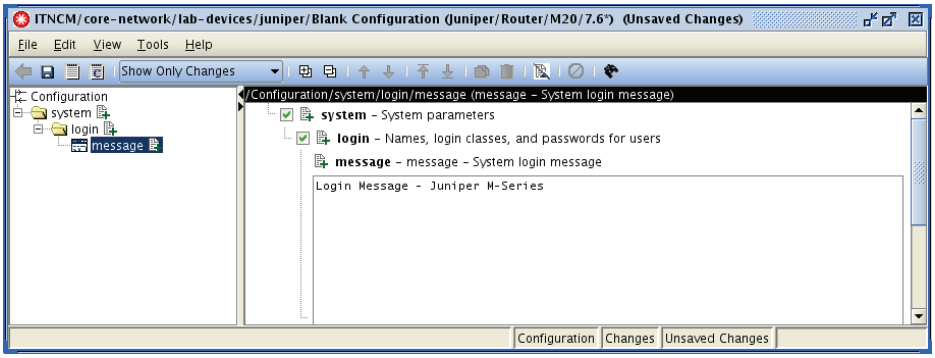
On the next few slides, you see how to use modeling to alleviate vendor syntactical differences. The user is provided a simple field to complete with banner information.

The device syntax is not eliminated from the configuration change. However, a user can manage the differences between vendors with less difficulty.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Setting the banner on Juniper M-series router




```
set system login message "Login Message - Juniper M-Series"
```

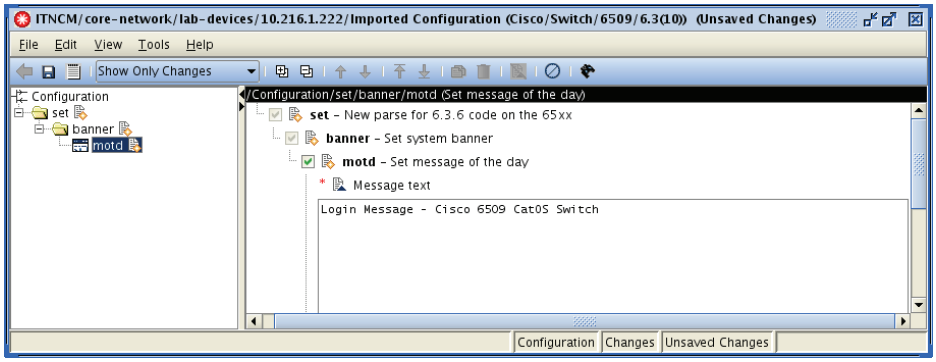
© Copyright IBM Corporation 2016

Setting the banner on Juniper M-series router

This slide illustrates the syntax to set a banner on a Juniper (www.juniper.net) M-series device. This device requires you enter a configuration mode, enter the command, and then *commit* it to the configuration in memory.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Setting the banner on Cisco CatOS switch




```
set banner motd ^CLogin Message - Cisco 6509 CatOS Switch^C
```

© Copyright IBM Corporation 2016

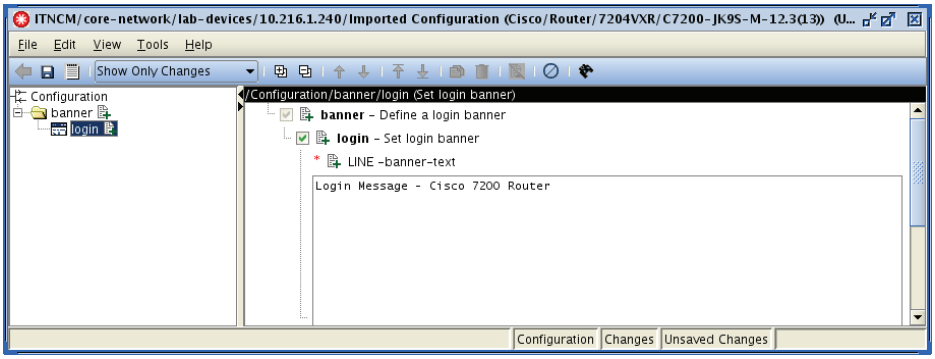
Setting the banner on Cisco CatOS switch

This slide illustrates the syntax to set a banner on a Cisco (www.cisco.com) CatOS device. You can enter the command directly to the device without entering any special model and it is immediately applied to the configuration. This device has only one configuration to apply changes to.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Setting the banner on Cisco IOS router




```
banner login ^CLogin Message - Cisco 7200 Router^C
```

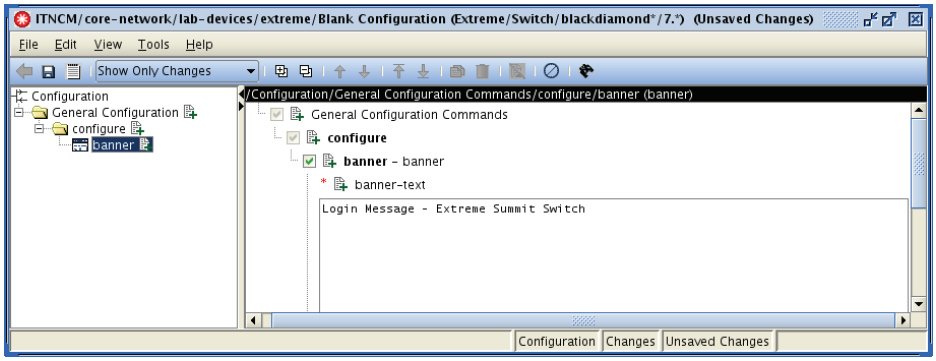
© Copyright IBM Corporation 2016

Setting the banner on Cisco IOS router

This slide illustrates the syntax to set a banner on a Cisco (www.cisco.com) IOS device. Unlike the CatOS switch, this device requires you to enter a configuration mode to enter the commands. They are immediately applied to the configuration in memory. After applying the change, you must save it to a startup configuration that is saved on disk. Saving the file allows the change to survive a rebooting of the device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Setting the banner on Extreme Summit switch




```
configure banner
Login Message - Extreme Summit Switch
```

© Copyright IBM Corporation 2016

Setting the banner on Extreme Summit switch

This slide illustrates the banner applied to an Extreme (www.extremenetworks.com) switch. Like a Cisco CatOS device, you do not have to enter a configuration mode to apply this change. However, you must save the change to a primary configuration file for it to survive a device restart.

The following items illustrate the important points from the previous slides:

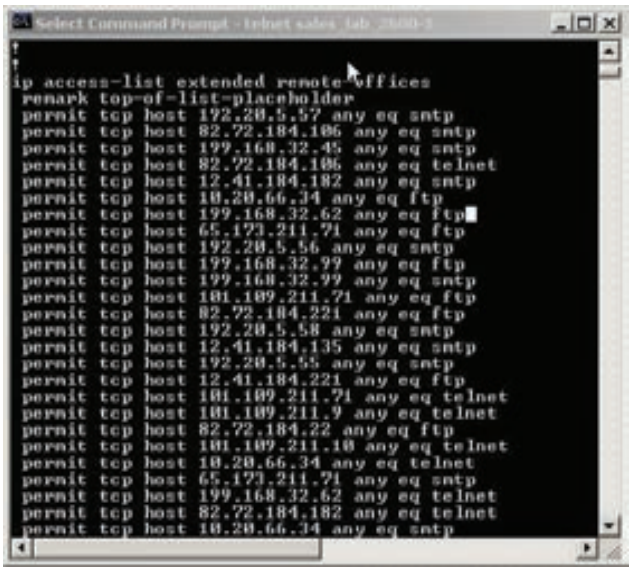
1. The syntax is unique in each example but the configuration editor GUI makes interacting with the syntax similar to the concepts of finding the command and entering data into an argument field

2. When applying the change to the device, you do not need to know the details of the application process as that is contained within the model itself.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Converting raw CLI to XML



© Copyright IBM Corporation 2016 11

Converting raw CLI to XML

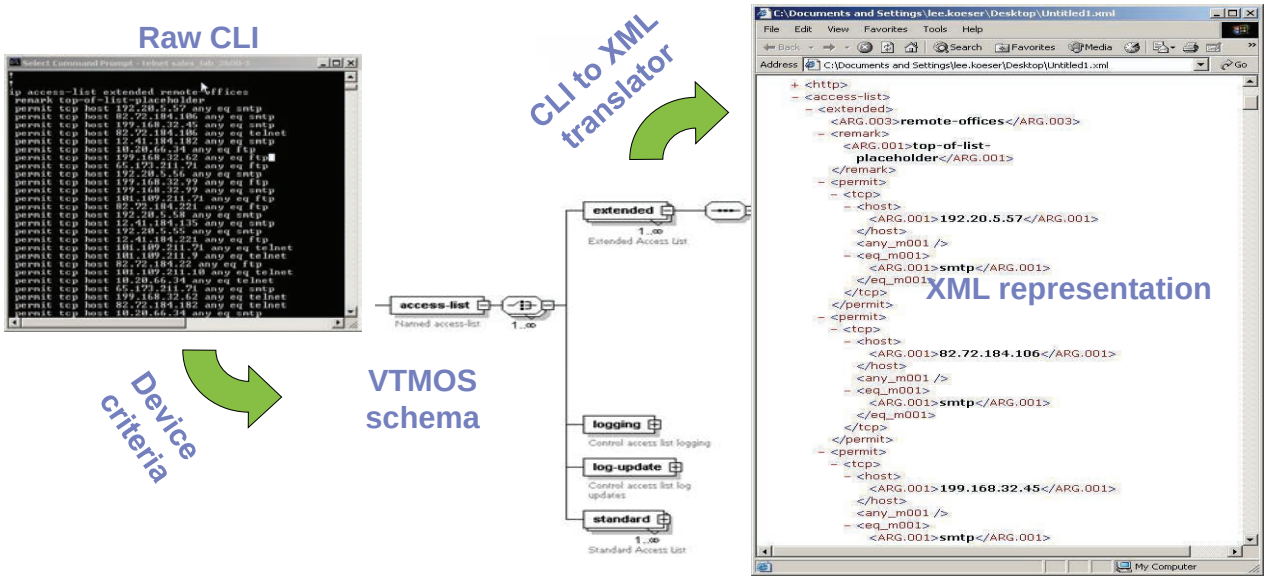
In Netcool Configuration Manager, modeling converts a device's native configuration language into XML. Most modeling converts a CLI configuration, that is, a text-based configuration file retrievable from a device, into an XML representation. However, you can convert configurations that are stored and retrievable through SNMP or TL-1 commands. Regardless of how it retrieves the data, modeling takes this text-based information and converts it into XML.

In the slide example is a named access control list from a Cisco IOS router. Notice the permit statements that authorize various IP addresses and associated protocols to access the network.

Note: TL-1 is a communication protocol that certain optical switches use. Retrieving configuration data with TL-1 requires a series of interactive commands. The configuration is retrieved one section at a time with these commands. One command is sent to the device and the response to that command requires more commands. For example, it uses one command to retrieve interface information. The results are shown, and then more commands must be sent to gather information from each individual interface.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using VTMOS schema to convert to XML



© Copyright IBM Corporation 2016

Using VTMOS schema to convert to XML

With modeling, the text-based configuration, regardless of how it is retrieved, is analyzed by using a device-specific schema. This device-specific schema is directly tied to the vendor, model, and operating system data that is part of the device itself. This VTMOS-specific schema takes the text-based data and decomposes it into specific elements through a **CLI-to-XML** translator. This decomposition tags each aspect of the command structure to create an XML representation of the configuration.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modeled device types

Command-line interfaces configuration

- Configuration is in ASCII text
- Entire configuration is retrieved with simple commands
- Includes most current SmartModel drivers

SNMP configuration only

- Configurable only through SNMP **set** and **get** commands

TL-1 devices

- Configuration is retrieved through call and response methodology
- Results of one command create a series of secondary and tertiary commands
- You can manage a small portion of the actual configuration

Proprietary interfaces

© Copyright IBM Corporation 2016 13

Modeled device types

You can model many types of network devices. Some devices do not qualify as network elements, for example, smart meters the electrical power distribution industry uses.

Most modeling is done on devices that have a command-line interface that is accessed through a VTY terminal. It stores the configuration as an ASCII text file and can retrieve it with simple commands. Most network devices have this interaction available to users.

Some devices require the use of SNMP to apply changes. The model of an SNMP management information base (MIB) is available to capture and change configuration data through **GET** and **SET** commands.

Other modeled devices use a TL-1 interface. You retrieve configuration data through a series of commands that you develop as information about the device is retrieved. Retrieving configuration data through TL-1 is a tedious task, and only a portion of the configuration is retrieved. In this case, the part of the configuration the customer wants to modify regularly is modeled. This method requires other traditional methods for saving and restoring the entire configuration in a disaster.

You can access some devices only through proprietary interfaces and direct access is impossible. Here interactions are modeled with the element management system (EMS) that configures the device and Netcool Configuration Manager works with the EMS to gather configuration data and apply changes.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using drivers to model devices

Use drivers to interface with devices

- Contain connection mechanisms
- Retrieve configurations and other data
- Submit and confirm changes
- Perform other management functions

Drivers contain these items:

- Model scripts, and sometimes OS scripts, that are specific for device interaction
- Schemas for command structure in SmartModels
- Translators to convert from vendor-specific language to XML in SmartModels
- Logic to convert XML changes back into vendor-specific language in SmartModels

© Copyright IBM Corporation 2016 14

Using drivers to model devices

You use a driver to manage a particular device. The driver contains everything necessary to interface with the configuration mode of the device. It contains the methods necessary to connect to the device, retrieve configuration information, and obtain hardware specifics. You can use the driver to change the device and for other management functions, such as upgrading its operating system.

A driver consists of both an XML schema of the device's command structure and translators that convert the native configuration data into XML and vice versa. It also has the scripts that it needs to communicate with the devices over the chosen protocol, typically a VTY session. A driver is specific to a vendor, type, model, and often times, an operating system.

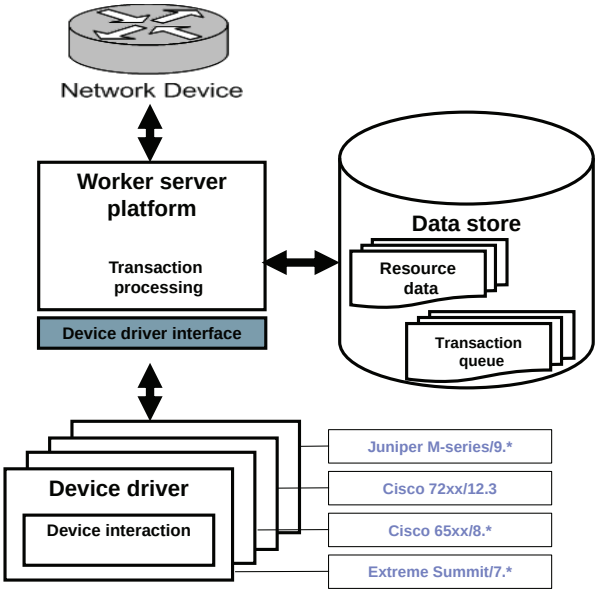
Important: A driver is either a standard driver or a SmartModel driver. For the purposes of most of this training, the underlying assumption is that the driver is using SmartModel technology. The key difference is that a standard model captures only CLI data from a device as text. SmartModels take that data and convert it into XML. Most Netcool Configuration Management tools are based on SmartModels.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Device driver architecture

Device driver architecture
- Code and metadata to support a set of devices
- Independent of application and other device drivers
- Well-defined, external, and extensible interface

Benefits
- Decoupled from product releases
- Live update of drivers on running system
- Isolated driver code improves reliability
- Optimize drivers based on unique device characteristics
- Enabler for future device driver development by external vendors



© Copyright IBM Corporation 2016

Device driver architecture

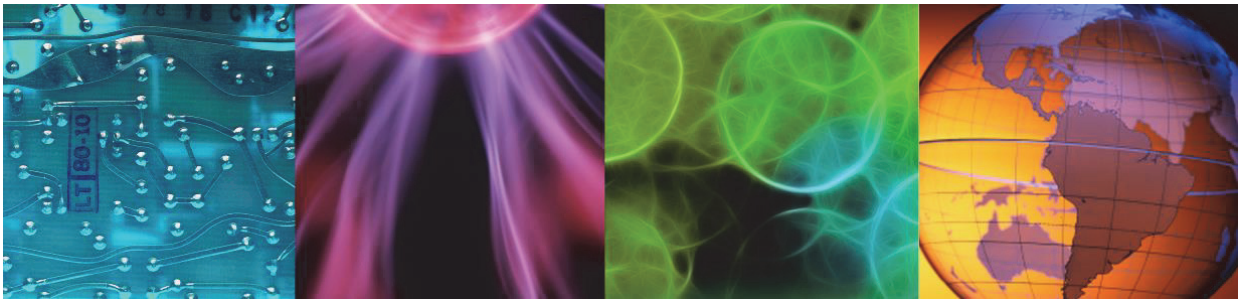
The device driver architecture, regardless if it is SmartModel or standard, is based on a series of flat files that exist on each of the servers. The flat files contain information that defines a driver, such as scripts, schema, and translators. Each set of flat files is designed to support a specific vendor, model, and operating system. Drivers are independent of the other drivers and are usually independent of the application revision level. A driver has a well-defined interface that connects to the worker server, which, in-turn, interacts with a physical device.

Device drivers are developed and released separately from the application. Drivers are optimized based on unique device characteristics and protocols. For example, customers can choose to model a small portion of a device's configuration to support the automation of a particular service. The interface between the worker and the driver is documented. External vendors or Business Partners use this documentation to develop their own drivers, for example, a federal customer with a proprietary device with a classified interface.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Security sets

Lesson 2 Security sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe a security set and its capabilities
- Build and implement a security set

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining security at a command level

Security sets add another layer of security to an application

Security sets extend a SmartModel to include view, add, modify, and delete rights to commands

Security sets add fine-grain configuration control

© Copyright IBM Corporation 2016 17

Defining security at a command level

The security set is a tool that layers security on top of modeled configurations. This concept is addressed here because it extends the XML model. It incorporates technology that is not necessary for managing the device, but supports security at the presentation layer.

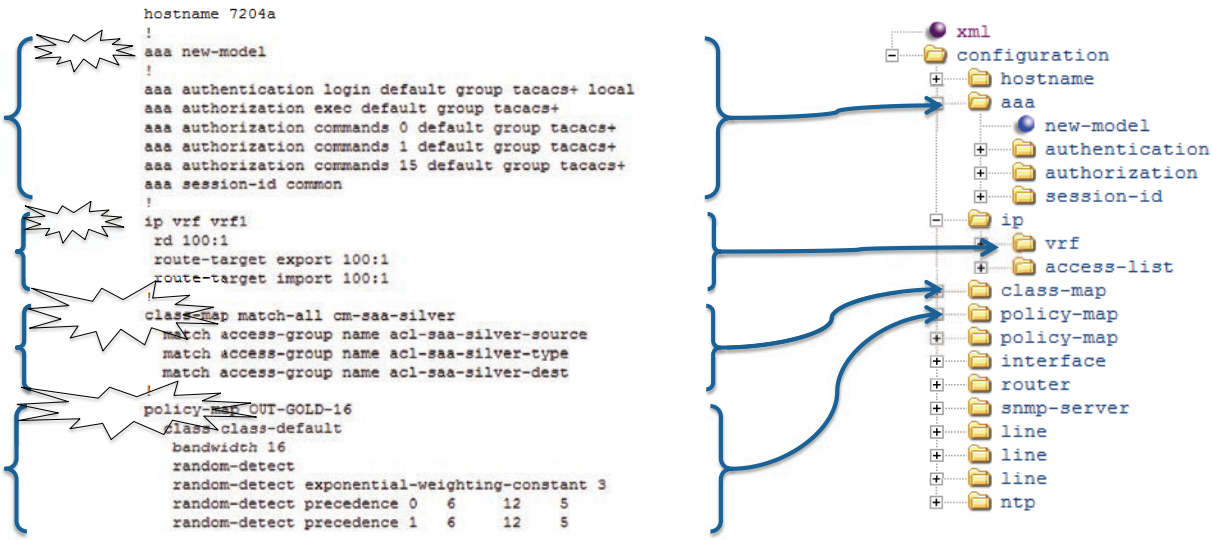
Security sets add permissions to the commands inside a configuration. It defines *view*, *add*, *modify*, and *delete* (VAMD) rights on various parts of the configuration. In the GUI, this information is either not available or not shown.

Security sets are applicable to any SmartModel. This feature is a function that, when you create it for one SmartModel, you can use it for any SmartModel.

Note: Organizations rarely have business policies that are detailed to this level of granularity. While this feature generates a large amount of interest, it is hard to implement because of organizational issues.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Mapping a native configuration to XML hierarchy



© Copyright IBM Corporation 2016 18

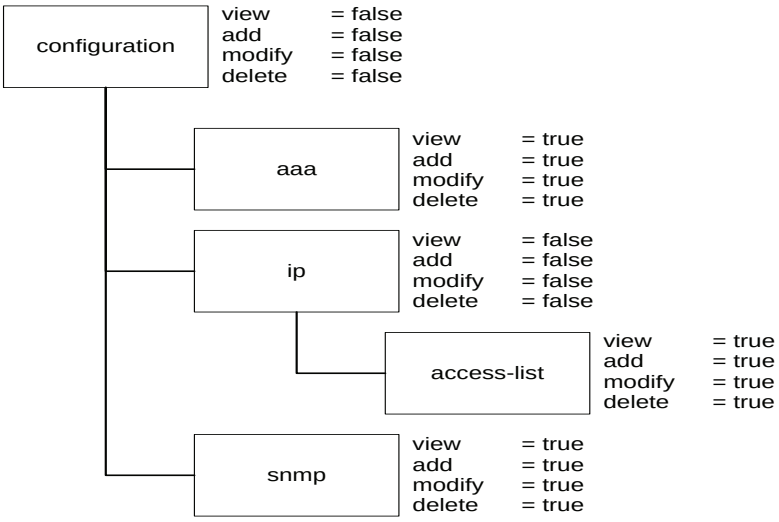
Mapping a native configuration to XML hierarchy

The decomposition of configuration data from the native form to XML creates a hierarchy of data. In the example here, it converts data from a Cisco format to XML format. The data is organized into parent-child relationships, which are based on the indentation that is used by Cisco. Commands that are indented are subcommands to a particular parent. Using this process, you can add VAMD rights.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Layering security sets onto a hierarchy

- Use XML configuration format for customization
- Overlay metadata onto XML structure
- Define VAMD rights on configuration nodes
- VAMD are inherited down the configuration



© Copyright IBM Corporation 2016

Layering security sets onto a hierarchy

Using the XML hierarchy for the configuration, the security set adds VAMD rights to the configuration and its subsequent nodes. In this example, the base node of the XML configuration is <configuration>. VAMD rights are all set to FALSE. If it is left this way and the security set is applied, the user sees nothing upon opening the configuration.

However, in subsequent nodes down the tree, the VAMD rights are modified. For example, the <aaa> node of the configuration has all its VAMD rights set to TRUE. If applied, the users see the AAA node of the configuration. The same is true for all the SNMP commands.

VAMD rights are inherited down the configuration tree. For all other commands not shown in this example, their VAMD rights are all FALSE.

In the <ip> node of the configuration, the security set is designed so that users can see and modify any access lists under the <ip> node of the tree. However, anything else under <ip> is hidden.

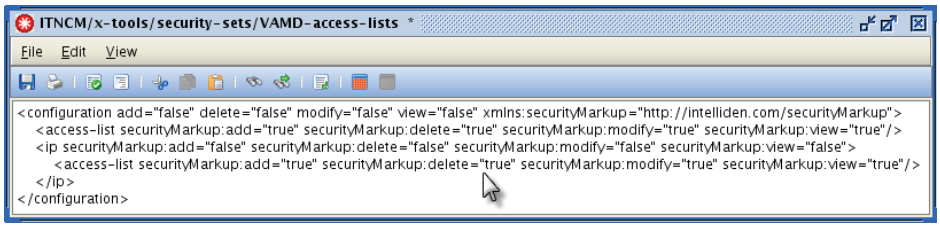
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating security sets

Security sets are resources created in the GUI

Security sets are schema-specific, but you can use them across devices with the same vendor and type

A security set must be in a realm where devices can resolve it



© Copyright IBM Corporation 2016

Creating security sets

Users create security sets in the GUI by selecting **File > New > Security Set**. You must place the security set in a realm where devices resolve to it like other generalized resources. Placing the security set in the realm does not initially affect users. Group security applies the security sets to the configurations, when users in that group open the configuration editor.

The security set is one of the last remaining resources that forces the user to edit a raw XML document. Users can use an external XML editor to create the document and cut and paste it into the editor that the GUI uses.

The security document is overlaid on a configuration, which means its node structure must align with the node structure of the configuration. Security sets are vendor-specific. You can often use them across devices with the same vendor and type. However, if the nodes do not align, the security sets have no effect.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Formatting security sets

```
<configuration
   add="false"
   delete="false"
   modify="false"
   view="false"
   xmlns:securityMarkup="http://intelliden.com/securityMarkup">
  <access-list
   securityMarkup:add="true"
   securityMarkup:delete="true"
   securityMarkup:modify="true"
   securityMarkup:view="true"/>
  <ip >
    <access-list
          securityMarkup:add="true"
          securityMarkup:delete="true"
          securityMarkup:modify="true"
          securityMarkup:view="true"/>
  </ip>
</configuration>
```

Required base node with VAMD settings (note closing tag at the bottom)

Open access to access-list node in configuration (note how this node is closed with /> at end)

The <ip> node inherits permissions from <configuration>. You open the access-list node by adding VAMD rights and setting them to true. (Note closing </ip> tag at bottom)

© Copyright IBM Corporation 2016

Formatting security sets

On this slide, you see an example of a security set. This security set is designed to hide everything about a configuration except its access control lists (ACLs). On a Cisco device, ACLs can exist in two locations:

- As a series of base commands in the configuration
- As a series of commands under the <ip> node of the configuration

In this security set, you are disabling all VAMD rights at the base node, which is known as <configuration>. For each right, the value is FALSE. You use the base command <access-list> to set the values for *add*, *delete*, *modify*, and *view* to TRUE. Notice the format differences between the base node and the subsequent nodes in the tree. The attribute is called securityMarkup:<type>.

The <ip> node is different. You want to prevent the user from accessing a number of commands under the <ip> subtree. The <ip> node inherits its parent's VAMD setting when you do not define them. In this case, <ip> has the same rights as <configuration>. The node <access-list> is a child of <ip> on a Cisco device. You open this node by setting all the VAMD rights to TRUE.

The closing tags at the bottom of the slide are not associated with security sets, but are a by-product of an XML document.

The following example illustrates the CLI that defines the two types of access lists that you can configure on a Cisco router:

```
...
access-list 100 remark generated by SDM firewall configuration
access-list 100 remark SDM_ACL Category=1
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

```
access-list 100 permit tcp any host 1.2.34.3 eq www
access-list 100 deny ip 1.2.3.0 0.0.0.255 any
access-list 100 deny ip host 255.255.255.255 any
access-list 100 deny ip 127.0.0.0 0.255.255.255 any
access-list 100 permit ip any any
...
ip access-list extended acl_guest_in
deny ip any 18.18.18.0 0.0.0.255
deny icmp any 18.18.18.0 0.0.0.255
deny ip any 118.118.118.0 0.0.0.255
deny icmp any 118.118.118.0 0.0.0.255
permit ip any any
```

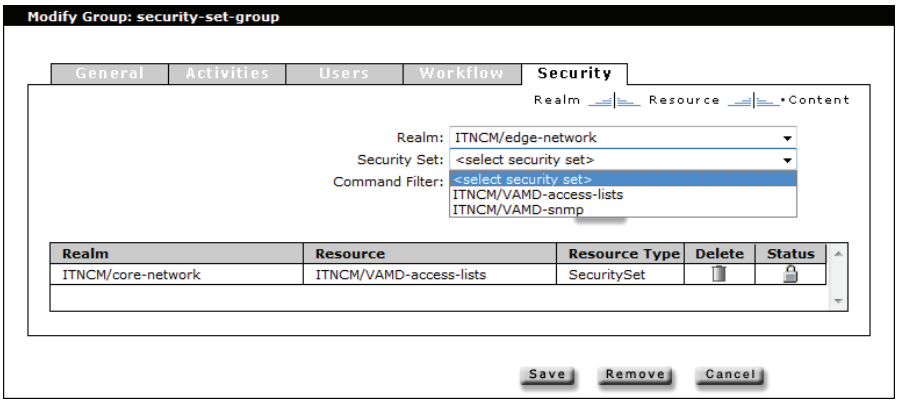
© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Applying a security set

Apply to a group and the realm pairs by using the Accounts tab

Use only one security set per group or realm pair



© Copyright IBM Corporation 2016 22

Applying a security set

After a security set is created, you must enable it so that you can use it to control configuration security. Security sets are applied to groups and realms through the account manager. In this example, you use the Content portion of the **Security** tab. You select a security set for each realm. In the example, this group has a security set placed on the realm **ITNCM/core-network**. This security set contains the access-list example shown in the previous slides.

The security set is initially created to have a VTMOS of Cisco / Router / * / *. This security set affects all Cisco routers in or under the core-network realm. When users of the security set group look at these configurations, the security set limits their view.

Currently, one security set per realm is supported. If more than one security set is applied to a device, the user receives a warning about having an empty configuration.

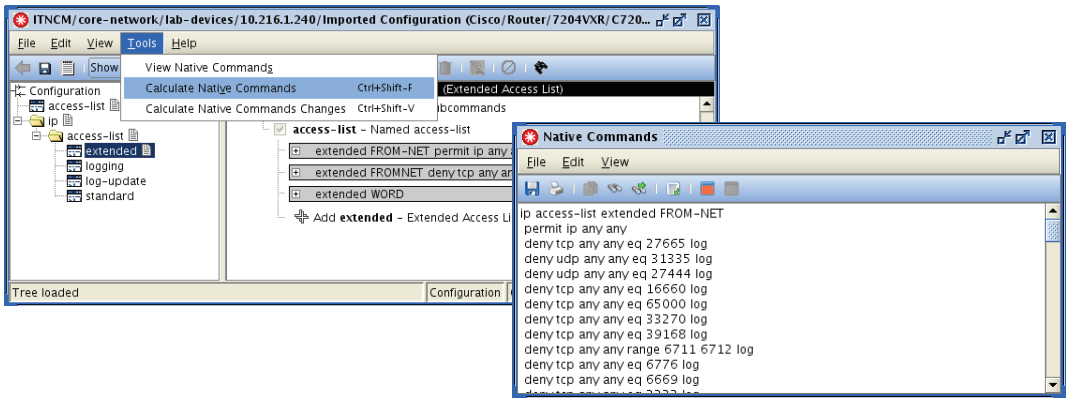
Note: A status column on the page contains an icon of a padlock. Be sure to save the group settings after adding a security set. The padlock must be in the locked position, or the security set is not saved to the group.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Results of applying a security set

You apply the security set to the configuration on the server side

You can click **Tools** > **Calculate Native Commands** to see only viewable data



© Copyright IBM Corporation 2016 23

Results of applying a security set

After the security set is applied and users in the affected group open a configuration that resolves to it, their capabilities are limited. The security set is applied on the presentation server and whatever is not viewable is not sent to the client.

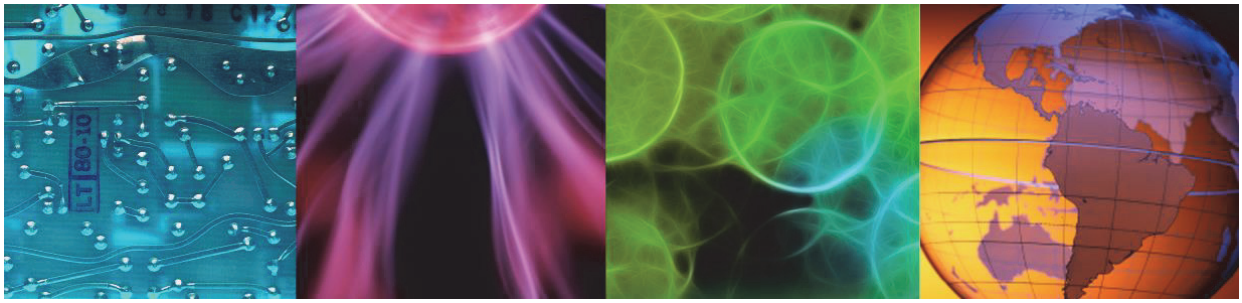
The security set also affects the *calculate native command* function that converts XML into CLI. The user sees only the commands that the security set applies.

An important aspect of security is making sure that you disable the activity **View Native Commands** for a group that has a security set applied. If you do not remove this activity, the user is able to see the entire configuration by making that menu selection.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Device interaction

Lesson 3 Device interaction



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe how Netcool Configuration Manager interacts with devices
- Describe the challenges of automating the management of network devices

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Overview

Most device interactions use Telnet or SSH connections

- SNMP is available when VTY configuration support is not available

Most devices include these default behaviors:

- Retrieves data off device with streaming
- Places data on device with file transfers
- Retrieving configuration at beginning of change task
- Saving in-memory configuration to disk after change is made
- Retrieving configuration at the end of change task

Some changes require interactive commands to confirm change

- Example: Deleting a VLAN on a CatOS switch
- Commands tracked in the logs but not seen in the GUI

© Copyright IBM Corporation 2016

Overview

For Netcool Configuration Manager to automate the management of network devices, it needs to connect to and interact with devices. Most interactions occur over a Telnet or SSH connection with a device. In cases where you cannot make configuration changes through a VTY session, use SNMP with appropriate SET and GET statements.

Device interaction to support configuration management includes these default behaviors:

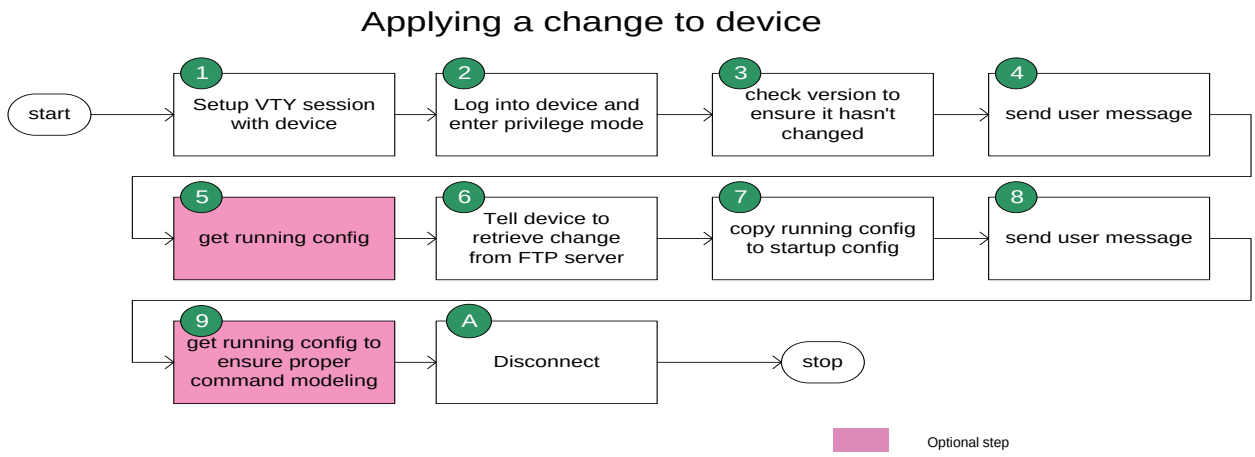
- Retrieving data from a device by streaming it over the VTY connection as ASCII text
- Placing data on the device by using a file transfer protocol
- Capturing the current configuration at the beginning of a change task to ensure that you can restore the device when a problem arises
- Saving the changed configuration in case the device is rebooted
- Retrieving the configuration from the device after a change is made to it

Some device interactions require interactive responses from the user. For example, when you delete a VLAN from a Cisco switch, you receive a message and must confirm that you want to perform this action. The normal scripts for device interaction do not contain the responses to these interactive commands, but they are part of the device schema. You do not see these interactive command responses in the CLI that is generated from marked-up XML documents. The only way to see these commands is to review the detailed driver logs that record the device interaction.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Device interaction example



© Copyright IBM Corporation 2016

Device interaction example

In the diagram on the slide, you see an interaction between the worker server and a Cisco router. This interaction is applying a change to a configuration, by using the steps in this workflow:

1. The worker server connects to the IP or host name that the device is named inside the application. It uses either a Telnet or an SSH connection based on the preferences configured by the system administrator.

2. After it connects to the IP address, the worker uses a script to access the device. The script looks for keywords like *user name* and *password* and responds to those keywords with the credentials defined in the application. After accessing the device, the script verifies that it is in the Cisco privilege mode. Extra commands ensure that the terminal session scrolls all data down the window without user interaction, for example, **terminal length 0**.

3. The script submits commands to the device to determine the device model and version information. The output of these commands, such as **show version**, is parsed for these two data items. It then compares these two data items with the database to verify that the device is still the same model and operating system.

4. A script sends a message to all logged-in users to ensure that they are aware that the worker server is processing the device.

5. A script retrieves the running configuration from the device. This retrieval either uses a file transfer protocol or streams over the VTY connection.

6. A script changes the device. The default behavior sends a command to the device to tell it to retrieve the change from an FTP server.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

7. A script copies the newly changed configuration to the startup configuration on the device.
8. Another message is sent to users to notify them that the worker retrieves the configuration again.
9. A script, the same as used in step 5, retrieves the newly changed running configuration from the device. This retrieval either uses a file transfer protocol or streams over the VTY connection.
10. An optional script explicitly disconnects from the device.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Challenges to device interaction

- Devices are not meant for automation
  - Worker servers are too fast for device interface
- Support infrastructure is not meant for automation
  - Authentication servers cannot handle worker server requests
- Vendors are inconsistent with their implementations
  - Data format changes from release to release
- Network design is challenging
  - Device access not straightforward
- Customers have unique corporate policies
  - No file transfers are allowed outside the subnet

© Copyright IBM Corporation 2016 27

Challenges to device interaction

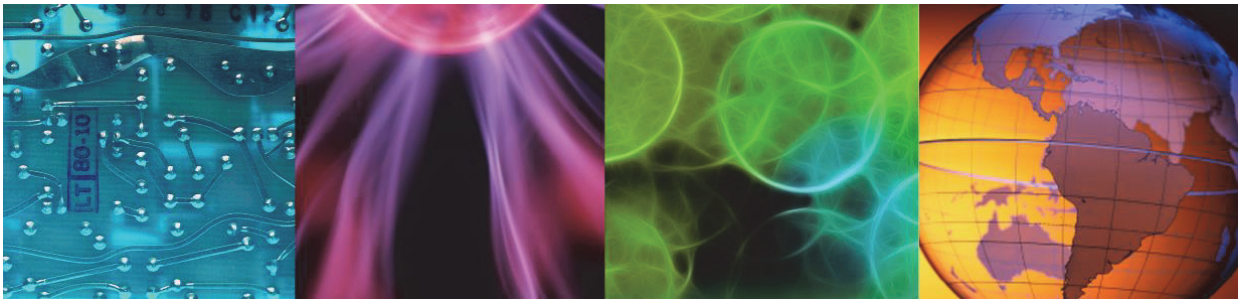
No multivendor standard exists for retrieving configuration data. A common expectation of vendors is that the person is interacting with the device through a VTY session. Another assumption is that this user is trained in the interaction process that the device requires. These assumptions lead to some of the following challenges:

- Devices are old or not meant for automation, and the device buffer overflows. The worker server sends data too fast to the device, and it drops data. The issue might also be the connection bandwidth between the worker server and device. Some connection methods, with limited bandwidth, are slow and cause timeouts on the worker server.
- The infrastructure for managing devices cannot handle automation in the external authentication servers and possibly in the file transfer servers. With multiple worker servers, with multiple threads, 80 or more requests for device authentication can exist at the same time.
- With Netcool Configuration Manager, you can parse data off devices to verify that you have a clear picture of the model and operating system of the device. Parsing requires search criteria to find this data. However, vendors might change the format of their device output, causing the search criteria to either pull too much or not enough data.
- Customers can create network designs that are challenging to automate. What is designed for manual interaction can make it difficult for worker servers to reach the devices in the network.
- Corporate policies can prevent the use of various protocols in the network. For example, certain government agencies do not authorize file transfers within their network. Making sure that the change management application respects these policies adds complexity to automation.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 Customizing device interaction

Lesson 4 Customizing device interaction



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe how a device resolves customization documents
- Describe the various documents that are used to customize the application

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Customizing the solution for a customer network

Customization determines how a series of questions are answered for each individual device under management

Customization typically occurs at two levels

- How to connect to a device
- How to interact with it after it is connected

Customizing connection

- What connection method to use, Telnet or SSH
- What passwords to use to authenticate to the device

Customizing interaction

- Stream or FTP
- Custom commands or parsing

© Copyright IBM Corporation 2016

Customizing the solution for a customer network

You can customize the solution to operate in a unique network. The intent is to minimize the effect on the network design. Customization refers to the ways that this product interacts with the physical devices it is managing.

Customization can be the way that Netcool Configuration Manager connects to a physical device or the way that it interacts after it connects. Customizing connection methods include communication protocols that access the device and the credentials that it requires to log in. It might also include deciding which worker server can access the device. Customizing the interaction after connection includes file transfer protocols and the specific interaction scripts that it needs for device interaction.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Customization at a realm level

Realms often determine divisions inside an organization

- Example: Core versus edge network groups

Different groups often have different means for accessing devices

- Different credentials
- Different routes for device access
- Different protocols for connecting

Each major realm can have different methods

Realm-level generalized resources allow customization

© Copyright IBM Corporation 2016

Customization at a realm level

Most customization occurs at the realm level so that the customization is shared across all the devices in or underneath that realm. Because realms often times reflect organizational or customer management structures, you can customize passwords for each group or customer.

An example is a customer with a group who manages the core network and another that manages the customers who make up the edge network. These two groups designed the network so that management applications must use separate VLANs to access devices. Also, each group has separate credentials and connection protocols. However, this product is designed to support this network implementation. Using realm-level generalized resources, the core realm can have different properties from the edge realm.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Realm-level generalized resources

What credentials to use to log in to a device
- Authentication resource

What FTP server to use to move data
- File transfer resource

Where it processes the work
- Worker server resource

What communication method and scripts to use
- Access resource
- Script resource

Name	Resource Type	Vendor	Type	Model	OS
authentication-resource	Authentication	*	*	*	*
device-script	Device Script	Cisco	Router	26*	*
file-transfer-resource	File Transfer	*	*	*	*
resource-access-document	Resource Access	*	*	*	*
worker-server-resource	Work Server	*	*	*	*

© Copyright IBM Corporation 2016

Realm-level generalized resources

Realm-level generalized resources ensure that each realm can answer questions that a unit of work needs to work with a device. Each device sends the following questions when it requires Netcool Configuration Manager to interact with it.

- What user name and password is required to connect to the device? An authentication resource is usually the answer to this question.
- What FTP server do you tell the device to use for file transfer purposes? A file transfer resource is the answer to this question.
- What worker servers should you use to connect to the device with? A worker server resource is always the answer to this question.
- What protocol and port do you use to connect to the device with and how do you communicate with it? An access resource and a scripting resource are the answers to these questions.

Other than the worker server resource, default answers for these questions are contained either in the database or in the device driver software.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Generalized resources that are specific to VTMOS

All devices often use generalized resources

- Authentication resource */*/*/*
- Worker server resource */*/*/*
- File transfer resource */*/*/*

Sometimes resources must be specific to VTMOS

- Access resource Cisco/Router/*/*
- Device script resource Cisco/Router/3560/*

© Copyright IBM Corporation 2016

Generalized resources that are specific to VTMOS

Generalized resources are often associated with a specific vendor, type, model, and operating system. Sometimes they are not. Some resources are generic, and all devices can use them. Some resources provide credentials, file transfer services, and locations where the UOW can process. Typically, this data is independent of the device that is processed. For example, regardless of the VTMOS, the user name and password to access the device is the same. However, these resources can be specific to VTMOS, if necessary.

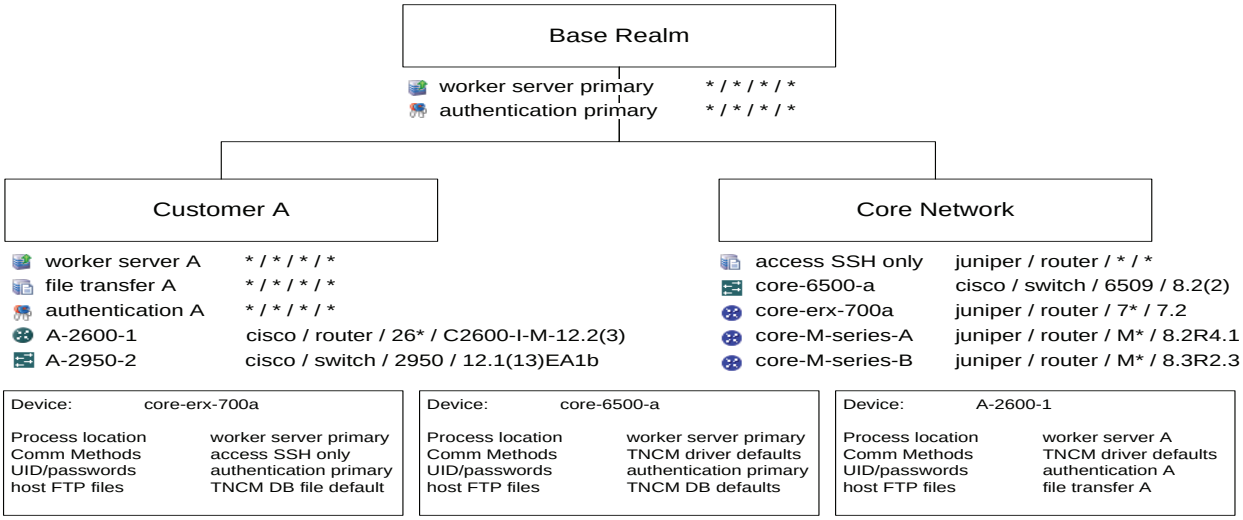
Some resources must be specific to VTMOS. Most notably is the script resource. The script resource has the communication scripts that you use to interact with the device. A Cisco 3560 device uses a different command structure than a Juniper M20. Using inappropriate scripts on devices cause *units of work* to fail, and more importantly, might affect the device state.

Note: The *type* in VTMOS is a construct that was developed early in the development of the product software. In 2001, device hardware was segmented in function. A device was either a router or a switch. As technology improved and competitive pressures increased, vendors created devices that are both a router and a multiport switch, and even a firewall. Type is irrelevant in the current hardware market. However, *type* is a distinct filter mechanism that this product uses and cannot be easily removed. A device must have a *type* for the product to function correctly.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Generalized resources resolution example



© Copyright IBM Corporation 2016 33

Generalized resources resolution example

Most generalized resources use a hierarchical resolution scheme that is based on these items:

- Where a device is located in the realm structure
- Where the generalized resource is located in the realm structure
- What the VTMOS is for both resources.

In the example on the slide, you see a realm structure where you have a service provider with two realms. One reflects their own core network, and the other is the customer whose network they provide services for. The following questions are asked whenever a change is applied to a device:

- What password is used to connect?
- Where are file transfers hosted?
- Which worker server can process this UOW?
- What connection protocol does it use, Telnet or SSH?

When a UOW processes a task, it seeks answers to these questions. These questions are answered in 1 of two ways: through a generalized resource or the defaults that a database or driver contains. For each question, the task does an upward search from its realm location and looks for the appropriate resource. If one is found, the VTMOS of the device and the generalized resources must align to use it. If none is found in the upward search, it uses the defaults.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

For example, the device core-erx-700a has a VTMOS of Juniper / Router / 7* / 7.2. To answer the previous questions, you find these answers:

- To determine where this UOW is processed, an upward search from the Core Network realm for a worker server resource finds the **worker server primary** that has a VTMOS of */*/*/*. Because the VTMOS aligns, it uses the worker server data in that document.
- To determine which connection protocol to use, an upward search is done looking for a resource access document that aligns with the VTMOS of the device. Here it finds the **access SSH only** document that aligns with its VTMOS. The task uses the information in this document, which states that it does connections only with SSH.
- To define which credentials to use, another upward search is done looking for an authentication resource that matches the Juniper / Router / 7* / 7.2 VTMOS. In the base realm is a set of passwords in the **authentication primary** resource. It uses these credentials to make a connection to the device.
- To resolve which file transfer server to use, an upward search for a file transfer resource does not find an appropriate generalized resource. The task then uses the file transfer data that is the default in the database so that it can move files back and forth to the device.

In the second example, a device exists in the customer network that the service provider manages. This device is unique compared to the devices in the core network. The device is A-2600-1, and it is a Cisco / Router / 26* / 12.3. When the same four searches are completed for this device, they resolve the following four resources:

- Worker server A
- No resource access document is found in an upward search. Use the default *access resource that is associated* with the driver to determine connection methods.
- Authentication A
- File transfer A

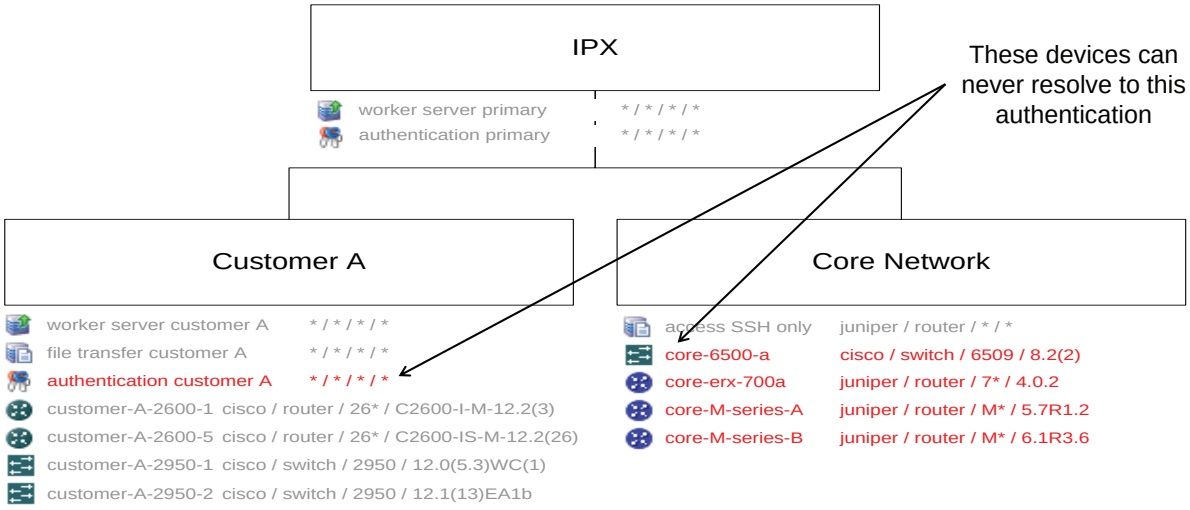
© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Generalized resource location

Location of the resource is critical

The resource must be either in a realm with a resource or in a direct line to the base realm



© Copyright IBM Corporation 2016 34

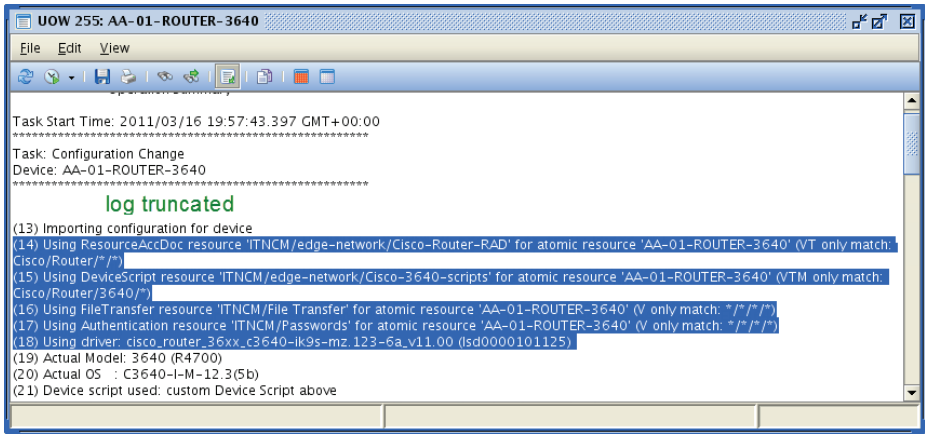
Generalized resource location

The key to generalized resources is the hierarchical nature of their search. The search is always up the realm structure. The search never goes across the realm structure. Devices in the Core Network never resolve to resources in the Customer A realm.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

UOW logs

- The UOW notes the location of the generalized resources needed to process task
- Some resources are in realms; other resources are the defaults



© Copyright IBM Corporation 2016 35

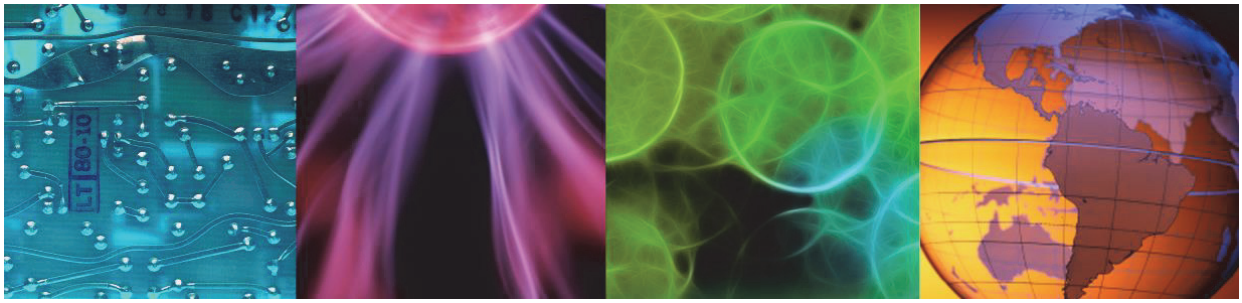
UOW logs

For each task that processes, notes are created in the *unit of work* log about which resources it uses to answer the questions about passwords, process location, and so forth. This documentation is useful for debugging tasks that fail for unknown reasons.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 5 Device authentication

Lesson 5 Device authentication



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe process for finding passwords
- Describe how to configure an authentication resource
- Configure the workers to cache successful authentication credentials

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Overview

You need credentials to log in to devices

External authentication is a good option, for example, RADIUS or TACACS

Two-factor security protocols are not an option

SNMP v1 and v2 devices require community strings

SNMP v3 might require credentials

© Copyright IBM Corporation 2016

Overview

To authenticate to a network device requires a set of credentials, typically, a user name and password. However, some devices can require an extra password to authenticate to the privileged mode of the device. The privileged mode supports configuration changes. This password is sometimes referred to as the *enable* password. If possible, use external authentication servers, such as TACACS or RADIUS.

Two-factor authentication is not an option for device access. Two-factor authentication typically entails a personal identification number and a randomly generated token, which the user holds to authenticate to the device. Tokens often expire between the time when the UOW is submitted and the time when the UOW runs.

If the device is using Simple Network Management Protocol (SNMP) to gather configuration data, community strings are necessary for SNMP v1 and v2, and credentials for SNMP v3.

Note: If network devices use external authentication, create a separate set of Netcool Configuration Manager credentials on the authentication server for device login and ensure that they do not expire. The software cannot automatically change its passwords, and expired passwords cause all *units of work* to fail.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Finding device credentials for VTY access

Step 1: Uses user-defined credentials that are set on a UOW

Step 2: Checks device properties or realm-based resource access document for nondefault user name or passwords

Step 3: Resolves the nearest, best-matching, realm-based authentication resource, the preferred method

Step 4: Uses default credentials in the database, if other options do not work

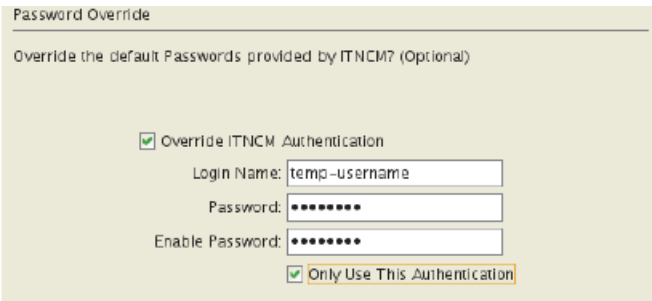
© Copyright IBM Corporation 2016

Finding device credentials for VTY access

You can place device credentials in several locations. The software does a search in four locations as it does its upward search. Initially it checks the unit of work to determine whether a user specified the credentials when creating the task. Next, it checks the access resource that it resolved for determining connection protocols. If no credentials are found in either of these searches, it searches for an authentication resource. Finally, if it finds no credentials, the user name and password for device connections is taken from the database.

The most logical places for credentials are defined either in the unit of work or through an authentication resource.

Corporate policy might require applying credentials to each unit of work. A financial customer designed its change management process to have temporary device credentials that are only good for a specific change window. Setting credentials on each *unit of work* is the only practical method for supporting this practice.



However, the preferred method is to use an authentication resource as it is designed specifically for credential management.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

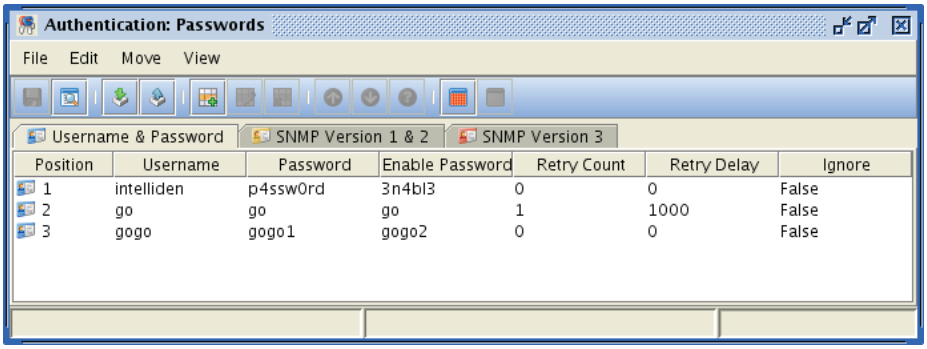
Authentication resource

You can create credential sets

Each set is tried, in order, until a successful connection is made

You can enable retries to support devices with erratic login behavior

Can maintain historical passwords to support local authentication



© Copyright IBM Corporation 2016 39

Authentication resource

The authentication resource uses credential sets. It uses each set to attempt access to the device. It tries them in order until it makes a successful connection. If no credential set works, the task in that UOW fails.

You create the authentication set in the highest level realm where the passwords take effect. When creating the resource, you must provide VTMOS criteria. In most cases, you use the default criteria, */*/*/*.

When you edit this resource, you create credential sets, which you need to enable by setting **Ignore** equal to False.

When you create or edit a credential set, you can also set the retry count and retry delay. You can set these values to support retries of the same set of credentials after waiting a specified number of milliseconds.

If you use local authentication on devices, credentials are saved in the configuration, the authentication resource can keep historical passwords. Keep these old passwords in case one of these versioned configurations with previous credentials is restored.

Important: When you create an authentication resource, be sure that you have modify rights on resources in the realm you are placing it. If not, you do not have the rights to edit the passwords in the document.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Successful credentials

After a successful authentication, cache the device credentials to prevent future login failures that cause side effects, such as security audits

Sticky credential behavior is set in the **rseries.properties** file

$INSTALL_HOME/config/properties/rseries.properties

```
#==============================================================================
# Core properties
#==============================================================================

# Switch on to identify this worker server
Core/loadedSchedulerUpdatePeriod=5000
Core/unloadedSchedulerUpdatePeriod=20000
Core/smtpServer=mail.ibm.com
Core/useCachedAuthCredential=true
Core/workerServerUpdatePeriod=30
```

© Copyright IBM Corporation 2016

Successful credentials

In many environments, a series of passwords are possible candidates for device login. The authentication resource can contain each of these credential sets. When accessing the device, each credential set is tried in order until it makes a successful connection. By default, this process repeats for each future connection. To prevent these extra attempts at authentication, you can configure the worker servers to use cached credentials that it saves after the first successful login. Subsequent logins use the cached credentials before trying others.

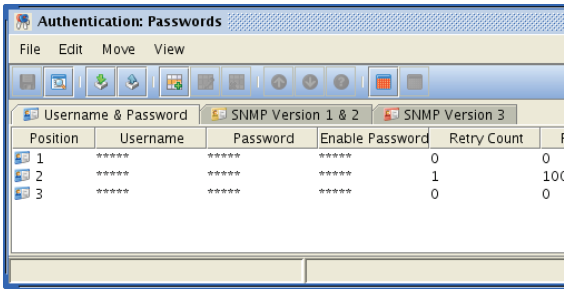
You must configure this sticky credential behavior on all worker servers and is in the rseries.properties file. Set the useCachedAuthCredentials to true and restart the worker server.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Encryption of authentication resource

Requires modification rights on authentication resource to see unencrypted credentials

Passwords are always encrypted in the database




