drops

User Guide

DROPS

Version 24.0

Publication Date: January, 2024

Prepared by the DROPS Software Documentation Team

North America & LATAM
1 N. State St, 15th Floor
Chicago, IL
USA
1-603-371-9074
1-603-371-3256 (support calls only)
sales-us@arcadsoftware.com

EMEA (HQ)
55 Rue Adrastée – Parc Altaïs
74650 Chavanod/Annecy
France
+33 450 578 396
sales-eu@arcadsoftware.com

Asia Pacific
5 Shenton Way #22-04
UIC Building
Singapore 068808
sales-asia@arcadsoftware.com

Contact DROPS Software

Headquartered in France at the foot of the Alps, DROPS Software offers global services and has offices and partners all over the world. ARCAD partners with leading-edge companies throughout the world to offer full services, close to home.

Visit our website to Contact Us and find out more about our company and partners, or to request a demo.

The ARCAD Customer Portal is intended for current and potential customers that have full or trial versions of ARCAD software. If you already use or are interested in using an ARCAD product, the portal lets you view all of your current licenses and generate your own temporary license keys for most ARCAD products. It grants you access to the ARCAD product knowledge base (new releases, release notes and current documentation).

Do you have a request for change or have you encountered a bug? Log into the ARCAD Helpdesk and create a ticket.

DROPS Software guarantees consultant support 24 hours a day, 5 days a week (24/5) to registered members. Calls received are redirected, according to the hour, to put you in contact with a support team in or near your timezone.

Country	Address	Account Contact	Support Contact
France	ARCAD Software (HQ) 55 Rue Adrastée 74650 Chavanod ARCAD Software 17 chemin de la plaine 07200, Saint-Didier-sous-Aubenas	+33 4 50 57 83 96 sales-eu@arcadsoftware.com	Worldwide 24/7: +1 603 371 3256 France only: +33 450 57 28 00 support@arcadsoftware.com ARCAD Helpdesk
Germany	ARCAD Software Deutschland GmbH c/o Pramex International GmbH Savignystr. 43, 60325 Frankfurt am Main		
China	ARCAD Software #2035, Yuehai Plaza, 180 Wanbo 2nd Road, Nancun, Panyu District, Canton	+86 (020)22324643 +86 (020)22324649 sales-asia@arcadsoftware.com	
India	ARCAD Software D-280/281/282, Vibhuti Khand Gomti Nagar, Lucknow		
Singapore	ARCAD Software 5 Shenton Way #22-04 UIC Building Singapore 068808		
USA	ARCAD Software 1 N. State St, 15th Floor Chicago, IL	+1 (603) 371-9074 +1 (603)-371-3256 (support calls only) sales-us@arcadsoftware.com	

Table 1: Contact DROPS Software

Preface

Document Purpose

This document is intended to guide DROPS Server Administrators through admin-level server management procedures.

This document is intended to guide all DROPS users through low-level studio configuration process and how to use the DROPS Studio.

Intended Audience

This document is intended for the following DROPS users, among others:

- administrators
- configuration managers
- release managers
- deployment process designers
- deployment operators

About the Entity Access Tables

At the beginning of most of the chapters in this document, an entity access table contains important pieces of information concerning the entity described in the chapter.

Required Roles	Environment Release Manager and/or Release Operator
Required Rights (by category)	Deployment Process Instances
Perspective	Operational Console Management
Access	DROPS Server → Deployment Process → Deployment Process Instances

1. If a **role** is required in order to carry out any action concerning the entity described, it is listed. Not all roles are required for all of the actions described.
2. If any **rights** are required for the user's profile in order to access, edit, delete, manage, etc. anything concerning the entity described, the parent right category is listed.
3. The **perspective** in which the entity is found.
4. The path to the entity in the **Configuration Explorer** or **Console Explorer**.

> ***Reference***
> For more information about how to navigate inside the DROPS Studio, refer to the Overview of the DROPS Studio on page 34.

Related Documentation

Related Documentation
ARCAD SSL Configuration Guide
DROPS Configuration Guide
DROPS Datasheet

Table 2: Related Documentation

Related Documentation
DROPS Installation Guide
DROPS Release Notes
DROPS Script Reference Guide
SYSchange User's Guide
ARCAD Installation Documentation

Table 2: Related Documentation

Publication Record

Unless stated otherwise, all content is valid for the most current version of DROPS listed as well as every subsequent version.

Product Version	Document Version	Publication Date	Update Record
≥ 24.0	3.2	January, 2024	No functional changes
23.3	3.1	November, 2023	User documentation updated for product evolution: • New option to re-execute Deployment Plans Instances • Enhanced options for the Azure DevOps plug-in
23.2	3.0	July, 2023	User documentation updated for product evolution: • New option to re-execute Deployment Plans Instances • Define deployment process with the new Process-as-code feature • Enhanced options for the Azure DevOps plug-in
23.1	2.9	April, 2023	User documentation updated for product evolution: • New automatic release resolver • New default scheduling for deployment processes • Updated instructions for the Azure DevOps plugin

Table 3: DROPS User Guide Publication Record

Contents

Contact DROPS Software 3
Preface 4
Contents 7
Tables 19
Figures 21

Introduction

1 About DROPS 26
1.1 The DROPS Server 27
1.2 The DROPS Studio 28
1.3 The DROPS Agent 28
1.4 The DROPS web app 29
2 The DROPS architecture & workflow 31
3 Main concepts 33
4 Overview of the DROPS Studio 34
5 Functional constraints 35

Getting Started

6 Managing the DROPS Server 37
6.1 Creating a server connection 37
6.2 Connecting to a DROPS Server 38
6.3 Configuring the TLS settings 38
6.4 Defining DROPS Server preferences 39
6.5 Configuring a DROPS Server 40
6.6 Viewing DROPS Server logs 40
6.7 Verifying DROPS Server versions 40
6.8 Deleting a DROPS Server connection 40
7 Changing your password 41

User Administration

Introduction to user administration 43
8 User accounts 44
8.1 Creating users 44
8.1.1 Create new users in the DROPS Studio 45
8.1.2 Import new users from an external LDAP (Active Directory) 45
8.2 Editing a user's ID 45

8.2.1 Define a user's login credentials 46
8.3 Adding users to groups from the editor 47
8.4 Assigning profiles to users 47
8.4.1 View the rights granted to profile(s) 48
8.5 Deleting users 48
9 User groups 49
9.1 Creating user groups 49
9.2 Adding users to groups from the editor 50
9.3 Deleting user groups 50
10 Importing LDAP Users 52
11 Assigning entity-level roles 54
12 Accessing your profile and rights 56

Configuring the import process

Introduction to configuring the import process 58
13 Applications 60
13.1 Creating applications 60
13.2 Editing application details 61
13.3 Deleting applications 66
14 Components 67
14.1 Accessing components 67
14.2 Creating components 68
14.3 Editing component details 69
14.4 Changing the type of a component 70
14.5 Changing the extension of a component 71
14.6 Deleting components 71
15 Import strategies 72
15.1 Accessing import strategies 73
15.2 Creating import strategies 73
15.2.1 Create a new strategy 74
15.2.2 Use or copy from strategy templates 74
15.3 Editing import strategy details 75
15.4 Deleting import strategies 78
16 Releases 79
16.1 Creating releases 80
16.2 Editing release details 80
16.3 Viewing environments to which a release has been deployed 82
16.4 Making comments 82
16.5 Viewing attachments 82

16.6 Viewing associated workitems 83
16.7 Locking and unlocking releases 83
16.8 Viewing the artifacts in releases 83
16.9 Archiving releases 83
16.10 Deleting releases 84
17 Working in release branching mode 85
17.1 Switching to release branching mode 86
17.2 Searching for release branches 86
17.3 Creating branched releases 87
17.4 Editing and viewing branched release details 87
18 Imported artifacts 88
18.1 Searching for imported artifacts 88
18.2 Viewing an artifact's import history 89

Configuring the deployment process

Introduction to configuring the deployment process 92
19 Multiple clients 94
20 Environments 96
20.1 Creating environments 97
20.1.1 Create a new environment 97
20.1.2 Duplicate an environment 98
20.2 Editing environment details 98
20.3 Adding infrastructure items 100
20.4 Defining allowed applications 101
20.5 Viewing the deployed releases 101
20.6 Viewing the deployment history 102
20.7 Changing the extension type 102
20.8 Executing compliance tests 103
20.9 Deleting environments 103
21 Environment groups 104
21.1 Accessing environment groups 104
21.2 Creating environment groups 105
21.3 Editing environment group details 105
21.4 Adding environments to groups 106
21.5 Deleting environment groups 107
22 Infrastructure items 108
22.1 Creating infrastructure items 109
22.1.1 Create new infrastructure items 109
22.1.2 Duplicate an existing infrastructure item 111

22.1.3 Import infrastructure items 111
22.2 Editing infrastructure items 115
22.3 Remote SSH options 121
22.4 Viewing the deployed releases 121
22.5 Viewing the deployment history 121
22.6 Changing the type of an infrastructure item 122
22.7 Changing the OS family of an infrastructure item 122
22.8 Downloading remote agent logs 122
22.9 Executing compliance tests 122
22.10 Provisioning infrastructure items 123
22.10.1 View provisioning results 123
22.11 Instantiating infrastructure items 123
22.11.1 View instantiation results 123
22.12 Managing an infrastructure item’s remote installation properties 124
22.12.1 Generating an infrastructure item’s remote installation properties 125
22.12.2 Creating remote installation information in the DROPS database 125
22.13 Testing the connection to an infrastructure item (agent) 125
22.13.1 Test the connection to an infrastructure item (agent) manually 126
22.13.2 Monitor the connection to an infrastructure item (agent) automatically 126
22.13.2.1 Accessing the Remote Agent Monitoring List 127
22.14 Deleting infrastructure items 128
23 SSH keys 129
23.1 Create new SSH keys 130
23.2 Import existing SSH keys 131
23.3 Editing SSH keys 131
23.4 Accessing the public key 131
23.5 Deleting SSH keys 131
24 Deployment schemas 132
24.1 Accessing deployment schemas 133
24.2 Creating deployment schemas 133
24.3 Editing deployment schemas 134
24.4 Managing deployment rules 134
24.5 Deleting deployment schemas 135
25 Replacement rules 136
25.1 Accessing replacement rules 137
25.2 Creating replacement rules 138
25.3 Editing replacement rules 138
25.4 Deleting replacement rules 139
26 Compliance rules 140
26.1 Creating compliance rules 140

26.2 Editing a compliance rule 141
26.2.1 Editing a compliance rule's properties 141
26.2.2 Editing a compliance rule's content 143
26.3 Importing and exporting compliance rules 143
26.3.1 Exporting a compliance rule 144
26.3.2 Importing a compliance rule 144
26.4 Executing compliance rules 144
26.4.1 Grouping compliance rules into rule sets 145
26.4.2 Executing compliance rule sets 145
26.5 Understanding compliance rule logs and compliance reports 146
26.5.1 Viewing compliance rule execution logs 147
26.5.2 Searching for compliance reports 148
26.6 Deleting compliance rules and rule sets 149
27 Scripts 150
27.1 Creating scripts 151
27.1.1 Create a new script 151
27.1.2 Duplicate a script 152
27.1.3 Create a new version of a script 152
27.2 Editing scripts 152
27.2.1 Edit script details 153
27.2.2 Edit script content 155
27.3 Importing scripts 157
27.4 Exporting scripts 157
27.5 Executing scripts 158
27.6 Deleting scripts 159
28 Deployment processes 160
28.1 Accessing deployment processes 160
28.2 Creating deployment processes 161
28.2.1 Create a new deployment process 161
28.2.2 Duplicate a deployment process 162
28.3 Changing a deployment process' status 164
28.4 Defining default deployment processes 164
28.5 Deleting deployment processes 165
28.6 Building deployment process diagrams 165
28.6.1 Manage deployment activities 165
28.6.1.1 Create new deployment activities 166
28.6.1.2 Use deployment activity templates 167
28.6.1.3 Define the order in which to deploy activities 168
28.6.1.4 Rearrange deployment activity boxes 169
28.6.1.5 Delete deployment activities 170
28.6.2 Manage deployment process stages 170

28.6.2.1 Add new stages to deployment activities 171
28.6.2.2 Add template stages to deployment activities 171
28.6.2.3 Delete stages from deployment activities 172
28.6.3 Manage deployment actions 172
28.6.3.1 Add deployment actions to activities and stages 172
28.6.3.2 Delete deployment actions 174
28.6.4 Create "Wait For" Connections 174
28.6.5 Print deployment process diagrams 175
28.7 Editing deployment process properties 175
28.7.1 Edit global deployment process details 175
28.7.2 Edit global deployment process properties 177
28.7.2.1 Use a properties template 181
28.7.3 Edit deployment activity properties 182
28.7.4 Edit deployment action properties 187
29 Deployment plans 191
29.1 Searching for deployment plans 192
29.2 Creating deployment plans 192
29.3 Duplicating deployment plans 193
29.4 Deleting deployment plans 194
29.5 Building deployment plan diagrams 194
29.5.1 Manage the application(s) to deliver 195
29.5.1.1 Add application groups 195
29.5.1.2 Add applications to groups to deploy 196
29.5.1.3 Delete applications from groups to deploy 197
29.5.2 Manage delivery stages 197
29.5.2.1 Add delivery stages 197
29.5.2.2 Delete delivery stages 198
29.5.3 Manage environment(s) 198
29.5.3.1 Add environments to delivery stages 198
29.5.3.2 Delete environments from delivery stages 199
29.5.4 Print deployment plans 199
29.6 Editing deployment plan properties 199
29.6.1 Edit global plan properties 200
29.6.2 Edit delivery stage properties 201
29.6.3 Edit environment properties 204
29.6.4 Edit application group properties 205

Templates

Introduction to configuration templates 208
30 Application templates 209
30.1 Creating application templates 210

30.2 Deleting application templates ... 211
31 Component types ... 212
31.1 Creating component types ... 213
31.2 Deleting component types ... 214
32 Import strategy templates ... 215
32.1 Creating import strategy templates ... 215
32.2 Editing import strategy templates ... 216
32.3 Deleting import strategy templates ... 217
33 Deployment process templates ... 218
33.1 Creating deployment process templates ... 219
33.1.1 Create a deployment process template manually ... 219
33.1.2 Create an activity or a stage template from an existing diagram ... 220
33.2 Editing deployment process templates ... 221
33.3 Viewing the process diagrams that use a template ... 221
33.4 Deleting deployment process templates ... 222
34 Environment provisioning templates ... 223
34.1 Creating provisioning templates ... 223
34.2 Editing provisioning templates ... 224
34.2.1 Edit provisioning template details ... 224
34.2.2 Edit provisioning script content ... 226
34.3 Associating provisioning templates with infrastructure items ... 227
34.4 Deleting provisioning templates ... 227
35 Instantiation templates ... 229
35.1 Creating instantiation templates ... 229
35.2 Editing instantiation templates ... 230
35.2.1 Edit instantiation script content ... 232
35.3 Associating instantiation templates with infrastructure items ... 233
35.4 Deleting instantiation templates ... 233

Main / Node Configuration

Introduction to main / node servers configuration ... 235
36 Configuring main and node DROPS servers ... 236
36.1 Registering node servers ... 236
36.2 Managing registered nodes ... 236
37 Synchronizing main and node DROPS servers ... 238
37.1 Synchronizing servers ... 238
37.2 Scheduling synchronization ... 239
37.3 Synchronization reports ... 240

Importing artifacts

Introduction to importing artifacts 242

38 Managing applications 243

38.1 Managing Allow-to-Release applications 243

38.2 Managing favorite applications 243

38.3 Viewing import history by application 243

39 Importing artifacts immediately by application 245

40 Import process instances 248

40.1 Searching for import process instances 248

40.2 Creating import process instances 250

40.2.1 Create new import process instances 250

40.2.2 Duplicate an import process instance 252

40.3 Executing import process instances 253

40.3.1 Pause scheduled import process instances 253

40.4 Understanding import process instance results 254

40.4.1 View the Logs 254

40.4.2 View the results for each import strategy 255

40.5 Viewing associated workitems 255

40.6 Deleting import process instances 255

Deploying releases

Introduction to deploying releases 257

41 Deployment process instances 259

41.1 Searching for deployment process instances 259

41.2 Creating deployment process instances 262

41.2.1 Create a deployment process instance 263

41.2.2 Create an instance to deploy a release set to an environment 264

41.2.3 Create multiple instances to deploy one release to an environment group 266

41.2.4 Duplicate a deployment process instance 268

41.3 Changing the release to deploy 268

41.4 Managing deployment tags 269

41.5 Defining an instance's deployment mode 270

41.5.1 Mark an instance to deploy in full-transfer mode 270

41.5.2 Mark an instance to deploy in transfer-only mode 271

41.5.3 Create installation-only instances 271

41.6 Executing deployment process instances manually 272

41.6.1 Execute full-transfer and transfer-only instances 272

41.6.2 Execute installation-only instances 273

41.7 Scheduling deployment process instances 273

41.7.1 Schedule full-transfer and transfer-only instances 274
41.7.2 Schedule installation-only instances 274
41.7.2.1 Managing an installation-only instance's automatic schedule 275
41.7.3 Modify any instance's schedule 276
41.8 Remote deployments on node servers 276
41.9 Understanding deployment process instance results 277
41.9.1 View the steps carried out 277
41.9.2 View the Logs 278
41.9.3 View the attachments 280
41.10 Viewing associated workitems 281
41.11 Re-executing failed deployments 281
41.12 Rolling back a deployed deployment process instance 282
41.12.1 Roll back a deployment manually 282
41.12.2 Schedule a deployment's rollback 282
41.13 Canceling deployment process instances 283
41.14 Deleting deployment process instances 283
42 Deployment plan instances 284
42.1 Creating deployment plan instances 285
42.2 Executing deployment plan instances 286
42.3 Understanding deployment plan instance results 287
42.3.1 View the instance status 287
42.3.2 View the steps carried out 287
42.3.3 View the Logs 288
42.4 Re-executing failed deployments 288
42.5 Rolling back a deployed deployment plan instance 289
42.6 Canceling deployment plan instances 289
42.7 Deleting deployment plan instances 289
43 Interruption messages 291
43.1 Responding to interruption messages 292
43.2 Reviewing processed interruption messages 293
44 Suspended actions 294
44.1 Responding to suspended actions 294
44.2 Reviewing processed suspended actions 295
45 Validation requests 296
45.1 Responding to requests 297
45.1.1 Fill out the validation request editor 297
45.1.2 Validate immediately 298
45.2 Reviewing processed requests 298

Managing campaigns

Introduction to Campaigns 300
46 Campaigns 301
46.1 Creating campaigns 301
46.2 Executing campaigns 304
46.3 Understanding campaign execution results 305

DROPS for ARCAD

Introduction to DROPS for ARCAD 308
47 Configuring ARCAD connections 311
47.1 Connecting to the ARCAD development server 311
47.2 Defining connections to target ARCAD-Servers 313
47.2.1 Creating connections to target ARCAD-Servers 313
47.2.1.1 Creating a new connection 314
47.2.1.2 Duplicate a connection 315
47.2.2 Deleting connections to target ARCAD servers 315
48 Generating ARCAD elements 316
49 Managing ARCAD remote templates 319
49.1 Working with remote application templates 319
49.1.1 Searching for remote application templates 319
49.1.2 Creating remote application templates 320
49.1.2.1 Create a new remote application template 320
49.1.2.2 Import an application's layout to create a template 321
49.1.3 Editing remote application templates 322
49.1.4 Linking remote environment templates 323
49.1.5 Linking remote application templates to infrastructure items 324
49.1.6 Deleting remote application templates 324
49.2 Working with remote environment templates 324
49.2.1 Searching for remote environment templates 324
49.2.2 Creating remote environment templates 325
49.2.2.1 Create a new remote environment template 325
49.2.2.2 Import an environment's layout to create a template 326
49.2.3 Editing remote environment templates 326
49.2.4 Deleting remote environment templates 328
49.3 Working with the target ARCAD runtime updater 328
49.4 Updating an ARCAD target's runtime 331

Plug-ins & APIs

Introduction to the DROPS plug-ins & APIs 334
50 Ant tasks 335
51 DROPS' Public REST API 343

51.1 Basic principles 343
51.2 Using DROPS' Public REST API to create releases 345
51.2.1 Opening the release 345
51.2.2 Creating the import instance 346
51.2.3 Running the import instance 348
51.2.4 Reading the result of the import instance execution 349
51.3 Using DROPS' Public REST API to deploy 350
51.3.1 Creating the deployment instance 351
51.3.2 Running the transfer instance, then the deployment instance 355
51.3.3 Rolling back 358
51.3.4 Tracking execution instances 360
52 DROPS for Jenkins 362
52.1 Prerequisites 362
52.2 Configuring the DROPS for Jenkins plug-in 362
52.2.1 Adding the plug-in to Jenkins 363
52.2.2 Setting the connection details 363
52.3 Adding DROPS steps to a project 364
52.3.1 Defining DROPS: Import steps 364
52.3.2 Defining DROPS: Deploy steps 366
52.3.3 Adding scripted steps to a pipeline project 367
52.4 Understanding results in the DROPS for Jenkins plug-in 368
53 DROPS for JIRA 370
53.1 Main concepts 370
53.2 Functional constraints 371
53.3 Prerequisites 371
53.4 Installing the DROPS for JIRA add-on 372
53.4.1 Uninstall the DROPS for JIRA add-on 373
53.5 Configuring the DROPS for JIRA add-on 374
53.5.1 Configure the DROPS Server connection 375
53.5.2 Define the type of issue to create in the DROPS project 376
53.5.3 Map your JIRA projects to your DROPS applications 377
53.5.4 Map your DROPS workitem provider to JIRA 378
53.6 Adding the request to deploy to your development project workflow 379
53.7 Define the request/requested by link between issues 384
53.8 Adding the instruction to launch the deployment to your DROPS project workflow 385
53.9 Adding custom fields to deployment request issues in your DROPS project 387
54 DROPS for JIRA Cloud 391
54.1 Installation 391
54.2 Configuring the DROPS Server connection 391
54.3 Configuring the JIRA user's roles and rights (permissions) 392
54.4 Mapping DROPS applications 393

54.5 Mapping DROPS environments 393
54.6 Importing 393
54.6.1 Importing at project level 393
54.6.2 Importing at issue level 394
54.7 Deploying 394
54.7.1 Deploying at project level 394
54.7.2 Deploying at issue level 395
55 DROPS for Azure DevOps 396
55.1 Compatibility 396
55.2 DROPS Server connection 396
55.3 Configuring DROPS's pipeline tasks 397
56 Workitems 401
56.1 Workitem providers 401
56.1.1 Creating workitem providers 401
56.1.2 Editing general workitem provider details 402
56.1.3 Editing JIRA workitems 403
56.1.4 Deleting workitem providers 404
56.2 Workitems 404

Appendices

Entity types 407
Properties 412
Server properties 414
Substitution variables 416
Roles 426
Rights 430
Component type extended configuration 432
Import strategy type configuration 433
Infrastructure item type extended configuration 455
Environment type extended configuration 458
Rollback 459
The Command Line Interface (CLI) 461
Automatic Release Resolver 466
OpenSSH Agentless connection for Windows OS 469
Troubleshooting 470
Glossary 475

Tables

Table 1: Contact DROPS Software 3
Table 2: Related Documentation 4
Table 3: DROPS User Guide Publication Record 6
Table 4: Edit environment groups 106
Table 5: Syntax for agent protocols 113
Table 6: Syntax for agentless protocols 113
Table 7: DROPS Agent Installation properties 124
Table 8: Deployment Rule example 132
Table 9: Compliance Rule results 147
Table 10: Edit provisioning engine properties: ANT processor 225
Table 11: Edit provisioning engine properties: Command-Line processor 226
Table 12: Edit provisioning engine properties: Ansible Playbook processor 226
Table 13: Edit provisioning engine properties: Microsoft Azure 226
Table 14: Edit provisioning engine properties: ANT processor 231
Table 15: Edit provisioning engine properties: Command-Line processor 231
Table 16: Edit provisioning engine properties: Ansible Playbook processor 232
Table 17: Edit provisioning engine properties: Microsoft Azure 232
Table 18: Import process instance statuses & results 249
Table 19: Deployment process instance modes 261
Table 20: Deployment process instance statuses & results 261
Table 21: Deployment plan instance statuses & results 287
Table 22: Campaign execution statuses 304
Table 23: Campaign execution results 304
Table 24: Edit application templates 322
Table 25: Edit environment templates 327
Table 26: ANT Task APIs: Configuration 335
Table 27: The drops.import.filter element’s attributes 335
Table 28: The drops.instance task attributes 335
Table 29: The drops.import.release task's attributes 336
Table 30: The release.deploy task's attributes 339
Table 31: Configure the DROPS for Jenkins plug-in 364
Table 32: Configure DROPS: Import steps in Jenkins 364
Table 33: Configure DROPS: Deploy steps in Jenkins 366
Table 34: Deployment process instance statuses in Jenkins 369
Table 35: Entity types: Configuration 407
Table 36: Substitution variables 416
Table 37: Import process substitution variables 417
Table 38: Export process substitution variables 419
Table 39: Deployment process substitution variables 420
Table 40: Activity-level substitution variables 424
Table 41: Action-level substitution variables 425
Table 42: Definitions of user roles and the required rights 427
Table 43: Server Administrator and other high-profile rights 430
Table 44: Component extension types 432

Table 45: ARCAD component extended configuration 432
Table 46: Import strategy types: Configuration 433
Table 47: ARCAD strategy extended configuration 435
Table 48: ARCAD Enhanced strategy extended configuration 438
Table 49: Azure DevOps strategy extended configuration 440
Table 50: Command Line strategy extended configuration 441
Table 51: CVS Repository import strategy extended configuration 443
Table 52: Docker strategy extended configuration 445
Table 53: File System strategy extended configuration 446
Table 54: GIT import strategy extended configuration 449
Table 55: IBM i strategy extended configuration 450
Table 56: Maven import strategy extended configuration 451
Table 57: SFTP import strategy extended configuration 452
Table 58: SVN import strategy extended configuration 454
Table 59: Infrastructure item types: Configuration 455
Table 60: ARCAD Runtime infrastructure item extended configuration 455
Table 61: ARCAD Server infrastructure item extended configuration 456
Table 62: Docker infrastructure item extended configuration 456
Table 63: IBM i infrastructure item extended configuration 456
Table 64: JDBC infrastructure item extended configuration 457
Table 65: Tomcat infrastructure item extended configuration 457
Table 66: Environment extension types 458
Table 67: ARCAD environment extended configuration 458
Table 68: Troubleshooting 470