```xml
<?xml version="1.0"?>
<authentication encrypted="true">
  <entry>
    <name>usernamepass</name>
    <set>
      <username>c2dcb2b946e9fe0b08f9</username>
      <password>db872068ff3334ae</password>
      <enablepassword>98c0ddc206e7</enablepassword>
      <retryCount>0</retryCount>
      <retryDelay>0</retryDelay>
    </set>
    <set>
      <username>ccc8</username>
      <password>ccc8</password>
      <enablepassword>ccc8</enablepassword>
      <retryCount>1</retryCount>
      <retryDelay>1000</retryDelay>
    </set>
    <set>
      <username>ccc80333</username>
      <password>ccc8033347</password>
      <enablepassword>ccc8033344</enablepassword>
      <retryCount>0</retryCount>
      <retryDelay>0</retryDelay>
    </set>
  </entry>
  <entry>
    <name>snmp</name>
    <!--<set><communityString>public</communityString><
  </entry>
</authentication>
```


© Copyright IBM Corporation 2016 41

Encryption of authentication resource

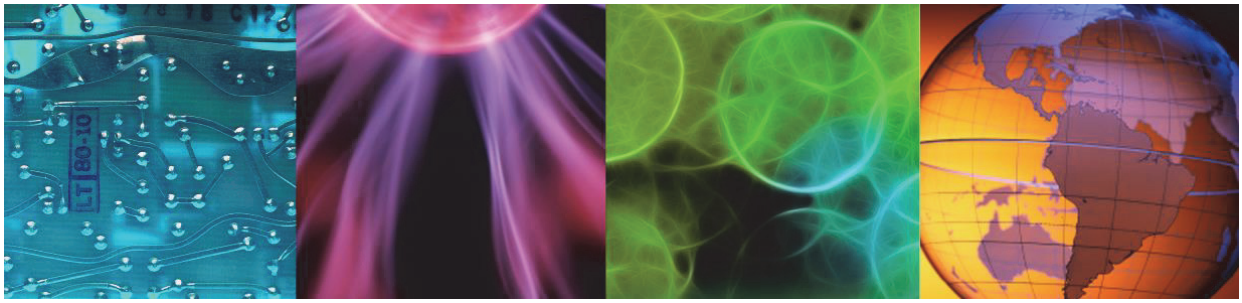
It is important that the authentication resource has encryption and security rights to prevent unauthorized viewing of user names and passwords for device access. For users to view credentials in clear text, they must have resource *modify* rights in the realm the authentication resource is located. You define this property for the group the user belongs to.

Regardless if a user has *modify* rights to view passwords in the authentication resource, be assured, the credentials are always encrypted in the database.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 6 Access resource

Lesson 6 Access resource

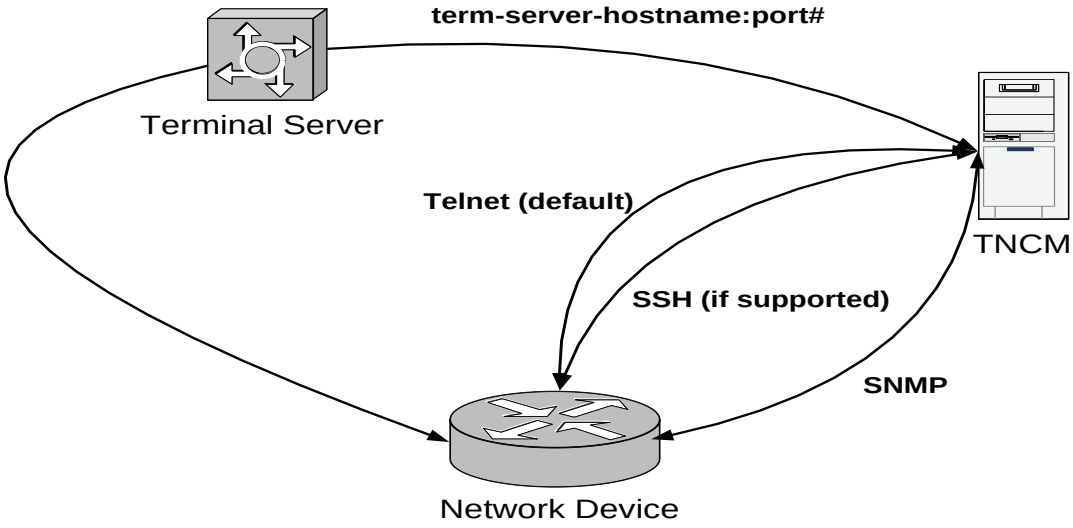


© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to describe the features that an access resource can configure.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Communication methods



© Copyright IBM Corporation 2016

Communication methods

Most device connections that Netcool Configuration Manager uses are over a virtual terminal (VTY) connection. Telnet is the default connection method for most drivers. However, if the device supports secure connections, it employs SSH connections.

In rare cases, a driver supports only an SNMP connection to devices to get and set configuration data. SNMP connections are rare because most devices have a command-line interface that you can use for configuration purposes.

You can also connect to the devices through a terminal server by specifying a host name and port number that the terminal server uses to connect to the device. This method is not for general communication, but you can use it as a backup. It requires customizations for each individual device that have terminal servers that are attached to them.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modifying default access methods

Communication method defaults:

- Telnet
- Streaming get
  Example: **`show running-config`**
- FTP or TFTP put
  Example: **`copy ftp running-config`**

Other configurable defaults:

- Rollback options
- Synchronize running and stored configurations
- Comparing the device configuration with the one in the database.

Modifications at two levels:

- Individual device, accessed through GUI
- Realm or VTMOS, accessed through GUI

© Copyright IBM Corporation 2016 44

Modifying default access methods

It was mentioned earlier that all device drivers have a default access resource that is part of their software. These access resources usually have the following default behaviors.

- Most drivers use Telnet to connect to devices.
- When it retrieves data from the device, it streams over the Telnet session.
- When it places data on the device, for example, a configuration change. The device requests it by using a file transfer protocol, either FTP or TFTP.

The driver sets other defaults, including these examples:

- Rollback options for when changes fail on devices
- Saving configurations when changes are made
- Checking the device configuration state against the database.

If these behaviors are acceptable to the customer, you might not need to customize the implementation. However, you can accomplish customizations at two levels in the application:

1. You can modify each individual device.
2. You can create a realm-level resource access document.

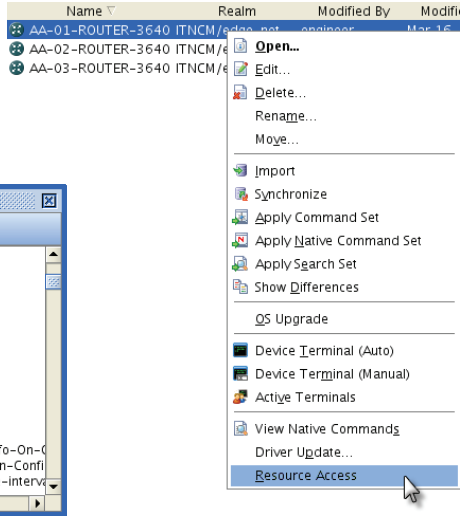
 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

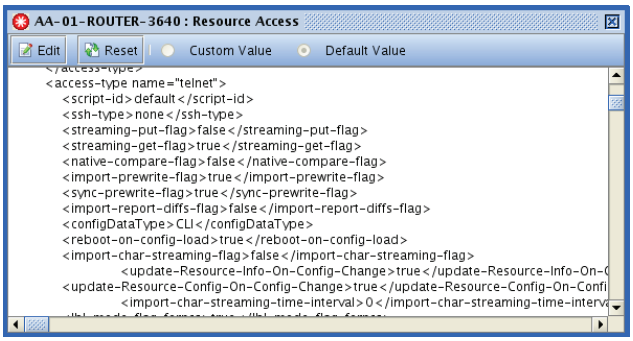
Modifying individual device properties

Use the **Resource Access** page for the device

Apply to the device you are modifying

Set nondefault documents as top priority





© Copyright IBM Corporation 2016

Modifying individual device properties

You can modify the default behaviors to reach individual devices under management. When you right-click a device, you can select **Resource Access** and set custom values that are unique to this device.

By changing this local resource access document for the device, you break the upward search for applicable realm-based resource access documents. If the realm-level document contains changes that apply to this device, they are not effective.

Whenever possible, use realm-level resource access documents and avoid this method of customization. In some instances, this local customization is necessary. The problem with making these kinds of granular customizations is that the application does not indicate that it is customized. The only place where you see that the device is customized is in the UOW log.

© Copyright IBM Corp. 2016

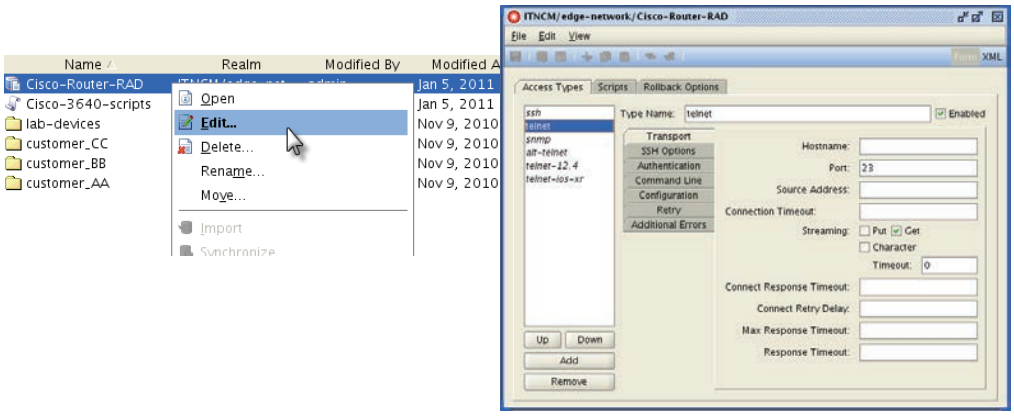
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modifying realm-level resource access

Uses a resource access document (RAD)

Is a VTMOS modification for devices on or below a realm

Requires at least the vendor and type



© Copyright IBM Corporation 2016 46

Modifying realm-level resource access

The realm-based resource access document (RAD) is a VTMOS-based document that affects the applicable devices at or beneath the realm where it is located. Use this method to modify the default interactions for a particular driver.

When you create a RAD, you must define at least a vendor and type for the document. This requirement means, for example, that all Cisco Routers at or beneath the realm use the values that are set in this document.

When you modify the resource access document, you can modify many defaults, based on what must be customized. Most values never change because they are for unique network management situations.

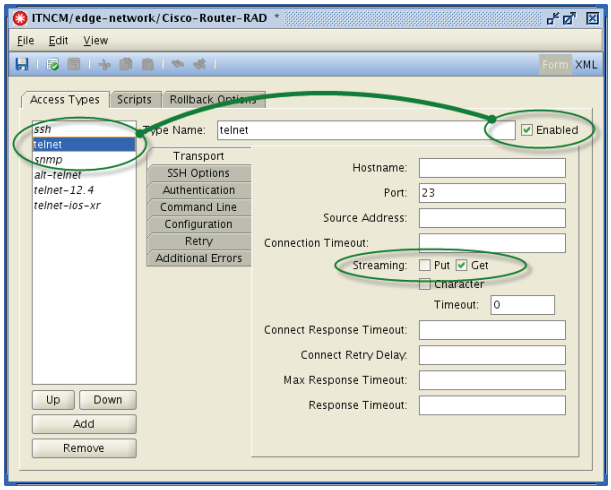
 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Access types and transport options

Determines which connection methods to try

Be sure to correctly order the methods you enable

Streaming or file transfer to move data



© Copyright IBM Corporation 2016

Access types and transport options

The most common reason for creating a RAD is to enable SSH connections to supported devices. The default enables only the telnet access type. Enabling SSH requires you to mark the appropriate check box and then the RAD attempts all *enabled* access types, in order, from top to bottom until it makes a successful connection.

A series of subtabs exist that you use to further configure the specific interaction that the access type uses. Any changes that you make to a subtab affect only that access type and no others. If you highlight Telnet and change its default streaming method, that change is not reflected in the SSH access type.

Common changes to the default RAD include these examples:

- Determining which connection protocols to use and in which order; for example, try SSH first and then Telnet
- Determining whether to use file transfer protocols to put and get data from the device

This training addresses the **Transport**, **SSH Options**, **Configuration** subtabs of **Access Types.**

The **Transport** subtab defines which port to connect to the device through and most importantly, which file transfer protocols to use to PUT data on and GET data off the device. On this slide and the following ones, refer to the student notes to get detailed information on the fields defined on the subtab.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The following items define the fields on the **Transport** subtab:

Hostname: Use this value to define a host name other than the device name used in the resource browser. The intent is to use it with a terminal server, where you connect to a host name and port number. This connection protocol uses this value.

Port: By default, either Telnet or SSH use this standard port number. A precedent exists for connecting to the VTY session on a nondefault port. The Cisco 15454 switch used port 380 for Telnet sessions. Finally, the terminal server connection and the console port use it. Use this final option only on an individual device RAD and not at a realm level.

Source Address: Use this parameter when source routing is implemented in the design of the solution. This feature is rarely used and was introduced to handle multiple private address spaces from a single worker server.

Connection Timeout: Use this value to determine how long to wait for a TCP connection to a socket on the network device. The operating system that the worker server is installed on determines the default of this value.

Streaming Put: This value determines how to place data on a device. Typically, this value is not selected. This option means that it uses a file transfer protocol, either FTP or TFTP. Selecting this check box determines which set of device scripts run at certain points in the device interaction.

Streaming Get: This option determines how to extract data from a device. The default is to stream data off the device. This option also determines which scripts run at various points in the device interaction.

Character: This option slows down how fast characters are sent to the device. You rarely use this option with current network hardware, but is included in the communication process to handle old network equipment, which have slow VTY connections. Selecting this option implements the character timeout.

Character Timeout: Use this value, which is defined in milliseconds, to place timeouts between each character that is sent to the device. You must select **Character** to use this timeout.

Connection Response Timeout: This timeout defines how long the application waits for the global connection script to complete. The global connection script includes accessing the device with authentication credentials and setting up the session for terminal interactions with Netcool Configuration Manager. The application is encoded with a default value of 20000 milliseconds.

Connection Retry Delay: Use this option to check sockets throughout the device interaction. This process is done in code. A socket check typically entails sending a carriage return to the device and waiting for it to return a prompt. If the device does not return a prompt in a timely fashion, the socket is considered dropped and a reconnection occurs after waiting for this period. The default value is 1000 milliseconds.

Max Response Timeout: This timeout determines the maximum amount of time the application waits for a value, as defined by the device script, to return from a device. If this timeout is exceeded, the task fails. The default value is 300000 milliseconds. The response that the application expects is usually the device prompt.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Response Timeout: This timeout serves a similar function to the Max Response Timeout. Its default is 600000 milliseconds, or 10 minutes. The timer is looking only for data that returns to the device in the stream. Any time that data is returned, the Response Timeout is reset to zero.

© Copyright IBM Corp. 2016

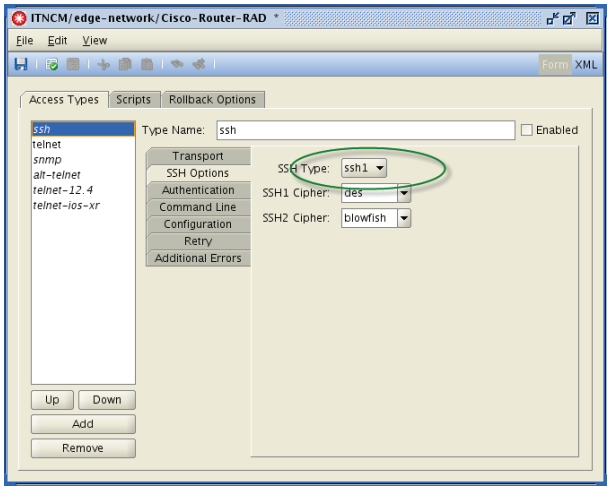
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

SSH options

Only applicable for SSH access type

SSH type is important

Cipher type is default preference



© Copyright IBM Corporation 2016

SSH options

Use the **SSH Options** subtab only to configure the **ssh** access type. This tab determines which version of SSH to use and selects preferred ciphers to connect to the device with. Selecting SSH1 enforces that it uses only that protocol. Selecting SSH2 attempts connections with that protocol but also tries SSH1 as a fall-back option. Selecting a cipher defines a preferred encryption method. If a device does not support that method, the SSH client auto-negotiates a cipher that the device supports.

The following items define the fields on the **SSH Options** subtab:

SSH Type: The options are **none**, **ssh1**, or **ssh2**. Selecting ssh1 means the first attempt to connect to the device uses the version 1 of the protocol. However, if that attempt fails, it tries to connect to the device with version 2. Selecting ssh2 means that it attempts to connect only with version 2 of the protocol. Devices supporting only version 1 fail to connect.

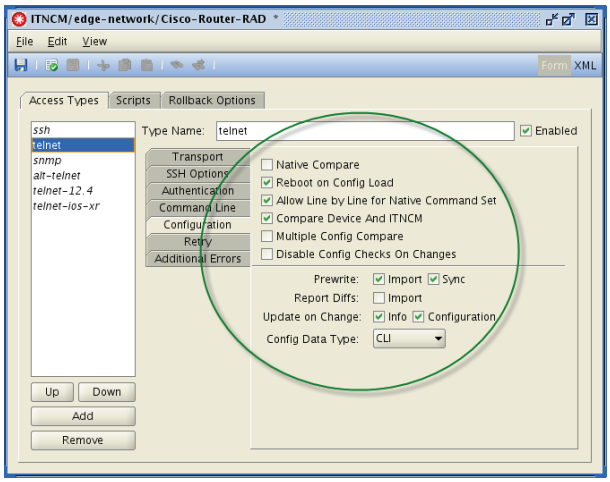
SSH1 Cipher: This value is the preferred cipher that is used for SSH1. If the device does not support it, MindTerm autonegotiates another cipher.

SSH2 Cipher: This value is the preferred cipher to use for SSH2. If the device does not support it, MindTerm autonegotiates another cipher.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuration

Controls many default behaviors with device interaction



© Copyright IBM Corporation 2016

Configuration

The **Configuration** subtab has the greatest ability to affect how devices that resolve to this RAD process *unit of work*. These fields control the workflow that a UOW uses and the checks that it undertakes. Changes on this page control how your network implements configuration management. For example, a default behavior of the work flow is to compare the configuration on the device with the configuration in the database when applying a change. When a difference is found, it stops processing the change. Clearing the **Compare Device with ITNCM** disables this behavior. Review the student guide to understand each of the features available on this subtab.

The following items define the fields on the **Configuration** subtab:

Native Compare: This option determines how a configuration is initially compared with the configuration in the database. By default, a running configuration is converted into XML first and then compared with the XML configuration in the database. This comparison is much more accurate than a raw comparison of configurations. However, converting large configurations to XML drastically affects performance. Selecting this check box causes a comparison of the raw configuration that the database stores with the raw configuration on the device. If they are the same, UOW processing continues. When it finds a difference, the device configuration is converted to XML, compared to the database, the results are reported, and the UOW fails. This check box depends on the **Compare Device And ITNCM** option.

Reboot on Config Load: This option determines the behavior when you select a disaster recovery UOW or certain rollback options. The scripting for these two situations includes taking the raw configuration for the device and downloading it into the startup configuration for the device. If you select this option, it submits a restart to the device to restore that startup configuration.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Allow Line By Line for Native Command Set: This option supports checking as each line of a configuration change is applied to a device. It works with the Retry page. Use this option only when a native command set is configured for *line by line* mode. The biggest effect of using this option is that it overrides FTP behavior. It forces the change to stream to the device so that each line of change is checked as it is applied.

Compare Device and ITNCM: Clearing this check box disables the comparison to verify that the device is synchronized with the configuration database. By default, it does not authorize changes into the network unless the configuration on the device matches that which is in the database.

Multiple Config Compare: This feature supports the comparison of the running and stored configurations of a device. When differences are found between these two configurations, the UOW fails.

Disable Config Checks on Changes: This feature is directly relates to the Update on Change: Config selection. A device might have what is referred to as a *stale configuration*, meaning a change was applied but the resulting configuration was never captured. The device has a change that is not synchronized, and the device is marked as stale. A device in a stale state, by default, cannot accept another change until it is synchronized. Selecting this option overrides this rule and authorizes unlimited changes to a device that is in a stale state.

Prewrite: With a Cisco-like device, this option causes a command to save the running configuration to the file system on the device. For Cisco devices, it is a **copy running-config startup-config** command. You can determine whether you want this action to happen when a device is imported and when it is synchronized. By default, it stores any changes to the running configuration.

Report Diffs: Selecting this option reports any differences between the running and stored configurations on the device when it is imported into the database. You use the **prewrite: import** option to remove these differences after the import.

Update on Change: These two check boxes determine what information updates in the database when a change is made to the device, either a command set or ad hoc change. By selecting **Info**, it processes the commands that are in the **diag** script. This option is for collecting the information that goes into the device hardware tab. Note, the **Configuration** selection determines whether it pulls the running configuration from the device after a change is made and stores it in the database. Clearing this option causes the device to become stale after any change is made to it. A stale device does not support any further changes to it until a synchronization action is processed to pull the changed configuration.

Config Data Type: This selection defines the nature of the data that is retrieved from the device. Data types include CLI, BIN, and CLIandBIN:

- The **CLI** data type implies that the data that is retrieved from the device is in an ASCII format and is saved as such. This option processes a script that pulls a single ASCII file from the device.
- The **BIN** data type means that the image file is an uneditable configuration file. It must be stored differently from a CLI. This option processes a script that is different from the CLI script.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

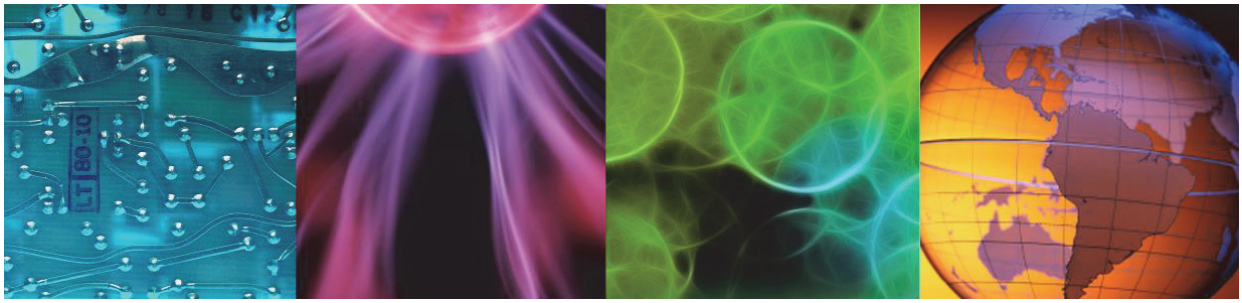
- The **CLIandBIN** data type is for a situation that stores configuration data in multiple locations. This option is the case with Cisco IOS switches, which use both a CLI configuration file and a VLAN.DAT file to hold data. This selection starts an extra script on top of the CLI scripts that run. This script retrieves the VLAN.DAT file and saves it as an uneditable file.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 7 File transfer resource

Lesson 7 File transfer resource



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to use the file transfer resource to designate a server that is used as a file transfer point between the worker server and the device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Moving data between workers and devices

Worker servers are responsible for moving data back and forth to a device

Data moves by either streaming or a file transfer protocol

Worker servers search for a realm-based file transfer resource

If none is found, they default to FTP server information, which is defined at installation

© Copyright IBM Corporation 2016 51

Moving data between workers and devices

The worker server moves data to and from the network device. It either resolves the resource access document or the defaults in the driver to determine how to move data back and forth.

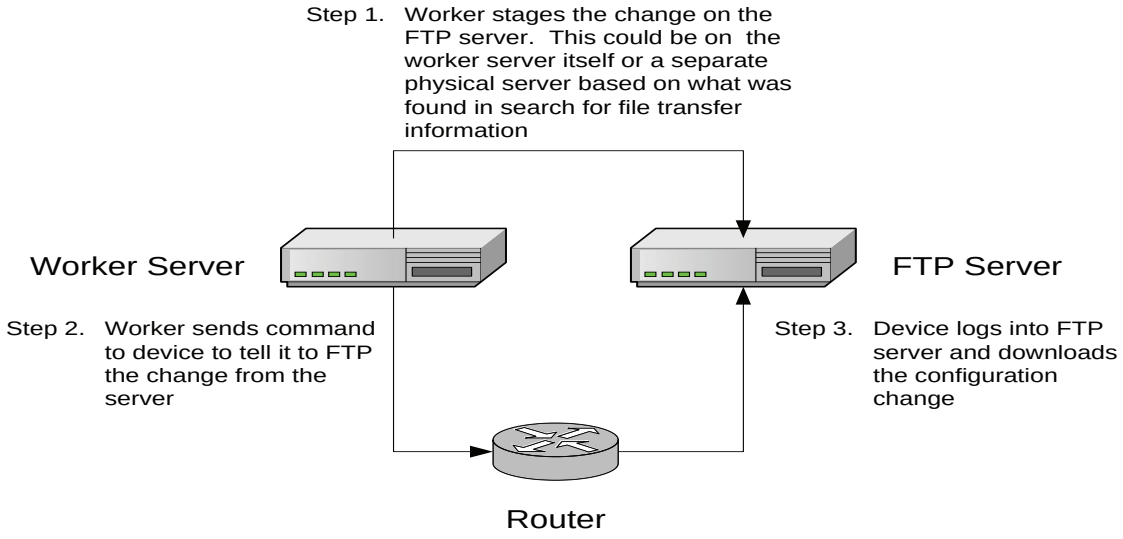
Based on these settings, the worker either streams the data over the VTY connection or uses a file transfer process. Both of these options are coded in the device scripts. By default, most interactions between devices use these processes:

- The worker server uses streaming when it retrieves data for various purposes.
- The worker server sends commands to the device to have it use a file transfer process to place data on the device.

When a task in the UOW requires file transfer data, the worker server does an upward search for a resolvable file transfer resource. If it does not find one, it uses default FTP data that is in the database.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Moving a configuration change to a device



© Copyright IBM Corporation 2016

Moving a configuration change to a device

The standard method for submitting a configuration change to a device is illustrated on the slide. An assumption is that an FTP server never runs on the network device and that the device acts only as an FTP client.

1. When a task is going to change a device, the worker server stages the change file on an FTP server. The architecture of the solution determines whether the file is hosted on the file system of the worker server or moved to a separate server. The IP address of the FTP server indicates whether it is a local move of the file or an actual transfer to another server.

2. The worker server sends a command to the device through the VTY connection to tell the device where to find the change file.

3. Using the command in step 2, the device pulls the configuration file from the FTP server. It then applies the change by using its standard change processes, which are included in the scripts.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Localizing all services on a worker server

A local worker server does all UOW activities

Workers have their own FTP servers

Requires a local name resolution strategy

Create a standard host name for the FTP server, such as ftplocalhost

- Each worker resolves the host name to its network-facing IP address
- Use the **/etc/hosts** file to resolve the name

© Copyright IBM Corporation 2016

Localizing all services on a worker server

A best practice with a worker server is to localize all task processing on it. This option makes your implementation more flexible and easier to debug. This includes the following task activities:

- VTY connections to devices, this procedure happens by default for worker servers
- Converting CLI to XML and back
- Moving files back and forth between worker and device

Keeping everything local to the worker server requires that it include an FTP server process. By having each worker act as its own FTP server, it activates these features:

- Removes the single point of failure that exists in a solution that has only one FTP server to host all changes
- Permits the worker servers to redeploy to other realms to enable better throughput
- Provides a change file that is local to the worker server that creates it.

To implement this localization, each worker server resolves a common host name that returns the IP address of that worker. This option also requires that the name resolution happens in the **/etc/hosts** file for each server.

For example, choose a host name that is unique to the network and does not resolve to any DNS entry, for example, ftplocalhost. In each worker host file, add this host name to the entry that defines the IP address of the network-facing interface.

© Copyright IBM Corp. 2016

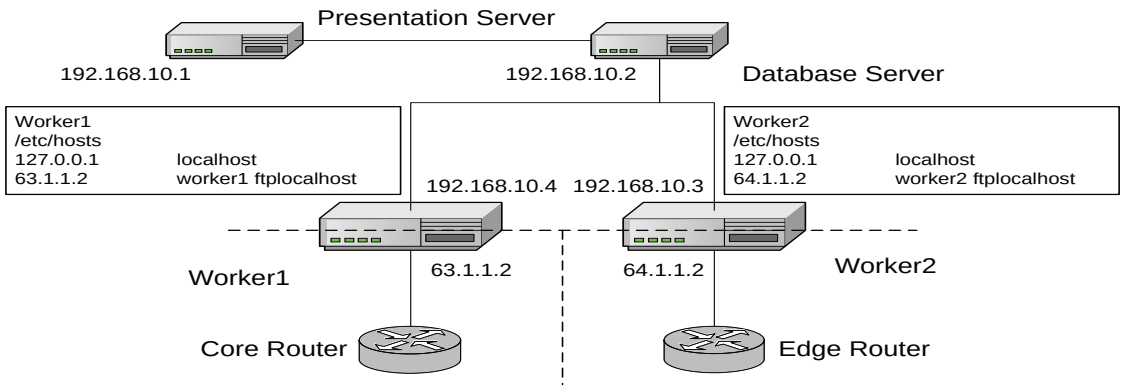
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Resolving the local FTP host name

Tivoli Netcool Configuration Manager operates in private address space

Workers straddle the private and public address spaces

Edge devices cannot route to resources in core



© Copyright IBM Corporation 2016

Resolving the local FTP host name

In the example on the slide, a private 192.168.10.0 address space hosts the application. The addresses of the network interface cards (NIC) of all the presentation and worker servers are in the 192.168.10.0 subnet. The workers must also communicate with devices that are in the public address space. Each worker has a second NIC that faces the public side of the network that they are going to manage.

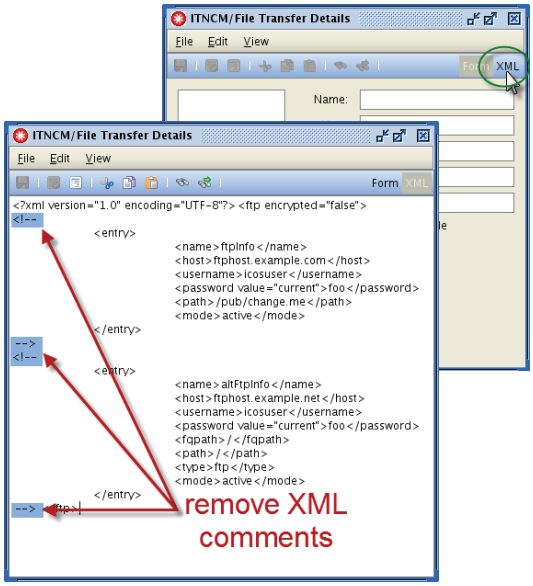
To further complicate this implementation, core devices and edge devices exist. They run on different network subnets, 63.1.1.0 and 62.1.1.0 subnets. Devices on these two subnets can communicate only with workers that are dedicated to their network. For example, Edge Router cannot communicate with Worker 1 that is associated with the Core network.

To solve this issue, each worker server implements an FTP server. The FTP host name resolves differently on each worker, ftplocalhost. Before Netcool Configuration Manager sends the command to the device that tells it to retrieve a file, the worker server resolves the host name to an IP address. It resolves through the name, ftplocalhost, to the network-facing IP address that is unique to the worker server. This step ensures that devices in the edge network do not resolve to an FTP server in the core network, which is unreachable.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Initializing the file transfer resource

1. Create file transfer resource in the appropriate realm
2. Edit the resource
3. Click **XML**
4. Remove XML comment tags
5. Click **Form** to return to the GUI



© Copyright IBM Corporation 2016

Initializing the file transfer resource

After you create a file transfer resource in the GUI, you must edit it and configure its settings. To initially configure the resource, you must remove the XML comment tags that are part of the default document. Click the **XML** option in the upper right corner. After that is complete, return to the **Form** page.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining data for ftpInfo

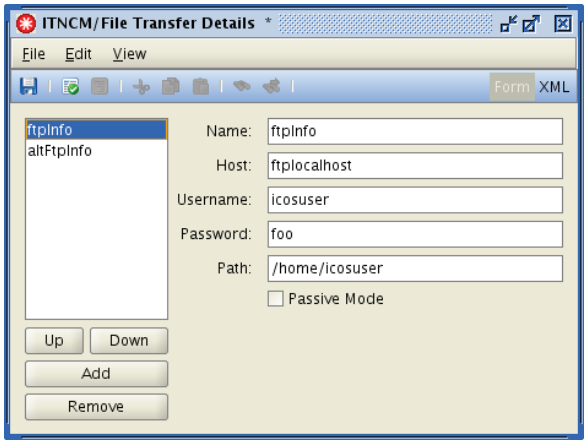
Name: Do not modify the name

Host: Worker server resolves IP address or host name

Username: FTP user name

Password: FTP password

Path: The file system location where the worker server stages the transfer files



© Copyright IBM Corporation 2016

Defining data for ftpInfo

On the Form page, you define the details of the FTP server. The application defines one server for hosting configuration changes that is ftpInfo. Use a secondary FTP server that is the altFtpInfo, for hosting image files to use in the OS Upgrade process. It is important that you do not modify these names as the worker servers search for them. You must define a host name for the FTP server. This value is a resolvable host name or an IP address. In the example, the host name is ftplocalhost, which resolves locally to the worker server IP. Next, you must define a user name and password that log in to the FTP server through an FTP client. Finally, the worker server is going to stage the configuration change files on a local file system directory that the path attribute defines. The worker then moves the file to the local or remote FTP server. The worker compares the FTP server host name address with its local address to determine whether it needs to move the file by FTP.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Describe the advantages of modeling
- Explain how models support security
- Identify methods of device interaction with software
- Explain how to use generalized resources to support customization

© Copyright IBM Corporation 2016

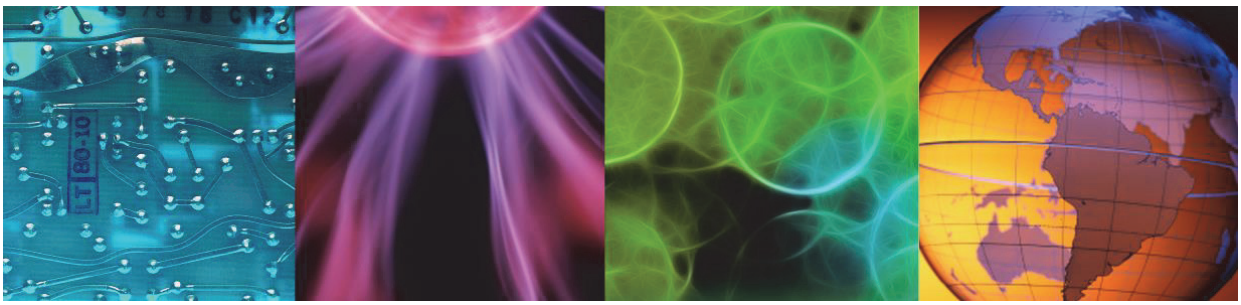
Summary

In this unit, you learned about managing devices and the customizations that support devices in a network. The unit covered the concept of modeling and how modeling can extend to enable security at a command level. You also learned about generalized resources, which are designed to customize interaction between this product and the devices.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

9 Using workflow and scheduling

Using workflow and scheduling



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches you about ticketing and scheduling. Netcool Configuration Manager directs tickets to the appropriate groups for approval. You learn how to estimate how long a UOW might take. You also learn about scheduling *units of work* and setting time zones.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Implement workflow realms to control ticketing
- Estimate how long a UOW takes to complete
- Create schedules for UOW starts
- Implement a recurring UOW
- Describe when units of work run, based on scheduled windows

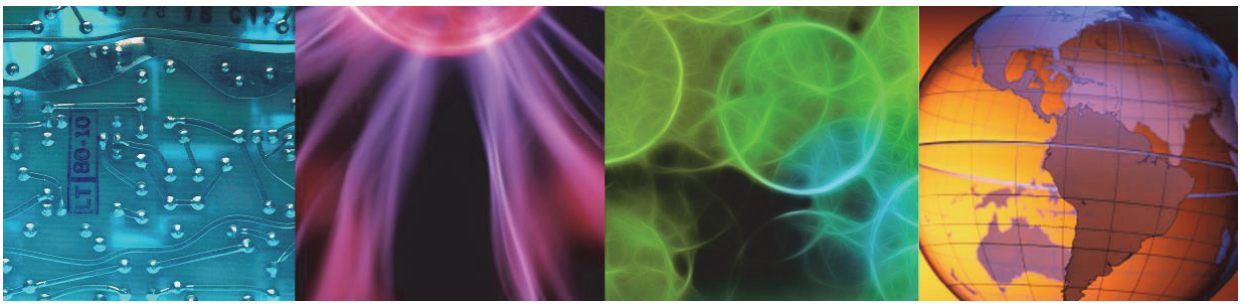
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Controlling the flow of tickets

Lesson 1 Controlling the flow of tickets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe how a workflow realm works
- Configure workflow realms for groups who manage work

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Workflow overview

Workflow is the concept of moving units of work through the queues

Approvals and schedule affect workflow

A UOW waits in *pending approval* until several approvals are met

You cannot control when tickets are delivered

- Tickets are delivered to all approvers as soon as the UOW is submitted
- No hierarchical approval exists

© Copyright IBM Corporation 2016

Workflow overview

Netcool Configuration Manager has four queues for *units of work*. It always uses three of them: *work waiting to execute*, *work that is executing*, *work that is finished*. It uses the fourth queue only when users submit *units of work* that require approval. The workflow engine moves the UOWs through each queue as each requirement is met.

Approvals and schedules affect workflow. *Units of work* wait in the *work pending approval* queue until enough approvals are received. Then, all tickets are delivered to all approvers simultaneously. If you need hierarchical approvals where one group approves a UOW and then the next group sees the approval ticket, you should consider a true ticketing system and design an integration with Netcool Configuration Manager.

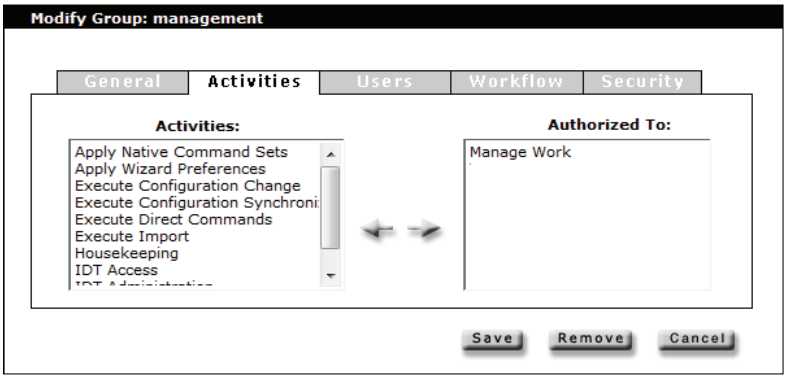
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Controlling the approval tickets

Tickets are delivered to approvers through the workflow realm hierarchy

Workflow realm is a property defined for the approval group

An approval group is a group with the *manage work* activity assigned to it



© Copyright IBM Corporation 2016

Controlling the approval tickets

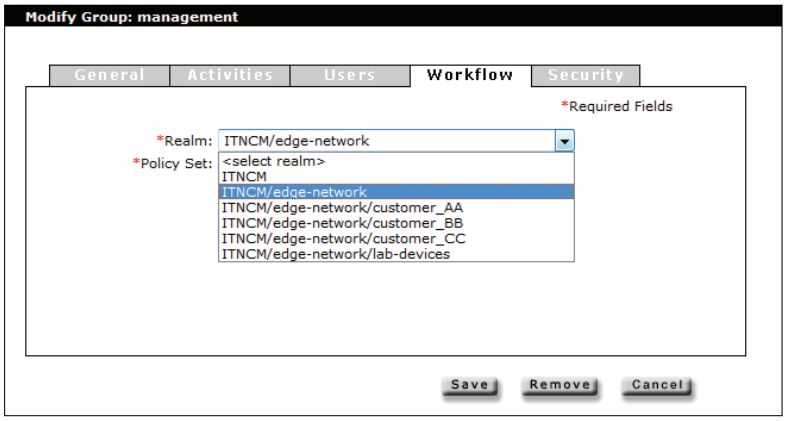
The application defines workflow realms, which are built on top of the resource browser realms. To develop this workflow realm and its associated hierarchy, you must start by defining approval groups. An approval group has the *Manage Work* activity.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Selecting the workflow realm

Groups need manage work rights for workflow realms to have meaning

Groups see tickets only for units of work where *all devices* are in or below the workflow realm



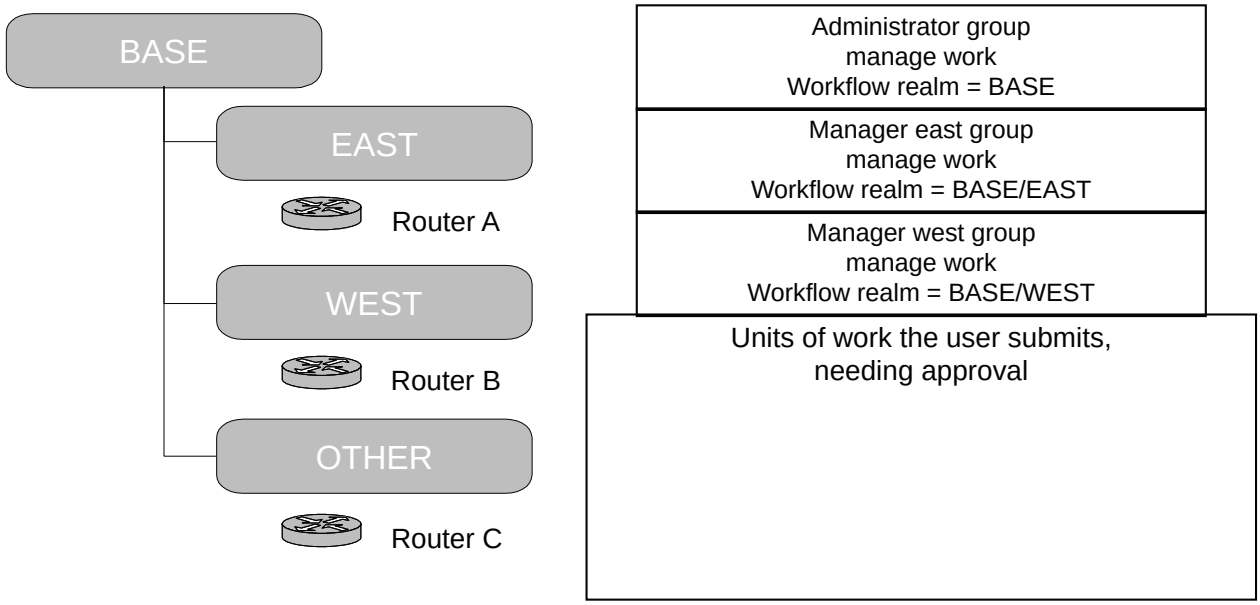
© Copyright IBM Corporation 2016

Selecting the workflow realm

Next, you must define a workflow realm for a group. The workflow realm is associated with a resource browser realm. The best way to describe it is to determine the highest level realm for which the approver has responsibility. It is important to remember that the workflow realm has meaning only for a group with *manage work* rights. This group sees tickets for *units of work* where all the devices in the UOW are in or below the workflow realm.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Approval ticket generation example



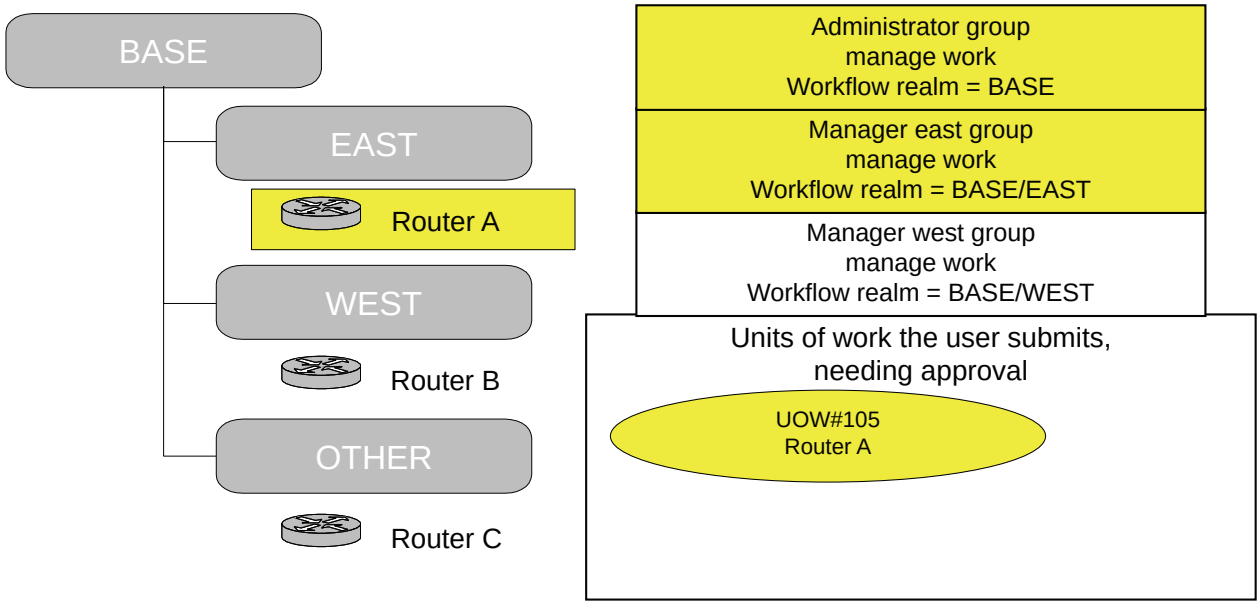
© Copyright IBM Corporation 2016

Approval ticket generation example

In this example, you have a realm structure and three approval groups. Each group has a different workflow realm. The manager east group is responsible for devices in or below the east realm. The same is true for the manager west group, who is responsible for the devices at or below the west realm. The third group is the administrator group, who is responsible for the base realm. When the UOW is generated and submitted, it generates and submits tickets to the appropriate groups.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Approval ticket generation example: East devices



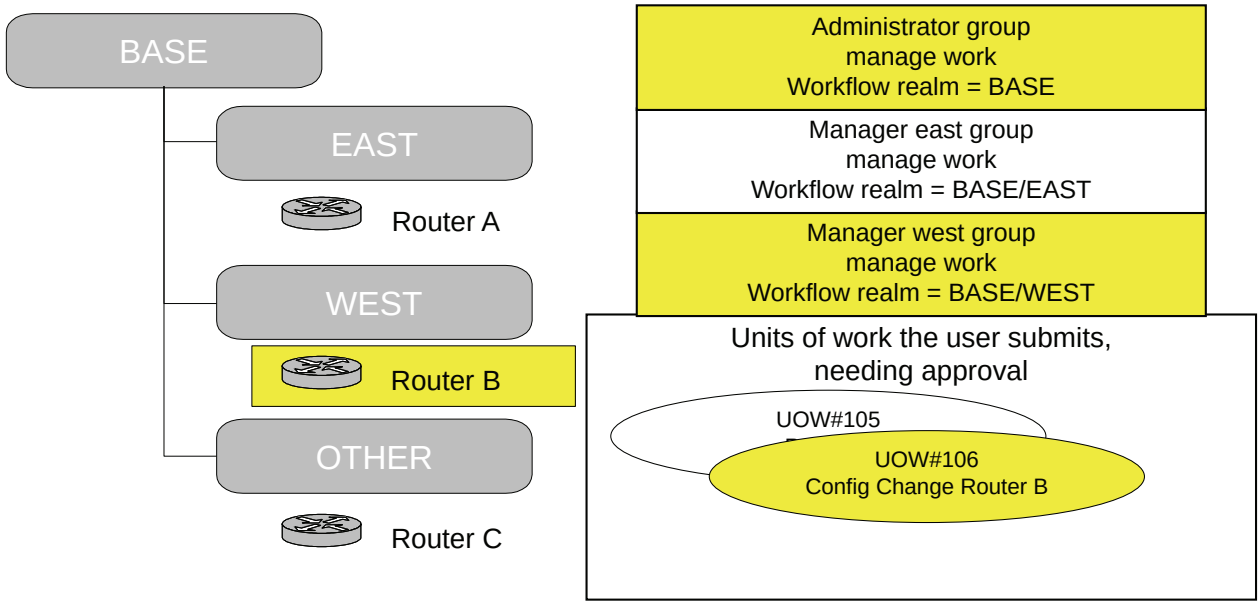
© Copyright IBM Corporation 2016

Approval ticket generation example: East devices

Here, an operator submits a UOW that requires approval. Inside that UOW is a device that is in the east realm. This submission creates tickets for the group or groups that have approval authority for *all* the devices in the UOW. Because this device is in the east realm, it creates tickets for the manager east group and the administrator group. These tickets are generated because all the devices are at or below the east workflow realm.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Approval ticket generation example: West devices



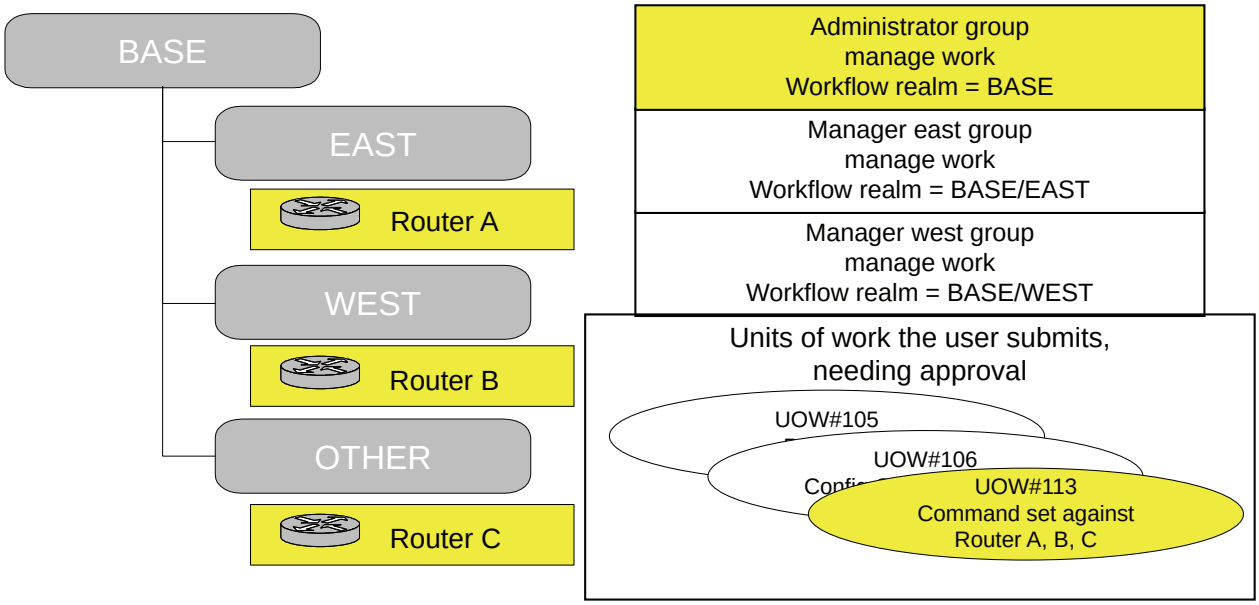
© Copyright IBM Corporation 2016

Approval ticket generation example: West devices

In this example, the same operator submits a UOW that requires approval. Inside that UOW is a device that is in the west realm. This submission creates tickets for the group or groups that have approval authority for *all* the devices in the UOW. Because the device is in the west realm, it creates tickets for the manager west group and the administrator group. These tickets are created because all the devices are at or below the west workflow realm.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Approval ticket generation example: All devices



© Copyright IBM Corporation 2016

Approval ticket generation example: All devices

An operator who needs approval submits a *unit of work* that applies a change to all the devices that are defined in the implementation. In this UOW, three devices are in three different realms. Tickets are sent to groups that have approval authority over all the devices in the UOW. Therefore, a ticket is only sent to the administrator group. The common realm between all three devices is only the Base realm, which is associated with the administrator group. If the administrator does not have the Manage Work activity, the UOW cannot be submitted.

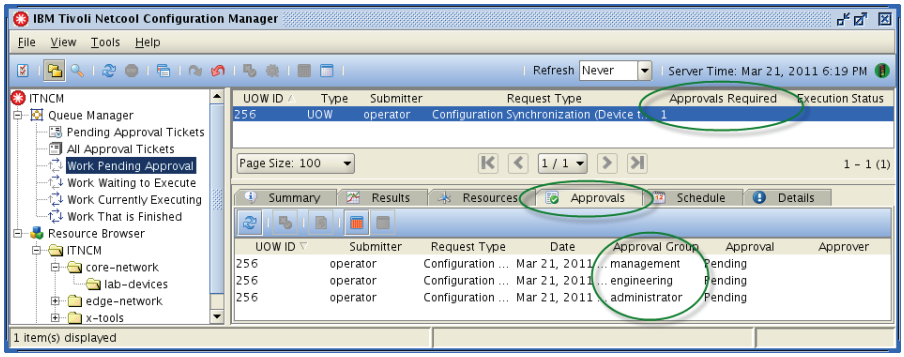
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Approval groups

You can view the number of approvals required

A member of one of the approval groups must accept the ticket

Approver approves the ticket from the *pending approval tickets* queue



© Copyright IBM Corporation 2016

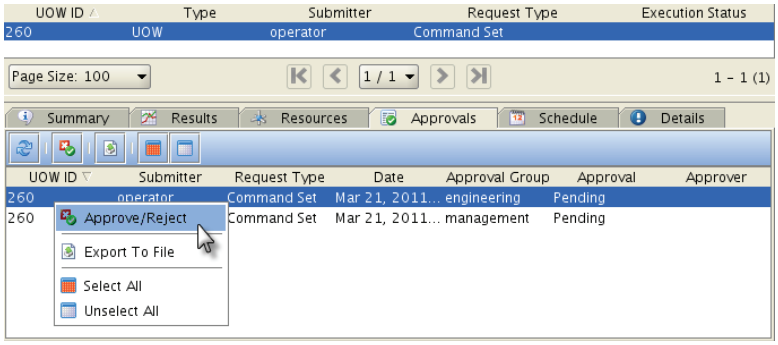
Approval groups

The presentation server assigns to the UOW which groups can approve the task. By selecting the UOW, a user can see which groups can approve it and the number of approvals it requires. While the approvals groups are shown in this view, approvers typically select the *pending approval tickets* queue to show only the tickets that can be approved. A user does not approve the *unit of work* but approves the tickets that are generated when the UOW is submitted.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Ticket approval policy

Only one approval per group is allowed



Units of work expire if they do not receive all their approvals by the scheduled end time

© Copyright IBM Corporation 2016

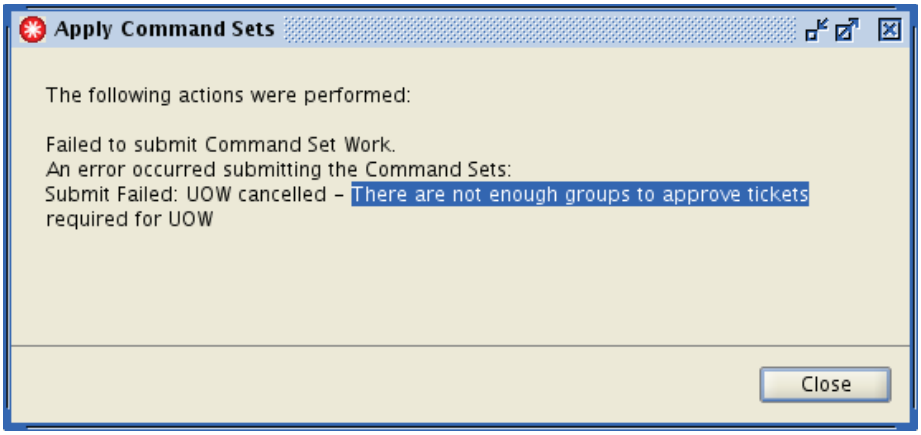
Ticket approval policy

A unit of work often requires multiple approvals to release it for execution. If so, it allows only one approval per group. For example, if it needs two approvals, an approval must come from each of two separate groups. However, if a user belongs to two groups that have the *Manage Work* activity and the appropriate workflow realms, the user can approve the UOW for both groups.

If the UOW does not receive the approvals by the scheduled end time, it expires and moves to the *work that is finished* queue.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Checking for approvers



© Copyright IBM Corporation 2016

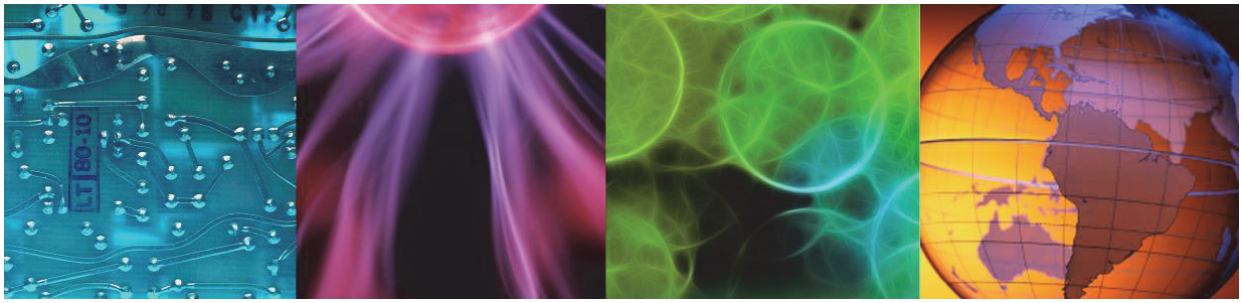
Checking for approvers

The workflow engine checks to see whether enough users are available to approve the UOW. If not enough users are available, you cannot submit the UOW. This problem typically happens when the operator collects too many network resources in the UOW and no one approver has authority for all of them.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Scheduling units of work

Lesson 2 Scheduling units of work



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

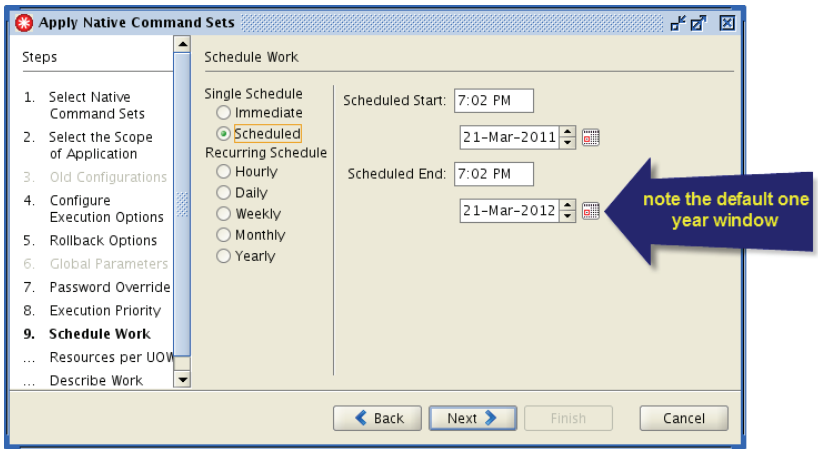
- Describe the scheduling window
- Calculate how long a UOW might take
- Implement a recurring unit of work

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Scheduling overview

All units of work can be scheduled

Scheduling defines a window



© Copyright IBM Corporation 2016

Scheduling overview

When you submit a UOW, you receive a prompt for when to process it. You can select to have the UOW process immediately or to set up a scheduled window for processing. The default window is one year in length. The default start time is the same as the current server time.

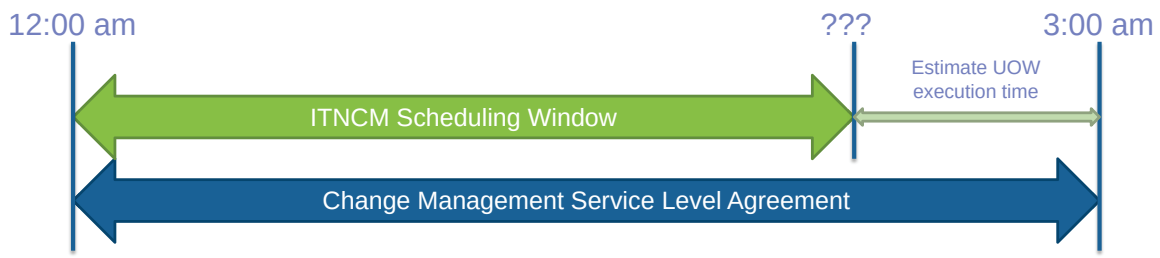
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Scheduling window

A UOW cannot start processing until *scheduled start*

A UOW must be approved and begin processing by *scheduled end*

- A UOW can and will continue processing past the *scheduled end*
- Adjusting *scheduled end* might be necessary, based on change management policies



© Copyright IBM Corporation 2016

Scheduling window

When defining the scheduling window, a worker server cannot pick up the UOW until the scheduled start time matches the server time. A worker server must pick up the unit of work before the scheduled end time that the user defined. These two situations can combine to prevent a UOW from processing by the scheduled end:

1. The UOW must receive approval before it begins execution. If the UOW is not approved before the scheduled end time, it expires and is sent to the *work that is finished* queue.

2. A worker server must pick up the UOW before the scheduled end time. If not, it expires and is sent to the *work that is finished* queue. A UOW might not get picked up because more work is scheduled in the execution window than the worker server can process.

The important aspect about the scheduling window is that it controls only the starting of *units of work*. It does not control the stopping of tasks. *Units of work* can and do continue processing past the scheduled end. Adjust the scheduled end of a UOW so it is sooner than the end of the maintenance window. Corporate policy or a customer service level agreement determines the maintenance window. Processing change tasks past either of these time windows might have consequences for the organization. Subtracting an estimated execution time for the UOW from the maintenance window time is a good practice.

As a reminder, when the user defines a schedule for a *unit of work*, the schedule is based on the time zone that the user configures in their preferences. This time zone is shown in the GUI and might be different for every user.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Estimating UOW execution time

Units of work are made up of tasks

- Number of devices times number of major tasks
  100 devices X configuration synchronization = 100 tasks
  100 devices X 2 command sets = 200 tasks

Tasks take time to process

- Processing time is directly related to size of configuration
  Configurations size < 500 lines = 1 minute per task

Worker servers have a pool of processing threads

- 1 worker server with 20 threads

Units of work execute concurrently or nonconcurrently

- OS Upgrade UOW is nonconcurrent
  10 tasks X 10 minute/task = 100 minutes
- Config synch is concurrent
  100 tasks / 20 threads X 1 minute/task = 5 minutes

© Copyright IBM Corporation 2016

Estimating UOW execution time

To estimate the processing time for a UOW, you must consider the following items.

- *Units of work* are a group of tasks. A task is a major operation against a device. To determine the number of tasks in a unit of work, you must determine how many devices are in the UOW. When multiple command sets are in a UOW, you take the number of command sets times the number of devices.
- Estimate how long a single task in the UOW takes to process. This estimation is difficult. Processing time is directly proportional to size of the configuration on the device for most *units of work*. OS Upgrades take up to 10 minutes. For small configurations fewer than 500 lines, it is fairly safe to assume 1 minute per task for a configuration change.
- The third variable is the number of threads that are available to process the UOW. The user must know something about how the worker servers are distributed and if all workers are eligible to process the tasks inside the *unit of work*. The thread number is related to the *max normal pool* size that is defined for each server. This number is a maximum number of threads that can process on a worker. It is a maximum that might not be reached because of memory limitations. Multiply the *max normal pool* size by the number of eligible worker servers to get the total pool size.
- Determine the concurrency of the UOW. A nonconcurrent UOW is processed in a single-threaded manner and is much slower. The only *units of work* that are nonconcurrent are OS upgrades and multiple command set *units of work*.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using this information, you multiply the number of tasks by the time it takes to complete a single task. If the UOW can run concurrently, you can then divide that time by the total number of threads available on the eligible worker servers.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

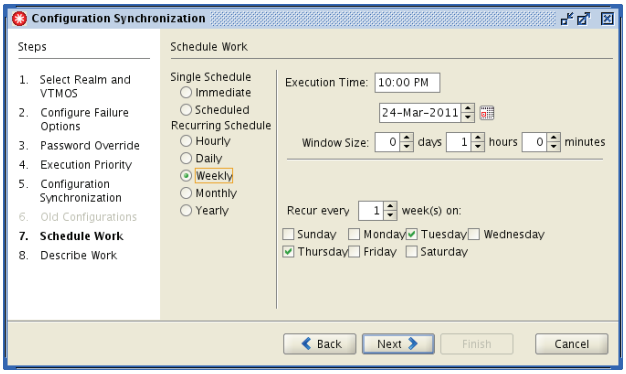
Recurring units of work

You can set the **Recurring Schedule** to create new units of work on a scheduled basis

Scheduled window is based on scheduled start and a window size

Creates a new UOW is after the previous UOW finishes

Stopping, or *dequeuing*, an executing UOW does not cancel a recurring UOW



© Copyright IBM Corporation 2016

Recurring units of work

Along with scheduling the UOW, you can set up the UOW to recur regularly. When defining the execution window, you select a start time and a window size. That start time is used for all future recurrences, regardless of the UOW's regularity. When you select a weekly schedule, your first execution might be any date. Any future executions occur on the days that are selected, given the start time, and window size.

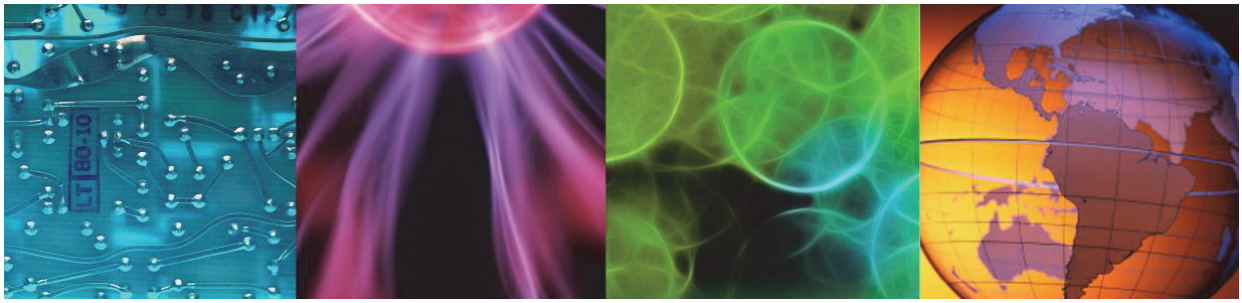
After the UOW is initially created and approved, it is more than likely in the *work waiting to execute* queue. It eventually is picked up and processed. It is only at the conclusion of its execution that a next recurrence is set up and placed in the *work waiting to execute* queue.

To stop the recurrence of a UOW, you must dequeue the UOW that is in the *work waiting to execute* queue. Dequeuing the processing UOW does not stop the recurring aspect of the UOW.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Understanding server time

Lesson 3 Understanding server time



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

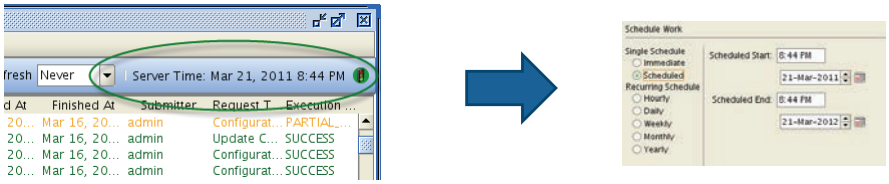
In this lesson, you learn how to perform the following tasks:

- Describe how local time zones affect processing *units of work*
- Describe how time is stored in database
- Describe possible implementation issues with multinational companies

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Overview

Server time defines when a UOW can begin its execution



Users can choose their preferred time zone

© Copyright IBM Corporation 2016

Overview

As mentioned earlier, the server time is a value that you set for the client GUI. Using the preferences window, you select in which time zone the server time is shown. The server time that is shown in the GUI determines when the UOW begins its execution. Users set their time zone preference in their *user preferences*. If a user chooses to change the time zone, it takes about a minute for that change to be reflected in the upper right corner of the GUI.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

When time zones are relevant

You are scheduling UOW for execution

You are managing devices in different time zones, this affects your execution window

Servers are installed in different time zones, this can affect server logs, such as the Intelliden.log and drivers.log

Users are located in different time zones, this might affect when units of work are approved

© Copyright IBM Corporation 2016

When time zones are relevant

Time zones are important when you submit a UOW in one time zone, and the device is in another time zone. The change window on the device might be in the local time zone where the device is located. You must calculate the difference between the user time zone and the device time zone.

Sometimes, the servers are in different time zones. The system administrators might want the operating system time in the local time zone. These differences do not affect the UOW execution times but affect logging time stamps. Synchronize the operating systems for Netcool Configuration Manager with network time protocols and a common time zone.

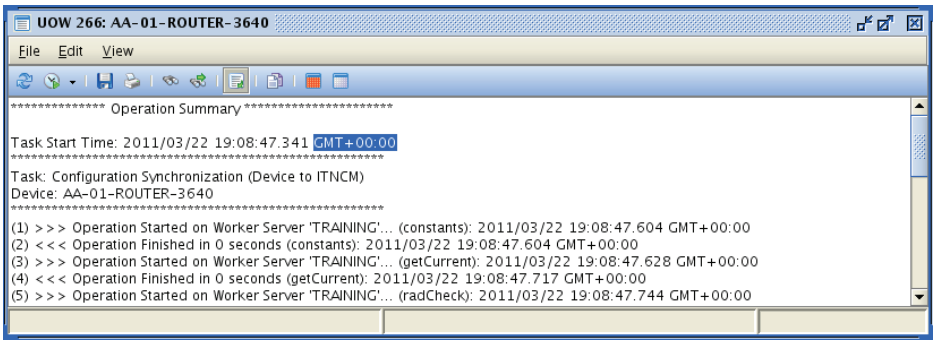
Users might be in various time zones. This issue affects what time *units of work* are approved.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Storing UOW timestamps in GMT

Converts all time to Greenwich mean time (GMT) for internal representation

UOW logs always show their timestamps in GMT



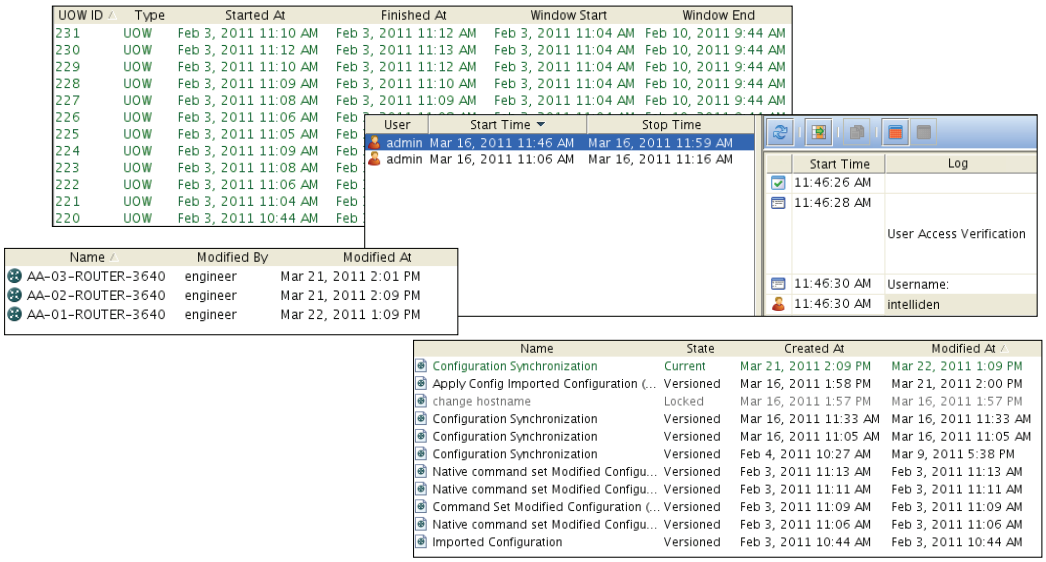
© Copyright IBM Corporation 2016

Storing UOW time stamps in GMT

When the UOW is submitted, all scheduling information is converted to GMT and stored in the database. The UOW log is always shown in GMT. The worker servers use a special Java argument to convert time to GMT, even though the OS clock is in a local time zone. It is important that server clocks are synchronized regardless of time zone so that differences in the seconds and minutes do not appear in the database. Workers that have a system clock that is 5 minutes behind the presentation server might cause *units of work* to sit in the *work waiting to execute* for that period.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Timestamps in local time zone



© Copyright IBM Corporation 2016

Time stamps in local time zone

Outside of the time stamps in the *unit of work* logs, all other time stamps in the GUI are converted into the user's local time zone.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Executing units of work based on local time zone

Units of work start, based on the window defined in the client, regardless of the location of the device

© Copyright IBM Corporation 2016

Executing units of work based on local time zone

Units of work run, based on the time zone that the user sets when they submit the task. Be careful when time zones are far apart. For example, an operator in the Denver network operations center submits a configuration change on a device that is scheduled to run at midnight local time. However, the operator is applying a change to a device that is in Singapore and is responsible for the trading floor. This difference in time zones means that the change is applied at 2 PM Singapore time during critical operation time for the device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Implement workflow realms to control ticketing
- Estimate how long a UOW takes to complete
- Create schedules for UOW starts
- Implement a recurring UOW
- Describe when units of work run, based on scheduled windows

© Copyright IBM Corporation 2016

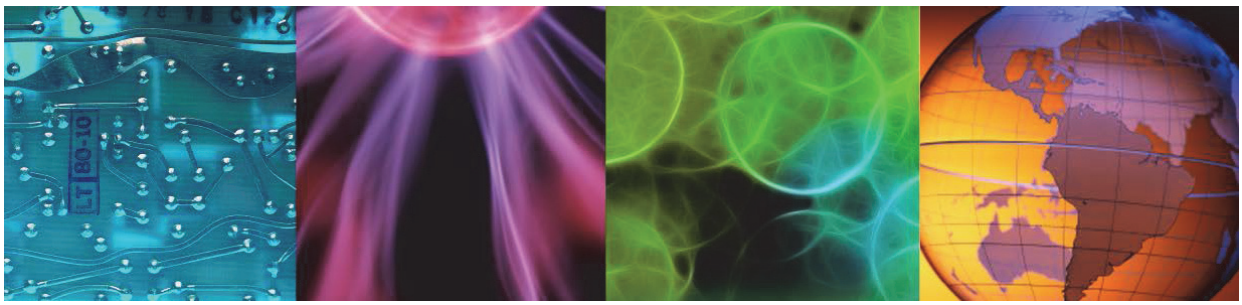
Summary

This unit taught you about ticketing and scheduling. Netcool Configuration Manager directs tickets to the appropriate groups for approval. You learned how to estimate how long a UOW might take. You also learned about scheduling *units of work* and setting time zones.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

10 UOW management

UOW management



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches how worker servers break down *units of work* into tasks and processes them. You learn how worker servers select and process tasks. Some tasks are processed in parallel with each other for efficient completion. Others are processed one after the other. This unit presents the quality checks that are part of the change management philosophy that the application supports.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Explain the difference between a unit of work and a task
- Describe concurrent and nonconcurrent tasks
- Explain which server processes tasks and in which order
- Describe how to optimize execution of tasks and steps to increase throughput
- Explain the quality checks for ensuring that changes are applied
- Describe rollback

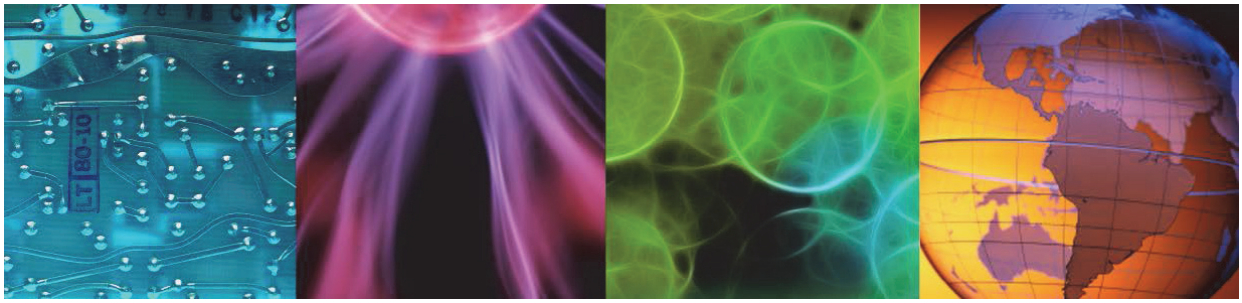
© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Units of work and tasks

Lesson 1 Units of work and tasks



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe what a task is
- Describe which *units of work* are run concurrently
- Describe execution order and priority

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Overview

- A unit of work is high-level means of tracking a change request
- A unit of work has one or more tasks associated with it
- A task consists of one major operation per device
- Examples
  - Applying a draft configuration to a device equals one task
  - Synchronizing 100 devices equals 100 tasks
  - Applying two command sets to 100 devices equals 200 tasks

© Copyright IBM Corporation 2016

Overview

The UOW is the high-level item that users track as it moves through the workflow engine. It is shown in the queue manager and other management systems reference it.

At a lower level in the application, the unit of work includes one or more tasks. The worker servers process these tasks. A task is a one major operation against one device. A major operation is an import, synchronization, configuration change, or one command set application. It is like one complete round trip to a device to perform an operation against it.

In the following examples, you can see how many tasks might be in a UOW:

- A UOW that applies a draft configuration against a single device has one task, one operation against one device.
- A UOW that synchronizes a realm of 100 devices has 100 tasks, 1 operation against 100 devices.
- A UOW that applies two command sets against 100 devices has 200 tasks, 2 operations against 100 devices.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Units of work are decomposed

- Worker servers decompose a UOW into its tasks
  - A special thread on the worker server picks up the UOW and breaks it down into tasks
- Decomposition adds important data, like where a task is processed
- Large units of work take time to decompose
- Decomposer threads determine if tasks run concurrently or nonconcurrently

© Copyright IBM Corporation 2016

Units of work are decomposed

When you submit a UOW and it is approved, if necessary, a worker server picks it up. A special thread decomposes it into tasks. Decomposition takes the UOW and creates a series of tasks in a special table in the database. Each task includes specific data that is defined in the original UOW, like execution window and execution order. It also includes data that the process generates at decomposition time. Decomposition also creates a list of worker servers that can process the task, which is based on the device location in the realm structure.

Units of work that process 1000 or more devices can take some time to decompose. The UOW sits in the *work waiting to execute* queue until it is fully decomposed and all its tasks are committed to the database.

One of the more important aspects of decomposition is determining whether the tasks can run concurrently. Concurrency determines whether a UOW is single-threaded. Single threaded processing in this context means that it processes tasks sequentially. Different threads can process tasks, but only in a specific order.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Concurrent and nonconcurrent tasks

- Some units of work run their tasks concurrently
  - All eligible worker server threads process their tasks in parallel
  - Concurrency provides efficient use of resources
  - Imports and configuration synchronization units of work
  - Single command set units of work
- Some units of work run their tasks noncurrently
  - Eligible worker server threads process nonconcurrent tasks sequentially
  - Nonconcurrent tasks provide control over error responses
  - Examples
    - Multiple command set units of work
    - Operating system (OS) upgrade units of work
    - Configuration change units of work

© Copyright IBM Corporation 2016

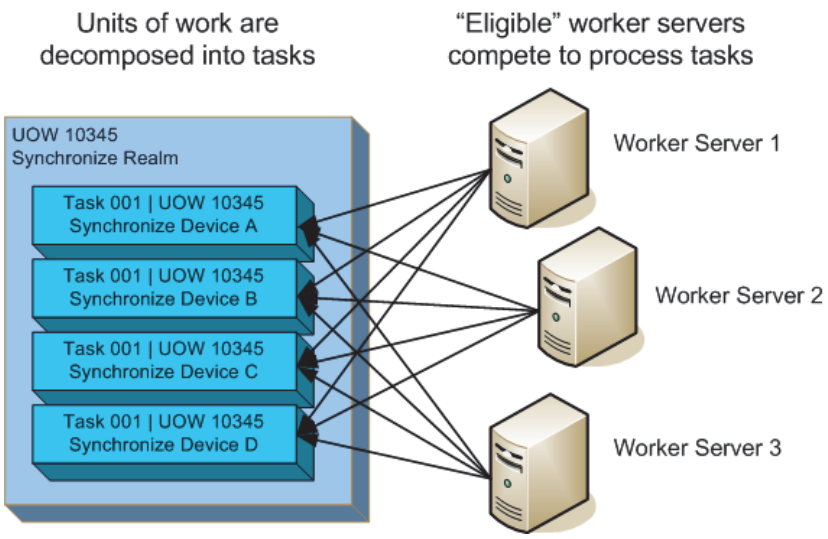
Concurrent and nonconcurrent tasks

A UOW can run concurrently or nonconcurrently. Concurrent *units of work* are multithreaded, which means that all available worker threads, which are eligible to process devices, pick up and process the tasks. This method makes efficient use of worker server resources. Use this method in configuration synchronization and import of multiple devices. Also, for a single command set applied to devices.

Nonconcurrent *units of work* are single-threaded. This wording is somewhat a misnomer in that any eligible worker server thread can pick up a task, but only one task processes at a time. Nonconcurrent processing ensures good control over error responses from individual tasks. This error control response manifests by stopping the UOW from processing if it reaches a certain level of errors. *M*ultiple command set and OS upgrade *units of work* affect this sequential processing. Configuration change *units of work* are by definition sequential because they use one task.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

How a UOW is run



© Copyright IBM Corporation 2016

How a UOW is run

After a UOW is submitted and decomposed, and its execution window opens, all the eligible workers servers compete for the tasks. It is a first-come, first-served mechanism. No round-robin distribution of tasks to worker servers exists to level the load. One worker server typically gets to the tasks first and fills up its threads before other workers can perform these tasks. Because each worker server polls at a different time, one worker server might find all the new work and fill its threads with it. Sometimes, the worker server that picks up the tasks is the one that is closest to the database server and has the shortest delay.

A nonconcurrent UOW uses references between each task. If a worker picks up a task that references a previous task, that previous task must complete before the worker can lock the new task.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Task execution order

Units of work are scheduled to execute in a maintenance window

Tasks are run first-in, first-out

SQL searches are based on the scheduled time and then on the creation time

UNIQUEKEY	CREATEDTIME	SCHEDULEDTIME
17721|task000	22-JUN-09 04.54.51.387000000 PM	26-JUN-09 10.00.00.000000000 PM
17722|task000	22-JUN-09 06.09.07.458000000 PM	26-JUN-09 10.00.00.000000000 PM
17723|task000	22-JUN-09 06.11.37.981000000 PM	26-JUN-09 10.00.00.000000000 PM
17723|task001	22-JUN-09 06.11.37.981000000 PM	26-JUN-09 10.00.00.000000000 PM
17724|task000	22-JUN-09 07.29.54.907000000 PM	26-JUN-09 10.00.00.000000000 PM
17741|task000	23-JUN-09 08.18.51.367000000 PM	26-JUN-09 10.00.00.000000000 PM
17741|task001	23-JUN-09 08.18.51.367000000 PM	26-JUN-09 10.00.00.000000000 PM
17741|task002	23-JUN-09 08.18.51.367000000 PM	26-JUN-09 10.00.00.000000000 PM
17741|task003	23-JUN-09 08.18.51.367000000 PM	26-JUN-09 10.00.00.000000000 PM
17741|task004	23-JUN-09 08.18.51.367000000 PM	26-JUN-09 10.00.00.000000000 PM
17741|task005	23-JUN-09 08.18.51.367000000 PM	26-JUN-09 10.00.00.000000000 PM
17741|task006	23-JUN-09 08.18.51.367000000 PM	26-JUN-09 10.00.00.000000000 PM
17761|task000	24-JUN-09 03.22.35.943000000 PM	26-JUN-09 10.00.00.000000000 PM
17762|task000	24-JUN-09 03.35.00.248000000 PM	26-JUN-09 10.00.00.000000000 PM
17763|task000	24-JUN-09 03.43.26.128000000 PM	26-JUN-09 10.00.00.000000000 PM
17764|task000	24-JUN-09 03.55.02.029000000 PM	26-JUN-09 10.00.00.000000000 PM
17765|task000	24-JUN-09 04.00.00.409000000 PM	26-JUN-09 10.00.00.000000000 PM
17766|task000	24-JUN-09 04.05.04.572000000 PM	26-JUN-09 10.00.00.000000000 PM

© Copyright IBM Corporation 2016

Task execution order

Typically, you do not care which tasks are picked up first. However, if *units of work* are scheduled to process during a maintenance window, the tasks process in a first-in, first-out order. In other words, if the maintenance window is on Friday night, the UOWs submitted on Monday morning are picked up before the UOWs submitted on Thursday.

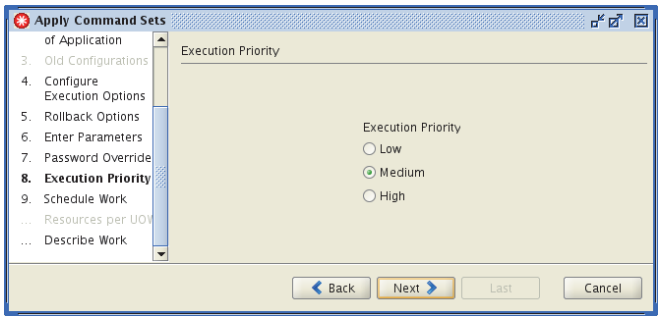
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Setting priority

Use the UOW wizard to set the priority

Default priorities are assigned to a UOW

- Low priority for imports and synchronizations
- Medium priority for configuration changes, command sets, OS upgrades



© Copyright IBM Corporation 2016

Setting priority

By using priority, you have some control over which *units of work* process first. All UOWs prompt priority and you can set it in the execution wizard. Imports and synchronizations are, by default, set to low priority. All other *units of work* are set to medium.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Impacts of priority

Types of priority:

- High priority tasks first
- Medium priority tasks after all high priority tasks are processed
- Low priority tasks after all high and medium priority tasks are processed

Relevance of priority:

- Maintenance windows
- Overutilized worker servers

UNIQUEKEY	SCHEDULEDTIME	PRIORITY
17721|task000	26-JUN-09 10.00.00.000000000 PM	3
17722|task000	26-JUN-09 10.00.00.000000000 PM	3
17723|task000	26-JUN-09 10.00.00.000000000 PM	3
17723|task001	26-JUN-09 10.00.00.000000000 PM	3
17724|task000	26-JUN-09 10.00.00.000000000 PM	3
17741|task000	26-JUN-09 10.00.00.000000000 PM	7
17741|task001	26-JUN-09 10.00.00.000000000 PM	7
17741|task002	26-JUN-09 10.00.00.000000000 PM	7
17741|task003	26-JUN-09 10.00.00.000000000 PM	7
17741|task004	26-JUN-09 10.00.00.000000000 PM	7
17741|task005	26-JUN-09 10.00.00.000000000 PM	7
17741|task006	26-JUN-09 10.00.00.000000000 PM	7
17761|task000	26-JUN-09 10.00.00.000000000 PM	1
17762|task000	26-JUN-09 10.00.00.000000000 PM	1
17763|task000	26-JUN-09 10.00.00.000000000 PM	1
17764|task000	26-JUN-09 10.00.00.000000000 PM	1
17765|task000	26-JUN-09 10.00.00.000000000 PM	1
17766|task000	26-JUN-09 10.00.00.000000000 PM	1

© Copyright IBM Corporation 2016

Impacts of priority

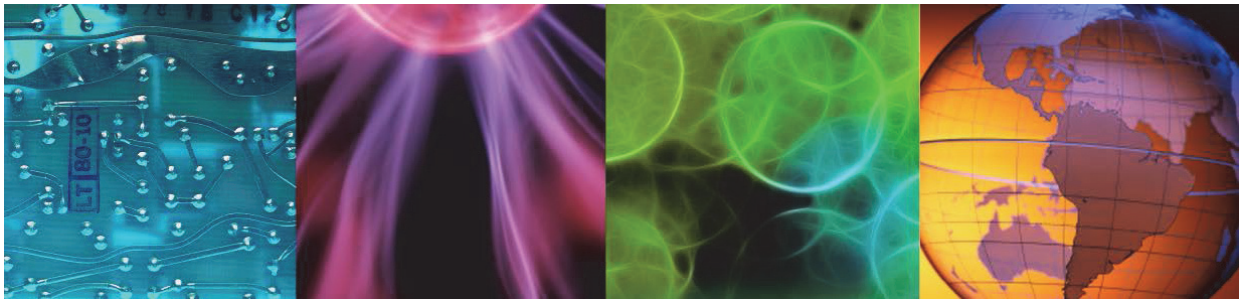
As you expect, setting priority reorders which tasks are picked up first with high priority tasks selected first.

Setting priority on *units of work* affects execution only when you overutilize worker servers. This situation arises in oversubscribed maintenance windows or when workers are fully loaded with tasks like a mass synchronization of the network.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Task processing

Lesson 2 Task processing



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

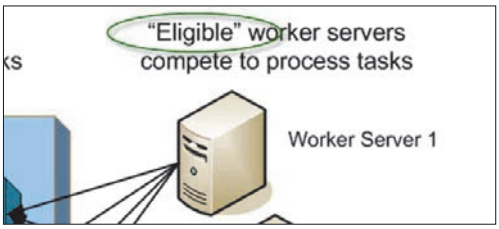
In this lesson, you learn how to perform the following tasks:

- Configure worker server resource to determine eligibility
- Describe the worker server processing checks
- Break up a UOW for better processing

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Designating work servers

Determine which worker servers process which devices



Use a realm-based worker server resource to determine eligibility

Use the following criteria to determine eligibility:

- Network access to the device: Can you route to a device
- Political or commercial boundaries: Who owns the worker server
- Nature of work being processed: How critical is the throughput

© Copyright IBM Corporation 2016

Designating work servers

You can designate which worker servers are eligible to process tasks. You use a generalized resource to determine the eligibility of workers and define the devices that a worker server can process.

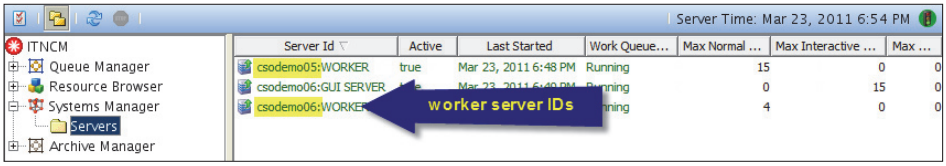
You use several factors to determine eligibility. The most important factor is network routing. The worker server must connect to the device to process it. In some network designs, that is impossible. In those cases, certain workers are excluded from processing. Other factors that determine eligibility arise from performance boundaries or company policies.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Worker server resource

A worker server resource determines which workers can execute a task on a device: Devices resolve them, like RAD or authentication resources

Worker servers were assigned names during installation: These names are listed in the Systems Manager, which is a group security setting



Worker server IDs are referenced in a worker server resource

© Copyright IBM Corporation 2016

Worker server resource

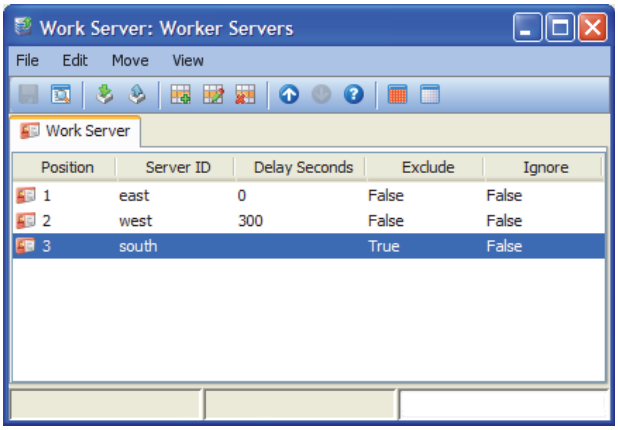
Use the worker server resource to determine worker server eligibility. It resolves in the same manner as most other generalized resources, that is, based on its realm location and VTMOS criteria. The decomposer thread that creates tasks performs an upward search from the device that it plans to process, looking for a worker server resource. When it locates the worker server resource, it uses the data inside that resource to note which workers are eligible to process.

The decomposer reviews the worker server resource and the worker server IDs configured inside the document. Those worker server IDs are defined at installation time. You can determine these names by looking for them in the systems manager.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring worker server resources

Worker server resources are placed in a realm and associated to a VTMOS, typically */*/*/*



© Copyright IBM Corporation 2016

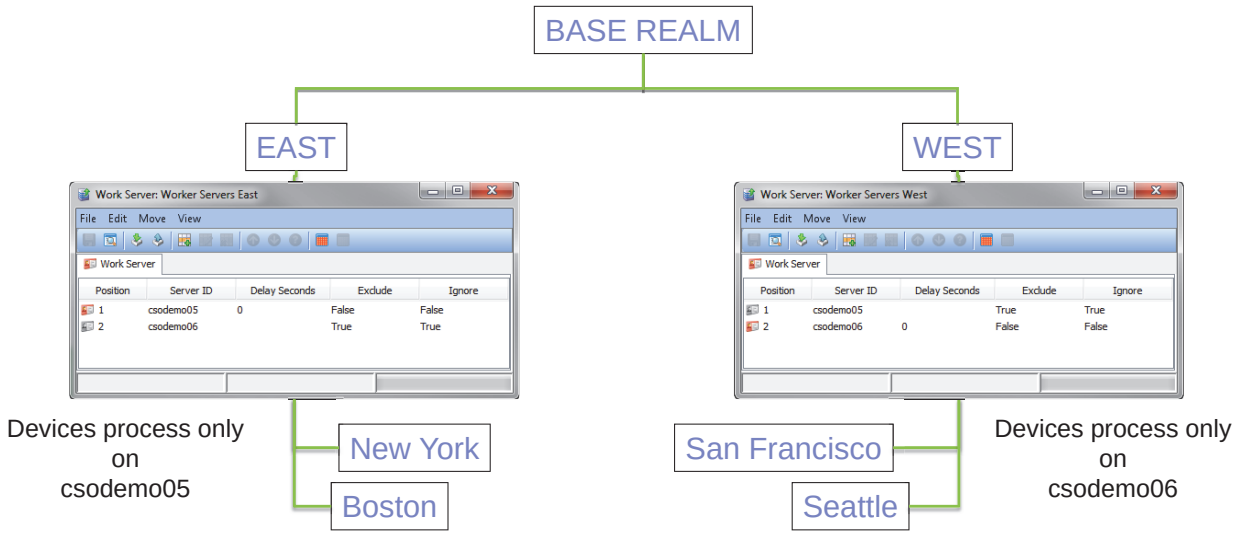
Configuring worker server resources

You must create a worker server resource in the implementation before a UOW can process. At a minimum, you need a worker server resource with a VTMOS of */*/*/* in the base realm. These columns are available on the Worker Server tab:

- **Server ID**: Determine the worker server IDs and define them in the worker server resource. Define all server IDs in the resource and then determine whether they can process the task. Defining all the worker server IDs makes it clearer which servers are allowed or disallowed from processing tasks.
- **Delay Seconds**: Determine the amount of time that the worker server must wait after the start window opens before it can pick up the approved task. Delay ensures worker servers act like backup servers. For example, the **west** server must wait 300 seconds after the window start opens before it can process the task. In that 5-minute wait period, the task might process the **east** server. However, if it is too busy with other tasks or inoperable, the **west** server processes the task after that wait.
- **Exclude (or Ignore)**: Set either one of these values to True to prevent a server ID from ever picking up a task. For example, the south server never processes devices that resolve this resource.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Worker server resource resolution example



© Copyright IBM Corporation 2016

Worker server resource resolution example

This slide illustrates an example of worker server resources that are defined in two separate realms. In this situation, you have *east* and *west* realms with devices organized into major metropolitan areas. This organization creates corporate political boundaries between the worker servers. One server processes devices on the east coast and another server processes devices for the west coast.

When you submit a unit of work and it enters the *work waiting to execute* queue, the decomposer thread breaks the UOW into its subsequent tasks. When creating the task, the decomposer looks at the name of the device and where it is in the realm structure. It then does an upward search and looks for a worker server resource. If the device is in the Boston subrealm, it searches and finds the Worker Servers East worker server resource. Using the information in that document, it tags the task with the csodemo05 worker server ID. Since the csodemo06 server ID is excluded, it is not added to the list of eligible workers that can process the task. This lookup occurs for each individual task. If you include a device in the *west* realm in the UOW, the search finds the other worker server resource and adds the csodemo06 server to the task.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Simultaneous worker server tasks

Define a maximum number of tasks per server

- Workers have a max normal pool size
- Pool size maximum is based on size of configurations

Self-tune the number of threads that are released at one time

- Monitor total memory use and estimate the amount of memory required for processing configurations
- Attempts to manage the available worker server memory
- Worker might never reach *max normal pool size*
- Attempts to avoid the Java *out of memory* error

© Copyright IBM Corporation 2016

Simultaneous worker server tasks

You can configure worker servers to process multiple tasks. By default, the worker server is configured with a *max normal pool size* of 5. You can configure the *max pool size* setting in the systems manager.

This setting means that the worker server can process five tasks simultaneously. Converting CLI to XML is a memory-intensive operation, which you must consider when processing tasks. The size of the configuration is the attribute that determines how much memory it needs. Setting this value to 20 is normal for most implementations.

To help manage the amount of memory that task processing uses, the workers use a memory manager to estimate how much memory is needed to process a task. It does not start a task unless the Java heap contains that much free memory. Using this feature, while processing tasks on a worker server, it might never reach the *max normal pool size*. The key to this feature is to prevent the Java out of memory error.

If a Java out of memory error is recorded on a worker server, or on a presentation server, you must restart that particular application server.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Workers and tasks

Workers poll the database looking for tasks that they can process

- Workers poll every 5 seconds
- Each worker retrieves all the tasks it can process and can begin execution

Each worker performs the following analysis:

- Determines if a task is locked by another worker
- Determines if task dependencies exist
- Determines if a device is locked by another task
- Locks a task and hands it off to a thread to begin processing it

© Copyright IBM Corporation 2016

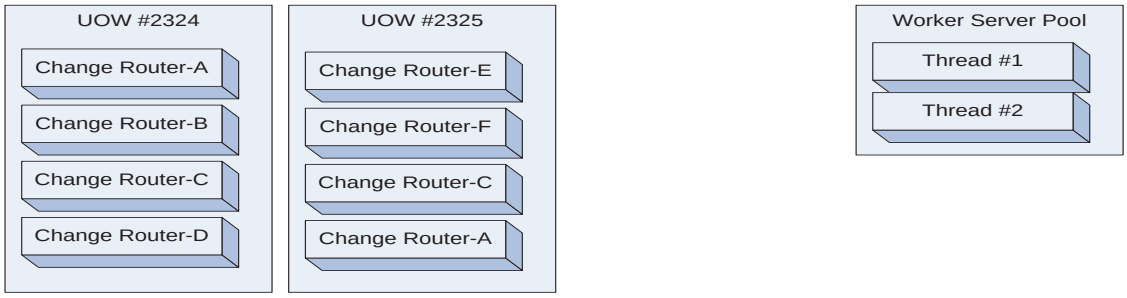
Workers and tasks

Worker servers pick up tasks by regularly polling the database for the decomposed tasks that a worker server decomposer places there. Each worker server polls the database every 5 seconds. It retrieves all the incomplete tasks that it is eligible to process and can begin processing. It applies these rules to the tasks:

1. Determines whether another worker server locks the task. If it is, it drops the task.
2. Determines whether any task dependencies exist. If it depends on another task and that is not complete, it drops the task.
3. Determines whether another task locked the device. If it is, it drops the task.
4. If it passes these criteria, it locks the task and the device that is associated with the task.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Worker server process scenario



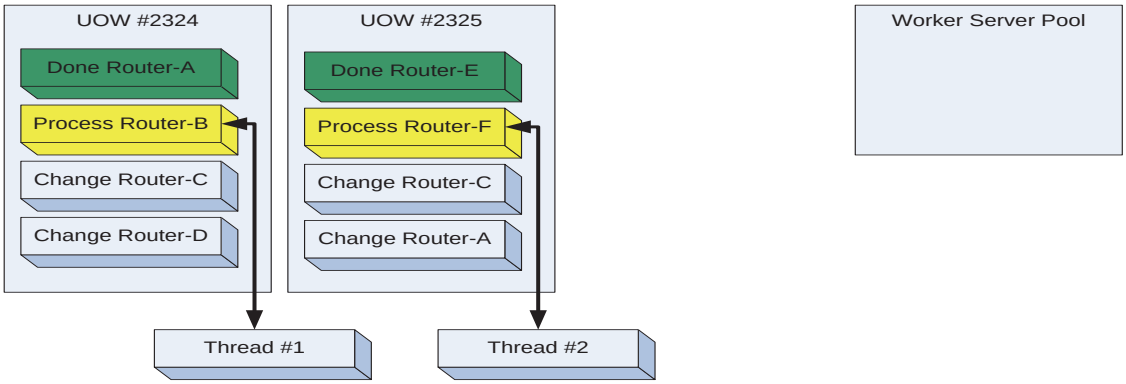
© Copyright IBM Corporation 2016

Worker server process scenario

Worker server threads are constantly looking for tasks to process. The following scenario might help you understand their behavior. In one worker server, the *max normal pool size defines two threads.* Two units of work are ready to process nonconcurrent changes on a series of devices.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Threads process two nonconcurrent units of work



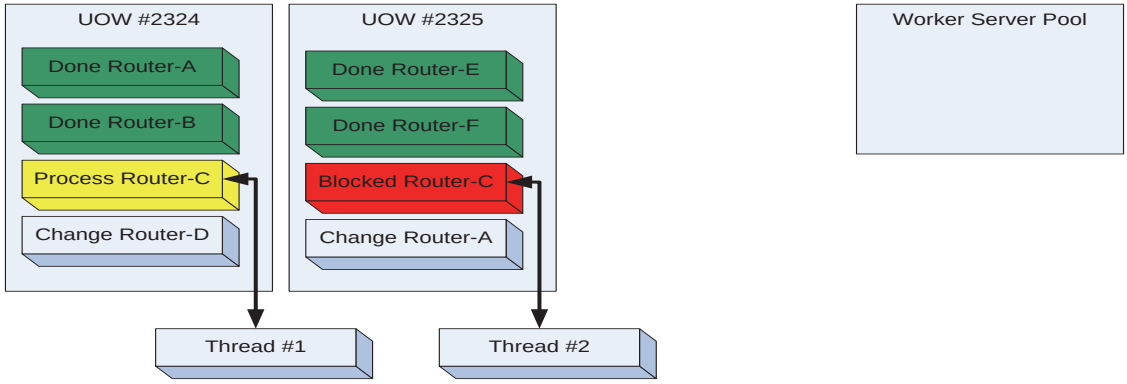
© Copyright IBM Corporation 2016

Threads process two nonconcurrent units of work

The threads begin processing the UOWs. Because these UOWs are nonconcurrent, they step through the tasks in a sequential order. In other words, two threads cannot process two tasks simultaneously in a single UOW. Each thread processes its device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Device is locked by a thread



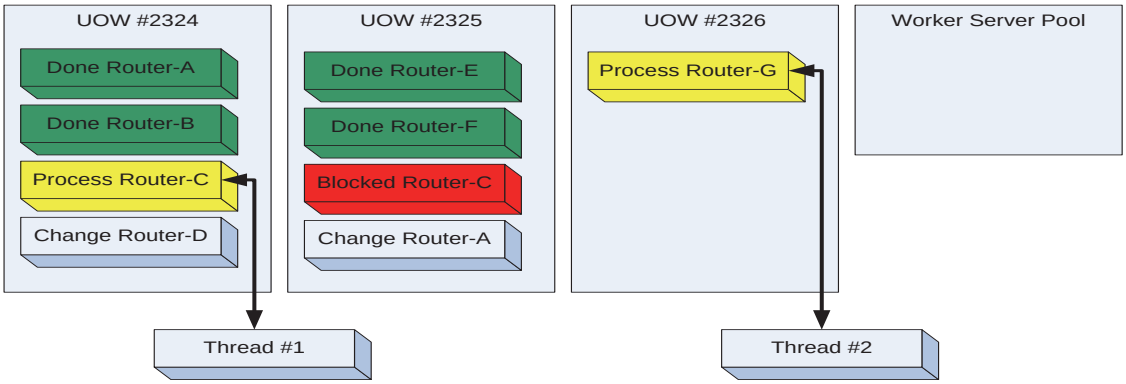
© Copyright IBM Corporation 2016

Device is locked by a thread

The *units of work* have a shared device in the same sequential location. Thread #1 picks up router-C and begins processing it. Router-C is locked and blocks thread #2 from processing its task. This UOW no longer moves forward until the lock is removed.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Thread looks for other work to do



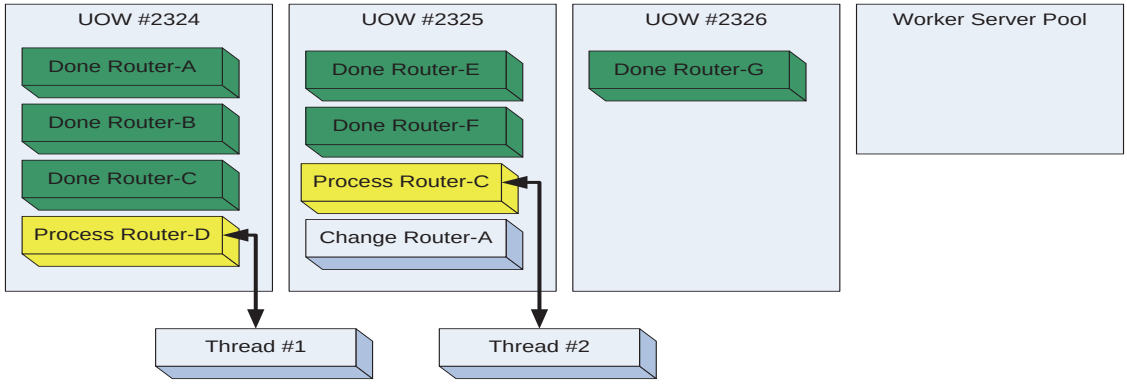
© Copyright IBM Corporation 2016

Thread looks for other work to do

Thread #2 does not wait for router-C to unlock but looks for other work to process. It picks up another UOW and processes its task. It looks like the application is actively processing three *units of work* when only two threads exist. In actuality, it is processing only two active tasks.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Thread returns to the previous UOW



© Copyright IBM Corporation 2016

Thread returns to the previous UOW

When thread #2 finishes the request for router-G, it returns to the previous UOW and resumes processing the task. Router-C must be unlocked for it to begin processing the next task in UOW 2325.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Improving performance for nonconcurrent units of work

Nonconcurrent units of work are single-threaded

A multiple command set UOW with many devices can take hours

You can split a large nonconcurrent UOW into multiple, smaller units of work

OS upgrades cannot be automatically split into multiple units of work

© Copyright IBM Corporation 2016

Improving performance for nonconcurrent units of work

As mentioned earlier, nonconcurrent *units of work* are single-threaded. Devices inside an OS upgrade or multiple command set UOW are processed one task at a time. While this technique ensures that users have control over execution, it slows down processing. A large command set execution with 100 devices and 2 command sets might take over 2 hours to complete:

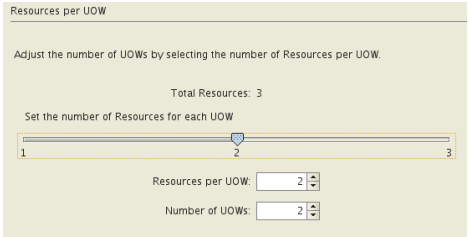
```
100 devices x 2 command sets / device x 1 minute/command set = 200 minutes
```

You can split multiple command set *units of work* into multiple *units of work*. Each UOW is still single-threaded, but the application can process multiple UOWs simultaneously. Because the operating system upgrades for a device can incur risk, you cannot split an OS upgrade UOW.

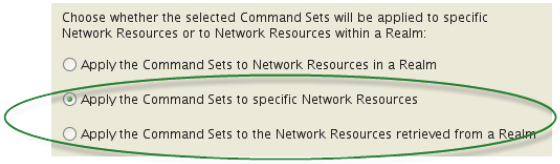
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Splitting a unit of work

- You can use the *command set* wizard to break up a UOW



- The scope of the application determines whether you can break up a UOW



© Copyright IBM Corporation 2016

Splitting a unit of work

You can use the execution wizard to split multiple UOWs. The system determines whether splitting is possible, depending on the scope of the application. You must provide the exact name and number of devices to split up.

If you select **Apply the Command Sets to specific Network Resources** or **Apply the Command Sets to the Network Resources retrieved from a Realm**, the number of devices is known. However, if you select **Apply the Command Set to Network Resources in a Realm**, the number is not known until execution time, which might be a week later. If the number of devices is not known immediately, the execution wizard does not let you split the *unit of work.*

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Reasons for splitting a UOW

You might have good reasons for breaking up a UOW

- Overall change is completed faster
- The scheduled end time is more reliable
- You have better visibility of devices in the queue manager

UOW ID	Type	Resources	Request Type	Started At	Submitter	Finished At
229	UOW	AA-01-ROUTER-3640	Native Command Set	Feb 3, 2011 11:10 AM	admin	Feb 3, 2011 11:12 AM
250	UOW	Multiple Devices	Configuration Synchronization ...	Mar 16, 2011 11:32 AM	admin	Mar 16, 2011 11:33 A

You have good reasons for not breaking up a UOW

- Failure options stop the UOW from processing
- Easier to dequeue and requeue one UOW
- Approving change request requires approval of all units of work

© Copyright IBM Corporation 2016

Reasons for splitting a UOW

A number of reasons exist why you should or should not break up a multiple command set's unit of work. Breaking up the UOW has the following advantages:

- Since a multiple command set UOW is single threading, breaking it up into many *units of work* allows the work to multi-thread and allows the work to complete faster.
- If you break up the command set into many *units of work*, especially when you have only one device per UOW, the schedule end time for *units of work* is more reliable. Remember that a UOW does not stop its execution when it reaches the scheduled end. A large UOW can run long past that window. If each UOW has only one device, its execution is much shorter. Even if a UOW is picked up just before the scheduled end time, it runs only for a short time after that window since it is processing a few command sets against one device.
- If you break up the UOW with only one device per UOW, the device name is shown in the queue manager. Seeing the name in the GUI makes it easier to debug issues when a failure of the change occurs.

Reasons also exist why you might not want to break up a UOW.

- When all the tasks are in one UOW and it is single threaded, it is easy to stop the execution of the work when it encounters a failure. Organizations with a low tolerance for risk, might prefer this option.
- It is easier to find and select a single UOW to either stop or restart it than searching for all the *units of work* that are created when it is split up
- It is also easier for an approver to select and approve one ticket as opposed to all the tickets that are generated when the UOW is split up into many *units of work*.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Unit of work performance

Number of tasks per hour

Variables that affect the performance of UOW tasks

- Configuration size
- Server central processing unit (CPU) speed
- Operating system working with the Java virtual machine (JVM)
- Database read and write speeds
- Worker server location
- Device performance

© Copyright IBM Corporation 2016

Unit of work performance

You can process thousands of changes per hour when you have the correct number of worker servers and the correct database processing speeds. However, overall performance is difficult to quantify.

The biggest factor of UOW and task performance is the size of the device configuration. The amount of time to convert a 500-line configuration to XML is shorter than 50,000 lines.

The processor speed of the worker server typically determines the time that it takes to convert CLI to XML. Having multiple, high clock rate processors is more important than the amount of memory that is on the worker servers. You can convert CLI to XML twice as fast using x86 processors as you can with RISC processors.

The amount of data that moves between the workers and the database is another issue. You must ensure that you have good bandwidth and that the database processor accomplishes reads and writes efficiently. This bandwidth is the reason that you collocate worker servers with the database. The amount of data that moves between the workers and the database is at least three times larger than that moved between the worker and the device.

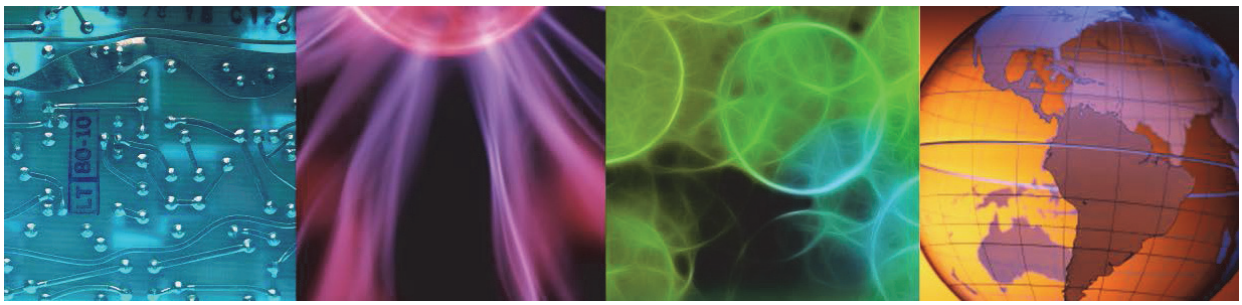
The performance of the device itself is also important. Line speeds and device response times for various configuration management tasks are often the slowest portions of the task processing.

While these values can be hard to extract from UOW timing logs, they are issues that the application cannot control.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Change management quality

Lesson 3 Change management quality



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe change management philosophies
- Describe how you implement change management philosophies in the application

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuration change management

Configuration change management is a philosophy

- You must define, analyze, and approve changes before you enter them into a system
- You must keep a record of how changes enter the system
- You must use version control to restore the state of a system

You can use Tivoli Netcool Configuration Manager to implement that philosophy

- You can use the configuration editor and command sets to define changes to stage and review
- You can use the approval policy for orderly release of changes to the network with an auditable trail of information
- All device states are available for restoration, if needed

© Copyright IBM Corporation 2016

Configuration change management

Configuration management is a philosophy that reflects itself in the processes that the organization uses. This philosophy is the same regardless of the system you are managing. You must define, analyze, and approve changes to a system, the network, before you make them physically. An audit trail must exist to track how the system arrives at its current state. Being able to restore to a previous state is also a requirement for any change management solution.

You can manage the state of the network and the devices in that infrastructure by using a configuration editor. Using this editor ensures that you define, stage, and review changes before you apply them to the network. You can implement approval policies that prevent changes from entering the network without approvals. You can audit and track all interactions with devices. Finally, you can restore all device states whenever necessary.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Changing a configuration

Are there any conflicts?

Does the device meet certain requirements?

Where there any errors?

Can you restore the device to its original state?

Has the application captured the new state?

© Copyright IBM Corporation 2016

Changing a configuration

When a change is to be applied to the system, in this case a network device, can you answer the following questions.

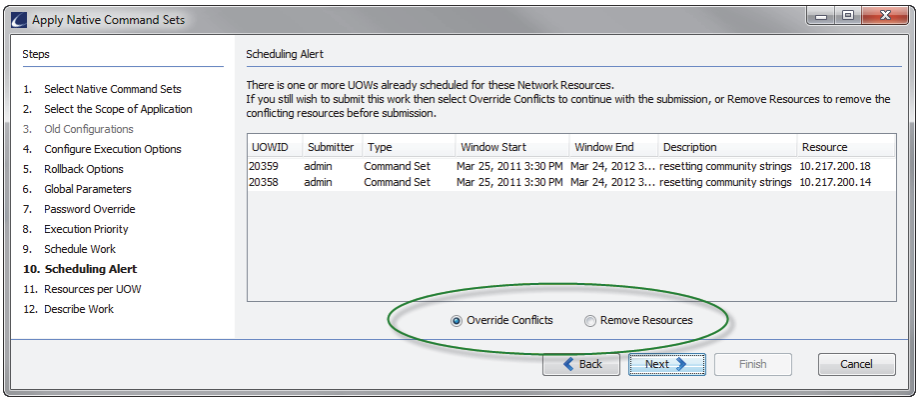
- Does this change impact another possible change that you are entering into the system, or in this case, the network device?
- Does the object that you are changing meet the requirements so that the change you apply does not cause unintended side effects?
- When you apply the change to the system, does it generate any obvious errors?
- Can you restore the object to its original state when you experience any errors?
- Did you capture the data about the object and update all references to reflect the new state after the change is complete?

In the following slides, you see how Netcool Configuration Manager addresses each of these issues when it applies changes to devices.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Scheduling alert

- You submit a change task to a device
- A warning is shown, indicating that other units of work are scheduled for the same device



© Copyright IBM Corporation 2016

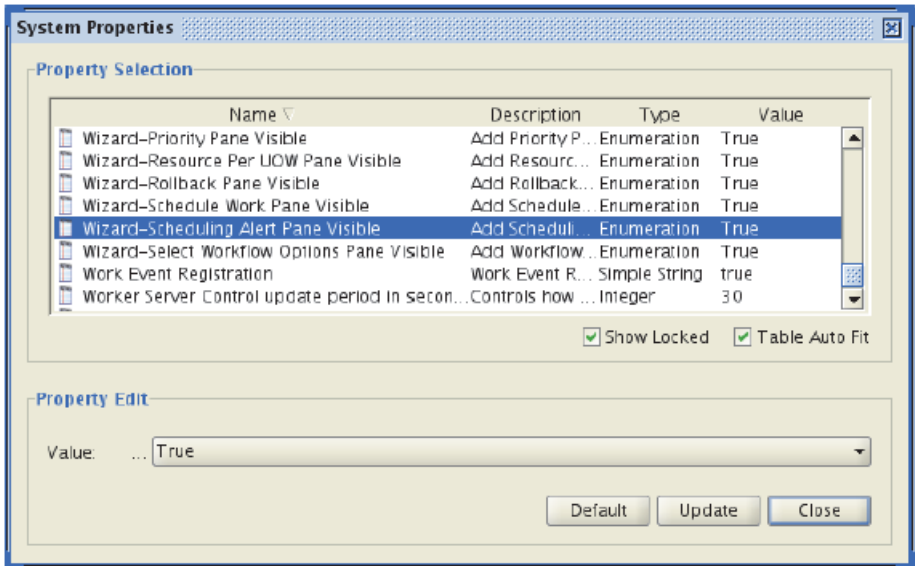
Scheduling alert

When submitting a unit of work, the application can check for any conflicting UOWs. The system checks the *work pending approval* and *work waiting to execute* queue for any conflicting devices.

You must know when another UOW in the system might change the configuration of a shared device. You might not know the nature of the configuration change. If the other unit of work is attempting to change the same area of the configuration that you are changing, you might receive unexpected results. It is worth your time to determine the nature of that change and make sure that it does not conflict with yours.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The Wizard-Scheduling Alert is not enabled by default. To enable this warning, the system administrator must enable it by using the System Properties manager. Select the **Wizard-Scheduling Alert Pane Visible** and set its value to true.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preparing for a device change

Software makes quality checks before applying a change

- Compares current device OS level with OS level in database
- Compares current device configuration with configuration in database

If comparisons pass, the change is applied

You might disable these checks to improve performance

© Copyright IBM Corporation 2016

Preparing for a device change

Before you apply changes to devices, verify that you are not overriding the changes that someone else made. Other experts might be working outside of the solution and modifying devices. The default behavior of the product is to check for these out-of-band issues and cancel its changes when it finds them. Two checks are made by default before you apply a change:

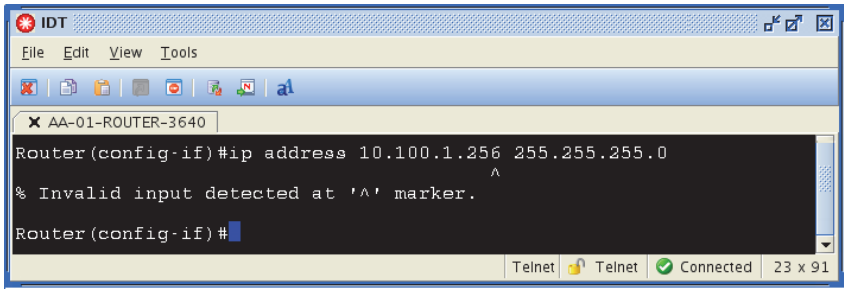
- The application requests version and model information from the device and compares it with what is in the database. This check is to verify that it is the same type of device since the previous interaction. Differences in these values cause the UOW to fail.
- The application pulls and caches the running configuration from the device. The workflow compares that configuration with what is in the database. If differences are found, the application assumes that an out-of-band change occurred and that the new change can overwrite it. If any differences between these configurations are found, the UOW fails and records an *out of synch* error state.

You can disable this check to improve speed. However, disabling this check defeats the purpose of implementing configuration management.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Ensuring that no errors are returned

Worker servers monitor for any errors as changes are applied



A list of error messages is included in the device scripts

© Copyright IBM Corporation 2016

Ensuring that no errors are returned

As the change is applied to the device, the worker server monitors for any messages that the device produces. It checks those messages against a list of error states that are defined in the device scripts. If any error message is captured, the UOW fails and the device rolls back to its original state. You can extend this list of error states by using the resource access document or the scripting resource.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Restoring to a previous state

Rollback is attempted to return the device to the original state

- Running configuration is cached before attempting change
- Changed configuration is checked against cached configuration
- Any changes are intelligently removed

You might disable rollback to improve speed

© Copyright IBM Corporation 2016

Restoring to a previous state

If errors are generated when a change is applied to a device, the worker server attempts to restore the device to its previous state. This restoration is accomplished by using these steps:

1. It pulls and caches the running configuration before it attempts the change.
2. It pulls the running configuration from the device after the change fails.
3. It compares the two running configurations for any differences. If it finds any differences, because of a partial change, it adds or removes commands to get back to the original running configuration. This process requires the device to be modeled with a SmartModel driver.

You can disable rollback to improve speed. However, disabling this check defeats the purpose of implementing configuration management.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Retrieving the configuration

Collect the resulting running configuration

Successful changes might incur unexpected results

- Data might be reformatted by the operating system
- New commands might appear if a feature is enabled

When applicable, the new configuration is stored to disk

You can disable the check to improve speed

Creates a *stale configuration* state on the device

© Copyright IBM Corporation 2016

Retrieving the configuration

If the worker server captures no errors when it applies the change, it determines that the change is successful. It pulls the running configuration one last time, converts it to XML, and saves both the CLI and XML version in the database as the new current configuration.

Sometimes, the successful application of changes leads to new commands in the configuration. Other times, the operating system interprets the changes and it reformats them into new syntax. This reformatting can happen with the different ways that you can write a Cisco permit statement.

Cisco IOS devices save the running configuration to the startup configuration. Any other device that uses the concept of storing the configuration to disk applies this step.

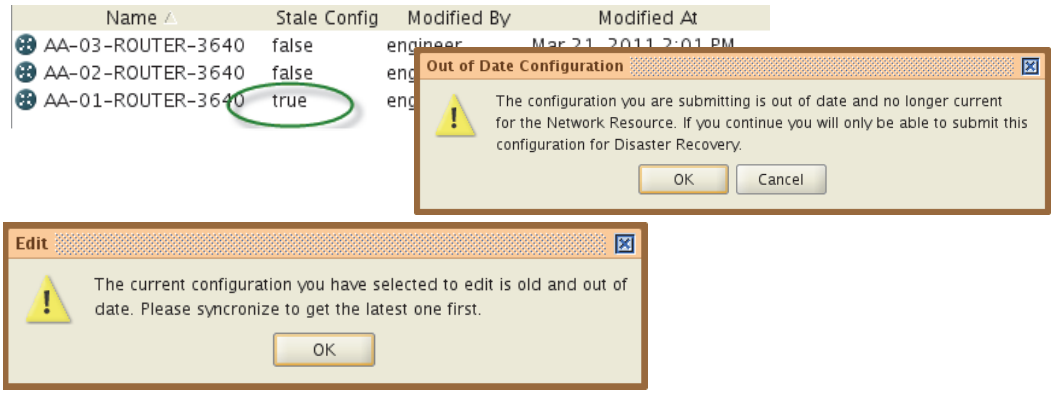
You can disable this behavior, but disabling this step defeats the purpose of configuration management and leads to a *stale* device. A *stale* device occurs when a device is modified, and the resulting configuration is not retrieved.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Stale configuration state

Certain functions do not work on a device with a stale configuration

You can override this behavior



© Copyright IBM Corporation 2016

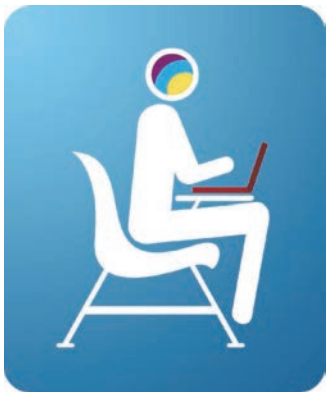
Stale configuration state

When Netcool Configuration Manager is configured to not retrieve a changed configuration, it is aware that the device has a new state that is not reflected in the database. When this situation occurs, the application marks the device as having a *stale configuration*. A *stale configuration* state does not allow another change to that device until it is synchronized.

As with the previous checks, you can also override this behavior. Overriding this behavior truly defeats the configuration management protections that the software enables and it becomes a simple provisioning system.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises



© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Explain the difference between a unit of work and a task
- Describe concurrent and nonconcurrent tasks
- Explain which server processes tasks and in which order
- Describe how to optimize execution of tasks and steps to increase throughput
- Explain the quality checks for ensuring that changes are applied
- Describe rollback

© Copyright IBM Corporation 2016

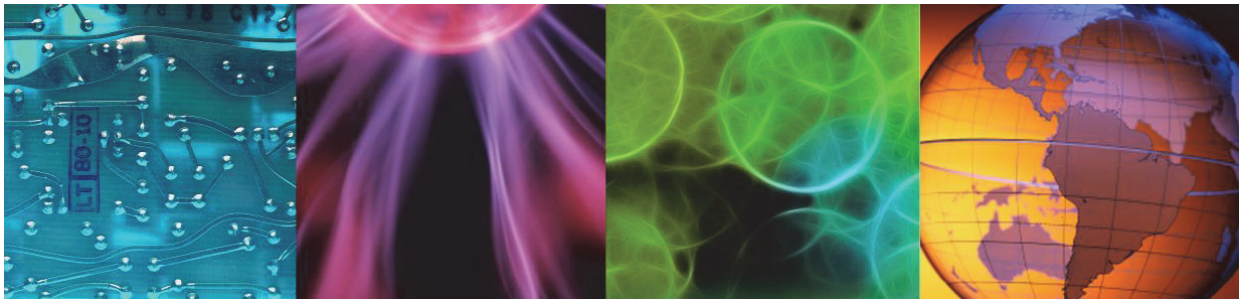
Summary

This unit showed how units of work are broken down into tasks and processed by worker servers. You learned how worker servers select and process tasks. Some tasks are processed in parallel with each other for efficient completion; others are processed one after the other. This unit also presented the quality checks that are part of the change management philosophy that the application supports.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

11 IBM device terminal

IBM device terminal



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit presents the IBM Device Terminal (IDT), which an authorized user can make direct VTY connections to the devices under management. You can filter the input for these connections to intercept restricted commands and collect keystroke logs to record all input.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Use the IBM device terminal and connect it to a device
- Create, apply, and test a command filter for a device terminal
- Describe the administration function of a device terminal

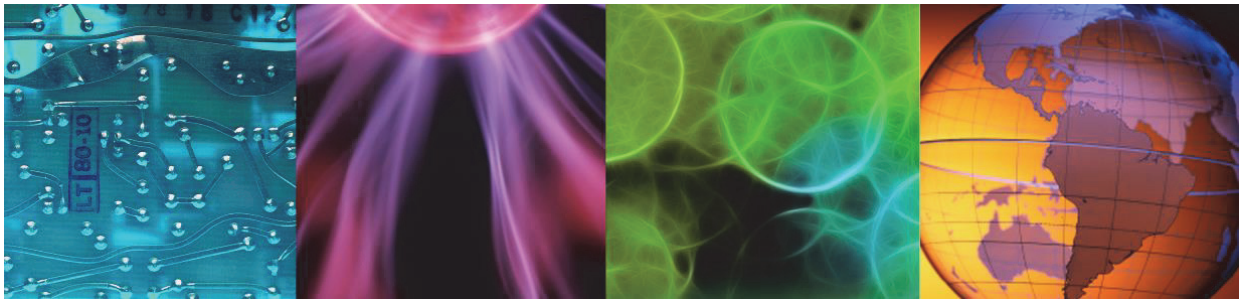
© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Device terminal overview

Lesson 1 Device terminal overview



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the features of the device terminal
- Describe how connections are made to devices
- Configure the tool to synchronize automatically
- Review keystroke logs of device terminal sessions

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IBM Device Terminal (IDT)

You can have direct access to command-line interfaces

Uses predictable paths to connect to devices

All interaction with a device includes keystroke logging

You can configure to use either automatic or user credentials for device login

© Copyright IBM Corporation 2016

IBM Device Terminal (IDT)

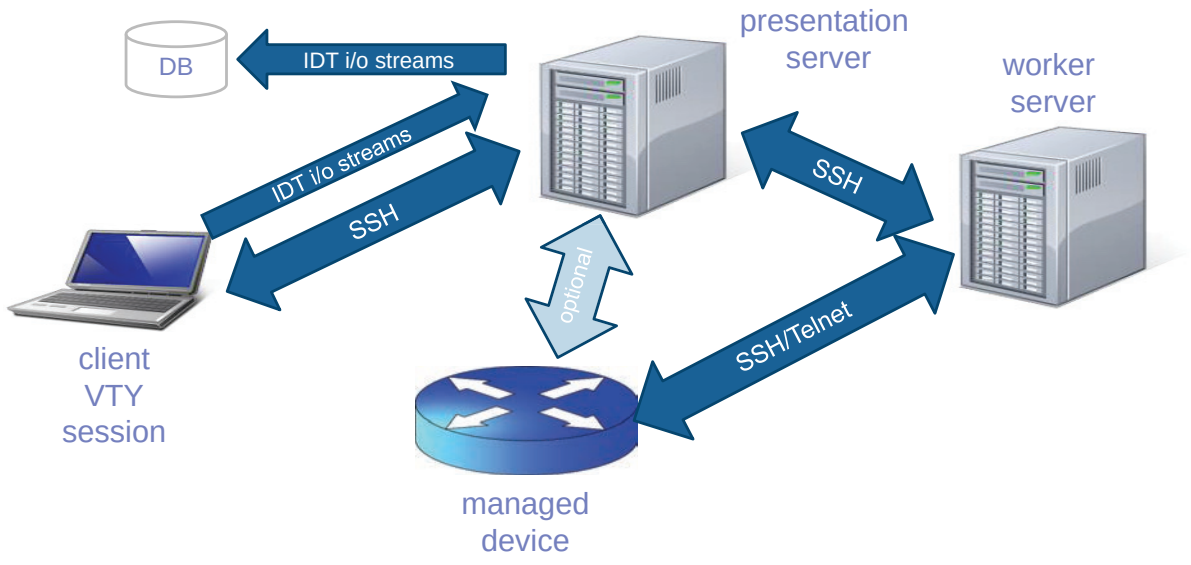
The IBM device terminal, IDT or device terminal, authorizes direct access to the command-line interface of a device through a virtual terminal (VTY.) The resource access document that the device uses to connect to process tasks determines the connection to the device.

The device terminal uses predictable paths of connection to the device. It uses the product infrastructure to create secure tunnels through the network that eventually lead to the command-line interface. Any interaction with the device terminal is logged and available for future audits.

Authentication to the device uses product credentials for automatic login and hides the device passwords from users or uses manual authentication, where users must provide their own credentials.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IDT architecture



© Copyright IBM Corporation 2016

IDT architecture

The device terminal architecture is designed to run on worker servers. You can configure some or all workers to run the device terminal process. The presentation server always has a worker server process that runs on that physical server, whether it is configured to process *units of work*.

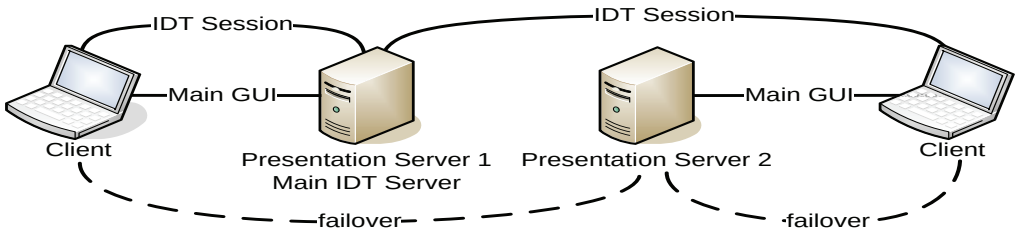
When a user initiates a device terminal session from the GUI, it creates a secure tunnel between the GUI and the presentation server. This SSH session is set up on a standardized port that is defined at installation. This port by default is 8104. IDT also uses the GUI port, by default, 16310, to send a copy of input and output streams to the presentation server for logging.

The presentation server can connect directly to the device, the optional arrow in the slide, or through another worker server. The key to this decision is the ability of the presentation server to route to the device. Often times, the worker server is the only server that has access to the network device, and routing from the presentation server fails. If the presentation server connects to a worker server, IDT uses the worker server resource to determine which worker can reach the device. Regardless of the connections, copies of the input and output streams are sent to the presentation server over port 16310 and saved to the database. These keystrokes and their responses are sent to the presentation server as clear text over port 7001 to the presentation server. While it is impossible to hijack the IDT session made to the device, the keystrokes are sniffable on port 16310. If this situation is a security violation, you want to use HTTPS for client connections to the presentation server.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Main IDT daemon server

All IDT sessions are initially directed to a main IDT daemon server



Main IDT daemon server is identified during installation