Figures

Figure 1: DROPS in the ARCAD for DevOps product suite 27
Figure 2: DROPS functional diagram 27
Figure 3: The DROPS functional diagram 31
Figure 4: The DROPS workflow 31
Figure 5: Accessing the DROPS Server options 37
Figure 6: Change your password (Preferences menu) 41
Figure 7: The Users search view 44
Figure 8: The User editor 45
Figure 9: The Groups search view 49
Figure 10: The User Groups editor 50
Figure 11: Assigning entity-Level Roles 55
Figure 12: Current user information (Preferences menu) 56
Figure 13: The Configuration Management perspective - artifact repository 58
Figure 14: The application editor 61
Figure 15: The Application editor - components 68
Figure 16: The Component editor 69
Figure 17: The component editor - import strategies 73
Figure 18: The Import Strategy editor 75
Figure 19: The Releases search view 79
Figure 20: The Release editor 81
Figure 21: The Release Branching search view 86
Figure 22: The Artifacts search view 89
Figure 23: The Artifact History view 89
Figure 24: The Configuration Management perspective - Infrastructure Repository 92
Figure 25: Views named for multiple clients 95
Figure 26: The Environments search view 96
Figure 27: The Environment editor 99
Figure 28: The Deployed Releases tab in the environment editor 101
Figure 29: The Deployment History tab in the environment editor 102
Figure 30: The Environment Groups search view 105
Figure 31: The Environment Group editor 106
Figure 32: The Infrastructure Items search view 109
Figure 33: Example of .csv file used to import multiple infrastructure items 112
Figure 34: The Infrastructure Item editor 116
Figure 35: Infrastructure agents included in the Monitoring List 127
Figure 36: The Remote Agent Monitoring List view 127
Figure 37: The SSH Keys view 130
Figure 38: The Deployment Schemas view 133
Figure 39: The Deployment Schema editor 134
Figure 40: Example of a .properties file prepared for replacement rules 137
Figure 41: The Replacement Rules view 137
Figure 42: The Replacement Rules editor 138
Figure 43: The Compliance Rule Repository search view 140
Figure 44: The Compliance Rule editor 141

Figure 45: The Compliance Rule's Ant editor 143
Figure 46: The Compliance Rule Sets search view 145
Figure 47: Executing Compliance Rule Cets 146
Figure 48: The Compliance Reports view 147
Figure 49: The Infrastructure Item Compliance log tab 148
Figure 50: The Compliance Reports search view 148
Figure 51: The Scripts search view 151
Figure 52: The Script editor 153
Figure 53: Edit script content 155
Figure 54: Example of an exported script 158
Figure 55: The Process List 161
Figure 56: The Deployment Process editor 165
Figure 57: Create a new infrastructure item activity 166
Figure 58: Activity templates in the deployment process diagram 168
Figure 59: Create ordered connections between deployment activities 169
Figure 60: Add stages to activities 171
Figure 61: Drag and drop actions into activities or stages 173
Figure 62: Create "wait for" connections between deployment actions 174
Figure 63: The deployment process details editor 176
Figure 64: The deployment process properties editor 177
Figure 65: Global-level properties templates in the deployment process diagram 182
Figure 66: The deployment activity properties editor 183
Figure 67: The deployment action properties editor 188
Figure 68: The Deployment Plans search view 192
Figure 69: The Deployment Plan editor 194
Figure 70: Deployment Plan: Order of deployment 195
Figure 71: Create new groups of applications to deploy 195
Figure 72: Drag and drop applications into the application group container 196
Figure 73: Drag and drop environments into delivery stages 198
Figure 74: Edit deployment plan properties 200
Figure 75: Edit delivery stage properties 202
Figure 76: Edit deployment plan environment properties 204
Figure 77: Edit application group properties 205
Figure 78: The Application templates search view 210
Figure 79: The Application Template editor 210
Figure 80: The Component Types search view 212
Figure 81: The component type editor 213
Figure 82: The import strategy templates view 215
Figure 83: The Deployment Process Templates search view 219
Figure 84: Viewing deployment process that use process templates 221
Figure 85: The Operational Console Management perspective - Importing 242
Figure 86: Importing artifacts by application 245
Figure 87: The Import Process Instance search view 248
Figure 88: Consulting Import execution results in real-time (Console) 253
Figure 89: The Import Results view - Logs 254
Figure 90: The Operational Console Management perspective - Deploying 257
Figure 91: The Deployment Process Instances search view 260

Figure 92: Deployment process instances for release sets 266
Figure 93: Tagged deployment process instances 269
Figure 94: Consulting Deployment execution results in real-time (Console) 272
Figure 95: Overview of the deployment process instance results 277
Figure 96: The deployment process result view - steps 278
Figure 97: The Deployment Results view - Logs 280
Figure 98: The Deployment Result view - attachments 280
Figure 99: The Deployment Result view - workitems 281
Figure 100: The Deployment Plan Instance search view 284
Figure 101: Follow the progress of a deployment plan 286
Figure 102: The deployment plan result view - steps 288
Figure 103: The Interruption Messages view 291
Figure 104: The Interruption Messages editor 292
Figure 105: The pending Suspended Actions view 294
Figure 106: The Suspended Action editor 295
Figure 107: The pending Validation Requests view 296
Figure 108: The Validation Request editor 297
Figure 109: The Operational Console Management Perspective - Campaigns 300
Figure 110: The Campaigns search view 301
Figure 111: The Campaign results view - Logs 305
Figure 112: Connect to the ARCAD development server 312
Figure 113: Connect to target ARCAD servers 313
Figure 114: The Remote Application Template editor 322
Figure 115: The Remote Environment Template editor 327
Figure 116: The ARCAD runtime ANT project editor 329
Figure 117: Manually updating the application layout on a target ARCAD runtime 332
Figure 118: Configuring the DROPS for Jenkins plug-in 363
Figure 119: Jenkins pipeline project: add DROPS steps 367
Figure 120: Jenkins pipeline syntax for DROPS steps 368
Figure 121: The JIRA settings menu: Add-ons 372
Figure 122: Upload external app to JIRA 372
Figure 123: The DROPS for JIRA add-on installed and ready 373
Figure 124: Uninstall the DROPS for JIRA plug-in 374
Figure 125: The JIRA settings menu 374
Figure 126: The DROPS Deployment Request configuration categories 375
Figure 127: Define the connection to the DROPS Server 376
Figure 128: Define the issue type for project dedicated to DROPS 377
Figure 129: Map your JIRA project to a DROPS application 378
Figure 130: Map your workitem provider to JIRA 379
Figure 131: The JIRA settings menu: Issues 380
Figure 132: Create the environment selection field 380
Figure 133: Create the environment selection screen 381
Figure 134: Modify your development project's workflow 382
Figure 135: Add a deployment request transition to your development workflow 383
Figure 136: Add a post function to the deployment transition 384
Figure 137: Create the link type to add to the issue pairs 385
Figure 138: Add an import and deployment execution transition to your DROPS workflow 387

Figure 139: The JIRA settings menu: Issues 388
Figure 140: Add the DROPS custom fields to the deployment request issue's screens 390
Figure 141: The Workitem Providers view 401
Figure 142: The Workitem provider editor 402
Figure 143: The workitem editor 405
Figure 144: The Component Types search view 407
Figure 145: Create environment types 409
Figure 146: Create infrastructure item types 411
Figure 147: The Properties manager 412
Figure 148: The Properties manager 414
Figure 149: Edit the release resolver script 466
Figure 150: Import with the release resolver 467
Figure 151: The Invalid Items view 471

Introduction

1 About DROPS

Automatically or selectively create packages on any environment and "drop" them on designated target environments

Previously, the software development lifecycle revolved around fixed requirements to be implemented by local in-house development teams, and delivered as "big bang" projects. Hundreds of modules were delivered at distant intervals, and then review by stakeholders was performed at the final delivery. This waterfall approach to development resulted in the majority of software projects failing to deliver in terms of cost, timeliness, and end-user satisfaction. Today, IT projects deal with evolving requirements, geographically dispersed development teams, ongoing stakeholder engagement, and incremental delivery. To stay ahead, massive software changes must be delivered with increasing frequency and at higher levels of quality. Agile methods were born to deal with these needs. Agile development allows organizations to react more quickly and deal with ever-changing requirements, while assuring quality with fewer resources.

Yet as Agile shortens development cycles, operations teams have struggled to keep up. Faster delivery increases the risk in each software release, as multiple teams push "change sets" in interdependent systems. The goal of operations – to keep applications available – conflicts with development that is measured on the speed and reactivity of their upgrades. To balance these conflicting interests, "dev" and "ops" need common tools to cooperate closely and become more proficient in rolling changes into production. For this reason, both agile methods and, in turn, the DevOps lifecycle are now central to Release Management, reconciling pragmatism, accountability, and a "need for speed".

At the same time, the introduction of Web and mobile components has made application architectures more complex. Mobile, cloud and Web-based interfaces make new demands concerning Release Management tools every year. The changes impact the downstream phases of software delivery and deployment, and in particular the boundary between development and operations departments. Information systems span multiple platforms, with often a legacy business application at the core. Release Management tooling must therefore manage the dependencies between diverse technologies and facilitate collaboration between disparate teams.

DROPS offers a unique solution for the management and automation of application and systems software deployment across multiple platforms - including IBM i, Windows, UNIX, Linux, and System z. In support of DevOps, DROPS is production-oriented, shifting the bias from development to operations.. DROPS focuses on the deployment of functions, rather than on technical components. This makes the solution intuitive and easy-to-use, assisting development and operations teams in working closely together to smoothly manage an accelerated delivery schedule. DROPS also supports continuous integration and delivery, so teams can use agile methods efficiently and deliver value to customers faster. DROPS deploys by process, to maximize automation and give real-time transparency on release status for all parties.

The real power of DROPS is derived from its comprehensive management capabilities, automation, and the diverse technical platforms it supports. The deployed software on open systems, IBM i and z/OS platforms can be traced at any time enabling Operations Managers to closely watch inflow of systems and application software changes at all times.

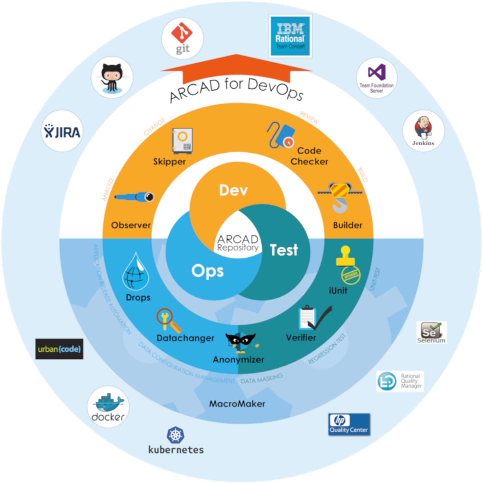


Figure 1: DROPS in the ARCAD for DevOps product suite

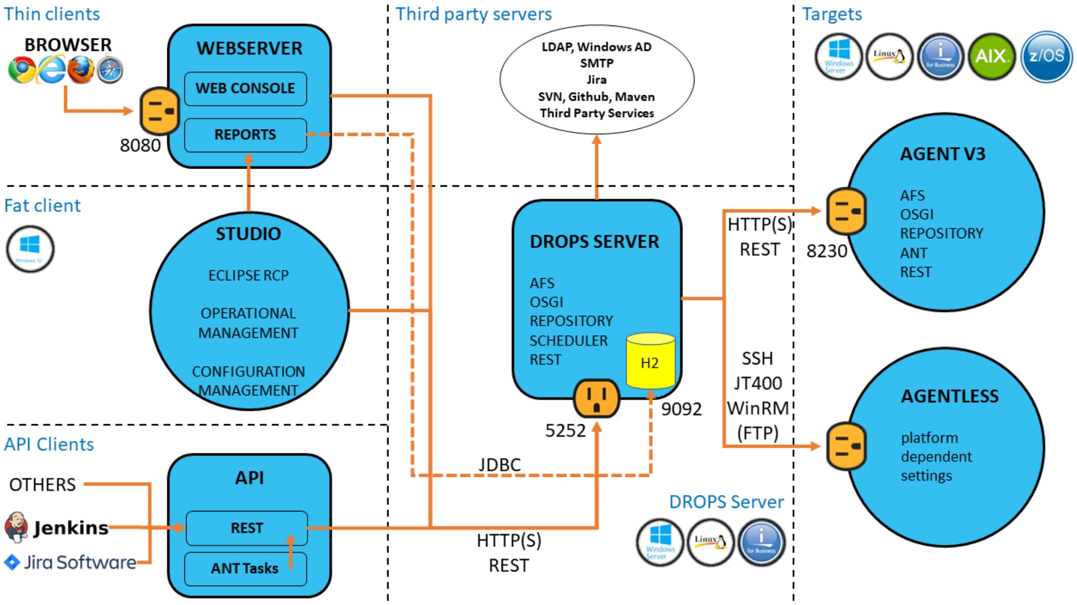


Figure 2: DROPS functional diagram

1.1 The DROPS Server

Host your deployment processes

The DROPS Server is an independent AFS server on which DROPS creates and stores specialized repositories to be used in your deployment process. It is an OSGi™ application server which hosts your references on an H2 database by default. The server's default port number is 5254.

It exploits REST web services in order to communicate with other applications in the system, for instance to create a continuous deployment process with Jenkins, and can be installed on Linux, Windows and IBM i. A scheduler is incorporated for planning and orchestrating the deployments.

The DROPS Server conforms to your company's security policies, authentication and delegation mechanisms and to your method(s) of assigning roles. It delegates process execution to the DROPS Agent which actually performs the action(s) requested on your target machine(s) while respecting the project architecture you describe.

The size limit of the DROPS Server depends solely on the size of your system's hard drive or the space available on the system.

The DROPS Studio, the DROPS web app and client APIs connect directly to the DROPS Server via http (s). The DROPS Server connects to the DROPS Agent via http(s). These connections enable you to execute any or all of the various import and deployment functions DROPS offers.

Reference

For more information about installing the DROPS Server, refer to the *DROPS Installation Guide*.

For more information about configuring the DROPS Server, refer to the *DROPS Configuration Guide*.

1.2 The DROPS Studio

Design your deployment processes

Created for Administrators, Release Managers, Deployment Process Designers, Deployment Operators and anyone else included in DevOps, the DROPS Studio simplifies the configuration and execution of the import and deployment processes involved in application deployment, all from a single, powerful yet easy-to-learn Eclipse RCP.

The DROPS Studio breaks the deployment process up into three phases: configuration, import and deployment. Roles and tasks are carefully defined for each phase and the powerful scheduler can automate your validated processes.

For reporting, the DROPS Studio relies on the complementary ARCAD product: ARCAD Dashboard.

Reference

For more information about installing the DROPS Studio, refer to the *DROPS Installation Guide*.

For more information about configuring the DROPS Studio, refer to the *DROPS Configuration Guide*.

1.3 The DROPS Agent

Connect to your target system(s)

DROPS is multi-platform by design which means it enables you to deploy content to any environment that can host a JRE. The DROPS Agent is a module that must be installed on every target system to which you intend to deploy that is incompatible with the agentless solution. It contains and relies on the Ant engine which actively contributes to the mechanical orchestration of the DROPS solution. It includes a library with over 300 built-in scripts which can be expanded with your own personalized scripts. The DROPS Agent is the remote administration engine used by the DROPS Server and from where all the deployment actions are started. In other words the DROPS Server orchestrates all deployment actions through the DROPS Agent and by using the capabilities of the agent on the target system, it also means that the server's action field on the target system depends the agent's process owner on this target system.

The DROPS Server connects to the DROPS Agent via http(s). The connection is defined in the infrastructure item associated with each agent. During deployment, the agent receives instructions from the DROPS Server and executes them on the target system.

The DROPS Agent can be remotely or manually installed and uninstalled.

Reference

For more information about installing the DROPS Agent, refer to the *DROPS Installation Guide*.

1.4 The DROPS web app

Execute your deployments via the web

The DROPS web app is a web application that runs on the DROPS Web Server by default but can also be installed manually on any Web Application Container. It is an RIA intended specifically for Deployment Operators which makes executing deployment processes easy. Operators can schedule sequential tasks and execute simultaneously or according to personalized procedures. Depending on the context, a release can be deployed individually, or in groups by creating cumulative releases. At any time and in real-time, an operator can execute and monitor an execution process as well as view deployment progress or history by using any internet browser.

To access the DROPS Web Server enter the following address in your browser:

<web_server_address>:*<web_server_port>*/drops

The DROPS web app is accessible from anywhere, from any web browser.

Note

The DROPS web app does not include access to the **Configuration Management** perspective. It only enables you to perform the actions available in the **Operational Console Management** perspective.

Important!

For the DROPS web app to function properly in case of an update, the following properties must be added in the **config.ini** file, located in the

configuration folder:

- The *com.arcadsoftware.csp.style.sources* parameter must be set to **'self' 'unsafe-inline'**.
- The *com.arcadsoftware.csp.image.sources* parameter must be set to **'self' data:**.

Reference
For more information about installing the DROPS web app, refer to the *DROPS Installation Guide*.

The web server is a Web Application Container (Jetty) (*IBM i, Windows and Linux*), which is a J2EE Application Server. It grants access to the web application. The web server can be added to an existing internal web server you may have, such as Tomcat, or you can use the lightweight jetty that DROPS Software delivers to install autonomously.If you already have a Web Application Container running on the machine (a Jetty or Tomcat), it is not necessary to install the DROPS Web Server to access the DROPS web app. You can install the *drops.war* for the DROPS web app manually using your current Web Application Container. To do this, copy the *.war* into the folder *C:\Program Files\ARCAD Solutions\Jetty\webapps*.

2 The DROPS architecture & workflow

Figure 3: The DROPS functional diagram

The DROPS architecture is based on the DROPS Server, which is a Java application that offers a REST interface on port 5254, by default. The various *clients* (fat client, thin client, APIs, etc.) communicate with the server based on the REST interface.

The DROPS Server drives deployments through the DROPS Agent, installed on the target servers. The agents are accessed either through a REST interface on the default port 8230 if they are installed manually, or via SSH when using the "agentless" solution (no installation on the target server that is reachable via ssh).

Deployments are made in 3 phases in DROPS:

1. Declaring the applications and provisioning of the releases to deploy.
2. Declaring the technical and logical targets.
3. Orchestrating the deployment of the releases to the target (logical) environments.

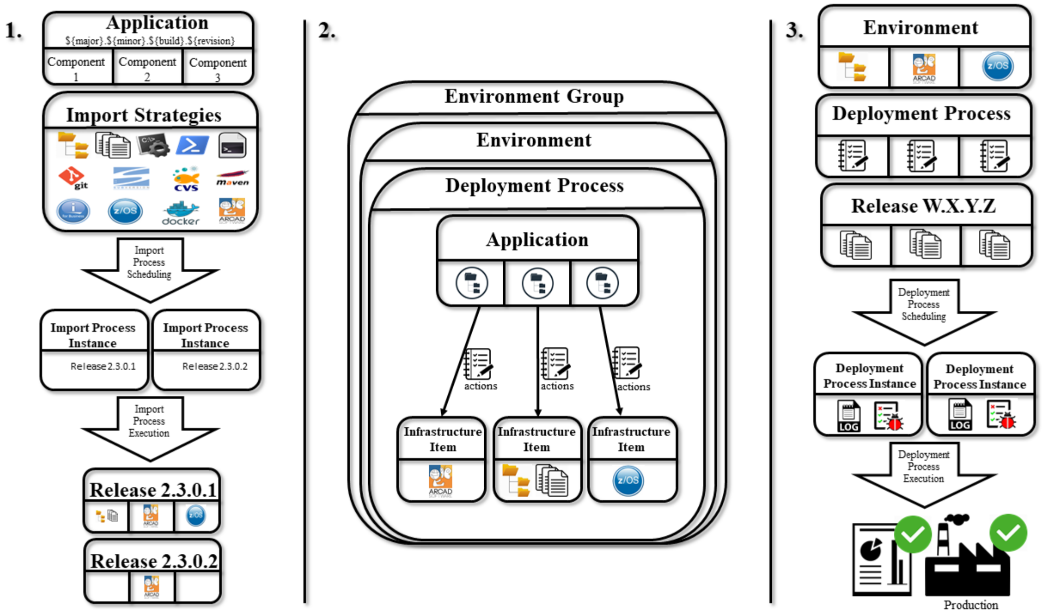


Figure 4: The DROPS workflow

Provisioning the releases to deploy

The administrator, or users with the *release configuration* roles, must first declare the **applications** that are a logical set of **components**.

For each component a set of **import strategies** is defined in order to collect or extract the different artifacts included in the component subset of the application. An **import instance** is the selective execution of the different import strategies of the components of the application.

The result of the execution of an import instance is the creation or the update of a **release**, which is the basic element that will be deployed by DROPS during the last phase.

Defining technical and logical targets

The administrator, or the users with the *environment configuration* roles, must first declare the **infrastructures items**, which are the technical targets (DB, application servers, file systems, third systems, etc.) to which will be deployed the artifacts of the different components of the application.

The infrastructure items are grouped into **environments** that are the logical targets (UAT, TEST, PRE-PROD, PROD, etc.) for deployment during the last phase. It is possible to group environments into **environment groups** which allows simultaneous deployment on a set of environments.

For each environment, **deployment processes** are published for the applications allowed in these environments. A deployment process is a deployment scheme in which components of the application are published to an infrastructure item using a list of actions to be executed. An environment can contain several deployment schemes for the same application, the latter implementing all or part of the available infrastructure items and components using specific action lists. Actions can interact with artifacts, technical targets, host systems and their file systems, and also third-party systems (via APIs, interfaces, etc.).

Orchestrating deployments

The last phase of operations is the implementation and orchestration of deployments. This is to select a logical target and an available deployment scheme for that target that therefore selects an application. For this application the selection of the release to deploy completes the creation of a **deployment instance** which is the actual deployment process on the logical target. The deployment instance is then executed which results in the deployment of the artifacts to the target system(s). The execution also produces statistics that can be retrieved in reports.

3 Main concepts

A multi-platform system automating continuous application deployment

Reference

Refer to the Glossary on page 475 for complete definitions of the terms below.

DROPS is intended to help you set up and automate the deployment process. To do this, DROPS uses dedicated repositories secured in the DROPS Server to organize your file structure and shape and store your releases.

Releases to deploy to target environments contain artifacts from the artifact repository. A release is created each time a pre-defined import strategy is launched and artifacts from your source location are copied to the artifact repository. The repository stores artifacts using the following structure: application/component/artifact.

Once the import is complete, the release is available to be deployed. Define infrastructure items to structure your deployment environment and connect to your target machine(s). Deploying to multiple machines is made easy by grouping infrastructure items in environments and organizing environments by client.

The deployment process is organized using a completely customizable deployment process diagram supplemented with the actions necessary for your deployment, all driven by ANT scripts. Applications and their components are partnered with environments and their individual infrastructure items in the deployment process diagram in order to make sure the correct content is always making it to its intended destination(s).

Multiple deployment processes can be run together via deployment plans which means you can also automate the deployment of any number of application-to-environment pairs in the same multi-destination process.

Scheduling imports and deployments makes it easy to create completely autonomous processes.

Checks and balances are taken care of in DROPS via validation requests and suspended actions which automatically pause processes if authorization is required or an error occurs.

DROPS enables you to track deployment stages in real time. In addition to providing full deployment logs, it keeps a history of all actions, executions, and transition times.

The option to roll back is always available. Before deployment, actions automatically register a copy of the existing release on your target machine(s) and the infrastructure item backs it up to a rollback directory.

Certain functions are common to all DROPS processes. Actors are defined for each configuration, together with their role(s) in the approval process. Your company's processes, roles and access rights, approval and validation requirements, scheduling, and reporting are all taken into account and can be managed inside DROPS.

Following the deployment process, DROPS is able to manage the installation of delivered software.

4 Overview of the DROPS Studio

DROPS is an independent Eclipse studio. There are two perspectives dedicated to DROPS that contain all the features available in the tool. Each is intended for different types of users.

A perspective defines the initial set and layout of views in the workbench window. Within the window, each perspective shares the same set of editors. Each perspective provides a set of capabilities aimed at accomplishing a specific type of task or working with specific types of resources. For example, the JavaTM perspective combines views that you would commonly use while editing Java source files, while the Debug perspective contains views that you would use while debugging a program. Perspectives contain views and editors and control what appears in certain menus and tool bars.

Views in DROPS are context sensitive. To see a view that is not currently active, open it by accessing the **Show View** dialog (*Window > Show view > Other*) and selecting the view. Re-position views by dragging and dropping them to the desired location in the studio. Click an edge and drag to re-size a view. Double-click on any view's toolbar to maximize it. This is helpful when a view contains a lot of information; maximizing the view makes it fill the entire window. Double-click again to minimize the view.

To re-set the perspective to the default layout, click *Window > Reset Perspective*.

The Configuration Management perspective

The Configuration Management perspective is intended for Release Managers and both Import and Deployment Process Designers because it provides access to all of the actions necessary to organize artifacts, to construct the many entities required for the import and deployment processes. Additionally, Reports are generated and entity-level configuration is managed inside this perspective.

Note

The DROPS web app does not include access to the **Configuration Management** perspective. It only enables you to perform the actions available in the **Operational Console Management** perspective.

Reference

For more information about this perspective, refer to the Introduction to user administration on page 43, Introduction to configuring the import process on page 58, Introduction to configuring the deployment process on page 92 .

The Operational Console Management perspective

The Operational Console Management perspective is intended for Release Managers and both Import and Deployment Operators because it provides access to all of the actions necessary to create, schedule, launch and monitor import and deployment processes.

Reference

For more information about this perspective, refer to the Introduction to importing artifacts on page 242 and the Introduction to deploying releases on page 257.

5 Functional constraints

Unattended actions

Like all Release Management software, DROPS cannot complete operations that contain manual manipulations necessarily carried out in a GUI or front-end terminal.

Example

- entering content in mandatory fields that have no default value;
- clicking buttons in an installation wizard.

However, DROPS provides solutions if your process contains manual actions in order to automate the majority of your deployment process seamlessly.

Getting Started

6 Managing the DROPS Server

Required Roles	n/a
Required Rights (by category)	Server Administration
Perspective	Configuration Management and/or Operational Console Management
Access	Connections

S

In order to use and manage the DROPS Server and access DROPS functions, a connection must be established between it and the DROPS Studio. Connections must be defined in both of the perspectives available in the DROPS Studio and in the DROPS web app.

Define a connection by first adding a DROPS Server to the DROPS Studio then connecting to it using the login credentials specific to it.

To access the DROPS Server options, right-click on it.

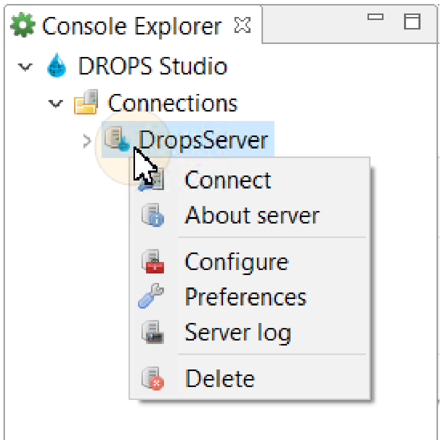


Figure 5: Accessing the DROPS Server options

6.1 Creating a server connection

Follow the subsequent steps to create the connection to the DROPS Server.

Step 1 Right-click on **Connections** and select **Create a server connection**.

Step 2 Define the server connection.

1. Enter a unique name for the server connection in the **Connection** field.
2. Enter the name or **IP address** of the server.
3. Enter the **Port** on which the server is listening. The port is defined when the server is installed.

4. Select whether or not to use a secure connection (**Use HTTPS**) or not.
5. Click **Finish**.

Result The server connection is created. New server connections can be accessed and managed in the **Connections** list.

> **Note**
> To connect to the server using a proxy, define the proxy parameter in the general network connection preferences of the eclipse platform (Preferences > General > Network Connections).

6.2 Connecting to a DROPS Server

Follow the subsequent steps to connect to a DROPS Server.

Step 1 To access the connection splash screen, either double-click on the DROPS Server, or right-click and select **Connect**.

Step 2 Each user has his own login/password. The admin-level user/password is: **admin**/**quadra**.

Step 3 Click **Connect** to connect to the server.

Result The DROPS Server connection is established and you have access to all the elements allowed for your user profile.

6.3 Configuring the TLS settings

> **Important!**
> This section does not describe the configuration required for TLS in the DROPS Server. The TLS configuration belongs to the AFS configuration layer. Refer to the DROPS Server Configuration Guide.

For security reasons, you may need to use an encrypted connection between the server and the studio. To do this you must activate the use of the HTTPS protocol, by checking the **Use HTTPS** option in the server connection parameters. However, this protocol, based on the TLS network protection layer, must be correctly configured to be implemented. Depending on the server settings, the client program will have to validate the server's TLS certificate and potentially use its own key.

In most cases, only the validation of the server certificate is required. If this certificate is the product of a chain of trust whose root is a recognized Certification Authority (CA) then there is nothing special to do on the client side. The Java virtual machine already recognizes and approves certificates from all the parties in the market. However, if the server's certificate was not issued by a CA, or your organization uses a specific CA, it is necessary to add the certificate of this CA, or directly the server's certificate in the studio configuration.

It is also possible that the server has been configured in such a way that the studio must also use a key pair (private, public) that requires the server to validate the studio's certificate.

In case the server uses a certificate that is not issued by a CA or requires the studio to use a key pair, the administrator will pass these materials to you. The most secure way to do this is to send them to you in a container, called a KeyStore, which is password protected. In case the HTTPS connection requires both the validation of the server's certificate and a key pair for the studio, it is possible that the administrator will send you two stores, one containing the certificate to be approved, this is called a TrustStore, the other containing only the studio's key pair. This last store keeps the name KeyStore.

It is also possible that the administrator transmits the certificate directly to you, without integrating it into a store, in this case make sure that the source of the transmission is not spoofed. Integrating the wrong trusted certificate can jeopardize the security of the entire installation.

> **Note**
>
> Certificates and key pairs have a limited validity period. This period varies, generally from 1 to 12 months. The following configuration procedure must therefore be repeated each time the equipment is updated. Knowing that an expired certificate or key pair will block the connection to the server.

Once you have the KeyStore(s) and their respective passwords, or, if not available, the server certificate, follow the steps below to redefine the HTTPS protocol settings:

Step 1 To access the **TLS Settings** dialog, right-click on the **DROPS** node and select **TLS Settings**.

Step 2 Edit the absolute paths to the **TrustStore** and **KeyStore** and their corresponding passwords.

Step 3 Click the **Import Certificate** button to import your own certificates into the file indicated in the **TrustStore** path.

You can click the **Import Certificate** button to import the server's certificate if it was sent to you as is. It will be included in the TrustStore defined in this window.

Step 4 Click **OK** to save your changes or **Cancel** to keep the default settings.

> **Note**
>
> The **Reset** button resets access information and passwords to the default values provided by the DROPS Studio.

6.4 Defining DROPS Server preferences

To manage the DROPS Server preferences, right-click on it in the **Connections** list then select **Preferences**.

> ***Reference***
>
> For more information about configuring the DROPS Server, refer to the *DROPS Configuration Guide*.

6.5 Configuring a DROPS Server

To configure the DROPS Server, right-click on it in the **Connections** list then select **Configure**.

> ***Reference***
>
> For more information about configuring the DROPS Server, refer to the *DROPS Configuration Guide*.

6.6 Viewing DROPS Server logs

To view a server's log, right-click on it in the **Connections** list then select **Server Log**.

6.7 Verifying DROPS Server versions

To verify the version of a DROPS Server, right-click on it in the **Connections** list then select **Server Information**. A dialog displays the selected server's version and the user with which you are currently connected.

6.8 Deleting a DROPS Server connection

To delete a connection to a DROPS Server, right-click on it in the **Connections** list then select **Delete**.

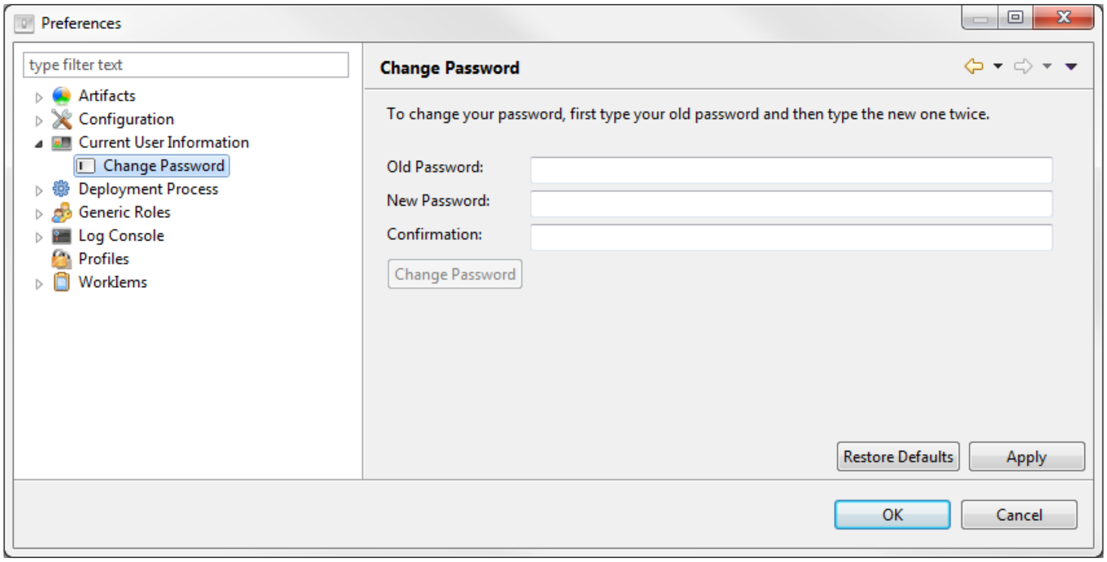
Deleting a server connection does not permanently affect the DROPS Server. You can reconnect to any server by following the steps outlined above.

7 Changing your password

Required Roles	n/a
Required Rights (by category)	Server Administration
Perspective	Configuration Management and/or Operational Console Management
Access	Connections → *right click on the* DROPS Server

To change your user's password, right-click on the active DROPS Server and select **Preferences**.

Open the **Current User Information** node in the **Preferences** menu and select **Change Password**.



To change your password, first type your old password and then type the new one twice.

Old Password:
New Password:
Confirmation:

Figure 6: Change your password (Preferences menu)

Click **Apply** to push the changes live without closing the **Preferences** menu.

Reference

For more information about users, refer to User accounts on page 44.

drops

User Administration

Introduction to user administration

DROPS administrators are in charge of managing users, roles and rights. It is important to define your team and associate the correct roles and rights with each user, because the individuals who use DROPS at different stages in the process must have accurate authority levels in order to access and, in some cases, create and/or edit the various processes.

User information is stored locally in DROPS and the login credentials for each user are managed by an administrator or any user with the corresponding rights. Any user can change his or her own password (see Changing your password on page 41).

This section covers all of an administrator's tasks associated with:

- Users (see User accounts on page 44),
- Roles (see Assigning entity-level roles on page 54),
- Profiles (see Accessing your profile and rights on page 56).

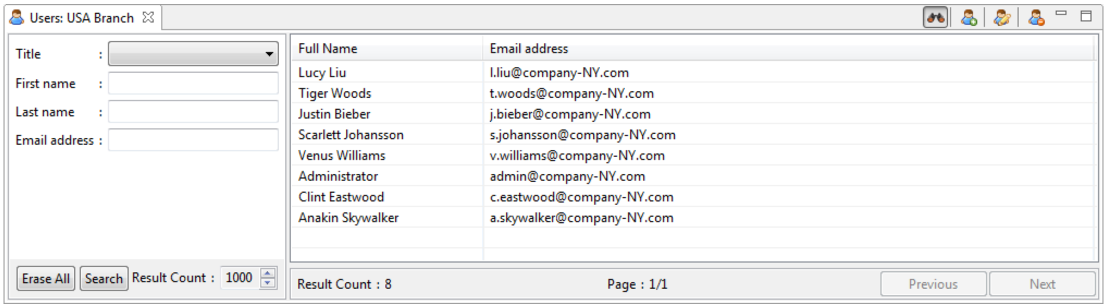
8 User accounts

Required Roles	n/a
Required Rights (by category)	User Management
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) Users

Chapter Summary

8.1 Creating users 44
8.2 Editing a user's ID 45
8.3 Adding users to groups from the editor 47
8.4 Assigning profiles to users 47
8.5 Deleting users 48

User accounts are what define the access rights for your team. Each account is associated with one email address and has a unique login. Users are assigned roles for the import and deployment processes and given rights to access certain functions.



Full Name	Email address
Lucy Liu	l.liu@company-NY.com
Tiger Woods	t.woods@company-NY.com
Justin Bieber	j.bieber@company-NY.com
Scarlett Johansson	s.johansson@company-NY.com
Venus Williams	v.williams@company-NY.com
Administrator	admin@company-NY.com
Clint Eastwood	c.eastwood@company-NY.com
Anakin Skywalker	a.skywalker@company-NY.com

Result Count : 8 Page : 1/1

Figure 7: The Users search view

The **Users** search view is accessed from the **Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

> ***Reference***
> For more information about managing multiple clients, refer to Multiple clients on page 94.

8.1 Creating users

User information can either be stored and completely managed in the DROPS Server, or imported from an external LDAP which manages login credentials.

8.1.1 Create new users in the DROPS Studio

Follow the subsequent steps to create a new user.

Step 1 To access the **Create a new user** wizard, either click the create icon in the **Users** search view or right-click anywhere in the view and select **Create a new user**.

Step 2 Complete at least one of the fields in order to create the account.

Step 3 Click **Finish**.

Result The new account is created and the **User** editor opens automatically.

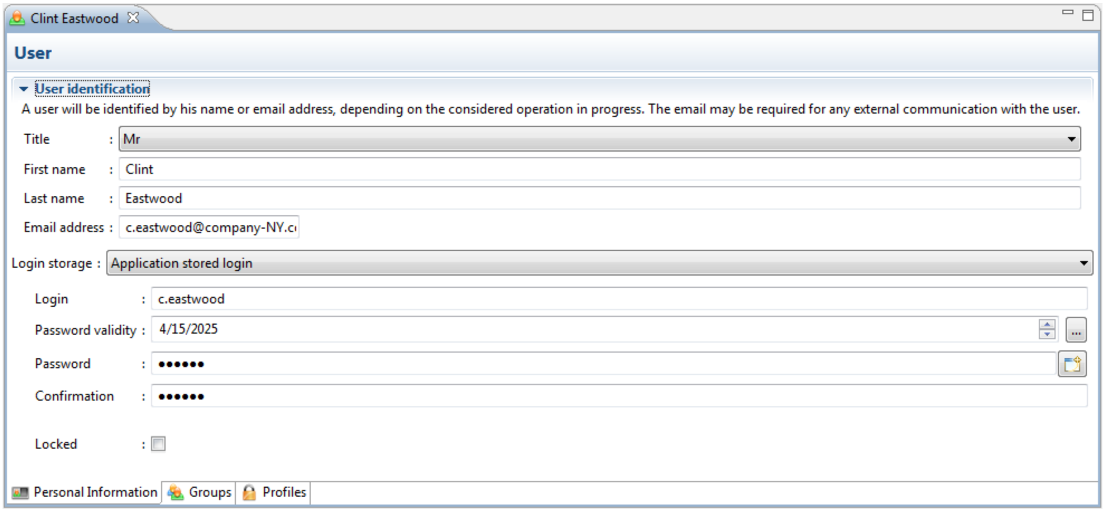
> **Important!**
> It is important to continue editing the new user's information right away to complete the user configuration and assign a login and password.

8.1.2 Import new users from an external LDAP (Active Directory)

Importing users from an external LDAP is done from the **User Import** view.

> ***Reference***
> Refer Importing LDAP Users on page 52.

8.2 Editing a user's ID



A user will be identified by his name or email address, depending on the considered operation in progress. The email may be required for any external communication with the user.

Title : Mr
First name : Clint
Last name : Eastwood
Email address : c.eastwood@company-NY.c
Login storage : Application stored login
Login : c.eastwood
Password validity : 4/15/2025
Password : ••••••
Confirmation : ••••••
Locked : □

Figure 8: The User editor

To view or edit a user's information, connection details or profile(s), either right-click on the user then select **Edit**, select the user then click the edit icon or double-click. The **User** editor is opened where all of its details are managed.

⚠ **Important!**
If users were imported via LDAP (Active Directory), their details are not managed in DROPS. There is nothing to configure or edit for imported users

Edit the details displayed in the **Personal Information** tab of the **User** editor by making changes in the fields necessary.

Changing the user's ID (display name and the email address used when sending automatic emails) are simple edits.

Save the changes (💾, Ctrl+S or **File** > **Save**).

8.2.1 Define a user's login credentials

If users are stored in the internal database, and not an external LDAP, their DROPS log-in credentials must be defined.

⚠ **Important!**
If users were imported via LDAP (Active Directory), their login and password are not managed in DROPS.

Select **Application stored login** from the **Login storage** drop-down list, then complete the necessary fields.

Login

The user name to enter when accessing DROPS.

ⓘ **Note**
Spaces cannot be used in user names.

Password validity

Select the date the user's login password will expire. Entering an expired date forces the user to change their password the first time they log in.

ⓘ **Note**
The server can impose a certain level of password complexity depending on its settings. By default, a password of at least 8 characters long, and containing at least one lower case, one upper case, one number and one symbol is required. These rules can be extended by the server administrator. Refer to the DROPS Server Configuration Guide.

Password

Either enter a password for the user or click the random password button to generate a random password. A confirmation window displays the random password generated by the system.

⚠ **Important!**
It is important to write down what it is before clicking **Yes** because after you accept a random password there is no way to recover it.

⚠ Any user can change his password after they access their account.

> **Note**
> The password field is encrypted. It will always display 6 bullets, no matter what the length of the real password is.

Confirmation

Re-enter the password chosen. The confirmation is automatically filled in for random passwords.

Locked

If this checkbox is ticked, the user cannot access DROPS. A user's account is locked automatically if the user tries to access DROPS too many times using the incorrect password. The administrator can lock accounts manually by ticking the checkbox.

Save the changes (, Ctrl+S or **File** > **Save**).

8.3 Adding users to groups from the editor

Grouping users together enables you to manage rights for multiple users at the same time. Profiles are assigned to groups just like they can be assigned to individual users. A user can be part of multiple groups.

Select the **Groups** tab of the **User** editor to view and manage the user's group(s).

To add a user to a group, click **Add** and select the group(s) from the list, then click **OK**. The list of groups is prepopulated with all of the groups available (for the current client, if applicable).

> **Note**
> User groups are created and managed in the **User Groups** editor. Users can also be added to groups in this editor.

To remove a user from a group, select the group, then click **Remove**.

8.4 Assigning profiles to users

At least one profile is required for every user because they grant the rights required to manipulate entities or items. If a user has no profile, they have no rights to access, create or edit anything.

Select the **Profiles** tab of the **User** editor to view and manage the user's profile(s).

To assign a profile to a user, click the **Add** button and select the profile from the list, then click **OK**. The list of profiles is prepopulated with the profiles already available.

To remove a user's profile, select it then click **Remove**.

> ***Reference***
>
> For more information about profiles, refer to Accessing your profile and rights on page 56.
>
> For more information about creating profiles and managing the rights for each one, refer to the *DROPS Configuration Guide*.

8.4.1 View the rights granted to profile(s)

Follow the subsequent steps to view the rights granted to a profile.

Step 1 Open the **Preferences** menu (right-click on the current DROPS Server → **Preferences**).

Step 2 Click **Profiles** to view the rights granted to a specific profile.

Step 3 Select the profile from the list and click **Rights**.

8.5 Deleting users

> ***Warning!***
>
> Deleted users cannot be accessed or recovered. If a deleted user was given a role, the role is no longer linked to that account.

To delete a user, search for it in the **Users** search view. Either right-click on it and select **Delete** or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the user.

9 User groups

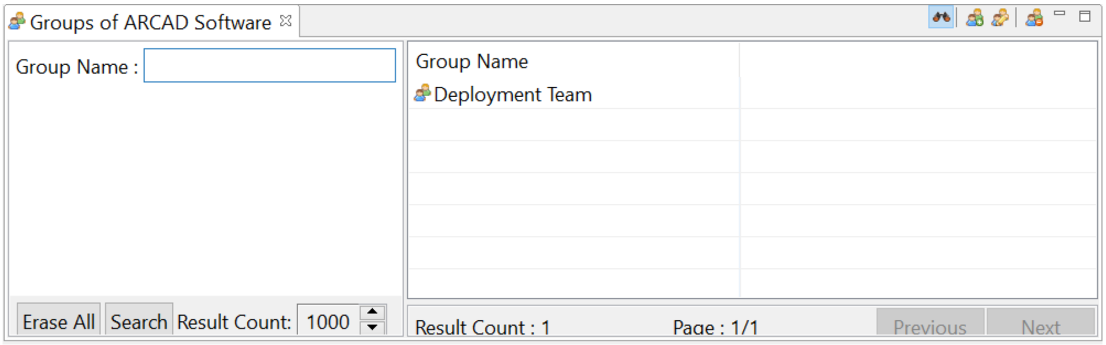
Required Roles	n/a
Required Rights (by category)	User Management
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) User Groups

Chapter Summary

9.1 Creating user groups 49
9.2 Adding users to groups from the editor 50
9.3 Deleting user groups 50

User groups are made up of two or more pre-defined users. groups can be assigned entity-level roles for the import and deployment processes just like individual users. If a group is selected for a role, all of the members of the group are given the same responsibilities.

Grouping users together makes it easy to manage rights for multiple users at the same time. Profiles are assigned to groups just like they can be assigned to individual users. A user can be part of multiple groups.



Group Name
Deployment Team

Result Count : 1   Page : 1/1

Figure 9: The Groups search view

The **User Groups** search view is accessed from the **Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

> ***Reference***
> For more information about managing multiple clients, refer to Multiple clients on page 94.

9.1 Creating user groups

Follow the subsequent steps to create a new user group.

Step 1 To access the **Create a new User Group** wizard, either click the create icon in the **Groups** search view or right-click anywhere in the view and select **Create a new User Group**.

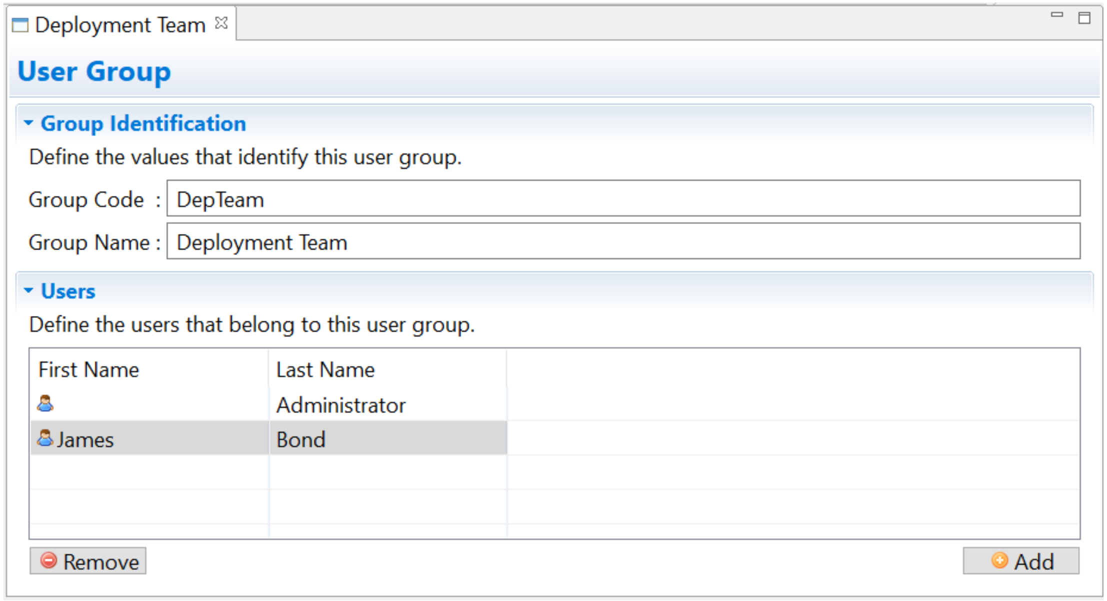
Step 2 Complete at least one of the fields. Click **Finish**.

Result The new user group is created and the **User Group** editor opens automatically.

To view or edit a user group's details or the users that belong to it, either right-click on the group then select **Edit**, select it then click the Edit icon or double-click it. The group's **Code** and **Name** are editable in the he **User Group** editor. Save the changes (, Ctrl+S or **File** > **Save**).

9.2 Adding users to groups from the editor

Note
Users can also be added to groups in the **User** editor.


Deployment Team
User Group
Group Identification
Define the values that identify this user group.
Users
Define the users that belong to this user group.
Remove
Add

Group Code : DepTeam

Group Name : Deployment Team

First Name	Last Name
	Administrator
James	Bond

Figure 10: The User Groups editor

To add a user to a group, click the **Add** button and select the user(s) from the list, then click **OK**. The list of users is prepopulated with all of the accounts (for the current client, if applicable).

To remove a user from a group, select it and click **Remove**.

9.3 Deleting user groups

Warning!
If deleted, a user group cannot be accessed or recovered. If a deleted group was given a role, the role is no longer linked to it.

To delete a user group, search for it in the **Users Group** search view. Either right-click on it, then select ⊗ **Delete** or select it, then click the ⊗ Delete icon. Click **OK** to confirm or click **Cancel** to keep the group.

10 Importing LDAP Users

From the **Import User** view, you can import your users in batch from an LDAP login source, and keep users synchronized. Furthermore, you can assign the DROPS profiles to the imported users immediately from this view.

Prerequisites

Before you start using the LDAP import option to manage users, make sure that:

- The user connected to DROPS is an LDAP user.
- The LDAP access is configured on your server.
- The LDAP Import Users feature is set to **true** in the **osgi.cm.ini** file in the **configuration** folder located in the DROPS Server repository, as follows:

```
[ REST Connection LDAP ]

[...]
userimportenabled = true
```

> **Note**
> Stop the DROPS Server to edit and save the changes made in the configuration file.

It is possible to include additional parameters to manage the LDAP operation with a predefined administrator:

- **search.user.dn**: sets the distinguished name of the LDAP administrator,

- **search.user.password**: sets the password corresponding to the **search.user.dn** parameter. The password is automatically encrypted at the first start of the server.

> **Important!**
> Remove the **dn.login.pattern** property if you define these two parameters.

When the connection to DROPS is done with an LDAP user, the **Import Users** option is displayed under the **Configuration** node in the **Configuration Explorer** panel.

Follow the subsequent steps to import users.

Step 1 Open the Import **Users** view under the **Configuration** node.

The **Import Users**view opens.

Step 2 Tick **LDAP** to select the login source from which to import the users.

> **Important!**
> The login source selection boxes are not displayed if the login sources are not already configured.

Step 3 Set the client of the imported user from the drop-down list in the **Client** option.

Step 4 Select the target profile in the drop-down list. The list shows all the profiles created in the DROPS Server.

Step 5 Define the LDAP **User Filter** option to retrieve the list of users that can be imported, then click on the **Search** button.

> **Note**
> The way this filter is defined depends on your LDAP, but this value is passed without transformation to the target LDAP.

The filtered results appear as a list.

Step 6 Check the box corresponding to the LDAP user you want to import, then click **Import**.

Result The user(s) is/are imported and displayed in the 👤 **Users** view.

11 Assigning entity-level roles

Required Roles	The user must have the Administrator role for the related entity or for all instances of that entity type (global-level user admin)
Required Rights (by category)	User Management
Perspective	Configuration Management
Access	DROPS Server → Artifacts → Applications DROPS Server → Artifacts → Apps → (*App*) → Releases DROPS Server → Configuration (→ Client) → Infrastructure Items DROPS Server → Configuration (→ Client) → Environments DROPS Server → Configuration (→ Client) → Environment Groups DROPS Server → Configuration → (*Client* →) Deployment Plans Preferences → Configuration → Environment Types

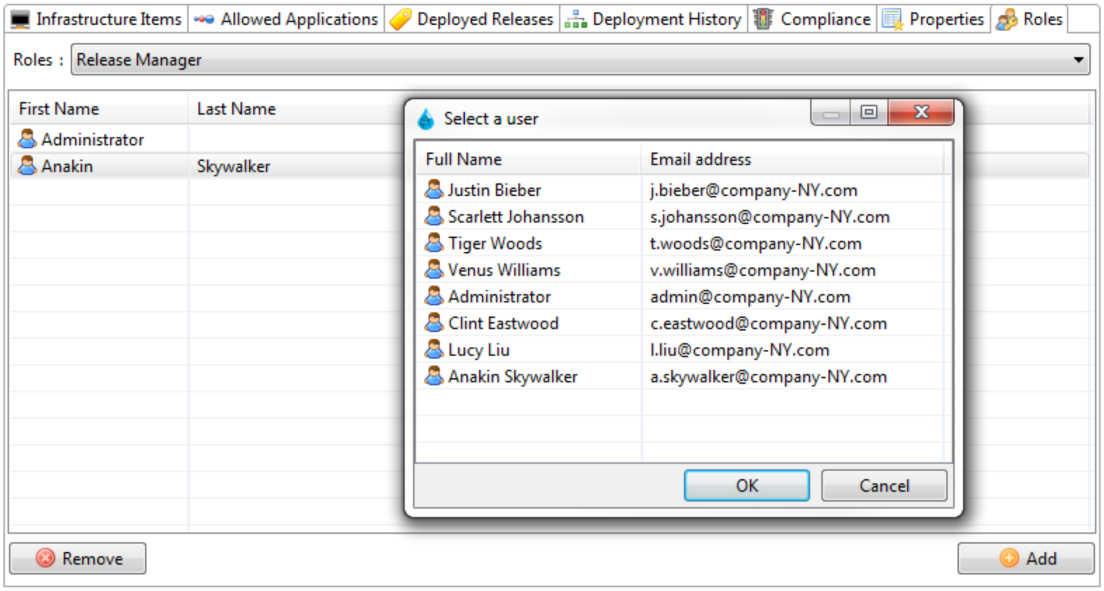
Assigning users and user groups roles enables an administrator to manage who is authorized to execute certain procedures during the import or deployment process or who can assign or carry out a task.