```
----------
Is this the main IDT Daemon Server ?

 ->1- Yes
   2- No

ENTER THE NUMBER OF THE DESIRED CHOICE, OR PRESS <ENTER> TO ACCEPT THE
   DEFAULT: 1
```

© Copyright IBM Corporation 2016

Main IDT daemon server

The IDT architecture also includes the concept of a main IDT daemon server. This feature is designed to initially direct IDT sessions to a specific presentation server. This feature makes sense only if there is more than one presentation server in the implementation. All IDT connections initially attempt to connect to this main IDT server. The IDT attempts access with another presentation server when the initial connection fails.

When installing presentation servers, it prompts the system administrator whether the installed server is the main IDT server. Selecting **yes** overwrites any other main IDT server daemon that is installed.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IDT controlled access

Device terminal employs the same security model as other functions

- The use of the device terminal is a group authorization
- If you cannot see the device, you cannot log in to it

You can incorporate a device terminal into a network design

- Device ACLs or network firewalls can filter unauthorized connections

© Copyright IBM Corporation 2016

IDT controlled access

One advantage of using IDT over other VTY clients is that IDT aligns with the configured security policies of controlling access to network devices. First, the use of the device terminal is an activity that you must add to the group. Second, if the device is not visible in the GUI, the user cannot connect with it even if authorized to use the tool.

Another advantage is augmenting Network security. The network design can include device ACLs and firewalls so that connections to devices can come only from worker servers. You can modify devices through a UOW task or an IDT session.

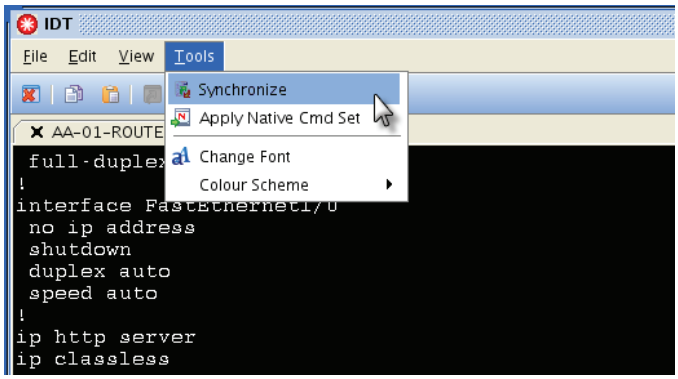
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IDT synchronized changes

You can synchronize devices from a device terminal

You can automate the synchronization process

- Synchronize every time a connection is made
- Synchronize when certain commands are seen in input stream



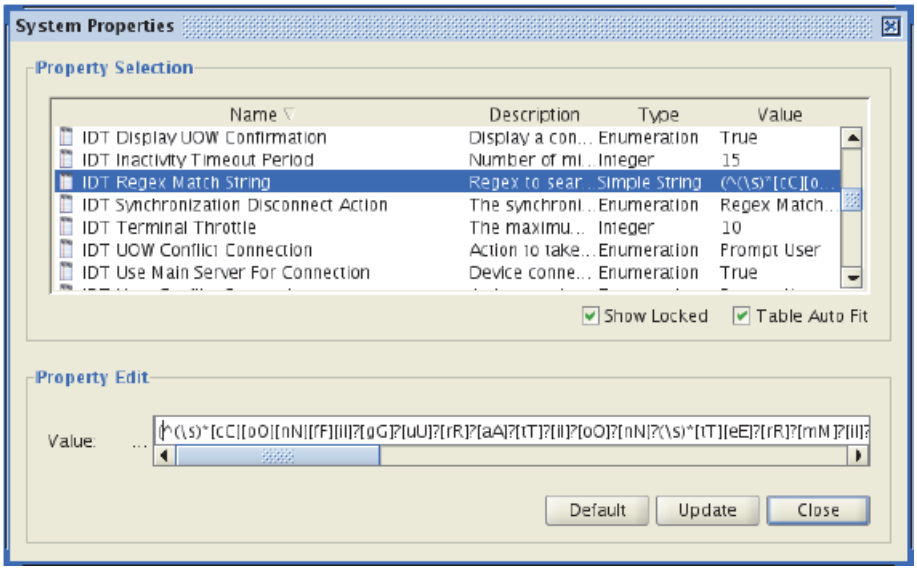
© Copyright IBM Corporation 2016

IDT synchronized changes

You can synchronize the device from an IDT session. This feature ensures that any changes you make to the device are captured in the configuration database.

You can automate the synchronization process. When specific keywords, like *configure, commit, copy, erase,* and *write,* are seen in the input stream to the device, the software triggers an automatic synchronization.

You can extend the list of keywords to include other vendors. The regular expression to use to search for these keywords is accessible through the systems manager.

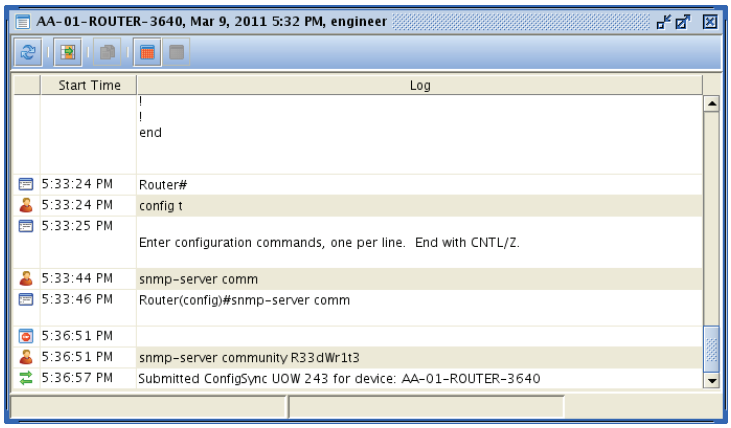


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IDT keystroke logging

Captures both the keystrokes and device output

Keystroke logging is not limited to managed network devices



© Copyright IBM Corporation 2016

IDT keystroke logging

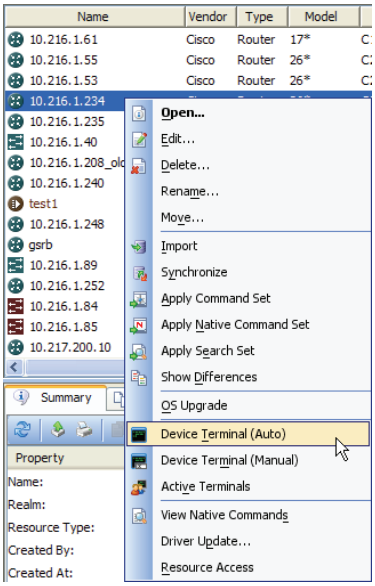
Another advantage of IDT is keystroke logging. The keystrokes that the user submits and the responses from the device are captured. This feature is not limited to network devices under management. You can use IDT to manage other host name or IP addresses that are accessible to the worker servers. You can create placeholders in the database for *unknown* devices. Unknown devices do not have configurations that are associated with them, but they can have keystroke logs maintained.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Access to a device terminal

Device terminal access is granted to a group, the option might be unavailable

You select and right-click a device; then, choose Device Terminal (Auto) or Device Terminal (Manual)



© Copyright IBM Corporation 2016

Access to a device terminal

To access to the IDT feature, you right-click a device and select Device Terminal (Auto) or Device Terminal (Manual). You provide group access to IDT by granting device terminal activities to the group.

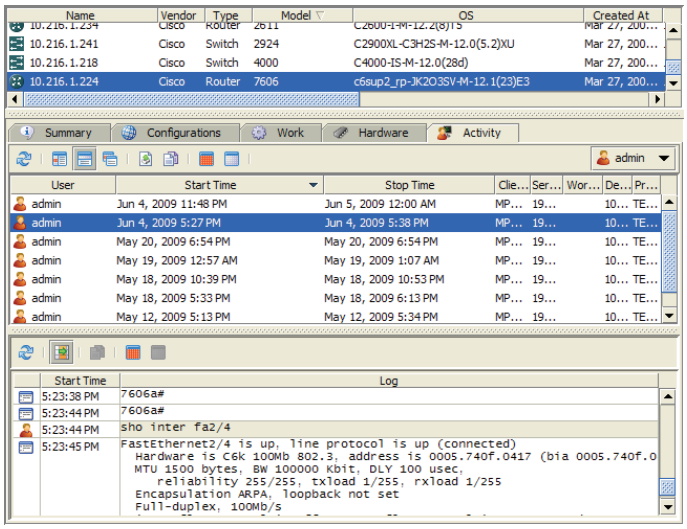
Device Terminal (Auto): IDT automatically accesses the device through the credentials that it uses for normal task processing. It uses the passwords in an authentication resource. Users do not see these passwords unless they can modify the authentication resource. This feature offers a layer of security. However, in other management solutions, for example, TACACS logs, the administrator sees the connection by the application user name, and not by the person who accesses the system.

Device Terminal (Manual): In IDT, you see the login screen to the device. You must provide credentials to log in.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

IDT logging

You can see only your previous interactions with a device



© Copyright IBM Corporation 2016

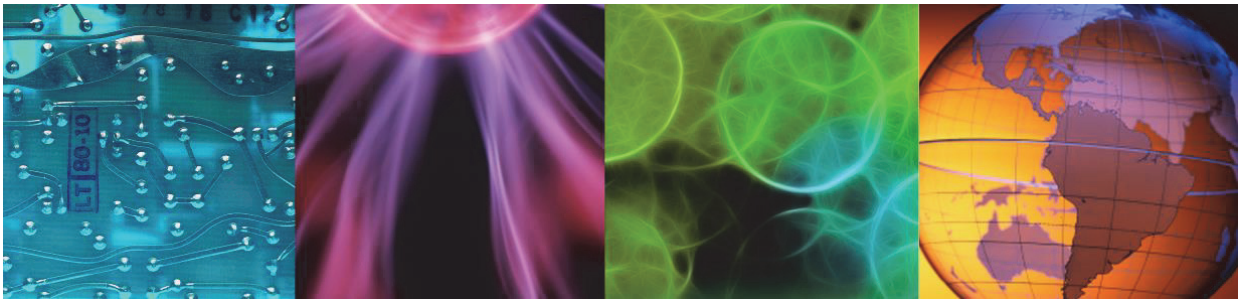
IDT logging

The IDT keystroke logs are available through the Activity tab after you select a device. Unless you give IDT administration rights to users, they can view only their own interactions with the device. This policy is different from *units of work*, where you can see all the *units of work* that your group submits.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Device terminal command filtering

Lesson 2 Device terminal command filtering



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe a command filter
- Create and configure a command filter
- Apply a command filter to a group

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

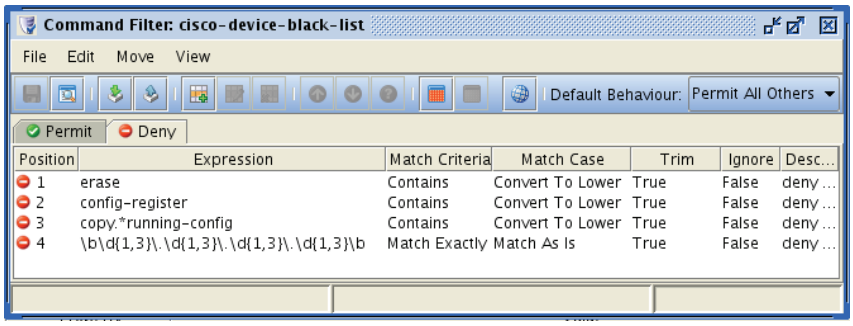
Command filtering

Is similar to security sets, it adds a security layer on devices

Is applied to group or realm combinations, like security sets

Determines which commands a user can or cannot use

Uses vendor-specific command filters



© Copyright IBM Corporation 2016

Command filtering

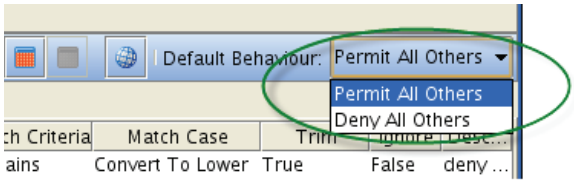
Command filtering is a tool that you can use to further extend the security features of IDT. This tool is similar to a security set. A security set places *view, add, modify,* and *delete* rights on commands in the configuration. And just like security sets, the command filter applies to group and realm combinations. The command filter defines what commands you can or cannot send to the device through the device terminal.

Because the command-line interface syntax differs between vendor devices, the command filters must have a VTMOS filter to ensure correct application. You can make the filter generic since most command-line syntax does not vary much between vendor models and operating systems.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring a command filter

Select **Permit All Others** or **Deny All Others**



Click the **Permit** tab or the **Deny** tab

- **Permit** tab used with **Deny All Others**
- **Deny** tab used with **Permit All Others**

© Copyright IBM Corporation 2016

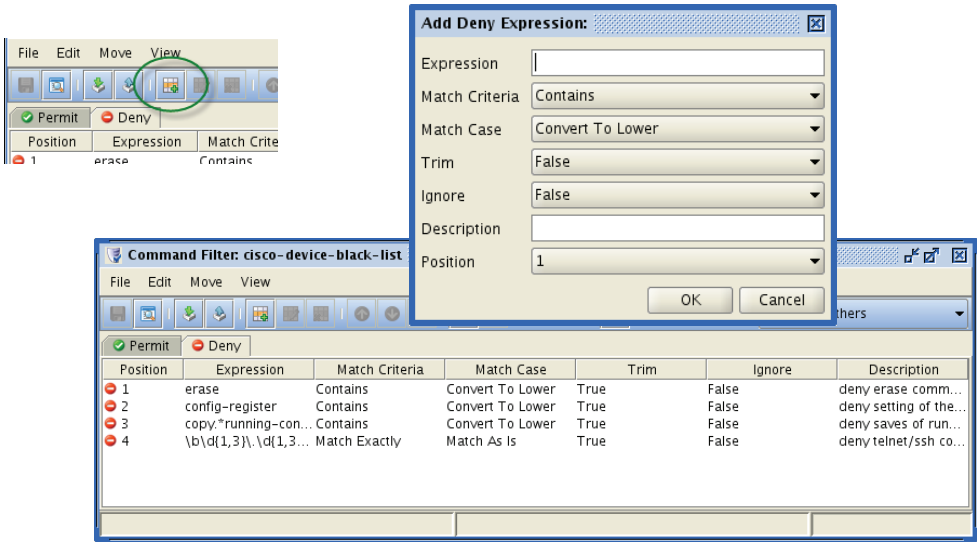
Configuring a command filter

When you create a command filter, you must determine whether you design it to deny or permit specific commands. It is similar to a *permit any any* or *deny any any* with regards to access list creation.

1. Select **Permit All Others** or **Deny All Others** in the Default Behavior menu.
2. Click the **Permit** tab or the **Deny** tab based on the default behavior that you chose in the previous step. In this tab, you define the expressions that you either permit or deny.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining the commands to permit or deny



© Copyright IBM Corporation 2016

Defining the commands to permit or deny

After you select a **Permit** or **Deny** tab, you must define the commands in the list. The command filter resource contains a series of examples. The key to any of these commands is that they are not enabled by default with **Ignore** set to **True**.

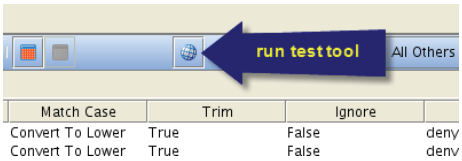
When you create an expression, you must complete these items:

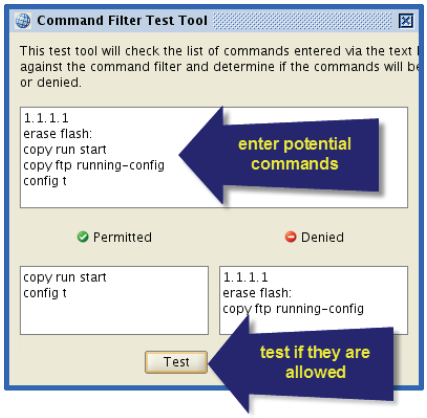
- **Expression**: Enter a regular expression to match against the input stream sent to the device.
- **Match Criteria**: Choose Contains or Match Exactly regarding the expression that you define.
- **Match Case**: Convert the input stream to Match As Is, Convert To Lower, or Convert To Upper. In most cases, you might want to convert the content to all upper or lowercase to make matching commands easier.
- **Trim**: Select True or False to specify whether the command filter must remove leading and trailing white spaces before matching.
- **Ignore**: Set this value to **True** to disable this permit or deny command.
- **Description**: Enter a description of the expression.
- **Position**: Place the permit or deny commands in a specific order to minimize command checking. Place the commands that are used most often at the top of the list.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Testing command filters

Use the GUI to test the command filters





© Copyright IBM Corporation 2016

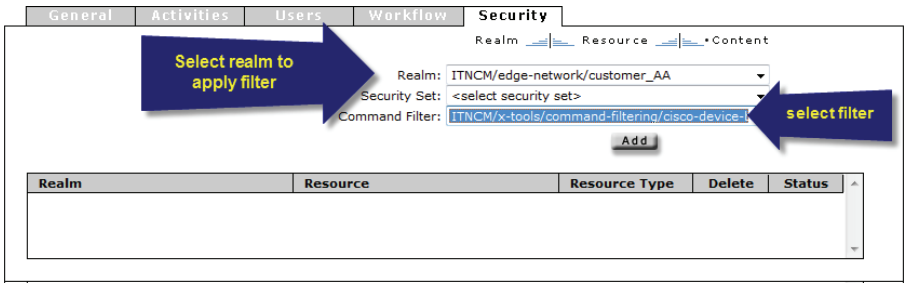
Testing command filters

Use the command filter interface to determine the commands that your filter permits or denies. Run the test tool by selecting the icon on the interface. Enter the commands that a user might enter in the IDT session and test to see which commands it permits or denies.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Applying command filters to groups

Use the Security tab for the group



One command filter is retrieved in an upward search, looking for one that matches VTMOS of device

© Copyright IBM Corporation 2016

Applying command filters to groups

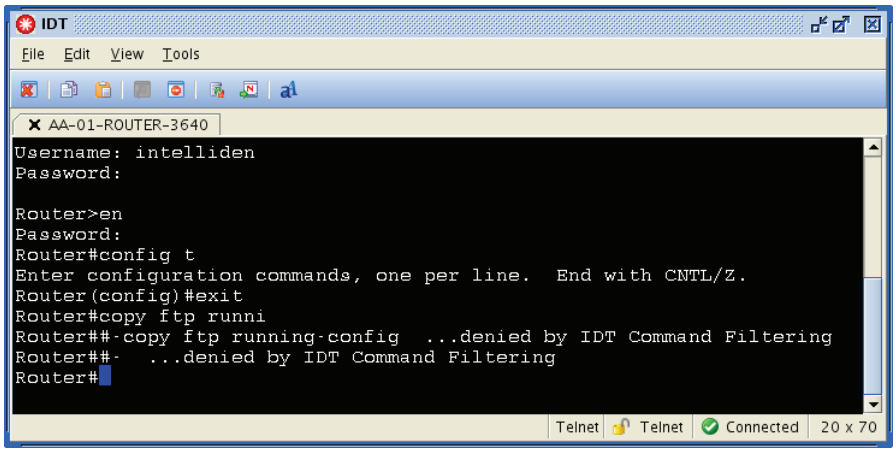
You apply command filters in the same way that you apply security sets to a group and a specific realm. When you open the IDT session, it does an upward search from the device to locate a command filter that matches the VTMOS of the device. It uses the first one that it finds.

Currently, in the 6.4.2 release, it finds only one command filter in the upward search. Having more than one filter resolvable in an upward search can lead to unpredictable results because nothing ensures which one it uses.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Results of applying command filter

A warning is shown: Denied by IDT Command Filtering



© Copyright IBM Corporation 2016

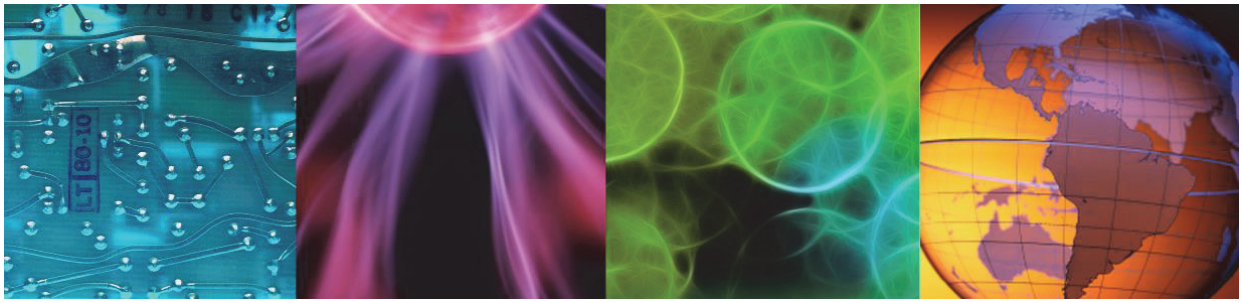
Results of applying command filter

After the command filter is applied to the group, it affects users in that group immediately. Whenever they access a device that is in a realm that is affected by the command filter, it monitors their device terminal sessions. If the command filter does not allow a command, a message prints to the device terminal that states it is denied.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Device terminal administration

Lesson 3 Device terminal administration



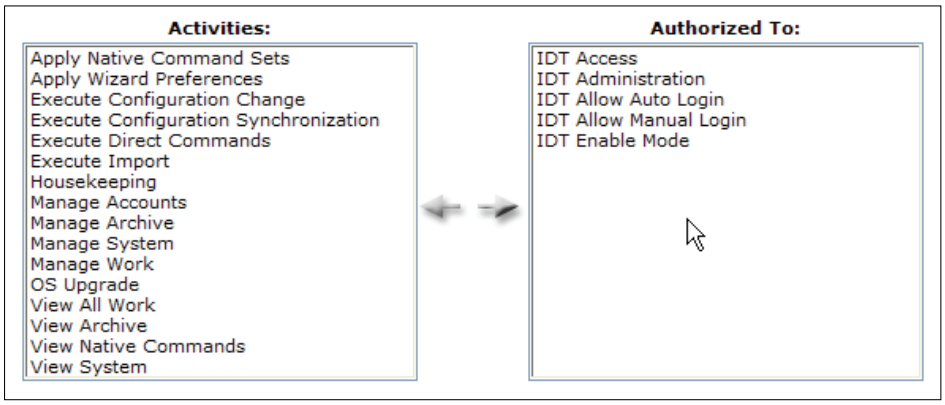
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the features that the system administrator can configure
- Describe the features available to users who have device terminal administration rights

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Providing group access to IDT



© Copyright IBM Corporation 2016

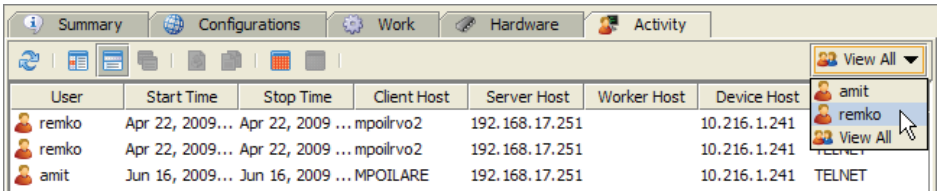
Providing group access to IDT

The user who has account management privileges assigns to a group their ability to use specific IDT menu selections on devices. To use IDT, the administrator must assign both **IDT Access** and a method for access, manual or automatic. The administrator can assign these additional options:

- **IDT Enable Mode**: This option is in addition to the **IDT Allow Auto Login** option. The user is authenticated to the privilege mode on a device. Privilege mode is a Cisco IOS term. In the privilege mode, the user can use the configuration editor. This mode implies that the credentials that are used in the automatic login do not go directly to the privilege mode by using TACACS.
- **IDT Administrator**: An IDT administrator gets authorization for these options:
  - View all user IDT keystroke logs.
  - View all active IDT sessions. Users can view active IDT sessions by right-clicking a device and selecting **Active Terminals**.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Viewing logs



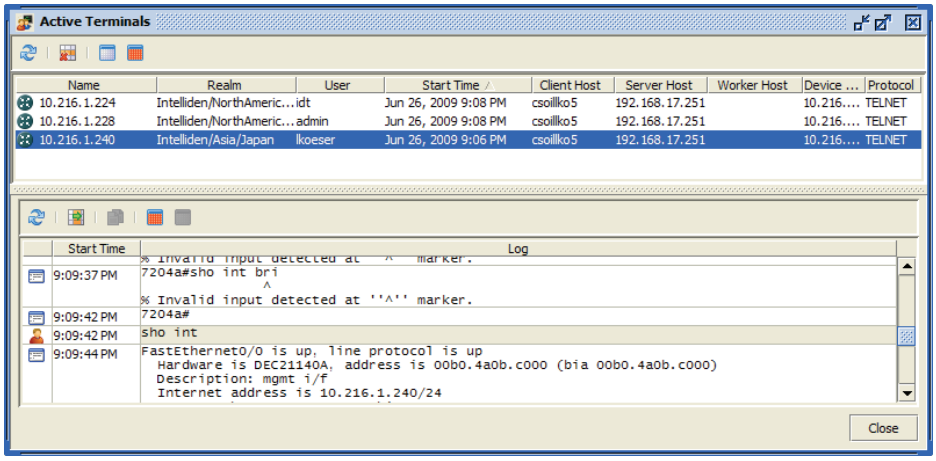
© Copyright IBM Corporation 2016

Viewing logs

If you give a group the IDT administration right, the users can see all the keystroke logs for a device. Normal users can see only their own keystroke logs.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

View active sessions



© Copyright IBM Corporation 2016

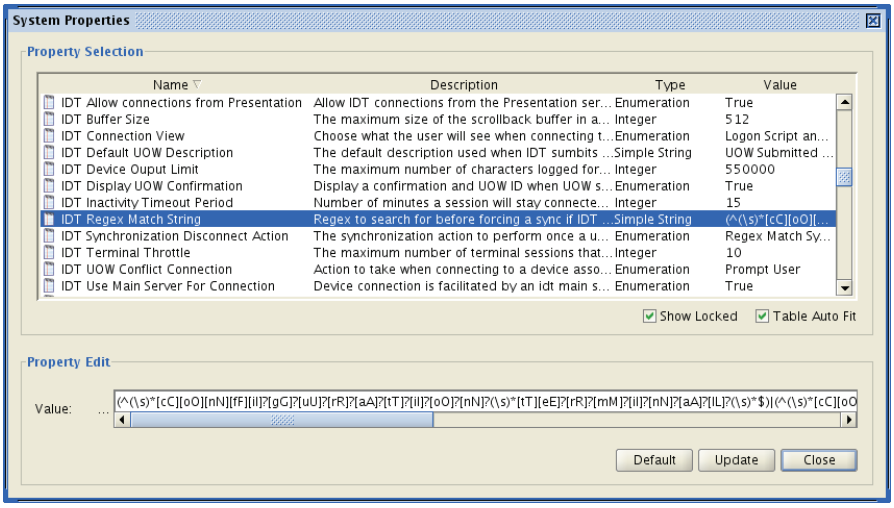
View active sessions

The IDT administrator can see all active IDT sessions and all keystrokes sent by the monitored user. The administrator can use the refresh icon to see any new commands that the user submits.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring other IDT options

Use the system properties manager to configure the options



© Copyright IBM Corporation 2016

Configuring other IDT options

The system administrator can implement specific IDT behavior through the System Properties pane. Only users who have System Manager activities that are associated with their group.can access this pane. Refer to the student notes to review the options that you can configure for IDT.

- **IDT Buffer Size**: Maximum size of scroll back buffer in IDT terminal. Use this property to configure the number of lines that the user sees when using the scroll function.
- **IDT Connection View**: This property sets the animation that is shown when connecting to a device.
- **IDT Default UOW Description**: This property sets the description that a default uses when IDT submits a UOW.
- **IDT Device Output Limit**: This property limits the number of characters that are logged for any device output. A limit of zero returns all output.
- **IDT Display UOW Confirmation**: This property shows a confirmation and UOW ID when the UOW is submitted successfully.
- **IDT Inactivity Timeout Period**: This property sets the number of minutes a session stays connected when the user is not active. The default setting is 15 minutes. However, any timeout that is set in the device configuration that is less than this value overrides this property.
- **IDT Regex Match String**: This property is a regular expression that is used to search all input streams. If anything matches this string, it forces a synchronization. This property is valid only if the IDT Synchronization Disconnect Action is set to Regex Match Synchronization.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

- **IDT Synchronization Disconnect Action**: This property is the Synchronization action to perform after a user disconnects from a device. The options are Prompt synchronization, Force synchronization, or use regular expression to match synchronization.
- **IDT Terminal Throttle**: This property configures the maximum number of terminal sessions that each client can run at any time.
- **IDT UOW Conflict Connection**: This property is the action to take when connecting to a device that is associated with active *units of work*.
- **IDT Use Main Server For Connection**: This property is the device connection that an IDT presentation server facilitates. Otherwise, a user's presentation server facilitates it.
- **IDT User Conflict Connection**: This property is the action to take when connecting to a device that has an active IDT session. This property determines whether to warn the user, prevent connection, or take no action.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Use the IBM device terminal and connect it to a device
- Create, apply, and test a command filter for a device terminal
- Describe the administration function of a device terminal

© Copyright IBM Corporation 2016

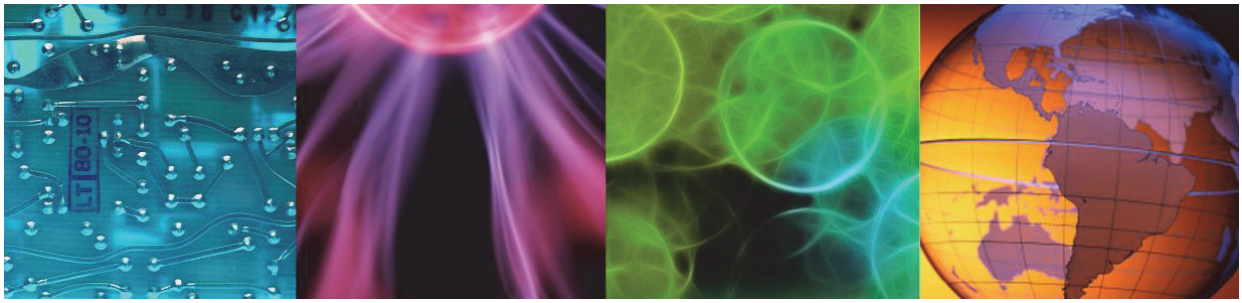
Summary

This unit presented the IBM device terminal, which allows authorized users to make direct VTY connections to the devices under management. You can limit the input for these connections with command filters that check commands before they are sent to the device. You learned about the keystroke log capabilities of the tool and how to administer it.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

12 Advanced command sets

Advanced command sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches you how to build command sets. You learn the advantages of choosing modeled or native command set types. You build and test modeled command sets by using their advanced search and replace features. You also learn about command set groups.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Describe the advantages and disadvantages of command set types
- Describe native command sets and their features
- Build and test modeled command sets that have advanced features
- Describe command set groups

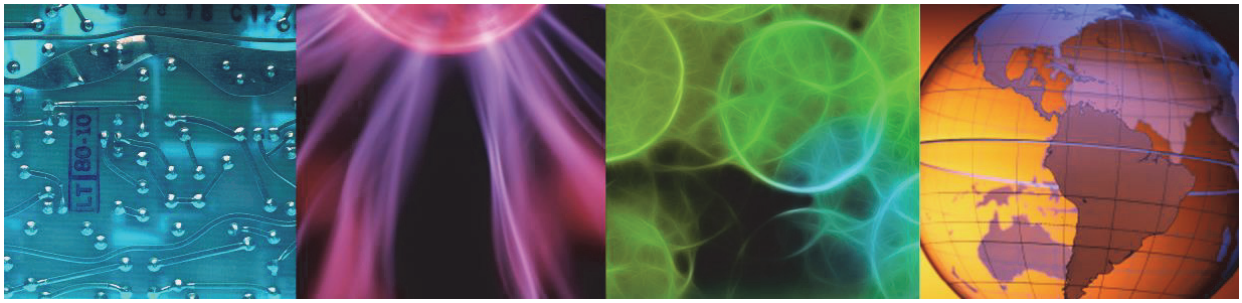
© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Command set types

Lesson 1 Command set types



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the advantages and disadvantages of command set types
- Describe native command sets and their features

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Command sets

You can apply a snippet or template of a resource configuration to one or more devices or realms

Command sets that you apply as a UOW can be scheduled and approved

You can apply command sets across models and operating systems, there is schema or command commonality at high levels

Command sets can use parameters that are defined at execution time (reusable)

© Copyright IBM Corporation 2016

Command sets

You can create change templates and apply standardized changes to the network. These templates apply one or more commands to a configuration to enable or disable a service that the network device is supplying. Use command sets to add, modify, or delete one or more commands. A simple command set is easier to manage, and makes command set names easier to maintain. You can apply one or more command sets to a series of devices in a unit of work.

Command sets are applied to devices as a unit of work. Like other *units of work*, the user identifies the items necessary for its execution. Also, you can schedule, prioritize, and approve the command set execution.

Command sets are vendor-specific. Command sets are designed to modify commands on vendor devices and are tied to the specific syntax of a device to enable, modify, or disable commands. For instance, command sets that are designed for a Juniper device do not function on a Cisco device because the syntax is different. However, within vendors, command sets can apply to various models and operating systems. For example, the creation of a banner for a Cisco device is syntactically the same since the earliest versions of the Cisco IOS. A command set for setting a banner typically works on all Cisco IOS devices, regardless of model or operating system. For more complex areas of the IOS command structure, you might have some difficulty when you apply the same command sets. Special features such as VPNs, quality of service, and traffic shaping are ideas that change between models and operating systems. Problems can arise when you assume the command sets work the same way in these areas.

Command sets are easy to parameterize. By adding parameters to them, they can become reusable. When users run these parameterized command sets, the execution wizard prompts the

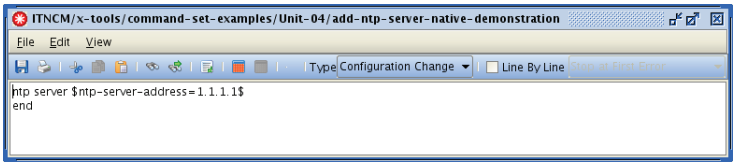
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

user to complete their values. If you apply command sets through an external API program, that utility can replace the parameters with values that the higher-level code supplies.

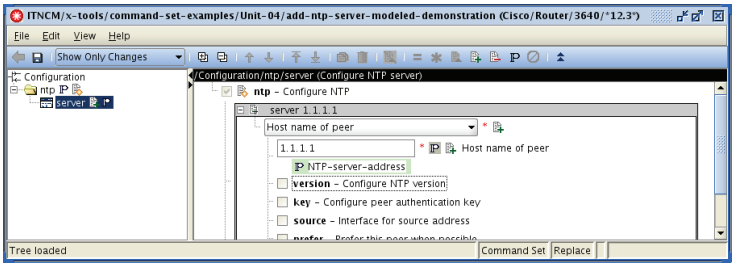
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Native and modeled command sets

Native command set



Modeled command set



© Copyright IBM Corporation 2016

Native and modeled command sets

The solution provides two types of command sets: native and modeled. As you saw earlier, a native command set uses the native syntax of the device to apply changes. The modeled command set applies a standard markup language to an XML representation of the configuration. On the slide, you see an example of applying network time protocol servers to a Cisco device by using the two different command set types.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Command set comparison

Feature	Modeled	Native
Syntax checking	☺	☹
Search and replace	☺	☹
Application rules	☺	☹
Testing support	☺	☹
Cut and paste support	☹	☺
Interrogate device and log results	☹	☺
JavaScript support	☹	☺

© Copyright IBM Corporation 2016

Command set comparison

The solution provides two types of command sets: modeled and native. Modeled command sets have these features:

- They use the SmartModel to support features like syntax checking.
- They support search and replace features to allow dynamic changes to device configurations.
- They provide a set of defined rules for ensuring that data is not accidentally overwritten.
- You can test them to predict the actual CLI that is sent to one or more devices.

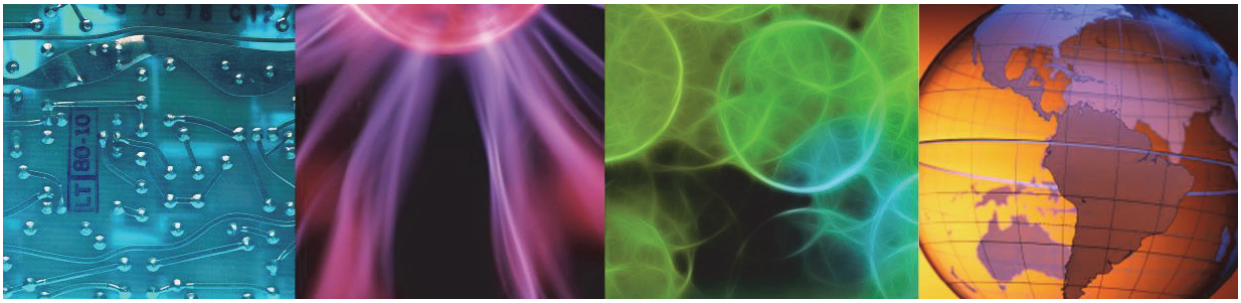
Native command sets have these features:

- They support cutting and pasting existing CLI templates to an editor, like Notepad. You can migrate cutting and pasting existing change templates to Netcool Configuration Manager.
- You can use them in an *interrogation* mode where commands are sent to the device, and the unit of work log saves their results.
- The native command set can use JavaScript to access external resources such as a database.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Native command sets

Lesson 2 Native command sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the native command set features
- Enable the security to allow their use
- Describe the use of Java Script to enhance functions

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Overview

Applies commands to a device using the native CLI syntax of that device

Applies commands in both configuration and nonconfiguration modes

- Commands must be correctly formatted

Supports integrations

- Applications that create dynamic configlets for devices
- WANDL, MPLS design and optimization

Customers can use current change templates

© Copyright IBM Corporation 2016

Overview

A native command set is designed to use the natural command language of a device to change its state. The commands that the user submits to the device are input in whatever way that the device accepts changes to the way that it operates. One of the keys to using a native command set is that the user supplies all the syntax and the command order.

The native command set has two modes. One mode is the standard *configuration change* mode, where the commands are designed to alter the state of the device. The other mode is the *interrogation* mode, which retrieves the status information from the device. Regardless of mode, the user supplies the syntax and ordering.

The original intent of a native command set is to support integration between the product and other applications that generate CLI to modify the device. You can see this principle illustrated in the following example:

- An MPLS design and optimization solution retrieves data from network routers and builds a graphical representation of the MPLS cloud. From there, the user can manipulate and optimize the MPLS network. These changes result in new commands that the user cuts and pastes onto the network devices to implement the design. Native command sets and the API can replace the cut and paste and automate a solution.

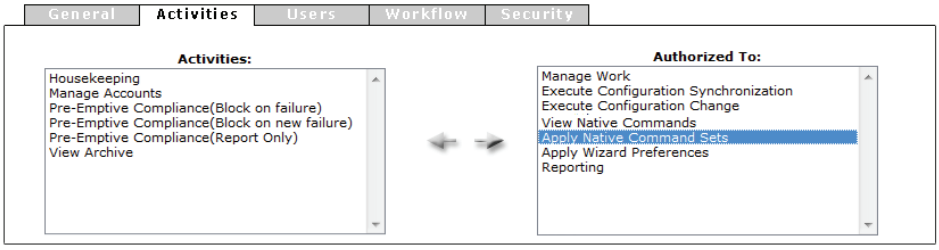
Native command sets allow customers to take current change templates and paste them into the editor. Customers can use command sets to adopt change management capabilities.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Security settings

A group must have the **Apply Native Command Sets** activity

The activity applies to both interrogative and configuration change command sets



© Copyright IBM Corporation 2016

Security settings

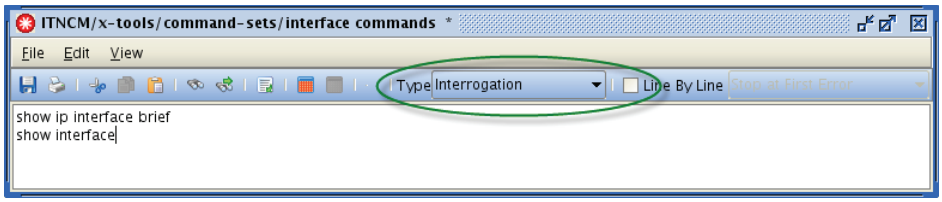
For a group to process native command sets, it must have the **Apply Native Command Sets** activity associated with the group. Regardless of the type of native command set used, it requires this activity.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Interrogation mode

Commands are submitted, and responses are captured

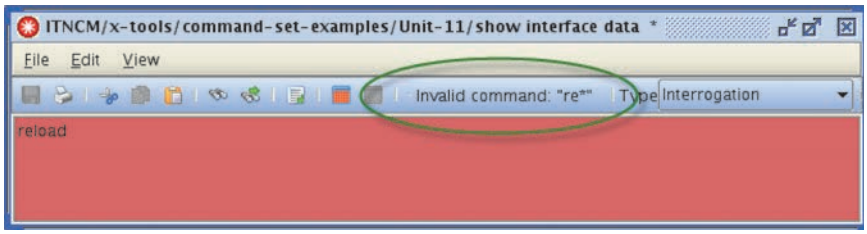
Some commands cannot be sent



© Copyright IBM Corporation 2016

Interrogation mode

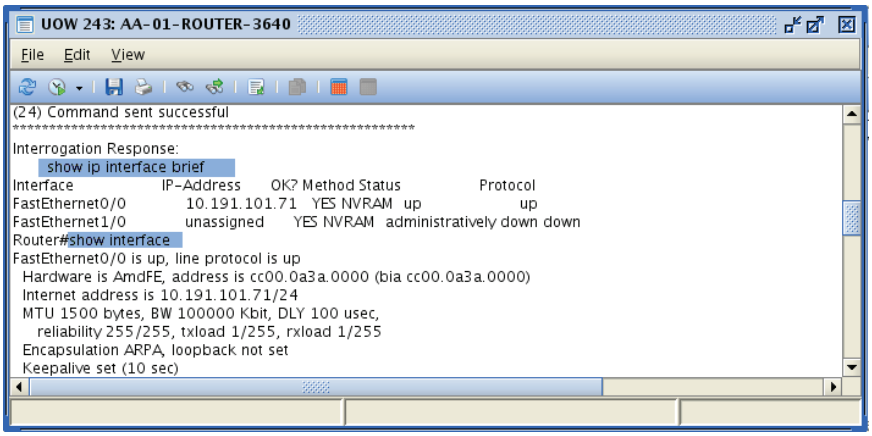
To use the *interrogation* mode to get device status information, you must set the type to **Interrogation**. The *interrogation* mode prevents the submission of various commands that can cause service disruption. These commands are similar to **reload** or **erase**. The *interrogation* mode does not authorize you go into a configuration mode and change the device. When you enter commands that are not authorized, you receive a warning.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Interrogation mode output

Saves output of each command in a native command set to the UOW log



© Copyright IBM Corporation 2016

Interrogation mode output

When you use an interrogative native command set, it outputs the data that the device responds with into the *unit of work* log.

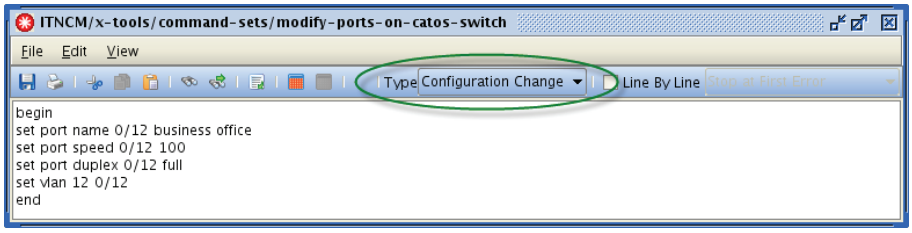
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuration change

Configuration change merges commands into the running configuration of the device, uses the RAD to determine application method

Commands must be correctly ordered

Configuration change must contain the correct deletion commands



© Copyright IBM Corporation 2016

Configuration change

You can also use a native command set to make configuration changes. You must set the type to **Configuration Change**. As mentioned previously, you must enter the commands in the correct order and with the correct syntax. You can use abbreviated commands when the device recognizes the syntax. You must delete commands also. For example, working with Cisco access control lists, you delete the list and re-create it whenever you modify it.

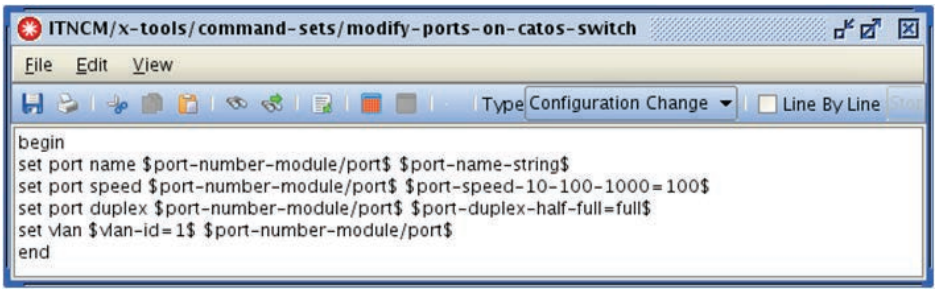
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Parameters

Simple string replacement is **$parameter-name$**

Can default to **$parameter-name=1.2.1.2$**

Can use with interrogation or configuration change modes



© Copyright IBM Corporation 2016

Parameters

Native command sets can have parameters to ensure their reuse. Setting parameters in the command set definition causes the execution wizard to prompt for them at execution time.

Placing dollar signs ($) around a parameter name defines the parameter. The system uses simple string replacement to fill the parameters with data at execution time. Using default data is also useful when creating parameters. You use the equals sign to precede the default you want to use. You can use these parameters on both configuration changes and interrogative command sets.

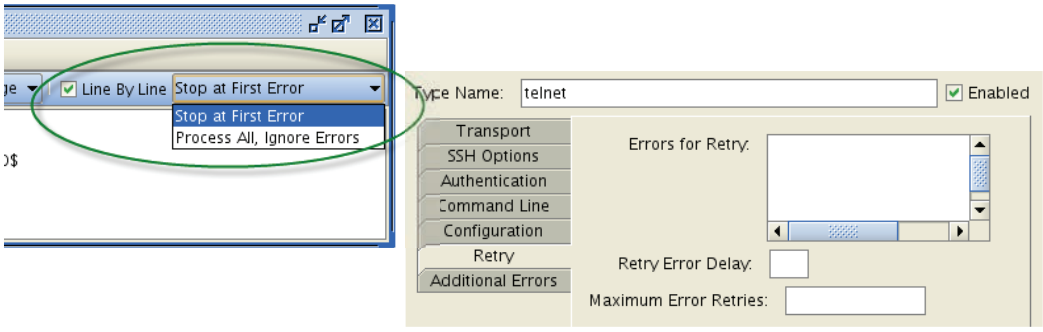
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Line-by-line mode

Default method for applying commands, apply all the commands and look for errors generated by device

Overrides FTP methods and uses streaming

Works with the **Errors for Retry** commands in the RAD



© Copyright IBM Corporation 2016

Line-by-line mode

The native command set has a feature that is unique. You can run the native command set in a *line by line* mode. With this mode, which is not available for modeled command sets, you have more control over how you submit commands to the device.

By default, native command sets are applied in the same manner as modeled command sets. All the commands are sent to the device, and the return buffer is viewed for any errors. The commands are retrieved through an FTP command and merged into the running configuration. The commands are searched for errors only after they are merged.

With native command sets, the *line by line* mode provides finer control over the way that it applies the commands. In this case, it applies each command and parses the results. By selecting *line by line* mode, the FTP behavior is overridden, and the commands stream to the device. Use the *line by line* mode with the *errors for retry* capabilities in the *resource access document*. This process ensures that commands are sent again if an error indicates that the device is not ready for the next command. With line by line, you can employ the following two options:

- **Stop At First Error**: The UOW fails on the first command that returns an error that is in the **default.errors** list in the device script.
- **Process All, Ignore Error**: All commands are sent to the device before the results of the commands are parsed for errors.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

JavaScript support

Native command set can execute JavaScript
- Supported in both interrogation and configuration change modes

Can use to employ complex application logic
- Check whether the IP is in use before applying to interface
- Retrieve the value from the database and apply to device configuration

Includes three classes
- ScriptLogger
- DeviceInterface
- SystemInterface

Supports parameter replacement

Scripting and its limitations are beyond the scope of this course

© Copyright IBM Corporation 2016

JavaScript support

Instead of placing CLI commands into a native command set, you can use the native command set to run a JavaScript. You can use this scripting to apply changes to a device and run interrogation commands against its operating system.

With scripting, native command sets use complex application logic and techniques to interact with a device, as the following example describes:

- You can use a scripted native command set to see whether an IP address is in use before applying it to a device interface. A practical application of this idea is applying an IP address to a management *loopback* interface. You can script the native command set to test a ping command against an IP address. If the ping responds, then the IP address is in use. If you apply that address to the loopback interface, the native command set fails.
- You can use a scripted native command set to retrieve a value from a database and use it in the JavaScript logic. You can also use the database value in commands that you apply to the device. Reserve this type of behavior for the Java API developers.

The JavaScript includes three classes for developing JavaScript native command sets:

- **ScriptLogger**: Appends messages to the unit of work log.
- **DeviceInterface**: Sends commands to the device that the worker server is interacting with. Returns responses to the JavaScript execution.
- **SystemInterface**: Sends system commands to the operating system of the worker server. Returns responses to the JavaScript execution.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

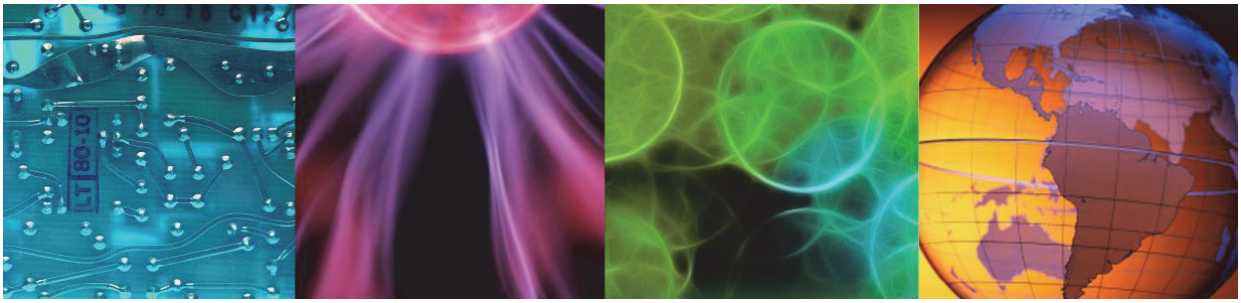
As with any native command set, you can define parameters by using the dollar sign ($) around a parameter name. You define the values of these parameters before the worker server runs the JavaScript.

JavaScript and its use with a native command set is beyond the scope of this training course.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Modeled command sets

Lesson 3 Modeled command sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe what a markup is and how to use it
- Create command sets to add, delete, and modify configurations
- Use parameters to make command set generic

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a modeled command set

A modeled command set is a combination of defining CLI snippets and markup
- Markup is for adding, deleting, or modifying commands on a device

Command sets are specific to VTMOS
- Are often applied to similar vendor and type

Creating a modeled command set is a two-step process
- Define the commands
- Apply the markup to the commands

You must have a plan for the intended change

© Copyright IBM Corporation 2016

Creating a modeled command set

In this lesson, you learn how to create modeled command sets. This feature might be the most complex tool that you use.

A modeled command set is a combination of a CLI snippet and markup with operations that tell the software what to do with the CLI. Markup determines whether the command is added to the device, is removed from it, or is used to modify the device.

Like native command sets, modeled command sets are VTMOS-specific, for example, you cannot use Juniper commands on a Cisco device. However, modeled command sets use VTMOS criteria to select a device driver and its schema to develop the command set. In this case, you must define the VTMOS to an operating system level so that it selects the correct driver. At high levels in the command structure, the schemas of a single vendor can align, for example, creating banner commands. However, the deeper into the command structure you go, the more they can diverge.

Creating a modeled command set is a two-step process:

1. You define commands in the command set editor. You open the editor and see an empty configuration catalog. You add commands manually or by importing a configuration or another command set.

2. You must add markup to the modeled command set to tell the software what it needs to do with the commands. If you have no markup, you have no change. In other words, you must have markup icons on the command so that the software recognizes the commands that are defined.

You use a native command set to apply commands to the device, even if those commands are not required. A modeled command set is much more judicious in its application and does not submit

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

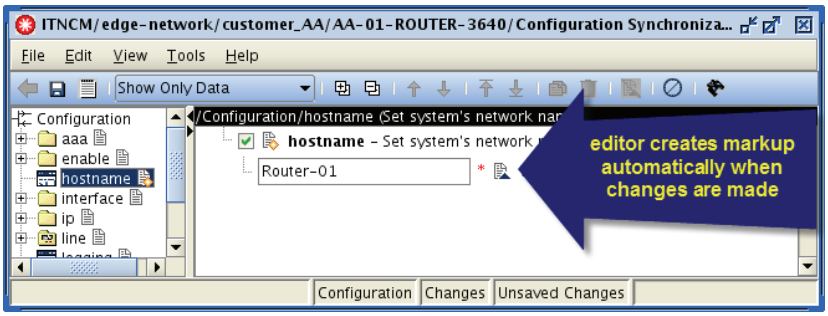
changes unless needed. Modeled command sets can also implement search and replace activities. Creating a modeled command set requires more thought to get the expected results. You must create a plan to implement the changes you intend.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining markup in a command set

You modify a configuration manually



Commands sets do not automatically create markup

Practicing a configuration helps you understand the markup

© Copyright IBM Corporation 2016

Defining markup in a command set

In previous exercises, you modified a configuration by opening the editor and manually changing it. You can add, delete, or modify commands in the editor. When you change an existing configuration, the original configuration and your modifications are available. These changes between the original and modified configuration are noted through markup.

When building a command set, you must tell the command set what to do with commands. For example, are you modifying a host name or adding one.

The command set editor has extra markup settings for searching. However, the concept of adding, modifying, and deleting is the same between a configuration change and a command set. You can change a configuration that reflects what the command set is attempting to do. You can use the markup created by the configuration edit as a guide for marking up the command set in the command set editor.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Adding commands to a device

Adding implies that commands do not currently exist on the device

You must add the markup in the correct location

BEFORE

```
interface FastEthernet2/0
 no ip address
 ip directed-broadcast
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
logging 10.191.101.52
snmp-server community wr1t3Str1ng RW
snmp-server community private RO
!
```

AFTER

```
interface FastEthernet2/0
 no ip address
 ip directed-broadcast
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
Logging 10.191.101.52
snmp-server community wr1t3Str1ng RW
snmp-server community R34d0nl33 RO
snmp-server community private RO
!
```

© Copyright IBM Corporation 2016

Adding commands to a device

You use the **add** markup icon to add markup to a command that you want to add to a configuration.

When creating a command set that adds commands, the command or any extended version of it cannot exist on the device. This rule is important because you might want to add commands to the device, but they are not authorized.

In the slide example, the following command is added:

```
snmp-server community R34d0nl33 RO
```

If that command or any version of it exists on the device, the command set does not add the command. Additionally, if any of the following commands are on the device, the command set does not apply the change.