DROPS users are assigned roles on the entity-and global-level. Assigning roles to users at the entity-level is done by the administrator or the user who already has the admin role for that entity.

> ***Reference***
>
> For more information about the different roles available for each entity, refer to Roles in the appendix.
>
> For more information about assigning roles at the global-level, refer to the *DROPS Configuration Guide*.

The **Application**, **Release**, **Infrastructure Item**, **Environment**, and **Environment Group** and **Deployment Plan** editors each contain a **Roles** tab.

> **Note**
>
> Environment types also require roles but they are only defined at the global-level.



Roles : Release Manager

First Name	Last Name
Administrator	
Anakin	Skywalker

Full Name	Email address
Justin Bieber	j.bieber@company-NY.com
Scarlett Johansson	s.johansson@company-NY.com
Tiger Woods	t.woods@company-NY.com
Venus Williams	v.williams@company-NY.com
Administrator	admin@company-NY.com
Clint Eastwood	c.eastwood@company-NY.com
Lucy Liu	l.liu@company-NY.com
Anakin Skywalker	a.skywalker@company-NY.com

Figure 11: Assigning entity-Level Roles

Follow the subsequent steps to assign a user to a role at the entity-level.

Step 1 In the entity's **Roles** manager, select a role from the drop-down list.

Step 2 Click the **Add** button to display the **Select a user** dialog. This dialog is prepopulated with all of the users defined for the current client.

Step 3 Select one or more users from the list (Ctrl+click for multi-selection), then click **OK**.

Result The user(s) is/are assigned the role.

To remove a user from his assigned role at the entity-level, select the role the user is currently assigned from the drop-down list in the entity's **Roles** manager. Select the user's name from the list, then click **Remove**.

The user no longer appears in the list for that role and is no longer associated with it at the entity-level. Roles can be reassigned if necessary.

Save the changes (, Ctrl+S or **File** > **Save**).

12 Accessing your profile and rights

Required Roles	n/a
Required Rights (by category)	User Management
Perspective	**Preferences** menu
Access	*right-click on the* DROPS Server → Preferences

Rights grant users the possibility to create, view and modify entities. Profiles are defined and granted rights by the administrator.

If you do not have the rights to access a certain DROPS entity, when you try to, an error message displays informing you that you are not allowed to use that feature.

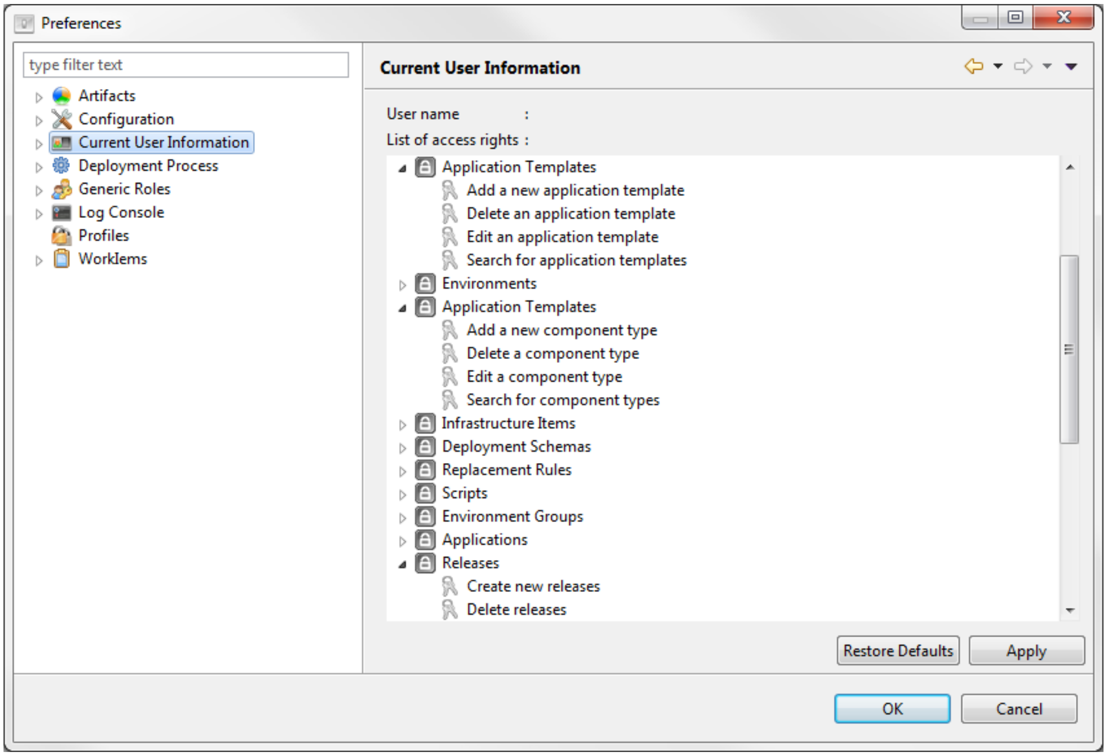
If you have the rights to access the **Users** view, you can see which profile(s) your user is assigned in the **User** editor.

If you cannot access the **User** editor, you can still view the list of your user's current rights in the **Preferences** menu. Click **Current User Information** to view all of the rights granted to your current user.

Reference

For more information about granting rights to user profiles, refer to the *DROPS Configuration Guide* or contact your administrator.

For more information about assigning profiles to Users, refer to User accounts on page 44.



Current User Information

User name :
List of access rights :

- Application Templates
  - Add a new application template
  - Delete an application template
  - Edit an application template
  - Search for application templates
- Environments
- Application Templates
  - Add a new component type
  - Delete a component type
  - Edit a component type
  - Search for component types
- Infrastructure Items
- Deployment Schemas
- Replacement Rules
- Scripts
- Environment Groups
- Applications
- Releases
  - Create new releases
  - Delete releases

Figure 12: Current user information (Preferences menu)

Configuring the import process

Introduction to configuring the import process

DROPS makes it easy to import and organize heterogeneous applications developed on different platforms and technologies to prepare them for deployment. For this, it uses a multi-platform artifact repository, structured by application and component, and loaded by import strategy. These entities, and more, must be configured in able to import the up-to-date releases which feed deployment processes.

The configuration described in this section covers process-level configuration completed using the **Configuration Management** perspective (see Overview of the DROPS Studio on page 34).

Note
Most of the high-level configuration necessary in the DROPS Studio must be performed by a DROPS admin who has the right(s) to configure global operations.

Reference
For more information about high-level configuration, refer to the *DROPS Configuration Guide*.

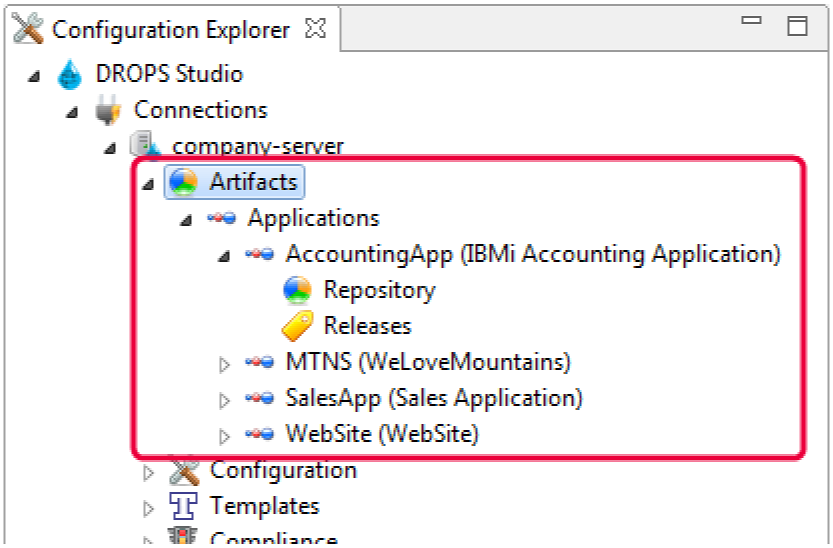


Figure 13: The Configuration Management perspective - artifact repository

The **Configuration Explorer** provides access to the entities required to set up the import process. This perspective is the configuration side of DROPS.

The “artifact repository” refers to the content managed from the **Artifacts** node under the currently connected DROPS Server. All of the applications defined for that server are configured, managed and accessed from the **Applications** node. Each application is displayed using its *code* and *(name)*. Double-clicking a node opens the **Application** editor where the following application-level configuration entities are managed:

- Applications (see Applications on page 60). Under each application node is the entry point to:
  - the artifacts it contains (see Imported artifacts on page 88),
  - the associated releases created after importing artifacts (see Releases on page 79 and Working in release branching mode on page 85).

- Components (see Components on page 67),
- Import strategies (see Import strategies on page 72).

To update the content of any menu list, right-click then select **Refresh**.

13 Applications

Required Roles	Application Administrator and/or Repository Manager
Required Rights (by category)	Applications
Perspective	Configuration Management
Access	DROPS Server → Artifacts → Applications

Chapter Summary

13.1 Creating applications 60
13.2 Editing application details 61
13.3 Deleting applications 66

Applications are the entities that contain the imported artifacts you intend to deploy. They are the highest level container in the artifact repository and house components, which contain individual artifacts. Each version of an artifact is contained in a release.

The ideal way to manage an application is to insure it contains all of the pieces necessary to complete the set of content you will deploy, no matter what their type, source location or target destination is.

> ***Example***
>
> You are deploying a software product that uses a web-based interface to connect to a server. In order to function correctly, your application might contain HTML files, the *.css* style sheet and all of the files required to install and launch the server. Even if these elements were created on different machines or if they must be deployed to different machines, they should be contained in the same application.

All of the applications defined for the current server are nested under the parent **Applications** node in the **Configuration Explorer**. They are listed, from top to bottom, in the order in which they were created.

13.1 Creating applications

Follow the subsequent steps to create a new application.

Step 1 Right-click the **Applications** node in the **Configuration Explorer**, then select **Create a new application**.

Step 2 Define the two mandatory properties: the App. Code and App. Name. These values can be edited later.

> ⚠ **Important!**
>
> Two applications cannot have the same ID. Every application's code must be unique.

Step 3 [*Optional*] Select an **application template** to apply to the new application.

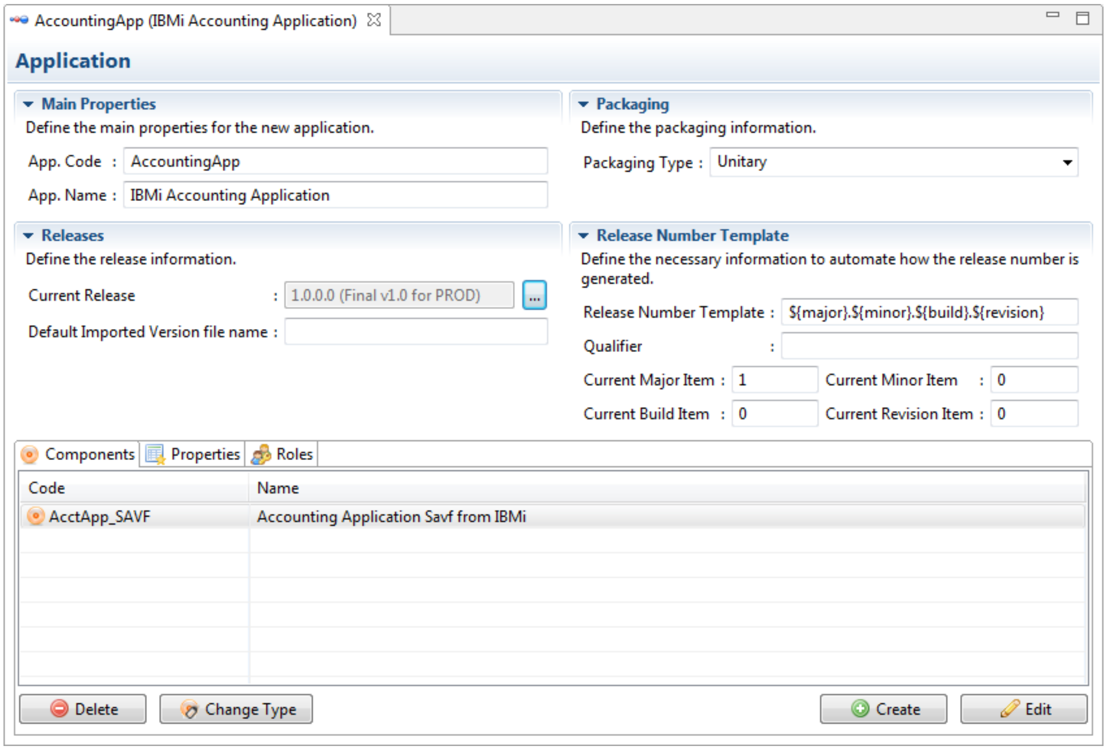
> ***Reference***
> For more information about **application templates**, refer to Application templates on page 209

Step 4 Click **Finish**.

Result New applications are nested under the **Applications** node.

13.2 Editing application details

To edit an application's details, either right-click on it then select **Edit**, or double-click on it. Save the changes (, Ctrl+S or **File** > **Save**). The **application** editor is opened where all of the entity's details are managed.


AccountingApp (IBMi Accounting Application)

Application

Main Properties
Define the main properties for the new application.
App. Code : AccountingApp
App. Name : IBMi Accounting Application

Packaging
Define the packaging information.
Packaging Type : Unitary

Releases
Define the release information.
Current Release : 1.0.0.0 (Final v1.0 for PROD)
Default Imported Version file name :

Release Number Template
Define the necessary information to automate how the release number is generated.
Release Number Template : ${major}.${minor}.${build}.${revision}
Qualifier :
Current Major Item : 1
Current Minor Item : 0
Current Build Item : 0
Current Revision Item : 0

Code	Name
AcctApp_SAVF	Accounting Application Savf from IBMi

Figure 14: The application editor

The information that defines applications can be edited at any time by users with the rights to do so.

> ***Reference***
> For more information about the different actions managed from the tabs in the **application** editor that are not described in this chapter, refer to:
>
> - Components on page 67
> - Properties on page 412
> - Assigning entity-level roles on page 54

App. Code

A unique string that identifies applications. This label is used throughout DROPS to select the entity.

App. Name

A string used to give additional information about the application. This label is used throughout DROPS to select the entity.

Packaging Type

This value is used by the deployment processes when the **Package Type** field is set to **From application**.

Set the default packaging type at the application-level to assure that every time the application is delivered by a deployment process, the packaging will be consistent. However, this default value can be overridden by choosing another value in the **Package Type** field when defining the deployment process (see Edit global deployment process properties on page 177).

[*default*] **Unitary** packaging deploys only the release selected.

Cumulative packaging deploys the selected release and all of the preceding releases that have been imported into DROPS but have not yet been deployed to the target environment. A 'cumulative packaging type' deploys only the most recent artifacts in each release. This ensures that all of the most up-to-date artifacts are deployed and that the deployed application is accurately updated.

> ***Example***
>
> You have imported 4 versions of your release but only deployed the first version to the target environment. The 4th release contains the same version of all the artifacts that the 3rd release contains, except for one file. Now, you select the 4th release to deploy and choose a 'cumulative packaging type'. Result: The 2nd, 3rd and 4th releases will all be deployed in order, finishing with the most current version of every artifact and release. The artifacts from the 3rd release are all current except the one, which is deployed with the 4th release.

Current Release

Select the release that should be considered current for the application. If **Current Release** is selected during the import process, the release defined in this field will be updated with refreshed artifacts. Instead of creating a new release with the new versions of the artifacts, DROPS replaces the old versions in the current release with the new versions.

The current release will also be used automatically if an external file management system calls DROPS and uses the current release to import its deliverable(s).

> ⚠ **Important!**
>
> The **Current Release** field is not automatically populated with the most recently imported release.

Click the browse button to open the **Release Selection** dialog, which is prepopulated with the releases that have already been imported for the current application.

You may use the Automatic Release Resolver to use a script to resolve release number when importing artifacts.

> ***Reference***
>
> For more information about the import process, refer to Introduction to importing artifacts on page 242.
>
> For more information about managing releases, refer to Releases on page 79.

Default Imported Version file name

Predefine which version of an artifact to import. This option can also be defined in the artifact **Import** wizard's **Component Filter** page by listing all of the versions from the potentially many external applications.

Alternatively, you can create a properties file that connects the desired version from the external system to the appropriate component and strategy in DROPS. This properties file must be stored on the same system that the DROPS Server is installed on.

> **Important!**
>
> You can use variables, however the syntax in this properties file *must* follow this standard: **component code.strategy code=external version.**

> ***Example***
>
> If version V1.10.00 should be used for a component with the code "Commands" and the strategy with a code "import", the syntax for one line of the properties file must be *Commands.import=V1.10.00*.

Enter the full path to the defined properties file in this field. The **External file name** field in the artifact **Import** wizard will be automatically populated with this path.

> ***Reference***
>
> For more information about the Artifact Import wizard, refer to Importing artifacts immediately by application on page 245.

Edit applications: Release flat mode

If the application is set to display **releases in flat mode**, the follow options are available to define how release versions are managed.

Release Number Template

Releases are organized by number. This version number is set by default when the release is created during the import process and is based on a predefined template that can include fixed text and/or variables. When a release is created multiple times, the version number increases corresponding to the variables defined in its application. The user defines which part of the version to increment for each import in order to conform with the increasing release versions according to the Automatic Release Number Generation Algorithm.

The default value for this field is defined in the **Preferences** menu (refer to the *DROPS Configuration Guide*).

To manually override the global default value and create a new default value for this application, edit the field to include any combination of fixed text and/or the version variables available.

- ${major}
- ${minor}
- ${build}
- ${revision}
- ${qualifier}

> ⚠ **Important!**
> The hierarchy of these variables is always in this order: **major > minor > build > revision**. Be sure to keep this in mind when entering the order of the variables in your template.

The Automatic Release Number Generation Algorithm is an intelligent counter that increases the version number of each release by one automatically according to the element selected to increase. When a new release is imported the algorithm follows these rules:

1. IF **${major}** variable is used **and** IF the selected element to increment is **Major**:
   - Increment the value in the **Current Major Item** field of the related application.
   - Use the new value as the replacement value.
   - Reset all values lower than major to 0.
2. IF **${minor}** variable is used **and** IF the selected element to increment is **MINOR**:
   - Increment value in the **Current Minor Item** field of the related application.
   - Use the new value as the replacement value.
   - Reset all values lower than minor to 0.
3. IF **${build}** variable is used **and** IF the selected element to increment is **BUILD**:
   - Increment the value in the **Current Build Item** field of the related application.
   - Use the new value as the replacement value.
   - Reset all values lower than build to 0.
4. IF **${revision}** variable is used **and** IF the selected element to increment is **REVISION**:
   - Increment the value in the **Current Revision Item** field of the related application.
   - Use the new value as the replacement value.

5. IF **${qualifier}** substitution variable is used:
    - IF the **Qualifier** field in the related application is empty, assign **${timestamp}**.
        - Replace **${timestamp}** variable with the current date and time.
        - Use the new value as the replacement value
    - IF the **Qualifier** field in the related application contains content, use that content.

Qualifier

The value entered here replaces the **${qualifier}** variable when a new release is created. You might use it as a suffix or a prefix for your version number.

If the **Qualifier** field is left blank, the **${qualifier}** variable will be automatically replaced by the special tag **${timestamp}**. The timestamp variable represents the current date and time.

If you enter a value into the **Qualifier** field, the timestamp variable will not be included in the release version unless you include it manually in the **Qualifier** field.

> ***Example***
>
> Version-${timestamp}_

Current Major, Minor, Build and Revisions Items

The values entered in these fields replace the **${major}**, **${minor}**, **${build}** and **${revision}** variables when a new release is created.

The values are automatically updated when a new release is imported and the version number is increased. However, these fields can be updated manually to control the next generation of your release Package. Only whole numbers can be entered in these fields.

Edit applications: Release branching mode

If the application is set to display **releases in branching mode**, the follow options are available to define how release versions are managed.

Release Tree Depth

Define the maximum number of sub-levels allowed for releases. In branching mode, minor releases can only be created from major releases and only a defined-number of sub-levels can be created.

> ***Example***
>
> If you want to allow the system to create three sub-levels of release versions, the highest you will be able to go is xx.xx.xx. If you add a 4th sub-level to the release branching you will be able to add the level xx.xx.xx.**xx**.

Release Policy (tab)

If the release branching mode is activated, the **Release Policy** tab is added to the Application editor. To edit the template for each level, select it and click the **Edit** button.

Reference

For more information about branching mode, refer to Working in release branching mode on page 85.

13.3 Deleting applications

Warning!

Deleted applications cannot be accessed or recovered. Any associated components are also deleted.

To delete an application, right-click its node in the **Applications** menu, then select **Delete**. Click **OK** to confirm or click **Cancel** to keep the application.

14 Components

Required Roles	n/a
Required Rights (by category)	Applications
Perspective	Configuration Management
Access	DROPS Server → Artifacts → Applications → *Application* editor → **Components** tab

Chapter Summary

14.1 Accessing components 67
14.2 Creating components 68
14.3 Editing component details 69
14.4 Changing the type of a component 70
14.5 Changing the extension of a component 71
14.6 Deleting components 71

Components are unique, functional subdivisions of applications used to structure releases created when artifacts are imported into the artifact repository. Components should group an application's artifacts by the intended deployment target because every artifact in a component will be deployed to the same infrastructure item.

A component can only belong to one application but an application can have many components. They are used to manage import strategies and are the parent entity for imported artifacts in the artifact repository's hierarchy. Artifacts that belong to a component can come from different sources which means that a component can contain multiple import strategies, which each provide DROPS access to the artifacts to import.

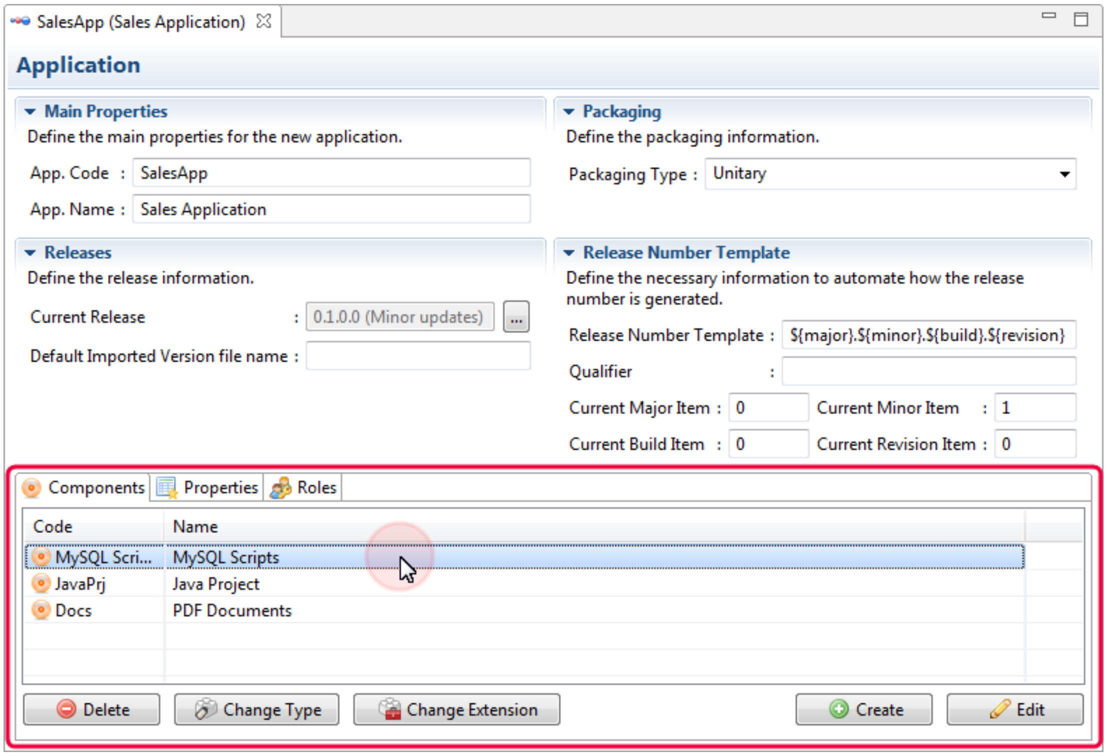
During the deployment process, the contents of components (artifacts) are deployed to target systems following the same directory structure defined in the artifact repository. The structure inside the component node is defined in each import strategy.

> ***Example***
>
> Your application is a software product that uses a web-based interface to connect to a server. Both the website and the server are required for the application to work correctly. The server is located in New York but the website's source HTML files and the .css style sheet must be hosted in Paris. The application should contain two components - one for each destination.

14.1 Accessing components

Components are defined and accessed in applications. To access the list of components defined for an application, select the application from the **Applications** node in the **Configuration Explorer** and refer to the **Components** tab.

This tab displays all of the components defined for the current application.



SalesApp (Sales Application)

Application

Main Properties
Define the main properties for the new application.
App. Code : SalesApp
App. Name : Sales Application

Packaging
Define the packaging information.
Packaging Type : Unitary

Releases
Define the release information.
Current Release : 0.1.0.0 (Minor updates)
Default Imported Version file name :

Release Number Template
Define the necessary information to automate how the release number is generated.
Release Number Template : ${major}.${minor}.${build}.${revision}
Qualifier :
Current Major Item : 0
Current Minor Item : 1
Current Build Item : 0
Current Revision Item : 0

Code	Name
MySQL Scri...	MySQL Scripts
JavaPrj	Java Project
Docs	PDF Documents

Figure 15: The Application editor - components

14.2 Creating components

Follow the subsequent steps to create a new component.

Step 1 Open the application for which you want to create a new component.

Step 2 Click the **Create** button in the **Components** tab of an application to open the **Create a new component** wizard.

Step 3 Define the two mandatory properties: the Component Code and Component Name. These values can be edited later.

Step 4 Select a pre-defined component type from the Component type drop-down list.

> **Note**
> This element is not mandatory but may help to organize components. It also allows the use of templates associated with the component type.

> ***Reference***
> For more information about component types, refer to Component types.

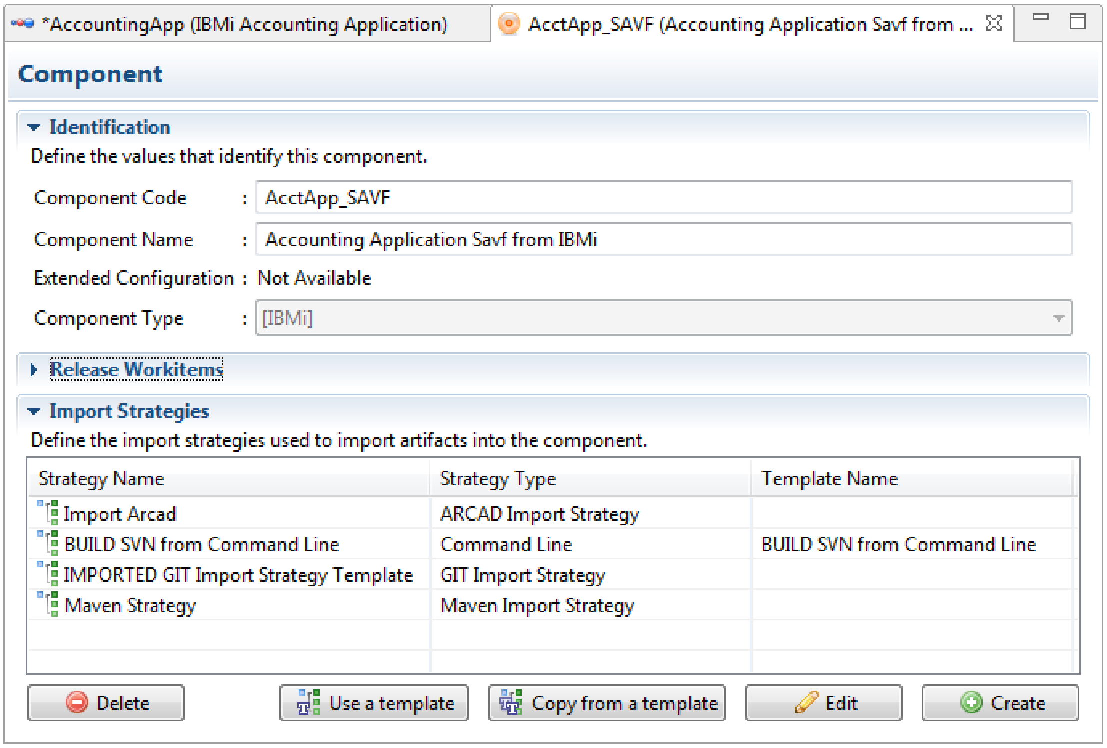
Step 5 If the component is to import an ARCAD application, it is recommended to define the Component Extension Type. Select the ARCAD type from the drop-down list. If you are not importing an ARCAD application, you can leave this field blank. Components are not required to have an extension type.

Step 6 Click **Finish**.

Result New components are displayed in the **Components** manager.

14.3 Editing component details

To view or edit a component, double-click it or select it then click **Edit** to open the **Components** editor.


*AccountingApp (IBMi Accounting Application)
AcctApp_SAVF (Accounting Application Savf from ...

Component

Identification

Define the values that identify this component.

Component Code	:	AcctApp_SAVF
Component Name	:	Accounting Application Savf from IBMi
Extended Configuration	:	Not Available
Component Type	:	[IBMi]

Release Workitems

Import Strategies

Define the import strategies used to import artifacts into the component.

Strategy Name	Strategy Type	Template Name
Import Arcad	ARCAD Import Strategy	
BUILD SVN from Command Line	Command Line	BUILD SVN from Command Line
IMPORTED GIT Import Strategy Template	GIT Import Strategy	
Maven Strategy	Maven Import Strategy	

Figure 16: The Component editor

The information that defines components can be edited at any time by users with the rights to do so.

> ***Reference***
> For more information about the different actions managed from the tabs in the **Components** editor that are not described in this chapter, refer to:
>
> - Import strategies on page 72.

Component Code

A unique string that identifies components. This label is used throughout DROPS to select the entity.

An application cannot contain multiple components with the same code. This means that all of the components in one application must have unique codes, but that two different applications can contain two different components even if they both use the same code.

Component Name

A label used to give additional information about the component. This label is used throughout DROPS to select the entity.

Extended Configuration

Some component types require extended configuration. If the component's current type does, this field will be a link to the type-specific configuration editor.

> ***Reference***
> For more information about extended configuration editors, refer to Component type extended configuration in the appendix.

Component type

A value used to assign a type to the component. This drop-down list is prepopulated with all of the component types defined by the administrator in the **Configuration Templates** node in the **Configuration Explorer**.

Save the changes (, Ctrl+S or **File** > **Save**).

14.4 Changing the type of a component

Component types are defined when creating a component and cannot be edited in the component's editor view. It can be changed in the **Application** editor.

To change the type of a component, click on the **Change type** button in the **Application** editor.

Select the new component type from the list, which is prepopulated with the component types available.

> ***Warning!***
> Component types may include extended template configuration. Changing the component type will delete and/or replace any preexisting configuration already defined for the component.

> ***Reference***
> For more information about component type configuration, refer to Component types on page 212.

14.5 Changing the extension of a component

The extended configuration for a component is defined when creating a component, and cannot be edited in the component's editor view. It can be changed in the **Application** editor.

To change the extension of a component, in order to gain access to any extended configuration, click on the **Change Extension** button in the **Application** editor.

Select the new extension from the list, which is prepopulated with the types available.

> ***Warning!***
> Changing the component's extension will delete and/or replace any preexisting extended configuration already defined for the component, even if the type of extension chose is the same as it was originally.

> ***Reference***
> For more information about extended configuration, refer to Component type extended configuration on page 432.

14.6 Deleting components

> ***Warning!***
> Deleted components are no longer associated with their parent application and cannot be accessed or recovered. Any associated import strategies are also deleted.

To delete a component, select the item in the **Components** tab of an application, then click **Delete**. Click **OK** to confirm or click **Cancel** to keep the component.

Save the changes (, Ctrl+S or **File** > **Save**).

15 Import strategies

Required Roles	n/a
Required Rights (by category)	Applications
Perspective	Configuration Management
Access	DROPS Server → Artifacts → Applications → *Application editor* → *Component editor* → Import Strategies manager

Chapter Summary

15.1 Accessing import strategies ... 73
15.2 Creating import strategies ... 73
15.3 Editing import strategy details ... 75
15.4 Deleting import strategies ... 78

The artifact repository is populated with artifacts imported from external sources, sometimes called *artifact providers*. Import strategies are dedicated engines that communicate with one particular artifact provider in order to find the defined artifacts and import them, nested under the associated component, into the artifact repository.

When defining import strategies, the link to the external source that points to the artifacts to retrieve creates, by default, the directory structure to respect when storing the imported artifacts. Strategies are created inside the **Component** editor so they are linked by default to that component. The imported artifacts are automatically stored under that component in the artifact repository. DROPS ensures that each artifact is associated with the correct application and release via the component.

An external artifact provider may be a simple file system or a complex SCM. Choosing the type of import strategy to use depends *solely* on the kind of source from which you are importing artifacts. Only one source type can be defined per import strategy but a component can contain multiple strategies in order to import artifacts from different sources.

Import strategies can be configured to recognize updated files in the defined paths and only import them, instead of the entire group of artifacts available.

Strategies reach the artifacts to import via the object's absolute path. The artifacts in this **Root Directory** are browsed after the import is launched to determine which to import or not.

Some import strategy parameters can be defined using properties.

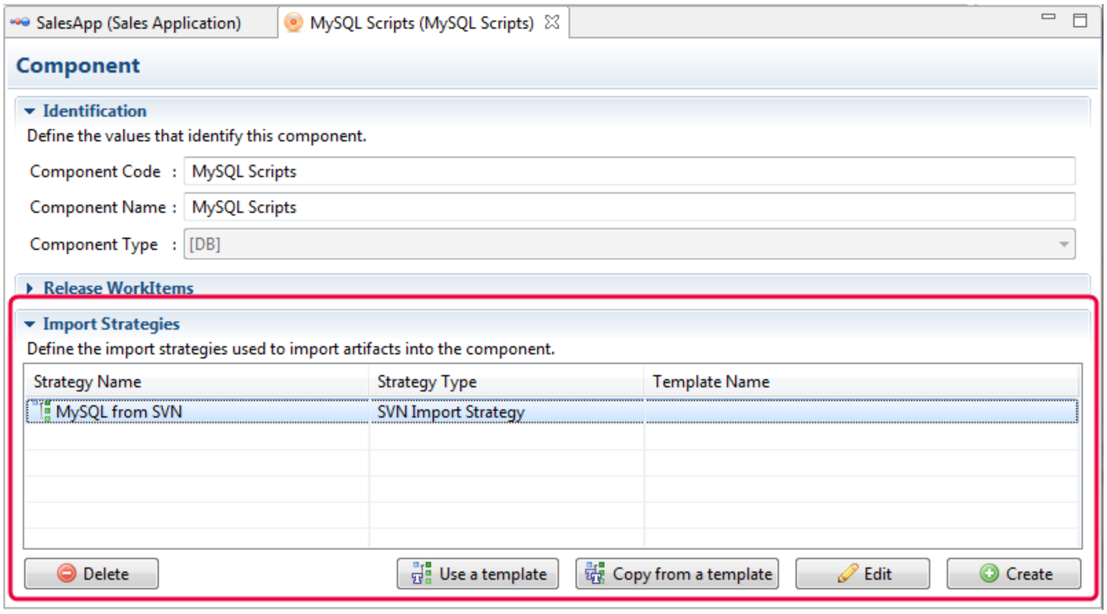
Example

The two components in your application are to be deployed to two different destinations (a server in New York and a website host in Paris). The server's installation files are all imported from the same location - there is only one import strategy necessary for the New York server component. However, the website component for Paris must contain HTML files that your web designer created and sent to you via email and the .css style sheet that your company hosts on a shared network. In order to import all of these artifacts from two different source locations, you will need two different import strategies: one for each source location.

15.1 Accessing import strategies

Import strategies are defined and accessed in components. To access the list of import strategies for a component, select it from the component**s** tab in the **Application** editor and refer to the **Import Strategies** section.

This tab displays all of the strategies defined for the current component.



Define the values that identify this component.

Component Code : MySQL Scripts
Component Name : MySQL Scripts
Component Type : [DB]

Define the import strategies used to import artifacts into the component.

Strategy Name	Strategy Type	Template Name
MySQL from SVN	SVN Import Strategy	

Figure 17: The component editor - import strategies

15.2 Creating import strategies

> **Note**
> The order in which import strategies are created is the order in which they will be carried out when the import process begins.

There are three ways to create import strategies:

1. Create a new strategy using the **Create a new import strategy** wizard.
2. Use or Copy from strategy templates to quickly create a strategy with predefined parameters.
   a. *Using a template* means that whenever the template is changed, the changes will be carried over to all of the Strategies that use it. There is a connection between the template and the new Strategy which permanently inherits the template's parameters.

b. *Copying from a template* creates a copy of the template's parameters which can be modified. There is no on-going connection between the template and the new Strategy.

> ⚠ **Important!**
> No matter how you create an import strategy, you must edit it once it is created to finish the configuration. However, how you configure a strategy depends on which type of strategy you create.
>
> Refer to the instructions that correspond to the type of strategy created to finish the import strategy configuration.

15.2.1 Create a new strategy

Follow the subsequent steps to create a new import strategy.

Step 1 Open the component for which you want to create a new import strategy.

Step 2 Click the **Create** button in a component's **Import Strategies** manager to open the **Create a new import strategy** wizard.

Step 3 Define the Import Strategy Code, Import Strategy Name and the Root Directory. These values can be edited later. By default the Root Directory is set to ***AUTO**.

Click **Next >** to continue.

Step 4 Select the Strategy Type to use.

> ⚠ **Important!**
> Take care when selecting the type to use based on the type of source your artifacts are coming from.

Step 5 Click **Finish**.

Result New import strategies are displayed in the **Import Strategies** manager. It is important to edit the Strategy and finish the configuration before using it.

15.2.2 Use or copy from strategy templates

Follow the subsequent steps to create a new import strategy by either using or copying from a template.

Step 1 Click the **Use a template** or the **Copy from a template** button in a component's **Import Strategies** manager. See above for more information about the differences between *using* and *copying from* strategy templates.

Step 2 Select the template from the **Strategy Templates** dialog.

The list of available templates is prepopulated with the templates created by users who have the rights to do so.

> ⚠ **Important!**
> Take care when selecting the type to use based on the type of source your artifacts are coming from.

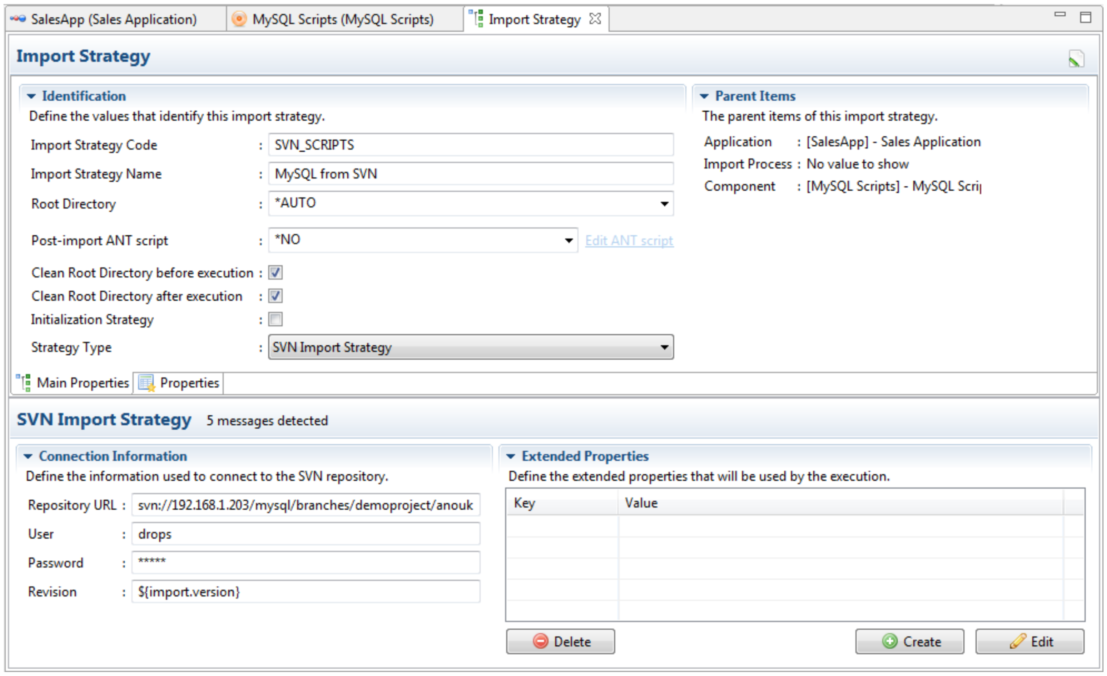
> ***Reference***
> For more information about creating strategy templates, refer to Import strategy templates on page 215.

Step 3 Click **OK**.

Result New import strategies are displayed in the **Import Strategies** manager. If a strategy *uses* a template, the name of the template is displayed in the template **Name** column to remind you that it is linked to the template. It is important to edit the Strategy and finish the configuration before using it.

15.3 Editing import strategy details

To view or edit an import strategy, double-click it or select it then click **Edit** to open the **Import Strategy** editor. It is important to edit import strategies before using them in order to configure them correctly based on your artifact provider.



Import Strategy

Identification
Define the values that identify this import strategy.

Import Strategy Code	SVN_SCRIPTS
Import Strategy Name	MySQL from SVN
Root Directory	*AUTO
Post-import ANT script	*NO (Edit ANT script)
Clean Root Directory before execution	☑
Clean Root Directory after execution	☑
Initialization Strategy	☐
Strategy Type	SVN Import Strategy

Parent Items
The parent items of this import strategy.

Application	[SalesApp] - Sales Application
Import Process	No value to show
Component	[MySQL Scripts] - MySQL Scri

SVN Import Strategy 5 messages detected

Connection Information
Define the information used to connect to the SVN repository.

Repository URL	svn://192.168.1.203/mysql/branches/demoproject/anouk
User	drops
Password	*****
Revision	${import.version}

Extended Properties
Define the extended properties that will be used by the execution.

Key	Value

Figure 18: The Import Strategy editor

The information that defines import strategies can be edited at any time by users with the rights to do so.

Reference

For more information about the different actions managed from the tabs in the **Import Strategy** editor that are not described in this chapter, refer to:

- Properties on page 412

The **Main Properties** tab contains the details common to all of the different types of import strategies, however each type requires very specific configuration as well.

The import strategy's parent entities are displayed in the **Parent Items** section.

Import Strategy Code

A unique string that identifies import strategies. This label is used throughout DROPS to select the entity.

Import Strategy Name

A label used to give additional information about the import strategy. This label is used throughout DROPS to select the entity.

Root Directory

The directory path where retrieved artifacts are temporarily stored before being imported into the artifact repository.

For all of the types of Import Strategies ***except*** **the File System Strategy**, the root directory is used to temporarily store the artifacts to be imported before DROPS actually imports them into the Artifact Repository. When the import is launched, DROPS:

1. [*clears* the root directory *IF OPTION IS SELECTED*];
2. fills it with the appropriate artifacts for temporary storage;
3. communicates with the DROPS Server parameters to learn what directory structure to use to store the artifacts in the Artifact Repository;
4. *copies* the artifacts to the Artifact Repository;
5. [*clears* the root directory *IF OPTION IS SELECTED*].

For the **File System Strategy**, the root directory must be the absolute root of all of the artifacts to be imported using this Strategy. Only the path up to the artifacts' *shared* root should be entered in this field. When the import is launched, DROPS:

1. [*clears* the root directory *IF OPTION IS SELECTED*];
2. analyzes this root path to find the artifacts to import;
3. communicates with the DROPS Server parameters to learn what directory structure to use to store the artifacts in the Artifact Repository;
4. *copies* the artifacts to the Artifact Repository;
5. subtracts the exact path of the **Root Directory** field from the absolute path of each artifact and

appends the results to the Artifact Repository path;

6. [*clears* the root directory *IF OPTION IS SELECTED*].

The path to the imported artifact is no longer *c://rootdirectory/../.artifact*, but is now *${root}/app${application}/cmp${component}/rls${release}/../.artifact* (where ${root} is the Repository Storage Area defined in the DROPS Server parameters).

Enter an absolute path here or keep the default value (***AUTO**) to point to the default path defined in the high-level server configuration. By default, this path leads to a relative **tmp** path where the DROPS Server is installed.

This path *must* be on the same machine as the DROPS Server.

For more information about defining the temporary and final locations involved in the import process, refer to the *DROPS Configuration Guide*.

Post-import Ant script

The selected script will be executed after the import process is finished but before the artifacts are copied to the artifact repository. Select the script to execute then click the **Edit Ant script** link to open the script to edit.

***NO** or nothing selected: no script is executed.

***AUTO**: the script specific to this import strategy. The core of the script exists already but the actions to execute must be defined.

Enter an absolute path to script: Enter the path to an existing script outside of DROPS. The script must be accessible to the DROPS Server. If no script already exists at the defined location, DROPS will generate the script template and save a new script file there for you (i.e. you can create new external Ant scripts using this method).

Clean Root Directory before execution

If ticked, the content of the Root Directory is cleaned *before* executing the import. By default, this option is disabled.

Be careful using this option as you may lose content that you did not intend to be cleaned! This is especially important if you have multiple strategies that point to the same Root Directory.

Clean Root Directory after execution

If ticked, the content of the Root Directory is cleaned *after* executing the import. By default, this option is disabled.

Be careful using this option as you may lose content that you did not intend to be cleaned! This is especially important if you have multiple strategies that point to the same Root Directory.

Initialization Strategy

If ticked, the import strategy is only used when running an initial import.

For more information about initial imports, refer to Importing artifacts immediately by application on page 245.

Strategy Type

This drop-down list contains all the available strategy types provided by DROPS. By default it is set to the type selected when the import strategy was created. Each type of strategy requires different extended configuration and it is essential that the type correspond to the method you use to collect artifacts to import.

If the extended configuration for an import strategy has already been defined, changing the type will delete **all** of the existing information.

To create a different type of strategy and keep the information defined in the current one, create a new import strategy.

For more information about the extended configuration necessary for each type of import strategy, refer to Import strategy type configuration in the appendix.

Save the changes (, Ctrl+S or **File** > **Save**).

15.4 Deleting import strategies

Warning!
Deleted import strategies cannot be accessed or recovered.

To delete an import strategy, select the item from the components editor then click **Delete**. Click **OK** to confirm or click **Cancel** to keep the strategy.

16 Releases

Required Roles	n/a
Required Rights (by category)	Releases
Perspective	Configuration Management
Access	DROPS Server → Artifacts → Applications → *Application* → Releases

Chapter Summary

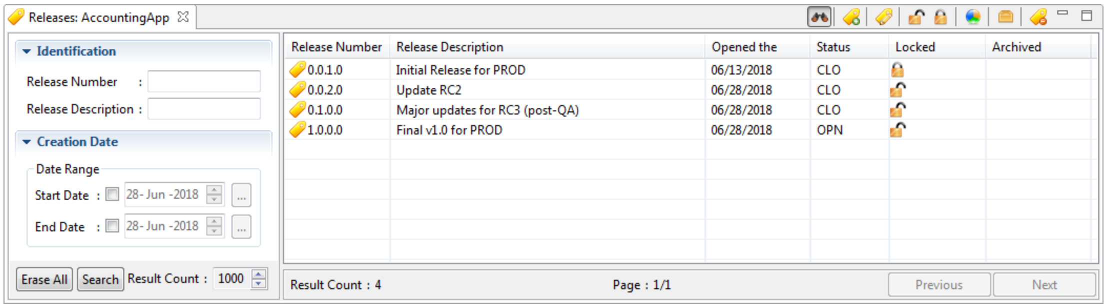
16.1 Creating releases 80
16.2 Editing release details 80
16.3 Viewing environments to which a release has been deployed 82
16.4 Making comments 82
16.5 Viewing attachments 82
16.6 Viewing associated workitems 83
16.7 Locking and unlocking releases 83
16.8 Viewing the artifacts in releases 83
16.9 Archiving releases 83
16.10 Deleting releases 84

Releases are consistent sets of artifacts identified by a unique number and created when artifacts are imported into the artifact repository. They are contained in components.

During the deployment process, releases are selected to deploy, not individual artifacts. All of the artifacts included in a release are deployed at the same time, unless release sets are deployed. Sets of releases are parsed to only include the most recent version of an artifact if the same artifact is included in multiple releases. The release with the most recent version of the artifact will include it during deployment and the releases with outdated versions of that artifacts will not.

Releases can be managed in two different ways: in flat mode and in branching mode. In flat mode, releases have no automated relationships with each other. This chapter describes how to work with releases in flat mode.

> ***Reference***
> For more information about working in branches, refer to Working in release branching mode on page 85.



Release Number :
Release Description :
Start Date : □ 28- Jun -2018
End Date : □ 28- Jun -2018
Result Count : 1000

Release Number	Release Description	Opened the	Status	Locked	Archived
0.0.1.0	Initial Release for PROD	06/13/2018	CLO	🔒	
0.0.2.0	Update RC2	06/28/2018	CLO	🔓	
0.1.0.0	Major updates for RC3 (post-QA)	06/28/2018	CLO	🔓	
1.0.0.0	Final v1.0 for PROD	06/28/2018	OPN	🔓	

Result Count : 4
Page : 1/1

Figure 19: The Releases search view

16.1 Creating releases

Releases are either created automatically during the import process or manually from the **Releases** search view. Releases can also be created by Jenkins if you launch the import process externally.

> ***Reference***
>
> For more information about the import process and how releases can be created automatically, refer to Importing artifacts on page 241.
>
> For more information about importing with Jenkins, refer to DROPS for Jenkins on page 362.

Follow the subsequent steps to create a new release manually.

Step 1 To access the **Create a new release** wizard, either click the create icon in the **Releases** search view or right-click anywhere in the view and select **Create a new release**.

Step 2 Define the Release Number, if applicable, and the Release Description. These values can be edited later.

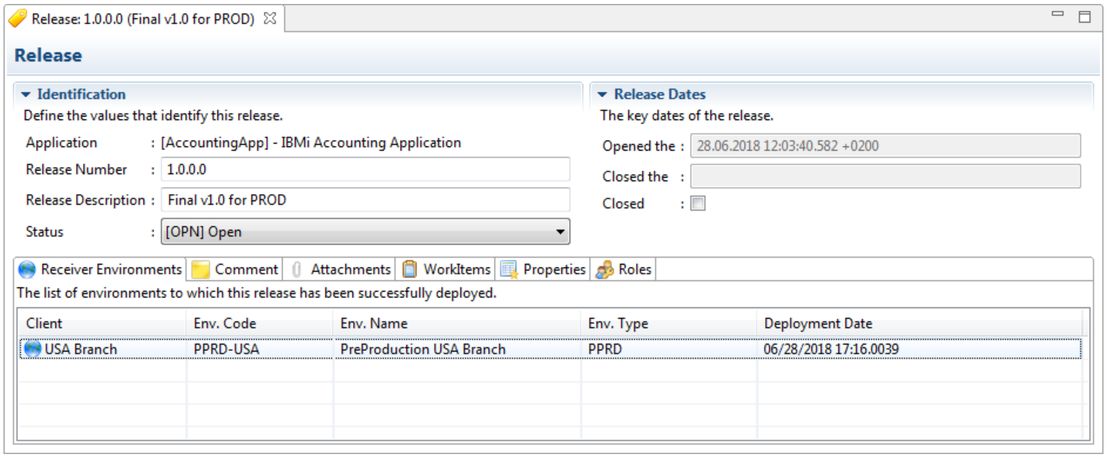
> **Note**
>
> The release number must be unique for the application.
>
> Variables cannot be used.