```
snmp-server community R34d0nl33 RW
snmp-server community R34d0nl33 RO
snmp-server community R34d0nl33 view test RO
```

You do not want to add commands that overwrite existing data. The **add** commands must be new and unique to the configuration before the code creates a change.

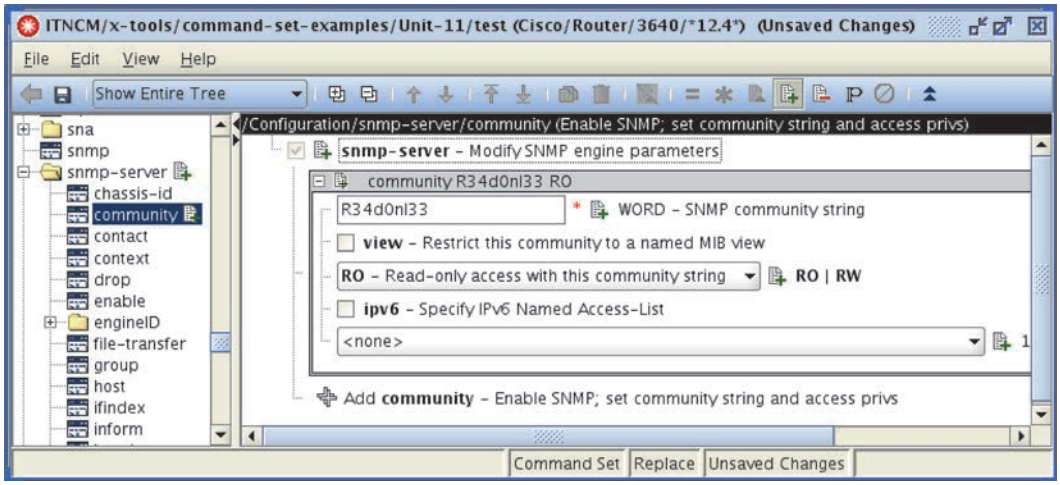
The command set editor has a Match window and a Replace window. The Match criteria are shown on the Match window. On the Replace window, you can use that criteria to add, modify, or delete data from the configuration. The **add** markup is enabled when you are on the Replace window.

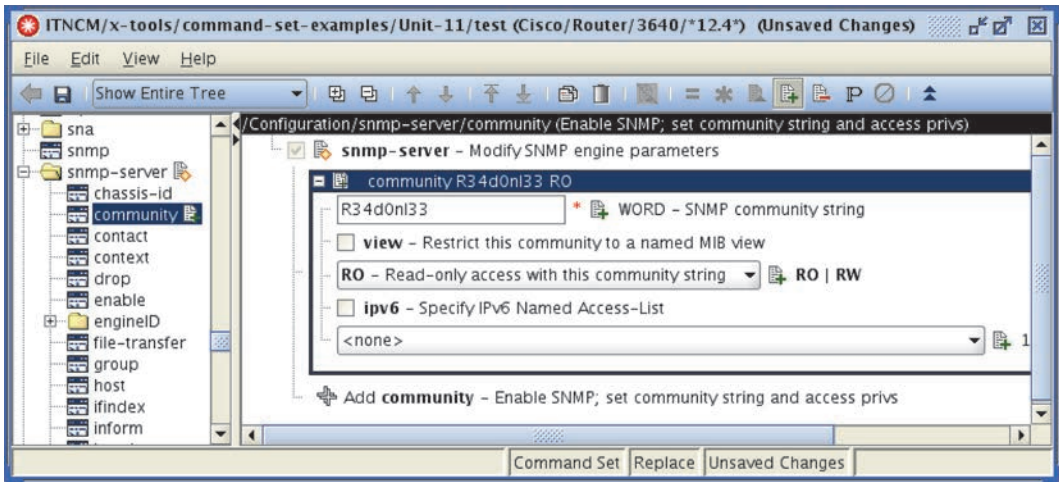
You use the double triangle icon to switch between the two windows.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

When the Replace window is shown, the add markup icon becomes active. You can place the markup on the command that you want to add.

You must take care where you put the **add** markup. The command set editor is encoded to prevent poor choices, but it is not foolproof. For example, look at the difference between these two command sets and their markup.





Notice in the upper command set that the **add** markup is on the SNMP-server node. In the lower command set, you see a yellow diamond markup icon on the SNMP-server node. Moving your cursor over this icon reveals its title, **child data will be changed**.

No changes are made to the configuration when you apply the upper command set because **snmp-server** commands are already on the device. In the second command set, the yellow diamond markup icon implies a change somewhere underneath the node. In this case, you are adding a community string to the SNMP-server node. It also implies in the **child data will be changed** node that the data in that node is used as search criteria.

Hint: An example of this command set is on the training server under the **TNCM/x-tools/command-set-examples/Unit-12/example-snmp-community-add-string** directory.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Deleting commands from a device

Deleting implies that commands currently exist on the device

Be careful where you place the delete markup

BEFORE

```
interface FastEthernet2/0
 no ip address
 ip directed-broadcast
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
!
logging 10.191.101.52
snmp-server community wr1t3Str1ng RW
snmp-server community private RO
!
```

AFTER

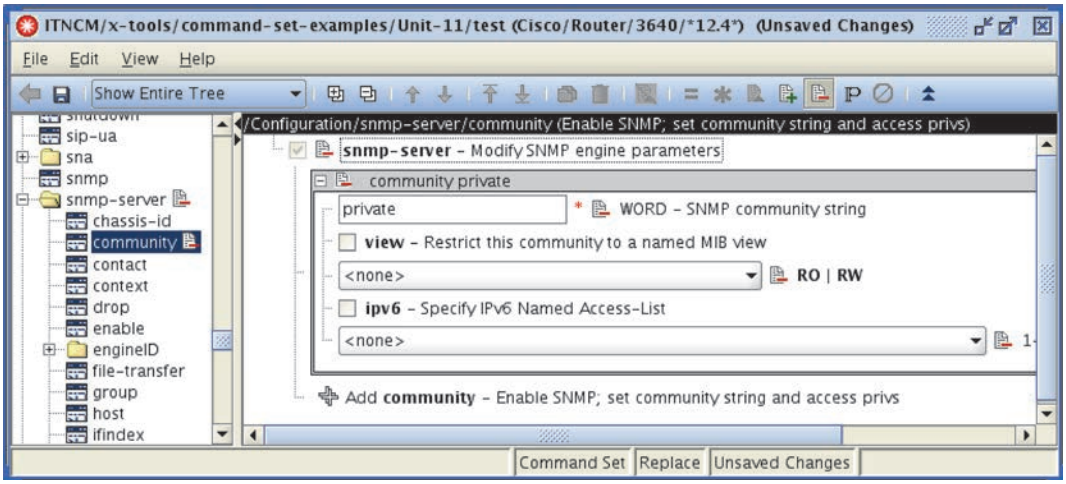
```
interface FastEthernet2/0
 no ip address
 ip directed-broadcast
 duplex auto
 speed auto
!
no ip http server
no ip http secure-server
!
!
logging 10.191.101.52
snmp-server community wr1t3Str1ng RW
!
```

© Copyright IBM Corporation 2016

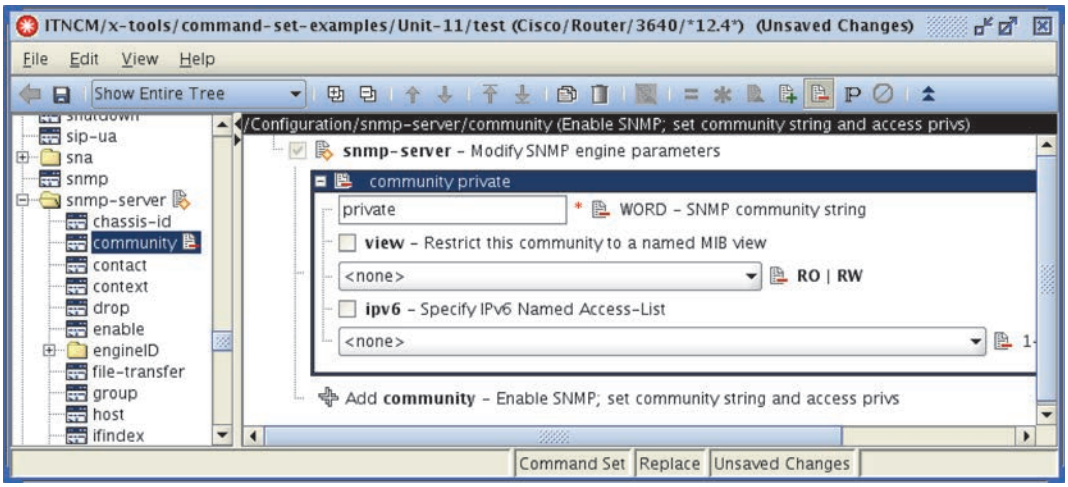
Deleting commands from a device

You use the Delete Markup icon to add delete markup to a command that you want to remove from a configuration. When you create a command set that deletes one or more commands from a configuration, one or more of the commands must exist on the device. You can save processing time when you do not send deletions to a device that has nothing to delete.

You can add only delete markup to commands when you are in the command Replace window. You must be careful where in the command set editor you put the deletion icon. For example, examine the difference between the following two command sets and their markup.



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.



You have the same situation as shown with the add command example. By placing the delete markup on the SNMP-server node, you place delete markup on the community string and all the other SNMP-server commands. You want to place the delete markup only in the community/private rollup window, highlighted in dark blue in the second screen capture on this page.

In the example, the type of community string, for example, read and write, is not defined. You are not required to define that for deletion. By defining the string type, it further filters the search. If you define the attribute as read-only (RO), the command must be **snmp-server community private RO**.

Hint: An example of this command set is on the training server under the **TNCM/x-tools/command-set-examples/Unit-12/example-snmp-community-delete-string** directory.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Importing data into a command set

You can import data from another command set or a configuration

Importing configurations imports only the data

Data without markup is ignored when applying a command set

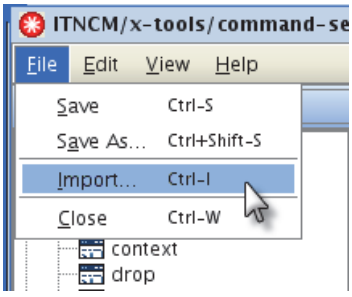
Importing command sets imports data and markup

© Copyright IBM Corporation 2016

Importing data into a command set

Importing data from a configuration or another command set helps you build a command set. Importing is especially useful when you create a complex template with many commands.

From the command set editor, you select **File** > **Import** and select either a configuration or a command set. The system filters for matching vendor and type.



You can import a configuration into a command set. If you do so, it imports the entire configuration into the command set. While this much data might be daunting, you must remember that without markup, all of that data is ignored. You must mark up the portion of the configuration that you want to make part of the command set. If you import a command set, you import both its data and the markup.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Important: When you import a command set, verify that the VTMOS of the command sets align. The import process brings in all details of the command set that you are importing. If differences exist between the schemas of the two command sets, you might have orphan XML in the resulting command set. If parameters exist in this orphaned data, those parameters appear in the execution wizard but are not visible in the command set editor.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Testing a modeled command set

Command set must align with a device with same vendor and type

You run a command set in *report only* mode to run as a UOW

You import a command set into a configuration, ensure that parameters have default data

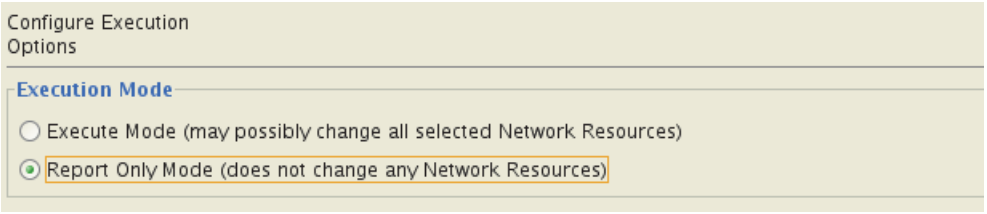
© Copyright IBM Corporation 2016

Testing a modeled command set

You can test modeled command sets. This feature is not available to native command sets.

To test a command set, you must have a device that is of the same vendor and type that is imported, and under management. The device must have a configuration state that has the data you are interested in manipulating with your command set. If you designed the command set to modify a policy map, it is important that the policy map exists on the device.

You can test a command set by using two methods. The first method is the most flexible because it uses the workflow. You can run a command set in *report only* mode. This mode determines what commands it sends to a device and presents them to the user. With *report only* mode, you can define values for parameters. This ability is useful for command sets that are ready for production.

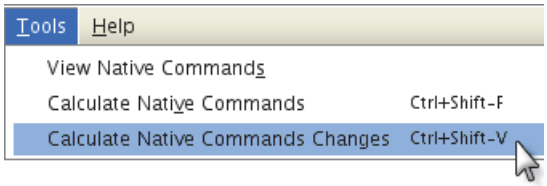


The other method of testing a command set is to import it into a modeled configuration. This method is not as flexible as running it in *report only* mode but provides quicker feedback. You must meet the following requirements when you want to test a command set in this manner:

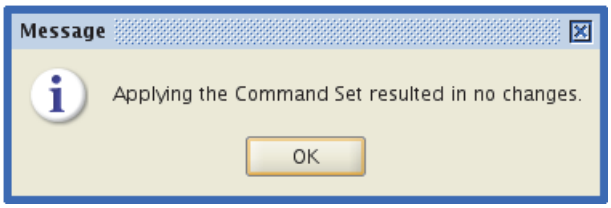
- If you define parameters, include useful default data. Importing the command set into a configuration does not support parameter modification. The default data must align with the configuration that you want to modify.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

- The command set must align with the vendor and type of the configuration that imports it.
- The configuration must have the correct type of data for that command set.
- You can view the results of the command when you select the **Calculate Native Command Changes** option.

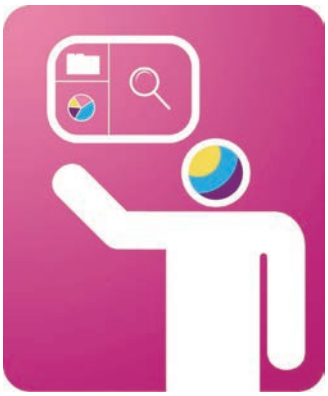


If the command set effected no changes, you receive a warning. This warning can help you work on complex markup problems when you know that changes must occur.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Instructor demonstration



© Copyright IBM Corporation 2016

Instructor demonstration

To help in the upcoming exercises, the instructor demonstrates the following activities:

- Build an *add* command set
- Test an *add* command set with *report only* mode
- Import a configuration into a command set
- Build a *delete* command set
- Test a *delete* command set by importing it into a configuration

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modifying commands on a device

Modify commands that exist on a device

Be sure to place the markup in the correct location in the editor

BEFORE

```
ip access-list extended management-servers
remark TNCM servers
permit tcp host 172.200.201.51 any
permit tcp host 172.200.201.52 any
permit tcp host 172.200.201.53 any
permit tcp host 172.10.201.6 any
permit tcp host 172.10.201.7 any
permit tcp host 172.10.201.8 any
remark VTY emergency server
permit tcp host 199.168.32.62 any
remark ITNM servers
permit tcp host 172.200.201.5 any
permit tcp host 172.10.201.99 any
remark TACACS servers
permit tcp host 192.20.5.58 any
permit tcp host 192.20.5.55 any
permit tcp host 192.20.7.22 any
```

AFTER

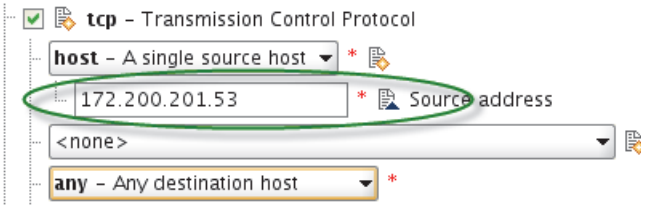
```
ip access-list extended management-servers
remark TNCM servers
permit tcp host 172.200.201.51 any
permit tcp host 172.200.201.52 any
permit tcp host 172.200.201.69 any
permit tcp host 172.10.201.6 any
permit tcp host 172.10.201.7 any
permit tcp host 172.10.201.8 any
remark VTY emergency server
permit tcp host 199.168.32.62 any
remark ITNM servers
permit tcp host 172.200.201.5 any
permit tcp host 172.10.201.99 any
remark TACACS servers
permit tcp host 192.20.5.58 any
permit tcp host 192.20.5.55 any
permit tcp host 192.20.7.22 any
```

© Copyright IBM Corporation 2016

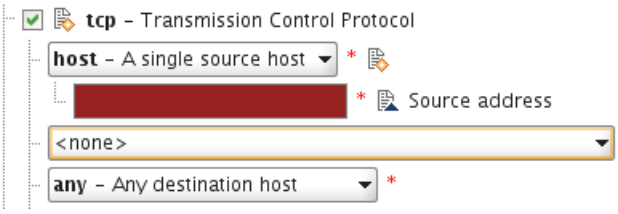
Modifying commands on a device

For modification markup, you use the modify icon.

You place modification markup on arguments that are associated with commands. The intent is that you are modifying a value associated with a command. To place modify markup on arguments, you use the Match window in the command set editor.



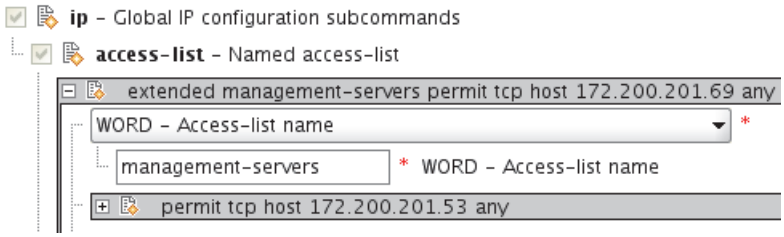
After you place modify markup on the Match window, you must switch to the Replace window to tell the system what to change the value to.



As with other markup, it marks the parent nodes of the modified command with the **child data will be changed** icon.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The nodes that are marked up become search criteria for the command set. The command set that reflects the previous change to a permit statement has the subsequent markup.



These automatic markup items become part of the search criteria, and the search must find these items first before a change is applied to the device. The implication is that the device called **management-servers** contains a named access list with correct capitalization. Next, it must find the permit statement **permit host 172.200.201.53 any** before the command set can modify the IP address.

Hint: An example of this command set is on the training server under the **TNCM/x-tools/command-set-examples/Unit-12/example-ACL-modify-permit-statement** directory.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using wildcards

Wildcards add flexibility to command sets

- You can use them in multiple locations
- In this example, the wildcard is 172.10.201.*

BEFORE

```
ip access-list extended management-servers
remark TNCM servers
permit tcp host 172.200.201.51 any
permit tcp host 172.200.201.52 any
permit tcp host 172.200.201.53 any
permit tcp host 172.10.201.6 any
permit tcp host 172.10.201.7 any
permit tcp host 172.10.201.8 any
remark VTY emergency server
permit tcp host 199.168.32.62 any
remark ITNM servers
permit tcp host 172.200.201.5 any
permit tcp host 172.10.201.99 any
remark TACACS servers
permit tcp host 192.20.5.58 any
```

AFTER

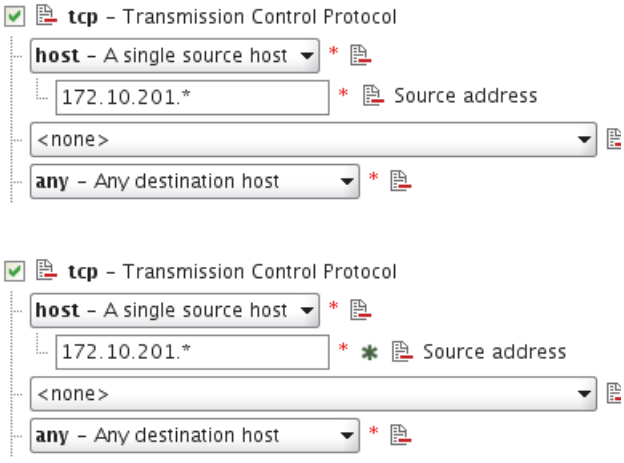
```
ip access-list extended management-servers
remark TNCM servers
permit tcp host 172.200.201.51 any
permit tcp host 172.200.201.52 any
permit tcp host 172.200.201.69 any
remark VTY emergency server
permit tcp host 199.168.32.62 any
remark ITNM servers
permit tcp host 172.200.201.5 any
remark TACACS servers
permit tcp host 192.20.5.58 any
```

© Copyright IBM Corporation 2016

Using wildcards

In the Match window, you can also use wildcards on arguments to search for multiple results inside a configuration. Use wildcards with other markup to modify or delete commands. When you work with wildcards, you see the wildcard icon.

Placing wildcard markup on an argument tells the command set interpreter that the argument is a regular expression and not an explicit string. An example is shown in the following two permit statements.

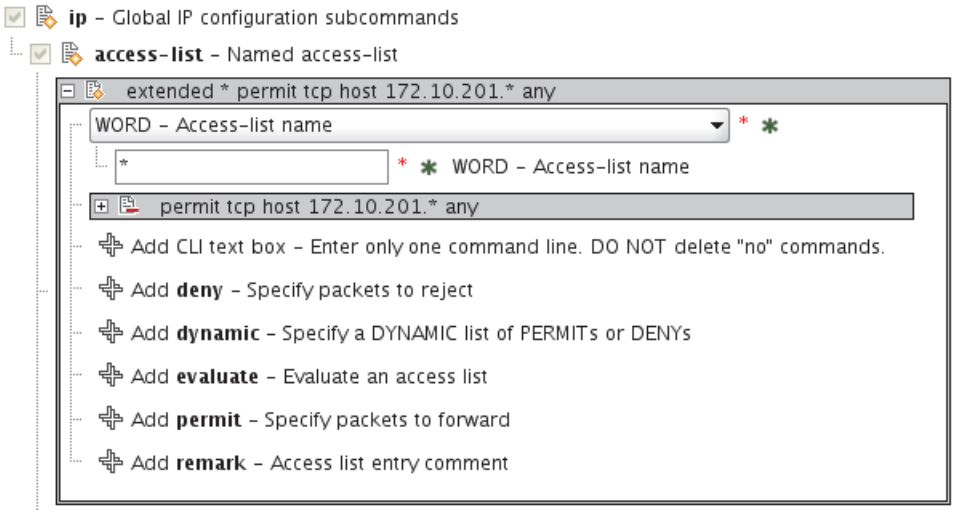


The second statement has the wildcard markup beside the argument **172.10.201.***. Without that markup, as in the first permit statement, the command set interpreter looks at the argument of **172.10.201.*** and searches for that explicit string in the configuration. It never finds that explicit

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

string. In the second example, the interpreter treats **172.10.201.*** as a regular expression, finds all matching permit statements, and deletes them from the configuration.

You can use wildcards in multiple locations in the command set. What is the result of this command set?



The answer is that it searches all the named access lists on the device for the IP addresses that match **172.10.201.*** for a host. Then, it deletes them.

Hint: An example of this command set is on the training server under the **TNCM/x-tools/command-set-examples/Unit-12/example-ACL-delete-permit-w-wildcard** directory.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using parameters

You can use parameters to create generic command sets

You can define them on the Match window and the Replace window

BEFORE

```
ip access-list extended <access-list-name>
…
…
permit tcp host <old-ip-address> any
…
…
…
…
…
```

AFTER

```
ip access-list extended <access-list-name>
…
…
permit tcp host <new-ip-address> any
…
…
…
…
…
```

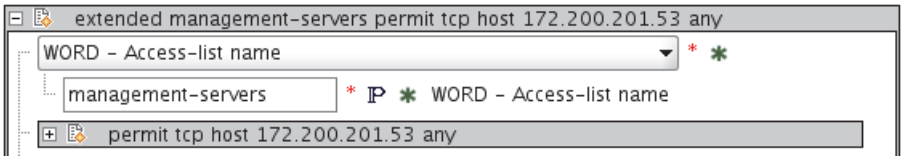
© Copyright IBM Corporation 2016

Using parameters

As with native command sets, parameters ensure that modeled command sets are reusable. You add parameters after you define and test the command set and after it works for at least one test case. After you have a functional command set, you can add parameters to those arguments that change between executions. The parameter markup icon is visible in both the Match and Replace windows.

You can place parameters in both the Match and Replace windows. The search criteria is modifiable at execution time. Assign default values to the argument that has the parameters. It requires the default data to import the command set into a configuration for testing.

You can use parameters with other markup, as shown in the following example:

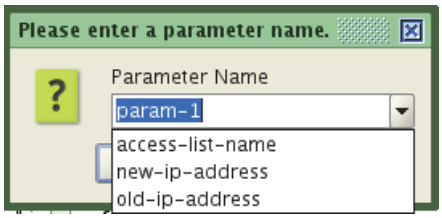


The wildcard markup is defined for an argument beside a parameter with a default value of **management-servers**. At execution time, that value can change to an asterisk (*) or some other search criteria to search for all the named access lists on the device.

When placing parameter markup on an argument, it prompts you for a parameter name. A good naming convention is helpful to the people who submit this command set. When you enter a

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

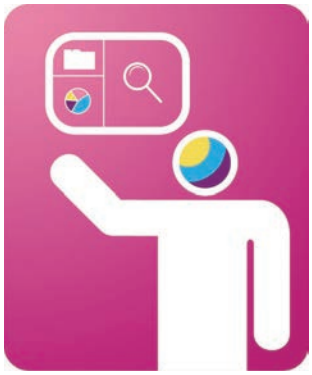
parameter name, for example, an interface number, you can see a list of all the previously created parameters. You can select that value instead of entering it again.



Hint: An example of this command set is on the training server under the **TNCM/x-tools/command-set-examples/Unit-12/example-ACL-modify-permit-w-parameters** directory.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Instructor demonstration



© Copyright IBM Corporation 2016

Instructor demonstration

To help in the upcoming exercises, the instructor demonstrates the following tasks:

- Build a *modify* command set with wildcards and parameters
- Test a *modify* command set by importing it into a configuration

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Matching on sibling commands

- Defines required criteria before another change can happen
- Currently allows only searches or changes between siblings

BEFORE

AFTER

```
ip access-list extended management-servers
remark TNCM servers
permit tcp host 172.200.201.51 any
permit tcp host 172.200.201.52 any
permit tcp host 172.200.201.53 any
permit tcp host 172.10.201.6 any
permit tcp host 172.10.201.7 any
permit tcp host 172.10.201.8 any
remark VTY emergency server
permit tcp host 199.168.32.62 any
remark ITNM servers
permit tcp host 172.200.201.5 any
permit tcp host 172.10.201.99 any
remark TACACS servers
permit tcp host 192.20.5.58 any
permit tcp host 192.20.5.55 any
permit tcp host 192.20.7.22 any
```

```
ip access-list extended management-servers
remark TNCM servers
permit tcp host 172.200.201.51 any
permit tcp host 172.200.201.52 any
permit tcp host 172.200.201.69 any
permit tcp host 172.10.201.6 any
permit tcp host 172.10.201.7 any
permit tcp host 172.10.201.8 any
remark VTY emergency server
permit tcp host 199.168.32.62 any
remark ITNM servers
permit tcp host 172.200.201.22 any
permit tcp host 172.200.201.5 any
permit tcp host 172.10.201.99 any
remark TACACS servers
permit tcp host 192.20.5.58 any
permit tcp host 192.20.5.55 any
permit tcp host 192.20.7.22 any
```

© Copyright IBM Corporation 2016

Matching on sibling commands

The final piece of markup allows extra matching criteria when you modify commands on a device. The match markup verifies that a command exists before a change can occur. You see the match icon when you use match markup.

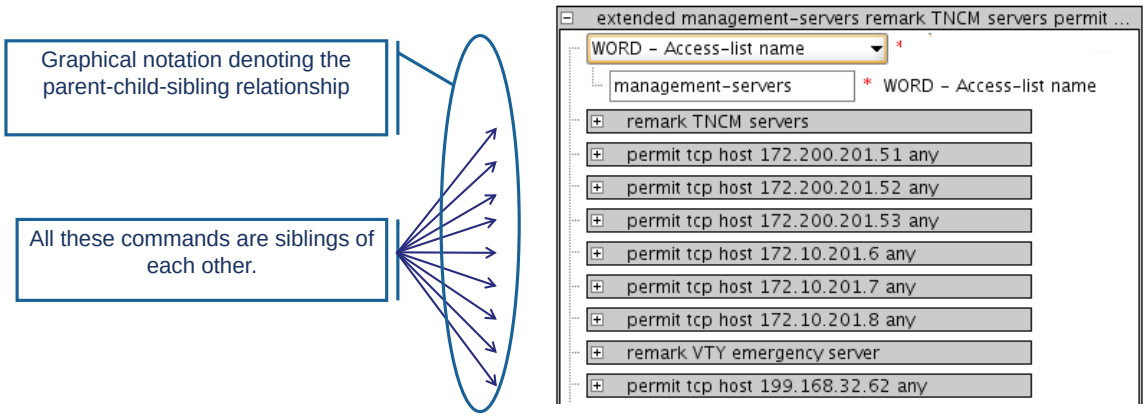
In the example, the command set is searching for the remark statement **remark ITNM servers** in a particular access list. If the remark is found, it places a permit statement with a specific IP address beneath the remark.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Parent-child-sibling relationships

Driven by the XML schema

Noted in the graphical user interface

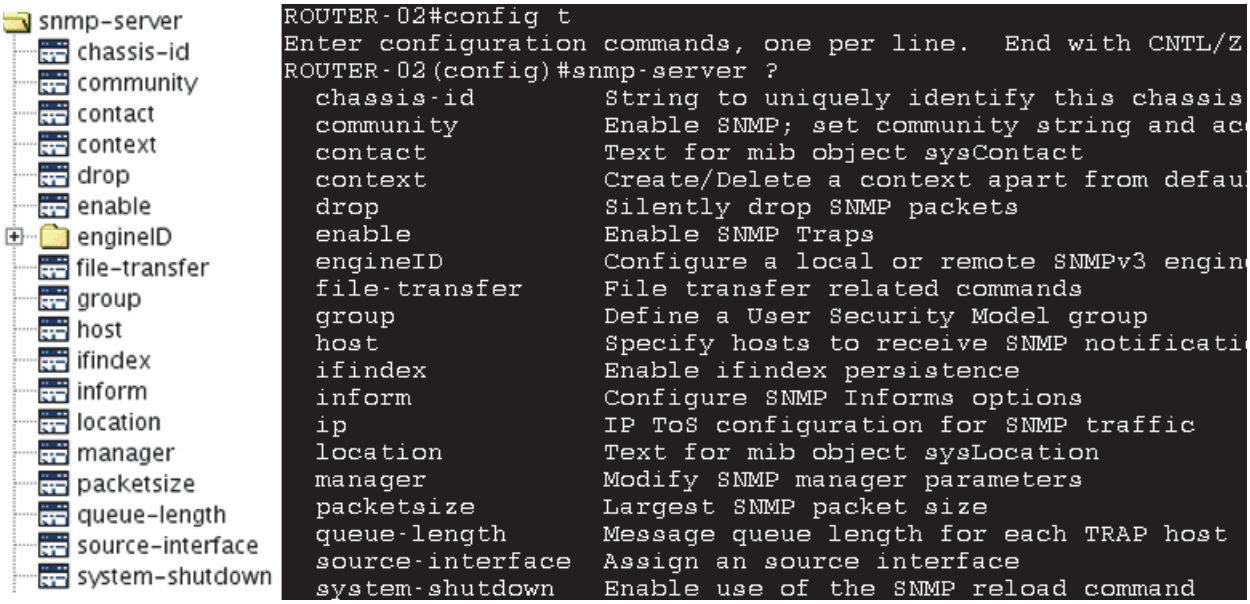


© Copyright IBM Corporation 2016

Parent-child-sibling relationships

Match markup determines the existence of a sibling command. If it finds a sibling command, you can add, modify, or delete another sibling command. So what is a sibling command?

When you model a device, it creates a hierarchy that defines parent-child relationships. For example, the command **snmp-server** has many children. You can see this hierarchy in both the configuration command set editor and in the help menus on a device.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In the example, you see several sibling commands that are children of **snmp-server**. For example, **community** is a child of **snmp-server** but is also sibling to **chassid-id** and **contact**.

You can determine the parent-child-sibling relationships. The command set editor shows graphical lines that connect commands to their parents. The XML schema is more important in this relationship than the command structure of the device. The sibling commands that you see when you view the native configuration are not necessarily the commands that you see when you view the XML.

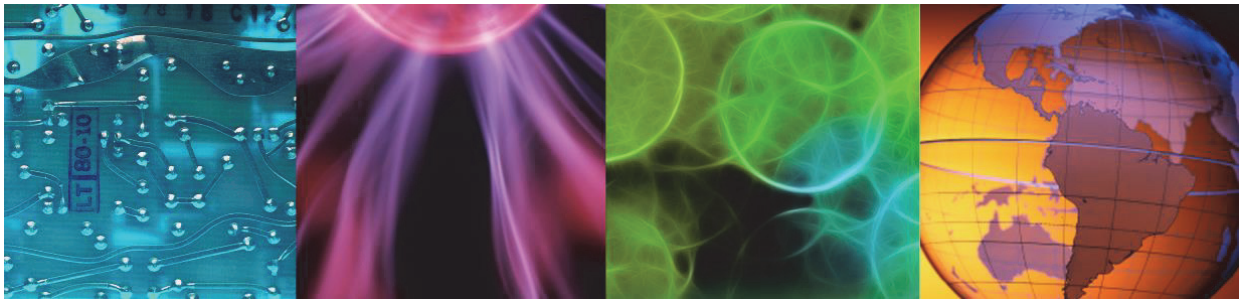
Hint: An example of this command set is on the training server under the **TNCM/x-tools/command-set-examples/Unit-12/example-ACL-add-permit-under-remark** directory.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 Applying command sets

Lesson 4 Applying command sets



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the options for running a command set
- Review the results in a unit of work log

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Overview

Command sets are applied against devices or realms and vice versa

There can be a many-to-many relationship between command sets and network resources or realms

Each command set is one round trip to a device

Failure criteria and rollback options can be set

Depending on command set, no changes can be applied

Command sets can be applied in *report only* mode, effective for single command set against one or more devices

© Copyright IBM Corporation 2016

Overview

After you create and test a command set, you can use it in a production setting for applying changes. You can use commands sets to apply a change to one or many devices. You can also apply the command set to a realm. The opposite is also true. You can select a realm, and apply a command set to it.

You can also apply multiple command sets to multiple devices in a realm. In this case, the change you are applying has these characteristics:

- The change is complex and you must apply it in steps to ensure that you handle it properly. The first command set creates a piece of policy on the device, for example, a class map. The second command set implements the policy by applying it to an interface.
- You do not necessarily know the state of the network. You want to implement a standard banner across all devices. Some devices have a banner and some do not. If you use modeled command sets, you must have two command sets. One adds a banner to a device that does not have one. The second modifies a banner that exists.

Each command set is a complete round trip to the device. In other words, each command set is treated as a single configuration change, and uses all the configuration management steps. This change includes checking the state of the device and database configurations, applying the change, and pulling the resulting configuration and storing it in the database.

You have failure options that you can set when you run command sets against more than one device. These failure options cause the UOW to stop processing when it reaches a certain level of failure.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

When you work with modeled command sets, command sets sometimes do not apply changes. Search criteria are not found, or the application rules for certain types of modeled command sets are violated.

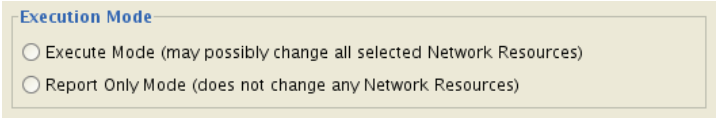
Modeled command sets can run in the *report only* mode. You can review what a command set does to a series of devices before changing the network. It is run like a UOW and produces logs that detail the commands it sends to the device.

Important: *You must run report only* command set execution with only one command set. Each command set that you test is applied to the current configuration in the database. If the multiple command sets have dependencies between them, it does not capture these dependencies because each command set is run against the static current configuration.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

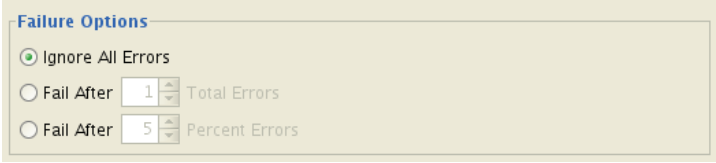
Execution mode and failure options

Execution Mode, modeled only



Application method

Failure Options



© Copyright IBM Corporation 2016

Execution mode and failure options

When applying command sets to a set of devices, the execution wizard prompts for various execution options. The most important option is the Execution Mode, which is shown only when running modeled command sets.

The application method is determining how you want the command sets to process across the devices. This method is relevant when multiple command sets are applied only to multiple devices. Do you want to apply all the command sets to a device or to apply each command set sequentially? The author cannot determine why one selection is chosen over the other.

Important: Selecting **Apply Command Set at a time** prevents the wizard from breaking up this operation in multiple *units of work*.

The following failure options ensure that the application can stop the processing of a unit of work when command sets create errors in the application:

- **Ignore All Errors**: The default selection that ensures that the execution of command sets proceeds across every device. It applies every command set to the device, regardless of a previous command set failure on that same device.
- **Fail After Total Errors**: This option fails the unit of work if a user determined number of individual tasks fails for whatever reason.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

- **Fail After Percent Errors**: This option fails the unit of work if a percentage of individual tasks fails. The total number of tasks are the number of command sets times the number of devices in the UOW. The UOW fails as soon as it reaches or exceeds the percentage. The percentage is calculated by using this formula:

  ```
  [(current total failed tasks in UOW)/(total tasks in UOW)]*100
  ```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

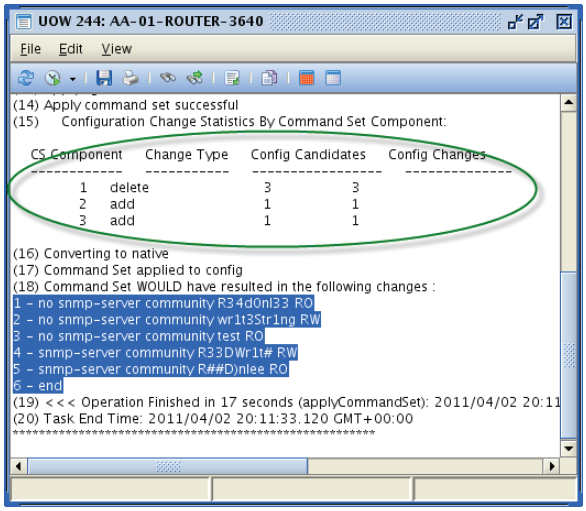
Report only command set UOW logs

Configuration change statistics

- Notes each change component in a command set
- Is helpful with large lists of items

CLI

- Is submitted to the device, if started



© Copyright IBM Corporation 2016

Report only command set UOW logs

When you look at a *report only* unit of work log for a command set execution, you look at two specific areas. The change statistics provide an overview of the adds, modifies, and deletes for the command set. It also shows how many items it searches to determine what to change.

The other item of interest is which CLI would have been submitted to a device if it had processed. In the following example, you can see a *report only* UOW log.

```
CS Component Change Type Config Candidates Config Changes
------------ ----------- ----------------- --------------
1     delete                        11                4
2     add                            1                1
3     PCDATAmodify                   3                1

(16) Converting to native
(17) Command Set applied to config
(18) Command Set WOULD have resulted in the following changes:
1 - snmp-server community R34d0nl33 RO
2 - no snmp-server community private RW
3 - snmp-server community Pr1v4t3 RW
4 - no ip access-list extended management-servers
5 - ip access-list extended management-servers
6 -  remark TNCM servers
7 -  permit tcp host 172.200.201.51 any
```


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

```
8 -  permit tcp host 172.200.201.52 any
9 -  permit tcp host 172.200.201.53 any
10-  remark VTY emergency server
11-  permit tcp host 199.168.32.62 any
12-  remark ITNM servers
13-  permit tcp host 172.200.201.5 any
14-  remark TACACS servers
15-  permit tcp host 192.20.5.58 any
16-  permit tcp host 192.20.5.55 any
```

Command set component 1 deletes permit statements, based on a wildcard that searches for a particular subnet. It searches through a potential list of 11 permit statements and finds four permits that it can delete. The generated CLI removes the access list **management-servers** and rebuilds it after it deletes the four permit statements.

Command set component 2 is adding a command to the device. If the command does not exist, it adds the command. This command is the **snmp-server community R34d0nl33 RO** command.

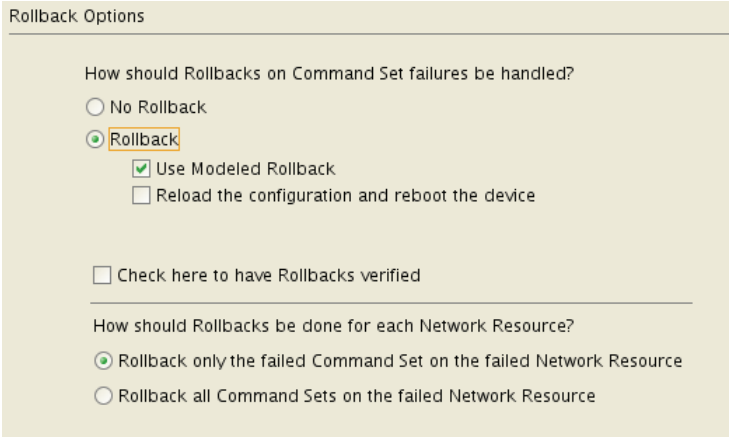
Command set component 3 searches through three candidates that it can modify. Only one meets its criteria. Component 3 modifies that candidate. It replaces one command with another command. This change is reflected in the **no snmp-server community private RW** and **snmp-server community Pr1v4t3 RW**.

Hint: After you review the *report only* unit of work and are satisfied with the results, the UOW can be requeued and set to *run* mode. If the current configurations on the devices are not changed, the changes are the same as the ones that are in the *report-only* logs.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Command set rollback options

You can set the default or hard-code the rollback process by using the *resource access* document



© Copyright IBM Corporation 2016

Command set rollback options

Rollback options are selections that are currently available to command sets only. You can choose how to roll back the device when a failure in the change management process occurs. Rollback is a process that restores the device to its original configuration before the change is attempted. Using a resource access document, you can set the preferences for the rollback options.

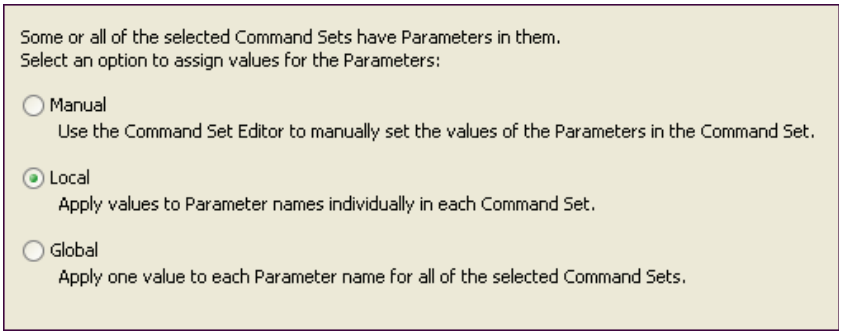
- **No Rollback**: The device remains in whatever configuration state it is in when the error happens.
- **Rollback - Use Modeled Rollback**: This method uses intelligent configuration management to compare the configuration on the device with what is originally on the device. Any differences are added or removed from the device to restore it to its previous state.
- **Rollback - Reload the configuration and reboot the device**: This process is also known as disaster recovery. For a Cisco-like device, it places the raw text-based configuration in the startup-configuration, and the device restarts to return it to its previous state.
- **Check here to have Rollbacks verified**: This option compares the rollback configuration with the database.

  When you have multiple command sets in the UOW, you can configure these next items to tell the system where to roll back the failed device:

  - **Rollback only the failed Command Set on the failed Network Resource**
  - **Rollback all Command Sets on the failed Network Resource**

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Entering values into parameters

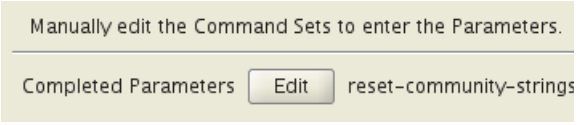


© Copyright IBM Corporation 2016

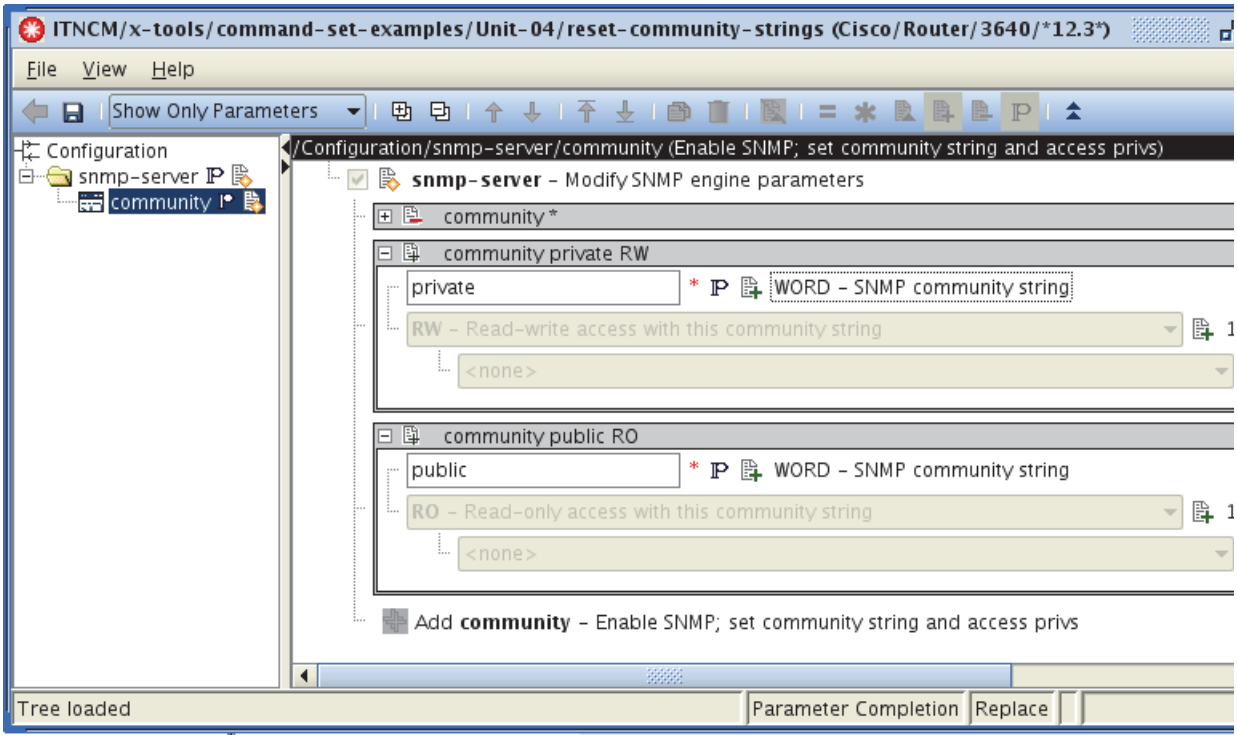
Entering values into parameters

If parameters in any of the command sets are applied, the wizard prompts you to define those values or overwrite the defaults. Three options to assign values for the parameters are manual, local, and global.

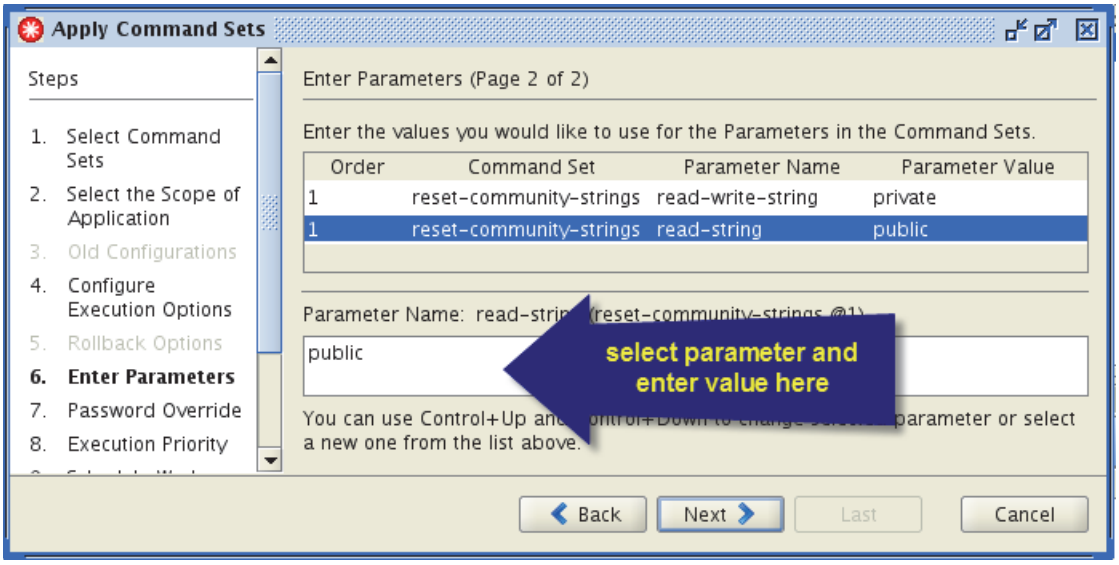
- **Manual**: The system opens a copy of the command set in the command set editor. You find the arguments to change. It is only shown when you use modeled command sets.



© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.



- **Local**: In this pane, you have an interface for setting parameter values.



- **Global**: This method is like the Local selection. This option is shown only when command sets share parameter names. The implication of having similar parameter names in command sets is that they are meant to process together. In this mode, you define the parameter value only once and both command sets use it.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modeled command set application rules

If there is no markup, there is no change

Adding and modifying are different

Matching criteria is case-sensitive

Commands are interpreted by IOS and potentially redefined

© Copyright IBM Corporation 2016

Modeled command set application rules

This slide has some final notes about running modeled command sets.

- If a modeled command set does not contain any markup, it applies no changes to the device.
- You must understand the difference between adding and modifying commands.
- Matching criteria are case-sensitive.
- You can place commands on a device in one format and the IOS interprets them in a different format, as shown in the following example:

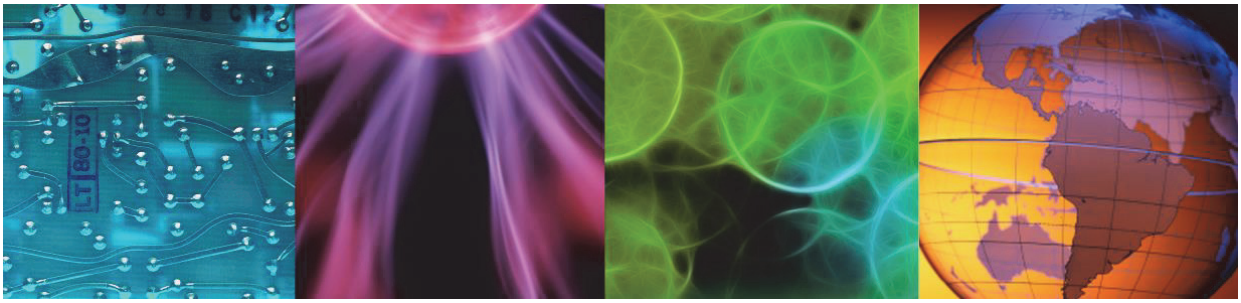
  `permit 12.13.14.1 255.255.255.255` is interpreted to `permit host 12.13.14.1`

  Any command set that searches for permit 12.13.14.1 255.255.255.255 never finds anything because the command is reformatted. The best way to avoid this issue is to look at the configuration. You can import the configuration into the command set and use that format.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 5 Command set groups

Lesson 5 Command set groups



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe a command set group
- Describe some of the steps that are needed to create a command set group

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Command sets are vendor-specific

Command sets are associated to a specific vendor and type, modeled command sets are associated to model and OS

A command set can only modify vendor specific devices

A heterogeneous network needs at least one command set for each vendor, for example, a command set to change the banner

Changing a common value across a heterogeneous network requires separate units of work executing a vendor-specific command set

© Copyright IBM Corporation 2016

Command sets are vendor-specific

As you learned in the previous slides, a command set is a template that applies syntax that is unique to the vendor that it is applied to. The application filters your selections so that a Juniper command set applies only to Juniper devices. If you are managing a network where you have devices of several vendors that are installed and want to apply a change across all these devices, you need a command set for each vendor. To create a banner that has the same legal language regardless of the device vendor forces you to have at least one command set per vendor, possibly more. Another issue with the vendor-specific command sets is that you can have only one vendor in a unit of work. The filtering mechanism the solution uses ensures that it does not apply a command set to an incorrect vendor. As a user, you need to create a unit of work for each vendor under management to implement the common banner.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Command set groups are vendor-neutral

A command set group is designed to be independent of the vendor(s) it is applied against

- VTMOS = */*/*/*

A grouping of command sets designed to implement a similar change

- Change the device banner
- Add a permit statement to an ACL
- Delete a community string

There should be a command set for each vendor model

© Copyright IBM Corporation 2016

Command set groups are vendor-neutral

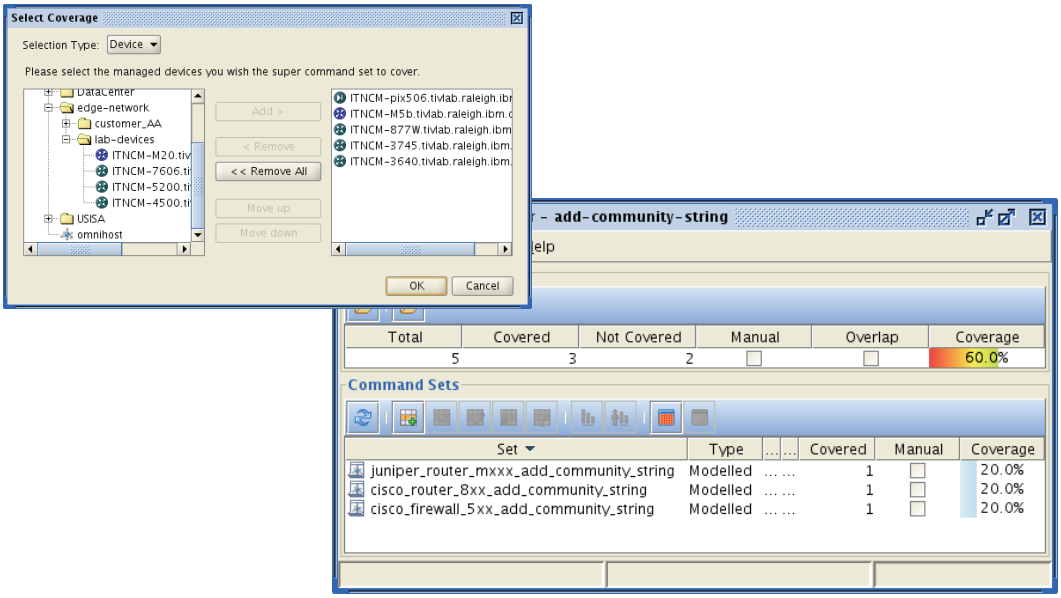
A command set group is an object that you can create in the GUI that is vendor-neutral and is designed to apply a common change to some or all the vendors in your network. When you create one, you are not allowed to define its *vendor, type, model,* and *operating system* as these fields are set to wildcards.

When you create a command set group, you are designing a change that might apply to two or more vendors in your network. The change has a common theme and usually a common result even though you must use different command syntax to accomplish the change. For example, modifying a common banner, adding a permit statement to an access list, or deleting a community string. These concepts have the same results but are implemented differently.

The command set group still requires that you develop a vendor-specific command set for each vendor, but it groups the command sets under a single object.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Add community string across network



© Copyright IBM Corporation 2016

Add community string across network

When developing the command set group, you select a series of devices or a realm that defines all the vendors you must create command sets for. When you finish selecting these devices, you are prompted to build command sets for all the different VTMOS that you chose. As you add the command sets to the group, the interface notes how many devices are covered.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Command set creation

You must create command sets inside the group interface

Consider parameter names

Enter Parameters

Enter the values you would like to use for the Parameters in the Command Set Group(s).

Order	Command Set Group	Parameter Name	Parameter Value
1	add-community-string	community-string-name	default

Enter Parameters

Enter the values you would like to use for the Parameters in the Command Set Group(s).

Order	Command Set Group	Parameter Name	Parameter Value
1	add-community-string	community-string-name	default
1	add-community-string	community-string-name-5300	default
1	add-community-string	community-string-name-juniper	default

© Copyright IBM Corporation 2016

Command set creation

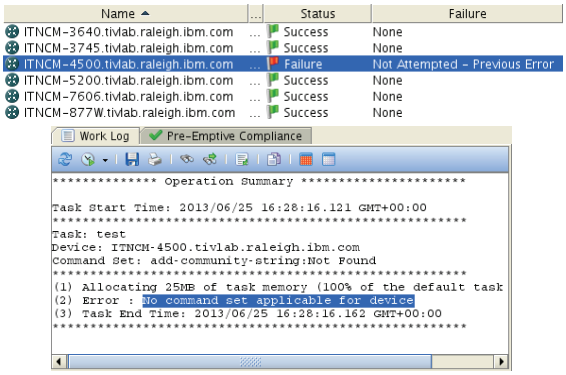
When you create the command set group, you are forced to create the command sets inside the command set group interface. Currently, it does not support dragging and dropping of existing command sets.

As you create you command sets, you are apt to add parameter names to your templates. You should consider implementing common parameter names between each command set so at execution time it prompts only for one value as opposed to many.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Applying command set group

Command set group is applied to a realm or a set of devices Each device is mapped to a command set in a group or generates an error



Processing of tasks depends on number of command set groups in UOW

© Copyright IBM Corporation 2016

Applying command set group

Applying a command set group is similar to applying a command set. The only difference is that you can select more than one vendor when choosing devices. Like command sets, a command set group can run in *report only* mode. If you select a realm when running the command set group, all devices are selected in that realm and its subrealms. When the UOW is processing, each device looks for a command set that matches its VTMOS inside the group. If it does not find a command set, it marks that device as failed, with a note in the UOW log that no command set was found. If a command set is found in the group, it processes in the same fashion as any other command set execution.

As with command sets, if it applies only one command set group to devices, its tasks process concurrently. However, selecting two or more command set groups to apply to a set of devices processes the tasks nonconcurrently.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Describe the advantages and disadvantages of command set types
- Describe native command sets and their features
- Build and test modeled command sets that have advanced features
- Describe command set groups

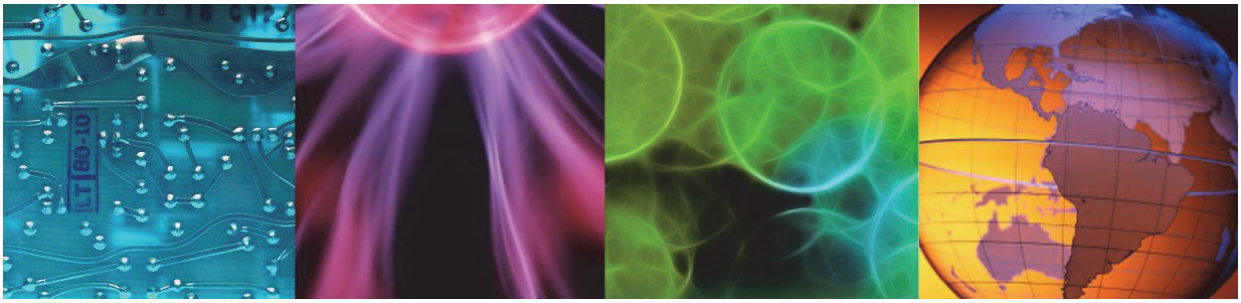
© Copyright IBM Corporation 2016

Summary

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

13 Device OS upgrade manager

Device OS upgrade manager



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit is about using Netcool Configuration Manager to upgrade the operating systems on network devices. You learn the upgrade process and the objects you need to create to ensure a successful upgrade.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Explain how the *upgrade manager* works
- Explain the requirements for a successful upgrade
- Modify an operating system registry
- Create an operating system specification
- Test the network for upgrade compatibility
- Verify the upgrade results

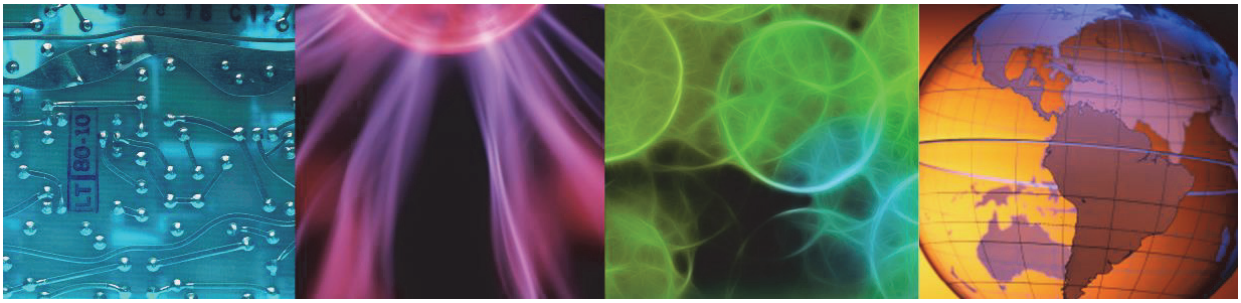
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 OS upgrade overview

Lesson 1 OS upgrade overview



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the OS upgrade process the solution uses
- Describe the requirements and tools that you need to upgrade devices

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Upgrade manager

Automates the manual upgrade processes for devices

Supports improved IOS upgrade process

- Validates available space and removal of files
- Can modify boot parameters in configuration to accommodate new IOS
- Maintains before and after configurations

Supports these devices:

- Cisco IOS
- Cisco CatOS
- Juniper JUNOS

Includes custom scripting and parsing support

© Copyright IBM Corporation 2016

Upgrade manager

Netcool Configuration Manager has an upgrade manager that it uses to upgrade the operating systems on various network devices. This process automates the manual steps that an operator uses to upgrade a device. The application requires a certain amount of setup to upgrade a device. You realize the business value of this feature when you must upgrade 5 or more devices to the same operating system level. Use the manual steps, that the vendor defines to upgrade a single device to a new OS.

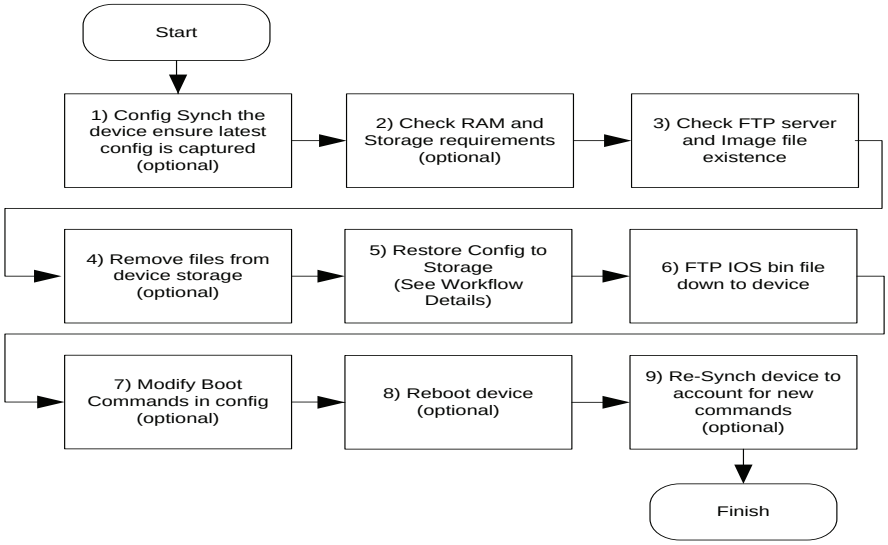
The OS upgrade manager supports an improved upgrade process. It validates that the device has enough memory and disk space to support the new operating system. It can also use command sets to modify configuration parameters that are associated with the new operating systems. It takes snapshots of the before and after configurations to verify that new commands are captured.

The currently supported devices that quality assurance tested, include Cisco IOS and CatOS devices and the Juniper JUNOS operating systems. Product marketing can obtain a list of the device models that IBM development tested.

Customizable parsers are available to check memory and disk space on the device to support fixes. You can also customize scripting to support different upgrade methods.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Upgrade process flow diagram



© Copyright IBM Corporation 2016

Upgrade process flow diagram

The following steps occur in an OS upgrade. During this upgrade, the OS upgrade UOW performs the following tasks:

1. Creates a configuration synchronization UOW to synchronize the device.
2. Logs in to the device and checks the disk space and memory availability. This step ensures that the device meets the requirements that the OS registry defines.
3. Checks the worker server to make sure it can connect to the FTP server that hosts the IOS image file. It also checks that the image file exists on the server.
4. Deletes the image file from the file system on the device to increase space for downloading the new image.
5. Obtains the raw CLI configuration from the database and applies it to the device. This step is for devices that have their configuration removed in step 4.
6. Sends the commands to the device to download the IOS image to the correct file system on the device.
7. Creates a command set UOW that modifies the configuration. This change tells the device to use the new IOS image on the device.
8. Sends a command to the device to restart it so that the new IOS loads into memory.
9. Creates a final UOW to resynchronize the device so that it captures any new commands.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Note: You can adapt this process to other vendor devices when they follow the same types of steps of downloading an image file to the device and installing it using command-line arguments.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

System preparation

Ensure properly tested IOS *.bin files are available

Ensure that the IOS *.bin files are located on the FTP site and in the correct directory

Ensure that the device has access to the FTP file

Ensure that the altFtpInfo is correct in file transfer resource

Know where the IOS is located on the device file system

© Copyright IBM Corporation 2016

System preparation

You must prepare for automated upgrades. The IOS image file that the *upgrade manager* transfers to the devices must not be corrupted. The network device hardware must also support it. Research and test these requirements before you automate them. The software can download images that do not function because of these two issues.

It is also critical that the FTP server that hosts the IOS files is accessible to the network devices. You must place the FTP information in the **altFtpInfo** portion of the resolvable file transfer resource of the devices that you are upgrading.

You must also know which file system on the device contains the image. This issue affects Cisco devices because several different file systems are on their devices. Cisco devices can have file systems that are named, for example, flash, disk0, and slot0.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Required resources

OS registry resource

OS specification resource

File transfer resource updated with altFtpInfo data

OS upgrade device script, optional

© Copyright IBM Corporation 2016

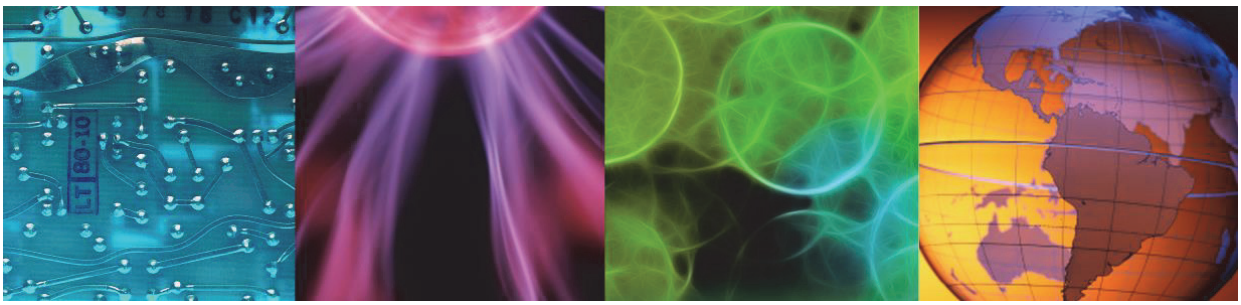
Required resources

To upgrade a device, you must create two resources, the model-specific OS registry resource and the operating system OS specification. You also must verify that the file transfer resource is correctly configured. Finally, depending on customer preference or device variations, you might need to create an OS upgrade device script.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 OS registry

Lesson 2 OS registry



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the OS registry file
- Configure the OS registry file

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

OS registry resource

An OS image catalog

Customer-defined list of approved images

IOS versions available for a specific VTM

- Image name
- Image size
- Memory requirements

© Copyright IBM Corporation 2016

OS registry resource

The OS registry resource is a document that defines all the operating systems that you can install on a particular device model. The customer defines this list of operating systems. It is good practice for you to minimize the number of operating systems that run on the network.

The OS registry defines the system requirements that a device needs to meet for you to load an image onto it. It does not define a comprehensive list of requirements for installing an image but covers two important ones: the memory that it requires and the disk space that it needs to host the image.

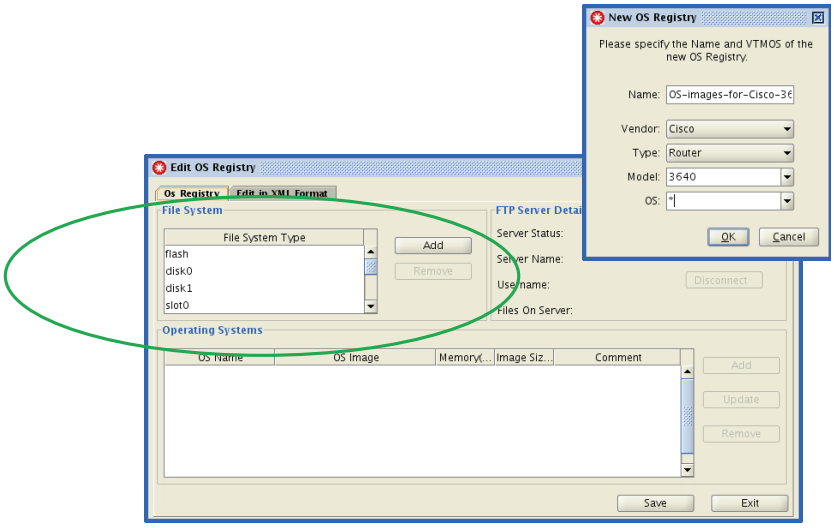
The OS registry is specific to vendor, type, and model. This requirement means that when you defined the resource, you must select, at least, a major model number from the menu.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring the OS registry

Create the OS registry and define it for a specific vendor, type, and model

Determine which file systems are supported on that model



© Copyright IBM Corporation 2016

Configuring the OS registry

The naming convention in this registry must reflect the specific model of devices it is used for. This convention helps users when viewing a series of registries in the GUI. You select the vendor, type, and model for the device the registry is used for. Defining the OS is not necessary because the registry is defined to handle multiple operating systems. The selections are used for downstream filtering so that registries line up with the devices upgraded. Selecting an OS can cause problems with this filtering later.

Unlike many other generalized resources that you must place in specific realms, you can place OS registries anywhere inside the realm structure. You might want to create an OS upgrade realm where the different OS registries exist and are protected from unauthorized users.

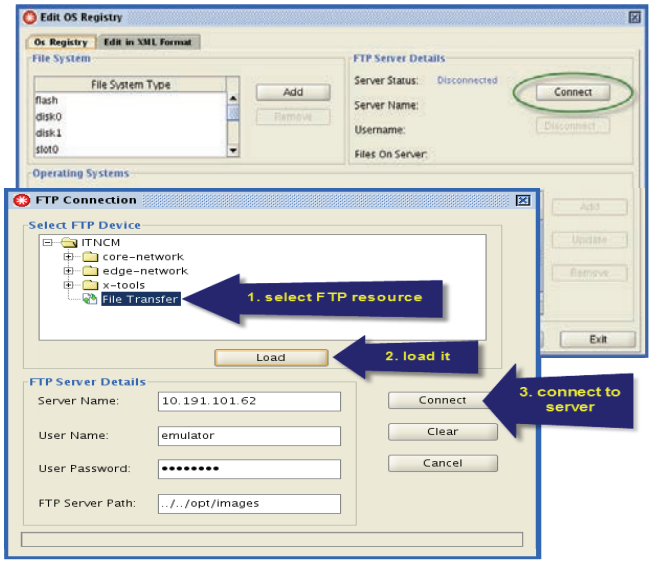
When you create the registry, you first want to determine what file systems are used by the device model that you are interacting with. The default list of file systems in the OS registry includes flash, disk0, disk1, slot0, and slot1. In many cases, for Customer Premise Equipment (CPE) gear, flash is the only file system on the router. Reduce this list to a minimum number of selections so that users do not select the wrong file system when they create an OS specification. Use the **Delete** function to remove the unused file system selections. Use the **Add** function to create a file system tag, if necessary.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Connecting to the image server

Connect to image server using file transfer resource

Connection occurs from GUI client



© Copyright IBM Corporation 2016

Connecting to the image server

The OS registry has functions to you use to connect to the image server where OS image files are stored. You can click the **Connect** button on the upper quadrant of the registry GUI to use this feature. The tool uses the data in the file transfer resource to make an FTP connection to the image server. Be sure to select the file transfer resource your devices resolve in normal UOW actions.

When prompted, you select the correct *file transfer resource* to complete the connection. Selecting the correct file transfer resource is important, depending on your implementation. You must configure the file transfer resource with the information it needs to connect to the image server for this function to work.

It is important to understand that this tool makes an FTP connection to the image server from the workstation where the GUI client is running. It is possible that firewall rules might prevent this connection.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

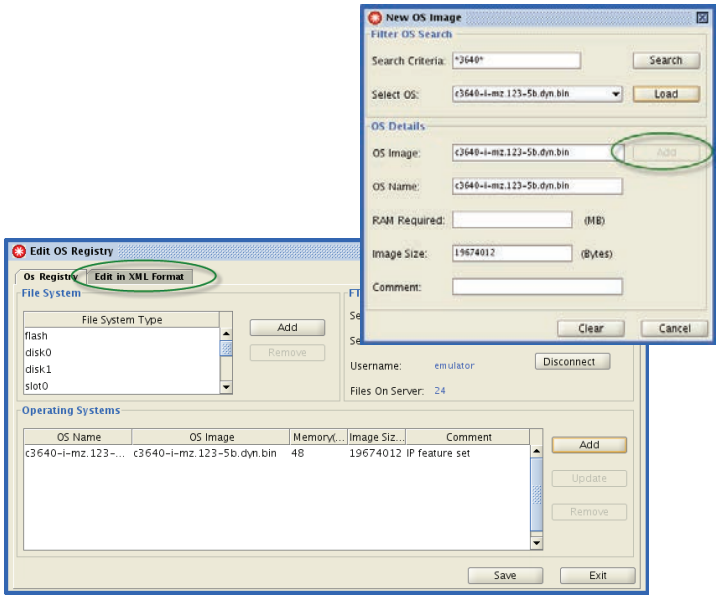
Loading the selected image

Search for correct image
Use search function to filter

Load selected image to get file size and correct name

RAM requirements are found in vendor documentation

Add image data to registry



© Copyright IBM Corporation 2016

Loading the selected image

After you connect to the image server, you can **Add** image file pointers to the OS registry. Selecting add opens a window where you can search the image server for specific files. You then select a file and load it. This process ensures that you define the file name correctly and the GUI correctly notes its image size. You are responsible for discovering the amount of memory that the image file needs to run. This information is usually available on the vendor's website.

When you are done defining the details of this image, you add it to your registry.

This process defines an entry into an XML document that defines the OS registry catalog. If you experience issues with connecting to the image server from the GUI, you can manually enter these details into the XML document with this same GUI by selecting the **Edit in XML Format** tab. The following text is an example of this XML document.

```
<?xml version="1.0" encoding="UTF-8"?><osregistry>
    <filesystems>
        <name>
            flash
        </name>
    </filesystems>
    <!--================================-->
    <os name="c3640-i-mz.123-5b.dyn.bin">
        <image>
            c3640-i-mz.123-5b.dyn.bin
        </image>
        <memory>
            48MB
        </memory>
```

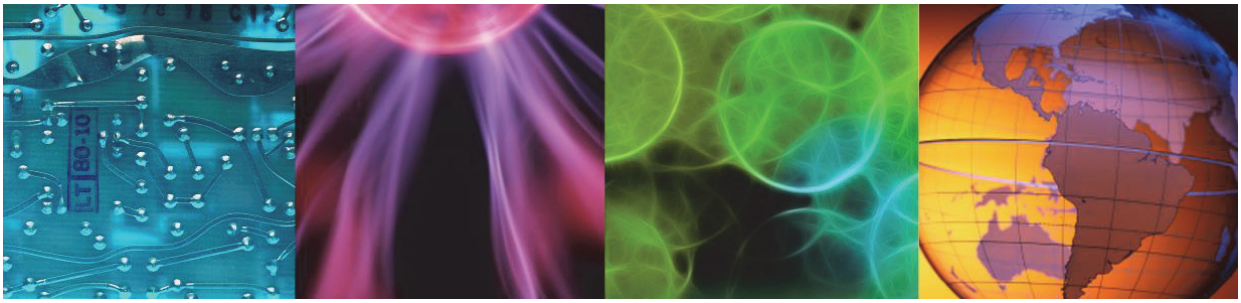

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

```
            <size>
                19674012
            </size>
            <commandset>
                <param name="FULL_IMAGE_PATH">
                    c3640-i-mz.123-5b.dyn.bin
                </param>
            </commandset>
            <details>
                IP feature set
            </details>
        </os>
        <!--==================================-->
        <os name="c3640-i-mz.124-25c.bin">
            <image>
                c3640-i-mz.124-25c.bin
            </image>
            <memory>
                64MB
            </memory>
            <size>
                12371292
            </size>
            <commandset>
                <param name="FULL_IMAGE_PATH">
                    c3640-i-mz.124-25c.bin
                </param>
            </commandset>
            <details>
                IP feature set
            </details>
        </os>
    </osregistry>
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 OS specification

Lesson 3 OS specification



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe an OS specification
- Configure an OS specification

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

OS specification resource

Connected to the OS registry resource

Defines the mechanics of the upgrade

Selected and applied to a device to upgrade its operating system

© Copyright IBM Corporation 2016

OS specification resource

After you complete the OS registry, you create an OS specification. An OS registry is a catalog of model-specific images. An OS specification is the resource that you select and apply to a device.

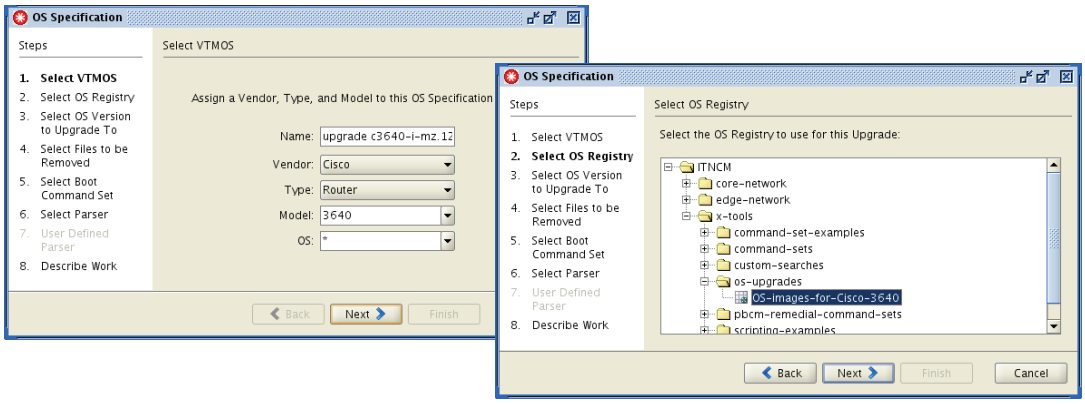
The OS specification takes an image from the OS registry and defines the mechanics of the upgrade. It also includes what data to parse from the device to verify that it meets memory and disk space requirements. The specification defines whether file systems can be erased to free up space for the download. It also supports the selection of an optional command set that modifies the configuration to change boot properties.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Specifying name and VTM

Create an OS specification and determine the VTM

Select the OS registry that has the catalog of approved images



© Copyright IBM Corporation 2016

Specifying name and VTM

When you create an OS specification, you must provide a unique name and vendor, type, and model (VTM) information. Operating system is not required because you are not concerned about the current operating system on the device but what it is after the upgrade.

For every model and image combination that you use the OS upgrade process on, you must have an OS specification. This requirement means having quite a number of specifications available at any one time. Using the image name in the OS specification is useful in locating it in the realm structure. Cisco's naming convention includes the device that the image is for, the operating system, and the feature set the image supports.

After you define the name and VTM, you select an OS registry that contains the image information. Remember that the OS registry is a catalog of image files. The specification selects an image from that catalog and uses the reference to upgrade a device to that specific OS. You must find the correct realm where the registry is located. The wizard presents only the registries that align with the model number that the specification is set to.

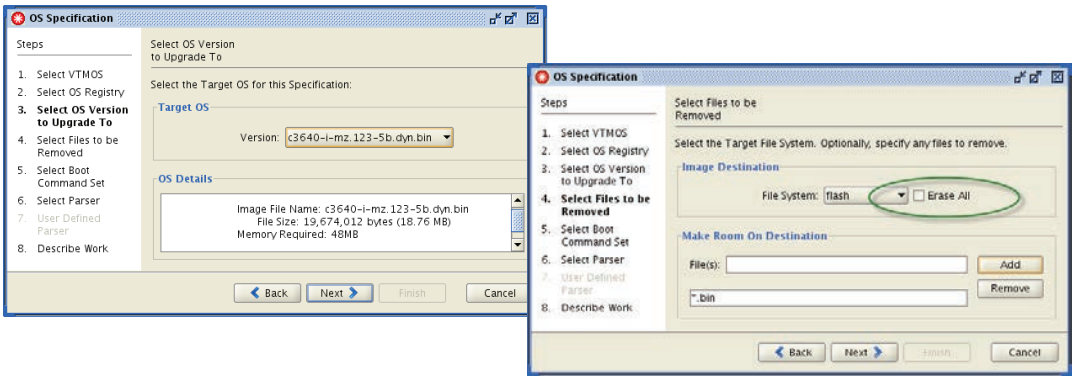
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Selecting the OS version and files

Select the specific image to which this specification upgrades

Select where to place the image

Indicate whether it can delete files from the device



© Copyright IBM Corporation 2016

Selecting the OS version and files

The next step in the specification definition is to choose the image from the OS registry that you select. This image is the one that is installed on the device if you apply this specification to the device in question. The image name can align with the name of the specification that you created in the first step.

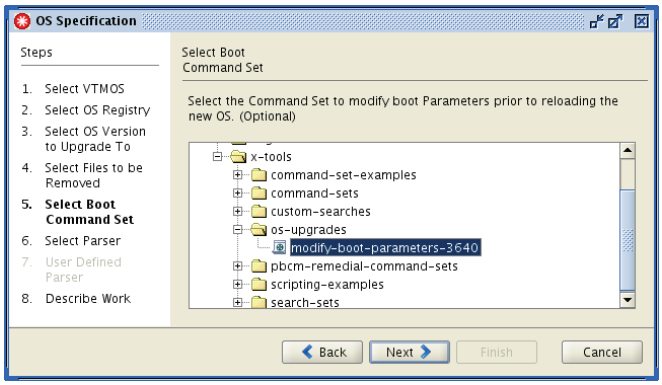
You must define which file system to place the image in and whether it can erase that file system. For example, you place the image file in flash. When you define the OS registry, you define a list of potential file systems for this particular model. Some larger Cisco devices have more than one file system, and an image might be in more than one location. If that is the case, the file system selection might have multiple file systems to choose from.

If the file system is to be erased, select the **Erase All** check box. An alternative is to remove specific files from the device. You can also use a regular expression with wildcards.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Selecting a boot command set

Indicate whether it can apply a command to modify boot parameters in the configuration file



© Copyright IBM Corporation 2016

Selecting a boot command set

Select an optional command set that defines boot parameters in the configuration on the device. These commands tell the operating system to look for a specific OS image file in a specific location.

These commands can often times be ignored. The boot loader on the device automatically searches for an image on the available storage systems and boots whatever it finds. You typically only have room for one image on smaller devices. However, on larger devices more than one image file can exist and so the configuration should point to a specific image to boot.

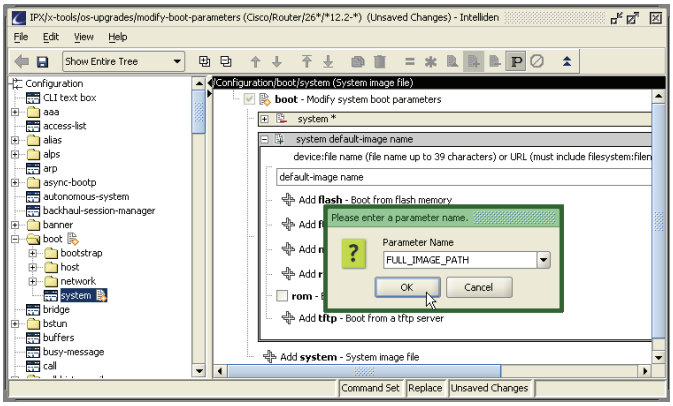
As with other parts of the OS upgrade process, you might want to keep the command sets in the same location as the other resources. This centralization makes it easier to work with all the resources needed to process upgrades.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Updating boot parameters

Upgrade process can use a command set to modify boot parameters

Command set must use the parameter FULL_IMAGE_PATH, which is defined in the OS registry



© Copyright IBM Corporation 2016

Updating boot parameters

If you chose to select a command set, you must configure it so that it uses a correct parameter name. This command set integrates into the specification and is used during the upgrade process to change the configuration after the image is downloaded.

The integration of the command set into the OS specification relies on the data that the OS registry contains. Inside the XML is a tag that is called <param> that is named FULL_IMAGE_PATH.

Each OS image that you define in the registry has this tag with the unique image file name. While the image file name is unique, the **name** is always equal to **FULL_IMAGE_PATH**. This label is a parameter name that is used exclusively by the boot system command set that you can optionally add to the specification. The boot system command set must have a parameter name that exactly matches the parameter name that you define in the registry.

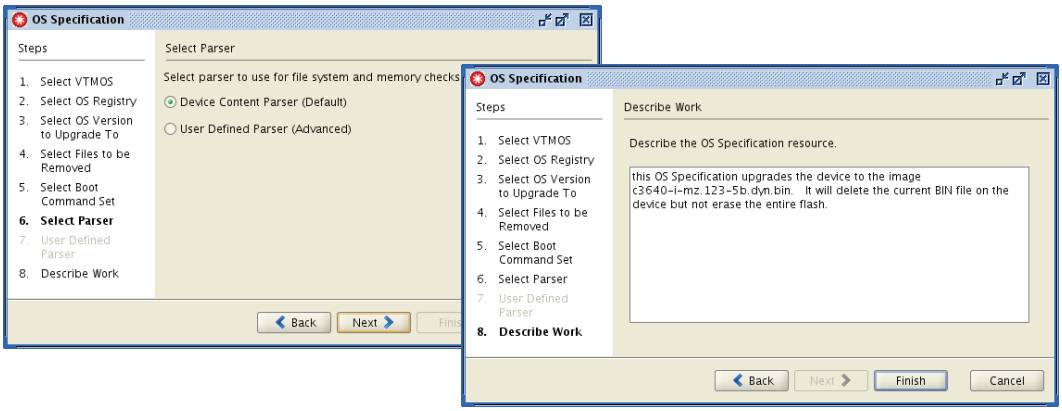
In the slide example, the command set is created to add the new image name to the configuration file. The value **default-image name** is a piece of default data that is replaced by the value that is assigned to **FULL_IMAGE_PATH** at execution time. The command set adds the command **boot system <value>** to the configuration. Note the **add** markup.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Selecting a parser and describing the work

Select the parser for determining file space and memory

Describe the OS specification



© Copyright IBM Corporation 2016

Selecting a parser and describing the work

The next step is to select a parser. You use these parsers to retrieve file system information and memory values that the device is using. By default, devices that are on the list of OS upgrades that are supported have a default parser that can retrieve this information.

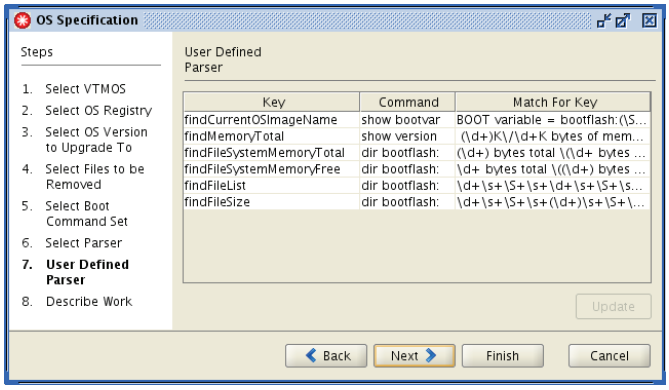
Finally, you must create a description of the OS specification. Describing the OS image upgrade and any other mechanics of the upgrade process is helpful.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Custom parsing

Custom parsing is the parsing of device information

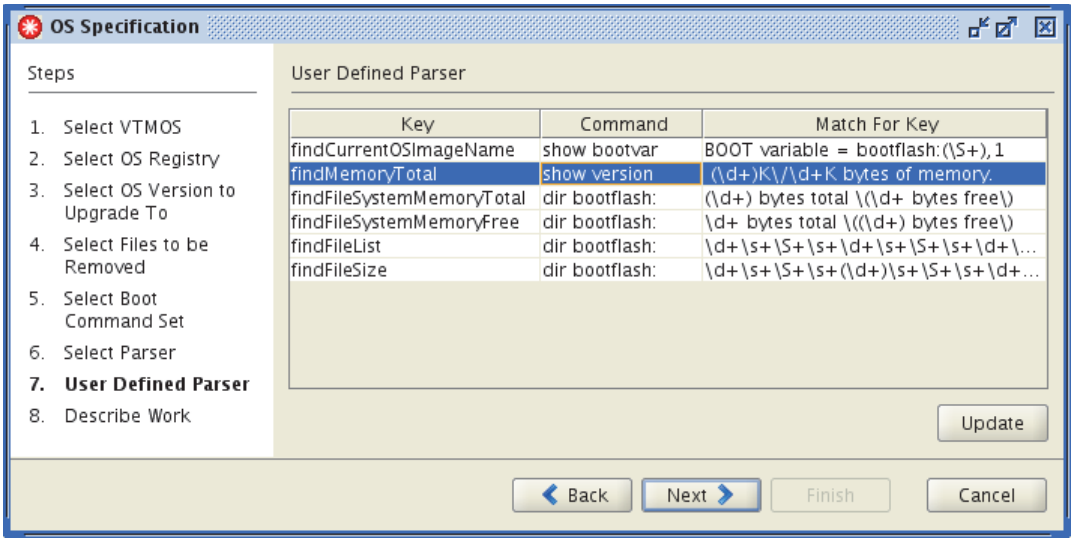
- To get total memory on a device
- To get total space on a device
- To list files that can be deleted



© Copyright IBM Corporation 2016

Custom parsing

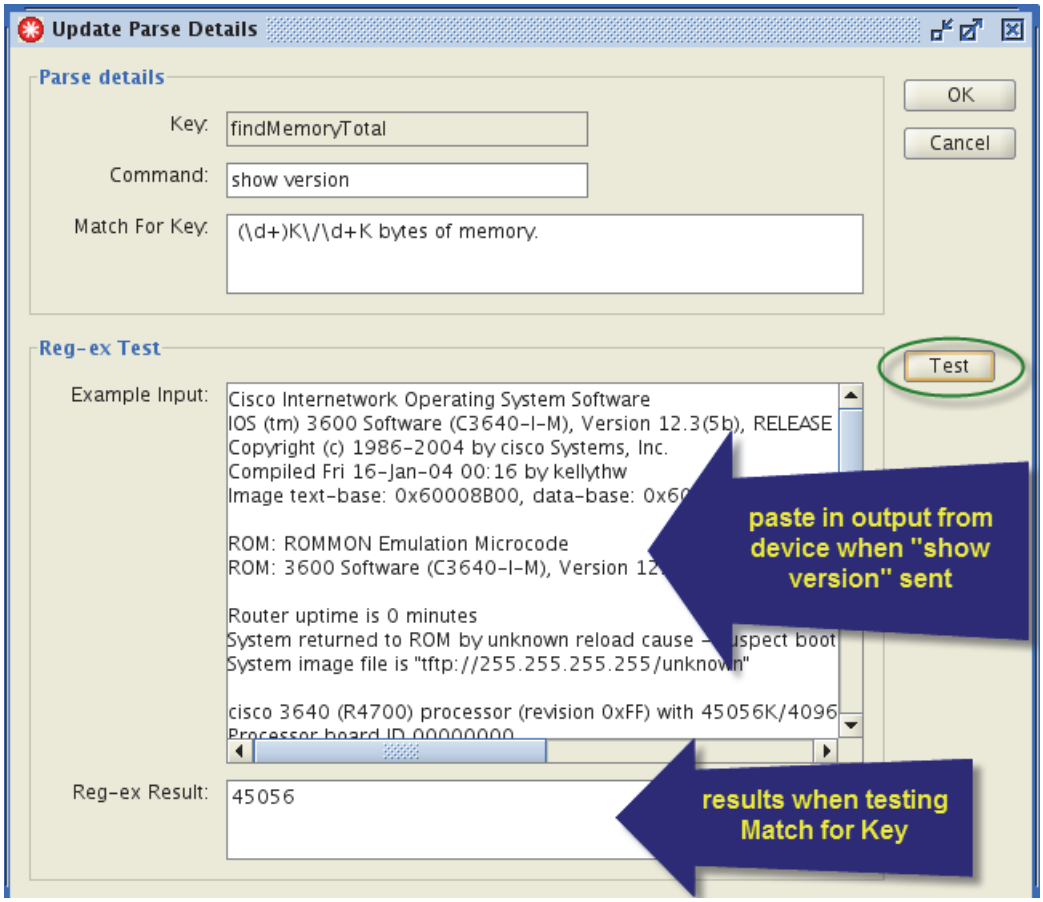
When building an OS specification, you can select whether to use standard parsers or custom parsers. When choosing to select a custom parser, the User Defined Parser pane is shown.



Select any of the values that you are parsing from the device and click the **Update** button. The system supports testing and updating the regular expression. This expression parses the data that

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

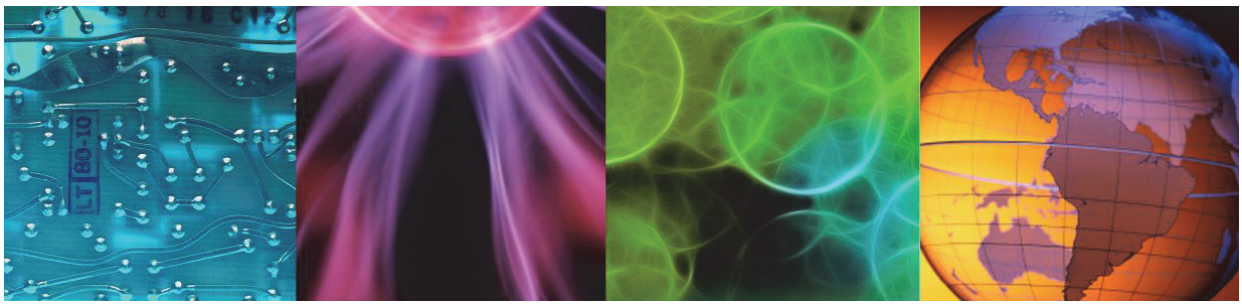
is retrieved from the command that you submit to the device. You can paste the output from the command and test the regular expression.



Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 Optional resources

Lesson 4 Optional resources



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Configure the file transfer resource for image server location
- Describe the OS upgrade script and when to use it

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Optional resources

File transfer resource altFtpInfo data, for defining a separate FTP server for downloading images

OS upgrade device script, for creating custom OS upgrade scripts

© Copyright IBM Corporation 2016

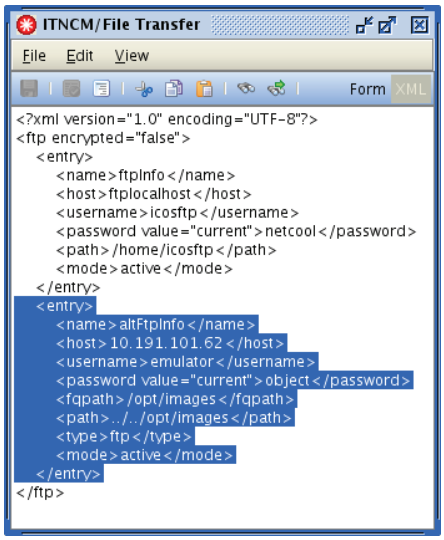
Optional resources

When you use a file transfer resource instead of the defaults in the database, filling out the altFtpInfo portion of the resource is straightforward. The creation of an OS upgrade script is only necessary when the default script in the driver is failing or you are creating an unsupported OS upgrade process for a unique device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using a file transfer resource

- You can use a file transfer resource to define a specific OS image server
- Use data in altFtpInfo for OS upgrades only
- Use data in ftpInfo for normal configuration management tasks
- You do not ever change <name>



© Copyright IBM Corporation 2016

Using a file transfer resource

Using the file transfer resource supports the definition of a separate FTP server to host images for use in upgrades. Only the OS upgrade process uses the data that you define in the altFtpInfo area. Do not change the names of altFtpInfo or ftpInfo, it generates errors.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Defining altFtpInfo data

```
<entry>
    <name>altFtpInfo</name>
    <host>10.191.101.62</host>
    <username>emulator</username>
    <password value="current">object</password>
    <fqpath>/opt/images</fqpath>
    <path>../../opt/images</path>
    <type>ftp</type>
    <mode>active</mode>
</entry>
```

© Copyright IBM Corporation 2016

Defining altFtpInfo data

The server that hosts image files is unique in the network. The worker server resolves the host name to its IP address and uses it in replacement variables. The network device that you upgrade must have access to this IP address.

The user name and password are the credentials to log in to the FTP server with the image files. Even if you use TFTP as the transfer mechanism, it requires a set of credentials so that it can determine the existence of the file.

The fully qualified path to the image file location is necessary so that its existence can be determined. The user name and password are used inside a remote method invocation to determine whether the file is on the server.

The path is used in the device script replacement variables. It is the relative location of the image files from the login location of the user name and password. In other words, if you manually access the FTP server, you are at a specific home directory, for example, **/home/emulator**. If the file is in /opt/images, the relative path from the initial login location is **../../opt/images**.

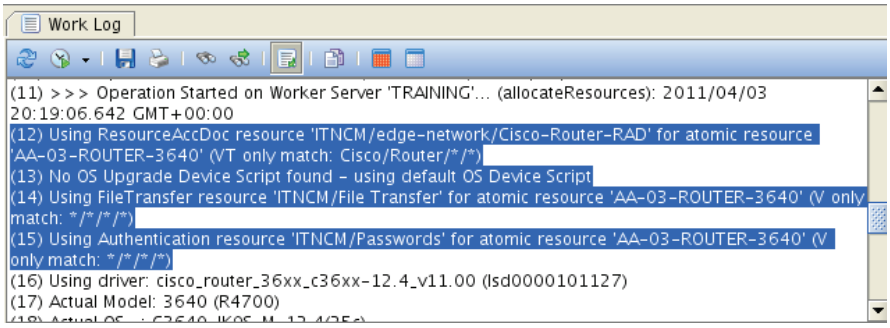
The *type* is the method that is used to transfer the image file. You can select TFTP, FTP, or SCP. The final definition determines whether the FTP connection is active or passive.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

OS upgrade device script

VTM-specific scripts to customize OS upgrade process

If no OS upgrade device script is resolved, it uses the default setting in the driver



© Copyright IBM Corporation 2016

OS upgrade device script

Devices that support OS upgrades have an OS upgrade device script that is defined as part of their drivers. Creating an OS device script occurs for one of two reasons.