Step 3 Click **Finish**.

Result The new release is available in the **Releases** search view and can be used to import artifacts.

16.2 Editing release details

To edit release details, either right-click on it in the **Releases** search view then select **Edit**, select the item then click the edit icon or double-click it. The **Release** editor is opened where all of the entity's details are managed.



Define the values that identify this release.

Application : [AccountingApp] - IBMi Accounting Application
Release Number : 1.0.0.0
Release Description : Final v1.0 for PROD
Status : [OPN] Open

The key dates of the release.

Opened the : 28.06.2018 12:03:40.582 +0200
Closed the :
Closed : □

The list of environments to which this release has been successfully deployed.

Client	Env. Code	Env. Name	Env. Type	Deployment Date
USA Branch	PPRD-USA	PreProduction USA Branch	PPRD	06/28/2018 17:16.0039

Figure 20: The Release editor

The information that defines releases can be edited at any time by users with the rights to do so.

Reference

For more information about the different actions managed from the tabs in the **Release** editor that are not described in this chapter, refer to:

- Properties on page 412
- Assigning entity-level roles on page 54

Application

A static field that displays the release's parent application.

Release Number

If a release was created automatically, the release number is automatically populated following the template defined in the Release Number Template field in the parent application.

Change the value in this field to modify the release number.

Note

The release number must be unique for the application.

Release Description

Manually enter a description for an existing release to easily identify it in the **Release** view and the deployment process.

Note

If you change the description of an release in Jenkins, this field will be updated.

Status

Select a status for the release that will enable users to easily recognize how or when to use it in the import and deployment processes.

Closed releases cannot be deployed.

The list is prepopulated with the statuses defined in the **Preferences** menu.

> ***Reference***
> For more information about defining release statuses, refer to the *DROPS Configuration Guide*.

Release Dates

This section displays the static dates that the release was created and closed. These fields cannot be modified. When the **Closed** checkbox is ticked, the Closed date is updated.

> **Important!**
> The CLO status for releases is not to be confused with the Closed checkbox in the release editor. The *status* must be updated manually and does not have a functional effect on the release or its ability to be deployed. It is only for information.

> **Note**
> Closed releases can not be imported or deployed.

16.3 Viewing environments to which a release has been deployed

If a release has been deployed, all of the target environments to which it was delivered, and the corresponding delivery dates, are displayed in the **Receiver Environments** tab in the **Release** editor.

16.4 Making comments

Use the **Comment** tab in the **Release** editor to enter any details that users should be aware of when using the release to import or deploy content.

16.5 Viewing attachments

Attachments can be automatically added to releases if an import strategy includes this action. The strategy defines which files can be saved as attachments. View any attachments added during the import process from the **Attachments** tab in the **Release** editor.

> **Note**
> The content of Save Files imported using the IBM i and Arcad Import Strategies are saved as .txt attachments by default. This makes it easy to refer to the content of the Save File outside of IBM i after it has been imported into the DROPS Studio.

Double-click on an attachment or select it and click the **Open** button to open it in a new tab in the studio.

Download an attachment by selecting it and clicking the **Save** button.

16.6 Viewing associated workitems

If the release was associated with a workitem, a link to the JIRA issue is included in the **Workitem** tab. Click the link to open the issue in your default explorer. The list of workitems in releases is populated automatically when the import process is launched.

16.7 Locking and unlocking releases

Locked releases cannot be used to import artifacts.

To lock or unlock a release and prevent users from using it to import artifacts, either right-click on it in the **Releases** search view then select **Lock**/**Unlock** or select the item then click the corresponding icon.

16.8 Viewing the artifacts in releases

To view the artifacts imported in a release, either right-click on it in the **Releases** search view then select **Show Artifacts** or select the item then click the artifacts icon.

Search for the artifacts in a release in the **Release Artifact** view by name, file type (**Extension**) or **Component**. To display the complete list of artifacts imported into the current release, click the **Search** button without entering any search criteria.

Imported artifacts are nested under their corresponding components' node and found in their source directory structure. Select an artifact and either right-click, then select **Show History**, or click the Show the release history icon to open the artifact **History** view.

This view displays when the artifact was added (its first release), and when it was modified (imported in other releases).

> ***Reference***
> For more information about this view, refer toViewing an artifact's import history on page 89.

16.9 Archiving releases

> ***Warning!***
> Unless they are archived in advance, archived releases cannot be accessed, used to import, deployed or recovered.

Archiving the artifacts from closed or outdated releases creates space on the DROPS Server and cleans up the artifact repository. The history of each release is maintained and the name(s) of the imported artifacts can still be found, but the artifacts themselves no longer appear in the repository.

> **Note**
> Releases must be locked before they can be archived.

Step 1 To archive a release, search for it in the **Release** search view. Either right-click on it then select **Archive** or select it, then click the archive icon.

Step 2 Tick the **Backup the release's Artifacts before archiving** checkbox to zip and save the artifacts in the release before archiving it. Backed-up artifacts are removed from the repository but the zip can still be consulted manually in the Archived Files Storage Area destination on the DROPS Server.

> ***Reference***
> For more information about configuring this storage area, refer to the *DROPS Configuration Guide*.

Step 3 Click **Finish** to confirm or click **Cancel** to cancel the request. If required, confirm the archive again.

Result The log file is opened automatically in your default text editor. If the backup was successful, the artifacts have been zipped and saved, then deleted from the repository. The archive icon is displayed in the release's instance in the **Release** search view.

16.10 Deleting releases

> ***Warning!***
> Deleted releases cannot be accessed, used to import, deployed or recovered. Deleting a release does not delete the artifacts that have already been imported by it. Only the trace of the release's container itself is removed. To remove the artifacts from the repository, they must be purged before the release is deleted.

To delete a release, search for it in the **Release** search view. Either right-click on it, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the release.

17 Working in release branching mode

Required Roles	n/a
Required Rights (by category)	Releases
Perspective	Configuration Management
Access	DROPS Server → Artifacts → Applications → *Application* → Releases

Chapter Summary

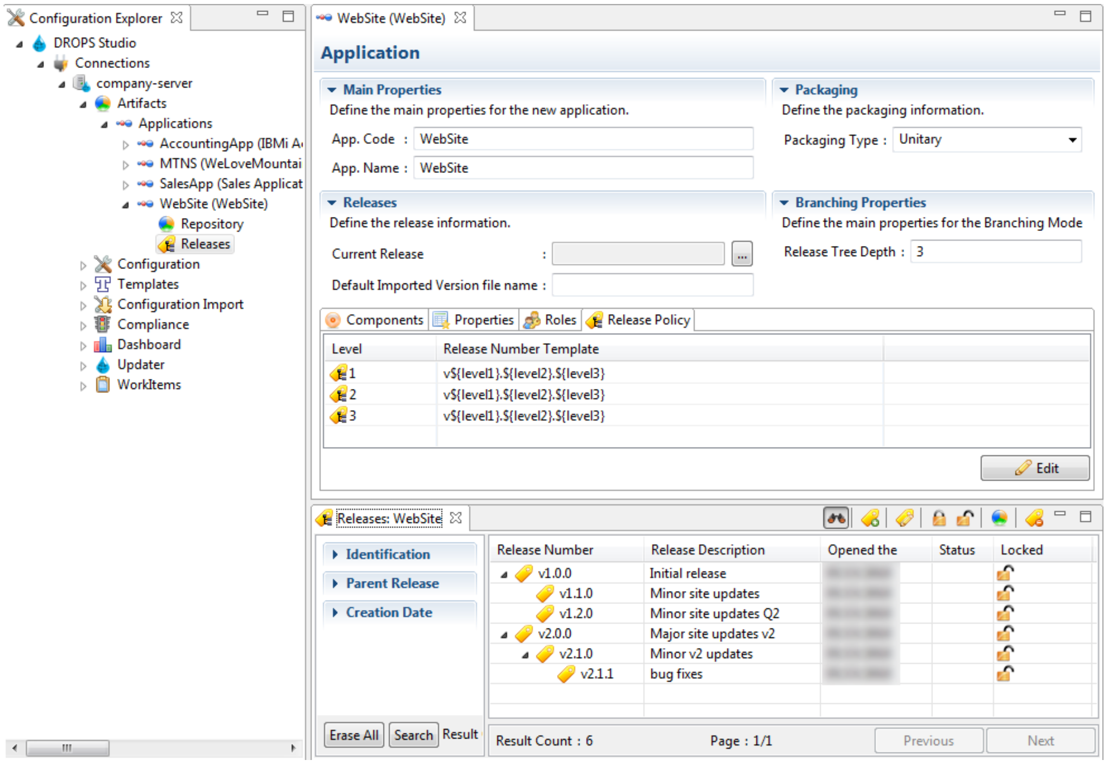
17.1 Switching to release branching mode 86
17.2 Searching for release branches 86
17.3 Creating branched releases 87
17.4 Editing and viewing branched release details 87

Releases can be managed in two different ways: in flat mode and in branching mode. In branching mode, releases are cumulative patches based on common ancestors.

This chapter describes how to work with releases in branching mode.

⚠ **Important!**
When creating releases automatically during the import process, DROPS does not allow you to create sub-branches of releases. Only new major, level-one releases can be created automatically during the import process.

Reference
For more information about working in flat mode, refer to Releases on page 79.



Define the main properties for the new application.

App. Code : WebSite

App. Name : WebSite

Define the packaging information.

Packaging Type : Unitary

Define the release information.

Current Release :

Default Imported Version file name :

Define the main properties for the Branching Mode

Release Tree Depth : 3

Level	Release Number Template
1	v${level1}.${level2}.${level3}
2	v${level1}.${level2}.${level3}
3	v${level1}.${level2}.${level3}

Release Number	Release Description	Opened the	Status	Locked
v1.0.0	Initial release			
v1.1.0	Minor site updates			
v1.2.0	Minor site updates Q2			
v2.0.0	Major site updates v2			
v2.1.0	Minor v2 updates			
v2.1.1	bug fixes			

Result Count : 6 Page : 1/1

Figure 21: The Release Branching search view

17.1 Switching to release branching mode

The release branching mode activated inside the parent application.

> ***Reference***
>
> For more information about switching to branch mode, refer to Edit applications: Release branching mode.

17.2 Searching for release branches

Follow the subsequent steps to search for branched releases that have already been created in an application.

Step 1 Expand the application.

Step 2 Double-click on the **Releases** node to open the **Releases** search view.

Step 3 Use the search criteria sections to filter the search.

- To search for an existing release by its **Release Number** or name, enter all or part of the information in the **Identification** section.
- To search for sub-level release branches, select a release from the in **Parent Release** section.

- To search for an existing release by when it was created or the last time it was modified, select a date range in the **Creation Date** section.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.

Result The **Releases** search view displays all of the releases that have been created in the current application that meet the search criteria. The releases are nested based on the parent/child relationships established when creating the releases.

17.3 Creating branched releases

Branched releases are created manually from the **Releases** search view.

> **Important!**
> Branch releases cannot be created automatically during the import process. Only first-level, major releases can be created automatically.

Follow the subsequent steps to create a new branched release manually.

Step 1 To create a new first-level major release, ensure that no releases are selected in the **Releases** search view then either click the create icon or right-click anywhere in the view and select **Create a new release**.

To create a sub-level release, either select the upper-level parent release then click the create icon or right-click on the parent release and select **Create a new release**.

Step 2 Continue as directed inCreating releases on page 80.

Result The new branched release is available in the **Releases** search view and can be used to import artifacts.

17.4 Editing and viewing branched release details

Branched releases are edited the same way as flat releases.

> ***Reference***
> For more information about editing releases, refer to Releases on page 79.

18 Imported artifacts

Required Roles	Application Repository Manager
Required Rights (by category)	Applications Artifacts
Perspective	Configuration Management
Access	DROPS Server → Artifacts → Applications → *Application* → Repository

Chapter Summary

18.1 Searching for imported artifacts 88
18.2 Viewing an artifact's import history 89

Artifacts are individual files imported into the artifact repository which together create functional applications ready for deployment. They are imported via releases and grouped into components in order to categorize different parts of applications. Artifacts are imported following the instructions in import strategies and their versions are managed in releases.

18.1 Searching for imported artifacts

Follow the subsequent steps to search for artifacts that have already been imported into an application.

Step 1 Expand the application into which the artifact was imported.

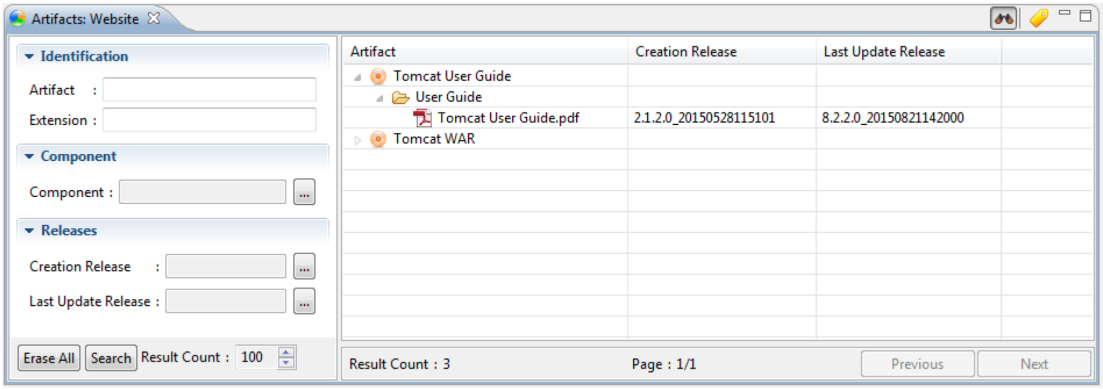
Step 2 Double-click on the **Repository** node to open the **Artifacts** search view for that application.

Step 3 When the search view is opened, the list of artifacts is empty. Use the search criteria sections to filter the search.

- To search for an existing artifact by name or file type, enter all or part of the information in the **Artifact** or **Extension** fields.
- To search for existing artifacts by **component**, click the browse icon and select the component from the list. This list is prepopulated with all of the components in the current application.
- To search for existing artifacts by release, click the corresponding browse icon. The **Creation Release** and **Last Update Release** lists are prepopulated with all of the releases that have been created for the current application.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.



Artifact	Creation Release	Last Update Release
Tomcat User Guide		
User Guide		
Tomcat User Guide.pdf	2.1.2.0_20150528115101	8.2.2.0_20150821142000
Tomcat WAR		

Result Count : 3 Page : 1/1

Figure 22: The Artifacts search view

Result The **Artifacts** search view displays all of the artifacts that have been imported into the current application that meet the search criteria.

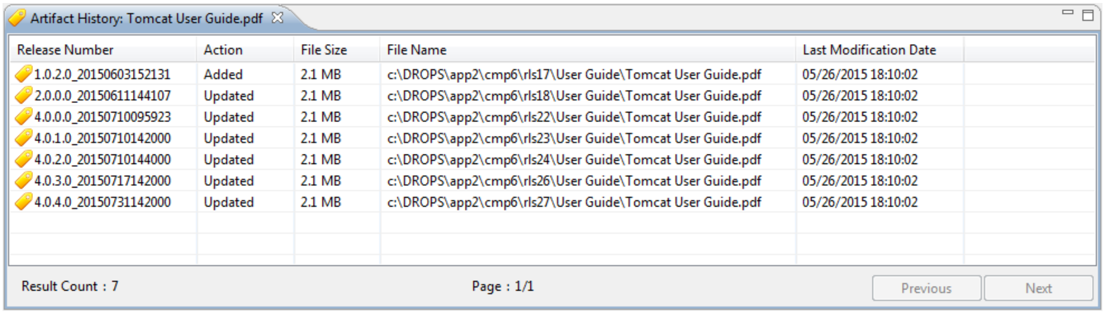
The **Artifact** column displays each component in the application that contains imported artifacts. Expanding each component displays the directory structure that DROPS copied from the artifact's source location. This directory structure is repeated in the deployment process and copied over to the target system.

The **Creation Release** and **Last Update Release** columns display the name of the first release the artifact was imported with and the most recent release that contained it, respectively.

18.2 Viewing an artifact's import history

To view an artifact's import history, select it in the **Artifacts** search view and either right-click, then select **Show History**, or click the release history icon.

The artifact's import history is displayed in the **Artifact History** view.



Release Number	Action	File Size	File Name	Last Modification Date
1.0.2.0_20150603152131	Added	2.1 MB	c:\DROPS\app2\cmp6\rls17\User Guide\Tomcat User Guide.pdf	05/26/2015 18:10:02
2.0.0.0_20150611144107	Updated	2.1 MB	c:\DROPS\app2\cmp6\rls18\User Guide\Tomcat User Guide.pdf	05/26/2015 18:10:02
4.0.0.0_20150710095923	Updated	2.1 MB	c:\DROPS\app2\cmp6\rls22\User Guide\Tomcat User Guide.pdf	05/26/2015 18:10:02
4.0.1.0_20150710142000	Updated	2.1 MB	c:\DROPS\app2\cmp6\rls23\User Guide\Tomcat User Guide.pdf	05/26/2015 18:10:02
4.0.2.0_20150710144000	Updated	2.1 MB	c:\DROPS\app2\cmp6\rls24\User Guide\Tomcat User Guide.pdf	05/26/2015 18:10:02
4.0.3.0_20150717142000	Updated	2.1 MB	c:\DROPS\app2\cmp6\rls26\User Guide\Tomcat User Guide.pdf	05/26/2015 18:10:02
4.0.4.0_20150731142000	Updated	2.1 MB	c:\DROPS\app2\cmp6\rls27\User Guide\Tomcat User Guide.pdf	05/26/2015 18:10:02

Result Count : 7 Page : 1/1

Figure 23: The Artifact History view

The **Release Number** for each individual release that contained the artifact.

The **Action** column displays the last action completed for the release (**Added** for the first import or **Updated** for when the artifact was reimported into the same release).

The **File Name** column displays the complete path to the artifact's location on the DROPS Server, which follows the template defined during high-level configuration (refer to the *DROPS Configuration Guide*).

The imported artifact's file size as well as the date the release was last modified are displayed in the corresponding columns.

Reference

For more information about artifacts in releases, refer to Releases on page 79.

Configuring the deployment process

Introduction to configuring the deployment process

DROPS makes it easy to deploy applications developed on different platforms and technologies to any target environment platform. For this, it uses a multi-platform deployment infrastructure repository, structured around environments and infrastructure items, and complemented by task-specific scripts, deployment schemas and replacement rules.

These entities, and more, must be configured in able to create functional elements to feed into deployment processes.

The configuration described in this section covers process-level configuration completed using the **Configuration Management** perspective (see Overview of the DROPS Studio on page 34).

Note
Most of the high-level configuration necessary in the DROPS Studio must be performed by a DROPS admin who has the right(s) to configure global operations.

Reference
For more information about high-level configuration, refer to the *DROPS Configuration Guide*.

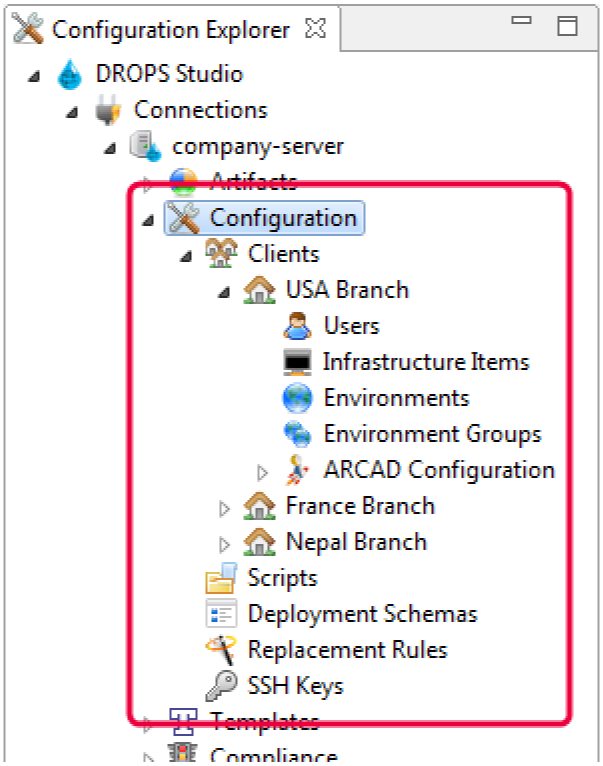


Figure 24: The Configuration Management perspective - Infrastructure Repository

The **Configuration Explorer** provides access to the entities required to set up the deployment process. This perspective is the configuration side of DROPS.

The “deployment infrastructure repository” refers to the content managed from the **Configuration** node under the currently connected DROPS Server. All of the following items are defined, configured, managed and accessed in the deployment infrastructure repository:

- Clients (see Multiple clients on page 94),
- Users (see User accounts on page 44),
- Infrastructure items (see Infrastructure items on page 108),
- Environments (see Environments on page 96),
- Environment groups (see Environment groups on page 104),
- Deployment processes (see Deployment processes on page 160),
- ARCAD Configuration (see Introduction to DROPS for ARCAD on page 308),
- Deployment plans (see Deployment plans on page 191),
- Scripts (see Scripts on page 150),
- Deployment schemas (see Deployment schemas on page 132),
- Replacement rules (see Replacement rules on page 136),
- SSH Keys (see SSH keys on page 129).

To update the content of any menu list, right-click then select **Refresh**.

19 Multiple clients

Required Roles	n/a
Required Rights (by category)	Clients
Perspective	Configuration Management
Access	DROPS Server → Configuration → Clients

The multiple-clients option is activated by users with the rights to do so. With multiple clients, the entities' search views are renamed in order to give you full visibility of your target structure.

> ***Reference***
> For more information about this high-level configuration, refer to the *DROPS Configuration Guide*.

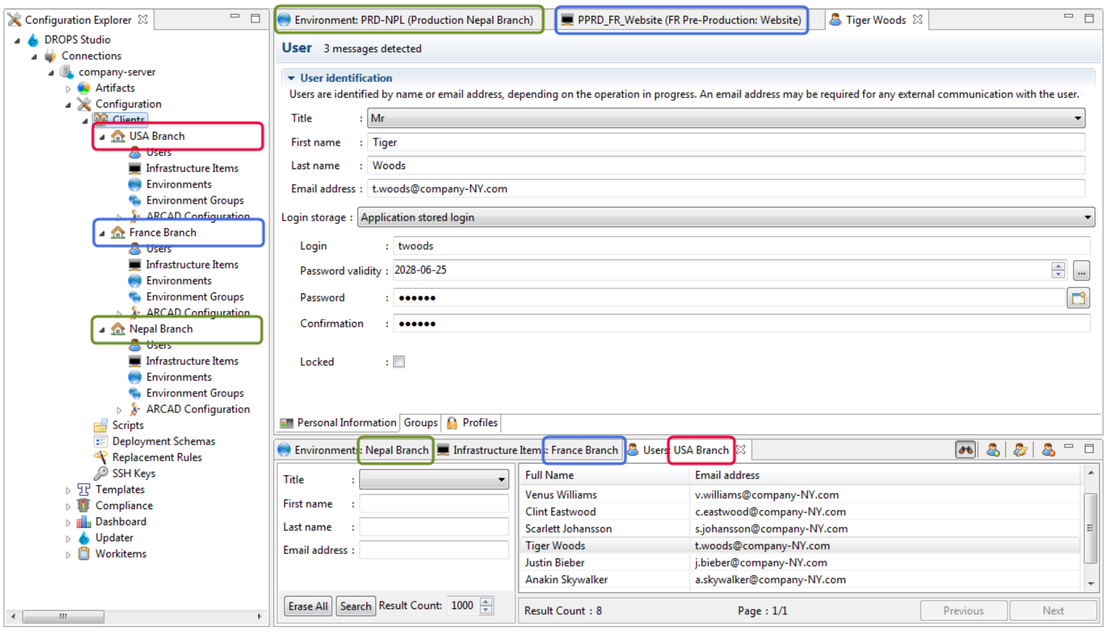
If the multiple-clients option is activated, one client is created by default called **DROPS Software**. Each client created contains its own set of users, user groups, infrastructure items, environments and environment groups and deployment plans. Having multiple sets of these elements enables you to organize them more efficiently per their deployment target.

To add additional clients, right-click on the **Clients** node and select **Create a new Client**. Define the client's code and name, then click **Finish**.

To change the names of a client, right-click on it and select **Edit**. Modify the client's code or display name in the **Client** editor. Reconnect to the DROPS Server to see the changes made.

To delete a client, right-click on it and select **Delete**. Click OK to confirm.

> ***Warning!***
> Deleted clients cannot be recovered. Any associated entities are also deleted.



User 3 messages detected

User identification

Users are identified by name or email address, depending on the operation in progress. An email address may be required for any external communication with the user.

Title : Mr

First name : Tiger

Last name : Woods

Email address : t.woods@company-NY.com

Login storage : Application stored login

Login : twoods

Password validity : 2028-06-25

Password : ●●●●●●

Confirmation : ●●●●●●

Locked : □

Title :

First name :

Last name :

Email address :

Result Count: 1000

Full Name	Email address
Venus Williams	v.williams@company-NY.com
Clint Eastwood	c.eastwood@company-NY.com
Scarlett Johansson	s.johansson@company-NY.com
Tiger Woods	t.woods@company-NY.com
Justin Bieber	j.bieber@company-NY.com
Anakin Skywalker	a.skywalker@company-NY.com

Result Count : 8 Page : 1/1

Figure 25: Views named for multiple clients

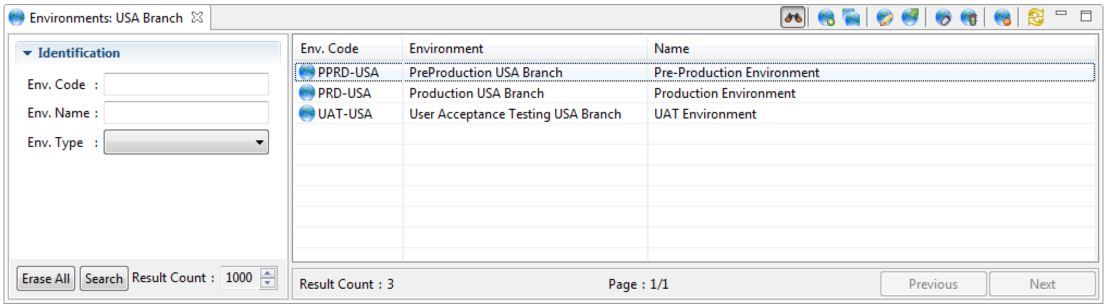
20 Environments

Required Roles	Environment Administrator, Release Manager and/or Release Operator
Required Rights (by category)	Clients Environments Infrastructure Items
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) Environments

Chapter Summary

20.1 Creating environments ... 97
20.2 Editing environment details ... 98
20.3 Adding infrastructure items ... 100
20.4 Defining allowed applications ... 101
20.5 Viewing the deployed releases ... 101
20.6 Viewing the deployment history ... 102
20.7 Changing the extension type ... 102
20.8 Executing compliance tests ... 103
20.9 Deleting environments ... 103

Environments are the highest-level container in the infrastructure repository; they define your target systems. This includes defining which applications can be deployed to certain systems, what is done during a deployment (instructions are defined in the deployment process) and to where deployed artifacts are delivered (defined in the infrastructure items). The different types of environments enable you to clearly outline where a target system is, who manages it and what it's used for.



Environments: USA Branch

Identification

Env. Code :
Env. Name :
Env. Type :

Erase All | Search | Result Count : 1000

Env. Code	Environment	Name
PPRD-USA	PreProduction USA Branch	Pre-Production Environment
PRD-USA	Production USA Branch	Production Environment
UAT-USA	User Acceptance Testing USA Branch	UAT Environment

Result Count : 3   Page : 1/1   Previous   Next

Figure 26: The Environments search view

Open the **Environments** view from the **Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

Reference

For more information about managing multiple clients, refer to Multiple clients on page 94.

Clicking on an environment also opens the **Process List**.

Note

It is recommended to open the **Process List** in a different area than the **Environments** view. If both views are tabs in the same area, it is difficult to access the **Environment** editor.

Reference

For more information about the **Process List**, refer to Deployment processes on page 160.

20.1 Creating environments

There are two ways to create environments:

1. Create a brand new environment using the **Create a new environment** wizard.
2. Duplicate an existing environments. Duplicating creates a copy of an existing environment that can be modified.

20.1.1 Create a new environment

Follow the subsequent steps to create a new environment.

Step 1 To access the **Create a new environment** wizard, either click the create icon in the **Environments** view or right-click anywhere in the view and select **Create a new environment**.

Step 2 Complete all of the fields. These values can be edited later.

1. Define the *Env. Code and *Env. Name.
2. Select the type. The *Env. Type drop-down list is prepopulated with environment types defined by the administrator in the **Preferences** menu.

   Reference

   For more information about entity types, refer to Entity types in the appendix.

3. [*Optional*] If you are working with ARCAD, select the ARCAD Extension Type.

Click **Next >** to continue.

Step 3 Select the application(s) which should be made available to be deployed on this environment. The list is automatically populated with the applications available for the current server (and client).

Step 4 Select the infrastructure item(s) to which releases should be deployed on this environment. The list is automatically populated with the items available for the current server (and client).

Step 5 Click **Finish**.

Result New environments are available in the **Environments** view after refreshing the list and can be used to configure the deployment process.

20.1.2 Duplicate an environment

Duplicating an environment is a quick way to apply an existing set of items, applications and process diagrams to a different target. You can only duplicate template environments.

> ***Example***
>
> You have similar environments that include processes made largely from templates. Instead of creating the multiple environments, create one and make it a template, then duplicate it and manually define which of the original items, apps and/or processes should be changed.

> **Note**
>
> The releases deployed for an environment are not duplicated into the new copy because they have not been deployed to that version of the target. The deployment history for these new environments is also empty.
>
> The users attributed to the original environment are carried over to the new version as well as the infrastructure items assigned to it and allowed applications. Alternatively, you can also decide to duplicate the infrastructure item if you wish.
>
> All of the original environment's deployment process diagrams and their complete configuration are also duplicated.

Follow the subsequent steps to duplicate a template environment in order to create a new one.

Step 1 To open the **Duplicate an environment** wizard, either select an existing template environment in the **Environments** view then click the duplicate icon, or right-click on it and select **Duplicate**.

Step 2 Define the new environment's *Env. Code and *Env. Name. These values can be edited later but must be unique.

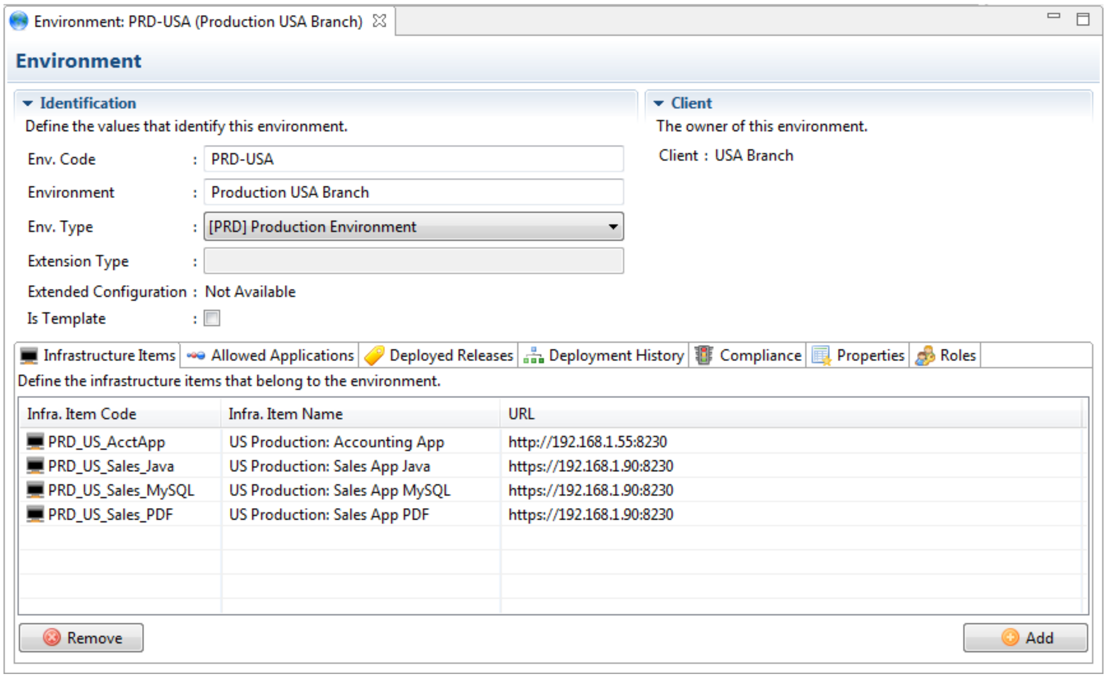
Step 3 Tick the **Create new infra.** box to create duplicate infrastructure items.

Step 4 Click **Finish**.

Result New environments are available in the **Environments** view after refreshing the list and can be used to configure the deployment process.

20.2 Editing environment details

To edit environment details, either right-click on it in the **Environments** view then select **Edit**, select the item then click the edit icon or double-click it. The **Environment** editor is opened where all of the entity's details are managed.



Define the values that identify this environment.

Env. Code : PRD-USA

Environment : Production USA Branch

Env. Type : [PRD] Production Environment

Extension Type :

Extended Configuration : Not Available

Is Template : □

The owner of this environment.

Client : USA Branch

Define the infrastructure items that belong to the environment.

Infra. Item Code	Infra. Item Name	URL
PRD_US_AcctApp	US Production: Accounting App	http://192.168.1.55:8230
PRD_US_Sales_Java	US Production: Sales App Java	https://192.168.1.90:8230
PRD_US_Sales_MySQL	US Production: Sales App MySQL	https://192.168.1.90:8230
PRD_US_Sales_PDF	US Production: Sales App PDF	https://192.168.1.90:8230

Figure 27: The Environment editor

The information that defines environments can be edited at any time by users with the rights to do so.

> ***Reference***
>
> For more information about the different actions managed from the tabs in the **Environment** editor that are not described in this chapter, refer to:
>
> - Properties on page 412
> - Assigning entity-level roles on page 54

The environment's parent entity is displayed in the **client** section.

*Env. Code

A unique string that identifies environments. This label is used throughout DROPS to select the entity.

*Env. Name

A label used to give additional information about the environment. This label is used throughout DROPS to select the entity.

*Env. Type

A value used to assign a type to the environment. This drop-down list is prepopulated with environment types defined by the administrator in the **Preferences** menu.

> ***Reference***
>
> For more information about entity types, refer to Entity types in the appendix.

Extension Type

A read-only value set to the environment's extension type selected when it was created.

Extended Configuration

Some environment extension types require extended configuration. If the current extension type does, this field will be a link to the type-specific configuration editor.

> ***Reference***
> For more information about the extended configuration editors, refer to Environment type extended configuration in the appendix.

Is Template

Tick this checkbox to transform the environment into a template. Template environments cannot be used to deploy content but may be duplicated to create other active environments that contain the pre-defined elements in the template.

You cannot select a template environment in a deployment instance to use it as a target for your deployment but you can filter the list of environments using a web service to select only the templates. Template environments cannot be added to environment groups.

> ⚠ **Important!**
> If an environment that is already part of an environment group is changed to be a template environment, NONE of the environments in the group can be used in any deployment instance until the template environment is either removed from the group or changed to not be a template any longer.

Template environments are indicated by the template environment icon in the **Environments** search view.

Save the changes (, Ctrl+S or **File** > **Save**).

20.3 Adding infrastructure items

To make the connection between infrastructure items and the target environment on which they run, add them to an environment. Populating the **Infrastructure Items** tab in the **Environment** editor with all of the items to which you can deploy content enables you to later select those items in deployment process diagrams associated with the environment.

> ⚠ **Important!**
> An environment must be linked to at least one infrastructure item in order to create a functional deployment process.

From the **Infrastructure Items** tab in the **Environment** editor, click the **Add** button. The **Select an infrastructure item** dialog is prepopulated with the items available for the current server (and client).

To remove an item from the list, select it in the manager and click the **Remove** button.

Save the changes (, Ctrl+S or **File** > **Save**).

20.4 Defining allowed applications

To make the connection between artifacts and the target environment to which they can be deployed, add their parent application to an environment. Populating the **Allowed applications** tab in the **Environment** editor with all of the applications that can be deployed to it enables you to later select those applications in deployment process diagrams associated with the environment.

> ⚠ **Important!**
> An environment must be linked to at least one application in order to create a functional deployment process.

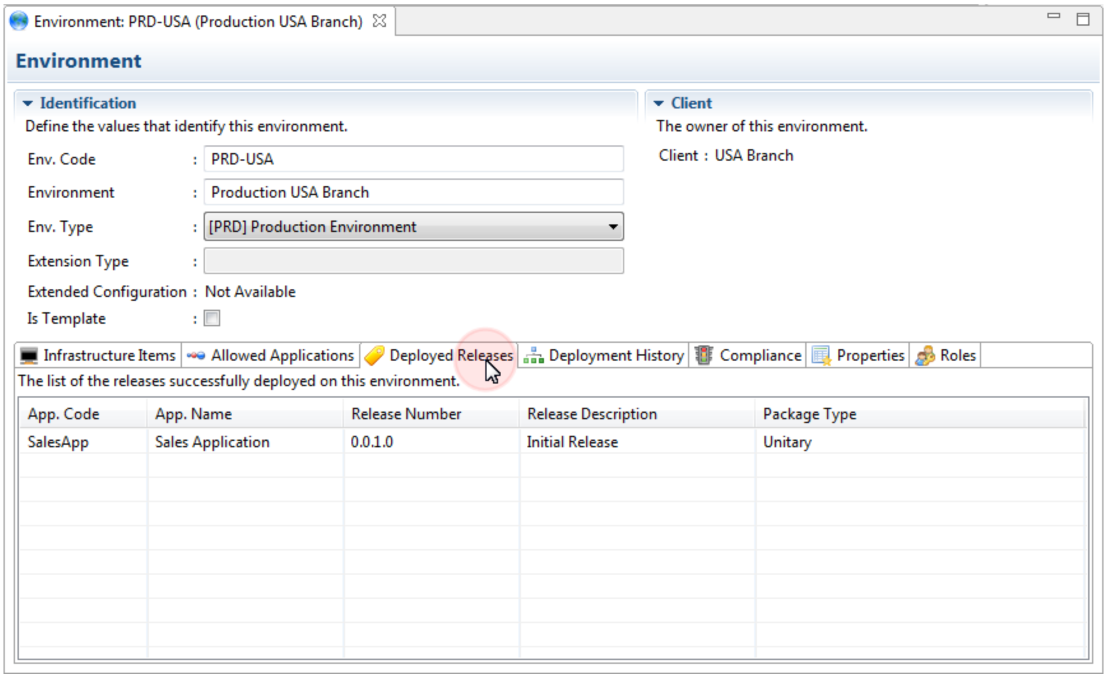
From the **Allowed applications** tab in the **Environment** editor, click the **Add** button. The **Select an application** dialog is prepopulated with the applications available for the current server.

To remove an application from the list, select it in the manager and click the **Remove** button.

Save the changes (, Ctrl+S or **File** > **Save**).

20.5 Viewing the deployed releases

The read-only **Deployed Releases** tab in the **Environment** editor displays the complete list of releases that have already been deployed on the environment.



Define the values that identify this environment.

Env. Code	PRD-USA
Environment	Production USA Branch
Env. Type	[PRD] Production Environment
Extension Type	
Extended Configuration	Not Available
Is Template	□

The owner of this environment.

Client : USA Branch

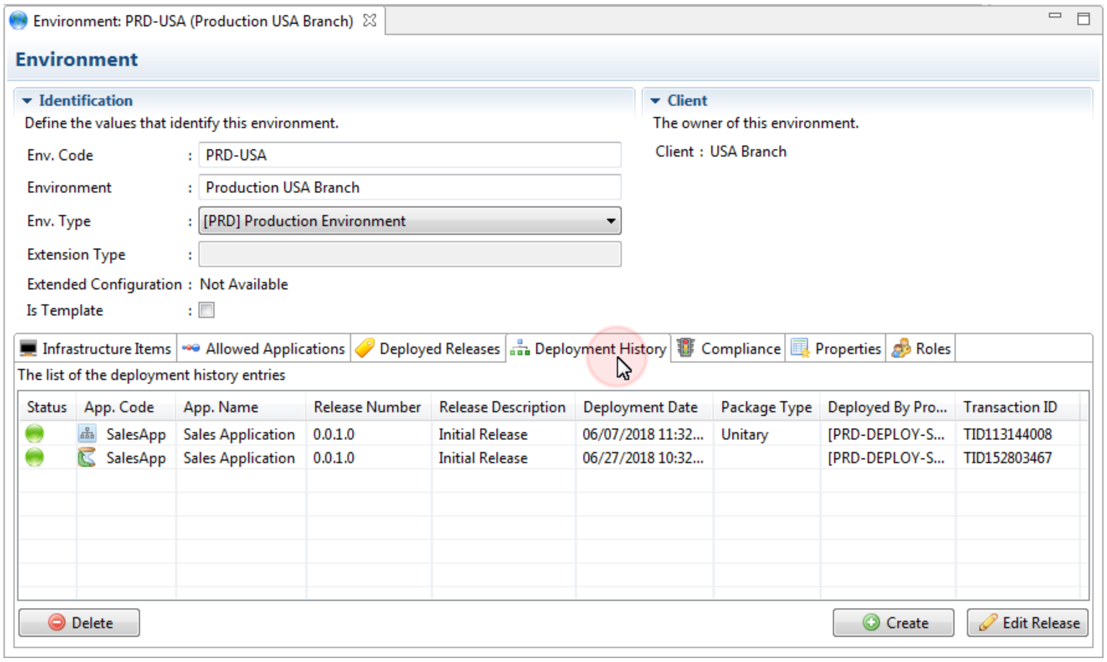
The list of the releases successfully deployed on this environment.

App. Code	App. Name	Release Number	Release Description	Package Type
SalesApp	Sales Application	0.0.1.0	Initial Release	Unitary

Figure 28: The Deployed Releases tab in the environment editor

20.6 Viewing the deployment history

The **Deployment History** tab in the **Environment** editor displays the complete history of the deployment process instances executed for the current environment.



Define the values that identify this environment.

Env. Code : PRD-USA
Environment : Production USA Branch
Env. Type : [PRD] Production Environment
Extension Type :
Extended Configuration : Not Available
Is Template : □

The owner of this environment.

Client : USA Branch

The list of the deployment history entries

Status	App. Code	App. Name	Release Number	Release Description	Deployment Date	Package Type	Deployed By Pro...	Transaction ID
●	SalesApp	Sales Application	0.0.1.0	Initial Release	06/07/2018 11:32...	Unitary	[PRD-DEPLOY-S...	TID113144008
●	SalesApp	Sales Application	0.0.1.0	Initial Release	06/27/2018 10:32...		[PRD-DEPLOY-S...	TID152803467

Figure 29: The Deployment History tab in the environment editor

The table displays complete information about the deployment including the final result, the application and release deployed, the date and time the deployment was finished, the package type and the deployment process used to deliver.

Select a deployment and click the **Edit Release** button or double-click it to open the associated **Release** editor.

In addition to the automatically-generated references to deployments, entries in this view can be created manually. Manually-generated history entries are not linked to any deployment process and do not reference any deployment process instance. They are for information only. The **Deployed By Process** column displays the term ***MANUAL** for any history entry that is not linked to a DROPS deployment. To create a deployment history reference manually, click the **Create** button and define the required application, release and package type.

To remove a deployment from the list, select it in the manager and click the **Delete** button. Deleting an item from the list in the Deployment History tab does not affect the **Deployment Process Instances** search view.

20.7 Changing the extension type

To change the Extension Type of an environment, either right-click on it in the **Environments** view then select **Change Extension Type**, or select the it then click the change type icon.

The drop-down list to select the new extension type is prepopulated with environment extension types available. Click **OK** then close the editor and refresh the view. When the environment is opened again, the change is visible.

> ***Warning!***
> If the extended configuration for an environment has already been defined, changing its extension type will delete **all** of the information.

> ***Reference***
> For more information about this, refer to Environment type extended configuration on page 458 in the appendix.

20.8 Executing compliance tests

Perform tests on entity to ensure that the target environment complies to specific requirements. Compliance rules are a complementary tool, available independently of the main DROPS process.

> ***Reference***
> For more information about compliance rules, refer to Compliance rules on page 140.

20.9 Deleting environments

> ***Warning!***
> Deleted environments cannot be accessed or recovered.

To delete an environment, either right-click on it in the **Environments** view, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the environment.

21 Environment groups

Required Roles	Environment Group Administrator
Required Rights (by category)	Clients Environment Groups
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) Environment Groups

Chapter Summary

21.1 Accessing environment groups 104
21.2 Creating environment groups 105
21.3 Editing environment group details 105
21.4 Adding environments to groups 106
21.5 Deleting environment groups 107

Environment groups are sets of environments that share the same application. They are used to create multiple deployment process instances, which all deploy the same release, at the same time.

Important!

The deployment instance(s) created using the environment group must always deploy the same release so it is important that all of the environments in a group allow the same application(s) and each have a deployment process defined to deploy the shared application(s).

Important!

If an environment that is already part of an environment group is changed to be a template environment, NONE of the environments in the group can be used in any deployment instance until the template environment is either removed from the group or changed to not be a template any longer.

Note

It is recommended to define a default deployment process for each environment in a group.

Reference

For more information about ensuring the above parameters are defined for all of your environments, refer to Defining allowed applications on page 101 and Deployment processes on page 160.

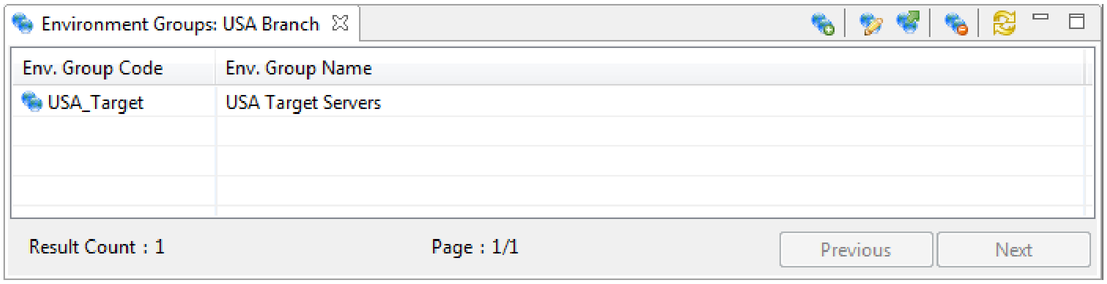
21.1 Accessing environment groups

To access the **Environment Groups** view, select the environment groups node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

> ***Reference***
> For more information about managing multiple clients, refer to Multiple clients on page 94.

The **Environment Groups** view displays all of the groups defined for the current server (and for the selected client, if applicable).



Environment Groups: USA Branch

Env. Group Code	Env. Group Name
USA_Target	USA Target Servers

Result Count : 1

Page : 1/1

Figure 30: The Environment Groups search view

21.2 Creating environment groups

Follow the subsequent steps to create a new environment group.

Step 1 To access the **Create a new environment group** wizard, either click the create icon in the **Environment Groups** view or right-click anywhere in the view and select **Create a new environment group**.

Step 2 Define the * Env. Group Code and * Env. Group Name.

Click **Next >** to continue.

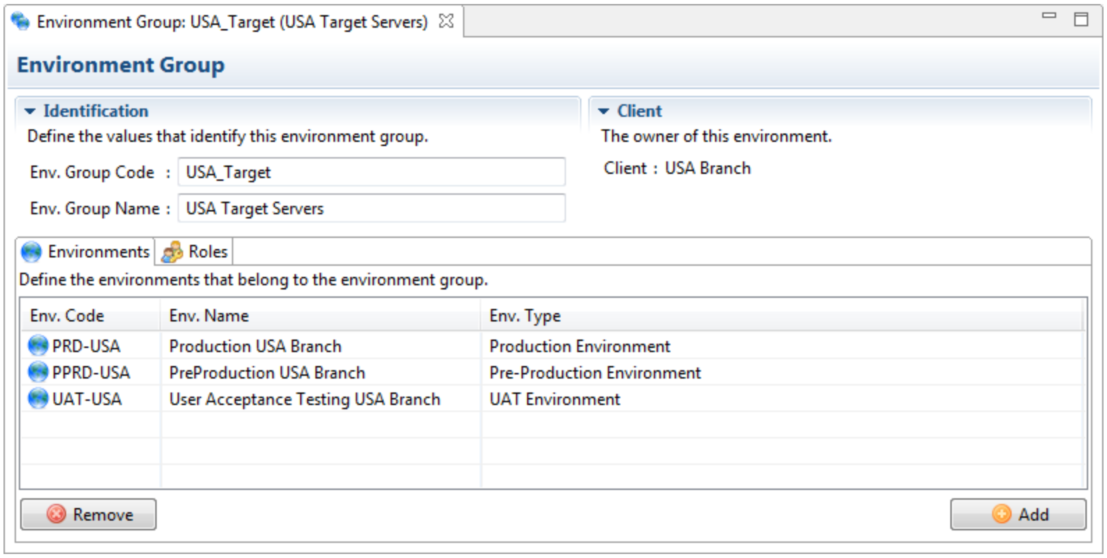
Step 3 Select the environments to include in the group. The list is automatically populated with the environments available for the current server (and client).

Step 4 Click **Finish**.

Result New environment groups are available in the **Environment Groups** view and can be used to configure the deployment process.

21.3 Editing environment group details

To edit environment group details, either right-click on it in the **Environment Groups** view then select **Edit**, select the item then click the edit icon or double-click it. The **Environment Group** editor is opened where all of the entity's details are managed.



Environment Group: USA_Target (USA Target Servers)

Environment Group

Identification
Define the values that identify this environment group.

Env. Group Code : USA_Target
Env. Group Name : USA Target Servers

Client
The owner of this environment.

Client : USA Branch

Define the environments that belong to the environment group.

Env. Code	Env. Name	Env. Type
PRD-USA	Production USA Branch	Production Environment
PPRD-USA	PreProduction USA Branch	Pre-Production Environment
UAT-USA	User Acceptance Testing USA Branch	UAT Environment

Figure 31: The Environment Group editor

The information that defines environment groups can be edited at any time by users with the rights to do so.

Reference

For more information about the different actions managed from the tabs in the **Environment Group** editor that are not described in this chapter, refer to:

- Assigning entity-level roles on page 54

The environment group's parent entity is displayed in the **Client** section.

Field	Description
* **Env. Group Code**	A unique string that identifies environment groups. This label is used throughout DROPS to select the entity.
* **Env. Group Name**	A label used to give additional information about the environment group. This label is used throughout DROPS to select the entity.

Table 4: Edit environment groups

Save the changes (💾, Ctrl+S or **File** > **Save**).

21.4 Adding environments to groups

To manage the list of environments in an environment group, add or remove them from the **Environments** tab.

From the **Environments** tab in the **Environment Group** editor, click the **Add** button. The **Select an environment** dialog is prepopulated with the environments available for the current server (and client).

To remove an environment, select it in the manager and click the **Remove** button.

Save the changes (, Ctrl+S or **File** > **Save**).

21.5 Deleting environment groups

> ***Warning!***
> Deleted environment groups cannot be accessed or recovered.

To delete an environment group, either right-click on it in the **Environment Groups** view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the environment group.

22 Infrastructure items

Required Roles	Infrastructure Item Administrator, Manager and/or Contributor
Required Rights (by category)	Clients Infrastructure Items
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) Infrastructure Items

Chapter Summary

22.1 Creating infrastructure items 109
22.2 Editing infrastructure items 115
22.3 Remote SSH options 121
22.4 Viewing the deployed releases 121
22.5 Viewing the deployment history 121
22.6 Changing the type of an infrastructure item 122
22.7 Changing the OS family of an infrastructure item 122
22.8 Downloading remote agent logs 122
22.9 Executing compliance tests 122
22.10 Provisioning infrastructure items 123
22.11 Instantiating infrastructure items 123
22.12 Managing an infrastructure item’s remote installation properties 124
22.13 Testing the connection to an infrastructure item (agent) 125
22.14 Deleting infrastructure items 128

Infrastructure items represent individual services on target systems. They connect DROPS to the target system via the DROPS Agent. Agents are installed on these deployment targets that run technical layers which can often support one or more services. Infrastructure items point to individual services. Because multiple services can point to the same agent on a shared machine, it is possible to create multiple infrastructure items in DROPS that each connect to a different service on the same system.

DROPS licenses may be based on the number of infrastructure items created because they represent the size, deployment workload and overall deployment needs of your company.