- The device does not have official OS upgrade support and must resolve to an OS device script for it to function correctly.
- The customer is either having problems with the existing OS upgrade device script or is choosing a custom method to upgrade the device.

When a device is upgraded, the worker server performs an upward search for a resolvable OS upgrade device script. If one is not found, the worker server uses the OS upgrade device script that is in the driver.

Important: The OS upgrade device script is a separate script from the typical device scripts. The scripts are nearly identical, but use the OS upgrade device scripts for all aspects of the upgrade process.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

OS upgrade device script example

Customer requests different upgrade process for Cisco 3560 switch

You create an OS upgrade device script and modify the tftpcopyDown script to use commands that are different from the defaults

Ensure that you use the altFtpInfo in file transfer resource **<type>tftp<type>**

```
tftpcopyDown.01.send=archive download-sw /overwrite tftp://$ftp_althostname$/$copy_input1$\r
tftpcopyDown.02.polling=count=1000,timeout=2000,duration=2000,token=#
#tftpcopyDown.01.responseTimeout=2000000
#tftpcopyDown.02.send=copy tftp $cd_input$:$copy_input1$\r
#tftpcopyDown.03.wait=?
#tftpcopyDown.04.send=$ftp_althostname$\r
#tftpcopyDown.05.wait=?
#tftpcopyDown.06.send=$copy_input1$\r
#tftpcopyDown.07.wait=?
#####Script Truncated for Example########
#tftpcopyDown.29.send=boot config-file $cd_input$:intelliden.cfg\r
#tftpcopyDown.30.wait=#
#tftpcopyDown.32.send=end\r
#tftpcopyDown.33.wait=#
#tftpcopyDown.34.sleep=1000
```

© Copyright IBM Corporation 2016

OS upgrade device script example

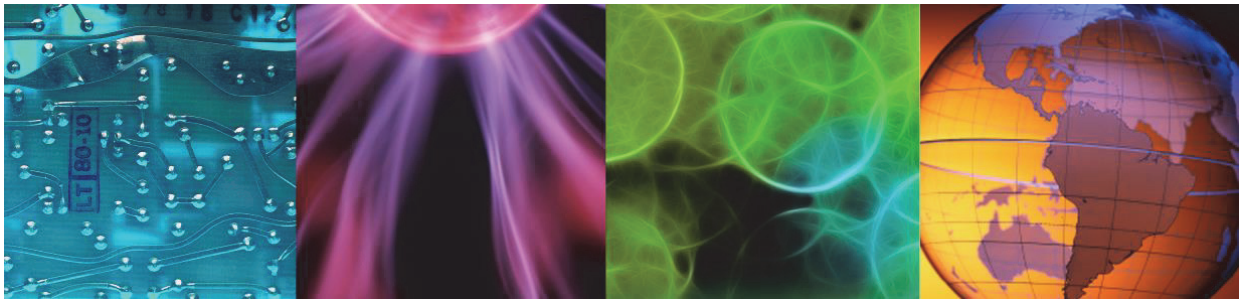
In this example, the customer chooses to use a different upgrade method that Cisco 3560 devices support. In the altFtpInfo, they choose to use TFTP, which calls the tftpcopyDown script. The customer comments out the script, and adds two commands. One command is to submit the upgrade command, and the other to poll for the pound (#) sign, which indicates that the process finishes successfully.

The customer chose this upgrade method because the particular device supports a more reliable method for upgrades that downloads the image into memory. If this process is successful, it then deletes the image on the disk and moves the new image from memory onto the flash drive.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 5 Launching an OS upgrade

Lesson 5 Launching an OS upgrade



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the OS upgrade UOW creation process
- Describe the results of the upgrade process

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Upgrading devices

Uses the OS specification

- Can run against realms or specific devices
- Can run in a *report only* mode

Executes as a UOW

- Might require approvals
- Can be scheduled

The UOW log is updated during the upgrading process

© Copyright IBM Corporation 2016

Upgrading devices

To upgrade a device, you select an OS specification and apply it to a realm or one or more devices. You can run it in *report only* mode, which checks the devices to verify that they have enough disk space and enough memory to authorize the upgrade.

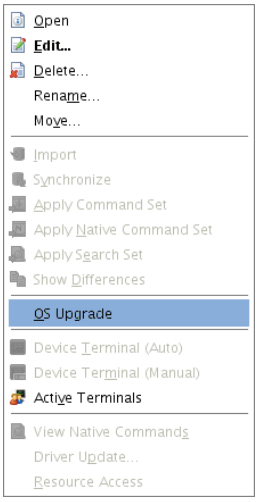
The OS upgrade is run as a UOW. Approvals are necessary, based on who submits the task. You can also schedule it.

As the OS upgrade processes it updates the UOW logs. You have a pseudo real-time listing of how the upgrade is progressing.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Performing an OS upgrade

Create the unit of work by selecting **OS Upgrade**



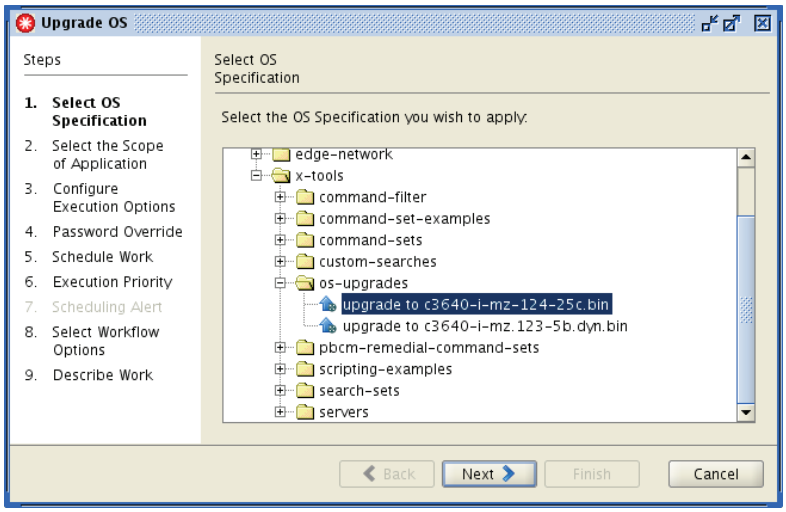
© Copyright IBM Corporation 2016

Performing an OS upgrade

First, you select **OS Upgrade** to create the unit of work. The Upgrade OS window opens.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Selecting an OS specification



© Copyright IBM Corporation 2016

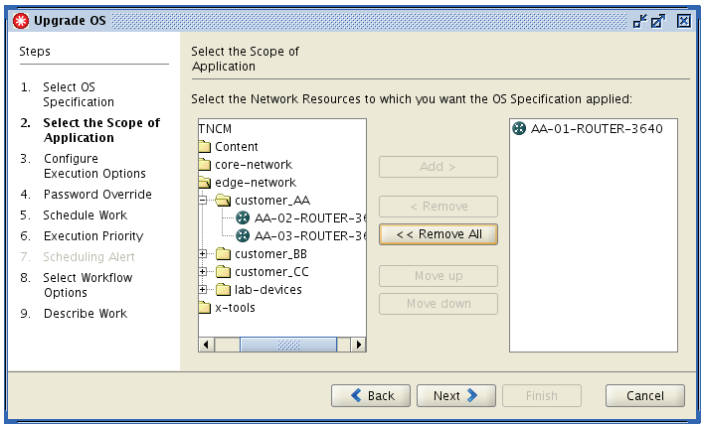
Selecting an OS specification

Select an OS specification. In this step, you can see why a good naming convention is important.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Selecting the device

Select the device that receives the upgrade



© Copyright IBM Corporation 2016

Selecting the device

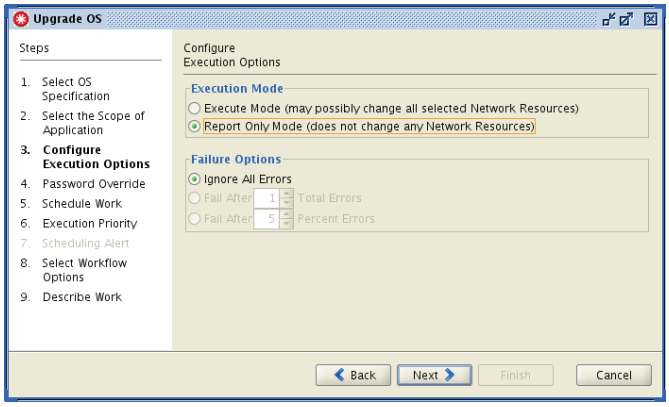
You can apply the OS specification to a realm, devices you select from a realm, or specific devices.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring the execution options

Select execution mode

Report Only Mode determines if device meets memory and space requirements



© Copyright IBM Corporation 2016

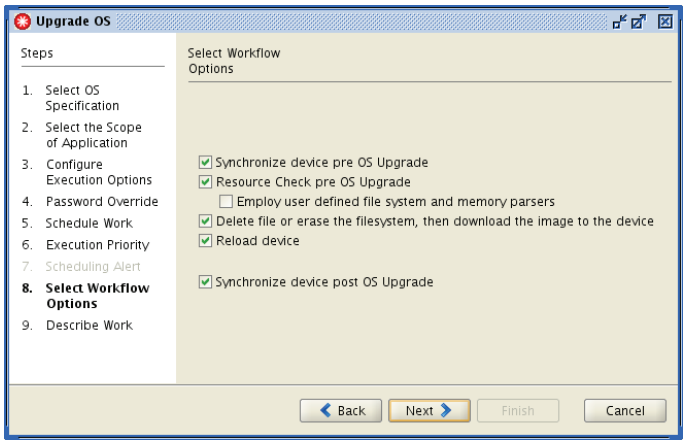
Configuring the execution options

Select the execution mode. The *report only* method can determine whether you can upgrade a series of devices to a new image based on memory and disk space requirements. Use this utility to determine whether hardware upgrades are necessary.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Selecting the workflow options

Select execution options
Some options are unavailable if you use report-only



© Copyright IBM Corporation 2016

Selecting the workflow options

Finally, select the execution options for the UOW. These options are applied to all the devices in the UOW:

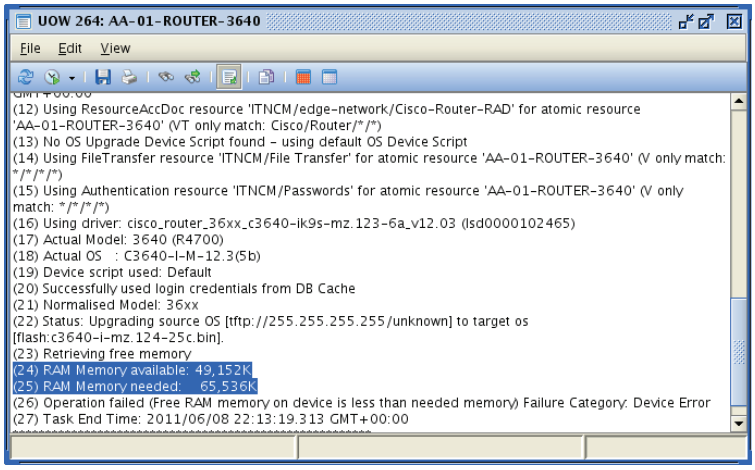
- **Synchronize device pre OS Upgrade**: This option synchronizes the configuration to verify that the device is ready to upgrade. This option not necessary when running in *report only* mode.
- **Resource Check pre OS Upgrade**: This option uses the parsers to determine whether enough memory and disk space exist. If custom parsers are created for the specification, select **Employ user-defined file system and memory parsers**.
- **Delete file or erase filesystem, then download image to the device**: This option deletes files or erases the file system based on the settings that are defined in the OS specification. Clearing this option downloads only the image and attempts to put it into the file system specified in the specification.
- **Reload device**: This option sends the appropriate commands to reload the device to use the new operating system.
- **Synchronize device post OS Upgrade**: This option synchronizes the device after the upgrade.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Monitoring execution

UOW log is appended during upgrade

Auto refresh the log to monitor progress



© Copyright IBM Corporation 2016

Monitoring execution

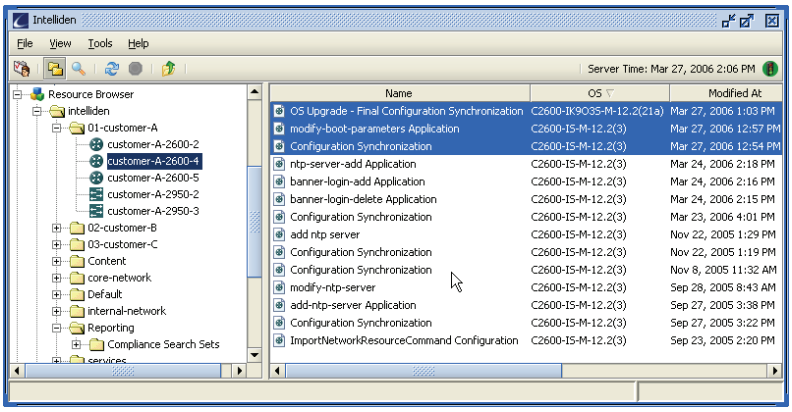
As the UOW runs, information about the upgrade process is appended to the log as actions occur. This feedback is a useful mechanism to monitor what is happening with each step that the upgrade goes through.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Completed upgrade

Creates two or three configurations

Note the change in OS



© Copyright IBM Corporation 2016

Completed upgrade

Depending on the workflow options that you select in the UOW wizard, you can end up with three new configurations in the database. One configuration is the synchronization that happens before the upgrade, the second configuration is the application of the optional command set, the third is the synchronization that happens after the device restarts the new OS.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Explain how the *upgrade manager* works
- Explain the requirements for a successful upgrade
- Modify an operating system registry
- Create an operating system specification
- Test the network for upgrade compatibility
- Verify the upgrade results

© Copyright IBM Corporation 2016

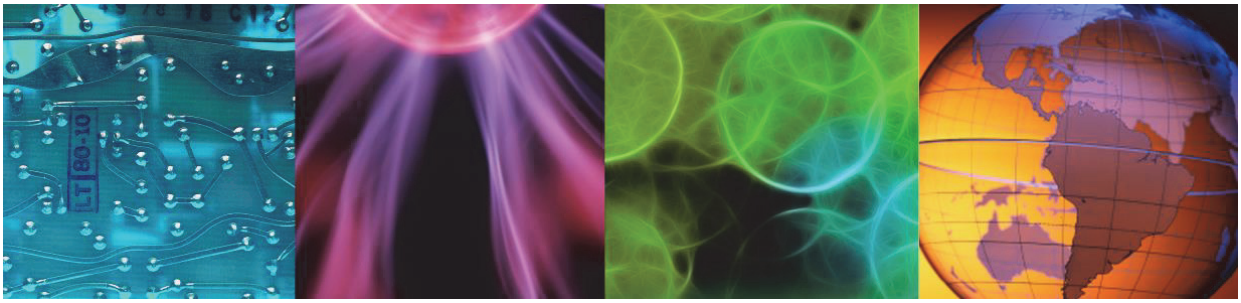
Summary

This unit was about using Netcool Configuration Manager to upgrade the operating systems on network devices. You learned the upgrade process and the objects you need to create to ensure a successful upgrade.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

14 Out-of-band change (OOBC) daemon

Out-of-band change (OOBC) daemon



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches you how to use the out-of-band change (OOBC) daemon. OOBC is an external application that monitors a syslog file and parses it for events that warrant synchronizing the device. You learn how to install and configure this utility to monitor any flat file that receives syslog messages from the network.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Explore how the out-of-band change daemon functions
- Install an OOBC
- Configure the daemon files
- Verify device synchronization
- Describe maintenance activities

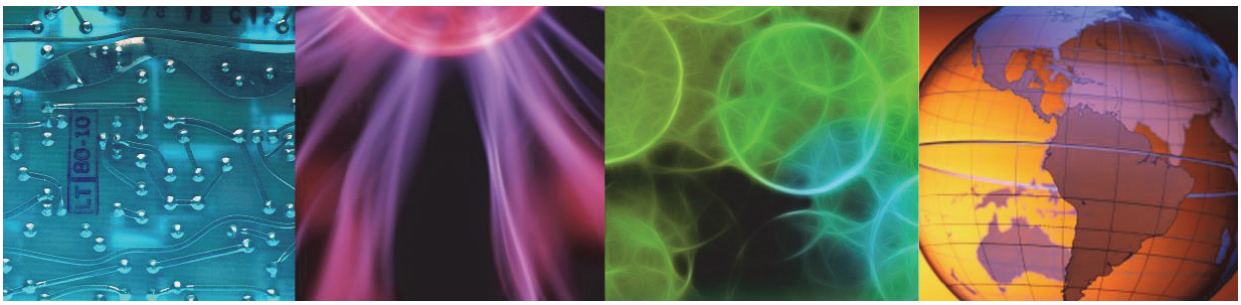
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Overview

Lesson 1 Overview



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe why OOBC is needed for the network
- Describe how OOBC picks up changes and analyzes them
- Describe the main features that can be configured

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Reasons for out-of-band change

You use out-of-band change for several reasons:

- Not all changes are processed by Tivoli Netcool Configuration Manager worker servers
- Not all VTY connections are made by the Tivoli Netcool Configuration Manager device terminal
- You can ensure that the configurations are always current within the database

© Copyright IBM Corporation 2016

Reasons for out-of-band change

The out-of-band change (OOBC) daemon is a tool that you can implement to capture changes that the Tivoli Netcool Configuration Manager software cannot monitor. OOBC addresses the following issues:

- Systems that initially implement Netcool Configuration Manager as a change mechanism in only one portion of the customer's infrastructure. Changes that Netcool Configuration Manager does not manage can occur If not all stakeholders in a network change use the solution.
- Netcool Configuration Manager device terminal does not manage all manual interactions with devices. When other VTY clients process device interactions, the session is not monitored.

The issue that arises from not using the Netcool Configuration Manager architecture for device interaction is that configurations can change, but the application does not monitor the change. Maintaining the current configurations in the database is important. This issue is especially the case when Netcool Configuration Manager implements changes, and the database and device must synchronize for the change to be successful.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Out-of-band change

A **syslog** event that comes from one of these sources:

- A nonworker server IP address
- An unauthorized user

An out-of-band change differs from an in-band change

- Out-of-band change syslog message

Jul 14 10:08:46 Sales_Lab_2600-3 11566: 7w6d: %SYS-5-CONFIG_I: Configured from console by lkoeser on vty0 (9.3.86.27)

- In-band change syslog message

Jul 14 15:28:53 Sales_Lab_2600-5 11567: 7w6d: %SYS-5-CONFIG_I: Configured from ftp://ftp:password@12.41.186.55/intelliden-Cisco1025558934493.cfg by itncm on vty0 (12.41.186.55)

© Copyright IBM Corporation 2016

Out-of-band change

The OOBC daemon is a utility that monitors syslog events. Syslog events are UDP logging events that are captured in a flat text file on a server that devices are sending messages to. Various levels are available when you enable logging on network devices. These levels mean that the device receives many log messages that reference different issues that the transmitting devices are reporting on. Only a small subset of these messages indicates that the configuration changed its state.

When a message of interest is found, it is parsed for various data items. These items include the user name of the individual who accesses the device and the IP address from which the connection is made. Using this information, this logic is used:

- If the IP address is present, it is compared to a list of worker server addresses. If it matches one of those addresses, the daemon assumes that the device is synchronized. If it does not match a worker server IP, the daemon schedules synchronization of that device.
- If the syslog event does not have a connection IP address, it selects the user name and compares it against a list of authorized user names. If the user name is authorized, it assumes that device synchronization occurred, and it ignores the message. If the user name is not authorized, it notes the event and the daemon schedules a synchronization.
- If neither of these data items are in the message, it notes the event and the daemon schedules a synchronization as a precaution to ensure that it saves any changes to the database.

Notice the difference between the out-of-band change and the in-band change messages. The assumption is that the worker server IP address is **12.41.186.55**, and the authorized user is **itncm.**

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

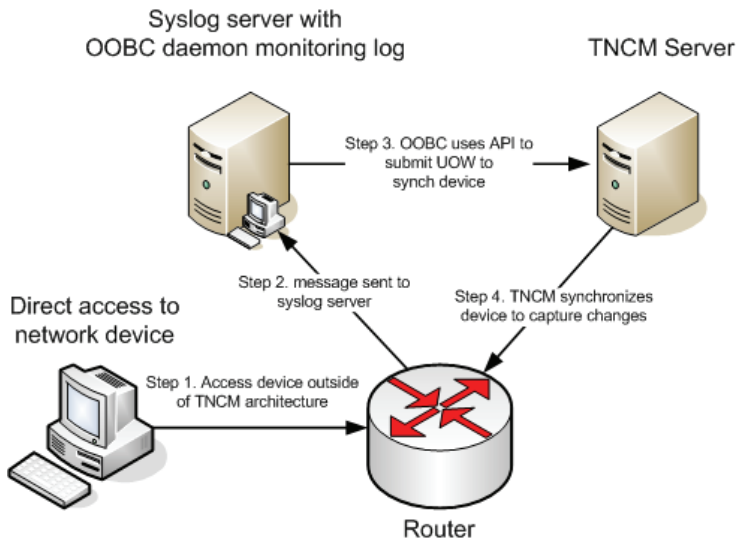
Important: Using the previous logic, the OOBC daemon ignores any VTY connection from a worker server. This situation implies that anyone who initiates Telnet/SSH sessions from a worker server is exempt from the OOBC daemon and the *units of work* that it creates. It is a potential security risk that someone might use to send unmonitored changes into the network. You must have tight controls on who can access worker servers.

Important: Another important point is that the device name in the syslog message must align with device name in the Netcool Configuration Manager database. In other words, resolve all IP addresses to the DNS name, and the DNS name is what the Netcool Configuration Manager registers. For example, a difference exists between the host name and a fully qualified host name.

Note: The device terminal always connects to a device from a worker server. Even though the user name is in the syslog message, if it includes the worker server IP address, it ignores the message. It assumes that the device terminal generates an appropriate synchronization.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

OOBC process



© Copyright IBM Corporation 2016

OOBC process

From an external perspective, the OOBC daemon is a piece of Java code that runs as a service on the server that collects syslog messages throughout the network. It is rare that the OOBC daemon runs on a Netcool Configuration Manager server because most organizations have a syslog server that they implement before installing Netcool Configuration Manager.

1. The process begins with a user, or application, logging in to a network device from a client computer that is not a worker server. This action comes from a desktop computer or an application server that applies changes to device configurations.

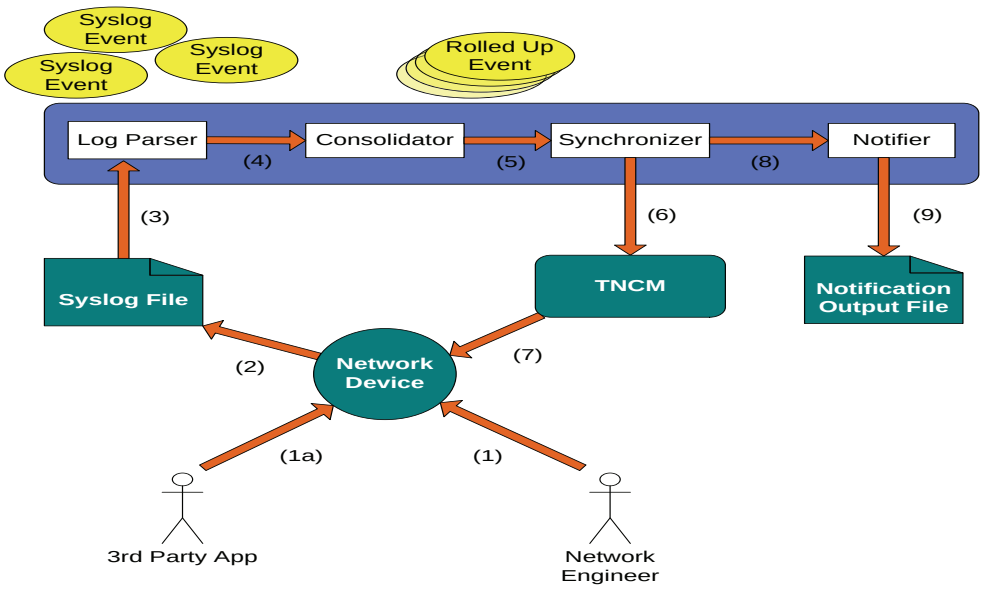
2. The network device is configured to send log messages when it detects a change to the configuration. This action has two implications:
   - The device is configured to send log messages and is sending them to the correct server.
   - The server receives the log messages. If the message uses UDP, it has an unreliable method of transmission.

3. The OOBC daemon uses its list of rules to parse through all log messages to find the ones that are significant. It finds the message that the router generates and parses information from. Using its logic, it notes that the message is an out-of-band change. It eventually connects to the Netcool Configuration Manager server through the API and submits a unit of work to ensure that the configuration is synchronized.

4. Netcool Configuration Manager processes the UOW, obtains the configuration from the device, and updates its database.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Data flow diagram



© Copyright IBM Corporation 2016

Data flow diagram

From an internal perspective, the daemon has logic to better control how Netcool Configuration Manager interacts with the device.

1. Steps 1 and 1a are obvious. A user or application logs in to a device and applies a change to the configuration.
2. The device sends a message to the syslog server, noting the occurrence of a configuration change event on the device.
3. At this step, the daemon uses its log parser and a series of log pattern data structures to sift through all the syslog messages. It retrieves the syslog change events that meet its criteria. Then, it applies its logic to each event to determine whether the event is an out-of-band change. If so, it notes the syslog event.
4. In this step, a consolidator takes all the noted syslog events from the log parsers and sorts them out for each individual device host name. In this sort, it creates a rolled-up event for all the messages that the device received. The next slide discusses the importance of this step.
5. The synchronizer parses each rolled-up event to find which event has the highest priority. The highest priority event determines which type of UOW is submitted against the device.
6. The synchronizer logs in to the Netcool Configuration Manager architecture through the API and creates the UOW.
7. Netcool Configuration Manager processes the UOW and retrieves the configuration from the device and determines whether updating the database with the new configuration is necessary.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

8. As a side effect, the synchronizer sends a message to the notifier about the action it took against the device.
9. The notifier sends a message to the notification log about the UOW it submits and the syslog messages it receives about the device.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Reasons for consolidating events

You consolidate to reduce the clutter of multiple syslog messages on the device

Consolidator must eventually submit a unit of work

- Timeout value
- Consolidation algorithm

© Copyright IBM Corporation 2016

Reasons for consolidating events

The consolidator is a useful tool that helps to reduce the number of *units of work* that are sent to a device. In a troubleshooting scenario, people might work on a device and generate many syslog messages. It might be resource-intensive to send a unit of work for each syslog event that is generated. The consolidator performs two functions:

- The consolidator takes multiple messages from the same host name and combines them into one message.
- The consolidator determines when to send a message to the synchronizer to log in to the API and submit the UOW. The criteria for this decision are configured in the OOBC properties file.

The consolidation strategy consists of two pieces of information:

- Timeout value that is set in seconds
- Consolidation algorithm

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Event consolidation algorithms

Static interval: Synchronizes every 30 minutes regardless of when the rolled-up event occurs

Dynamic interval: Synchronizes every 30 minutes after the first rolled-up event occurs

Idle timeout: Synchronizes every 30 minutes after the last rolled-up event occurs

© Copyright IBM Corporation 2016

Event consolidation algorithms

The OOBC daemon has three consolidation algorithms that it uses.

The static interval consists of a single timer that starts as soon as the OOBC process is starts. If the OOBC process starts at the beginning of the hour, synchronizations occur at regular intervals that are based on the timeout value. With a default of 30 minutes, it submits synchronization *units of work* at the top and bottom of every hour.

The dynamic interval creates a timer for each individual device that it sees an event for. The timer begins as soon as it notes the first event for a device. The consolidator sends a message to the synchronizer after it reaches the timeout value.

The idle timeout is used frequently in production and reduces the number of *units of work* to a minimum. Like the dynamic timeout, an individual timer exists for each device that starts at the first event that it notes for a host name. A countdown starts to send a UOW. Unlike the dynamic interval, this timer resets after each subsequent event that it notes for the device. If no event is seen after the configured timeout period, it can synchronize the device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Synchronizer actions

Actions determine the type of UOW action to send to Tivoli Netcool Configuration Manager

- Default actions are ConfigSyncIn, Import, and Reload
- Customized actions can be developed in Java

Consolidated events are *matched* against action rules

- Each log pattern has an action associated with it
- Priority flag is associated to action
- Highest priority action is submitted against the device

© Copyright IBM Corporation 2016

Synchronizer actions

The synchronizer determines from the rolled-up events in the consolidator what type of unit of work is submitted against a device. The synchronizer has three configured actions:

- **ConfigSynchIn**: Synchronizes the device, and updates the database with any changes to the configuration. Notes any differences in the UOW log.
- **Import**: Imports the device into the database. Importing the device does not compare the actual and database configuration, and it reports no differences to the UOW log.
- **Reload**: Applies a user-defined command set to the device, and synchronizes its configuration.

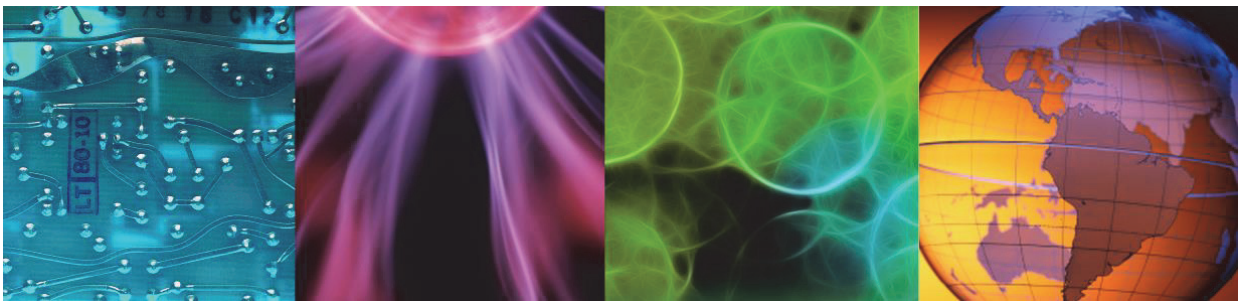
Because the OOBC code calls a Java class, you can develop a customized action that can interact with the Tivoli Netcool Configuration Manager API to process a custom set of steps.

The consolidator submits to the synchronizer all the rolled-up events on a device, which is based on the event consolidation algorithm. Each event has an associated action that tags to it when it matches a rule in the oobc.properties.xml file. A priority is associated with each action. The events are sorted by this priority, 1 being the highest. The highest priority event and its associated action define the UOW type that is submitted against a device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Installing OOBC

Lesson 2 Installing OOBC



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to install the OOBC daemon.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Installation prerequisites

You must install the OOBC where it has read access to a text-based syslog file

Each installation can read only one syslog file, can have multiple installations on a syslog server

Each installation requires 500 MB of space

Each installation requires 1 GB of memory

Application must connect to a Tivoli Netcool Configuration Manager presentation server, typically on port 7001

Requires a Tivoli Netcool Configuration Manager user name and password

© Copyright IBM Corporation 2016

Installation prerequisites

The OOBC daemon must be able to read the text-based syslog file. You typically install the daemon on the same server where the syslog messages are captured. If the syslog file is an NFS mount point, it is possible to read it from a separate server.

Each OOBC process can monitor only one file at a time. A server might have multiple installations to monitor multiple log files.

Each installation takes about 500 MB of disk space. Most of this space, 400 MB, is for logging. The installation also requires up to 1 GB of memory. This value is negotiable, depending on how many syslog messages it must parse in a second. Contact IBM support if memory on the syslog server is limited.

The OOBC process must be able to connect up to a Netcool Configuration Manager presentation server to use the API.

Finally, you need a user name and password to log in to the presentation server. Ensure that the user name and password are unique and reflect that the OOBC process submits the *units of work*, in other words, OOBCUser. By having a unique user to run an OOBC task, you can find those tasks in the queue manager. This method makes it easy to generate reports on how much OOBC activity is generated.

Verify that the user belongs to a group that has limited capabilities. The OOBC user needs only synchronization activities and a workflow policy of zero so that *units of work* require no approval. The group requires view rights on realms and view and execute privileges on resources.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Installing the OOBC software

Find the oobc.zip software, included in the Tivoli Netcool Configuration Manager Base tar file

Move it to syslog server and extract to the appropriate location

```
[icosuser@itncmdemo software]$ cd OutOfBandChange/
[icosuser@itncmdemo OutOfBandChange]$ ls -la
total 460
drwxrwxr-x 7 icosuser icosgrp   4096 Nov  8  2010 .
drwxr-xr-x 3 icosuser icosgrp   4096 Jun 10 11:05 ..
drwxrwxr-x 2 icosuser icosgrp   4096 Nov  8  2010 etc
drwxrwxr-x 2 icosuser icosgrp   4096 Nov  8  2010 install
-rw-rw-r-- 1 icosuser icosgrp     50 Nov  8  2010 ITNCMBASE0602.SYS2
drwxrwxr-x 2 icosuser icosgrp   4096 Nov  8  2010 lib
drwxrwxr-x 4 icosuser icosgrp   4096 Nov  8  2010 license
-rw-rw-r-- 1 icosuser icosgrp 432659 Nov  8  2010 OOBC_Install_Guide.pdf
drwxrwxr-x 4 icosuser icosgrp   4096 Nov  8  2010 tools
```

Change directory to install

Run the install.sh script as root

Software installs the executable software under a **run1** directory

© Copyright IBM Corporation 2016

Installing the OOBC software

To install the application, you must locate the OOBC software. This software is in the Netcool Configuration Manager base archive file that you download and use to install the Netcool Configuration Manager application. Move the oobc.zip file to the syslog server where it is installed.

Determine the location of the base file system, and extract the file. This action creates the OutOfBandChange directory. This file system is where the installer builds all the executable files during the installation process.

Move into the **install** directory and run the **install.sh** script as the **root** user. You require root privileges because start and shutdown scripts are placed in the RC directories. This step ensures that OOBC restarts if the syslog server restarts. While not a hard requirement, running the installer as a different user causes the installation to generate a message that it *fails.* This failure is probably a false positive, and the OOBC can start. If you install as root, the permissions on all the files are changed to reflect the user and group the installer prompts for.

This installation creates a **run1** directory, where it places all the executable files. If subsequent installations are necessary to monitor more than one file, it creates a **run2** directory, and others as necessary.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Installation questions

User and group owner of the software installation

Tivoli Netcool Configuration Manager presentation server host name and port

Tivoli Netcool Configuration Manager user name and password

User name for Tivoli Netcool Configuration Manager access to network devices

IP address of the worker server

Authorized user name of who can access network devices

Full path to the syslog file that is monitored

Full path to a file that saves all interesting syslog events

You can modify these details in the oobc.properties.xml file

© Copyright IBM Corporation 2016

Installation questions

The installer prompts you to answer a series of questions:

- **User and Group owner of the software installation**: This value defines a Linux user and group that is already configured on the server. This user and group require read access to the monitored syslog file. If installed as root, the permissions of all installation files are changed to these values.
- **TNCM presentation server host name and port**: This data directs the OOBC daemon to the Tivoli Netcool Configuration Manager installation. The default port is 7001.
- **TNCM user name and password**: Use these credentials to log in to Tivoli Netcool Configuration Manager and submit the UOW.
- **User name used for TNCM access to network devices**: This value is the user name that Tivoli Netcool Configuration Manager uses to log in to network devices. Typically, it is a

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

TACACS user name. If more than one user name is used for device connectivity, manually configure the oobc.properties.xml file after this installation.

- **IP address of the worker server**: This value is the IP address of the worker server.
- **User names that generate synchronizations but not notifications**: These user names modify devices that must be synchronized but do not send messages to notification.log. The installer refers to them as 3rdPartyUsers.
- **Full path to the syslog file that is monitored**: This path is the location of the syslog file that the OOBC daemon monitors.
- **Full path to a file that saves all interesting syslog events that generated UOWs**: This file captures all syslog events of interest.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Postinstallation

Installation is complete in approximately 5 minutes

Change the directory to **$OOBC_home/run1**

Start OOBC with the **./oobc.sh start** command

- Monitor $OOBC_home/run1/oobc.log for any errors
- OOBC reads entire syslog looking for events of interest

© Copyright IBM Corporation 2016

Postinstallation

After installation, move to the **run1** directory and start the OOBC process. Initially, it is useful to monitor the **oobc.log** that is in the **run1** directory to verify a successful start.

After the daemon starts successfully, it reads the entire syslog file. This step generates messages in the **oobc.log**, and more importantly, generates units of work. After it reads the entire syslog, it begins actively monitoring that log and polls it at regular intervals for any new messages. The daemon keeps a log marker to track the file that is read most recently. Any inconsistency between the log marker and the syslog file size indicates that the syslog file rolled, and OOBC must adjust.

In the following sample, you see a successful startup message in the **oobc.log** file:

```
WARN 10 Jun 2011 13:38:34 Activating com.intelliden.oobc.OutOfBandChangeDaemon
INFO 10 Jun 2011 13:39:09      Recovered 0 PRE rolled-up Events (with 0 child e
vents) from last run.
INFO 10 Jun 2011 13:39:09      Recovered 0 POST rolled-up Events (with 0 child
events) from last run.
INFO 10 Jun 2011 13:39:09       Started com.intelliden.oobc.NotifierThread
INFO 10 Jun 2011 13:39:09       Started com.intelliden.oobc.RollupThread
INFO 10 Jun 2011 13:39:09       Started com.intelliden.oobc.ParserThread
WARN 10 Jun 2011 13:39:09 com.intelliden.oobc.OutOfBandChangeDaemon now Active.
INFO 10 Jun 2011 13:39:09

=====================================================
ITNCM Host:               itncmdemo
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

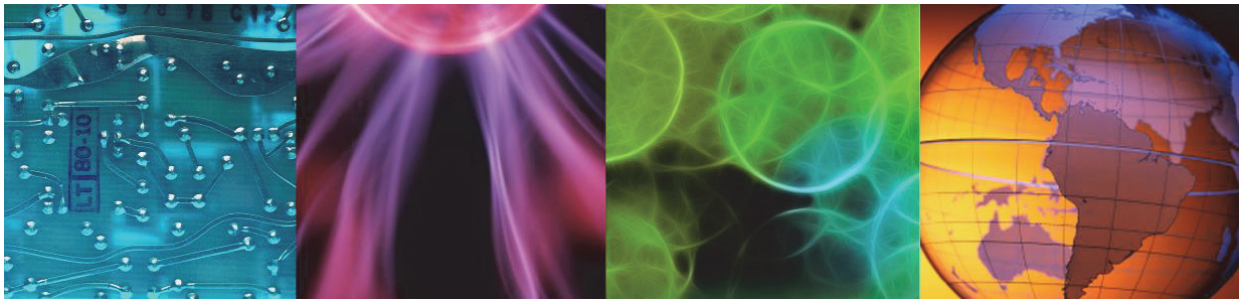
```
ITNCM Port:                7001
ITNCM Protocol:            iiop
ITNCM User:                OOBCUser
Ignored host(Worker Server):    10.191.101.52
Input Log File:                /var/log/syslog
Input Log File Poll Seconds:   5
Marker File:                   /opt/OutOfBandChange/run1/log.marker
Recovery File(s) Prefix:       /opt/OutOfBandChange/run1/log.recovery
Notify on Unmanaged Device:    true
Event Consolidation Algorithm: IdleTimeout
Consolidation Freq Seconds:    30
Fatal Restart Seconds:         15
ITNCM User:                    intelliden
3rd Party User:                3rdPartyUser
Number of Log pattern rules:   6
Number of Notification Rules:  1
Number of Action Rules:        3
=====================================================
```

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Configuring OOBC

Lesson 3 Configuring OOBC



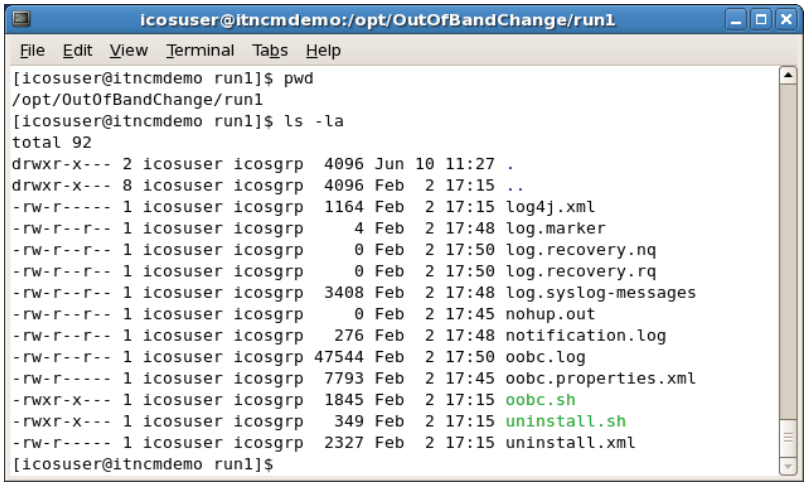
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Configure the daemon to handle multiple user names
- Configure the daemon to handle multiple worker servers
- Configure the daemon to handle different syslog messages

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Files in the execution directory



© Copyright IBM Corporation 2016

Files in the execution directory

The **run1** directory holds all the executable and log files that you need for review and troubleshooting. An important file is the **oobc.sh** script that you use to start and stop the daemon. Another important file is the **oobc.log** file that notes any trouble with the daemon and messages it retrieves from the syslog file. The **oobc.properties.xml** file is important as it has all the configuration settings that the daemon uses. Refer to the student notes for more information on other useful files in this directory.

These files are ones that you can configure to customize the software:

- **oobc.properties.xml**: You might need to configure this file after installation.
- **log4j.xml**: This file controls the logging format of the messages in the oobc.log and log.syslog-messages files. It also determines when to roll the log files. The current default is to roll them daily. You can change this file to roll them based on size.

These files are used for the monitoring process and syslog file:

- **log.marker, log.recovery.nq, log.recovery.rq**: These files monitor the last place in the syslog file that the application is read from. Also, use them to recover if the OOBC process fails. These files include any events that did not generate *units of work*, which are addressed at the next startup of the daemon.
- oobc.pid: This file contains the process ID number to run the OOBC process.

Use these files for logging purposes:

- **oobc.log**: This file has all the startup, shutdown, and error code information. This file captures all syslog messages of interest. Not all significant messages generate *units of work*. If one

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

does, information about the UOW and its submission is included. The startup messages are noted previously. The following messages are from a device that is configured out-of-band and not included in the Netcool Configuration Manager database:

```
INFO 10 Jun 2011 13:48:21 Jun 10 13:48:04 cc-03-router-3640 17: *Mar  1
00:03:02.119: %SYS-5-CONFIG_I: Configured from console by console
INFO 10 Jun 2011 13:49:39 UN-AUTHORIZED User console on cc-03-router-3640
INFO 10 Jun 2011 13:49:39 UN-AUTHORIZED Configured from console by console
INFO 10 Jun 2011 13:49:39 Taking action on event: cc-03-router-3640 | console |
Fri Jun 10 13:48:04 CDT 2011 |  console
INFO 10 Jun 2011 13:50:03 Did not find network resource: cc-03-router-3640
INFO 10 Jun 2011 13:50:03 Notify Device Not Found event: cc-03-router-3640 |
console | Fri Jun 10 13:48:04 CDT 2011
```

- **notifications.log**: Notifications are sent to this log when an OOBC daemon that is not an authorized user, submits a UOW. Messages are of the following format:

  <time-stamp> | <hostname> | <alarm> | <user name> | <# of events and last message>

  Note the alarms in the following sample message:

```
20110202171754Z|cc-01-router-3640|IntellidenAlarm|intelliden|3 of 3 syslog
events. Configured from console by intelliden
 on console
20110202174809Z|cc-02-router-3640|IntellidenAlarm|intelliden|1 of 1 syslog
events. Configured from console by intelliden
 on vty0 (10.191.101.1)
20110610134821Z|cc-03-router-3640|IntellidenDeviceNotFound|console|Device
cc-03-router-3640 not currently managed by ITN
CM. 1 of 1 syslog events. Configured from console by console
```

- **log.syslog-messages**: This log contains the raw syslog messages that are significant. They might or might not generate *units of work*. The following message is a sample:

```
Feb  3 12:05:40 cc-02-router-3640 16: *Mar  1 00:31:46.967: %SYS-5-CONFIG_I:
Configured from ftp://icosftp:*****@10.191.
101.52/int-cisco1296756336744.cfg by intelliden on vty0 (10.191.101.52)
Feb  3 12:07:35 cc-03-router-3640 17: *Mar  1 00:33:24.099: %SYS-5-CONFIG_I:
Configured from ftp://icosftp:*****@10.191.
101.52/int-cisco1296756450373.cfg by intelliden on vty0 (10.191.101.52)
Feb  3 12:08:58 cc-01-router-3640 14: *Mar  1 00:35:31.823: %SYS-5-CONFIG_I:
Configured from ftp://icosftp:netcool@10.19
1.101.52/int-cisco1296756533679.cfg by intelliden on vty0 (10.191.101.52)
Feb  3 12:10:06 cc-02-router-3640 20: *Mar  1 00:36:13.047: %SYS-5-CONFIG_I:
Configured from ftp://icosftp:*****@10.191.
101.52/int-cisco1296756602641.cfg by intelliden on vty0 (10.191.101.52)
Jun 10 13:48:04 cc-03-router-3640 17: *Mar  1 00:03:02.119: %SYS-5-CONFIG_I:
Configured from console by console
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Configuring OOBC

Most configuration is in the **oobc.properties.xml** file

- Specific syslog file is for monitoring and how often to monitor
- Consolidation algorithm and timeouts
- Presentation server details, user name, and password
- Worker server addresses
- Authorized users who log in to network devices
- The regular expressions used to find syslog messages of interest

© Copyright IBM Corporation 2016

Configuring OOBC

After the installation is complete, the daemon typically requires more configuration to run properly. Most of this configuration includes extra worker servers and user names.

For this extra configuration, you use the **oobc.properties.xml** file. This file is an XML file that defines everything that is important about the OOBC daemon. Any changes to this file require restarting the daemon.

This file contains the following data:

- The location of the syslog file to monitor
- The timeout value and consolidation algorithm to use
- The credentials and location of the presentation server
- The worker server IP addresses and device user names
- The regular expressions to parse the syslog server file for significant messages

These items typically require little change if correctly defined after installation:

- Syslog file to determine what and how to monitor, recovery files to use, the polling frequency, algorithm, and when to submit a UOW

```
<!-- Properties to configure what and how we monitor -->
<monitor>
   <logFile>/var/log/syslog</logFile>
   <markerFile>/opt/OutOfBandChange/run1/log.marker
   </markerFile>
```


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

```
    <syslogMessageSaverFile>/opt/OutOfBandChange/run1/
    log.syslog-messages
    </syslogMessageSaverFile>
    <recoveryFile>/opt/OutOfBandChange/run1/log.recovery
    </recoveryFile>
    <pollFrequencySeconds>5</pollFrequencySeconds>
    <rollupAlgorithm>IdleTimeout</rollupAlgorithm>
    <rollupFrequencySeconds>30</rollupFrequencySeconds>
    <fatalRestartSeconds>15</fatalRestartSeconds>
    <notifyOnUnmanaged>true</notifyOnUnmanaged>
</monitor>
```

- Netcool Configuration Manager server details and login credentials

```
<intelliden-server>
    <host>itncmdemo</host>
    <port>7001</port>
    <user>OOBCUser</user>
    <password>f62fb89d73347cb9fc8bdc9dedf5620e</password>
    <protocol>iiop</protocol>
    <initialContextFactory>com.ibm.websphere.naming.
    WsnInitialContextFactory
    </initialContextFactory>
</intelliden-server>
```

These three items require more configuration:

- Multiple worker server addresses
- Multiple Netcool Configuration Manager users and authorized users who change network devices
- Regular expressions to find significant syslog messages

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Addressing multiple worker servers

<worker-servers>

- Defines the IP addresses of worker servers
- Assumes any interactions from this IP address are by Tivoli Netcool Configuration Manager processes

Example syslog message:

Apr 17 11:53:24 test_3-2 12984: Apr 17 07:52:23.318 EDT: %SYS-5-CONFIG_I: Configured from console by vty0 (**204.16.213.15**)

```
<!-- List of Intelliden worker servers -->
<worker-servers>
            <server>204.16.213.15</server>
            <server>204.16.213.16</server>
</worker-servers>
```

© Copyright IBM Corporation 2016

Addressing multiple worker servers

Most implementations have more than one worker server. This list captures the IP addresses of worker servers. These addresses must align with the address that is in the syslog message itself, for example, NAT addresses. Add to or delete from this list as necessary. Any worker server not in this list can create two *units of work*, one that changes the device and one that is treated as an out-of-band change.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Addressing multiple authorized users

<syslog-users>

Indicates whether to send UOW when IP address is not available

Example syslog message:

Apr 17 11:53:24 test_3-2 12984: Apr 17 07:52:23.318 EDT: %SYS-5-CONFIG_I: Configured from console by **intelliden** on console

```
<syslog-users>
    <!-- Intelliden system users do work on behalf of Intelliden.
         Syslog entries that contain these user ID's are skipped. -->
    <intelliden-users>
        <user>intelliden</user>
    </intelliden-users>
    <!-- Third party system users do work that needs synchronization
         into Intelliden but nobody needs to be notified. -->
    <authorized-users>
        <user>3rdPartyUser</user>
    </authorized-users>
</syslog-users>
```

© Copyright IBM Corporation 2016

Addressing multiple authorized users

The <syslog-users> section of the **oobc.properties.xml** file addresses when a syslog message does not have an IP address. If Netcool Configuration Manager has a set of users that it uses to access devices, you can include them in <intelliden-users>. Because the Netcool Configuration Manager software is using these user names, assume that Netcool Configuration Manager is also synchronizing the database and OOBC actions are not necessary.

If other applications change the network, assume that they are using separate user names. You can capture those user names in the <authorized-users> section. Any syslog message that is missing an IP address but has an <authorized-users> user name generates a UOW but not a message in the notification log.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Adding pattern matching rules

<log-pattern>

Provides a search pattern for specific syslog messages

- <pattern> Regular expression
- Can contain up to five pieces of data extracted
  Time stamp, host name, user name, source IP (optional), description
- Specifies action upon successful match

```
<!-- this pattern matches the most common log messages like
        Apr 17 11:53:24 test_3-2 12984: Apr 17 07:52:23.318 EDT: %SYS-5-CONFIG_I:
        Configured from console by lkoeser (10.0.0.1)
    -->
<log-pattern actionName="ConfigSyncIn" notifyName="FileNotifier">
    <pattern>([A-S][a-u][b-y]\s+\d{1,
    2}\s+\d{2}:\d{2}:\d{2})\s+\[?([\.A-Za-z0-9_-]+)\]?[\.\S+]*\s+.*(Configured\sfrom.*by+(.*)+v
    ty+(.*)$)</pattern>
    <timeStampIndex>1</timeStampIndex>
    <dnsNameIndex>2</dnsNameIndex>
    <userIdIndex>4</userIdIndex>
    <descriptionIndex>3</descriptionIndex>
    <sourceHostIndex>5</sourceHostIndex>
</log-pattern>
```

© Copyright IBM Corporation 2016

Adding pattern matching rules

Unfortunately, the default oobc.properties.xml file does not contain a comprehensive list of regular expressions to determine whether it can create an action against a device. Most log patterns are based on Cisco routers, and even those patterns are not comprehensive. You can add log patterns to the properties files to manage any syslog message.

The regular expression retrieves one required piece of data and up to four optional pieces:

- **timeStampIndex**: Time stamp of the syslog message (optional)
- **dnsNameIndex**: Name of the device as resolved by the syslog server (required)
- **userIdIndex**: User name of the person who logs in to the device (optional)
- **descriptionIndex**: Syslog description (optional)
- **sourceHostIndex**: IP address of the source client that is logged in to the device (optional)

Also defined in the log-pattern is the actionName, which defines the action to take when the log pattern is matched. In most cases, this action is defined as **ConfigSyncIn**.

The following sample log pattern is created for this class. Use it for syslog messages when you use the console on the virtual devices.

```
<!-- this pattern matches the most common log messages like
     Jun 10 11:55:51 cc-03-router-3640 17: *Mar  1 01:43:29.643:
%SYS-5-CONFIG_I: Configured from console by console
-->
```


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

```
<log-pattern actionName="ConfigSyncIn" notifyName="FileNotifier">
   <pattern>([A-S][a-u][b-y]\s+\d{1,2}\s+\d{2}:\d{2}:\d{2})
   \s+\[?([\.A-Za-z0-9_-]+)\]?[\.\S+]*\s+.*
   (Configured\sfrom(.*)by\s(.*)$)
   </pattern>
   <timeStampIndex>1</timeStampIndex>
   <dnsNameIndex>2</dnsNameIndex>
   <userIdIndex>5</userIdIndex>
   <descriptionIndex>3</descriptionIndex>
   <sourceHostIndex>4</sourceHostIndex>
</log-pattern>
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Startup and shutdown

Single script for each runtime configuration

- **$OOBC_Install_Home/run1/oobc.sh {start|stop|restart}**
- Creates and uses **$OOBC_Install_Home/run1/oobc.pid**
- Simple **kill** causes an orderly shutdown

```
myhost# cd /opt/OutOfBandChange/run1
myhost#./oobc.sh start
myhost# cat ./oobc.pid
myhost#./oobc.sh stop
```

© Copyright IBM Corporation 2016

Startup and shutdown

After modifying the *oobc.properties.xml* file, you can start the application with the **oobc.sh** script. It takes the options of start, stop, and restart. Starting the daemon creates a file with the process ID that is kept in the **run1** directory until the process is shut down.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

User name or password

User name rarely changes

Password should change on a regular basis

```
myhost# cossun13% ./install.sh gen-pass
Enter clear text password:
mySecretPass
Encrypted password is ce6f3ea926ad712172f8f9d7a67ccc07
Edit the oobc.properties.xml file and update the password field.

BUILD SUCCESSFUL
Total time: 7 seconds
```

© Copyright IBM Corporation 2016

User name or password

The only other maintenance the application might need is that you periodically change the password that the daemon uses to log in to the Netcool Configuration Manager API. This password must be hashed in the oobc.properties.xml file. When the OOBC user in the Netcool Configuration Manager changes its password, you must change the password in the oobc.properties.xml file or you receive errors in the oobc.log file. Use the install.sh script in the **OutOfBandChange/install** directory that is created when you expand the installation files. It has an option of gen-pass that takes a clear text password as an input and generates the hashed password. Replace the encrypted password in the oobc.properties.xml file with this new hash and restart the daemon.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Explore how the out-of-band change daemon functions
- Install an OOBC
- Configure the daemon files
- Verify device synchronization
- Describe maintenance activities

© Copyright IBM Corporation 2016

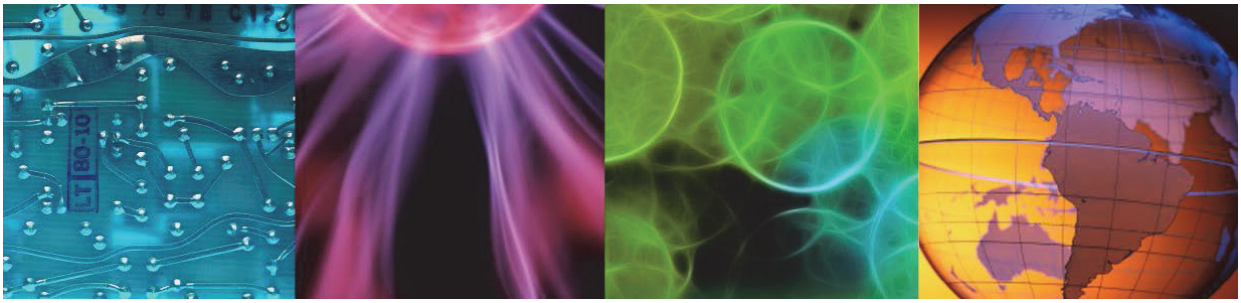
Summary

This unit provided the details of how to use the out-of-band change (OOBC) daemon. OOBC is an external application that monitors a syslog file and parses it for events that warrant synchronizing the device. You learned how to install and configure the utility to monitor any flat file that is receiving syslog messages from the network.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

15 Compliance manager interface

Compliance manager interface



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit is an introduction to the client graphical user interface (GUI) for the compliance manager. It provides an overview of the client technology and the way the software downloads to the client computer. You learn the main aspects of the compliance user interface.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Access and download the application
- Log on to the graphical user interface
- Browse through the main portions of the application

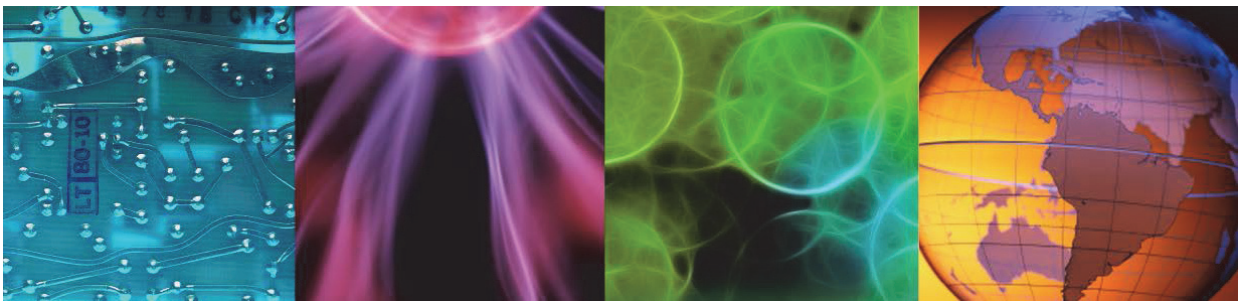
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Accessing the application

Lesson 1 Accessing the application



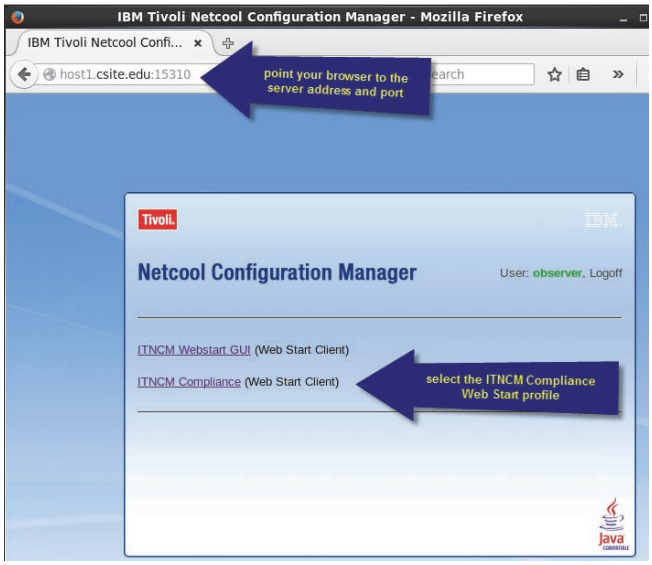
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Download the Compliance Manager GUI
- Describe the major features of the GUI

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Initial access through web browser



© Copyright IBM Corporation 2016

Initial access through web browser

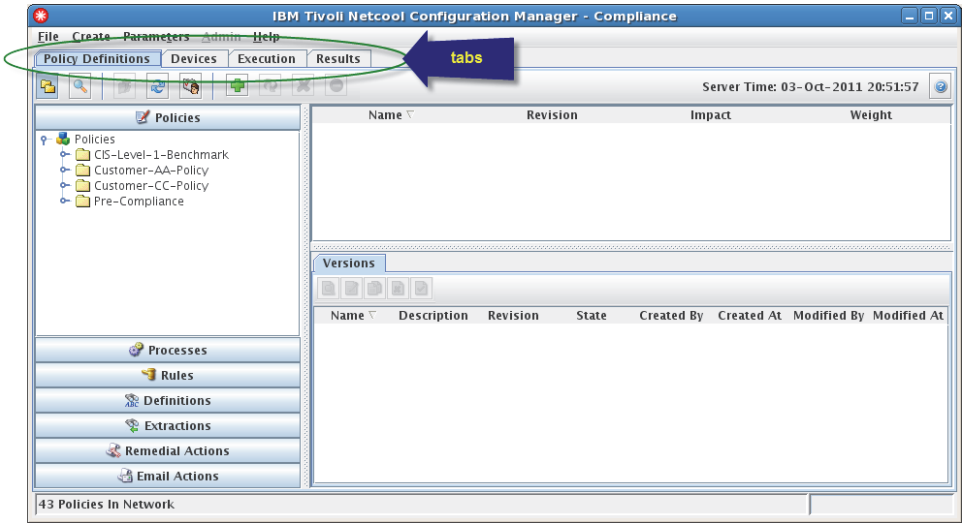
Using the same method that you use to access the Netcool Configuration Manager GUI, you point a browser to the Netcool Configuration Manager presentation server and the appropriate port and it presents two profiles. The top profile starts the Netcool Configuration Manager base client. For compliance, select the **ITNCM Compliance (Web Start Client)** profile to download the application and run it. The software downloads into the user's home file system.

The JRE downloads the files that the client needs. It might require you to accept certain security certificates before the GUI starts.

To authenticate to the GUI, use the same user name and password that you use for authenticating to the configuration manager.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The Compliance user interface



© Copyright IBM Corporation 2016

The Compliance user interface

When the GUI starts, it presents you with the following series of tabs along the top of the GUI:

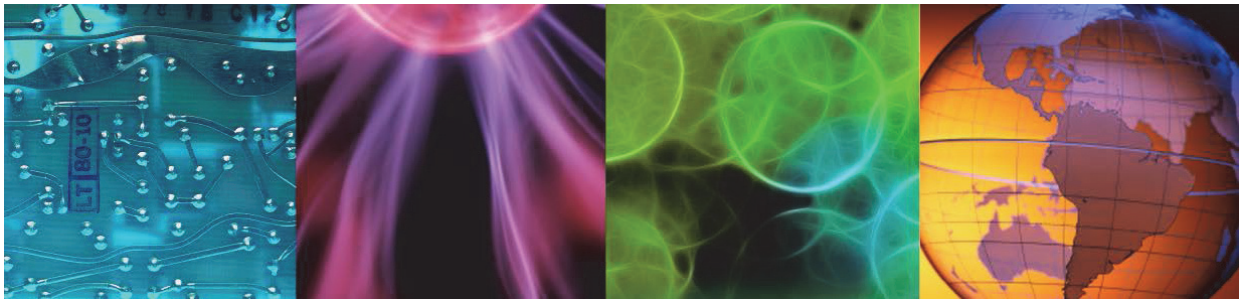
- **Policy Definitions**
- **Devices**
- **Execution**
- **Results**

The following lessons contain information about each tab.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 The Policy Definitions tab

Lesson 2 The Policy Definitions tab



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the **Policy Definitions** tab
- Describe how to create policy objects
- Describe how to test policies.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the Policy Definitions tab

Build *definitions* to search configurations

Create *rules* to determine noncompliance

Assemble *policies* to handle multi-vendor network

Build *processes* to run policies on regular basis

Allows testing of policies

© Copyright IBM Corporation 2016

Key points of the Policy Definitions tab

The **Policy Definitions** tab is where you build policies.

The advanced user creates a *definition* object, which searches through a data structure to determine the presence or absence of specific strings. The *rule* structure uses the *definition* to determine whether the presence or absence of the data means that the device is compliant or noncompliant.

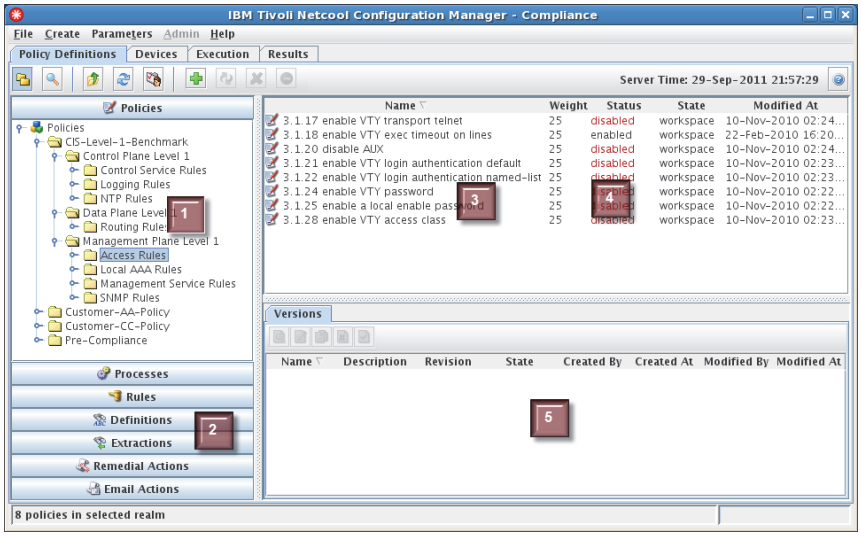
Rules are specific to vendor, model, and sometime operating specific device. Combining *rules* that are vendor-specific, but are logically the same, creates a *policy*.

After you develop *policies*, you gather them into a *process* structure. The *process* determines how often and when to analyze a series of devices for compliance purposes. When you run a *process*, all of the dependent policies, rules, and definitions also run.

The **Policy Definitions** tab allows users to test policies against the devices in the database.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The Policy Definitions tab GUI



© Copyright IBM Corporation 2016

The Policy Definitions tab GUI

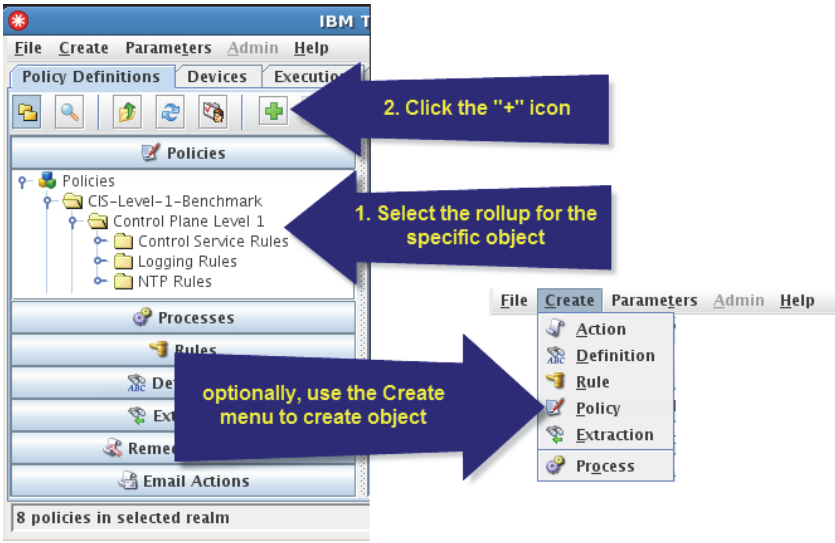
The **Policy Definitions** tab has the following panes:

1. This pane shows all the policy realms that the user has authorization to view. A user can browse through this structure to view the policies that are defined for the implementation.
2. Rules and definitions make up policies. They are also associated with extractions, remedial actions, and email actions. To view and create these structures, select a roll-up to show the policy realms and filter them for these specific objects.
3. Selecting a policy realm shows the objects that are defined in that realm. The slide shows policies because that is the roll-up currently active on the left.
4. Notice the status of these policies. Most of the policies are not enabled. These policies are not enabled to minimize the number of active policies for training. A status of *not enabled* means that a process cannot include the policy.

The lower pane is for showing all the versions of a policy. Selecting a policy shows its version history. Version control is not implemented for any of the policies in the training server.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating objects in the Policy Definitions tab



© Copyright IBM Corporation 2016

Creating objects in the Policy Definitions tab

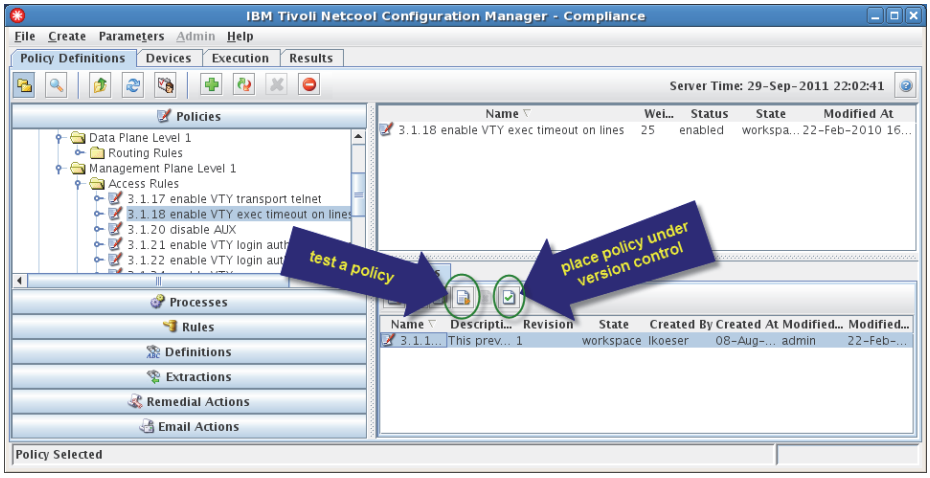
On the **Policy Definitions** tab, you create the various objects that you need to build a policy. You can build these objects in 1 of two ways.

1. Select the specific roll-up of the object you want to create. In the slide, the user selects the **Policies** roll-up. After that roll-up is selected, the user clicks the green **+** icon to create the object. This action starts a wizard that guides the user through creating the object.

2. Alternatively, users can use the **Create** menu at the top of the GUI. Selecting the **Create** menu allows the instantiation of the selected object. The wizard from step 1 is present here.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Testing and policy versions



© Copyright IBM Corporation 2016

Testing and policy versions

Testing of a policy is an important role for the **Policy Definitions** tab. Selecting a policy in the left pane allows it to show in the right pane. In the version pane in the lower right, a user selects the policy to test. Selecting the testing icon presents the user with a wizard to test the policy against one or more devices.

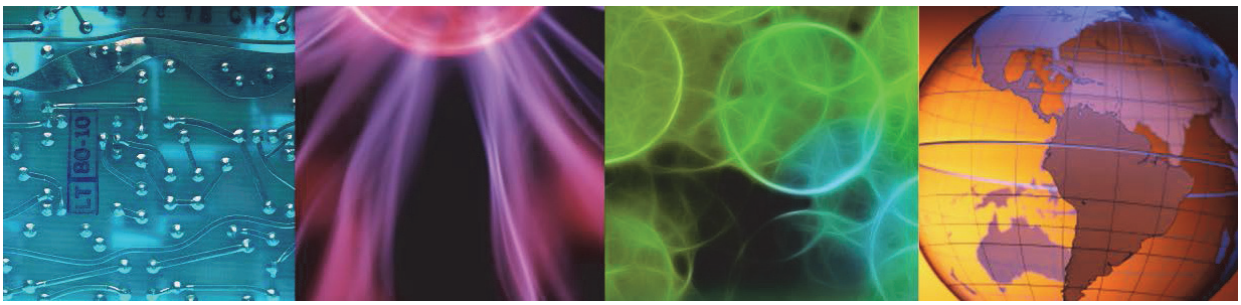
The **Policy Definitions** tab is where you can also place a policy under version control. After you place a policy in version control, it is always in version control unless you delete it. Refer to product documentation for an overview of version control.

Note: You must select the policy in the left window to view it in the version control pane. If you cannot see the policy in the lower right pane, you cannot test it. If more than one policy is shown in the upper right pane and you select one, it is not shown in the lower left pane.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 The Devices tab

Lesson 3 The Devices tab



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the available data in the tab
- Describe the security that is employed on the tab
- View the current compliance status of devices

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the Devices tab

- View device configurations
  - CLI
  - XML
  - Show commands
- Netcool Configuration Manager security is employed
- View by realm or VTMOS
- Show current compliance by device

© Copyright IBM Corporation 2016

Key points of the Devices tab

The **Devices** tab is where you can find the devices that you are authorized to view from a compliance perspective. The XML and raw CLI version of each device's configuration is available for viewing. Each device also has a series of **show** commands that are cached from the last time the *configuration manager* synchronized the device. You can use any of these data items for evaluating compliance policies.

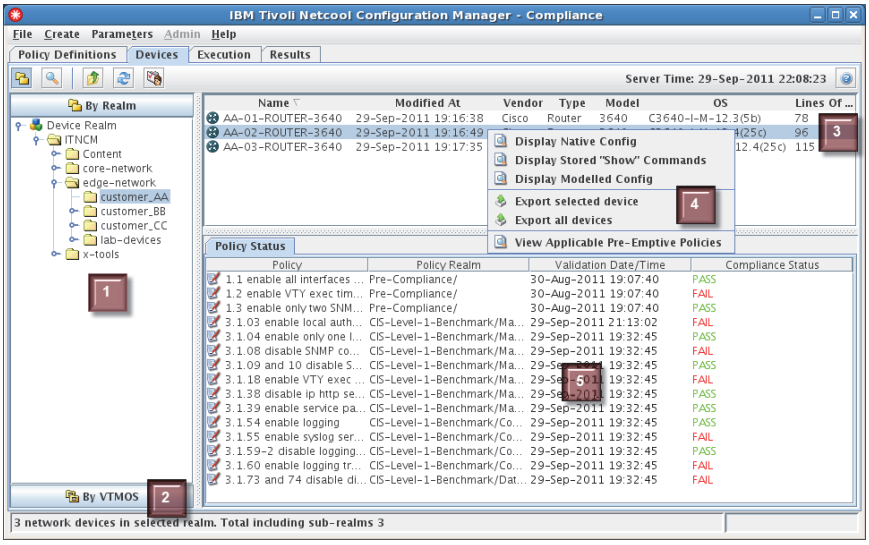
The group permissions in the *configuration manager* determine the authorization to view devices in *compliance manager*. If you cannot view the device in the *configuration manager*, you cannot view it in the *compliance manager*.

The GUI not only shows the devices by the realm structure that you define in the *configuration manager* but also by VTMOS. This view is useful when you define policies for specific versions of operating systems.

When you select a device, the **Devices** tab shows all the policies that process against it. This list shows where the device is and is not compliant with policy.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The Devices tab GUI



© Copyright IBM Corporation 2016

The Devices tab GUI

The following features comprise the **Devices** tab GUI:

1. You can browse the Netcool Configuration Manager realm structure for the devices you are authorized to view. This structure is a one-to-one match with what the *configuration manager* presents. However, you can view only network resources. It filters out all other resources, for example, authentication resources and access resources.

2. You can choose to view the devices with the **By VTMOS** view. You see the devices that you are authorized to view organized by *vendor*, by *type*, by *model*, and then by *operating system*.

3. The device data is similar to what is shown in the *configuration manager*. One new item of interest presents the total number of configuration lines in the raw configuration. This value is useful information, processing times for both compliance and configuration management depend on configuration size.

4. Right-clicking a device opens a series of menu options. For each device, you can see the following information:
   - **The native configuration**. This menu option shows the raw configuration that it retrieves from the device.
   - **The stored show commands**. This menu option shows the command responses that it retrieves and stores from the device when an import or synchronization, and sometimes configuration change, occurs.
   - **The modeled configuration**. This menu option shows the XML representation of the device configuration.

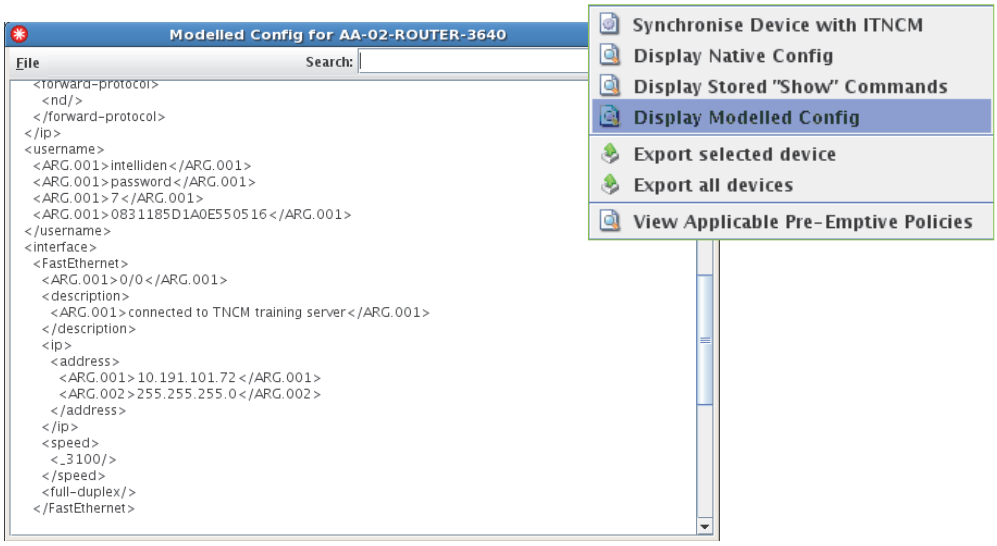
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

- **Exporting data**. This menu option exports the columns on the **Devices** tab to a CSV file for further review.
- **View pre-emptive policies**. This menu option shows the pre-emptive policies that are defined for this device. This option does not specify which groups the pre-emptive policies affect.

5. By selecting a device, the lower pane shows all the policies that process against the device. It also denotes whether the policy was *pass*, *fail*, *exempt*, or *not assessed*.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

View SmartModel configuration



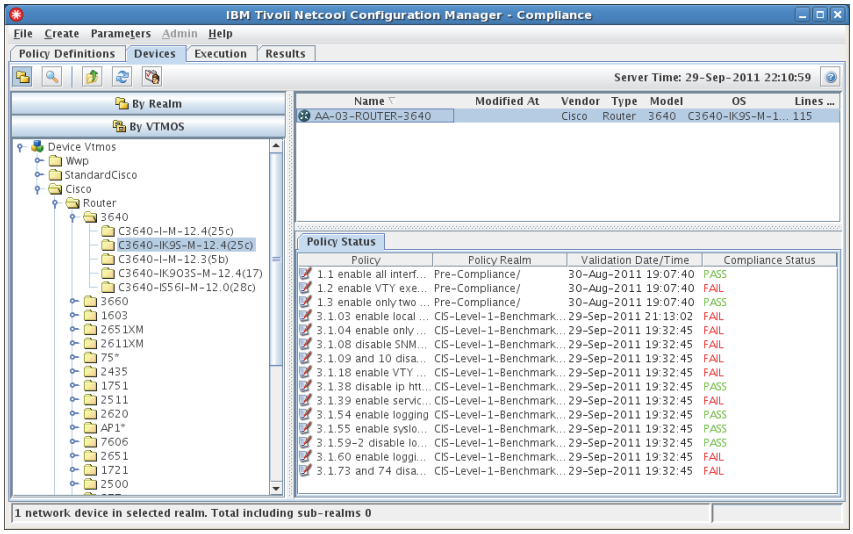
© Copyright IBM Corporation 2016

View SmartModel configuration

The *compliance manager* is the one place a user can easily retrieve the modeled version of a configuration. It is useful for you to view the modeled version of a configuration because you can use it to support security set development, search sets, and modeled definitions.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

View by VTMOS



© Copyright IBM Corporation 2016

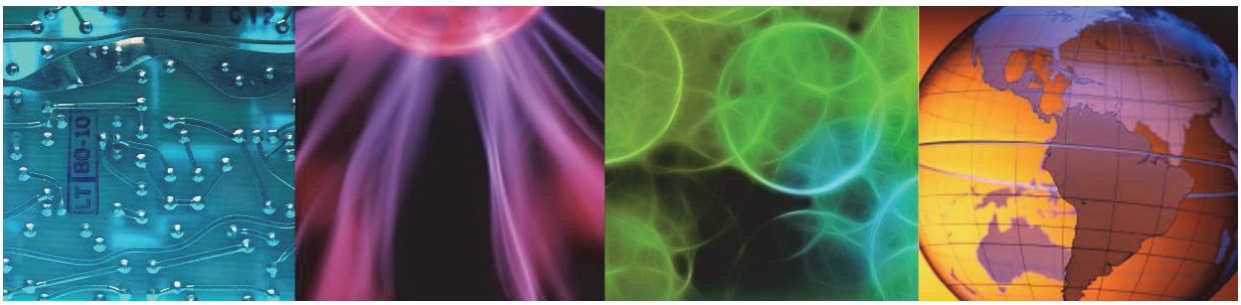
View by VTMOS

Viewing devices by VTMOS is useful in understanding how many different operating systems you have in your network. It is also helpful when you must create a policy to address a specific operating system. A vendor might send out a warning about a serious bug that is found in an operating system and provide workarounds. Ensuring the workaround is implemented is an example use case of a compliance policy.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 The Execution tab

Lesson 4 The Execution tab



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to run policies and processes.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the Execution tab

Start official compliance processes, generate side effects

Starts ad hoc and scheduled processes

Starts policies against devices or vice versa

© Copyright IBM Corporation 2016

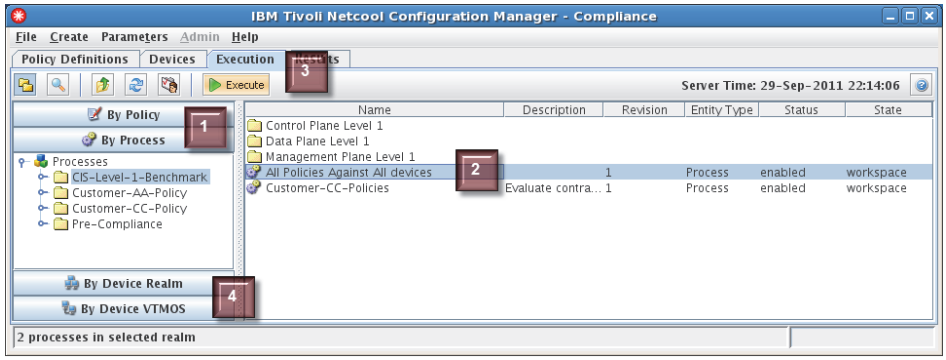
Key points of the Execution tab

The **Execution** tab is where users manually start policies and process. The user must have the appropriate group privileges to manage policies and processes.

If a process is defined, the user can manually start it from this tab regardless of how it is scheduled: unscheduled, scheduled, or recurring. Starting a process generates associated side effects such as impacting compliance scores, sending emails, and creating SNMP traps. You can select a series of devices in either a device or VTMOS realm to have policies that process against them.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The Execution tab GUI



© Copyright IBM Corporation 2016

The Execution tab GUI

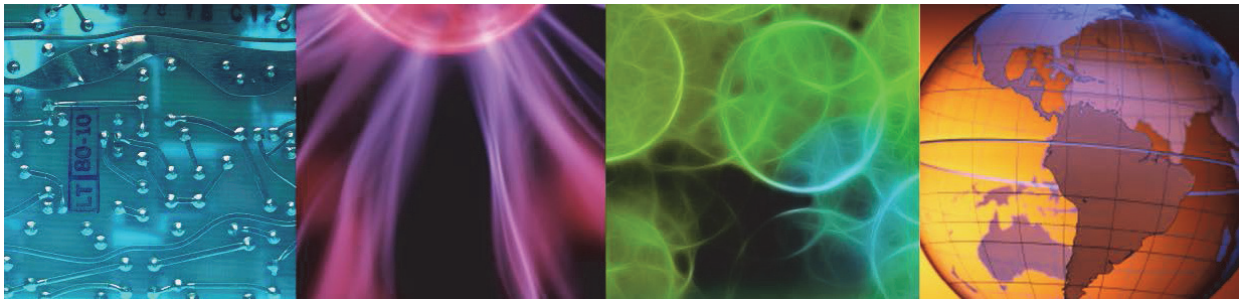
The **Execution** tab GUI has the following features:

1. A user with permission to process policies can either select a process or a policy for manual execution. Selecting the appropriate roll-up shows the policies or processes the user can select.
2. Select a process by browsing the policy realms on the left side of the GUI.
3. Clicking **Execute** starts the wizard that runs the policy or process. If it runs against a policy, the wizard prompts for device or VTMOS realms to run against it.
4. Alternatively, the user can select series of devices from a device or VTMOS realm and have a policy that processes against them.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 5 The Results tab

Lesson 5 The Results tab



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

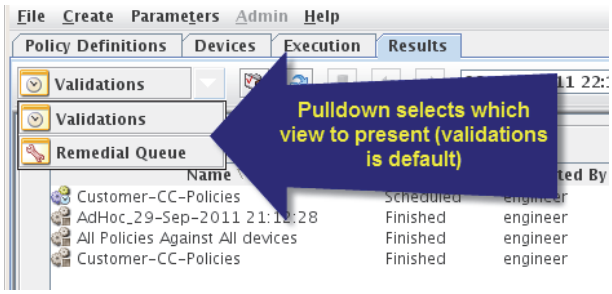
In this lesson, you learn how to perform the following tasks:

- Describe the details in the validations view
- Describe how to use the remediations queue to fix compliance violations

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the Results tab

- Shows details of validation activities
- Provides access to the remediation queue



© Copyright IBM Corporation 2016

Key points of the Results tab

The **Results** tab provides views of different features in the compliance GUI. These views include the following data:

- The validations view shows that the compliance server recently completed validation activity.
- The remedial queue view shows devices that have pending remedial actions.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the validation view

Shows the date and type of past validations

Shows data about the number of devices

Provides details of the execution

Details should be deleted on a regular basis (housekeeping)

© Copyright IBM Corporation 2016

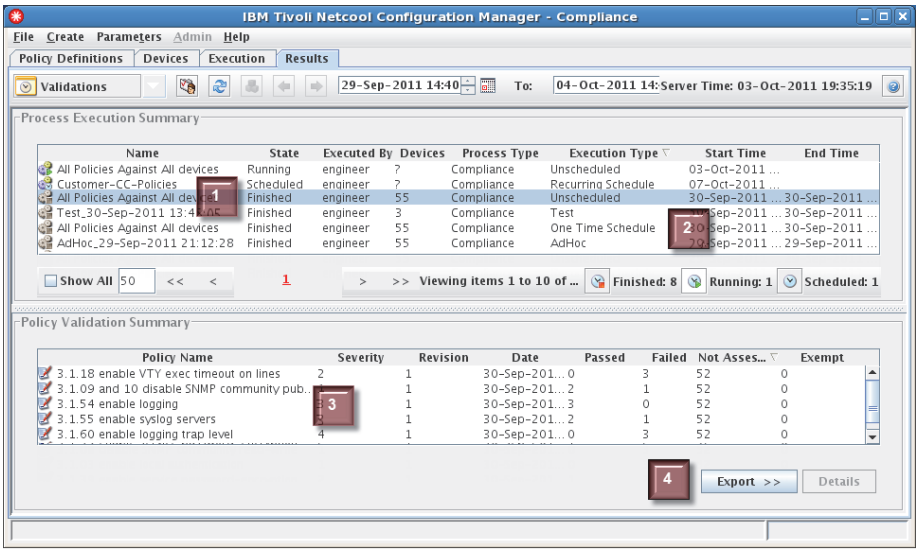
Key points of the validation view

The validation view shows when and what type of validations occur on the compliance server. By selecting a completed validation process, it shows the number of devices that are processed and the policies that are part of the validation. It gives the user the opportunity to see the details of the devices that either pass, fail, are exempt, or not assessed.

The details that are available in the validations view quickly become dated and you must remove them regularly. Housekeeping is done daily and is based on the housekeeping parameters that you define in the administration menu choices.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The validation queue GUI



© Copyright IBM Corporation 2016

The validation queue GUI

The following features make up the validation queue GUI:

1. It presents a list of recent validations. If you configure recurring processes, you might need to expand the dates of the search to see results. Select a validation to see its details. The GUI shows the total number of devices that are analyzed. Also, note the state of the validation. These states include:

   c. **Scheduled**: A validation that is scheduled to process at a future date.

   d. **Running**: When a validation is running, use the refresh icon on GUI to get updated results.

   e. **Finished**: Completed validations for that specific period. The results might already be dated and the pass/fail/exempted/not assessed values might no longer be correct.

2. The validation queue defines different execution types. The execution types include:

   a. **Recurring Schedule**: A process that runs on regular basis.

   b. **One Time Schedule**: A process that runs only once at a specific time and date.

   c. **Unscheduled**: A process that is unscheduled, and the user and must start manually from the **Execution** tab.

   d. **Test**: You select a policy to run against a set of devices. If version control is implemented, you can test an earlier or working version of a policy.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

   e. **Ad hoc**: You select a policy to run against a set of devices. This policy is the current or activated version of the policy, if version control is implemented.

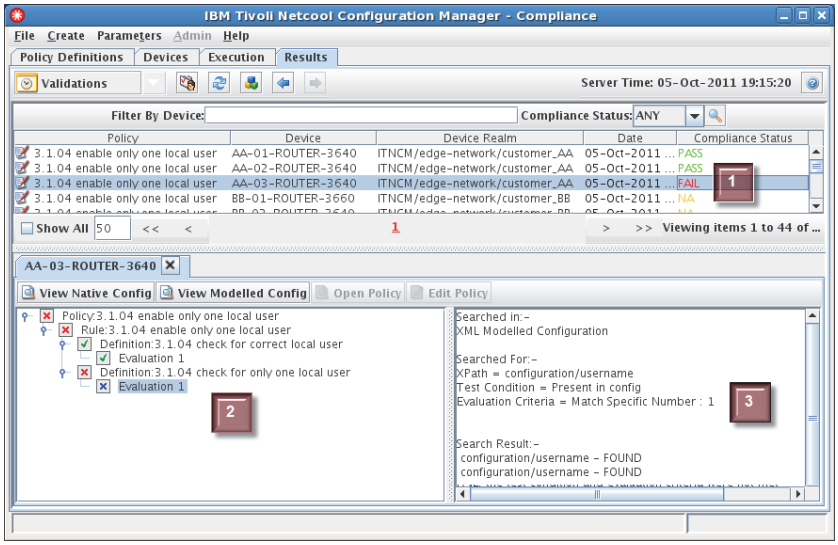
   f. **Automated**: A process that you define for automatic validations. When a state change is indicated on a device, the policies in this process are run against this device to check its compliance.

3. Select a validation in the upper pane to present the details of the policies that you use in the execution. It notes each policy and it details the pass/fail/exempt/not assessed values for the total devices that it analyzes.

4. Select a policy in the lower pane to see the details of each device configuration that it evaluates. You can also export details.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Drill into the compliance results



© Copyright IBM Corporation 2016

Drill into the compliance results

After you select a policy and click **Details**, the GUI presents a list of all the devices that are summarized in the previous table. Here you can filter the results for pass, fail, exempt, and not assessed. The following features are available on the GUI:

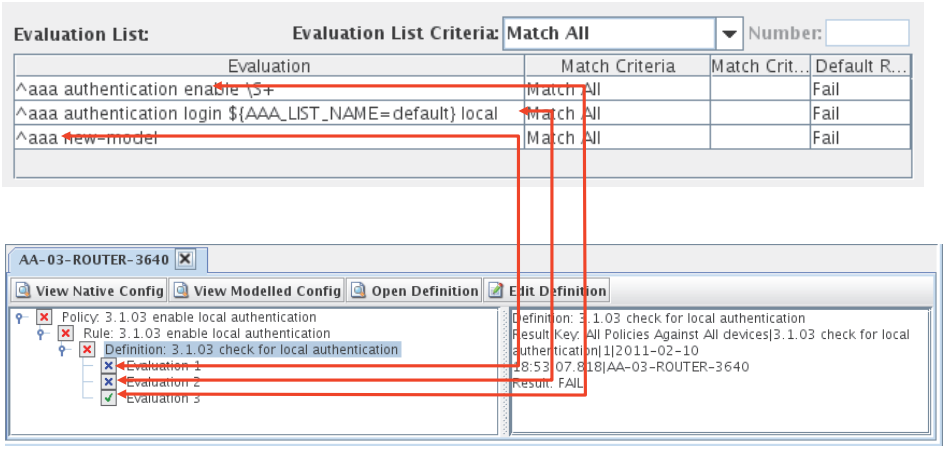
1. When you select a device name and double-click it, the GUI opens the pane on the lower half of the screen.

2. The left side of the screen shows the policy and its associated rules, definitions, and evaluations. A future lesson presents these three items. By selecting an evaluation, you can see the results of the evaluation against this device. In this policy, it attempts two searches on the configuration. The first definition looks for a specific user name in the device configuration. The green check mark indicates that it finds the command. The second definition verifies that only one user name is defined inside the configuration. The blue X indicates that the definition finds more than one user name.

3. The lower right pane shows the results of the evaluation that you select in the left screen. In the example, the evaluation searches on user names to verify that only one user name is configured in the device. The search results show that two user names are defined in the configuration.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Interpreting the details



© Copyright IBM Corporation 2016

Interpreting the details

In the slide on this page, you note that one evaluation for each definition exists. However, a definition can have multiple evaluations. Future sections present this concept in detail. For now, understand that this definition has three evaluations or searches of the configuration. Each evaluation searches for specific information. As shown in the slide, two evaluations have a blue X beside them and one has a green check mark. These marks mean that two of the evaluations did not find their search target and the third did. In this particular situation, the Evaluation List Criteria at the top of the window is set to Match All. All three evaluations must return a green check mark for the policy to be compliant. The rule determines the question of compliance.

Selecting any of the evaluations presents the command that it searches for in the configuration.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the remedial queue

Optional side effect of a noncompliant device

Releases units of work to fix devices noncompliant issue

Uses command sets defined in Netcool Configuration Manager

Can associate parameter values to command sets

© Copyright IBM Corporation 2016

Key points of the remedial queue

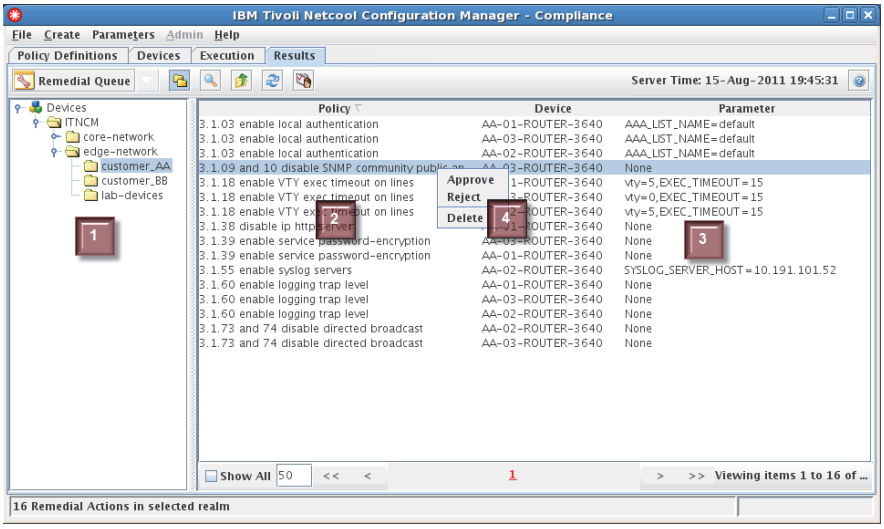
A remediation is an optional object that the remedial queue presents. These remediations are associated with policies when they are defined. The queue is where you release these remedial actions to create units of work that fix the issues that cause the noncompliant situation.

Remedial actions use command sets defined in the configuration manager. The remedial action can associate parameter values to the command set.

Important: It is possible to configure the compliance manager so that remedial actions create *units of work* automatically. This feature is configured for one or more groups that are starting specific processes. So it is possible that certain policies with remediations are associated with a process and these remediations modify devices in an automated way. Review this practice thoroughly before implementing it.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

The Remedial Queue GUI



© Copyright IBM Corporation 2016

The Remedial Queue GUI

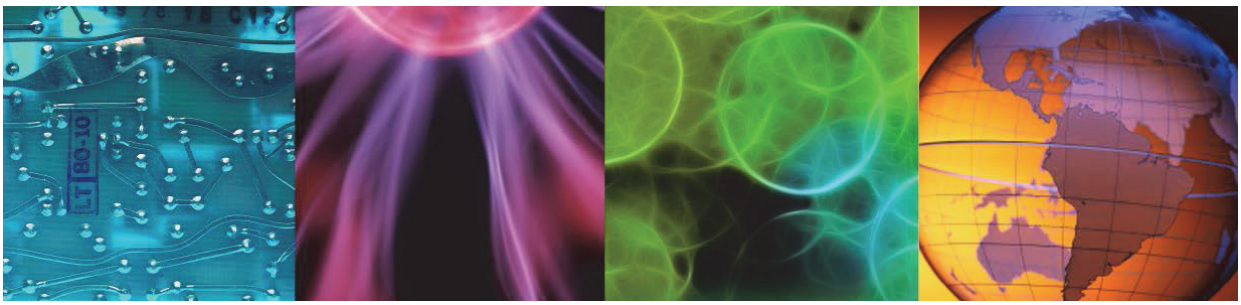
The Remedial Queue GUI has the following features:

1. The left pane shows the device realms. You can browse to the realms that have remedial actions that you need to approve. Selecting a realm presents the remedial actions ready for release.
2. The right pane presents the policy that is found noncompliant and has a remedial action. Each policy or device combination is presented in this pane.
3. Some of the remedial actions have parameters. These values can align with the command set in the *configuration manager*. These parameters can be explicit or implicit.
4. To approve or reject the remedial action, select one or more of these remedial actions and right-click. After you approve the actions, the *compliance manager* tracks the *units of work* that are submitted to the *configuration manager*. Refreshing this view presents these updates.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 6 Integration with configuration management

Lesson 6 Integration with configuration management

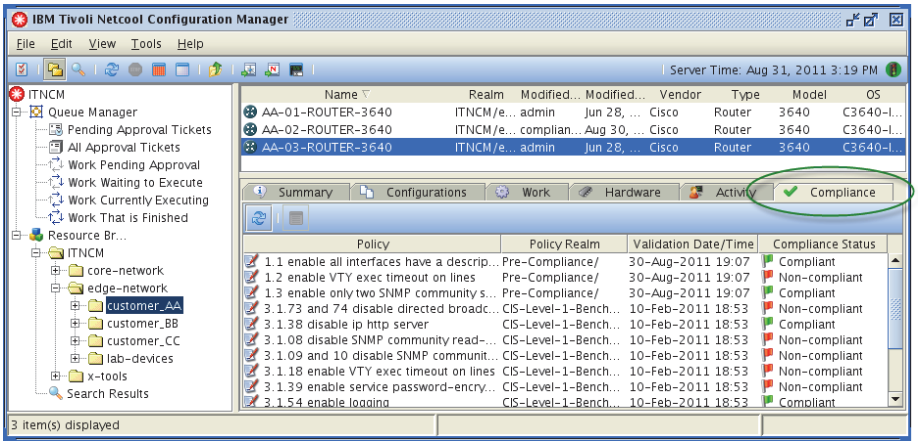


© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to view compliance results in *configuration manager*.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Compliance results are available



© Copyright IBM Corporation 2016

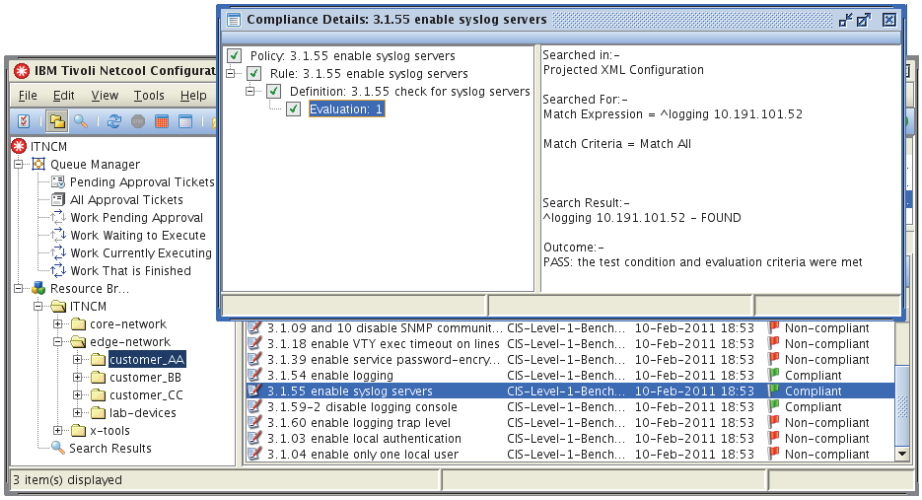
Compliance results are available

In the *configuration manager*, the user can view the results of the *compliance manager*. Select a device and then select the **Compliance** tab. The *configuration manager* shows the policies that you have the rights to view.

Note: The policies that you have the rights to view are based on the policy realms your group has the rights to view. Reference the compliance security features for more information.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Compliance details are also available



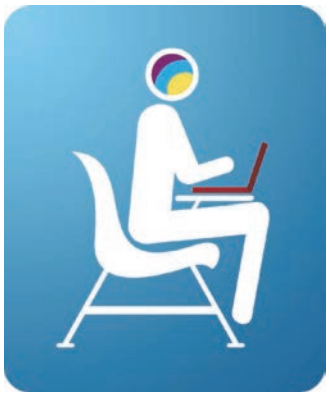
© Copyright IBM Corporation 2016

Compliance details are also available

Users can access details of why a device is not compliant by double-clicking the policy in the *configuration manager*.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises



© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Access and download the application
- Log on to the graphical user interface
- Browse through the main portions of the application

© Copyright IBM Corporation 2016

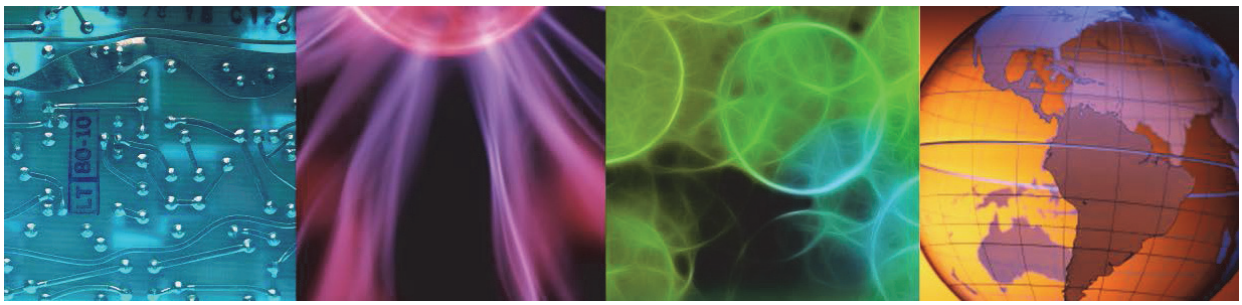
Summary

This unit introduced you to the client graphical user interface (GUI) for the *compliance manager*. It provided an overview of the client technology and the way the software downloads to the client computer. You learned the main aspects of the compliance user interface.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

16 Compliance reports

Compliance reports



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit presents the user with an overview of the compliance reports that are available in Tivoli Common Reporting. It shows how to access these reports and provides examples. You learn how to calculate a score to measure the compliance of your network.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

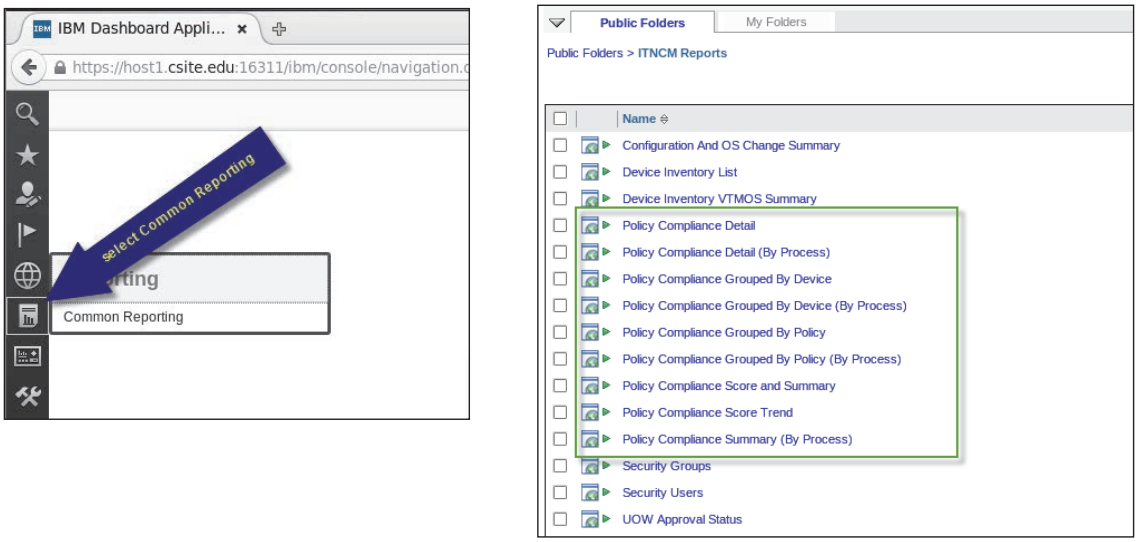
- Access reports
- Describe the reports available in compliance manager
- Calculate compliance scores

© Copyright IBM Corporation 2016

Objectives

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Use Dashboard Application Services Hub to access reports



© Copyright IBM Corporation 2016

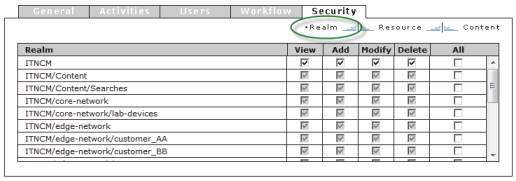
Use Dashboard Application Services Hub to access reports

You can find compliance reports under **Reporting > Common Reporting > ITNCMReports**. All the compliance reports begin with the word **Policy**. To access the Dashboard Application Services Hub interface, use the same credentials that you use for accessing the compliance GUI.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

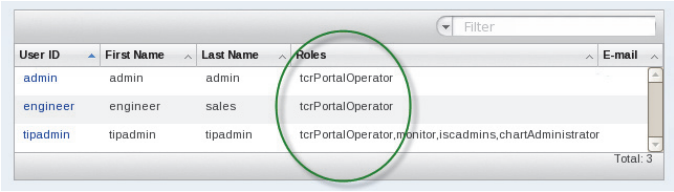
Security settings affect reports

Netcool Configuration Manager security affects what devices can be reported on



Dashboard Application Services Hub allows access to reports

Dashboard Application Services Hub and Netcool Configuration Manager user names and passwords must align



© Copyright IBM Corporation 2016

Security settings affect reports

An important aspect of the compliance reports is that reporting uses the configuration manager security model to ensure that you include only authorized data in reports. These reports include only the devices and policies associated with your configuration manager group or groups. Dashboard Application Services Hub is where users are given access to the actual reports.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Report types

High-level management reports
- Score and summary

Mid level device reports
- Policy
- Device

Low-level details report
- Validation

© Copyright IBM Corporation 2016

Report types

A series of compliance reports are available to users. They vary from high-level summary reports to detailed validation reports. Review the student notes for examples of the major reports that you can create.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Example: Policy Compliance Score & Summary report

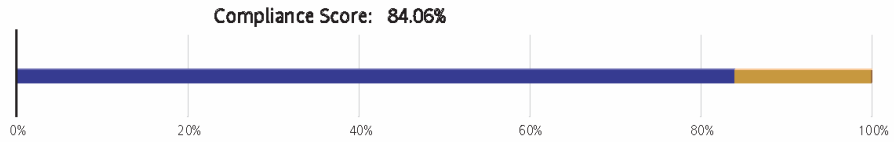
Tivoli. software

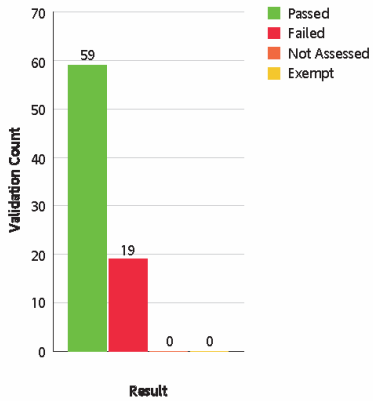
IBM

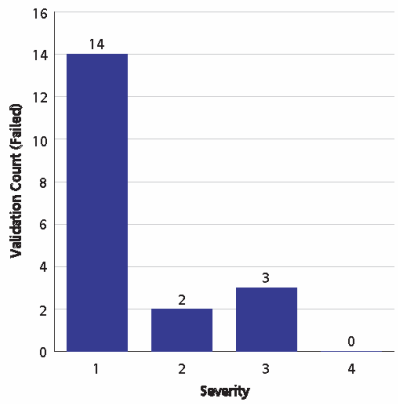
Policy Compliance Score & Summary

Compliance Score

This score represents the weighted compliance score for the devices in the selected device realms, as evaluated by the selected policies. The weighted compliance score is based on the most current validation results for the selected devices and policies and the relative importance of each of the policies as defined by the weight of a policy.







The following table summarizes the compliance results for each selected policy and all selected devices.

Policy Name	Severity	Passed	Failed	Not Assessed	Exempt	Policy Weight
1.1 enable all interfaces have a description	3	0	3	0	0	25
1.2 enable VTY exec timeout on lines	2	3	0	0	0	25
1.3 enable only two SNMP community strings	2	1	2	0	0	25
3.1.03 enable local authentication	1	0	6	0	0	25
3.1.04 enable only one local user	1	4	2	0	0	75
3.1.08 disable SNMP community read-write	1	0	6	0	0	20
3.1.09 and 10 disable SNMP community public and private	1	6	0	0	0	100
3.1.18 enable VTY exec timeout on lines	2	6	0	0	0	25
3.1.38 disable ip http server	1	3	0	0	0	25
3.1.39 enable service password-encryption	2	6	0	0	0	100
3.1.54 enable logging	3	6	0	0	0	50
3.1.55 enable syslog servers	3	6	0	0	0	50
3.1.59-2 disable logging console	4	6	0	0	0	25
3.1.60 enable logging trap level	4	6	0	0	0	25
3.1.73 and 74 disable directed broadcast	2	6	0	0	0	25
Policy Count	**15**					

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Example: Policy Compliance Grouped by Device report

Tivoli. software

IBM

Policy Compliance Grouped By Device

Device Name AA-01-ROUTER-3640
Device Realm ITNCM/edge-network/customer_AA
Device VTMOS Cisco/Router/3640/C3640-I-M-12.3(5b)

Policy Name	Policy Realm	Severity	Result
1.1 enable all interfaces have a description	Pre-Compliance/	3	FAIL
1.3 enable only two SNMP community strings	Pre-Compliance/	2	FAIL
3.1.03 enable local authentication	CIS-Level-1-Benchmark/Management Plane Level 1/Local AAA Rules/	1	FAIL
3.1.03 enable local authentication	CIS-Level-1-Benchmark/Management Plane Level 1/Local AAA Rules/	1	FAIL
3.1.08 disable SNMP community read-write	CIS-Level-1-Benchmark/Management Plane Level 1/SNMP Rules/	1	FAIL
3.1.08 disable SNMP community read-write	CIS-Level-1-Benchmark/Management Plane Level 1/SNMP Rules/	1	FAIL
Policy Count	**6**		

Device Name AA-02-ROUTER-3640
Device Realm ITNCM/edge-network/customer_AA
Device VTMOS Cisco/Router/3640/C3640-I-M-12.4(25c)

Policy Name	Policy Realm	Severity	Result
1.1 enable all interfaces have a description	Pre-Compliance/	3	FAIL
3.1.03 enable local authentication	CIS-Level-1-Benchmark/Management Plane Level 1/Local AAA Rules/	1	FAIL
3.1.03 enable local authentication	CIS-Level-1-Benchmark/Management Plane Level 1/Local AAA Rules/	1	FAIL
3.1.08 disable SNMP community read-write	CIS-Level-1-Benchmark/Management Plane Level 1/SNMP Rules/	1	FAIL
3.1.08 disable SNMP community read-write	CIS-Level-1-Benchmark/Management Plane Level 1/SNMP Rules/	1	FAIL
Policy Count	**5**		

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Example: Policy Compliance Grouped by Policy report

Tivoli. software

IBM.

Policy Compliance Grouped By Policy

Realms	ITNCM/edge-network/customer_AA	**Policies**	1.1 enable all interfaces have a description, 1.2 enable VTY exec timeout on lines, 1.3 enable only two SNMP community strings, 3.1.03 enable local authentication, 3.1.04 enable only one local user, 3.1.07 disable SNMP if not used, 3.1.08 disable SNMP community read-write, 3.1.09 and 10 disable SNMP community public and private, 3.1.11 disable SNMP community without ACLs, 3.1.17 enable VTY transport telnet, 3.1.18 enable VTY exec timeout on lines, 3.1.20 disable AUX, 3.1.21 enable VTY login authentication default, 3.1.22 enable VTY login authentication named-list, 3.1.24 enable VTY password, 3.1.25 enable a local enable password, 3.1.28 enable VTY access class, 3.1.34 and 36 and 37 disable finger service, 3.1.35 disable identd service, 3.1.38 disable ip http server, 3.1.39 enable service password-encryption, 3.1.42 and 44 and 46 enable ntp servers, 3.1.50 clock timezone - GMT, 3.1.51 forbid clock summer-time - GMT, 3.1.52 enable service timestamps logging, 3.1.53 enable service timestamp debug, 3.1.54 enable logging, 3.1.55 enable syslog servers, 3.1.57 enable logging buffer, 3.1.59-1 enable logging console critical, 3.1.59-2 disable logging console, 3.1.60 enable logging trap level, 3.1.62 and 64 disable tcp-small-servers, 3.1.63 and 65 disable udp-small-servers, 3.1.66 disable ip bootp server, 3.1.67 disable cdp run, 3.1.68 disable service config, 3.1.69 enable service tcp-keepalives-in, 3.1.70 disable tftp-server, 3.1.73 and 74 disable directed broadcast, 3.1.75 disable ip source routing, enable syslog servers customer-AA, enable syslog servers customer-CC
Vendor*		**Type**	*
Model	*	**OS**	*
Name	*	**Policy Severity**	1, 2, 3, 4, 5
Policy Result	Fail		

Policy Information:

Name	3.1.04 enable only one local user
Description	Create only one local user with password to be used in case external authentication servers are not available
Revision	1
Severity	1

Device Information:

Device Name	Device Realm	Vendor	Type	Model	OS	Result
AA-03-ROUTER-3640	ITNCM/edge-network/customer_AA	Cisco	Router	3640 (R4700)	C3640-IK9S-M-12.4(25c)	FAIL
AA-03-ROUTER-3640	ITNCM/edge-network/customer_AA	Cisco	Router	3640 (R4700)	C3640-IK9S-M-12.4(25c)	FAIL
Device Count	**2**					

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Example: Policy Compliance Detail report

Tivoli. software IBM.

Policy Compliance Detail

Device Name AA-02-ROUTER-3640
Device VTMOS
Device Realm ITNCM/edge-network/customer_AA
Policy 3.1.08 disable SNMP community read-write **Policy Result** FAIL

Searched For	Search Result	Outcome
Match Expression = ^snmp-server community .* RW Match Criteria = Match All	1 match snmp-server community wr1t35tr1ng RW	PASS: the test condition and evaluation criteria were met
	Result Count	1

Device Name AA-02-ROUTER-3640
Device VTMOS
Device Realm ITNCM/edge-network/customer_AA
Policy 1.1 enable all interfaces have a description **Policy Result** FAIL

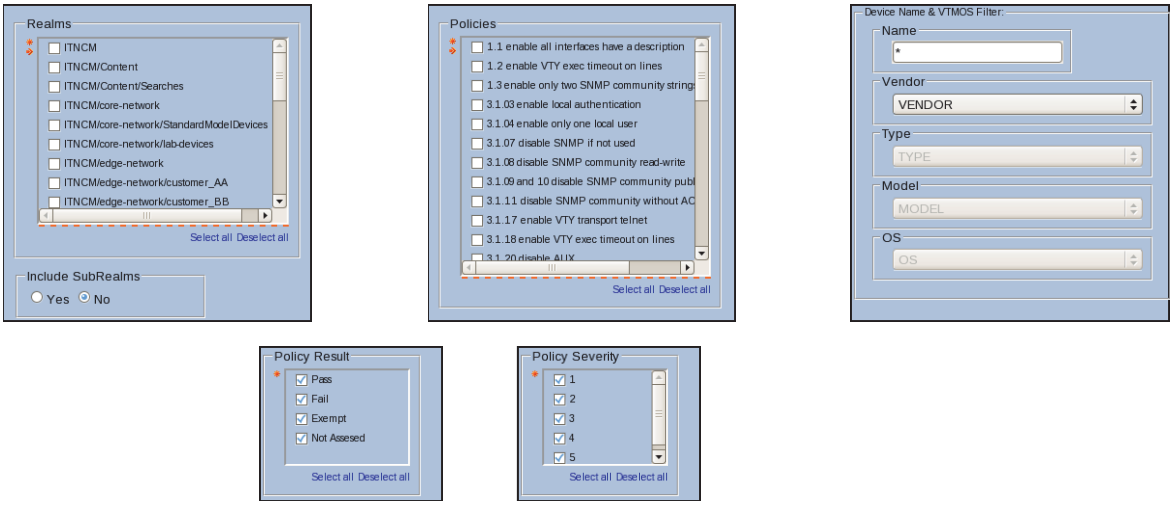
Searched For	Search Result	Outcome
Context XPath = configuration/interface/* Defined XPath = description Context Nodes = description Test Condition = Present in config Evaluation Criteria = Match All	description=null description - NOT FOUND description=null description - NOT FOUND description=null description - NOT FOUND description=null description - NOT FOUND	FAIL: the test condition and evaluation criteria were not met
	Result Count	1

Device Name AA-02-ROUTER-3640
Device VTMOS
Device Realm ITNCM/edge-network/customer_AA
Policy 3.1.03 enable local authentication **Policy Result** FAIL

Searched For	Search Result	Outcome
Match Expression = ^aaa authentication enable \S+ Match Criteria = Match All	NOT FOUND	FAIL: the test condition and evaluation criteria were not met
Match Expression = ^aaa authentication login default local Match Criteria = Match All	NOT FOUND	FAIL: the test condition and evaluation criteria were not met
Match Expression = ^aaa new-model Match Criteria = Match All	1 match aaa new-model	PASS: the test condition and evaluation criteria were met
	Result Count	3

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Report filters



© Copyright IBM Corporation 2016

Report filters

When you create a report, you are presented with a series of filters that you use to create a targeted set of report data. The following filters are available in various reports.

- **Realms**: This filter is the list of device realms the Dashboard Application Services Hub user is authorized to see in Netcool Configuration Manager. Select one or more realms for the report. It is important to select **Include SubRealms** to see lower-level devices and realms.
- **Policies**: This filter is the list of policies to include in the report. Notice that some policies might not apply to the devices in the selected realms.
- **Device Name & VTMOS**: If a logical naming convention for device names or for select specific types of devices exists, you can use this filter.
- **Policy Results**: Select which policy results to include in the reports. Most reports are interested in the *fail* result.
- **Policy Severity**: Select the level of severity to include in the report. Selecting severity 1 shows only the policies that have a severity 1 setting.

Some reports might use only a subset of these filters.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

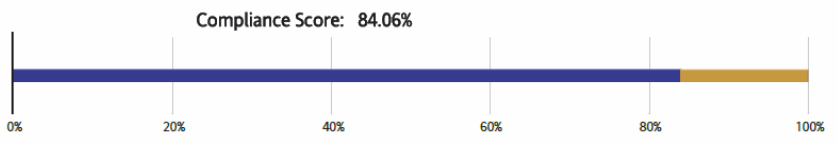
Processes generate compliance score

Shown in the Policy Compliance Score & Summary report

Generated by selecting device realm and policies

Weighted average of policy results and policy weight

Example: Customer-AA devices (3) and applied policies (15)



© Copyright IBM Corporation 2016

Processes generate compliance score

When a process starts and its policies are evaluated against devices, it creates a compliance score.

It generates the compliance score from a combination of selecting a device realm and a policy realm.

The policy score is useful for tracking the compliance of the network over time as it provides a quantitative measure that you can place on a trend chart. A report is available that shows this value as it changes over time.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Compliance score calculation

Policy	Weight	Passed	Failed	Exempt	Not Assessed	Score	Weighted Average	Weighted Score
Policy A	25	0	4	0	0	0%	0.125	0.00%
Policy B	25	0	4	0	0	0%	0.125	0.00%
Policy C	25	4	0	0	0	100%	0.125	12.50%
Policy D	25	0	4	0	0	0%	0.125	0.00%
Policy E	25	0	4	0	0	0%	0.125	0.00%
Policy F	25	0	4	0	0	0%	0.125	0.00%
Policy G	25	2	2	0	0	50%	0.125	6.25%
Policy H	25	0	4	0	0	0%	0.125	0.00%
							Compliance Score	18.75%

$$score_x = \frac{(passed_x + exempt_x)}{(passed_x + failed_x + exempt_x + not\ assessed_x)} * 100$$

$$weighted\ average_x = \frac{weight_x}{\sum_{i=1}^{n} weight_i}$$

$$weighted\ score_x = score_x * weighted\ average_x$$

$$Compliance\ Score = \sum_{i=1}^{n} weighted\ score_i$$

© Copyright IBM Corporation 2016

Compliance score calculation

The compliance score is a weighted average of success and failure in terms of policies and the number of devices analyzed. To calculate it, you must perform the following steps:

1. First, select a set of policies and devices, for example, Policy A through Policy H as noted in the slide. A total of four devices make up the device realm.
2. For each policy, calculate a score that is based on the number of devices that pass, fail, are exempted, and are not assessed. The score is a percentage of success. Take the sum of the passing and the exempt devices, divide it by the total number of devices, and make the result into a percentage.
3. Create a weighted average of that score. In this step, you take the weights that are defined for each policy and create a summation of them. You take the weight of each individual policy and divide it by the total weight. In the example, each policy has the same weight so the weighted average is the same (25 / (8*25) = 0.125).
4. Take the weighted average value and multiply it by the score percentage to come up with a weighted score, which is a percentage.
5. The compliance score is the summation of all the weighted scores, which is expressed as a percentage.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

You can automate and distribute reports

Processes generate the data for reports, recurring schedule allows for regular refreshing of policy results

Automate report creation, recurring schedule allows for regular reports

Distribute either through email or by saving on the server

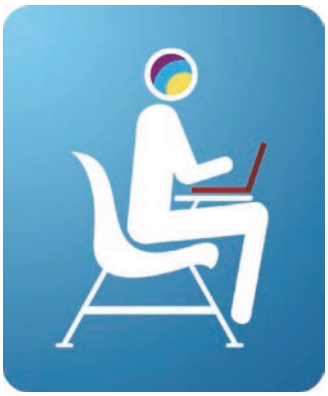
© Copyright IBM Corporation 2016

You can automate and distribute reports

You have the option in Tivoli Common Reporting to automate the creation and distribution of reports. You can define a schedule for report creation. You can save the report output as a file on the server for subsequent viewing. Or, you can send the output to one or more users through email.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises



© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Access reports
- Describe the reports available in compliance
- Calculate compliance scores

© Copyright IBM Corporation 2016

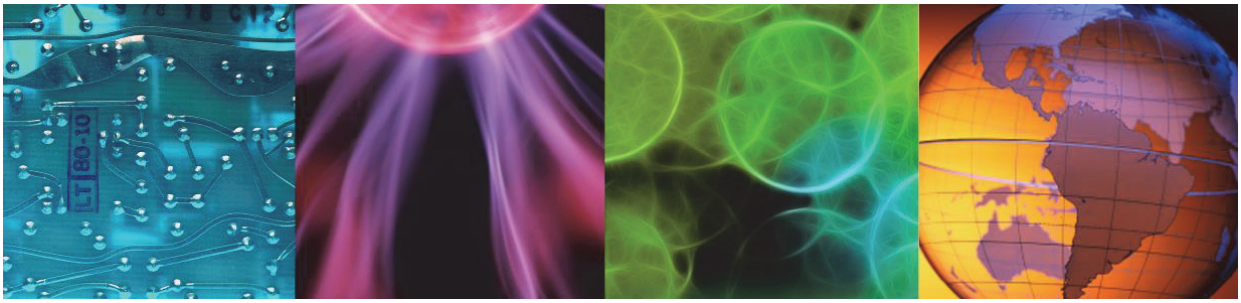
Summary

This unit presented an overview of the compliance reports that are available in Tivoli Common Reporting. It showed how to access these reports and provided examples. You learned how to calculate a score to measure the compliance of your network.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

17 Build and run a policy

Build and run a policy



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit provides an overview of the objects that make up a compliance policy and the role of objects in the compliance process. This unit shows how to create and test a simple policy. The unit also teaches the student how to exempt devices from a policy and how to include a policy into a production process for scheduled execution.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Define the components that make up a policy
- Create and test simple policies
- Exempt devices from policies
- Build and run a process

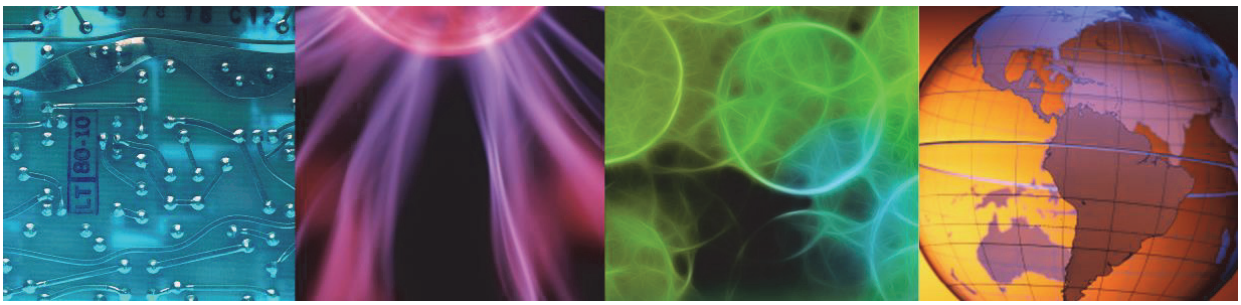
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Policy overview

Lesson 1 Policy overview



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the components of a policy
- Describe how to use documentation to build policy
- Describe how a policy realm is used in security.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies

Policy A

Defines a logical condition

Defines severity & weight

Sends SNMP traps, optional

VTMOS filter, optional

© Copyright IBM Corporation 2016

Policies

A policy is a logical condition that is independent of a vendor device. It reflects some condition on a device that is considered not compliant, under normal conditions.

Consider the following logical conditions:

- Encrypt passwords in configurations
- Where applicable, make device connections through Secure Shells (SSH)
- Configure only approved SNMP community strings

Policies also have severity levels that are associated with them. Severity levels are defined from 1, the most severe, to 5, the least severe. Most organizations define a policy by its level of severity, and most reports show severity as a component. The policy also defines weight as a value 0 - 100. Weight aligns with severity where a severity of 1 reflects a weight of 100. Use weights to calculate a compliance score for a group of policies and devices.

You define whether an SNMP trap is generated when a device violates the policy.

Policies can have VTMOS filters that define that they apply only to specific vendors, models, and operating systems. It is better to apply VTMOS filters with a lower-level rule object.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies run optional actions

Sends email on a per-policy basis for noncompliant devices

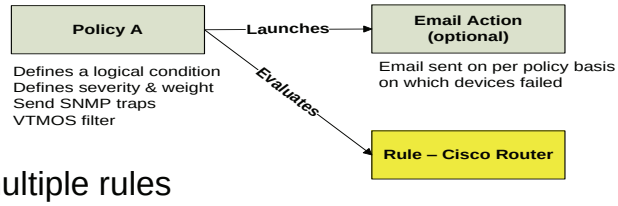
© Copyright IBM Corporation 2016

Policies run optional actions

You can configure a policy to run an email action when devices violate the policy. The email contains the names of the devices that violate the policy.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies evaluate rules to determine compliance



Policies can have multiple rules

Defines VTMOS specific test

Defines whether device is compliant

Can use AND and OR logic

© Copyright IBM Corporation 2016

Policies evaluate rules to determine compliance

Policies use rules to determine whether a device is compliant. A policy can have multiple rules. Each rule is a VTMOS-specific test that checks for the logical condition that the policy describes. No device overlap with rules must exist. A device must not resolve to two rules inside a single policy. In theory, a rule for each type of device in your network can exist.

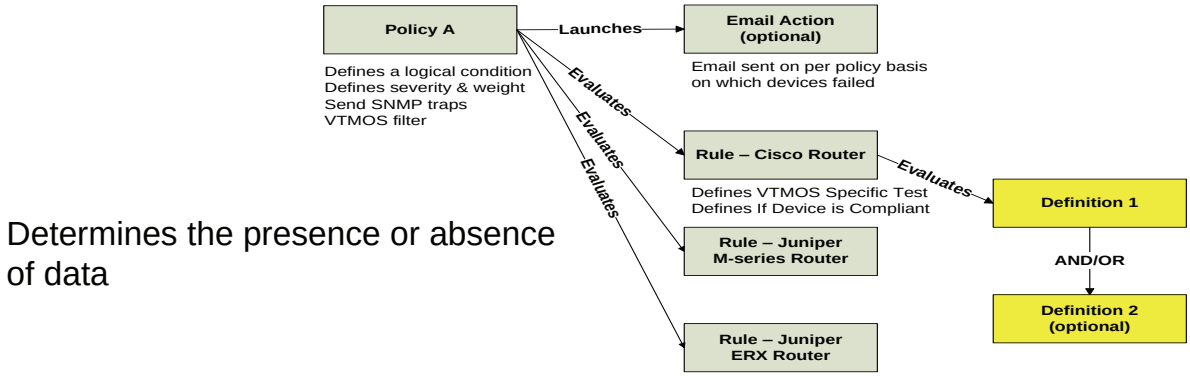
Rules must always be vendor, model, and sometimes OS-specific. The device type is optional, depending on the vendor. When creating the rule, you have a series of VTMOS filters to ensure that the rule applies to the correct device. These VTMOS filters might use advanced options.

Rules define whether a device is compliant or not. It takes a true or false condition and turns it into a compliant or noncompliant condition.

You use rules to build logical AND and OR gates that are used to check several conditions to verify that a device is compliant.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Rules evaluate one or more definitions



Determines the presence or absence of data

© Copyright IBM Corporation 2016

Rules evaluate one or more definitions

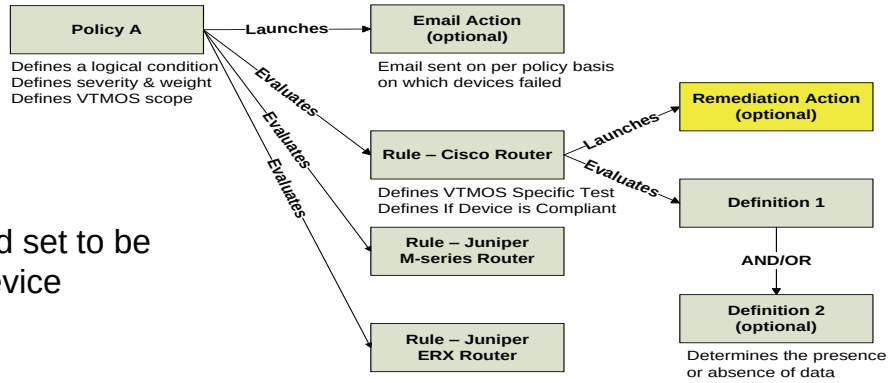
Rules use definitions to determine whether a device is compliant or not. Definitions determine the presence or absence of data in a data structure. They are the most technical aspect of policy creation. Definitions are where data structures are interrogated with tools like regular expressions and XPaths.

You can organize definitions into AND and OR logic gates that are part of the rule object.

A definition always returns a *true*, *false*, or *not assessed* answer. Definitions do not determine whether a device is compliant or not compliant.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Rules can run remedial actions



Prepares a command set to be submitted against device

© Copyright IBM Corporation 2016

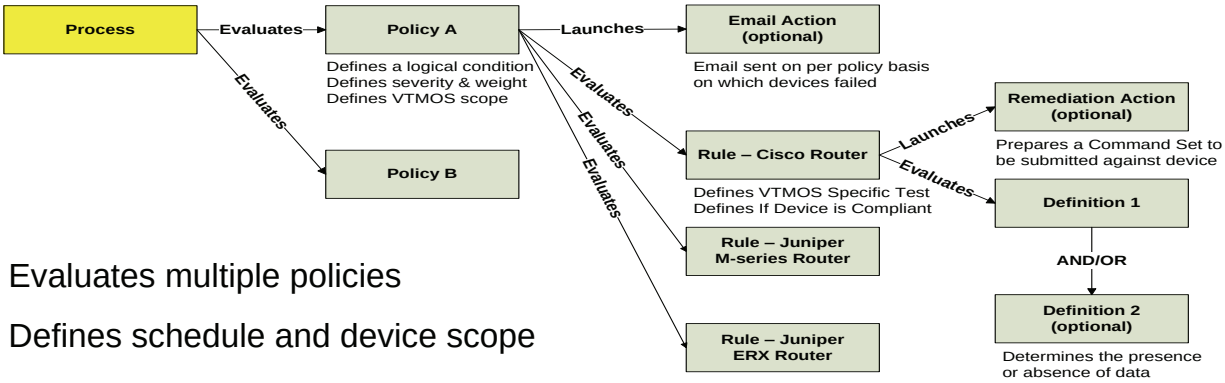
Rules can run remedial actions

Rules use definitions to determine whether a device is compliant or not. After a rule determines that a device is not compliant, it can use an optional object to begin the process to fix the problem. A remedial action is where compliance integrates into the *configuration manager*. Remedial actions link to command sets. The command sets modify the device configuration to make it compliant.

Remedial actions can be approved either automatically or manually. After a remedial action is approved, it submits a *unit of work* to the *configuration manager* that combines the linked command set and the device. *Compliance manager* tracks the progress of the UOW and notes whether it is a success or a failure.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Processes evaluate policies against devices



Evaluates multiple policies

Defines schedule and device scope

© Copyright IBM Corporation 2016

Processes evaluate policies against devices

After you configure a policy at the definition level, it can incorporate with other policies in an executable structure that is a process. Processes take a series of policies and devices and run those policies according to a schedule you define. Each policy within a process tests for different compliance conditions. You can also use processes to generate data for reports.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Building policies

Policies come from multiple locations

- Vendors
- Industry standards
- Corporate directives

Center for Internet Security

- http://cisecurity.org
- Security Configuration Benchmark For Cisco IOS, Version 2.4

© Copyright IBM Corporation 2016

Building policies

You use existing network procedures, practices, and directives to design a policy. A common practice for policy design is to reference some directive that is outside the system. These directives can come directly from vendors as they discover potential flaws or security holes in their operating systems. They can come from industry standards that government agencies or user groups develop. Sometimes directives come from the organization itself as it tries to simplify and standardize its networks.

The Center for Internet Security (CIS) created example industry standards. This nonprofit organization defined a set of security standards for various networked operating systems. One of the standards from this organization is the Security Configuration Benchmark for Cisco IOS. This document describes and details how to secure Cisco IOS devices. It provides the commands that you search for in a configuration to determine whether it meets a particular policy.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Center for Internet Security example

3.1.55 IOS - set syslog server

Description	Designate one or more syslog logging servers
Applicability	11+ IOSGlobal configuration mode
Rule Type	Control Plane Level 1⇒Logging Rules Level 1
Documentation	See section 3.2.34.
Action	`router(config)# logging $(SYSLOG_HOST)`

3.2.34 IOS - set syslog server

Security Impact	Cisco routers can send their log messages to a Unix-style syslog service. A syslog service simply accepts messages, and stores them in files or prints them according to a simple configuration file. This form of logging is the best available for Cisco routers, because it can provide protected long-term storage for logs.
Importance	5
For More Info	See RSCG page 130 for more information.
Rule Actions	See section 3.1.55.
Rule Match	`logging $(SYSLOG_HOST)`

This example is taken from the Center for Internet Security Gold Standard Benchmark for Cisco IOS Version 2.1, Sept 2003

© Copyright IBM Corporation 2016

Center for Internet Security example

The slide shows an example of a policy from the CIS document that encodes the policies that you find on the training server.

Notice that this policy ensures that devices are pointing to a syslog server to capture any significant messages from the device. This documentation is useful in that it describes why having a syslog server is so important. Cutting and pasting this documentation into the policy structures is useful for other users who review compliance reports but do not have access to the original document.

Notice that this document further references the Router Security Configuration Guide (RSCG) which the United States National Security Agency developed.

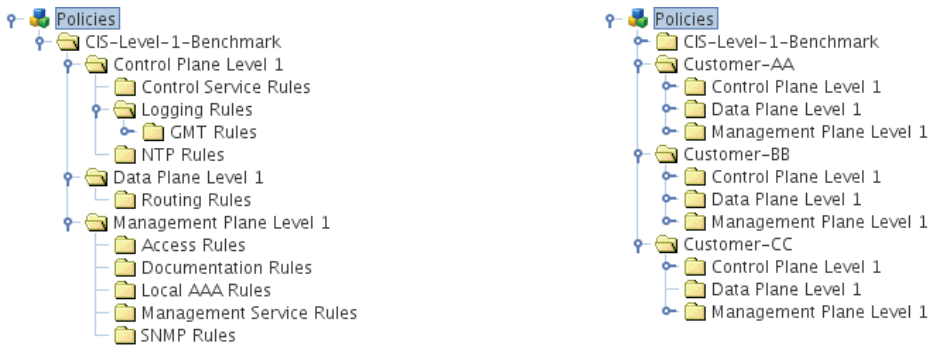
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policy realms

Policies are in policy realms

Policy realms can be organized in multiple ways

- By policy type
- By customer



© Copyright IBM Corporation 2016

Policy realms

Before building a policy, this section reviews the concept of policy realms. Like device realms that you define in configuration management, you define policy realms to organize policy. Policy realms are distinct from device realms, but serve the same purpose.

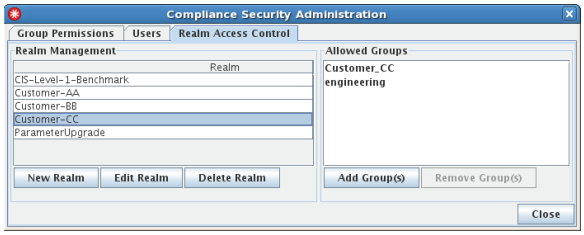
In many organizations that use the *compliance manager* to manage their private network, the policy realm is a means to organize policy by function. Assume the CIS Gold Standard Benchmark for Cisco IOS as a guide. The policy realm on the left in the slide is organized by the various roles and services that you need to configure on a router. The CIS guide divides the router configuration into three separate planes: a control plane, a data plane, and a management plane. These categories are subdivided into more specific functions. Here, policies exist in only one location and are not unique to the organization.

Service providers can use policy realms to break up policies by customer. In the scenario that the right side of the slide describes, you see a series of customers. Each customer has a series of policies that might or might not be unique to their organization. The intent is to isolate users to specific policy realms. You can use this technique to implement policies differently from customer to customer. These differences are most apparent when it comes to defining parameter values.

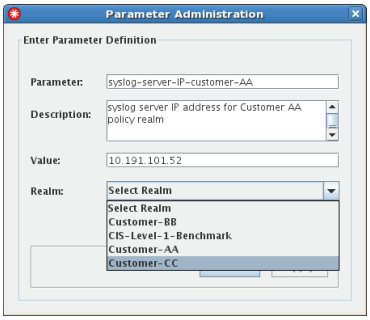
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Base policy realms support security

Security admin gives group rights to view policy realms



Can associate global parameters to a base policy realm



© Copyright IBM Corporation 2016

Base policy realms support security

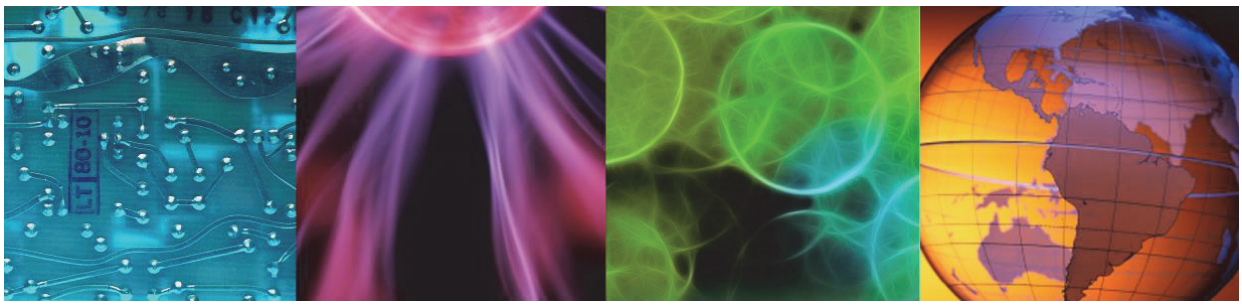
When policy realms need security like device realms need it, it is done only for base realms. Here the *compliance manager* has a security option where *configuration manager* groups associate with specific base realms. If a group is not associated with a policy realm, a user in that group is able to access the *compliance manager* but cannot see any policies. This security allows customers to see only the policies that are associated with their policy realm. This concept needs to align with the device realms to ensure that it fully enforces security.

This policy realm defines global parameters. The global parameter is defined for a specific policy realm and is only used in policies that reference it in that realm.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Building a definition

Lesson 2 Building a definition



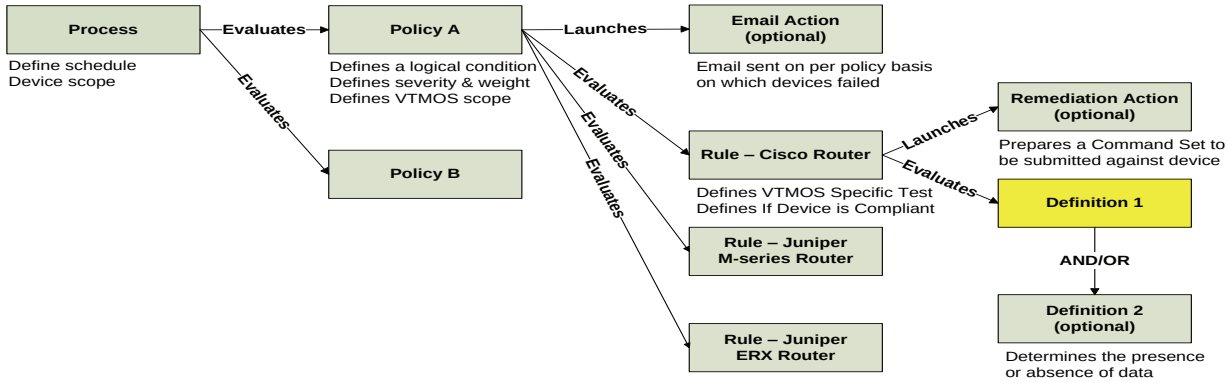
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe a definition
- Choosing which data element to search

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a definition



© Copyright IBM Corporation 2016

Creating a definition

Remember where the definition exists in the policy hierarchy.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Definition overview

Most granular part of process

Returns either *true*, *false*, or *not assessed*

Searches for the existence of data

- Configurations
- Cached show commands
- Live device data

Uses regular expressions, AND and OR, and XPath (SmartModels®)

- Can specify *match any*, *all*, *some*, or *none*
- Can use parameters, extractions

Definitions can use JavaScript to run complex searches on data structures

© Copyright IBM Corporation 2016

Definition overview

The definition is the most granular part of the policy creation process. It is the most complex of the policy objects that you must create. Definitions are simple or complex depending on what you are attempting.

Definitions return *true*, *false*, or *not assessed* to the higher-level rule that calls them.

In most cases, definitions search for the existence or nonexistence of data inside a configuration or a blob of data from various **show** commands that are retrieved from a device. If you require, you can retrieve data directly from a device.

Definitions use string matching, regular expressions, and for SmartModel devices, XPath to search configurations or device data to look for specific strings. Definitions try a series of evaluations. In this case, the definition uses matching criteria for these multiple evaluations and uses the concepts of *match all*, *match any*, *match none*, or *match some*.

Parameters and extractions allow reuse of definitions for different implementations.

In more complex cases, you can use JavaScript to create a definition. The script still returns a true or false, but you can use it for more complex analysis of the data structure you select.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating the definition

Includes reference to documentation

Example from CIS Cisco Benchmark

3.1.55 IOS - set syslog server

Description	Designate one or more syslog logging servers
Applicability	11+ IOSGlobal configuration mode
Rule Type	Control Plane Level 1⇒Logging Rules Level 1
Documentation	See section 3.2.34.
Action	`router(config)# logging $(SYSLOG_HOST)`

3.2.34 IOS - set syslog server

Security Impact	Cisco routers can send their log messages to a Unix-style syslog service. A syslog service simply accepts messages, and stores them in files or prints them according to a simple configuration file. This form of logging is the best available for Cisco routers, because it can provide protected long-term storage for logs.
Importance	5
For More Info	See RSCG page 130 for more information.
Rule Actions	See section 3.1.55.
Rule Match	`logging $(SYSLOG_HOST)`

Determines whether to use CLI, modeling, or native commands to detect whether device is compliant

© Copyright IBM Corporation 2016

Creating the definition

The CIS benchmark document is a great resource for building definitions. It calls out the specific lines of Cisco CLI that you must find in the configuration. Notice that the 3.2.34 supporting documentation defines the rule match.

The definition determines whether it investigates the raw configuration, the modeled configuration, or **show** commands. If you use AND and OR logic at the rule level, It is a combination of all three that ensures compliance.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Differences between CLI and modeled

CLI searches
- Uses a familiar syntax
- Work well with base commands in a configuration
- Regular expressions can be used

Modeled commands allow for context
- Searches subcommands and returns context pointer
- Schema editor helps to build command, generates an XPATH
- Can use regular expressions

CLI

```
tacacs-server host 10.179.20.21
logging 10.55.160.30
no ip http server
ntp server 10.73.130.24 prefer
```

Modeled

```
interface FastEthernet 0/0
description Interface to Customer A
no ip redirects
no ip unreachables
no ip proxy-arp
interface FastEthernet 0/1
description Interface to Customer B
no ip redirects
no ip unreachables
no ip proxy-arp
```

© Copyright IBM Corporation 2016

Differences between CLI and modeled

When you determine that a definition needs to search the device configuration, you must decide whether to search the raw CLI configuration, or the modeled configuration. Either one is acceptable for searching, but reasons exist to choose one over the other.

Searching through the CLI configuration includes the following features:

- You work with a language that is familiar to most network engineers and it is easy to create the definition.
- Works well with base commands. A base command is a command that does not have any subcommands. Often, a base command has one instance of itself as you use it to turn on or off a feature of the device.
- Complex searches can use regular expressions.

Searching through the modeled configuration includes the following advantages:

- Modeled searches allow the easy search of subcommands. The underlying XML that models the configuration represents the concept of context. A context selects one or more nodes of data from the configuration and then further processes each individual node. For example, a context is the fast ethernet interfaces in a configuration. Each interface is retrieved and then each context is further analyzed to search for a specific command. It can also return a pointer or

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

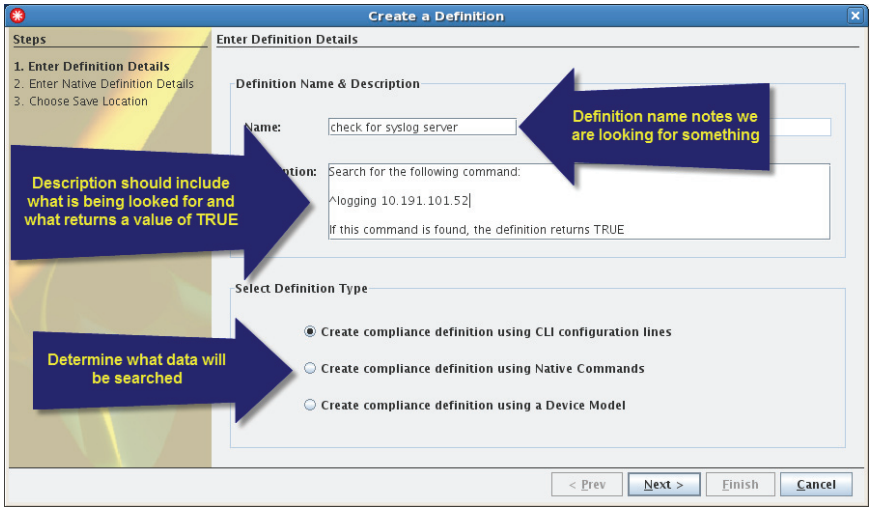
value of the context that does not meet the matching criteria. For example, it can return the interface number.

- Modeled searches begin with an XPath, which is a standard XML search technology. A tool is available to help you build these XPaths based on a command schema of a device.
- You can extend The XPath with regular expressions to make powerful search criteria without deep understanding of either technology.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a definition



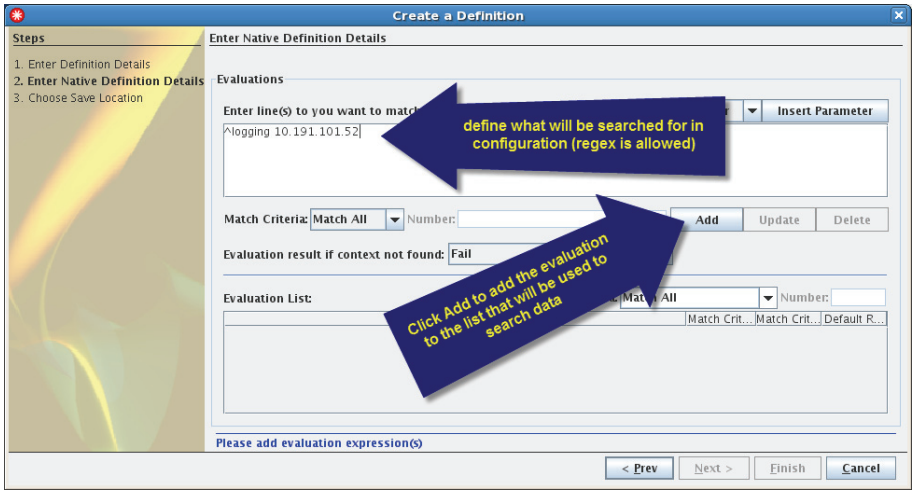
© Copyright IBM Corporation 2016

Creating a definition

Creating a definition object requires that the user has manage compliance policy rights for their group. Use the **Policy Definitions** tab to create the definition. It requires a name. Use a verb like *check* or *search* to define the name. In the training server, it also includes the policy number as a reference. It does not require a description, but it is useful for users who look at the details of a policy analysis.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Enter the search terms



© Copyright IBM Corporation 2016

Enter the search terms

Click **Next** to open the window where you can define the search. This window varies depending on what you are going to search. A future lesson explains these differences.

This definition is searching the raw configuration. You define the search criteria with simple strings or more complex regular expressions. The example on the slide uses a little of each. It uses the ^ character to start the search from the beginning of the line. From there, it is a simple string match for **logging 10.191.101.52**. After you define the string that you want to match, click **Add** and add it to your evaluation list.

Clicking **Next** saves the definition in one of your policy realms.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Notes about definitions

Searches for commands: returns *true*, *false*, or *not assessed*

Definitions do not determine good or bad

© Copyright IBM Corporation 2016

Notes about definitions

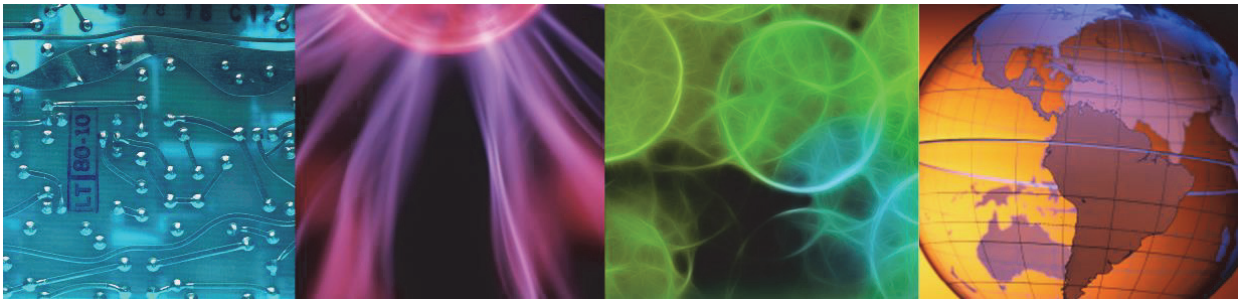
Remember, a definition searches for one or more commands and returns a status of true, false, or not assessed.

Definitions do not determine anything good or bad about a configuration, just the presence or absence of data. The definition can mislead because of words like *pass* and *fail*. These terms do not determine goodness and badness, like someone who passes or fails a class. For clarity, convert *pass* and *fail* into words like *true* and *false* to better align what is done in the higher-level rule.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Building a rule

Lesson 3 Building a rule



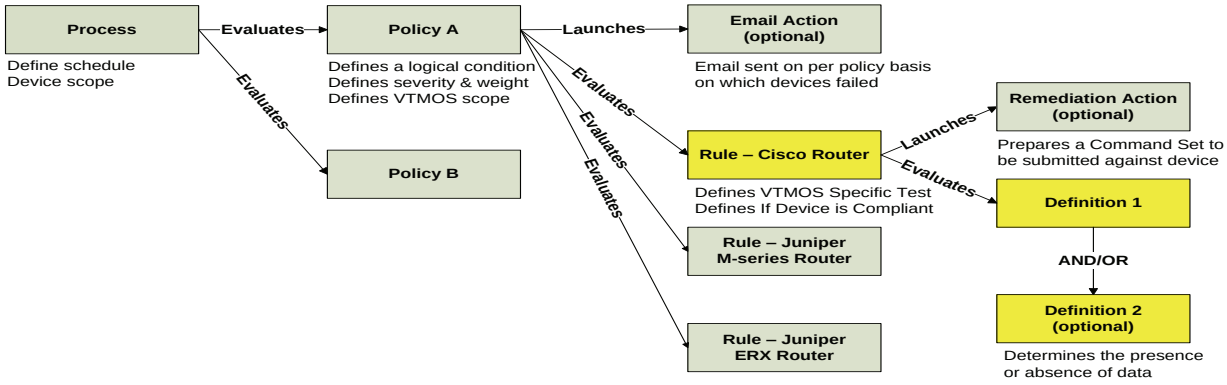
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe a rule
- Configure AND-OR logic in a rule
- Describe advanced VTMOS filtering on a rule

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a rule



© Copyright IBM Corporation 2016

Creating a rule

Remember that a rule is a VTMOS-specific tool that uses definitions to find data and then determines whether the result of that search means that the device is compliant or not.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

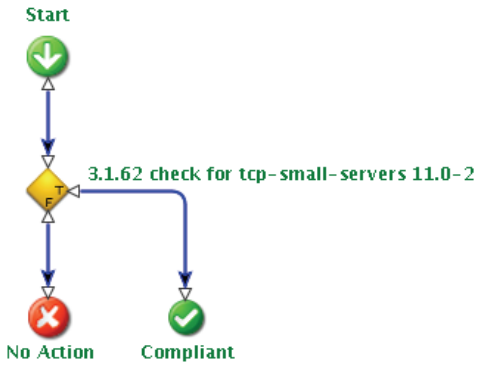
Rules determine noncompliant devices

Rules evaluate vendor-specific definitions

Definitions return a *true*, *false*, or *not assessed*

not assessed devices are neither *compliant* or *noncompliant*

True or *false* is translated to *compliant* or *noncompliant*



© Copyright IBM Corporation 2016

Rules determine noncompliant devices

Rules evaluate vendor-specific definitions. The definition is the yellow diamond icon in the slide. The definition searches for one or more strings inside of a list of data, for example, a configuration. If you are using a string that is syntactically aligned with Cisco IOS devices, you need to ensure that it searches only Cisco IOS configurations. Searching configurations from other vendors probably does not find that particular string. The device might be configured correctly, but you are searching for the wrong data.

Definitions usually return *true* or *false*. However, definitions can also return *not assessed*. In this situation, the data structure does not meet certain criteria to allow further searching. A later lesson describes the details of a not-assessed definition. Regardless, a device that is not-assessed is neither compliant or noncompliant.

The definition uses logic gates that turn the concept of *true* or *false* into concepts of *compliant* and *noncompliant*. The administrator must understand what *true* and *false* mean in the context of a rule. Does the definition search for commands that must be configured or must not be configured?

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Rules are vendor-specific

Rules must be vendor-specific because of definitions

Rules that are run on the wrong vendor can produce an incorrect result (a false positive)

© Copyright IBM Corporation 2016

Rules are vendor-specific

When you create a definition, you do not determine that the definition is only run on Cisco IOS devices. You determine this filtering in the rule. Definitions that are written for Cisco IOS must never run on Juniper JUNOS devices. Rules must always have some level of VTMOS filtering.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Rules can be OS-specific

Cisco IOS changes its default states, what command is displayed in configuration

Cisco IOS 11.0 through 11.2, search for **`no service tcp-small-servers`**

Cisco IOS 11.3 and higher, search for **`^service tcp-small-servers`**

Name	Vendor	Type	Model	OS
3.1.62 disable tcp-small-servers 11.0-2	cisco	*	*	$range:11.0-11.2
3.1.64 disable tcp-small-servers 11.3+	cisco	*	*	$greaterThanOrEqual:11.3

Also, this is useful to solve problems between Cisco IOS and CatOS switches

© Copyright IBM Corporation 2016

Rules can be OS-specific

It is also possible that commands might be operating system-specific. For instance, as Cisco IOS moves between major releases, the commands are often compatible with later versions. However,

Cisco might reformat the commands or change their default states.

Cisco IOS has a habit of not showing the default state of some services. By not showing these default states, it is hard to know whether the service is enabled. To make matters worse, sometimes the default state changes between IOS releases, or even worse, within an IOS release.

Take the example of the command **service tcp-small-servers**. You should disable this service. In Cisco IOS 11.1 and 11.2, the command to search for in the configuration is **no service tcp-small-servers**. If you find this command in the configuration, you are assured it is not enabled. In the 12.x line of IOS Cisco changed this behavior, and now you must search for the command **service tcp-small-servers**. This behavior is the opposite of the IOS 11.x search. If you see this command, you must disable it. By disabling the command in the 12.x line of code, it removes the command completely from the configuration. In this situation, where you have two different behaviors, you need to have two separate rules that look at the different operating systems.

Not setting VTMOS filtering on rules easily lead to incorrect results. For example, Cisco has two types of operating systems on its switching gear. The Cisco IOS and CatOS operating systems use different syntax. Notice the difference between the two when defining a **syslog** host.

Cisco IOS command:

```
logging 10.191.101.52
```


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

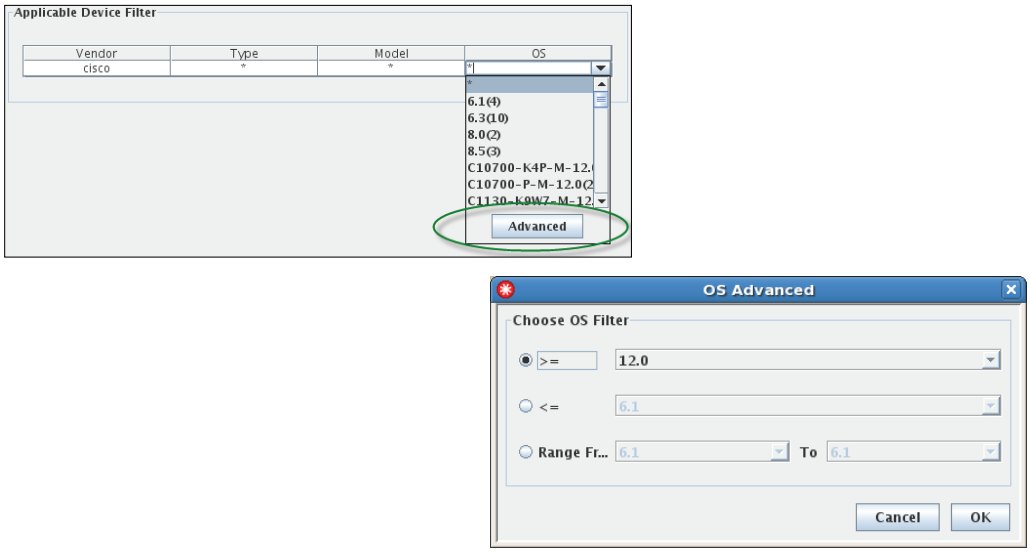
Cisco CatOS command:

```
set logging server 10.191.101.52
```

Searching for a command from the wrong IOS is likely to cause problems.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Advanced VTMOS selection



© Copyright IBM Corporation 2016

Advanced VTMOS selection

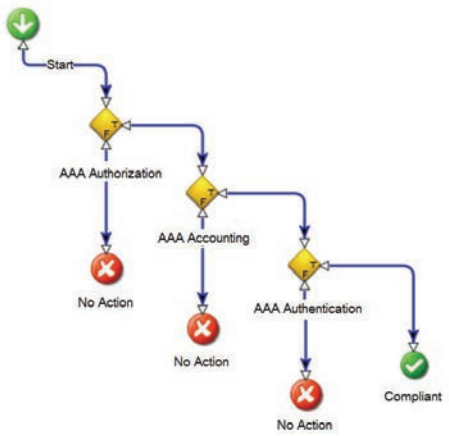
Fortunately, rules have some good VTMOS filtering options. Unlike *configuration management*, the *compliance manager* can ignore type and model and can have filters for vendor and operating system. More importantly, an advanced operating system selection capability is available.

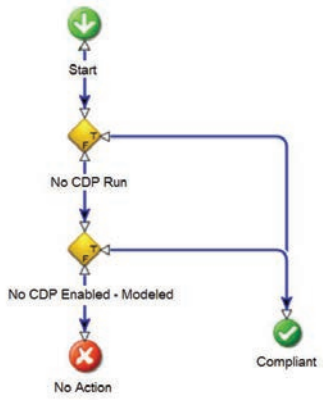
Using the advanced OS filter, you can select operating systems less than or greater than a certain level. The filter also allows for a range of OS values.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Rules use logic gates

- AND gates
- OR gates





© Copyright IBM Corporation 2016

Rules use logic gates

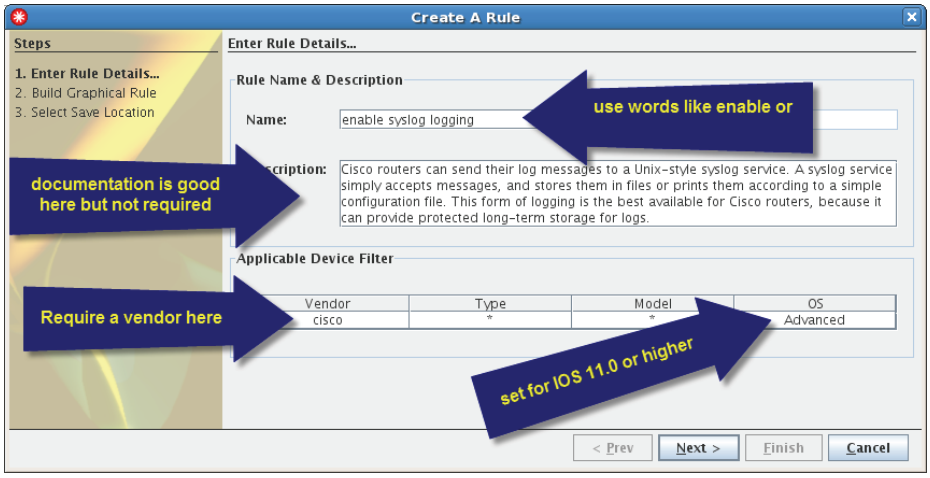
Rules allow the use of logic gates so that you can use AND and OR functions to take multiple definitions and use them to determine compliance.

For example, the AND logic gates test for three specific sets of commands on the device. Each definition must return *true* for the rule to indicate that the device is compliant. Each definition checks for a different portion of the authorization, accounting, and authorization (AAA) structure in the configuration.

In the OR logic, two different ways exist to determine whether the device is compliant. Cisco Discovery Protocol (CDP) is a feature that you can disable globally in the configuration with the command **no cdp run**. You can also disable the feature for each individual device interface with the command **no cdp enable**. Detecting *true* from either definition indicates that the device is compliant.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Create a rule



© Copyright IBM Corporation 2016

Create a rule

Creating a rule object requires that the user has *manage compliance policy* rights for their group.

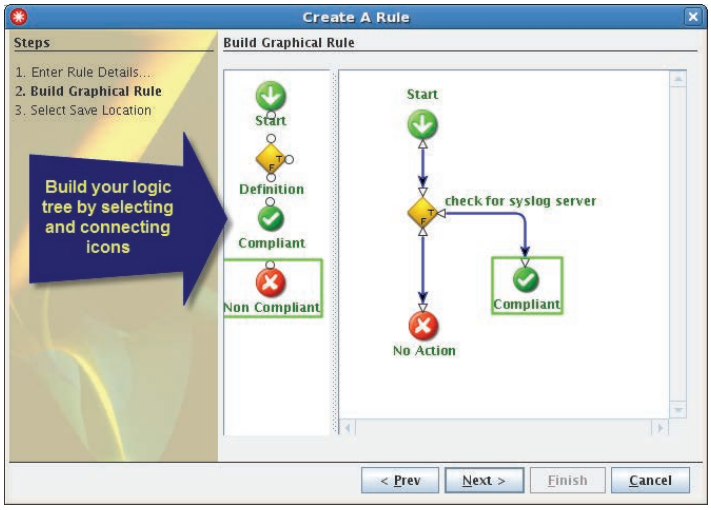
Creating the rule on the **Policy Definitions** tab brings up the initial screen the slide notes. It requires a name. In the naming convention, it makes sense to use words like *enable* or *disable*.

Documentation helps users to interpret a rule result. You document the purpose and function of a rule with the description field. The details section of a compliance result shows the rule description.

You must always define at least the vendor in the device filter. In the example on the slide, the rule applies only to operating systems IOS 11.0 or higher. This setting ensures that the rule does not run on Cisco CatOS devices that have an operating system of 8.x or lower.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Create the compliance logic



© Copyright IBM Corporation 2016

Create the compliance logic

You click **Next** to create the compliance logic. Here you must select one or more definitions and associate *true* with a compliant or noncompliant condition. First, drag a start icon onto the screen. Next, drag the definition icon. It opens a window that you use to either select a definition that exists or to create a new one.

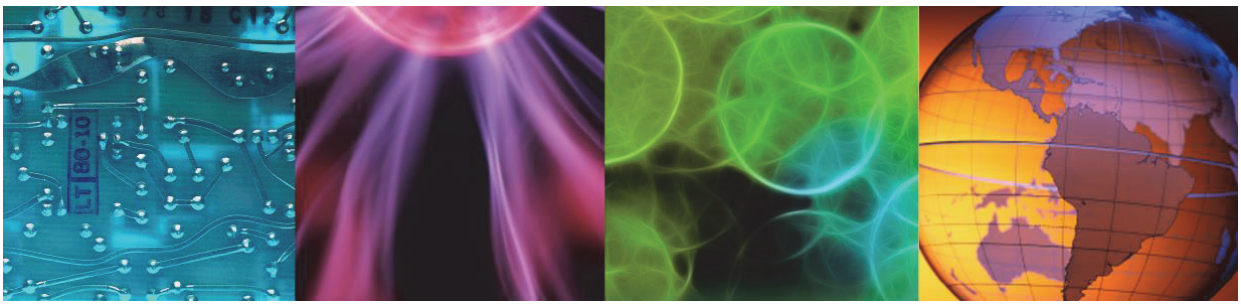
Afterward, add the compliant and noncompliant icons to define the values of *true* or *false*, Dragging the noncompliant action into the rule prompts you to select a remedial action to associate with this rule.

Be sure to connect all the objects with the green lines by clicking the white nodes on each object and dragging them to the correct, corresponding node.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 Email action

Lesson 4 Email action



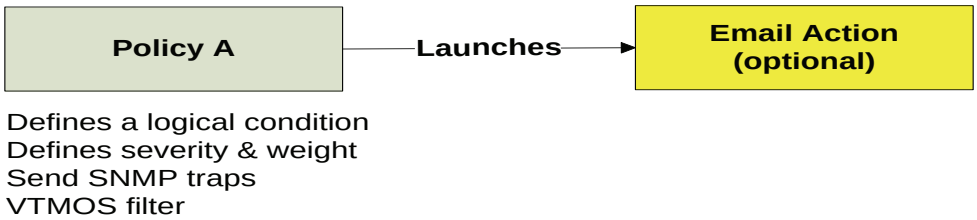
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe what is included in an email actions
- Describe when an email is sent to users

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies start optional email actions



© Copyright IBM Corporation 2016

Policies start optional email actions

It is possible to provide email notification for devices that violate a policy. This notification is done through an email action, which you build and associate with a policy.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Email action overview

Sends email on a per-policy basis for noncompliant devices

Sends email to an SMTP server defined at installation time

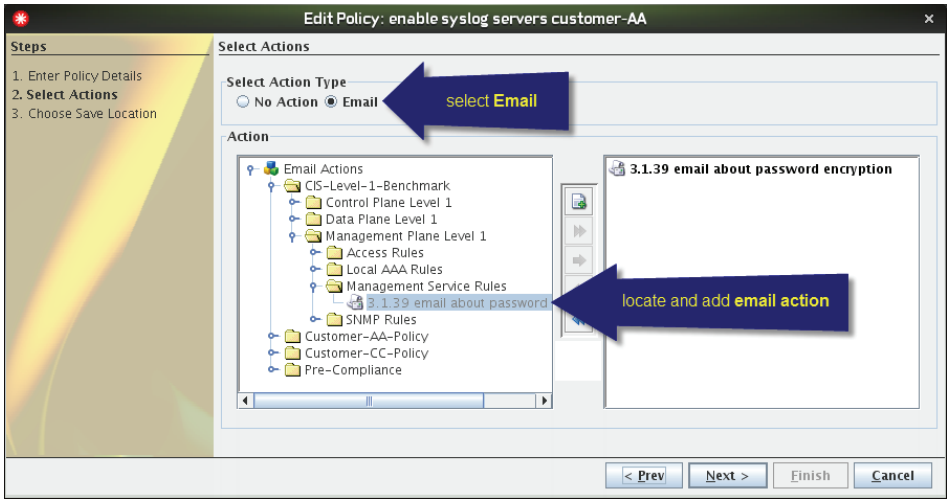
© Copyright IBM Corporation 2016

Email action overview

You use the action to create an email template that defines a distribution list of addressees, a subject, and an email body. Using this template, it creates an email that has the list of all the devices that violate the policy. This action creates an email on a per-policy basis for any devices that are found in violation. The emails are forwarded to a simple mail transfer protocol (SMTP) server that relays them to the addressees.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Associate email action to policy



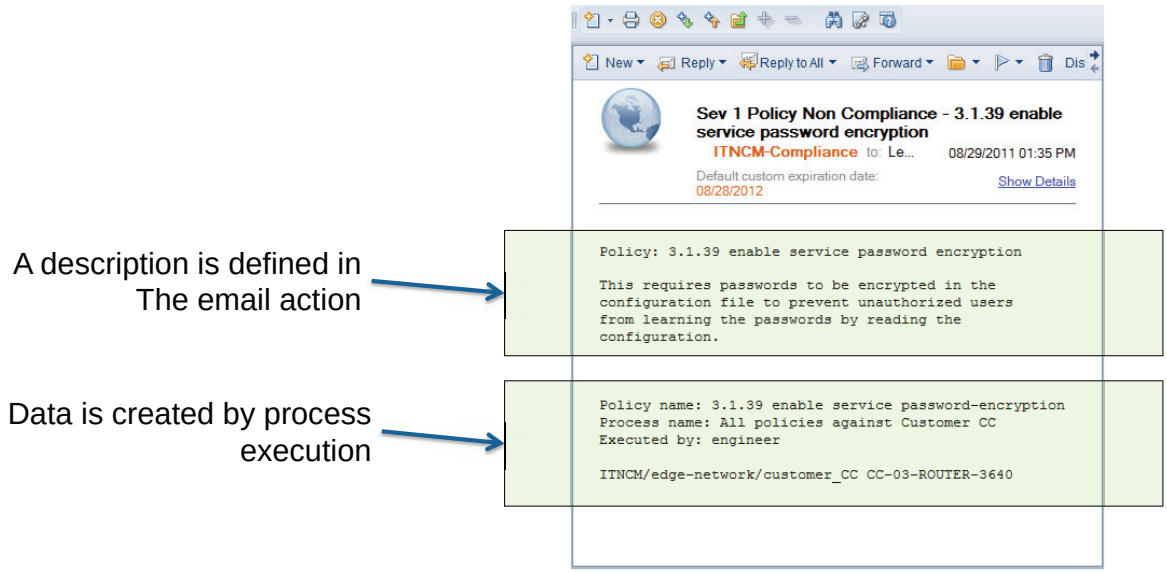
© Copyright IBM Corporation 2016

Associate email action to policy

When you build a policy, it prompts you to determine whether you want to add an email action. If so, search for the email action that you created and add it to the list. You can add email actions to a policy at any time.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Email example



© Copyright IBM Corporation 2016

Email example

The slide contains an example of the email that is sent to a recipient. It contains the information that you place in the template and the data that the *compliance manager* adds. The added data describes the devices that violate the policy.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Emails are only created by a process

Only processes create side effects like email

Emails reference only noncompliant devices that are selected in process

Email addresses are the same regardless of the process that uses the policy

© Copyright IBM Corporation 2016

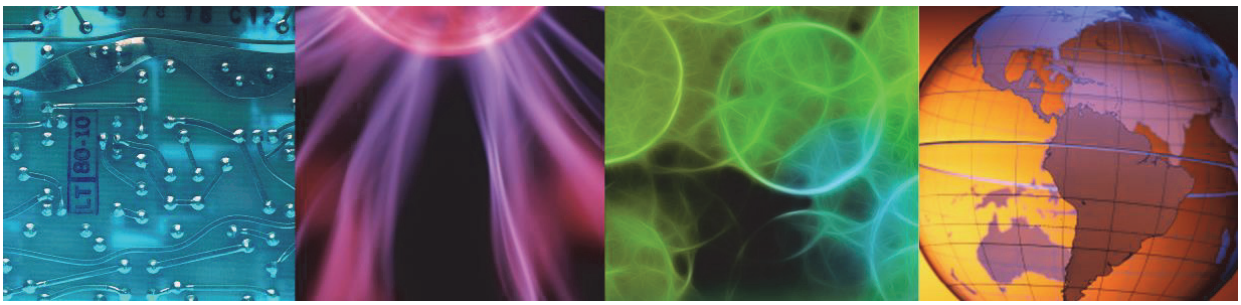
Emails are only created by a process

Emails are sent only when processes run. The emails send only the names of the devices that violate a policy. An email action has a hardcoded list of recipients. Use caution when you use a policy in more than one process.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 5 Build and test a policy

Lesson 5 Build and test a policy



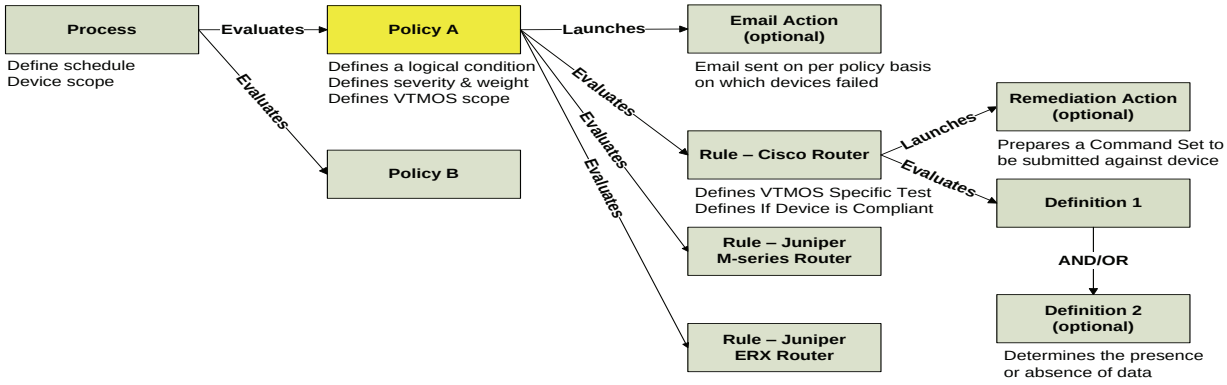
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the features of a policy
- Test a policy against one or more devices

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a policy



© Copyright IBM Corporation 2016

Creating a policy

When you fully define a rule, you can create a policy. After you create a policy, users can test their definitions.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies define a vendor-neutral condition

Passwords should be encrypted

Unused interfaces should be shut down

Connections should be through secured terminals

© Copyright IBM Corporation 2016

Policies define a vendor-neutral condition

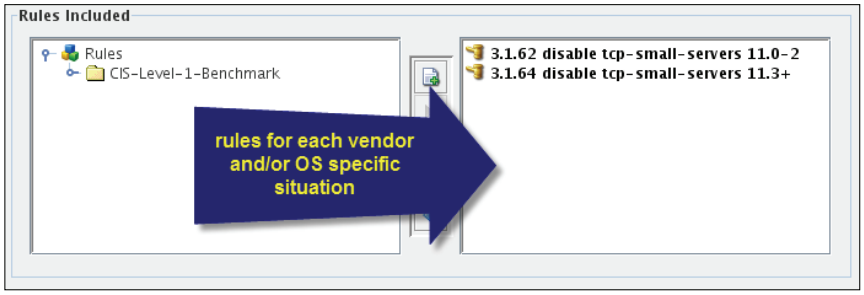
Policies are typically vendor-neutral conditions that you should configure on devices. You write policies to address a specific vendor issue, especially when it is an OS problem that is limited to a small segment of the network. However, the idea is to have generic policies that multiple vendor-specific rules address. A policy that verifies that SNMP is correctly configured across a multi-vendor network is an example of a generic policy.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies are a collection of vendor-specific rules

Each rule should be addressing the generic condition in a vendor-, model-, and OS-specific way

A device should resolve to only one rule in a policy



© Copyright IBM Corporation 2016

Policies are a collection of vendor-specific rules

Each rule should address the generic condition in a VTMOS-specific way. For example, to verify that the device is correctly configured for **tcp-small-servers**.

When you create a policy, you associate one or more rules with it. If there is more than one rule, it is imperative that the rules use VTMOS filtering. It is important when a policy processes against a device that the device uses only one of the rules. If it applies both rules to a single device, a good chance exists that you receive a false-positive response.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Severity and weight

Severity is usually how policy is defined

- Corporate policy usually defines severity level
- Severity levels are 1 (most severe) to 5 (least severe)
- Severity is used in most reports

Weight is used to develop compliance score

- You can place more importance on various policies

Severity and weight should align

© Copyright IBM Corporation 2016

Severity and weight

You must associate a severity and a weight with a policy.

You define a problem in the network in terms of its severity, for example, a severity 2 bug. The *compliance manager* has severity levels that match the severity levels in Netcool/OMNIbus. Five levels exist. A severity 1 policy is most important, and severity 5 is the least important. Use severity in reporting, to group compliance issues by each level.

You use weight to develop a compliance score. Its value is 1 - 100. The compliance score is shown in certain reports to provide a metric that you can evaluate over time as you implement policies in the network. The higher the weight, the more effect it has on the compliance score. One policy violation with a weight of 100 has the same effect on the compliance score as five policy violations with a weight of 20.

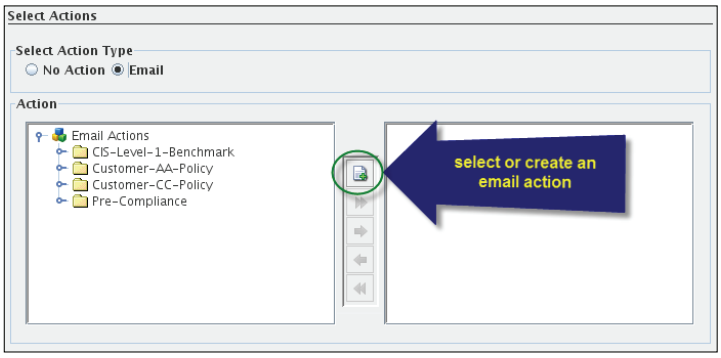
It is important that weight and severity align.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies can create side effects

SNMP traps sent for noncompliant devices

Emails can be sent when devices are noncompliant, but you must create an email action



© Copyright IBM Corporation 2016

Policies can create side effects

You can configure policies to create side effects when a device is not compliant.

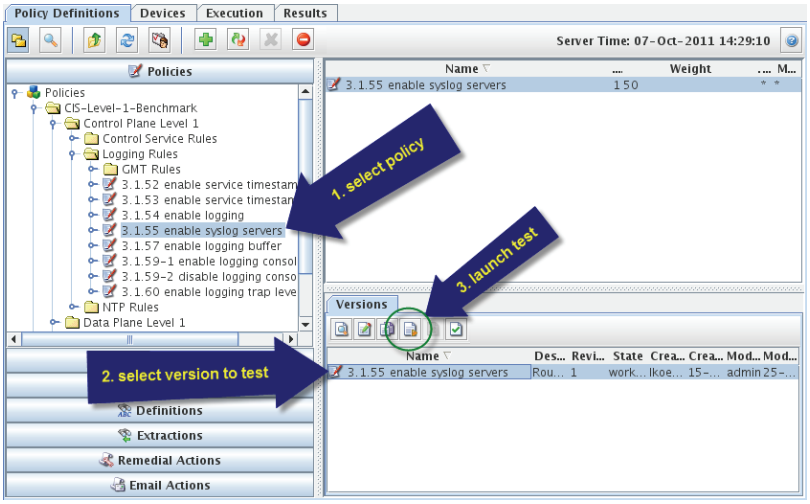
For example, you can configure a policy to send SNMP traps to an event listener. When the policy processes against a series of devices, it creates a trap for each device that it finds noncompliant.

A policy can also use an email action to send an email to concerned parties when it finds noncompliant devices. In this situation, only one email is sent with a list of all noncompliant devices it finds. The email is sent after the policy finishes with the analysis of all devices.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Policies can be tested

After a policy is defined, it can be tested



© Copyright IBM Corporation 2016

Policies can be tested

Finally, one of the more important aspects when you create the policy is that you can test it. Here you determine whether the expressions you use in the definition work as you expect. Use the following steps to test a policy:

1. Select the policy from the left pane of the **Policy Definitions** tab.

   The **Policy Definitions** tab GUI shows the policy in both the upper and lower panes.

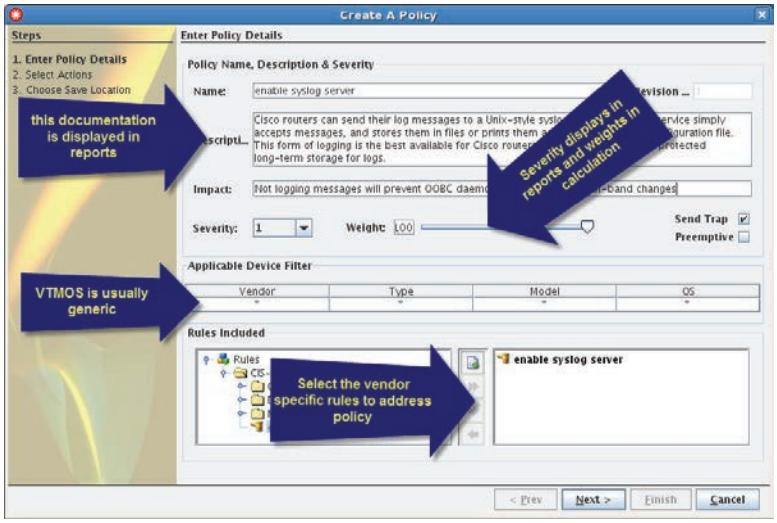
2. Select a version of the policy in the lower screen.

   When version control is in place, you might have more than one version of the policy.

3. Click the **Test Policy** icon to begin the testing process.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Create a policy



© Copyright IBM Corporation 2016

Create a policy

Creating a rule object requires that the users have *manage compliance policy* rights for their group.

The policy name and documentation is important at this stage as it is used throughout reporting. Having good documentation helps people to understand why this policy is in place.

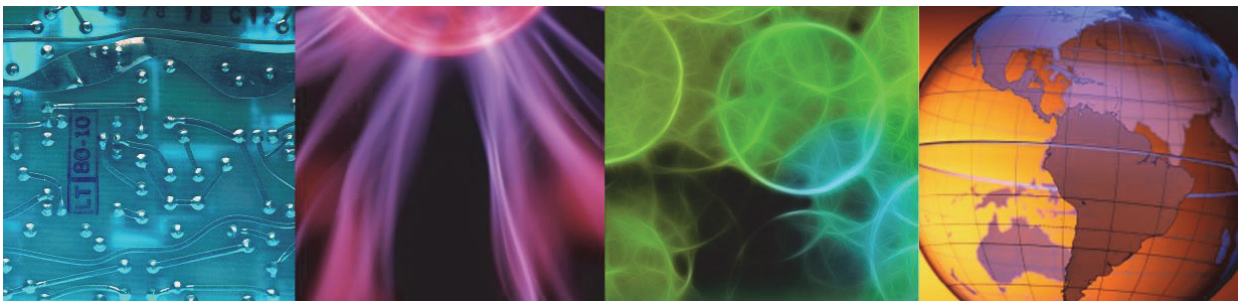
Notice on this screen that you can configure the **Severity**, **Weight**, and **Send Trap**. You can set VTMOS criteria for this policy. For most policies, this filter is left with full wildcard characters. You must also select one or more rules that are part of this policy.

You use the next step in this wizard to add an email action to the policy. The final step is used to save the policy in the policy realm structure.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 6 Exempting a device from a policy

Lesson 6 Exempting a device from a policy



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to describe policy exemption and its impacts.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Reasons to exempt devices from a policy

Exemptions are different from not-assessed devices

Importing new devices can affect compliance score

Exempted devices do not count against compliance score

Exemptions should have limited time horizon

© Copyright IBM Corporation 2016

Reasons to exempt devices from a policy

The *compliance manager* allows the organization to exempt devices from specific policies. Do not confuse exemptions with not assessed.

A device that is not assessed means that either the VTMOS does not align with a rule in a policy, or that the configuration does not have the prerequisite data for a full analysis. The *compliance manager* code indicates the not assessed condition.

The compliance administrator configures a device that is exempted to not evaluate against a policy. This exemption comes from organizational decisions.

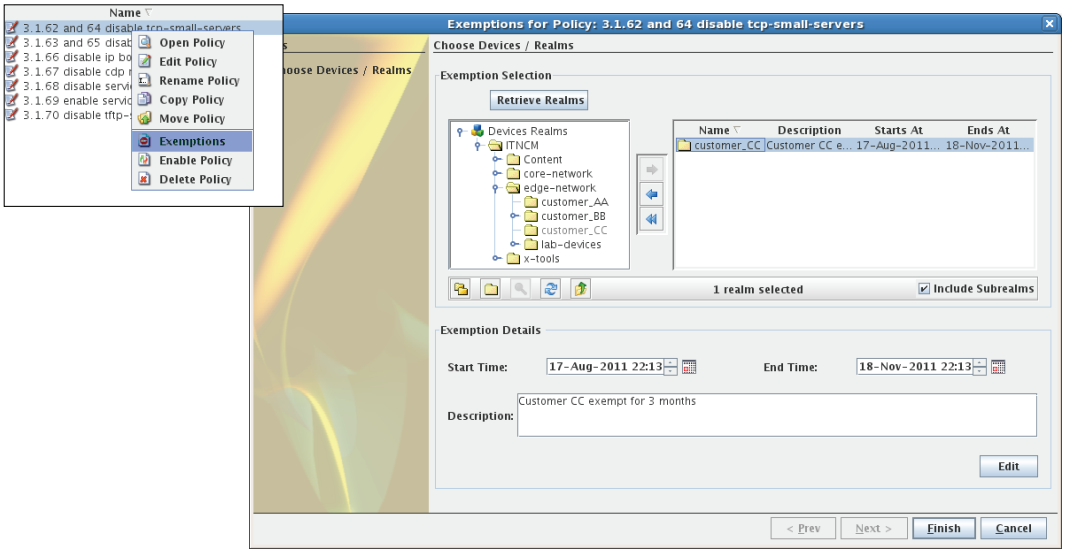
An example of this exemption is importing a new network from an acquisition. This network reflects poorly on the compliance score when policies are run against it. By exempting this network, you prevent a major change in the compliance trend lines.

Exempted devices do not count against your compliance score.

These exemptions must have a time at which they expire.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Exempting devices



© Copyright IBM Corporation 2016

Exempting devices

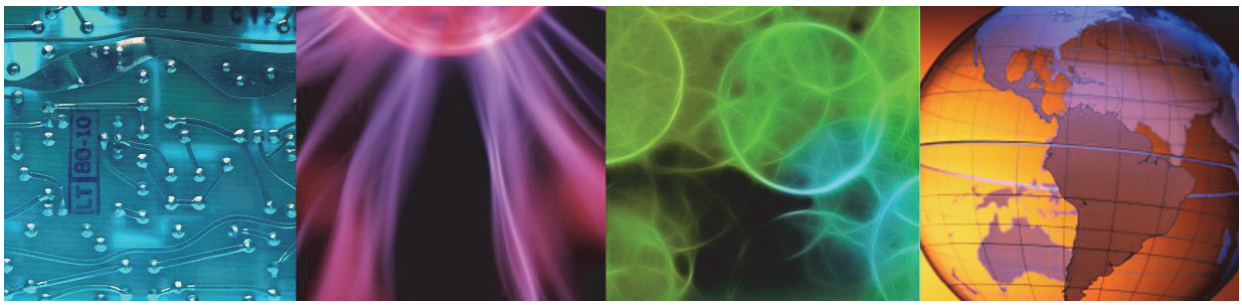
Exempting devices happens with individual policies. To exempt a device, right-click the policy name to open a menu, select **Exemptions**. The **Exemptions** GUI opens.

You can select specific devices, device realms, and even VTMOS realms that you want to exempt. When you select the device level, the GUI prompts you for a start and end time for the period that the exemption is in effect. You must also enter a description of why you require the exemption.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 7 Build and run a process

Lesson 7 Build and run a process

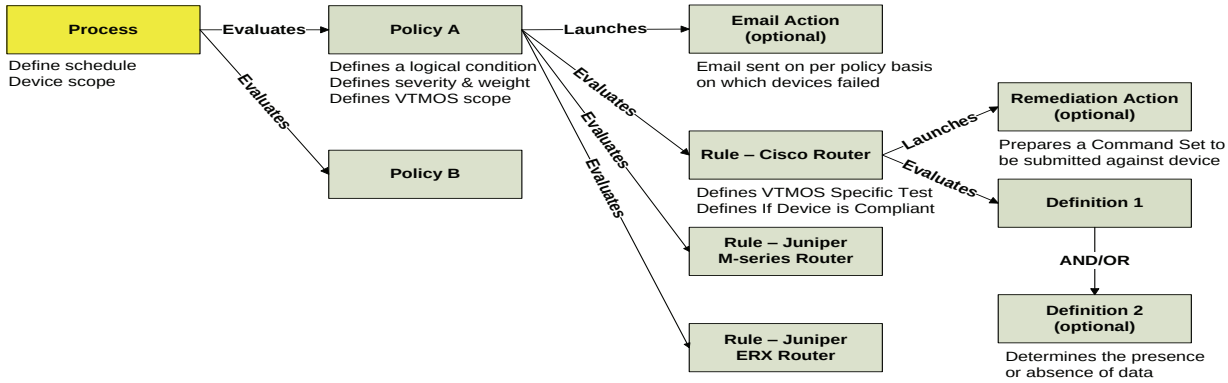


© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to configure a process.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a process



© Copyright IBM Corporation 2016

Creating a process

After you create one or more policies, you can associate them with a process. Processes define a list of policies, a list of devices, and a schedule of when to run the analysis.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Processes evaluate policies against devices

Select policies

Determine precompliance (described later)

Select devices or realms

Set parameter values (described later)

Determine schedule and recurrence

© Copyright IBM Corporation 2016

Processes evaluate policies against devices

The most important aspect of a process is the selection of policies. Policies determine compliance for the many services that you can configure on a single device. A process gathers a series of policies and uses them to analyze a portion of the network.

The process is where you can enable the function *precompliance*. Precompliance allows policies to analyze changes you propose to the network and determine whether they are compliant.

When you create a process, you select the devices, device realms, or VTMOS realms that the policies run against. You can also define values for policy definitions that have parameters that are associated with them. Finally, you must configure the schedule that determines when and how often the process runs.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

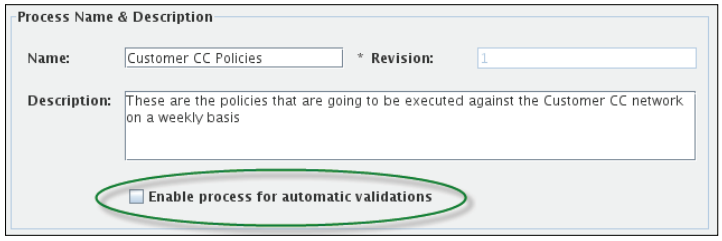
Automatic validations

Any Netcool Configuration Manager state changes to device causes reanalysis of policy

- Configuration imports
- Synchronizations
- Configuration changes and command sets

Affects only devices in process

All policies in process are applied to device



© Copyright IBM Corporation 2016

Automatic validations

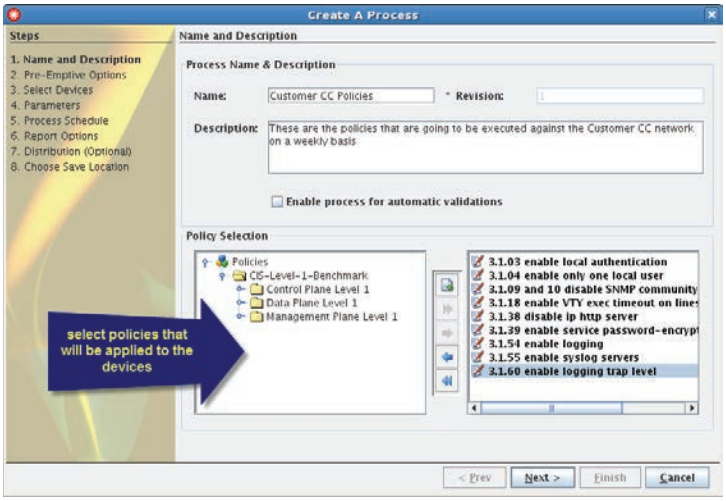
You can define the process so that it automatically validates its policies against one of its devices when a state change occurs. A state change on a device is when one of the following unit of work actions is run on it.

- Import
- Configuration synchronization
- Configuration change
- Command Set application

The *compliance manager* monitors for events that occur in the *configuration manager*. These events signal a possible change in the configuration of the device. If a process is enabled for automatic validations, the event that is signaled from the configuration manager is analyzed. It checks the device name that has the potential configuration change against all the enabled processes. If one or more of these automatic processes includes that device, each process runs it policies on that device and updates the database.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Select the policies to include in the process



© Copyright IBM Corporation 2016

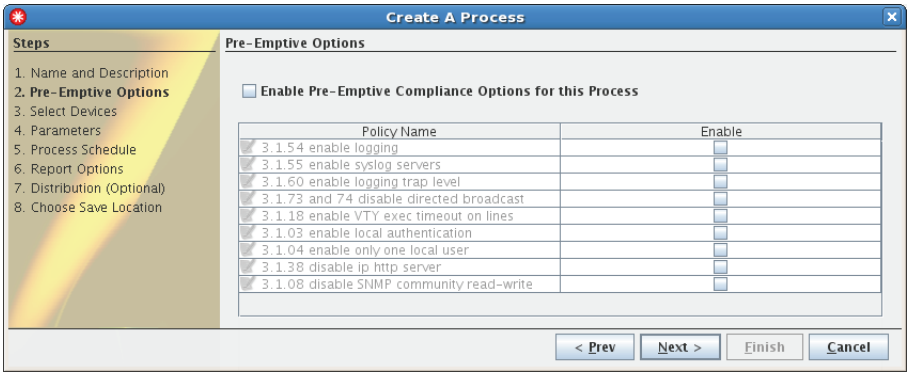
Select the policies to include in the process

The first step in creating a process is to select the policies to include.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preemptive compliance

The topic of preemptive compliance is described in an upcoming chapter



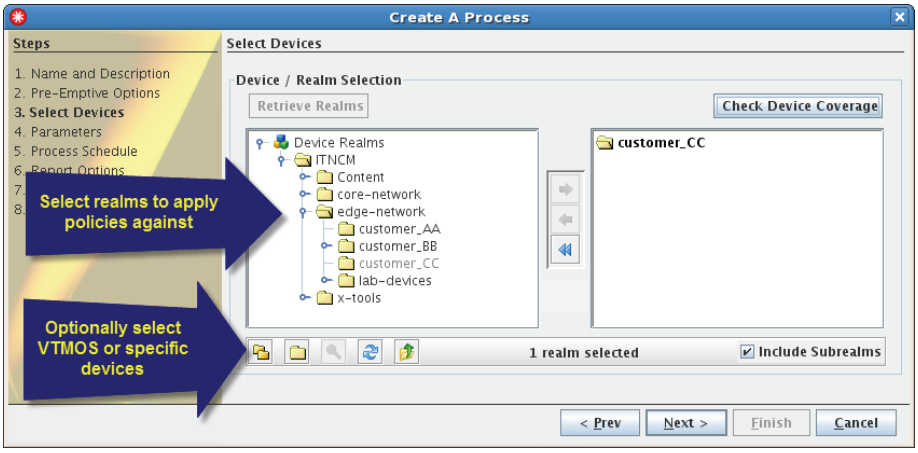
© Copyright IBM Corporation 2016

Preemptive compliance

Preemptive compliance allows certain policies to check changes that units of work submit. It provides an inspection-like service to SmartModel changes that can prevent changes that take devices out of compliance.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Select realms or devices



© Copyright IBM Corporation 2016

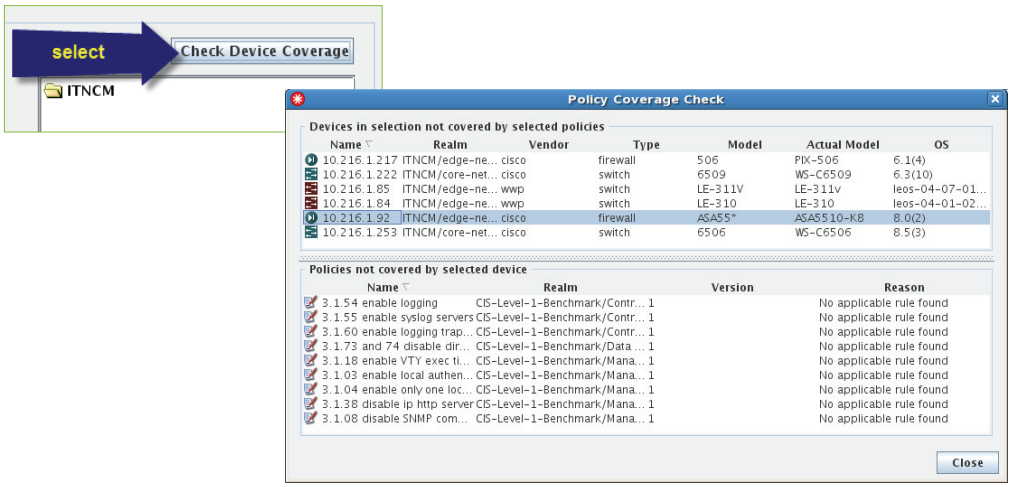
Select realms or devices

The third step is to decide which devices to include in the process. Most processes use the device realms that are defined in the configuration manager. However, you can use VTMOS realms to support a process that is specific to certain operating systems.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Device coverage

Show devices that are not analyzed



© Copyright IBM Corporation 2016

Device coverage

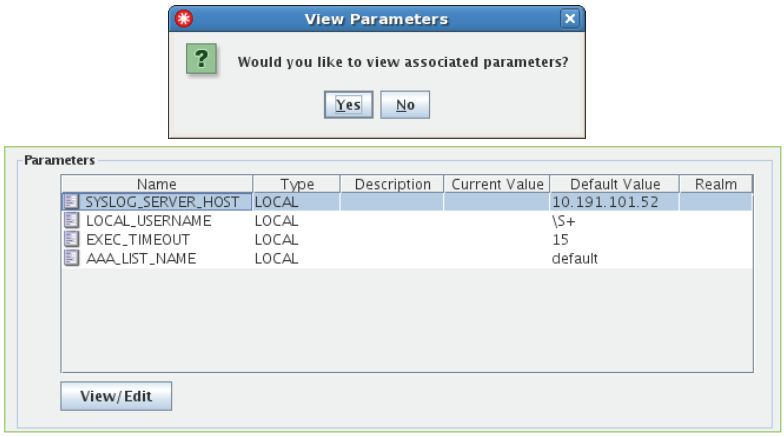
After you select a series of realms or devices, you can check the device coverage. When you click **Check Device Coverage**, the Policy Coverage Check window opens. Use this window to see all the devices that the process might not analyze. You can click a device name to view a list of policies that the selected device.does not cover. The most common reason for a policy not being analyzed is that no applicable rule exists for the VTMOS of that device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Setting local parameter values

Set local parameter values

A later unit describes parameters



© Copyright IBM Corporation 2016

Setting local parameter values

If the policy definitions you select contain any parameters, you can view the parameters and change their default values for this particular process.

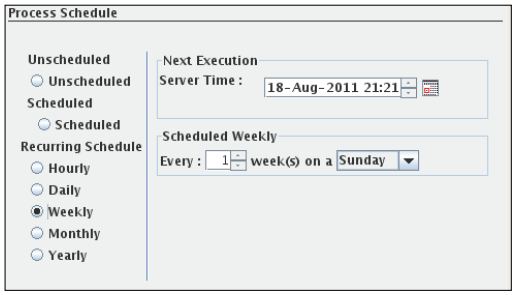
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Scheduling process

Unscheduled
- Must be run manually

Scheduled
- Runs only once at some future date

Recurring Schedule
- Launches at scheduled time
- Relaunches at specified intervals (hours, days, months



© Copyright IBM Corporation 2016

Scheduling process

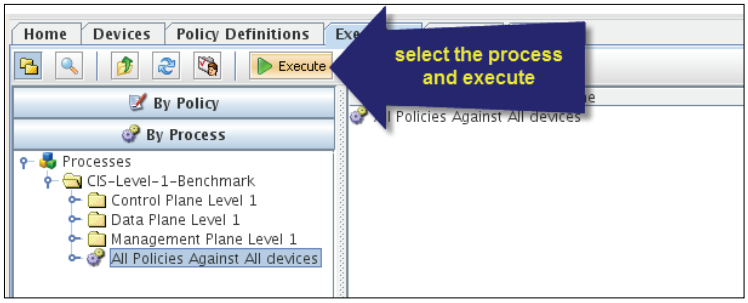
The user who creates the process must determine when the process can run. The user can choose three options.

- **Unscheduled**: The process runs only when a user manually starts it. In other words, the process never runs unless a user goes to the **Execution** tab, selects it, and starts it.
- **Scheduled**: The process runs only once at some future date. Later, when run, the user must manually start this process from the **Execution** tab in the GUI.
- **Recurring Schedule**: The process runs regularly on an hourly, daily, weekly, or monthly schedule.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Running processes

Need to run unscheduled processes



Processes that run create side effects

- Remedial actions
- SNMP traps
- Email

© Copyright IBM Corporation 2016

Running processes

If you create an unscheduled process, you must run it from the **Execution** tab. Select the process by going to the process section, selecting it, and clicking **Execute**. This process generates side effects such as remedial actions, SNMP traps, and email.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Define the components that make up a policy
- Create and test simple policies
- Exempt devices from policies
- Build and run a process

© Copyright IBM Corporation 2016

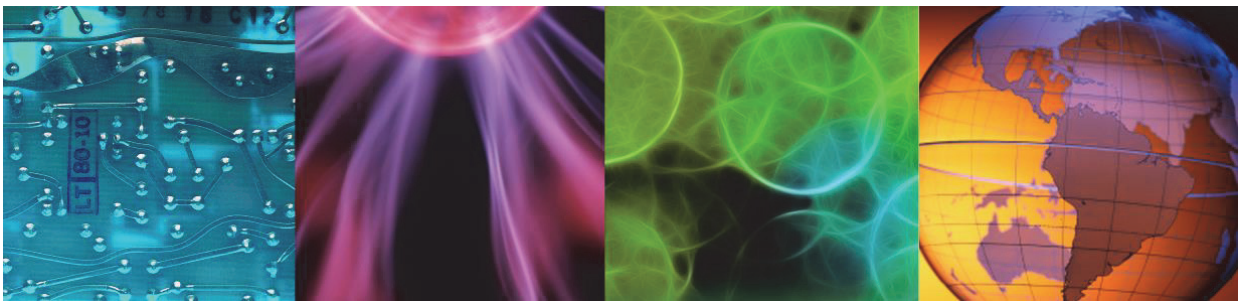
Summary

This unit provided an overview of the objects that make up a compliance policy and the role of objects in the compliance process. This unit showed how to create and test a simple policy. It also taught the student how to exempt devices from a policy and how to include a policy into a production process for scheduled execution.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

18 Remediation

Remediation



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Remediation is an important feature of the *compliance manager*. This unit describes how remediation works and how you configure it. It explains how remedial actions are approved and how to use parameterization to ensure that the changes sent to remediate devices have the correct values. You learn how to implement approvals for remedial actions.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Create a remedial action and describe how it works in Netcool Configuration Manager and compliance workflows
- Link command sets to remedial actions
- Describe how parameters link remedial actions to command sets
- Implement compliance approvals

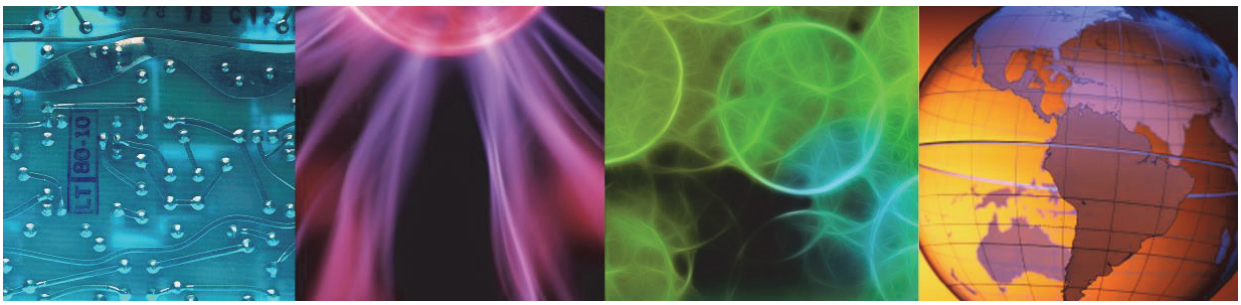
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Remediation actions

Lesson 1 Remediation actions



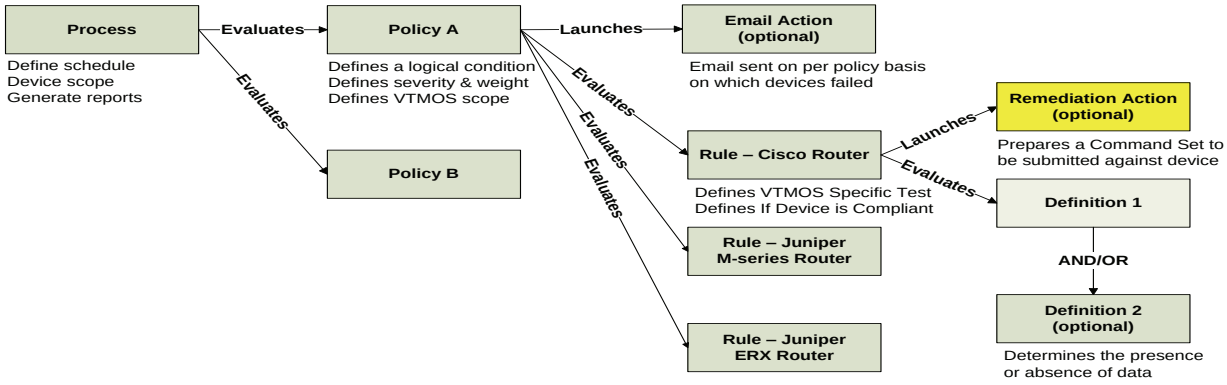
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe how to link a command set to a remedial action
- Describe the options available to a remedial action

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a remediation action



© Copyright IBM Corporation 2016

Creating a remediation action

A remediation action, or remedial action, is an object that you create and associate with a rule. Its purpose is to prepare a command set with appropriate values to fix a configuration that is not compliant. The remedial action, when approved, creates a unit of work in the configuration manager to fix the problem that compliance discovers.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Key points of the remediation action

Optional object that is associated with a rule

Must create before associating with the rule

Remedial actions point to a command set in the configuration manager

© Copyright IBM Corporation 2016

Key points of the remediation action

Remedial actions are optional and you do not need to include them in the compliance solution. However, they are probably the most useful portion of the software. This object is where the compliance solution and configuration management allow quick discovery and resolution of poorly configured devices. Remedial actions link into the solution through their addition to rules.

While technically not necessary, you must create remediation actions before attempting to apply them to rules. You can create remedial actions on an as needed-basis when creating rules, but this method is not as effective.

When you create a remedial action, you need to point it to a command set that you pre-define in the *configuration manager*.

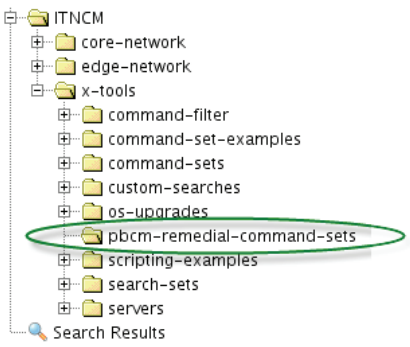
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

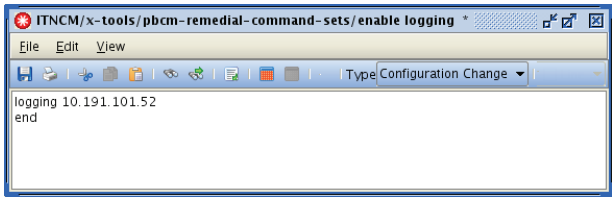
Verify that the command set is defined first

Compliance cannot build command sets

Build command set in Netcool Configuration Manager in specific command set realm

Synchronize the command set to compliance, not required





© Copyright IBM Corporation 2016

Verify that the command set is defined first

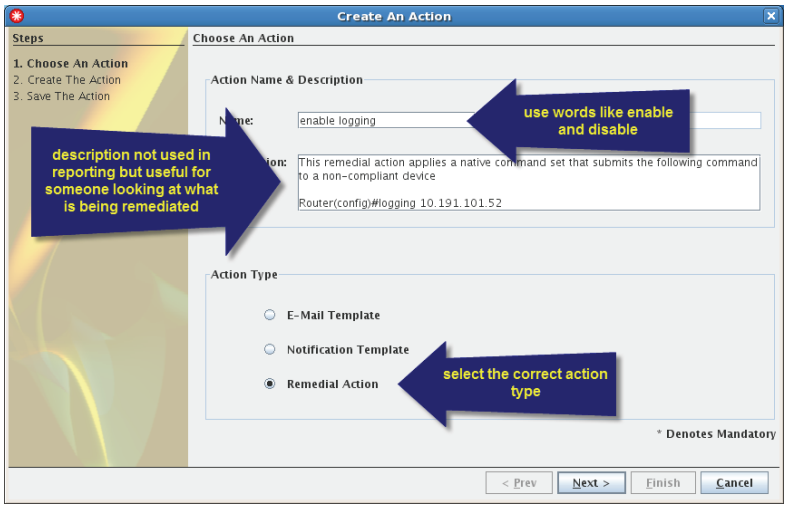
The *compliance manager* currently cannot build command sets. You must build a command set in *configuration manager* before you build the remedial action. The command set can either be a native or modeled command set.

Create specific realms in *configuration manager* to contain remedial command sets. The *configuration manager* does not indicate whether a command set links to a remedial action or not. By keeping the command sets inside this realm, it makes it obvious from the *configuration manager* that the command sets link to compliance and you should treat them carefully.

Compliance directly looks into the tables that define command sets to create the linkage. As long you are authorized to view the command set realm in the *configuration manager*, you can view them in compliance.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Build a remediation action



© Copyright IBM Corporation 2016

Build a remediation action

You create remediation actions in the **Policy Definitions** tab with the same methods that you use to create policies, rules, and definitions.

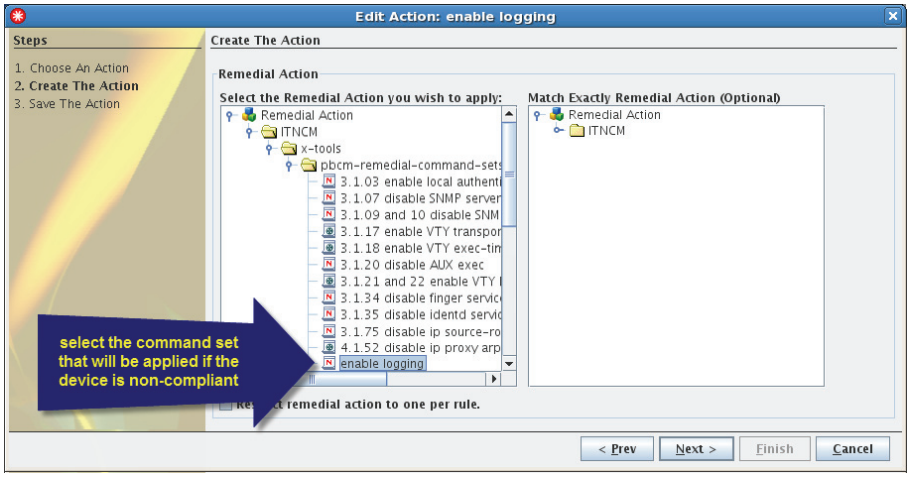
A remedial action is an object that enables or disables a service by changing a configuration. Use a naming convention that reflects this type of activation and de-activation.

Reporting does not use the description, but a description is useful for administrators who review policies and policy goals. In the Description field, include the CLI that is sent to the device to enable or disable a service.

Be sure that you select the **Remedial Action** option before clicking **Next**.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Link a remediation action to command set



© Copyright IBM Corporation 2016

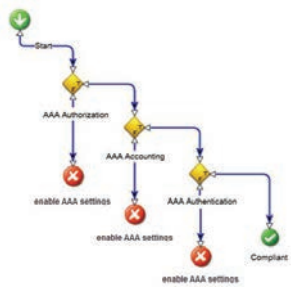
Link a remediation action to command set

After clicking **Next**, you choose a command set from the *configuration manager*. This step creates a link back to the *configuration manager*.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Optional remediation steps

Restrict remedial action to one per rule



Match exactly remedial action

© Copyright IBM Corporation 2016

Optional remediation steps

You can use two optional steps to configure the remedial action.

1. Use the **Restrict remedial action to one per rule** option when you create a rule with AND logic. Each definition evaluates regardless of its position in the AND logic tree. If the rule is found to be noncomplaint, all the definitions that it finds noncompliant generate a remedial action. In the AND logic, as the three definitions evaluate, it generates up to three remedial actions. However, each remedial action uses the same command set to fix the problem. To prevent the automatic generation of three *units of work* for this device, click the **Restrict remedial action to one per rule** option. This option limits a device to only one UOW.

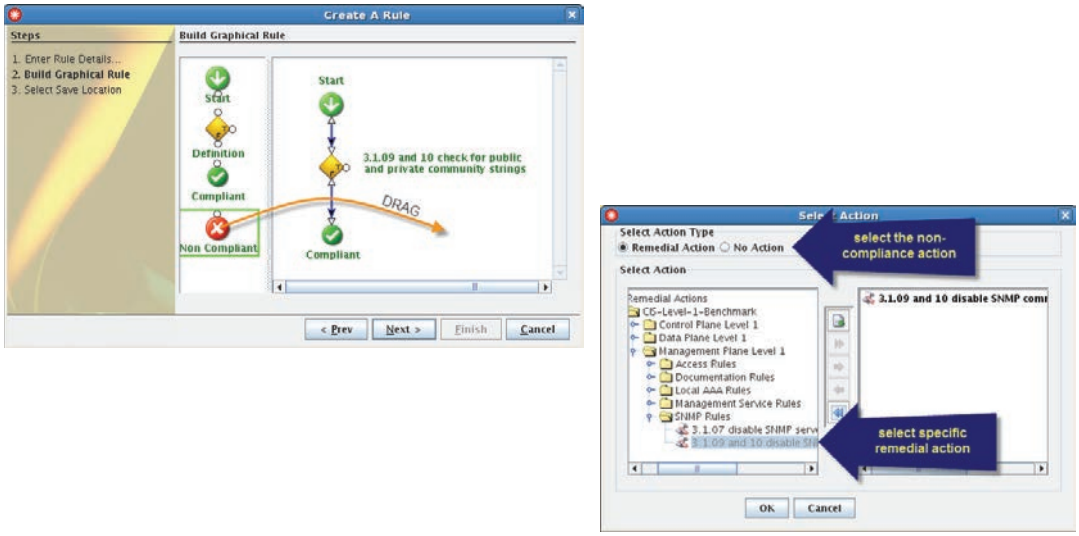
2. Use **Match Exactly** with group or extraction parameters. In these cases, you search for a set of values and only those values. If you add one of these parameters to your modeled or native evaluation, you get the option to add **Match Exactly** in the evaluation criteria.

The optional **Match Exactly** command set is another use case. You seldom use this option. It is available in remedial actions. Use **Match Exactly** when an evaluation in a definition is configured with a group or extraction parameters. These parameters search for a series of values. By using **Match Exactly**, it allows only those values in a configuration. If you have more values or less, it marks the rule as noncompliant.

To remediate **Match Exactly** failures, you need two command sets instead of one. Use one command set to add the required commands. Use the second to remove the extra commands that should not be there.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Associating the remediation action to a rule



© Copyright IBM Corporation 2016

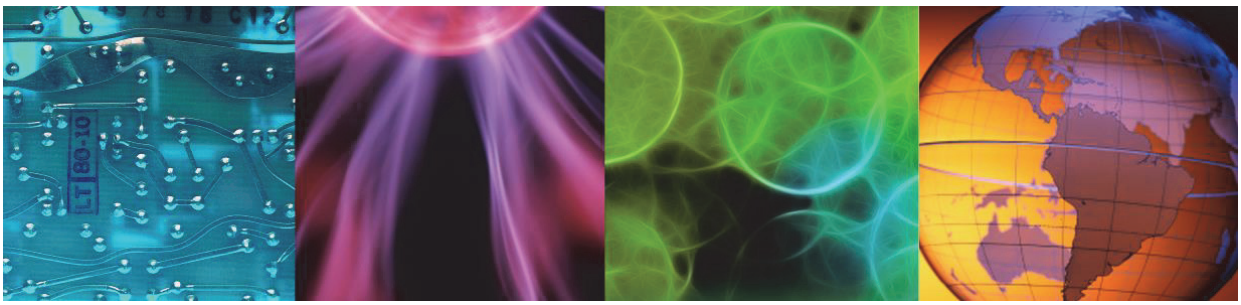
Associating the remediation action to a rule

After you create your remedial action, you associate it with a rule. When you drag the noncompliant icon onto the **Create a Rule** window, it prompts you to select either a remedial action or no action. Click the **Remedial Action** radio button, and in the policy realm navigation tree, select the remedial action that you create earlier.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Parameterization

Lesson 2 Parameterization



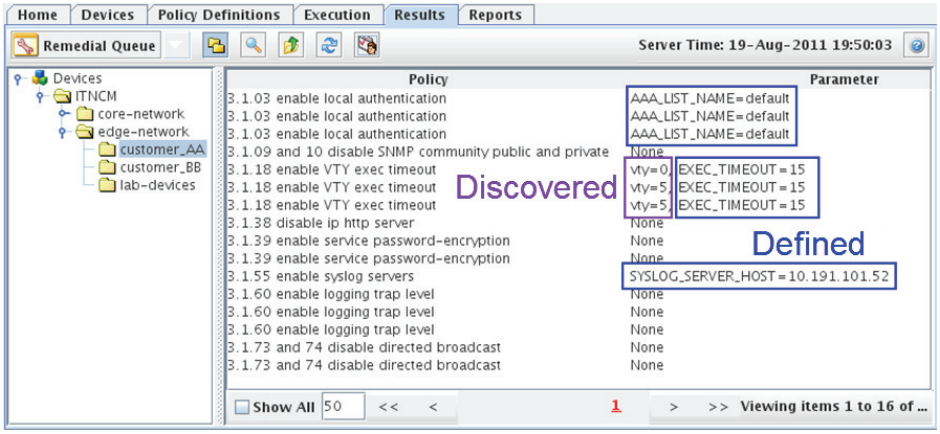
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to describe the differences between discovered and defined parameters.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Remediation notes parameter values

Parameters are defined or discovered



© Copyright IBM Corporation 2016

Remediation notes parameter values

A powerful aspect of remediation is the ability to pass the parameters that it uses or discovers in the analysis to the remedial actions that it uses to fix devices. These parameter values are the following two types.

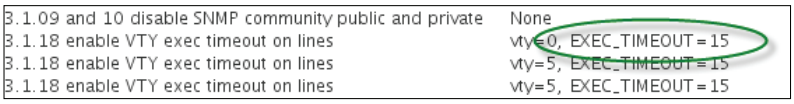
1. The defined parameter value is a value that is either the default value for a parameter or the value that you modify before the analysis starts. Explicit parameter values are defined in the evaluations that are run on the configurations. An example in the slide is **SYSLOG_SERVER_HOST**. This user-defined parameter is used in the **3.1.55 enable syslog servers** policy. In this case, its value is determined when the process is run and prompts the user to view or modify parameter data.

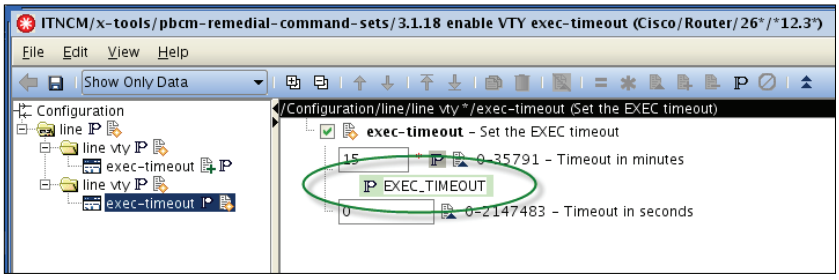
2. You find the discovered parameter value by searching through a modeled configuration. These values are present when you use a modeled definition with a context. The example in the slide is the **VTY** number. You discover this value when you analyze one of its subcommands for **EXEC_TIMEOUT**. When **EXEC_TIMEOUT** does not match the required value of 15, the modeled definition returns the **VTY** that is incorrectly configured. You discover the VTY value and can use it in a command set.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Parameter values can pass to command sets

Parameter must align between policy and command set





© Copyright IBM Corporation 2016

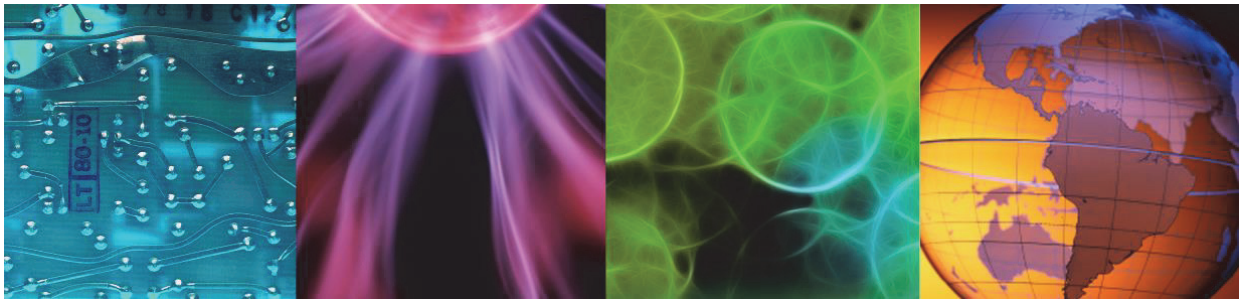
Parameter values can pass to command sets

You can use these parameters and their values, whether discovered or defined, in command sets. While parameters are not a requirement for remediation, they make it more flexible if they are present. The command set must have parameter names that exactly match the parameter name that evaluation finds or the parameter name that the modeled definition returns.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Approvals

Lesson 3 Approvals



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

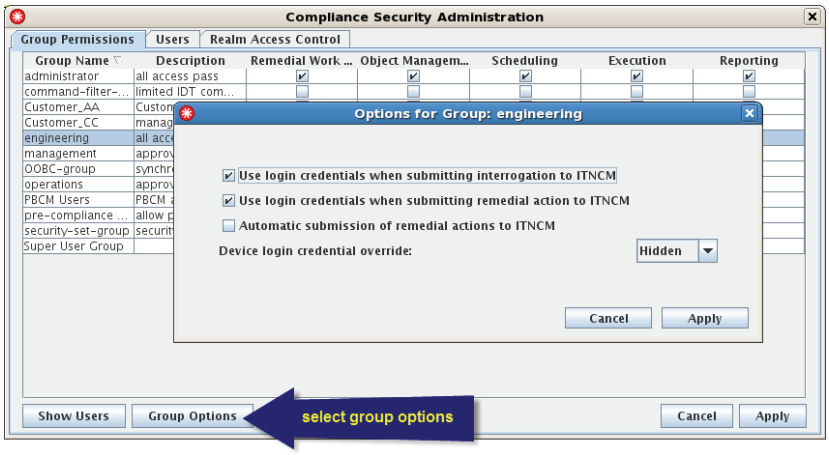
In this lesson, you learn how to configure remedial action approvals.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Remediation actions can be approved

Approvals can happen in both compliance and Netcool Configuration Manager

Based on the user who submits the UOW



© Copyright IBM Corporation 2016

Remediation actions can be approved

You can configure remedial actions so that they need approvals before the *compliance manager* creates the UOW for the *configuration manager*. Depending on the user who submits the UOW, the UOW might also need a *configuration manager* approval to run.

The *configuration manager* sets the approval of the remedial actions based on group membership. Group configuration is done through the *compliance security manager*. You can configure each group for the following two features for remedial actions.

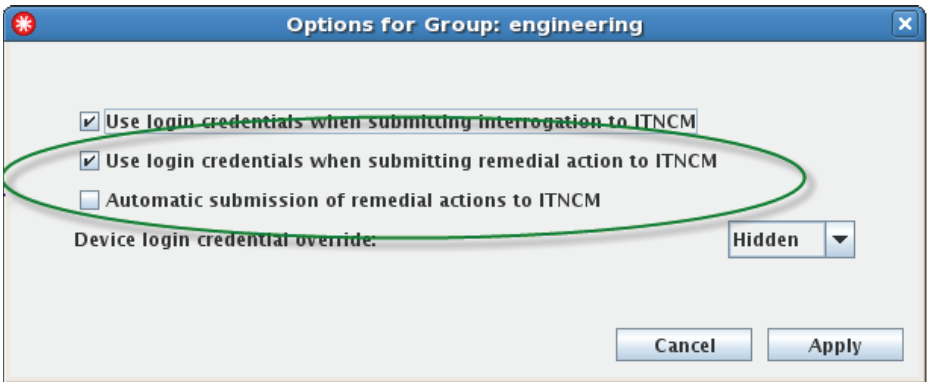
- **Automatic submission of remedial actions to ITNCM**: Selecting this option negates the need to use the remedial queue to manually approve or reject the action. Use of this option is infrequent. You need to carefully consider the impact of this option before you use it.
- **Use login credential when submitting remedial action to ITNCM**: By selecting this option, you override the need of the standard compliance user to submit the UOW to the *configuration manager*. The API handles the submission of the UOW. The API needs a set of credentials to log on to the *configuration manager*. By default, a standard user, that is defined at installation time, submits remedial actions. By checking this box, the credentials you use to log on to the *compliance manager* are the same credentials that you use to submit the UOW. This selection affects the UOW in two ways.
- The user name that is associated with the UOW is either the standard user name or the user who approves the remediation.
- The *configuration manager* workflow settings for the user who submits the UOW are in effect and might require approval of the UOW. This approval occurs from the queue manager in *configuration manager*.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Each group can have different settings

Does remediation really need to be approved in compliance?

What user do you want to see in configuration manager?



© Copyright IBM Corporation 2016

Each group can have different settings

Since each group can have different settings, it is possible to have different policies for each group.

The group settings come into play based on who submits a process for execution. If a user in a group selects **Automatic submission of remedial action to ITNCM**, and submits a process for execution that has remediations, the remediations create *units of work* without intervention.

While customers might be reluctant to enable this feature, use cases exist where it is useful. You might have configuration settings that have little impact on service delivery but improve security, for example, encrypting passwords. You can enable this feature on a Cisco device with little impact on its function and can apply it without issue. In this case, doing this configuration change automatically increases the security of the network without affecting productivity.

In this automation scenario, it also makes sense to use the standard compliance user to create this UOW. You can then filter these *units of work* in the queue manager with its sorting mechanisms.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Create a remedial action and describe how it works in Netcool Configuration Manager and compliance workflows
- Link command sets to remedial actions
- Describe how parameters link remedial actions to command sets
- Implement compliance approvals

© Copyright IBM Corporation 2016

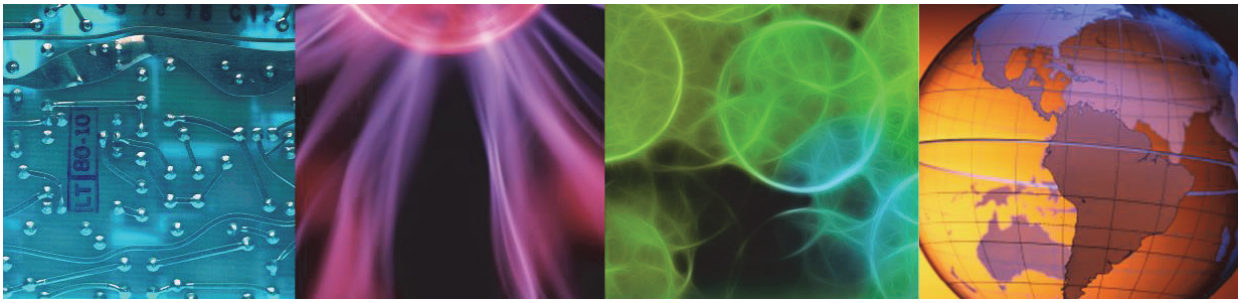
Summary

Remediation is an important feature of the *compliance manager*. This unit described how remediation works and how it is configured. It explains how remedial actions are approved and how parameterization is used to ensure that the changes sent to remediate devices have the correct values. You learned how to implement approvals for remedial actions.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

19 Advanced definitions

Advanced definitions



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches you about the more advanced features to determine whether a configuration is compliant. This unit describes the different data structures that you can analyze and the different methods for analyzing them. It describes parameterization of definitions in detail. Finally, it introduces the use of JavaScript.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Describe the types of definitions that you can create
- Describe the methods to use parameters in definitions
- Describe how to use JavaScript to enhance definitions
- Create a policy that uses XPath
- Create a policy that uses an extraction
- Test the new policies
- Create a process

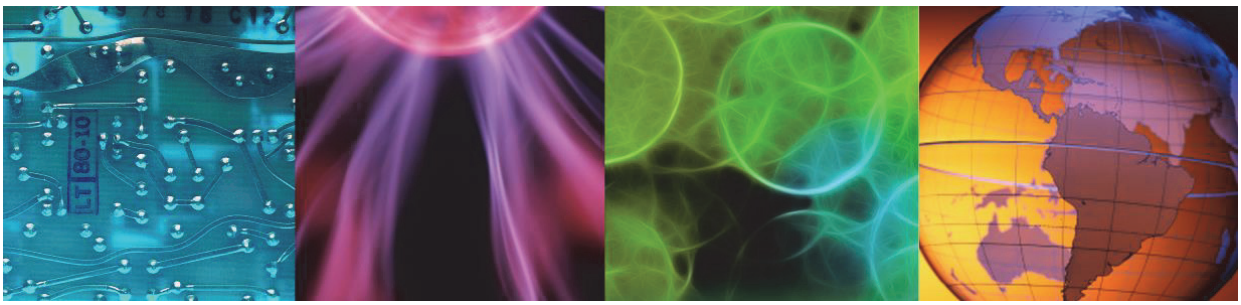
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 General concepts

Lesson 1 General concepts



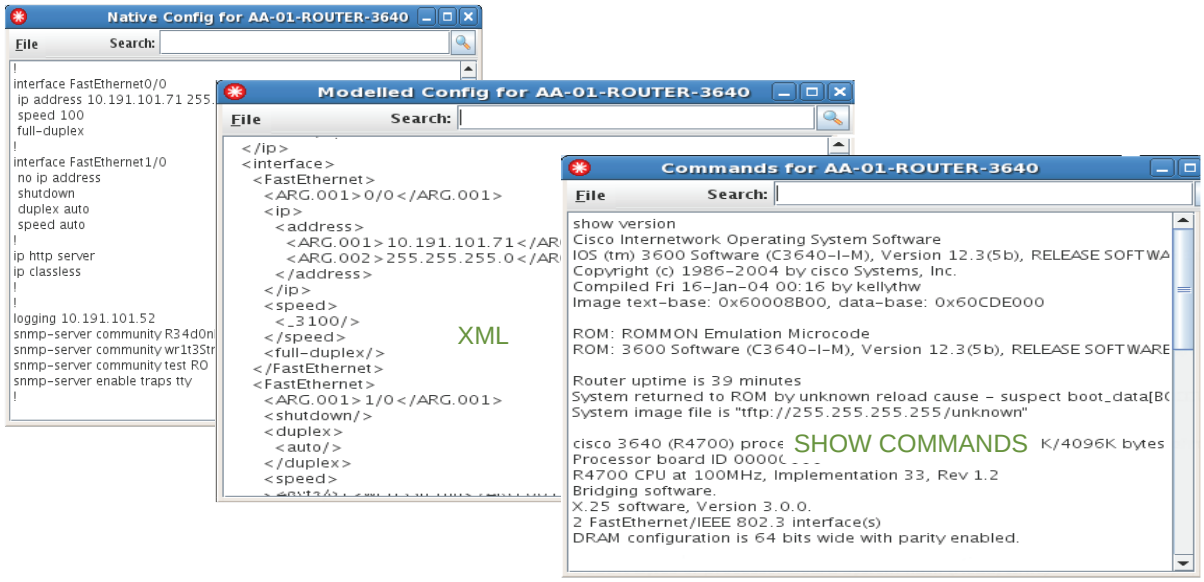
© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe regular expression use in definitions
- Describe how to use evaluations to inspect data
- Describe optional evaluations and not assessed results

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Definitions search through a BLOB of data



© Copyright IBM Corporation 2016

Definitions search through a BLOB of data

Traditionally, definitions are designed to search through a list of data that is either a configuration or data that it retrieves from the device. The advent of JavaScript objects allows this concept to expand to look at other types of data. This unit describes JavaScript later.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Most searches use regular expressions

A ***regular expression*** is a special text string for describing a search pattern in a field

Example: **\b[A-Z0-9._%-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b** can find email addresses in a BLOB of text

Introduction to regular expressions:
http://www.regular-expressions.info

A good practice is using a regular expression editor

© Copyright IBM Corporation 2016

Most searches use regular expressions

When searching through data, compliance most often uses regular expressions to match on text. A regular expression is written in a language that allows it to match on patterns in the text.

The language itself is cryptic and teaching it is beyond the scope of this class. In the slide illustrates a regular expression that finds an email address inside a block of text data. The Internet has a wealth of materials to provide introduction to and useful examples of regular expressions.

A useful tool for developing regular expressions is to acquire a regular expressions editor. These editors allow for you to develop an expression and test it immediately against test data.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Definitions use regular expressions

Any definition type can use regular expressions

Some definition types need it more than others, SmartModels® use XPath that can be augmented with regex

Often times use simple string searches, ^ntp server 10.191.101.51

More complex searches can be done, ^snmp-server community (?=.{8,})(?=.*[a-z])(?=.*[A-Z])(?=.*[\d\W])

© Copyright IBM Corporation 2016

Definitions use regular expressions

Any definition can use regular expressions. The CLI definition requires a regular expression even if it is a literal string like **no http server**. The SmartModel definition uses an XPath as its basis for searching an XML configuration and does not need to use a regular expression. However, you can use a regular expression to further hone a search to retrieve only specific nodes of the document.

In many of the security examples this training includes, the definition searches for a simple string in the configuration. The ntp server search uses a literal string. Note the carat (^) at the beginning of the string. This character is the regular expression character that starts the search at the beginning of the line. This character *prevents* a match on a string that might have a no command in front of it.

However, you can do more complex searches with more complex regular expressions. The example in the slide uses assertions against the community string to make sure that it is properly complex. The string contains the following four assertions.

- (?=.{8,}) The community string is at least 8 characters long
- (?=.*[a-z]) The community string has at least one lowercase letter
- (?=.*[A-Z]) The community string has at least one uppercase letter
- (?=.*[\d\W]) The community string has at least one digit or one special character

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Definitions search for presence or absence of data

Definitions should not determine good or bad

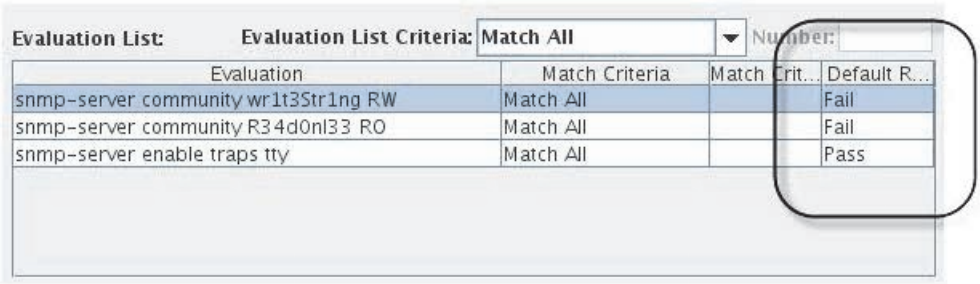
Definitions return a *true* or *false*

Definitions often use words like *pass* and *fail*

© Copyright IBM Corporation 2016

Definitions search for presence or absence of data

A key concept about a definition is that it is a neutral tool that searches for the presence or absence of data. All a definition can return is a true or a false result. Did it find what it was looking for? Due to some terminology the application uses, some confusion might occur because of words like *pass* and *fail*. In this example, two definitions return *fail* and one returns *pass*.



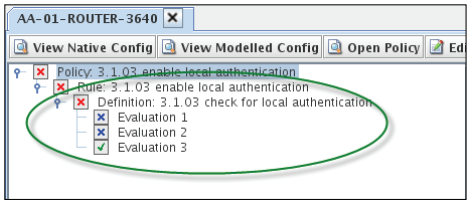
To make things easier to work with, any time words like *pass* or *fail* are seen in a definition, convert them to *true*(*pass*) or *false*(*fail*).

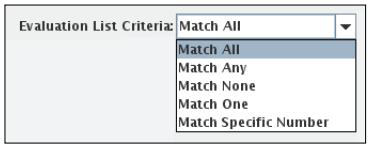
In the end, a definition does not determine the concept of good, bad, compliant, or noncompliant. Rules assign these concepts.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

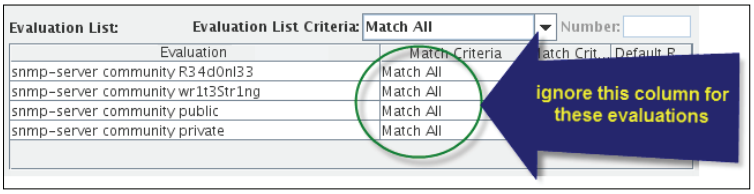
Definitions are made up of one or more evaluations

Evaluation List Criteria





Decide on the criteria to use



© Copyright IBM Corporation 2016

Definitions are made up of one or more evaluations

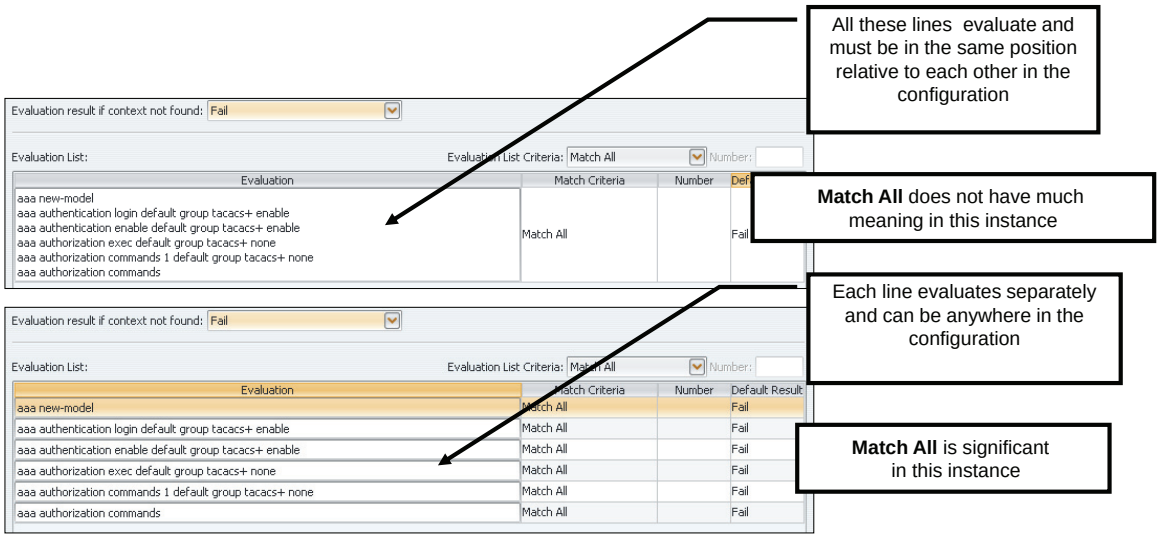
Definitions consist of one or more evaluations. These evaluations are the actual regular expressions or XPaths that interrogate the data structure. The data structure is usually a configuration. By having multiple evaluations, you must determine a criterion by which to test these evaluations. You must select one of the following options from the **Evaluation List Criteria**.

- **Match All**: All evaluations must return a match in the data structure for the definition to return *true*.
- **Match Any**: At least one evaluation must return a match in the data structure for the definition to return *true*.
- **Match None**: None of the evaluations return a match for the definition to return *true*.
- **Match One**: Only one of the evaluations returns a match for the definition to return *true*
- **Match Specific Number**: Only a specified number of evaluations can return a match for the definition to return *true*. No more and no fewer.

As shown in the slide, each evaluation has an individual match criterion. Ignore this column for now. Use when you use group parameters and extractions in the evaluation.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Evaluations can use one or more lines



© Copyright IBM Corporation 2016

Evaluations can use one or more lines

Notice the difference between these two definitions. They both check for the same AAA settings in a configuration. However, one is done as a single evaluation, while the other is done with multiple evaluations. The compliance analyzer processes these two definitions differently.

The top-most example has the entire set of commands for the AAA settings as one evaluation. This scenario means that the compliance analyzer starts one search. The configuration must have all those commands in the same exact order for the evaluation to return a match. Since it is only one search of the configuration, the concept of *match all* does not have much significance. Since this evaluation does only one search through the configuration, this scenario is an efficient search.

The bottom example has each line of the AAA settings as a separate evaluation. This scenario searches the configuration is six times for each individual string. Since the evaluation must find each string to verify the security settings, the concept of *match all* plays an important role. Using *match any* in this example causes drastic implications because the definition returns *true* when it finds only one of the evaluations. It is also apparent that this definition uses six searches through the configuration as opposed to one, which implies potential performance issues. The bottom example also allows detailed understanding of the actual line that is missing in the configuration when you review results.

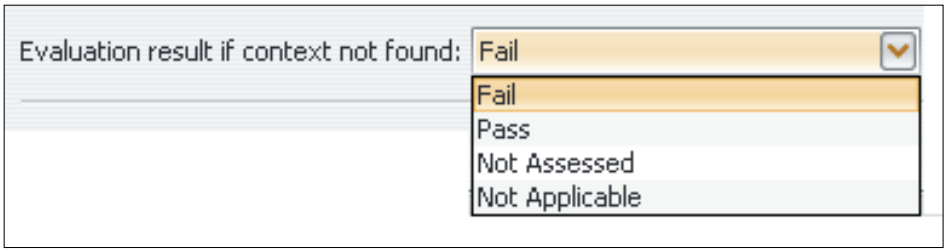
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Optional evaluations in a definition

Multiple evaluations

Evaluation result if search does not find anything

- Fail = false
- Pass = Always returns a result of true (useful?)
- Not Applicable = Optional search, neither true nor false
- Not Assessed = Used to ignore a policy



© Copyright IBM Corporation 2016

Optional evaluations in a definition

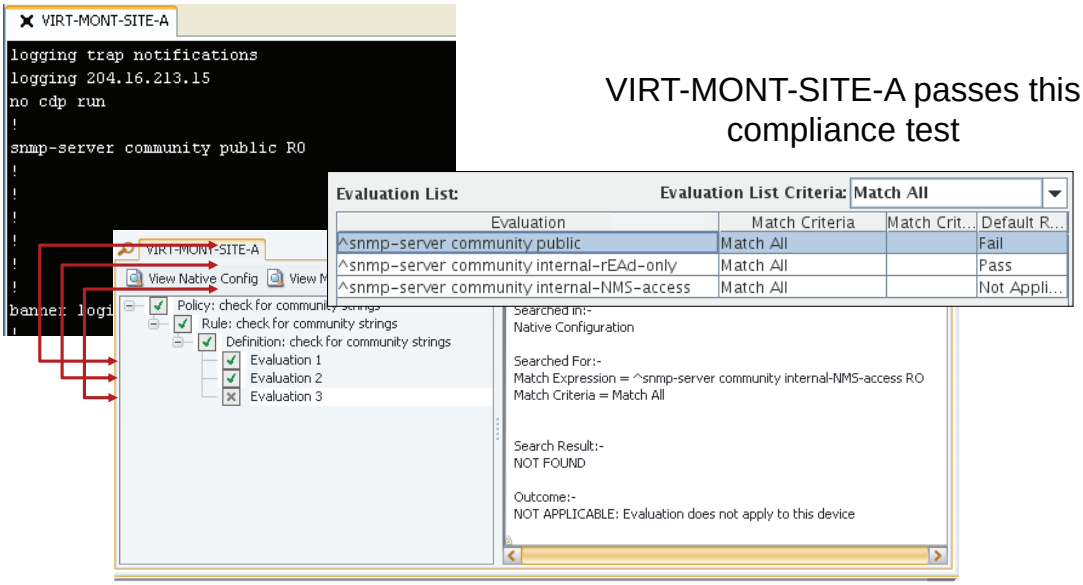
Definitions can make evaluations optional. You can use this option to treat individual evaluations differently when the search does not find them in the data. Use the option **Evaluation result if context not found** to define whether an evaluation is optional.

If an evaluation does not find what it searches for, it can respond in one of four ways.

- The evaluation returns *fail*. To avoid confusion, rename *fail* to mean *false*. *Fail* is the result that you set most evaluations to.
- The evaluation returns *pass*. To avoid confusion, rename *pass* to mean *true*. This result means that the evaluation always returns a *true* result, regardless if the evaluation matches the data or not. The next slide describes the use of this result.
- If the evaluation is truly optional, *not applicable* is a better selection than *pass*, as it tests the evaluation and provides better feedback if something is or is not found.
- The *not assessed* option is a special case where the definition-rule-policy combination does not apply to a device. An upcoming slide describes this option.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Optional evaluations in a definition example



© Copyright IBM Corporation 2016

Optional evaluations in a definition example

The slide shows an example of each of these options and how they function in a use case. The use case is the router VIRT-MONT-SITE-A. The device is configured with one **snmp-server community** string named **public**.

The definition has *match all* criteria and three evaluations that search for three different community strings of public, internal-rEAd-only, and internal-NMS-access. Note the *default responses* of fail, pass, and not applicable if it does not find the context.

Even though the router has only one community string, this definition returns *true*.

- If the search does not produce a match, the first evaluation returns fail, or false. The first evaluation finds data in the configuration. If it does not find the **public** community string, the definition returns false. Notice the green check mark for **Evaluation 1** in the slide.
- If the search does not produce a match, the second evaluation returns pass, or true. Here the evaluation looks for a community string that is not on this particular router. It does not find it and it still returns *true* and receives a green check mark for **Evaluation 2**.
- If the search does not produce a match, the third evaluation returns not applicable. Like the second evaluation, this one does not find the community string that it searches for. However, this evaluation produces the gray X as you see for **Evaluation 3** in the slide. This statement is optional because it does not affect the outcome of the definition. It is either a green check or the gray X. This result provides more information than the *pass* evaluation.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Not assessed prevents false positive results

First, check for existence of configuration data

- Allows for the policy to be ignored if data is not there
- Affects reports and compliance score

Second, check whether it is correctly configured

Evaluation List: **Evaluation List Criteria:** Match All

Evaluation	Match Criteria	Match Crit...	Default R...
policy-map POC-OUT-G7/0/7.129	Match All		Not Asse...
policy-map POC-OUT-G7/0/7.129 description PE_10000_Ethernet_PDSCP_POC_omsadmin_07:15_30/01/2008_POC class pe_mgmt_bun_output bandwidth 94 random-detect dscp-based random-detect exponential-weighting-constant 5 class class-default bandwidth 9212 random-detect dscp-based random-detect dscp 0 22 35 10 random-detect dscp 48 192 307 10	Match All		Fail

© Copyright IBM Corporation 2016

Not assessed prevents false positive results

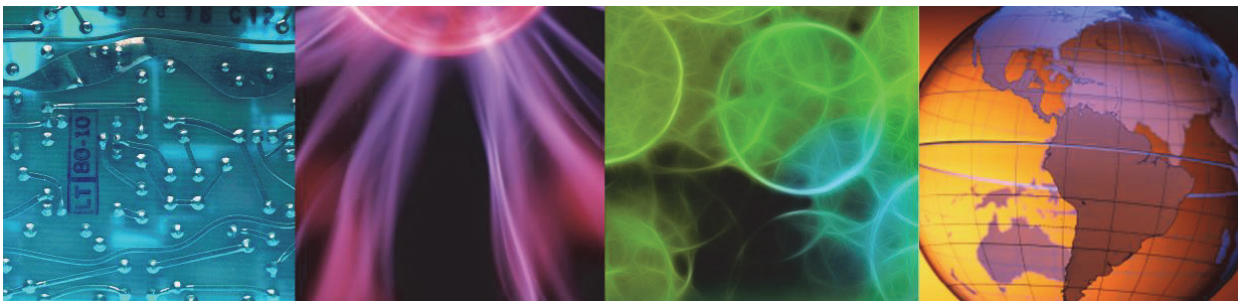
The final selection that you can set **Evaluation result if context not found** to is *not assessed*. This option has wide-ranging behavior if an evaluation does not return a match. If the evaluation is set to *not assessed* and no match is made, it defines the entire policy structure as *not assessed* for that device and the analysis continues to the next device. This function produces the same *not assessed* behavior if the VTMOS in the rule does not match that of a device.

The following use case shows why *not assessed* is useful. The network contains a policy map that manages the bandwidth usage of a customer. This policy map applies to only routers that the customer uses. If you search each router for the policy map, it is configured only a small portion of them. However, it is challenging to maintain a list of routers where this policy map is configured. By using the *not assessed* option, you can search for one line of code. If you do not find it, the policy for that device is marked as *not assessed* and the analyzer moves on to the next device. However, if you find it, use the other evaluations. These other evaluations verify that the policy map is configured correctly. If the policy map is not configured correctly, the evaluation causes the definition to return false and eventually, a noncompliant device.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 CLI definitions

Lesson 2 CLI definitions



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to configure a CLI definition.

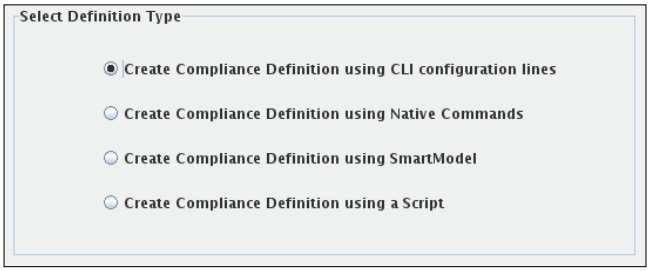
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Evaluate the raw configuration

Evaluating the current configuration from device

The raw configuration format is determined by the device

Fully reliant on regular expressions



© Copyright IBM Corporation 2016

Evaluate the raw configuration

Evaluating the raw configuration that is retrieved from the device is the easiest method for determining compliance, especially when you look for one or more base commands. Remember, a base command is a command that does not have any subcommands. Often times a base command has only one instance of itself as it turns on or off a feature of the device.

When you evaluate the configuration, you are evaluating the current configuration that is in the configuration management database. This configuration is retrieved from the device before the SmartModel converts it.

These definitions use only simple strings and regular expressions for searching.

An important aspect of raw configuration is that you can retrieve it in different formats. For example, the Juniper Junos device has a format that most operators see when interacting with the device. However, the device provides an XML version, which is useful for system processing. This XML version is the raw version that is retrieved from the device that configuration management uses. A native command definition requires the regular expression match this format and not the user view version. Refer to the student notes to see examples of these different formats.

User-friendly display of the JUNOS configuration

```
system{
   host-name itncm-m20;
   location{
      postal-code BT3 9DT;
   }
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

```
root-authentication{
   encrypted-password $1$YSgljaqA$93AE0D;
}
name-server 204.16.213.101;
name-server 204.16.213.102;
```

XML version of the JUNOS configuration that can be retrieved

```
<system>
   <host-name>itncm-m20</host-name>
   <location>
      <postal-code>BT3 9DT</postal-code>
   </location>
   <root-authentication>
      <encrypt-password>$1$YSgljaqA$93AE0D</encrypt-password>
   </root-authentication>
   <name-server>
      <name junos:key="key">204.16.213.101</name>
   </name-server>
   <name-server>
   <name junos:key="key">204.16.213.102</name>
   </name-server>
```

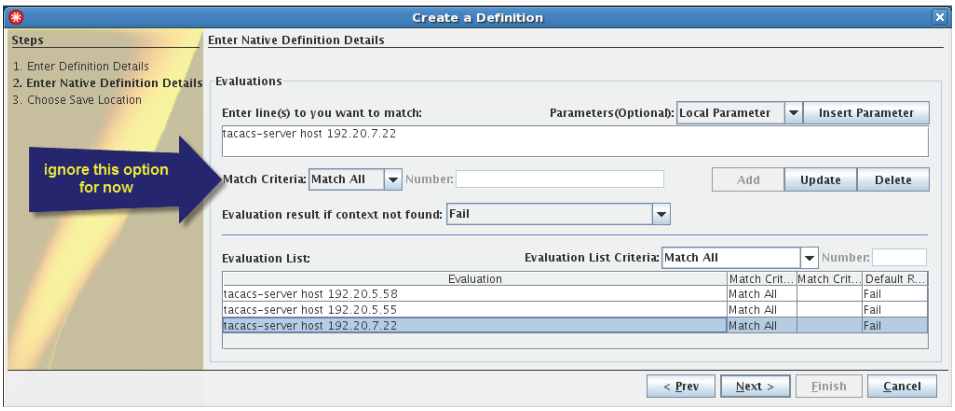
© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a CLI definition

Create the evaluation list

Define evaluation list criteria



© Copyright IBM Corporation 2016

Creating a CLI definition

Similar to the previous unit, it is fairly easy to define a CLI or native definition. You create a list of string blocks, which are typically strings only one line long. Each string is matched against the current raw configuration for the device that is evaluated. Each string returns either a *found* or *not found* response.

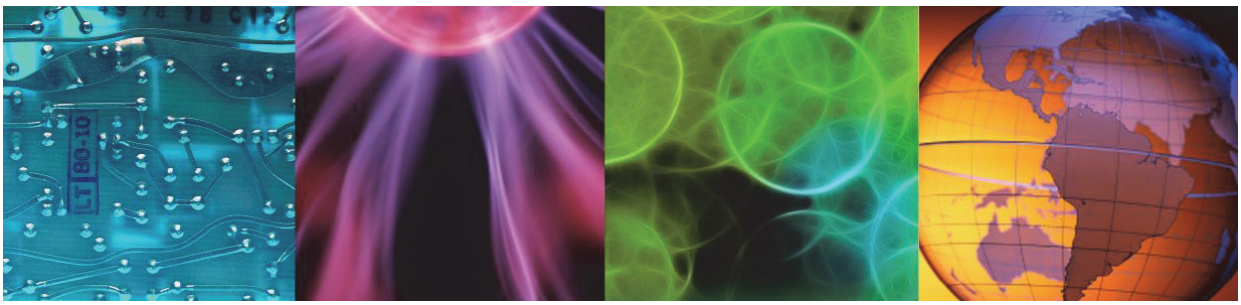
The **Evaluation List Criteria** determines what to do with all the returned *true or false* conditions.

Ignore the **Match Criteria** selection for now. This value is necessary when you use group and extraction parameters. A later slide describes this feature.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 3 Show command definitions

Lesson 3 Show command definitions



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to configure a show commands (native commands) definition.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Evaluate data that is retrieved from device

Evaluating the data that is retrieved from device command line

Fully reliant on regular expressions

Any device interaction occurs through Tivoli Netcool Configuration Manager

© Copyright IBM Corporation 2016

Evaluate data that is retrieved from device

In this scenario, you evaluate data that is retrieved from a device. You take the output of a command that is sent to the device and search through it with a regular expression for specific information. The only difference between this definition and one that searches the raw configuration is the data that it searches. They both search a blob of raw text and use the same methods to develop their searches.

Like a CLI definition, you use regular expressions to find string matches.

The *configuration manager* always retrieves the data that is evaluated. *Configuration manager* has drivers to log on to and retrieve data from the devices that are in the *compliance manager* database.

The name of this option is not helpful. Most times when you describe native commands, you reference a raw configuration. However, in this context, you are talking about the responses that you receive when you submit native commands to the device, typically **show** commands.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

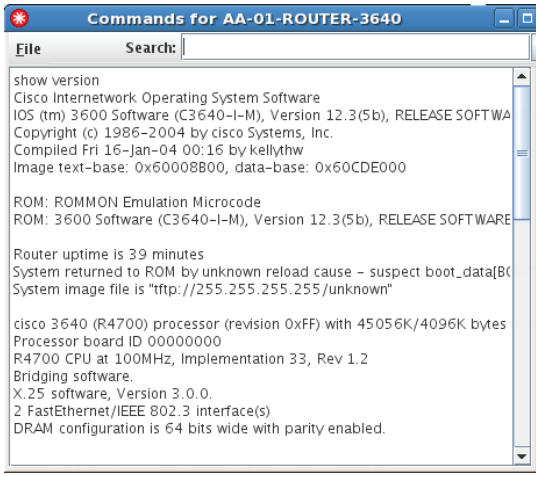
Devices already cache show data

This data is retrieved by the configuration manager during normal interactions

Typical Cisco data retrieved

- show version
- show diag
- show interface
- show ip interface brief
- show vlan
- show file system
- show snmp

Can be extended to included other commands



© Copyright IBM Corporation 2016

Devices already cache show data

Device drivers retrieve a chunk of potentially useful data whenever they import or synchronize a device. You can configure it to do this data retrieval when it submits device changes. The *configuration manager* does not use this list of data for any important activities.

The data is the result of a series of **show** commands. A typical list of commands asks for version data, interface information, and information about the SNMP setup for the device. These commands vary from vendor to vendor. Many times a device is not able to respond to a command. For example, the router might not support the command **show vlan** and you get a syntax error. In this case, the driver ignores the result of the command and it continues to the next device. A single list of data that is associated with the device in the *configuration manager* contains the results of all these commands.

You can extend the list of commands that are sent to the device to contain other commands that a customer might want to include. Adding commands to this list requires you to create a custom script resource.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Using cached commands is faster

Analyze Customer-AA devices for specific processor data

- Three devices

Show version, cached

- Analyzed devices in one second
- Simple SQL query and search

Show hardware, not cached

- Analyzed devices in 57 seconds
- Submitted and ran three units of work to gather data

© Copyright IBM Corporation 2016

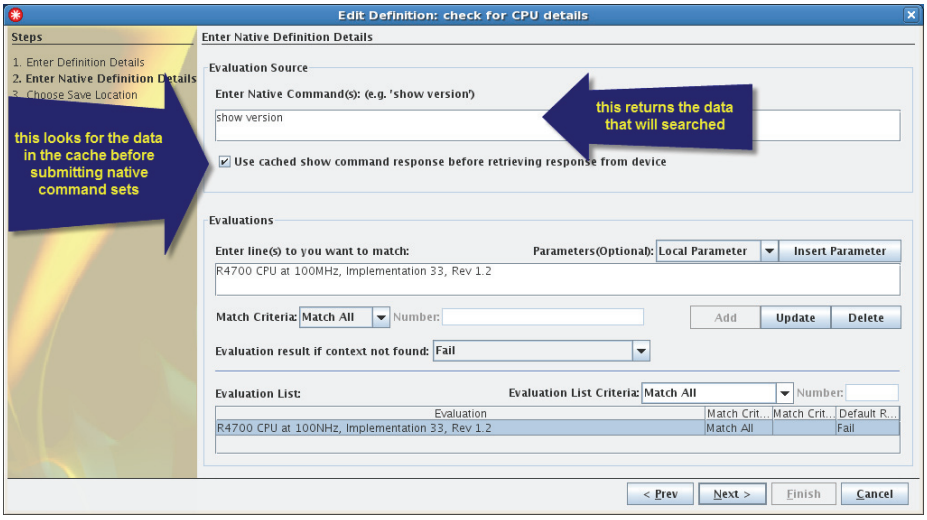
Using cached commands is faster

When using a show-commands definition, caching the blob of data in the database greatly improves the performance of the analysis. It is an acceptable option to pull live data from the network device. However, you must consider the time that it takes to log on to the device, run the command, and retrieve the results.

For example, in the training server, you can test the following by running a definition against all three devices. Running a definition that analyzes cached data uses a simple SQL query and completes the analysis in less than a second. However, if you retrieve live data from the device, this retrieval significantly increases the processing time. Three *units of work* are run on devices to retrieve the *show* data and the analysis takes almost 1 minute.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a native command definition



© Copyright IBM Corporation 2016

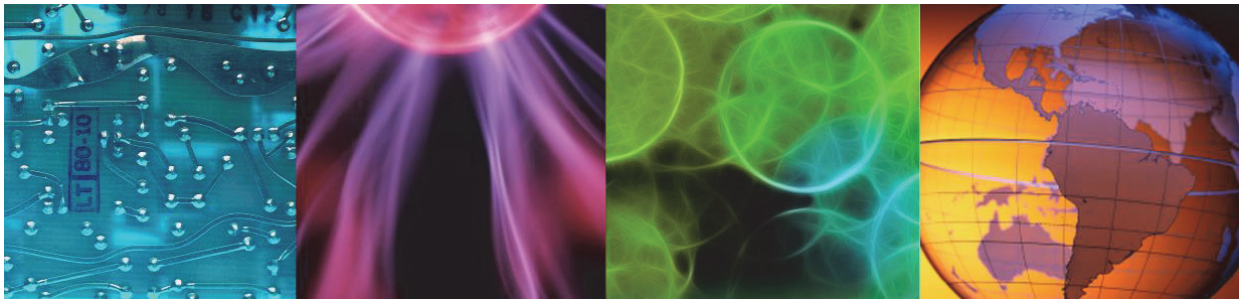
Creating a native command definition

When creating a native command or a show-command definition, you must enter the commands that retrieve the data you want to analyze, for example, **show version**. The next decision is to look for that data in the cached data that is available in the *configuration manager*. When you select this option, compliance searches the cached data for the **show version** command. If it finds the command, it uses the cached data. However, if compliance does not find it, it creates a UOW and it retrieves the results of the **show** command directly from the device.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 4 SmartModel definitions

Lesson 4 SmartModel definitions



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe searching an XML document
- Configure a SmartModel definition
- Configure optional data items to show in the GUI

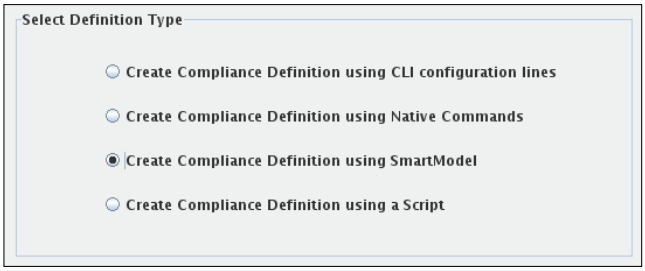
 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Evaluate the modeled configuration

Evaluating data in the XML configuration

Fully reliant on Xpath, XPath is a query language for selecting nodes in an XML document

Regular expressions can also be used extend XPath



© Copyright IBM Corporation 2016

Evaluate the modeled configuration

The SmartModeled definition is a powerful way to search through configurations. In this case, instead of searching through a blob of raw text, you search through a structured XML document. All of the data in the configuration is tagged, making it easy to find specific items. As mentioned earlier, modeled searches allow the search of subcommands. The underlying XML that models the configuration understands the concept of context. A context is where it selects one or more nodes of data from the configuration and then further processes each individual node.

Searching a device model requires an understanding of XPath. XPath is an industry-standard method for searching and retrieving data from an XML document. To build a device model definition, you must be familiar with the XPath syntax.

The device modeled definition can also use regular expressions to help filter the results that XPath retrieves from the modeled configuration.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

SmartModel definitions search XML

XPath returns a set of zero to N nodes
- Definition can further examine nodes for other data

Much easier to evaluate unbounded commands

Bounded, has zero to one instance of a command
- hostname, service password encryption, aaa new-model

Unbounded, has zero to many instances of a command
- interface FastEthernet, tacacs-server host, ip access-list
- Often times commands have subcommands, interfaces

© Copyright IBM Corporation 2016

SmartModel definitions search XML

When you search through a modeled configuration, you use XPath to retrieve a set of nodes that match the criteria. The XPath returns **0** to **n** nodes of data, depending on what it is searching. By retrieving a series of nodes from the document, this feature makes it much easier to evaluate an unbounded command.

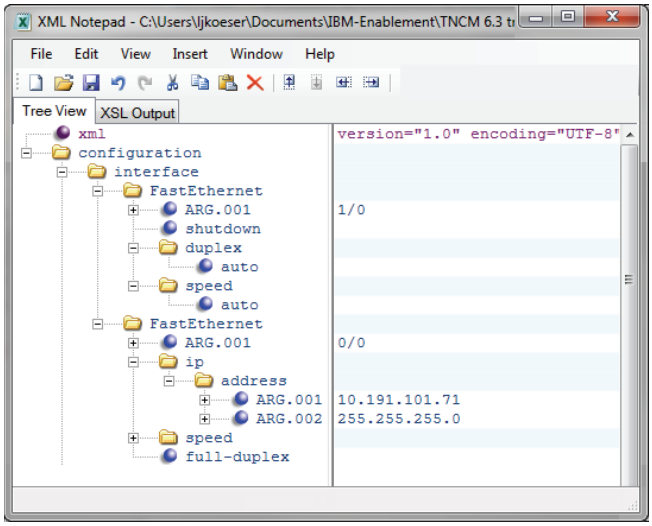
A bounded command is a command that you find a single instance of in a configuration. A bounded command defines a unique attribute for the device, like its host name, or enables and disables a global service for the device, for example, password encryption or security protocols.

An unbounded command is where you can have a whole set of instances of one particular command. For example, devices usually have multiple interfaces. The **interface** command starts each interface. That command is followed by the type of interface, a numbering scheme, and eventually commands that configure that specific interface. Another place that you commonly find unbounded commands is in definitions of lists. Access lists are lists of permit and deny statements that can be long. Other lists of data include multiple authentication servers, community strings, and others. Unbounded commands often have subcommands associated with them.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

UNIX-like syntax for finding nodes

/configuration/interface/FastEthernet



© Copyright IBM Corporation 2016

UNIX-like syntax for finding nodes

After a person is familiar with the XPath syntax, it is easy to equate it to the UNIX syntax for finding directories in a file system. An XML document is hierarchical like a file system. An XPath of **/configuration/interface/FastEthernet** retrieves all the FastEthernet nodes from a configuration, regardless of whether 1 or 100 interfaces exist.

Similar to a UNIX path for finding a directory, you need to understand the document structure to be able to retrieve things from it. The software provides tools to help you in this effort.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

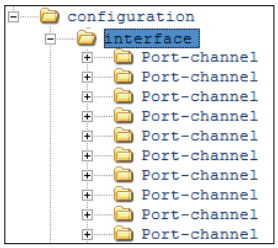
Modeled definitions allow contextual XPath

Similar to creating two searches

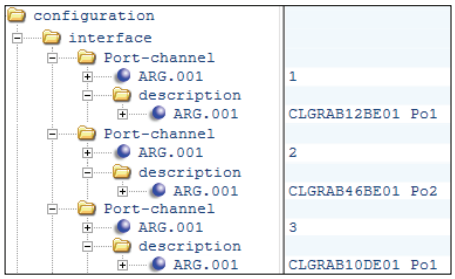
The first search returns a set of nodes from the document

The second search analyzes a subcommand in each node

configuration/interface/Port-channel



description



© Copyright IBM Corporation 2016

Modeled definitions allow contextual XPath

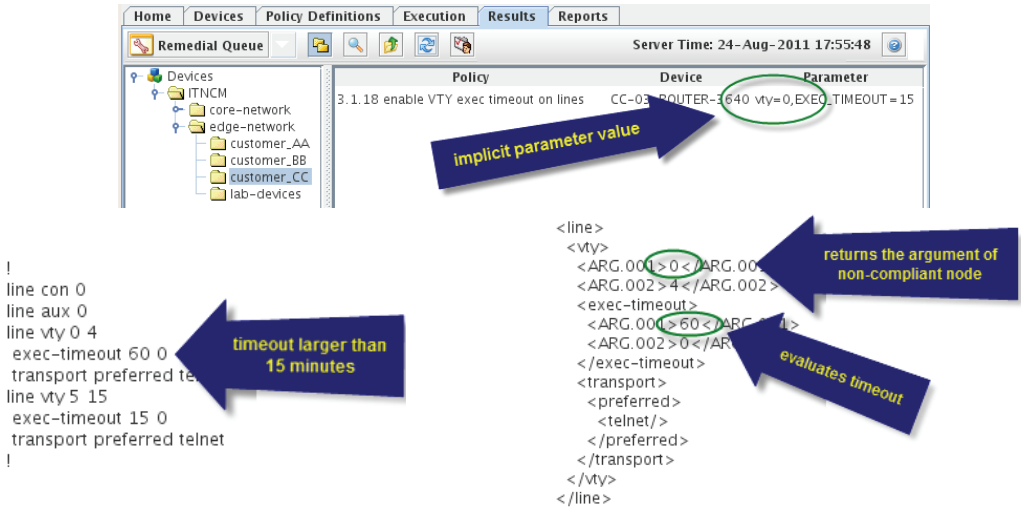
One of the more powerful aspects of working with the modeled version of the configuration is using the concept of a context. Contexts are useful when you work with configurations that use subcommands. When you create a contextual XPath, it is like you create two searches. The first search retrieves a set of nodes from the document that meet a specific criterion. This list of nodes are considered the contexts. The second search then inspects each individual context for something specific and determines whether it is successful or not.

In this example, an XPath search exists that retrieves all the port-channel interfaces from a configuration. This search creates a list of contexts. The second search then goes into each port-channel interface and determines whether a description node exists. You can further enhance this second search to include a regular expression to verify that the descriptions meet a specific format.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Modeled definitions provide additional data

Model definitions return specifics about non-compliant nodes



© Copyright IBM Corporation 2016

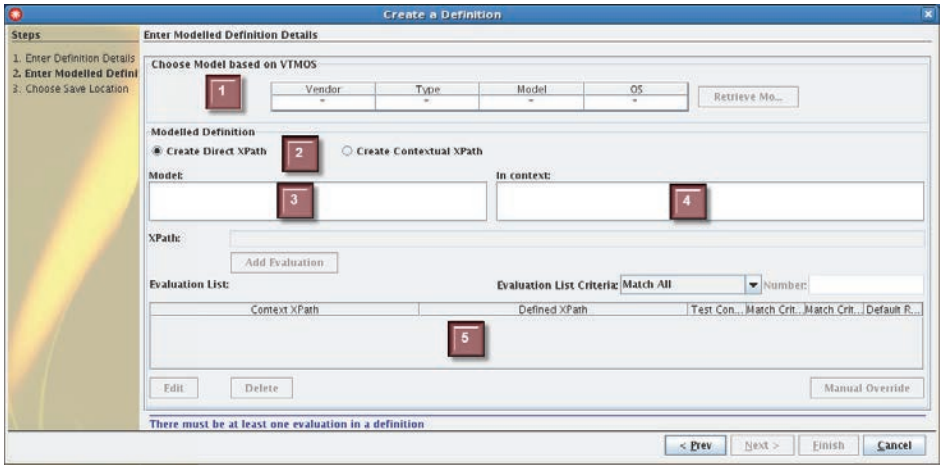
Modeled definitions provide additional data

What is important to learn about the concepts of a modeled definition and contexts, is that the software returns information about the context that fails the search criteria. Specifically, the product returns the identifying value that makes one context different from another. For example, it returns an interface number or a VTY argument.

In this example, you are ensuring the configured VTY sessions timeout after a specific period. The period in this example is 15 minutes. The raw configuration in the lower left corner of the slide shows a configuration with two VTY terminals. The first **VTY 0 4** is configured to time out after 60 minutes where the second **VTY 5 15** timeout is 15 minutes. Per the policy of a timeout of 15 minutes, the **VTY 0 4** terminal is configured incorrectly. By using a modeled definition to analyze this configuration with a context, the software returns the argument that makes **VTY 0 4** unique. In this example, the argument is the value **0** in **VTY 0 4**. By knowing this value, you know which instance of VTY needs modification. You can include that value of 0 in a command set to modify just that portion of the configuration. The vty=0 noted that in the GUI snapshot in the slide is an implicit parameter, use it to target the command set execution.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating a modeled definition



© Copyright IBM Corporation 2016

Creating a modeled definition

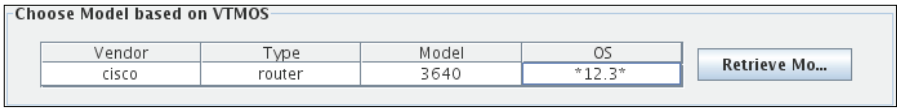
Creating a modeled definition is a more involved process than you saw in the earlier definition types. The following list describes the main items in the GUI.

1. To build a modeled definition, you must select a VTMOS so you can select and use a schema to help build the XPath.
2. The type of command you are inspecting determines whether you choose a direct or contextual XPath.
3. You need to select the command that you use for the initial search.
4. If the command of interest is a subcommand of an unbounded command, you need to create a context.
5. This area shows the set of evaluations that are part of this definition.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Select a SmartModel to build XPath

Selecting a SmartModel helps build definition



This configuration retrieves the VTMOS schema that has command structure

Same limitations that come with building command sets

© Copyright IBM Corporation 2016

Select a SmartModel to build XPath

The first step in creating a modeled definition is to select the SmartModel that best represents the devices that you are analyzing. You must select a representative VTMOS. The VTMOS selection pulls from the drivers database the schema or command structure of that device.

Use this command structure to help build an XPath for the command you are interested in searching for in the XML configuration.

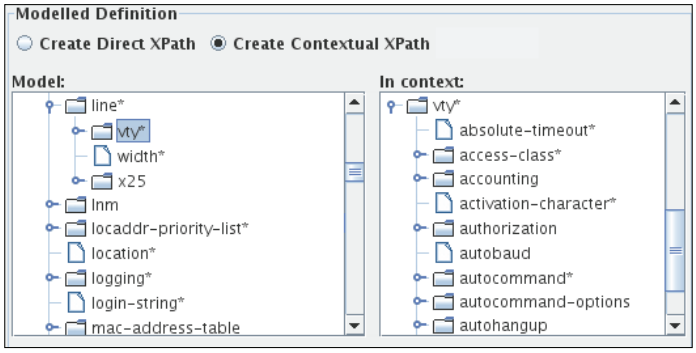
Selecting the VTMOS for the device schema has the same limitations that are associated with command sets. For example, if you select the newest schema it usually has commands that the older operating systems do not include.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Select commands that are retrieved and searched

Direct Xpath, bounded commands

Contextual Xpath, unbounded commands with subcommands



© Copyright IBM Corporation 2016

Select commands that are retrieved and searched

After you select a schema and load the model, a command structure shows in the window. This structure follows the same alphabetical layout that you see in the configuration editor in the *configuration manager*.

An important decision that you must currently make is to determine whether you are selecting a direct XPath or want to create a contextual one.

A direct XPath in most cases aligns with bounded commands where only one instance exists in a configuration.

Note: Note, if the entity is not unique, or there is more than one node, a direct XPath evaluates only the first occurrence of the entity that it searches for.

A contextual XPath mostly aligns with command structures that have subcommands like the interfaces and access lists.

Refer to the student notes for examples of bounded and unbounded commands.

Examples of bounded commands for a Cisco router:

```
aaa new-model
service password-encryption
no ip http-server
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Examples of bounded commands for a Cisco router.

```
service password-encryption
hostname router-3
no logging console
enable password 7 124A0B4310075F
aaa new-model
no ip http server
interface FastEthernet0/0
 ip address 10.191.101.72 255.255.255.0
 description connection to the internet
 speed 100
 full-duplex
interface FastEthernet1/0
 no ip address
 description currently not in use
 ip directed-broadcast
 duplex auto
 speed auto
interface FastEthernet2/0
 no ip address
 description currently not in use
 shutdown
 duplex auto
 speed auto
ip access-list extended management-servers
 remark TNCM servers
 permit tcp host 172.200.201.51 any
 permit tcp host 172.200.201.52 any
 permit tcp host 172.200.201.53 anyy
snmp-server community T3stM3 RO
snmp-server community Iw1llW0rk RW
line vty 0 4
 exec-timeout 15 0
 transport input all
 logging synchronous
line vty 5 15
 exec-timeout 20 0
```

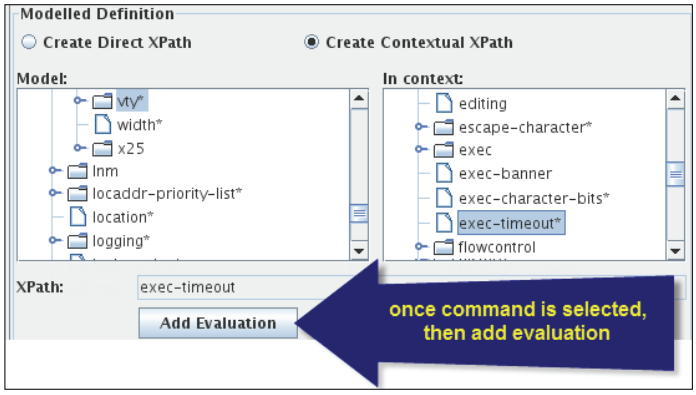
© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Select the command to analyze

Regardless of direct or contextual XPath

Add Evaluation opens another wizard



© Copyright IBM Corporation 2016

Select the command to analyze

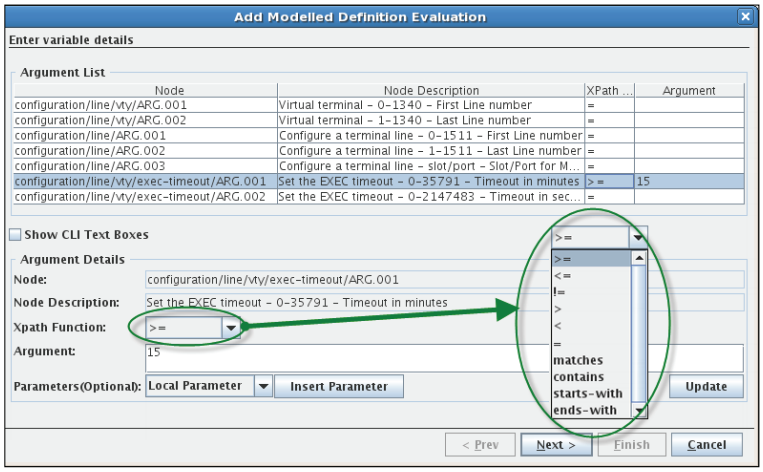
In the slide example, the definition looks at the **exec-timeout** subcommand that is associated with the parent command-line **vty** and so it is using a context. Regardless of whether you choose a direct XPath or a contextual one, you eventually must select **Add Evaluation**. This selection opens a window that addresses details you can set for the XPath.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Set the analysis on the nodes of interest

Can place criteria on multiple nodes

Use Update to confirm changes



© Copyright IBM Corporation 2016

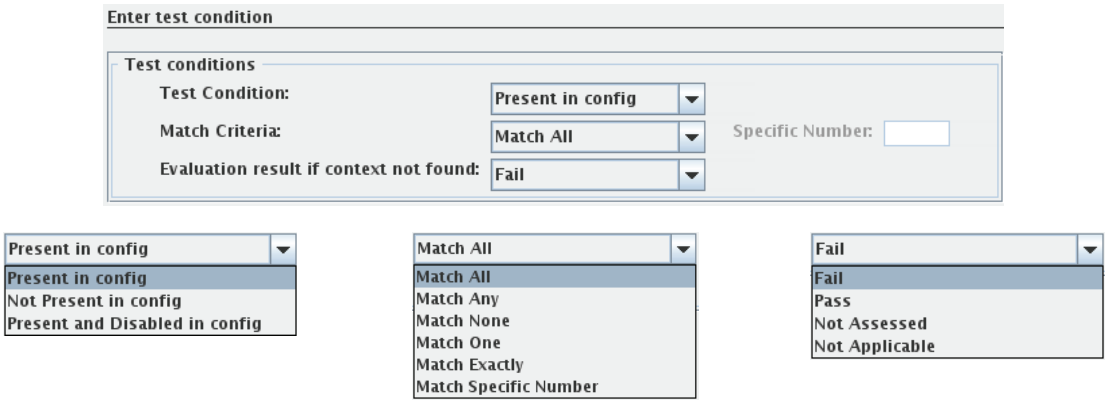
Set the analysis on the nodes of interest

In the previous slide, you create a contextual XPath that addresses the **line vty** command structure.

You use the GUI that opens to set criteria on the entire XPath that leads from the **exec-timeout** command all the way to the **line** base command. It is possible to select only one specific **line** and check for the **exec-timeout** if necessary. Often, you want to inspect all instances of **line vty** to verify that they are configured with **exec-timeout**. You also ensure that the **ARG.001** is equal to 15. Looking at the slide, you can see that **ARG.001** is described as the timeout value in minutes. Here you set the XPath function to **=**. Depending on the argument, you inspect, other functions might be more appropriate.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Select the test condition for XPath



© Copyright IBM Corporation 2016

Select the test condition for XPath

After you set criteria on the XPath arguments, you can then determine the test conditions on the XPath search.

- **Test Condition**: This selection asks should the XPath return a result or not. In most cases, you search for the existence of commands and so **Present in config** is the correct selection. It is in the higher-level rule that you determine whether finding the command is a good or a bad thing.

  Another important issue is that a negative like *no* can precede commands in the configuration. For example, **no logging console**. **Disabled** replaces the **no** command in the XML. For commands preceded by the word **no**, you select **Present and Disabled in config**.

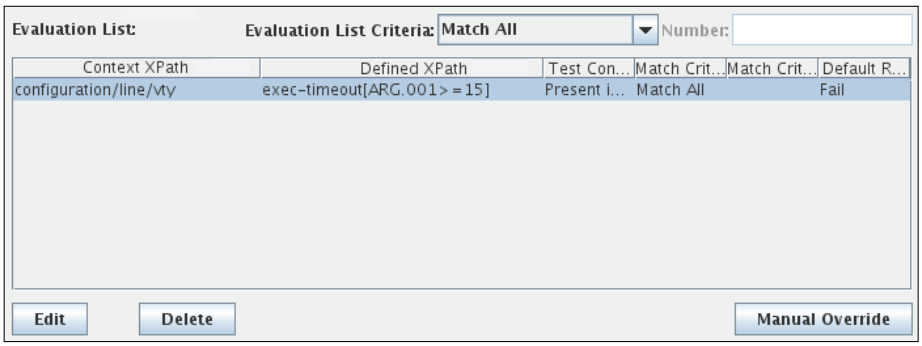
- **Match Criteria**: This criteria is similar to what you use in a CLI definition. However, with the concept of a context, an XPath search can return 0 to n results. If the search finds multiple contexts, you need to understand the effects of selecting *match all*, *none*, *some*, and others.

- **Evaluation result if context not found**: As with a CLI definition, these selections determine whether the definition returns *true*, *false*, or *not assessed*. The same rules apply here as with the CLI definitions.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Completed evaluation

Determine the Evaluation List Criteria for multiple evaluations



© Copyright IBM Corporation 2016

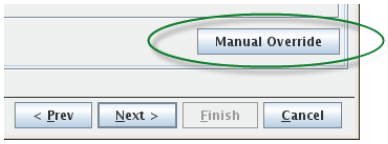
Completed evaluation

The Evaluation List Criteria selection is only useful when there is more than one evaluation that the Evaluation List defines.

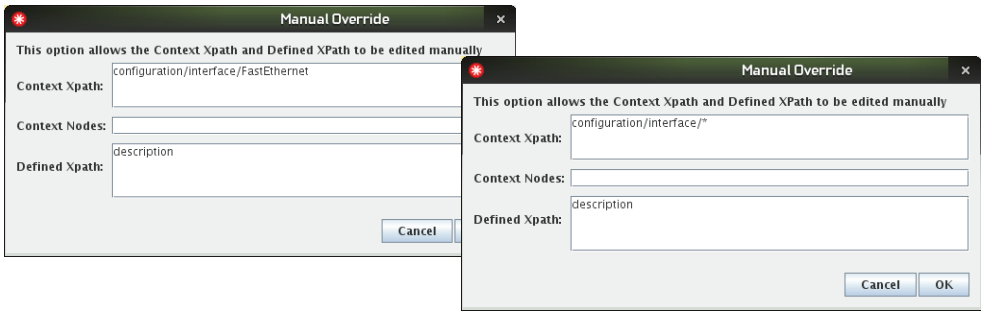
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Manual override

Override the XPath that is generated by the GUI



Example: Ensure that all interface types have descriptions



© Copyright IBM Corporation 2016

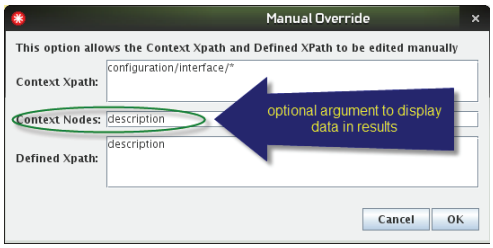
Manual override

Rare occasions exist where you might choose to override the XPath that the GUI generates. If for example, your policy is to ensure that all interfaces have a description. This particular policy is difficult to implement with raw text. However, using an XML context, it is easy to find all FastEthernet interfaces in a configuration and check each one for a description. Unfortunately, the GUI forces you to select a specific interface type, for example, FastEthernet or serial. Using the manual override button for an evaluation, you can manually edit the XPaths that are generated. In this example, you can replace **FastEthernet** with the character ***** and all interfaces, regardless of type, are found in the configuration and are checked for descriptions.

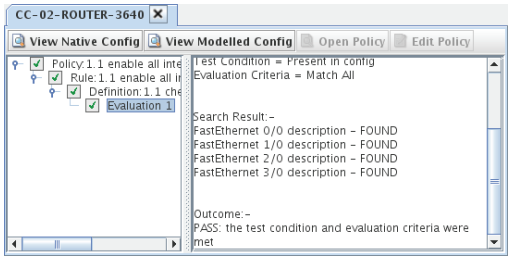
Important: You must test the manual override option and understand the schema and command structure. A problem with the previous example is that not all interface types might support the **description** command. In that case, you might generate false positives for interfaces that you cannot configure with descriptions.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

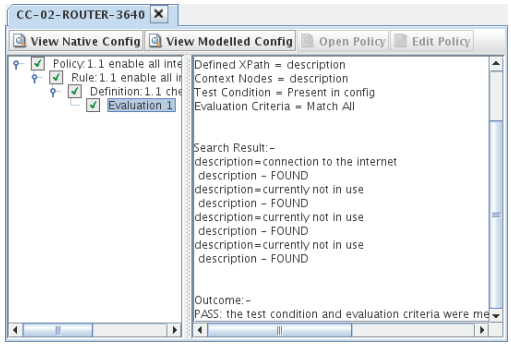
Show optional data about context



Without context nodes



With context nodes



© Copyright IBM Corporation 2016

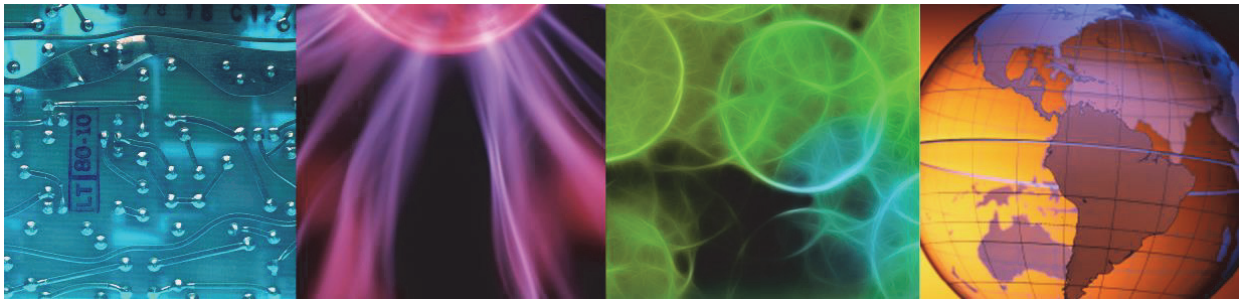
Show optional data about context

The manual override process allows the addition of context nodes, which show data about the node in the detailed results window. The context node is the name of the actual node in the XML context that is retrieved from the main configuration document. This optional field does not affect the evaluation in any way but does provide data in the log that might be useful for users.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 5 Using parameters in definitions

Lesson 5 Using parameters in definitions



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

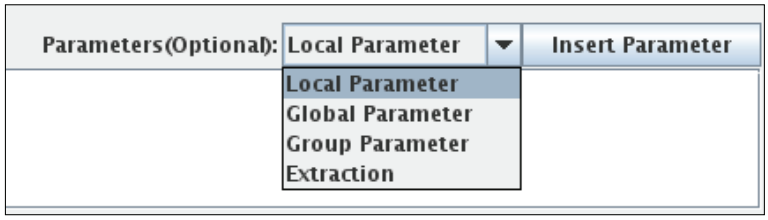
In this lesson, you learn how to use parameters in definitions.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Parameters make definitions reusable

Use parameters in one or more evaluations

Different networks use different values



© Copyright IBM Corporation 2016

Parameters make definitions reusable

In this lesson, you modify the definitions to include parameters. Parameters allow the following capabilities:

- You can make policies generic. At execution time, define the parameters in the definitions and make the policies specific. For example, one part of the network has specific read-only community string while another part has a different one.
- You use parameters to test multiple values by using a group parameter or extraction. By using these types of group values, you verify that a list of values are configured on a device.

Use parameters exclusively in the building of evaluations for a definition.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Parameter types

Local parameter
- Single value with an optional default
- Defined for a process execution

Global parameter
- Single value
- Defined at a global level for a policy realm*

Group parameter
- List of one to many values
- Defined at a global level for a policy realm*

Extraction
- List of zero to many values
- Values that are extracted from a data source

Scripted
- Uses JavaScript to determine the value of the parameter
- Defined at a global level for a policy realm*

* Policy realms have a security role in this context

© Copyright IBM Corporation 2016

Parameter types

Four types of parameters exist that you can insert into an evaluation definition.

- **Local Parameter**: You can place a local parameter in an evaluation. At execution time, the search of the data structure uses a value that replaces your parameter. Local parameters are only valid for the process execution. If a policy uses a definition with a local parameter, each execution of the policy can have a different value for that parameter. You can assign a default value to a local parameter.
- **Global Parameter**: A global parameter is a uniquely named parameter that is set for the system and applies to a policy realm. It represents one value. After you define a global parameter, anywhere else you use that parameter name, it has the same value. For example, if a TACACS server is defined for an entire implementation, you can define a global parameter and use it in multiple policies that associate with TACACS compliance evaluations.
- **Group Parameter**: A group parameter is a special type of global parameter that defines a list of one or more values. When you associate a group of parameters with an evaluation, that evaluation runs for each of the values that are in the group parameter list.
- **Extraction**: An extraction is not technically a parameter as the other three are. You cannot define a value for an extraction. Use an extraction with a regular expression or XPath to search for values in a list of data. An extraction retrieves one or more values from the data structure. The extraction then uses those values in an evaluation like a group parameter does.
- **Scripted Parameter**: A scripted parameter is one or more values that returns from the execution of a JavaScript that has access to internal and external data and processing resources.

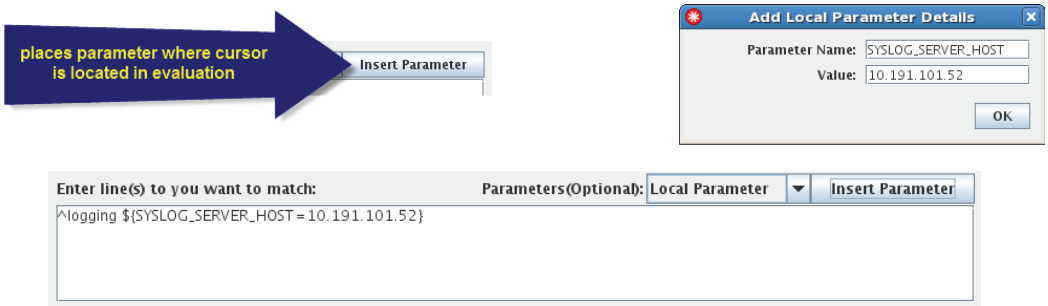
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Local parameters

Values defined or overwritten when creating a process

No limit to local parameters in an evaluation

Can change between process executions



© Copyright IBM Corporation 2016

Local parameters

A local parameter is easy to define. You can place it in any type of definition. When you insert it into the definition, you must define a parameter name and provide a default value. After you create it, you can see the syntax that it creates.

When you add this definition to a rule and policy combination, the wizard prompts you to view the parameter values and you can change the default. The value that you select is part of the process until you manually change it by editing the process or use the **Parameter Administration** tool.

Since a local parameter defines a single value, you can have as many as you need in one evaluation.

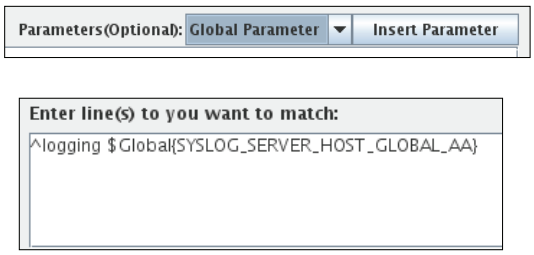
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

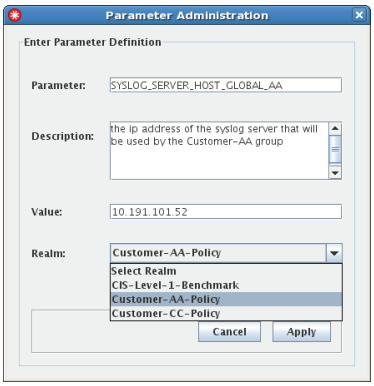
Global parameter

Parameters are associated to policy realms for security reasons

All references to the parameter in use the same value

No limit to global parameters in an evaluation





© Copyright IBM Corporation 2016

Global parameter

A global parameter is a parameter that you define for a policy realm. The intent of defining global parameters for specific policy realms is to support managed services providers. If you assign a customer to a policy realm, the customer has access only to the global parameters associated with that realm. When you create the parameter, you assign it to a policy realm that you are authorized to view.

Give all global parameters unique names. All references to the global parameter use the same value that you define when you create it and then manage it with the parameter administration tool.

As with local parameters, you can define as many as you need in a single evaluation.

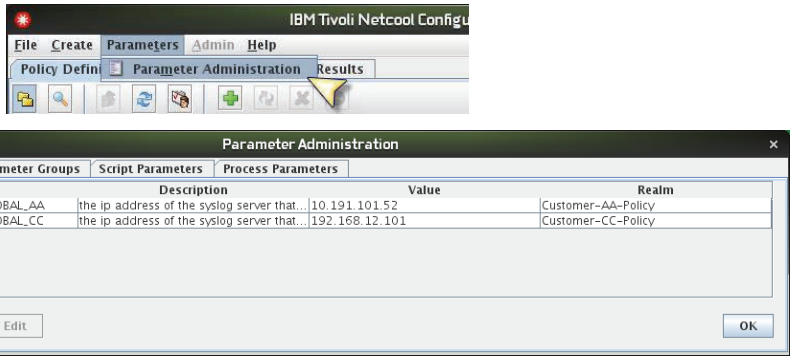
© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Creating global parameters

Global parameters are managed at an application level

Need to create them using a GUI

Create them on-the-fly or ahead of time



© Copyright IBM Corporation 2016

Creating global parameters

Manage global parameters with the Parameter Administration GUI. Select it using the **Parameter > Parameter Administration** option. Using this GUI, you can create, edit, or delete global parameters. You can choose to create parameters with this administration tool or you can create them when working with the definition. Regardless, when you create a new one, a GUI is started that requires a parameter name, description, value, and a policy realm to associate the parameter with.

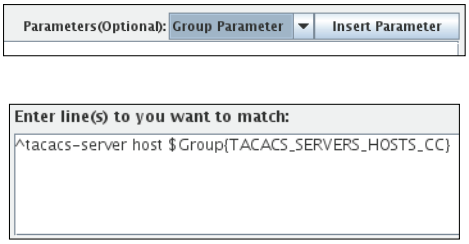
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

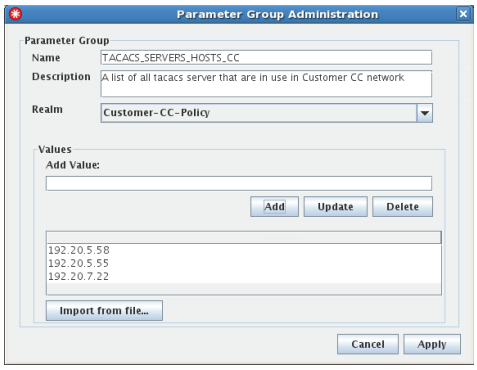
Group parameter

Associated to a policy realm like a global parameter

A list of user-defined values to test against for an evaluation

Only one group parameter is allowed per evaluation





© Copyright IBM Corporation 2016

Group parameter

A group parameter is similar to a global parameter. You associate it with a policy realm for security reasons. A group parameter differs from a global parameter in that it is a list of values. When an evaluation includes a group parameter, every item in the group parameter list of values is evaluated. The list of **tacacs-server host** commands that are shown in the slide is one example. A configuration can point to multiple TACACS servers for authentication fallback scenarios. The evaluation **^tacacs-server host $Group{TACACS_SERVER_HOSTS_CC}** evaluates three times: once for each IP address in the list of values.

It is not possible to have more than one group parameter per evaluation because that would cause iterating between each set of values.

When you create a group parameter, you are presented a GUI similar to a global parameter where you must provide a name, description, and a policy realm. You must also provide a list of values that the evaluation uses sequentially. You can import this list from the client system, if available. Otherwise, you can add, modify, and delete values individually.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

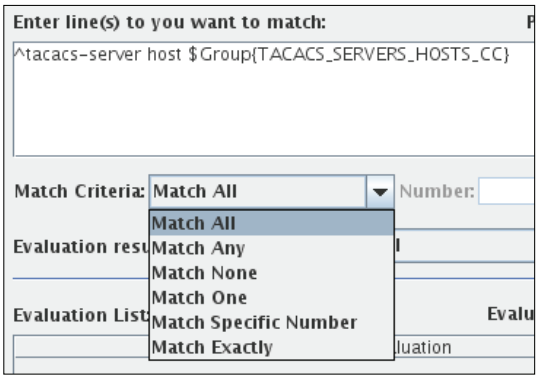
Group parameters use evaluation criteria

Implied are multiple values in a group

After searching through the data with each value, what match criterion should be used

- Match All
- Match Any
- Match None
- Match One
- Match Specific Number
- Match Exactly

Evaluation returns *true* or *false*



© Copyright IBM Corporation 2016

Group parameters use evaluation criteria

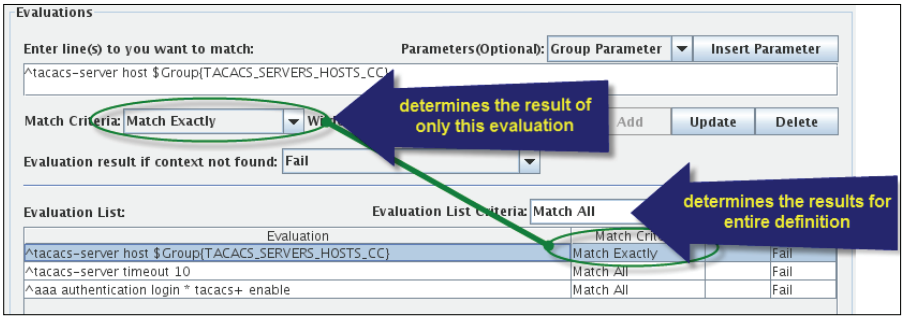
Up to this point in the training, a match criterion exists that is not described. The **Match Criteria** option is set for evaluations that use multi-value parameter sets, group parameters, and extractions. When you use a group parameter in an evaluation and it has more than one value, you must select the match criterion. If you use the **^tacacs-server host $Group{TACACS_SERVER_HOSTS_CC}** example, you need to decide whether to select *match all*, *any*, *one*, *specific number*, or *exactly* when you inspect the configuration for all three commands. Depending on the match criterion, the evaluation returns either *true* or *false*.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Difference between matching criteria

Match Criteria is determining the results of only one evaluation

Evaluation List Criteria is determining the results of the definition as a whole



© Copyright IBM Corporation 2016

Difference between matching criteria

The **Match Criteria** field is only effective on the evaluation that you apply it to. It produces a *true* or *false* only for that evaluation. The **Evaluation List Criteria** field determines the results for the entire definition. Even if a match criterion produces a *false* result for a specific evaluation, that does not necessarily mean that the definition returns a false, especially if you define more than one evaluation.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Match exactly is a useful tool

Checking for two and only two community strings

Any other strings cause definition to return false

TRUE

```
Search result-
Group Parameter Value(s)
R34d0nl33
wr1t3Str1ng
snmp-server community R34d0nl33 – FOUND
snmp-server community wr1t3Str1ng – FOUND

Outcome
True
```

FALSE

```
Search result-
Group Parameter Value(s)
R34d0nl33
wr1t3Str1ng
snmp-server community R34d0nl33 – NOT FOUND
snmp-server community wr1t3Str1ng – FOUND

Outcome
False
```

```
Search result-
Group Parameter Value(s)
R34d0nl33
wr1t3Str1ng
snmp-server community R34d0nl33 – FOUND
snmp-server community wr1t3Str1ng – FOUND
snmp-server community test – Not in list

Outcome
False
```

© Copyright IBM Corporation 2016

Match exactly is a useful tool

One of the match criteria that you can use for a group parameter and its associated evaluation is *match exactly*. In this case, the evaluation checks for a specific number of results and there can be no more and no fewer. This example is checking for two specific community strings: R34d0nl33 and wr1t3Str1ng.

You must find only these two strings for the evaluation to result in a true when it uses *match exactly*. Notice how the *false* results in the slide determine that either one of the strings is missing or that it found an extra string.

Use the *match exactly* result to apply an extra command set during a remediation.

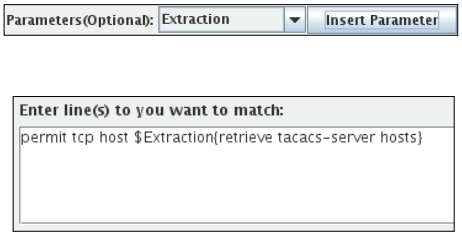
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

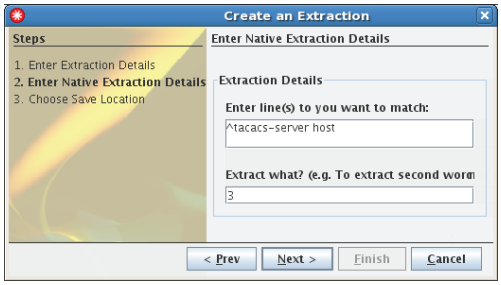
Extractions

Retrieves one or more values from device data and uses them in evaluation

Only one extraction is allowed per evaluation

Use **Match Criteria** similar to what group parameters use





© Copyright IBM Corporation 2016

Extractions

The next parameter is not like the previous examples. Extractions do not define user-coded values. An extraction is a tool that searches for a set of one or more values in a data structure. An extraction uses the retrieved values as input into an evaluation that searches the same or a different data structure.

Only one extraction per evaluation is allowed since it might have multiple values.

As with group parameters, the results of the extraction evaluations must have a *matching criterion* set to determine whether the evaluation should return *true* or *false*. The default is *match all*.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Extraction object in GUI

Like creating a remedial or email action

Select a data source to search

Define search criteria

Determine which piece of data to extract from search results

© Copyright IBM Corporation 2016

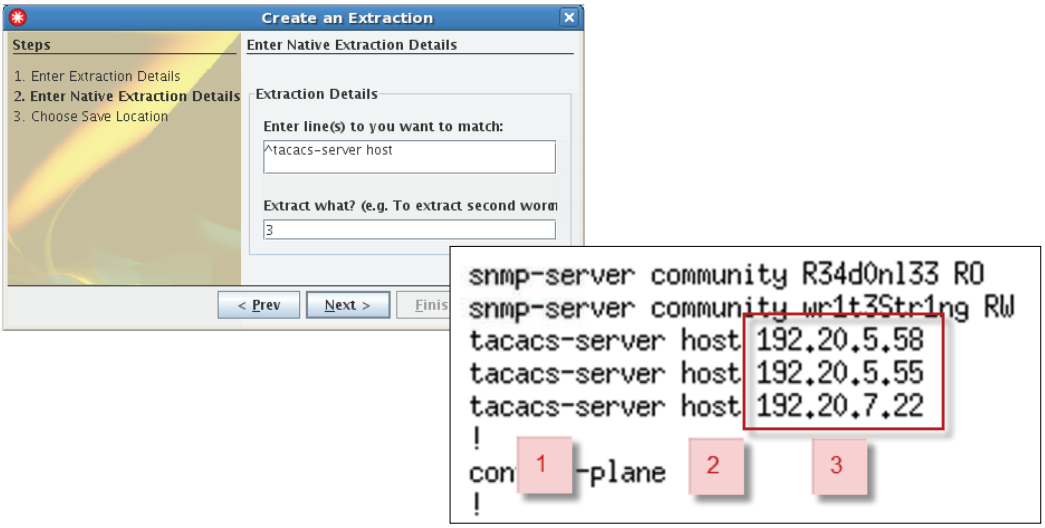
Extraction object in GUI

To use an extraction in an evaluation, you must create an extraction object. An extraction object is built the same way that you create a remedial or email action. Click **Create > Extraction** to start the extraction wizard.

The wizard prompts you for the data structure you want to search. As with definitions, you can search the raw config, the modeled config, or retrieve data directly from the device. Selecting the raw config or device data options assumes that you are searching raw text. It prompts you for search criteria to select specific lines of text from the data blob. After you select the lines of data you are interested in, you indicate to the wizard the location of the significant data.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Retrieve TACACS server example



© Copyright IBM Corporation 2016

Retrieve TACACS server example

The slide shows an example extraction. This example is extracting the IP addresses of the TACACS servers that a router uses for authentication. You want to know these addresses so you can verify that they are included in the access list that device management uses.

The first step is to enter the line or lines you want to match against. In this case, you are searching for the command **^tacacs-server host**. When you search through the configuration for those commands, you receive from 0 to many results. In this case, you retrieve three lines in the configuration. You now must extract from those results the data you need. In this case, the significant data is in the third position. In this example, position 1 = tacacs-server, position 2 = host, and position 3 = IP address. By placing 3 in the **Extract what** field, the extraction returns the IP addresses of the retrieved tacacs-server commands.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Example of an extraction in an evaluation example

1. Extraction retrieves these values

```
snmp-server community R34d0nl33 RO
snmp-server community wr1t3Str1ng RW
tacacs-server host 192.20.5.58
tacacs-server host 192.20.5.55
tacacs-server host 192.20.7.22
!
```

2. Places those values in the evaluation

Evaluation List: Evaluation List Criteria: Match All Number:

Evaluation	Match Crit...	Match Crit...	Default R...
permit tcp host $Extraction{extract tacacs server IP address}	Match All		Fail

```
remark ITNM servers
permit tcp host 172.200.201.5 any
permit tcp host 172.10.201.99 any
remark TACACS servers
permit tcp host 192.20.5.58 any
permit tcp host 192.20.5.55 any
permit tcp host 192.20.7.22 any
logging 10.191.101.52
snmp-server community wr1t3Str1ng RW
snmp-server community private RO
```

3. To find these lines of configuration

© Copyright IBM Corporation 2016

Example of an extraction in an evaluation example

After you define your extraction, you can use it in an evaluation. In this example, you place the extraction in a different command. This different command must have IP addresses that match the ones that you extract earlier. In this case, you use the extraction with a series of permit commands that are in an access control list. You search the configuration for permit commands that have the IP addresses of the TACACS servers.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

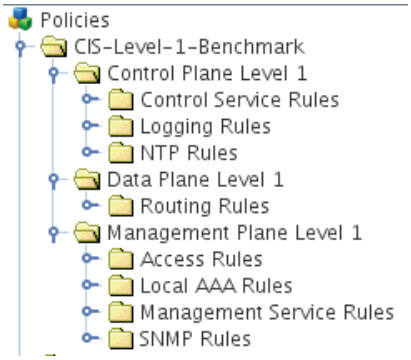
Policy realm implications

Policy realms help organize policies into logical groups

Policy realms are associated with groups and customers

Can hide policy realms can from groups and customers

Global and group parameters are associated with a policy realm

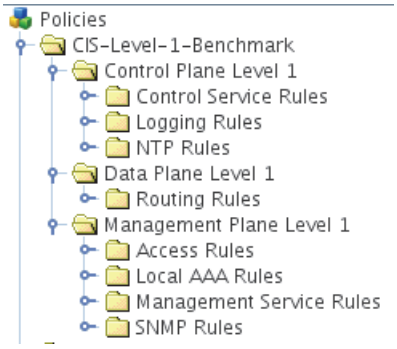


© Copyright IBM Corporation 2016

Policy realm implications

As mentioned, global, and group parameters are associated with a policy realm. Depending on the environment, the application might affect how you use policy realms. Use policy realms in two general ways:

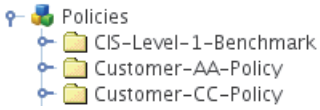
1. *As a method to organize policies*. An example is the CIS-Level-1-Benchmark policy realm. This realm structure is designed to segment policies into specific buckets of analysis. Per the Center for Internet Security, routers are broken down into three planes: *control*, *data*, and *management*. These main areas are further broken down into specific domains that encapsulate relevant policies.



2. *As a means to enforce security for viewing realms*. In this case, it uses the policy realm as a security measure to control which groups can see which realms. For instance, as a managed service provider, you might manage several customers who implement compliance services. Giving the customer access to the compliance software is a means that they can check your management. However, it is correct that the customer sees only their policies and not the

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

policies of another customer. So at a high level, you use policy realms to enforce view rights. In many instances, a base policy realm associates with a customer name.



You control what policy realms and which global and group parameters are viewable to various groups through the **Admin > User Security Options**.

© Copyright IBM Corp. 2016

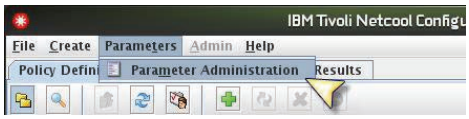
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

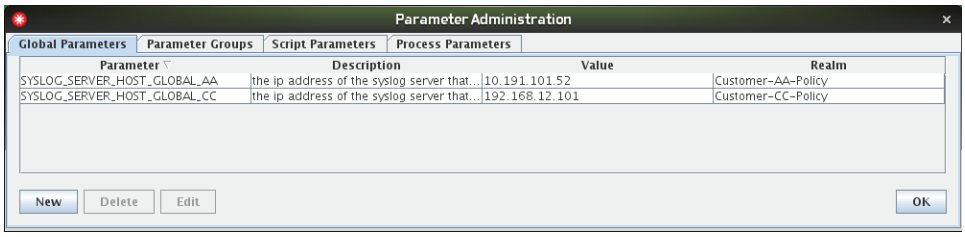
Parameter administration

Allows the creation and editing of global and group parameters

Edit process, local, parameters

Use edited values in the next process execution





© Copyright IBM Corporation 2016

Parameter administration

As mentioned, you can use the Parameter Administration GUI to view global, group, and process parameters. Process parameters are local parameters that associate with a process. That means that you create a process that includes policies that have definitions that use local parameters.

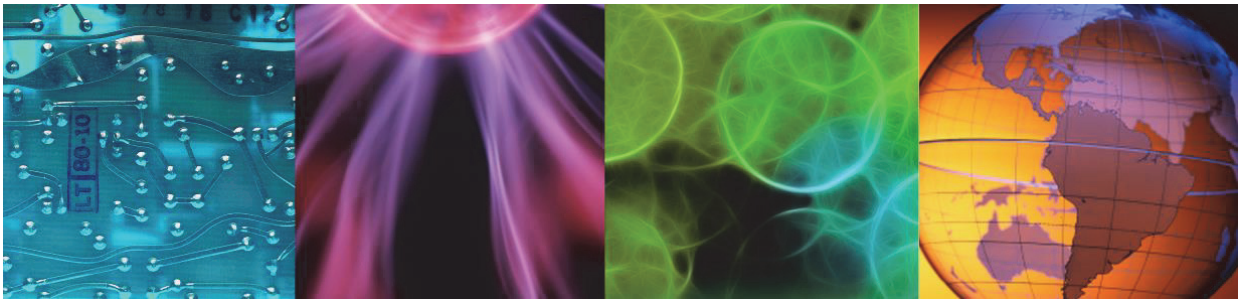
Scripting parameters are associated with a JavaScript function.

Regardless of parameter type, you can use this interface to modify the parameter values.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 6 Using JavaScript

Lesson 6 Using JavaScript



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to describe scripted parameters and definitions.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

JavaScript allows additional flexibility

JavaScript is supported in creating parameter data and definitions

Allows access to external resources to determine compliance

- Databases
- Applications
- Worker server operating system

Allows more complex analysis of internal data

© Copyright IBM Corporation 2016

JavaScript allows additional flexibility

The use of JavaScript in the *compliance manager* is available to you for both the creation of parameter data and for determining whether a device meets a specific condition or not. You can use this scripting in the creation of a parameter and for a definition.

JavaScript adds flexibility in the *compliance manager* by allowing policies to gather data from external databases and applications. You can use this external information to determine compliance for a device.

JavaScript allows more complex searching and analysis of device and configuration data.

This additional flexibility comes at a cost. Policy development requires the user to be familiar with JavaScript and what it can and cannot do.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

JavaScript parameter

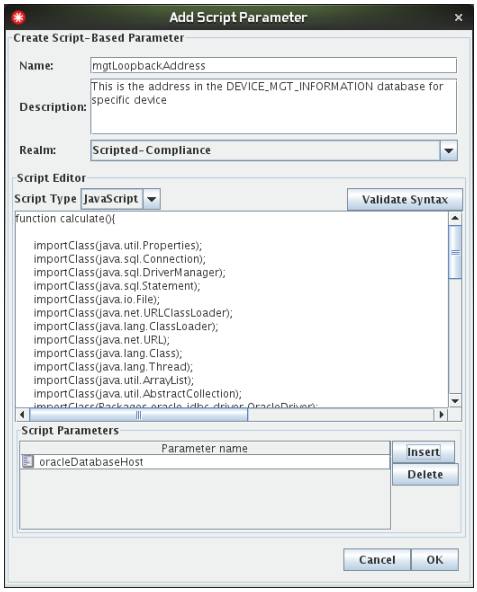
Returns either a value or an ArrayList of values

Defined at a system level

- Policy realm security

Can declare variables in script

- Implicit
- Explicit



© Copyright IBM Corporation 2016

JavaScript parameter

The application allows the creation of a parameter whose value or values a JavaScript function determines. The JavaScript must return a single string or an ArrayList of strings. JavaScript can return something other than a string, such as an integer or float, but it is converted into a string.

Define a JavaScript parameter at a system level like a global parameter, where policy realm security you can employ it to control who can use the parameter. It also shares features of an extraction, where returns one or more values, which are determined at execution time.

The developer can define a set of external parameters for the JavaScript. Define these parameters as *group*, *global*, or *extraction* parameters that pass into the JavaScript. You insert these parameters in the lower half of the GUI and can reference them in the script. If the parameter is a group or extraction, you must write the script to handle multiple values.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

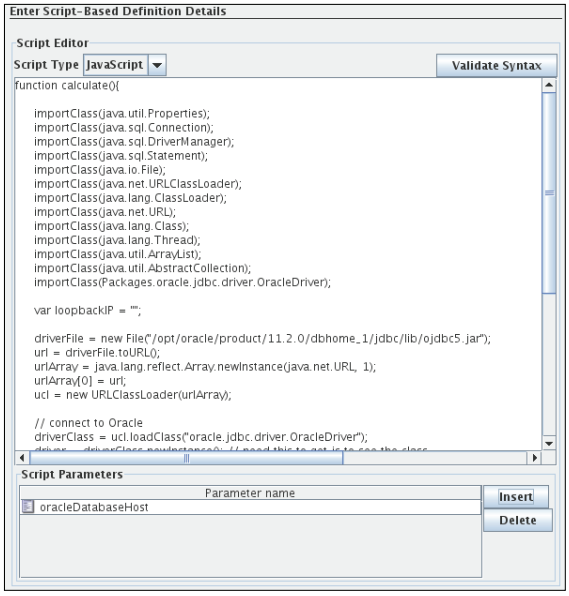
JavaScript definition

Returns either a *true* or *false*

Can declare variables in script

- Implicit
- Explicit

Can output statements to evaluation logs



© Copyright IBM Corporation 2016

JavaScript definition

You can use the application to create a definition whose return value is the Boolean of either *true* or *false*.

The JavaScript you create has access to a set of implicit parameters that pass into the script automatically, which includes data about the device it is evaluating.

You can define a set of external parameters in a similar manner as with the JavaScript parameter and you add them in the lower part of the screen.

You can use the JavaScript definition to send messages to the compliance log so you can track what the script evaluates.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Describe the types of definitions that you can create
- Describe the methods to use parameters in definitions
- Describe how to use JavaScript to enhance definitions
- Create a policy that uses XPath
- Create a policy that uses an extraction
- Test the new policies
- Create a process

© Copyright IBM Corporation 2016

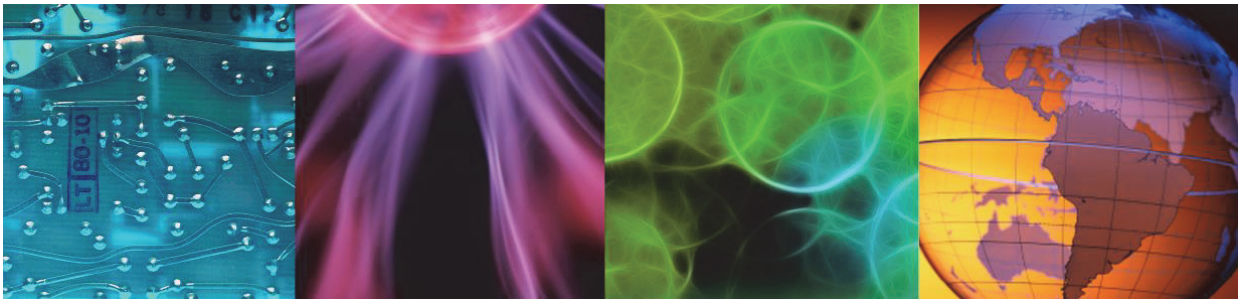
Summary

In this unit, you learned about the more advanced features to determine whether a configuration is compliant. This unit described the different data structures that you can analyze and the different methods for analyzing them. Parameterization of definitions is described in detail. Finally, it introduced the use of JavaScript.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

20 Preemptive compliance

Preemptive compliance



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

This unit teaches how you can use compliance to inspect changes that you make in the *configuration manager* and notify users when changes take a device out of compliance. This unit teaches how to configure the application to enable preemptive compliance and how it affects users.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Objectives

In this unit, you learn to perform the following tasks:

- Describe preemptive compliance
- Describe its effects on UOW execution
- Implement and test preemptive compliance

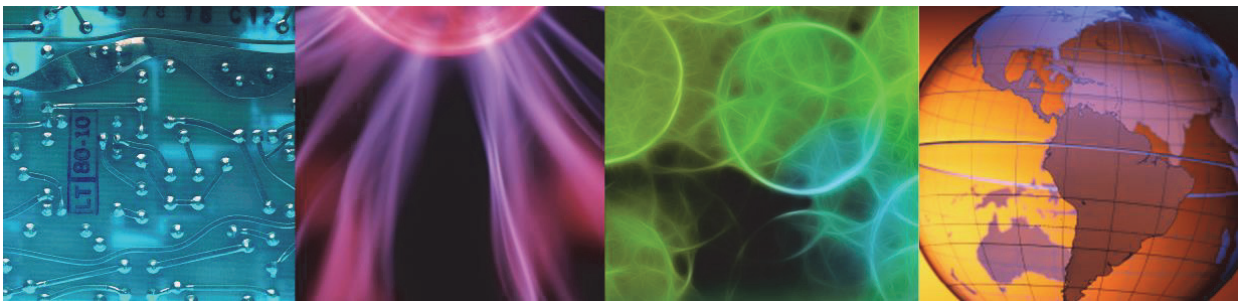
© Copyright IBM Corporation 2016

Objectives

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 1 Overview

Lesson 1 Overview



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe precompliance
- Describe how it impacts submitting and processing *units of work*.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preemptive compliance overview

Integrate the change management with compliance policy

Policies analyze a proposed change to determine whether it is compliant

Can prevent changes from entering network

Think of it as a lot of inspection for changes

© Copyright IBM Corporation 2016

Preemptive compliance overview

Preemptive compliance is where change management and policy compliance unify to inspect changes before applying them to devices. It analyzes a proposed change to check whether the results cause a device to become noncompliant.

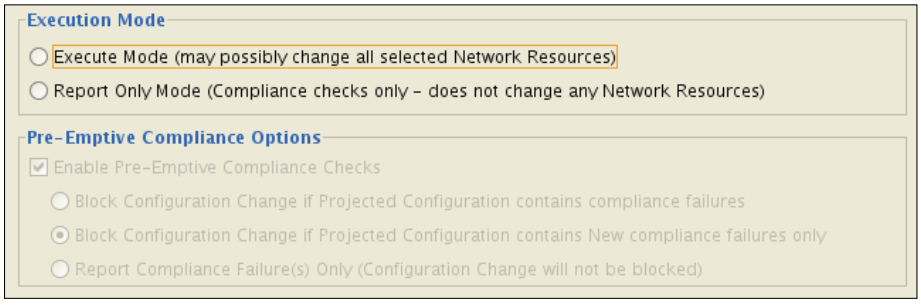
You can configure the inspection technique to prevent a change from being applied to a device. The manufacturing equivalent of precompliance is to inspect every part before it ships to a customer.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preemptive compliance is part of UOW wizards

Most users cannot change preemptive compliance selection

User can test change against policies with report only



© Copyright IBM Corporation 2016 5

Preemptive compliance is part of UOW wizards

From the user perspective, precompliance appears when the user submits units of work and their group is configured to use precompliance. Most users see the precompliance information in the unit of work wizard but are not authorized to change the preemptive compliance options.

Preemptive compliance allows the user to test their change in a *report only* mode. This report shows the changes that violate any preemptive compliance policies. This report is a quick analysis since this analysis is done on the server and does not access devices.

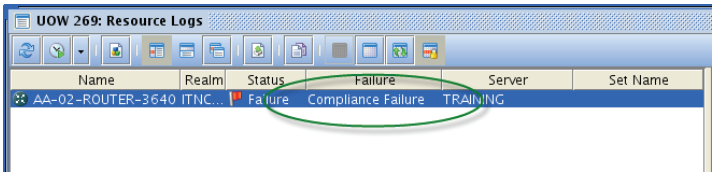
Report only mode is a method that you use to test modeled command sets. This option is the one time a user can test an ad hoc configuration change before applying it.

© Copyright IBM Corp. 2016

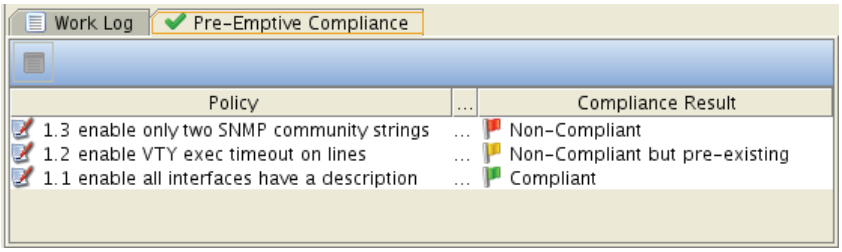
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preemptive compliance affects units of work

Noncompliant changes might or might not be allowed



Determines whether change is making a device less compliant



© Copyright IBM Corporation 2016

Preemptive compliance affects units of work

Preemptive compliance affects a UOW in that it might prevent its execution that depending on the level of preemptive compliance that applies to a group. Based on this criterion, a UOW might fail with the failure noted as **Compliance Failure**.

Preemptive compliance allows a user to see which, if any, preemptive compliance policies are violated. Three levels of compliance exist:

1. **Compliant**: The change either makes a device compliant or does not cause the device to become noncompliant for a particular precompliance policy.
2. **Non-Compliant but pre-existing**: The change does not improve the compliance of a device but neither does it make it less compliant. The device is already noncompliant for this particular policy and the change does not make that noncompliance worse
3. **Non-Compliant**: Applying the change causes the device to become noncompliant for a particular policy.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preemption starts after UOW submission

Change is checked when a UOW enters the *work pending approval* or *work waiting to execute* queue

Change is evaluated against precompliance policies for all processes that:

- Have included the device in the process execution
- Have precompliant policies that are defined and enabled

© Copyright IBM Corporation 2016

Preemption starts after UOW submission

Precompliance checking occurs upon submission of the unit of work and its entry into the workflow. Depending on approvals, this check occurs either when the unit of work is in the *work pending approval* queue or is in the *work waiting to execute* queue.

If a user submits a unit of work that is part of a precompliance group, the change is tested if it meets the following two conditions.

1. The change applies to one or more devices that associate with a process in the *compliance manager.*
2. The associated process defined and enabled preemptive policies.

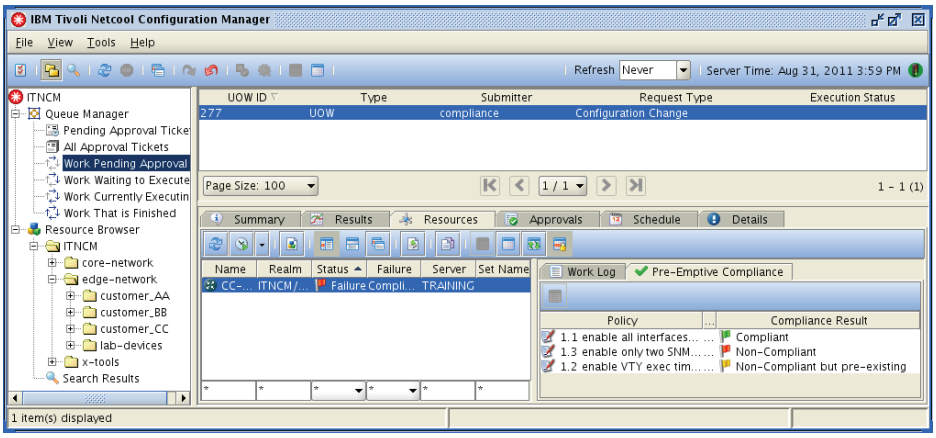
© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preemption affects approvals

Approvers view preemptions in *work pending approval*

Approved units of work run regardless of preemption policy



© Copyright IBM Corporation 2016 8

Preemption affects approvals

If a UOW requires an approval and is from a group that requires preemptive compliance, the approver can select the unit of work. The approver can also select the **Resources** tab, and determine whether the UOW passes the preemptive compliance policies. The approver can also look at the unit of work log, which has log messages that note the precompliance results.

```
*****************************************************
Task: Configuration Change
Device: CC-03-ROUTER-3640
*****************************************************
(1) >>> Operation Started on Worker Server 'itncm'... (preComplianceCheck):
2011/12/09 17:39:13.321 GMT+00:00
(2) Using driver: cisco_router_36xx_c36xx-12.4_v14.00 (Isd0000103872)
(3) Started Pre-emptive Compliance Check
(4) Options: On Compliance Failure :Stop
(5) Getting the new config
(6) Running compliance checks against current and projected configs
(7) Got applicable policies
(8) Finished running policies
(9)
Policy Name                                   Current Config   Projected Config
---------------------------------------------   --------------
----------------
1.1 enable all interfaces have a description   FAIL             FAIL
```


© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

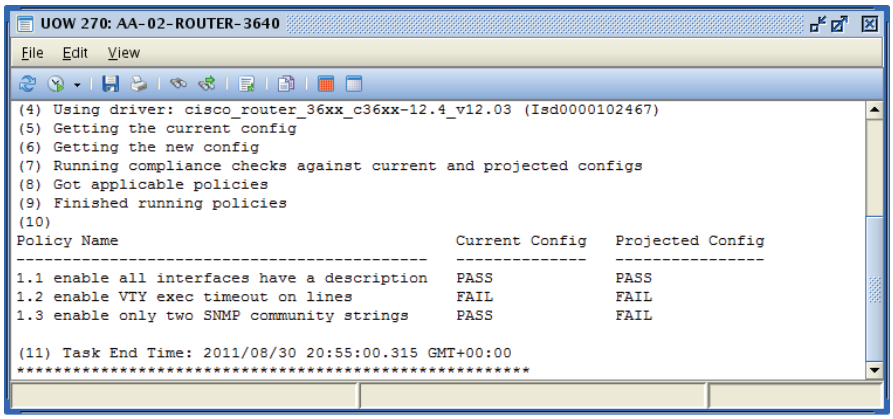
```
1.3 enable only two SNMP community strings     PASS            PASS
1.2 enable VTY exec timeout on lines           FAIL            FAIL

(10) Task End Time: 2011/12/09 17:39:17.589 GMT+00:00
```

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Interpreting the unit of work log

Notice the differences between Current Config and Projected Config



© Copyright IBM Corporation 2016 9

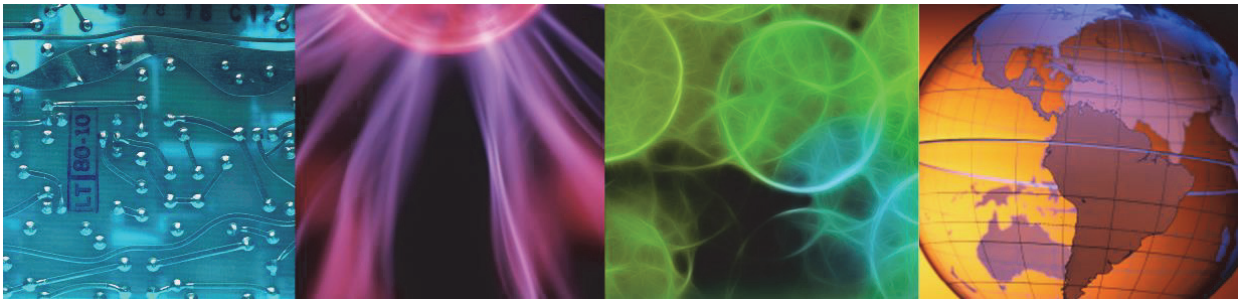
Interpreting the unit of work log

When the UOW log is shown and you employ preemptive compliance, the log shows a message that lists all the preemptive compliance policies that apply to the device. It also shows whether the current configuration is compliant or not compliant with each policy. It then shows whether the proposed change violates any of the preemptive policies. In the example in the slide, the change causes the device to go from compliant on policy **1.3 enable only two SNMP community strings** to noncompliant.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Lesson 2 Requirements and configuration

Lesson 2 Requirements and configuration



© Copyright IBM Corporation 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

In this lesson, you learn how to perform the following tasks:

- Describe the requirements to use precompliance
- Configure the application to use precompliance

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Preemptive compliance reliance on SmartModels

Device must be SmartModeled

The configuration change must be SmartModeled, no native command sets

Precompliance policy must be SmartModeled

© Copyright IBM Corporation 2016 11

Preemptive compliance reliance on SmartModels

It is important to understand that preemptive compliance is a solution that is reliant on SmartModel technology. Precompliance works because SmartModels not only know the contents of the current configuration of a device, but also can predict what the configuration is after a change is applied to it. By knowing this present and future state of the configuration, you can use SmartModels to analyze those states against compliance policies.

To implement preemptive compliance, all aspects of the change management and compliance must use SmartModels. These requirements include the following parameters:

- The device must be SmartModel. The configuration the device uses is modeled and in XML.
- The change that is applied to the SmartModeled device must also be SmartModeled. This action means that the change is either using the configuration editor or a modeled command set to apply a change.
- The definition that is part of the preemptive policy is modeled and uses XPath.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Steps to enable preemption features

Compliance manager

1. Create preemptive policies
2. Associate preemptive policies to process
3. Select appropriate devices in process to apply preemption

Configuration manager

1. Add preemption activities to appropriate group

© Copyright IBM Corporation 2016

Steps to enable preemption features

To enable preemptive compliance, you must configure both the *configuration manager* and the *compliance manager* applications.

From a *compliance manager* perspective, the subject matter expert needs to create preemptive policies and processes. You must complete the following steps to use the preemptive compliance feature:

1. Create policies that use modeled definitions and enable preemption.
2. Create a process that includes these preemptive policies and enables preemption.
3. Select the devices that these preemptive policies apply to.

Modify group security settings in *configuration manager* to implement the preemptive compliance workflow. You must complete the following steps to allow a group to use the preemptive compliance feature:

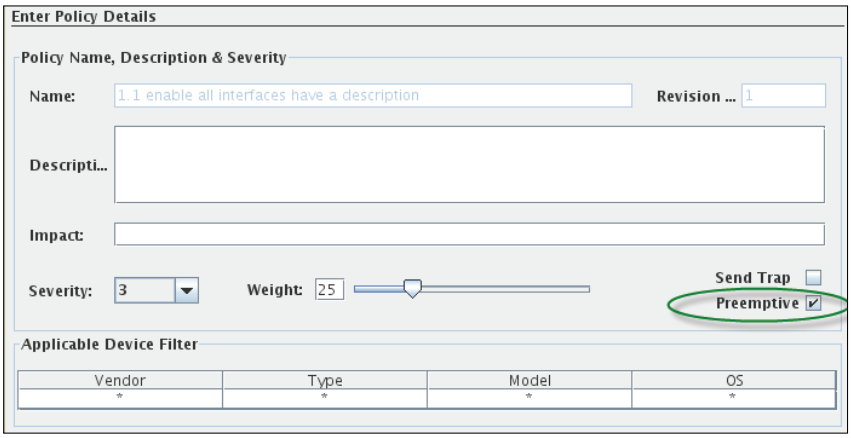
1. Add *execute compliance policy* activities to the group.
2. Add one or more *preemptive compliance* activities to the group.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Define policies as preemptive

User must know that all definitions are modeled



© Copyright IBM Corporation 2016

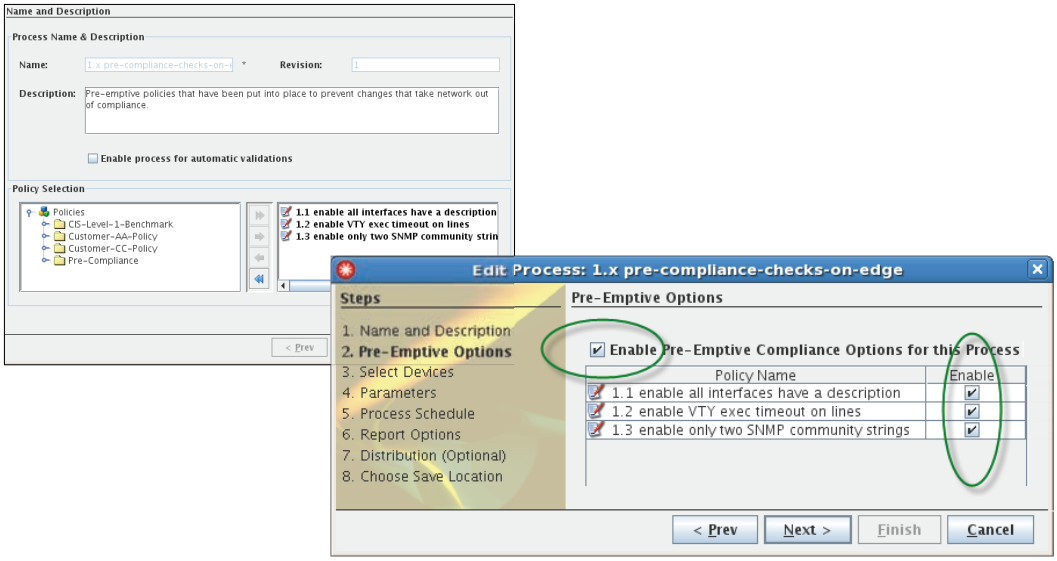
Define policies as preemptive

The first step of preemptive compliance configuration is to enable a policy so the preemption workflow uses it. Select the Preemptive option in the policy wizard. Selecting this option alone does not cause policies to be preemptive and does not affect the policy when it is not used in a preemptive mode.

Important: The policy is not intelligent enough to know the nature of the definitions that are part of the rules that are associated with itself. It is up to the subject matter expert to know that the underlying definitions in the policy structure are modeled.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Process must declare if it has preemptive policies



© Copyright IBM Corporation 2016 14

Process must declare if it has preemptive policies

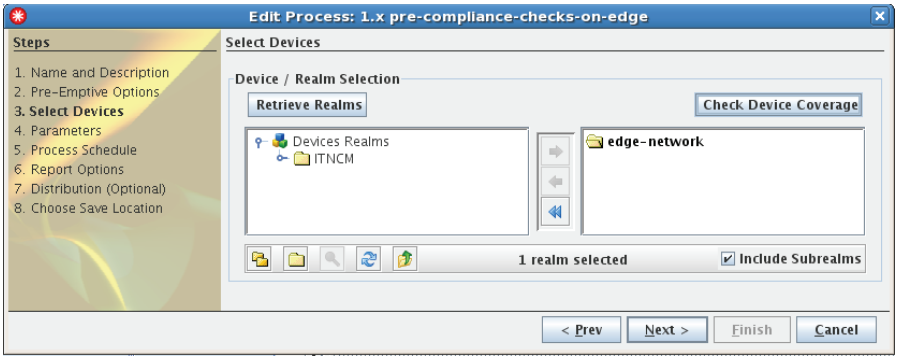
The second step is to enable the preemption workflow in a process. The process must include preemptive policies. It is in this step that a preemptive policy affects the UOW submission. The process creation wizard contains a step that enables the preemptive workflow and the policies that you want as part of the preemption process.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Process must include devices to apply preemption

Preemption is applied only to the devices in the process



© Copyright IBM Corporation 2016

Process must include devices to apply preemption

The third step requires the selection of the devices that the policies affect. It is important to understand that preemption is limited to the policies and devices that are in this process.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Add preemptive activities to appropriate group

Execute Compliance Policy required

Pre-Emptive Compliance (Block on failure)

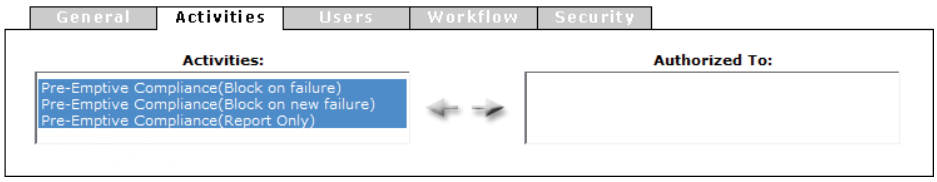
Blocks any modeled change operations from completing if any precompliance failures exist against the projected modeled configuration. For example, fixing the non-compliance before any other change is allowed.

Pre-Emptive Compliance (Block on new failure)

Blocks any modeled change operations from completing if the change causes any new precompliance failures to occur. For example, non-compliance issues can exist on device for now.

Pre-Emptive Compliance (Report Only)

Reports only the status of preemptive compliance in the audit log



© Copyright IBM Corporation 2016 16

Add preemptive activities to appropriate group

To implement preemptive compliance in *configuration manager*, you must modify group security settings to the group that the analysis affects.

First, the group whose *units of work* are checked must have *execute compliance policy* included in their activities. If this activity is not included, the next step has no effect. If the following message is seen in the UOW log and you are expecting preemptive policies to run, it is because you did not include this activity.

```
*****************************************************
Task: Configuration Change
Device: CC-02-ROUTER-3640
*****************************************************
(1) >>> Operation Started on Worker Server 'itncm'... (preComplianceCheck):
2011/12/09 13:42:46.284 GMT+00:00
(2) Using driver: cisco_router_36xx_c36xx-12.4_v14.00 (Isd0000103872)
(3) Started Pre-emptive Compliance Check
(4) Options: On Compliance Failure :Stop
(5) Getting the new config
(6) Running compliance checks against current and projected configs
(7) There are no applicable Compliance Policies
(8) <<< Operation Finished in 0 seconds (preComplianceCheck): 2011/12/09
13:42:46.780 GMT+00:00
(9) Task End Time: 2011/12/09 13:42:46.817 GMT+00:00
```


Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Second, select one or more preemptive compliance activities to add to the group. Note the level of restriction associated with each activity.

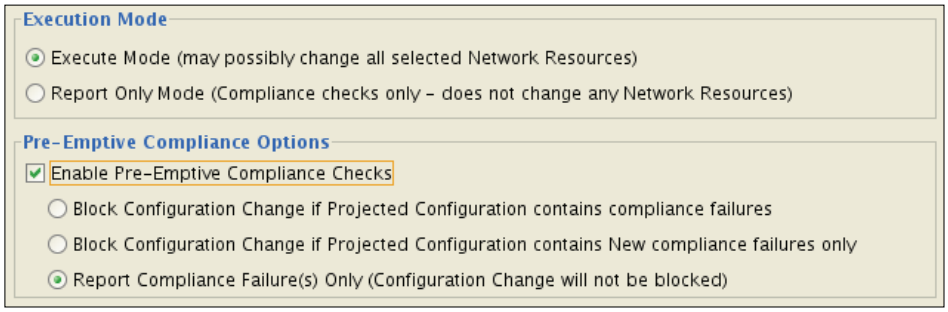
The following list contains the effects of the preemptive activities:

- **Block on failure**: This activity is the most restrictive of preemption policies in that no change can apply to a device until all preemptive policies pass. This activity means if a device has any preemptive failures, you must fix all failures before any other change to the device is allowed.
- **Block on new failure**: This preemption policy allows any changes to the device if they do not cause a new policy violation. This activity means if preemptive policies are noncompliant on a device you are not required to fix them to apply a change. However, the change cannot cause another preemptive policy to become noncompliant.
- **Report only**: Allows any change on the device and reports the preemptive violations in the UOW log.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Groups who can disable preemptive compliance

Groups with *manage work* activities can override preemptive compliance



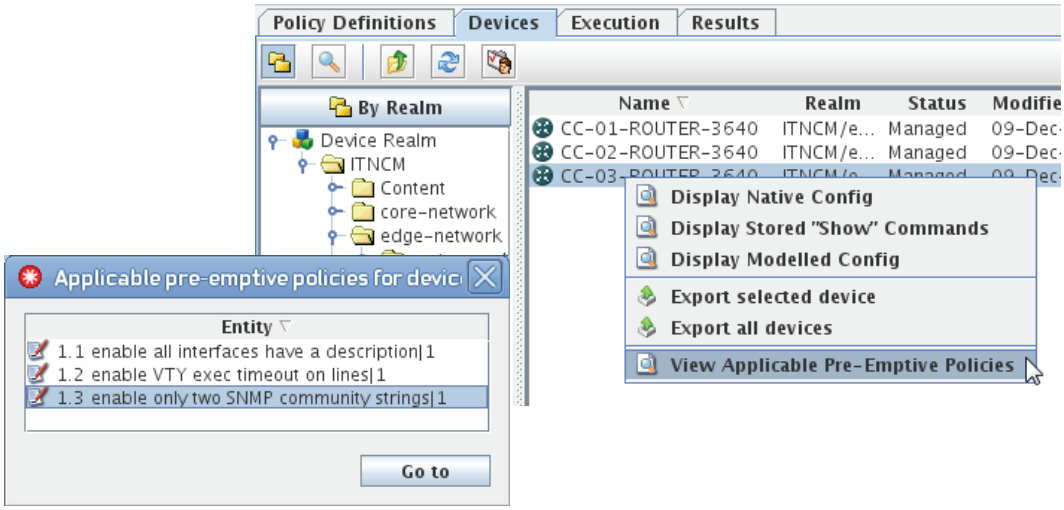
© Copyright IBM Corporation 2016 17

Groups who can disable preemptive compliance

In general, when a group is assigned Pre-emptive Compliance options, they are not available to edit and cannot change. However, if a group has *manage work* activities and can approve work, they can select their own preemptive compliance option or disable it altogether.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Viewing preemptive policies for a device



© Copyright IBM Corporation 2016 18

Viewing preemptive policies for a device

If you are in the *compliance manager*, you can select a device in the *compliance manager* and see the preemptive policies that are assigned against it. Selecting a policy and clicking **Go To** shows you the policy in the **Policy Definitions** tab.

 © Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Student exercises

© Copyright IBM Corporation 2016

Student exercises

Open your *Course Exercises* book and perform the exercises for this unit.

© Copyright IBM Corp. 2016

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Summary

You now should be able to perform the following tasks:

- Describe preemptive compliance
- Describe its effects on UOW execution
- Implement and test preemptive compliance

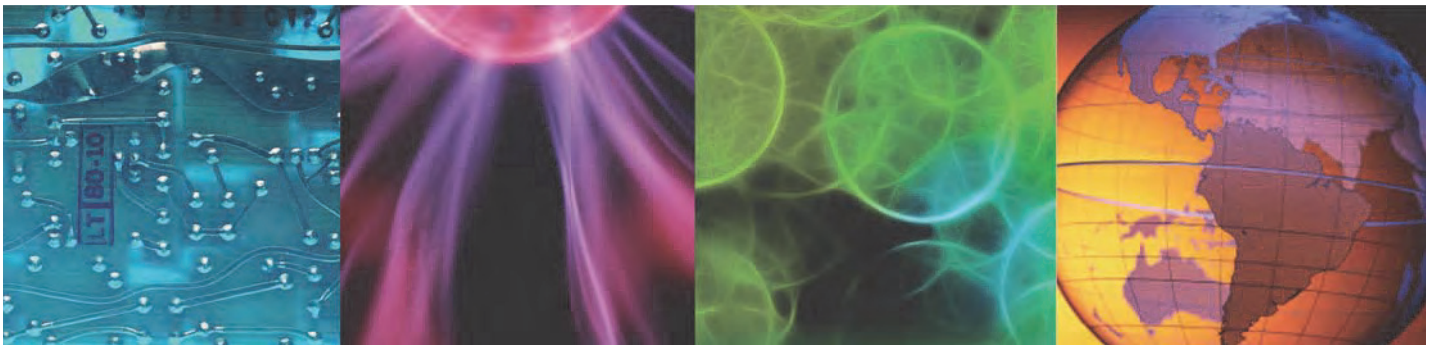
© Copyright IBM Corporation 2016

Summary

This unit covered how you can use compliance to inspect changes that you make in the configuration manager and notify users if changes take a device out of compliance. This unit also covered how to configure the application to enable preemptive compliance and how it affects users.

© Copyright IBM Corp. 2016
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

TOD44 1.0



ibm.com/training

Authorized IBM Training