It is recommended to create one infrastructure item for each of the various services you may deploy to, no matter how many run on the same system. Doing this will enable you to manage the correspondence between infrastructure items and components in the deployment process diagram accurately. In the process diagram, infrastructure items and components are matched one-to-one, therefore, having unique infrastructure items for each service will ensure components are deployed accurately to only the item required and not to multiple services on the system.

It is also recommended to create your infrastructure items before installing the DROPS Agent.

An infrastructure item can be added to multiple environments and an environment can contain multiple infrastructure items.

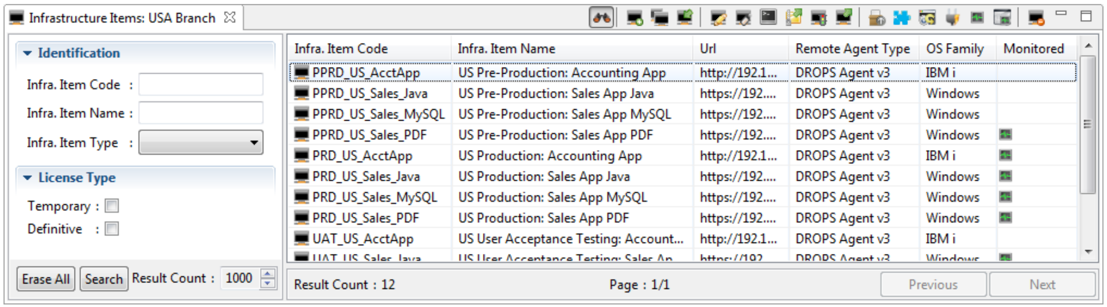
Note

If an infrastructure item is not correctly connected to a DROPS Agent or the license access is restricted, when it is used in a deployment process, the process will be suspended.

Reference

For more information about suspended processes, refer to Suspended actions on page 294.

For more information about the DROPS Agent, refer to the DROPS *Configuration* and *Installation Guides*.



Infrastructure Items: USA Branch

Identification

Infra. Item Code :

Infra. Item Name :

Infra. Item Type :

License Type

Temporary : □

Definitive : □

Erase All | Search | Result Count : 1000

Infra. Item Code	Infra. Item Name	Url	Remote Agent Type	OS Family	Monitored
PPRD_US_AcctApp	US Pre-Production: Accounting App	http://192.1...	DROPS Agent v3	IBM i	
PPRD_US_Sales_Java	US Pre-Production: Sales App Java	https://192....	DROPS Agent v3	Windows	
PPRD_US_Sales_MySQL	US Pre-Production: Sales App MySQL	https://192....	DROPS Agent v3	Windows	
PPRD_US_Sales_PDF	US Pre-Production: Sales App PDF	https://192....	DROPS Agent v3	Windows	🟩
PRD_US_AcctApp	US Production: Accounting App	http://192.1...	DROPS Agent v3	IBM i	🟩
PRD_US_Sales_Java	US Production: Sales App Java	https://192....	DROPS Agent v3	Windows	🟩
PRD_US_Sales_MySQL	US Production: Sales App MySQL	https://192....	DROPS Agent v3	Windows	🟩
PRD_US_Sales_PDF	US Production: Sales App PDF	https://192....	DROPS Agent v3	Windows	🟩
UAT_US_AcctApp	US User Acceptance Testing: Account...	http://192.1...	DROPS Agent v3	IBM i	
UAT_US_Sales_Java	US User Acceptance Testing: Sales Ap	https://192	DROPS Agent v3	Windows	🟩

Result Count : 12 | Page : 1/1 | Previous | Next

Figure 32: The Infrastructure Items search view

The **Infrastructure Items** search view is accessed from the **Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

Reference

For more information about managing multiple clients, refer to Multiple clients on page 94.

22.1 Creating infrastructure items

There are several ways to create infrastructure items:

1. Create new infrastructure items manually using the **Create a new infrastructure item** wizard.
2. Duplicate an existing infrastructure item and modify its definition and properties.
3. Import infrastructure items by uploading a .csv file that contains the required minimum information. This will mass-produce all of the items you need.

22.1.1 Create new infrastructure items

Follow the subsequent steps to create a new infrastructure item.

Step 1 To access the **Create a new infrastructure item** wizard, either click the create icon in the **Infrastructure Items** search view or right-click anywhere in the view and select **Create a new infrastructure item**.

Step 2 Complete all of the fields. These values can be edited later.

1. Define the new item’s *Infra. Item Code and *Infra. Item Name.
2. Select the item's Infra. Item Type. The drop-down list is prepopulated with infrastructure item types defined by the administrator in the **Preferences** menu.

> ***Reference***
> For more information about entity types, refer to Entity types in the appendix.

3. Select the *Remote Agent Type that has been or will be installed on the item. This decision will affect the options available in the **Protocol** field. There are two options:
   - DROPS Agent: relies on HTTP(S) connection to the DROPS Agent.
   - Agentless: there is no target agent - the connection is secured by SSH or SSL.
4. Select the *OS Family that the agent is running on. There are three options:
   - Windows
   - Linux
   - IBM i
5. Select the Protocol to use to connect to the target machine and its Host name & Port and port.
6. If you selected to use an **SSH Protocol**, select the pre-defined SSH key from the list.
7. Enter the Login & Password and password required to connect to the infrastructure item.

Step 3 Click **Finish**.

Result New infrastructure items are available in the **Infrastructure Items** search view and are ready to be configured and added to environments.

> ⚠ **Important!**
> It is important to continue editing a new infrastructure item's information right away to complete the configuration for the infrastructure item type.

> ***Reference***
> For more information about configuring Items, refer to Infrastructure item type extended configuration in the appendix.

22.1.2 Duplicate an existing infrastructure item

Duplicating an infrastructure item is a quick way to create a new infrastructure item with similar settings and properties. Follow the subsequent steps to duplicate an existing infrastructure item.

Step 1 Select an existing infrastructure item from the **Infrastructure Items** search view, and click the duplicate icon or right-click on the item and select **Duplicate** from the list to open the duplication wizard.

> **Note**
> If the duplicated infrastructure item does not appear in the list, refresh the search view.

Step 2 Modify the definitions and properties of the duplicated infrastructure item in the wizard. These values can be edited later. Click **Finish**.

> ***Reference***
> For more information about infrastructure item edition, refer to theEditing infrastructure items on page 115.

Result New infrastructure items are available in the **Infrastructure Items** search view and are ready to be configured and added to environments.

22.1.3 Import infrastructure items

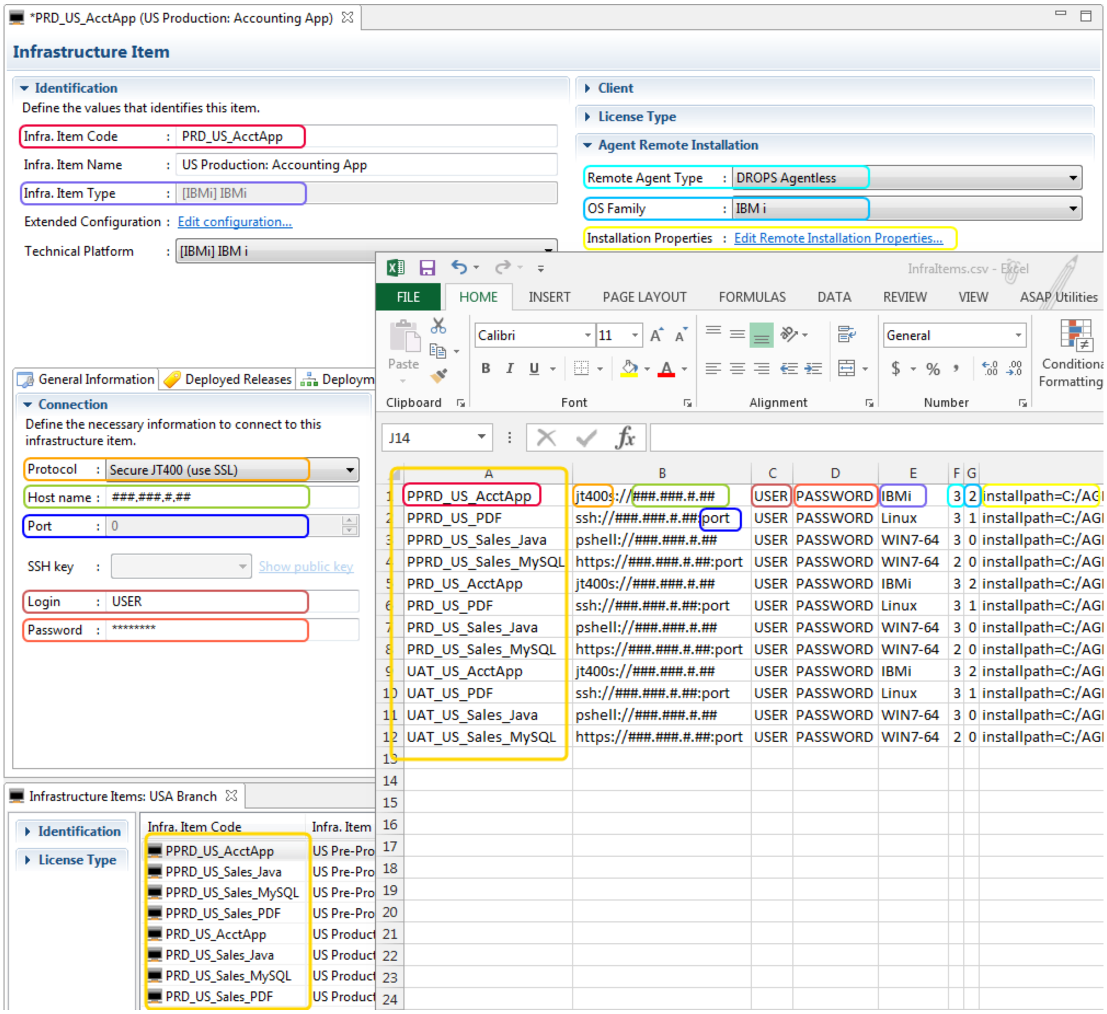
Follow the subsequent steps to create multiple infrastructure items automatically by importing a pre-configured *.csv* file.

Step 1 Create the *.csv* file to upload.

The *.csv* file to upload *must* contain a minimum amount of information for each infrastructure item to create. The details should be configured in a specific order - the information corresponding to each required field must be included in the correct column in order for the upload process to be successful.

The details defined in the *.csv* file will be automatically transferred to each item DROPS creates. The corresponding fields will be automatically populated. The columns can be remapped but the default order is as follows.

> **Important!**
> Do not include a header row in the *.csv* file.



*PRD_US_AcctApp (US Production: Accounting App)

Infrastructure Item

Identification

Define the values that identifies this item.

Infra. Item Code : PRD_US_AcctApp
Infra. Item Name : US Production: Accounting App
Infra. Item Type : [IBMi] IBMi
Extended Configuration : Edit configuration...
Technical Platform : [IBMi] IBM i

Client
License Type
Agent Remote Installation

Remote Agent Type : DROPS Agentless
OS Family : IBM i
Installation Properties : Edit Remote Installation Properties...

General Information | Deployed Releases | Deploym

Connection

Define the necessary information to connect to this infrastructure item.

Protocol : Secure JT400 (use SSL)
Host name : ###.###.#.##
Port : 0
SSH key : Show public key
Login : USER
Password : ********

Infrastructure Items: USA Branch

Identification
License Type

Infra. Item Code	Infra. Item
PPRD_US_AcctApp	US Pre-Pro
PPRD_US_Sales_Java	US Pre-Pro
PPRD_US_Sales_MySQL	US Pre-Pro
PPRD_US_Sales_PDF	US Pre-Pro
PRD_US_AcctApp	US Product
PRD_US_Sales_Java	US Product
PRD_US_Sales_MySQL	US Product
PRD_US_Sales_PDF	US Product

InfraItems.csv - Excel

J14

	A	B	C	D	E	F	G	
1	PPRD_US_AcctApp	jt400s://###.###.#.##	USER	PASSWORD	IBMi	3	2	installpath=C:/AG
2	PPRD_US_PDF	ssh://###.###.#.##:port	USER	PASSWORD	Linux	3	1	installpath=C:/AGI
3	PPRD_US_Sales_Java	pshell://###.###.#.##	USER	PASSWORD	WIN7-64	3	0	installpath=C:/AGI
4	PPRD_US_Sales_MySQL	https://###.###.#.##:port	USER	PASSWORD	WIN7-64	2	0	installpath=C:/AGI
5	PRD_US_AcctApp	jt400s://###.###.#.##	USER	PASSWORD	IBMi	3	2	installpath=C:/AGI
6	PRD_US_PDF	ssh://###.###.#.##:port	USER	PASSWORD	Linux	3	1	installpath=C:/AGI
7	PRD_US_Sales_Java	pshell://###.###.#.##	USER	PASSWORD	WIN7-64	3	0	installpath=C:/AGI
8	PRD_US_Sales_MySQL	https://###.###.#.##:port	USER	PASSWORD	WIN7-64	2	0	installpath=C:/AGI
9	UAT_US_AcctApp	jt400s://###.###.#.##	USER	PASSWORD	IBMi	3	2	installpath=C:/AGI
10	UAT_US_PDF	ssh://###.###.#.##:port	USER	PASSWORD	Linux	3	1	installpath=C:/AGI
11	UAT_US_Sales_Java	pshell://###.###.#.##	USER	PASSWORD	WIN7-64	3	0	installpath=C:/AGI
12	UAT_US_Sales_MySQL	https://###.###.#.##:port	USER	PASSWORD	WIN7-64	2	0	installpath=C:/AGI

Figure 33: Example of .csv file used to import multiple infrastructure items

Column A (0)(*the first column*): the infrastructure item's *Infra. Item Code.

Column B (1): the Protocol://Host name & Port:Host name & Port that points to the target system on which the DROPS Agent is installed.

> ⚠ **Important!**
>
> This URL must follow the syntax exactly:
> <protocol>://<hotname>:<port>
>
> JT400, JT400s and PowerShell do not use ports. If you use these protocols, the syntax is:
> <protocol>://<hotname>

The syntax for each protocol is:

Agent

Agent type	Syntax to enter in the URL
HTTP	**http**://hostname:port
HTTP(s)	**https**://hostname:port

Table 5: Syntax for agent protocols

Agentless

OS	Agent type	Syntax to enter in the URL
Linux	SSH	**ssh**://hostname:port
Windows	PowerShell ≥4	**pshell**://hostname
IBM i	SSH	**ssh**://hostname:port
	JT400	**jt400**://hostname
	Secure JT400 via SSL	**jt400s**://hostname

Table 6: Syntax for agentless protocols

Column C (2): the Login & Password required to connect to the target system.

Column D (3): the Login & Password required to connect to the target system.

Column E (4): the infrastructure item's Infra. Item Type. Enter only the item type code, which is defined when entity types are created in the **Preferences** menu.

> ***Reference***
> For more information about entity types, refer to Entity types in the appendix.

Column F (5): the *Remote Agent Type that is running on the remote system. There are two types of agents. Enter the *code* that corresponds to the type that the item connects to on the target system. The codes are:

- **2** for the DROPS Agent
- **3** for the agentless solution

Column G (6): the *OS Family that the remote system is running on. Enter the *code* that corresponds to the OS on which the DROPS Agent is running. The codes are:

- **0** for Windows (*default*)
- **1** for Linux
- **2** for IBM i

Column H (7): [*Optional*] the remote Installation Properties. Enter the properties following this pattern:

installpath=*<DROPS Agent install location>*;storagepath=*<DROPS Agent opt/dropsagent/storage path>*;installuser=*<user>*;installuserpassword=*<password>*

Each key=value pair must be separated by a semi-colon (;). The following properties are shared by all OS families:

- installpath: installation path
- storagepath: the local repository storage area
- installuser: the user used or to use during the installation process
- installuserpassword: the installation user's password

The OS-specific properties to add depending on your target are:

Windows

- serviceuser: the user profile that will run the agent
- serviceuserpassword: the service user's password

Linux

- serviceuser: the user profile that will run the agent

IBM i

- library: target IBM i installation library
- iasp: the installation library's iASP

Step 2 Import the *.csv* file.

To open the *.csv* import wizard, either right-click anywhere in the **Infrastructure Items** search view and select **Import from a .csv file** or click the *.csv* import icon.

Step 3 Complete all of the fields.

> ***Warning!***
> These fields cannot be edited later.

1. Enter the exact path to the *.csv* file created in Step 1. Click the browse button to navigate to the file.
2. If you did *not* change the order of the columns as suggested in Step 1, do not modify the fields in the **Column Index Mapping** section.

> **Note**
> If you *changed* the order of the columns in the *.csv* file, modify the corresponding fields' indicator(s) in the **Column Index Mapping** section.
>
> '0' corresponds to the first column on the far-left (*column A*), '1' to the next column to the right (*column B*) and so on.

Step 4 Click **Finish** to launch the import.

Result All of the infrastructure items listed in the *.csv* file are created. The number of the items created that are automatically active depends on your license key. If the *.csv* file contains more items than your license allows, the extra items will be created but they will be temporary.

Example

If there are 45 rows with accurate and complete information in the *.csv* and you have a license that permits 50 items, 45 items will be created. If you have a license that only permits 40 items, all 45 items will be created, but the final 5 in the *.csv* will be temporary items. If your license does not allow the full number of items to create, an email is sent to the user who uploaded the *.csv* file.

Reference

For more information about temporary items, refer to Temporary Infrastructure Item.

New infrastructure items are available in the **Infrastructure Items** search view and are ready to be configured and added to environments.

Important!

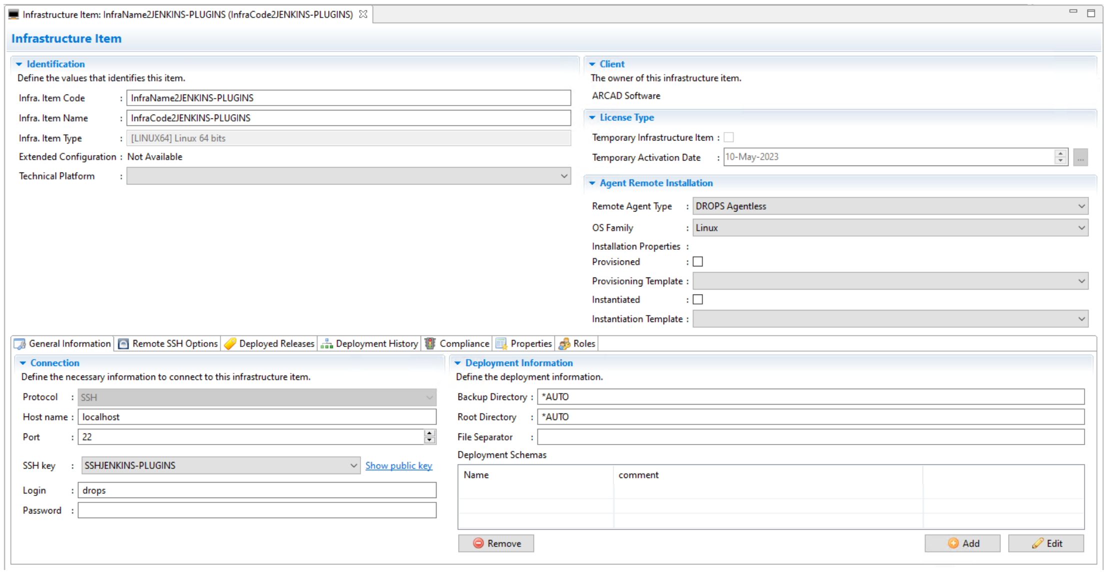
It is important to continue editing a new infrastructure item's information right away to complete the configuration for the infrastructure item type.

Reference

For more information about configuring Items, refer to Infrastructure item type extended configuration in the appendix.

22.2 Editing infrastructure items

To edit infrastructure item details, either right-click on it then select **Edit**, select the item then click the edit icon or double-click it. The **Infrastructure Item** editor is opened where all of the entity's details are managed.



Infrastructure Item: InfraName2JENKINS-PLUGINS (InfraCode2JENKINS-PLUGINS)

Infrastructure Item

Identification

Define the values that identifies this item.

Infra. Item Code : InfraName2JENKINS-PLUGINS

Infra. Item Name : InfraCode2JENKINS-PLUGINS

Infra. Item Type : [LINUX64] Linux 64 bits

Extended Configuration : Not Available

Technical Platform :

Client

The owner of this infrastructure item.

ARCAD Software

License Type

Temporary Infrastructure Item : □

Temporary Activation Date : 10-May-2023

Agent Remote Installation

Remote Agent Type : DROPS Agentless

OS Family : Linux

Installation Properties :

Provisioned : □

Provisioning Template :

Instantiated : □

Instantiation Template :

Connection

Define the necessary information to connect to this infrastructure item.

Protocol : SSH

Host name : localhost

Port : 22

SSH key : SSHJENKINS-PLUGINS

Login : drops

Password :

Deployment Information

Define the deployment information.

Backup Directory : *AUTO

Root Directory : *AUTO

File Separator :

Deployment Schemas

Name	comment

Figure 34: The Infrastructure Item editor

The information that defines infrastructure items can be edited at any time by users with the rights to do so.

Reference

For more information about the different actions managed from the tabs in the **Infrastructure Item** editor that are not described in this chapter, refer to:

- Properties on page 412
- Assigning entity-level roles on page 54

For more information about an item's extended configuration, refer to Infrastructure item type extended configuration in the appendix.

Note

The values entered in the **Identification** and **Connection** sections can be referenced by Substitution variables.

*Infra. Item Code

A unique string that identifies infrastructure items. This label is used throughout DROPS to select the entity.

*Infra. Item Name

A label used to give additional information about the infrastructure item. This label is used throughout DROPS to select the entity.

Infra. Item Type

A read-only value set to the infrastructure item type selected when the item was created.

> ***Reference***
>
> For more information about infrastructure item types, refer to Entity types in the appendix.

Extended Configuration

Some infrastructure item types require extended configuration. If the item's current type does, this field will be a link to the type-specific configuration editor.

> ***Reference***
>
> For more information about extended configuration editors, refer to Infrastructure item type extended configuration in the appendix.

Technical Platform

[*Optional*] Select the type of OS that the infrastructure item points to. The drop-down list is prepopulated with the platforms defined by an administrator in the **Preferences** menu.

Technical platforms refer to the service to which an infrastructure item points. Knowing the type of service Item points to is helpful when selecting platform-specific scripts when building the deployment process.

> ***Reference***
>
> For more information about Technical Platforms, refer to the *DROPS Configuration Guide*.

The item's parent entity is displayed in the **Client** section.

The **License Type** section displays critical information concerning the current server's Activation Key.

> **Note**
>
> To view the number of permanent and temporary License Units available, click the **Configuration** node in the **Preferences** menu (*right-click on the* DROPS Server → Preferences → Configuration).

Temporary Infrastructure Item

If this checkbox is ticked, the activation key associated with the current DROPS Server does not have any more available License Units.

Temporary infrastructure items can only be created if every license unit in the current activation key's is consumed. The number of temporary license units available depends on the number of purchased License Units that the current activation key contained.

If no activation key has ever been registered for the current DROPS Server, only five (5) temporary infrastructure items can be created and used for a limited period of time (up to one month). This enables you to use DROPS while your activation key is being generated.

A temporary infrastructure item becomes definitive as soon as a new license unit is available (as soon as a new license unit pack is registered).

Temporary Activation Date

Temporary infrastructure items are available for one month starting from the current date (the date the item was created).

> **Note**
>
> When a deployment involves a temporary infrastructure item and if its temporary activation date is over, when it is used in a deployment process, the process will be suspended. If the temporary activation date is not over, a warning will be produced in the deployment log to remind you that you are using a temporary infrastructure item.

> ***Reference***
>
> For more information about suspended processes, refer to Suspended actions on page 294.

The **Agent Remote Installation** section displays important information about the DROPS Agent on the infrastructure item and the target machine's OS.

*Remote Agent Type

Select the type of DROPS Agent that is running or will be installed on the remote system to which you are connecting.

If the DROPS Server must connect to an agent using HTTP(S), select **DROPS Agent**.

If there is no target agent because you are running the agentless solution, select **DROPS Agentless**.

*OS Family

The type of OS on which the DROPS Agent is installed. The OS can be changed in the **Infrastructure Items** search view. Depending on the OS family defined for the item, the Installation Properties and the security Protocol options will change.

> ***Warning!***
>
> If you change the OS Family for an existing infrastructure item, any install properties or connection protocol details defined will be lost.

Installation Properties

The properties required to define the installation path and connection to the remote host of the DROPS Agent. These properties are only used when the when the DROPS Agent is installed or updated remotely on the infrastructure item.

> ***Reference***
>
> For more information about remote installation refer toManaging an infrastructure item’s remote installation properties on page 124 and the *DROPS Installation Guide*.

Provisioning Template

Select a provisioning template to use if your infrastructure item must be configured before deploying content. The list is pre-populated with the templates defined on the current DROPS Server.

> ***Reference***
>
> For more information about these templates, refer to Environment provisioning templates on page 223 andProvisioning infrastructure items on page 123.

If the **Provisioned** checkbox is ticked, the item has been provisioned.

Instantiation Template

Select an instantiation template to create a new VM for the item. The list is pre-populated with the templates defined on the current DROPS Server.

> ***Reference***
>
> For more information about these templates, refer to Instantiation templates on page 229 andInstantiating infrastructure items on page 123.

If the **Instantiated** checkbox is ticked, the item has been instantiated.

The **Connection** and **Deployment Information** section in the **General Information** tab are crucial to configuring an infrastructure item. They enable you to define the connection information and directories that serve as destination locations on the target system during the deployment process. The values entered here can be referenced in substitution variables during the deployment process.

Protocol

Select the security protocol to use to connect to the infrastructure item. The options available depend on the combination of *OS Family and the *Remote Agent Type:

IBM i	Agent	HTTP(s)
	Agentless	JT400
	Agentless	Secure JT400 via SSL
	Agentless	SSH
Linux	Agent	HTTP(s)
	Agentless	SSH
Windows	Agent	HTTP(s)
	Agentless	PowerShell ≥4

Host name & Port

Enter the host name and port that point to the infrastructure item.

SSH key

If the Protocol selected is **SSH**, for any OS, select the SSH key to use to connect to the infrastructure item. The list is prepopulated with all of the SSH keys available on the current DROPS Server.

> ***Reference***
> For more information about SSH keys, refer to SSH keys on page 129

Login & Password

Enter the valid (domain) user's login and password to connect to the target system.

Backup Directory

The directory path on the target system used to stock a copy of the content that is currently in the same location defined in the deployment process.

The copy of the current content stored here serves as a backup. If the deployment of artifacts does not work and you need to roll back the deployment, the previous version of content from the target location will be reinstated. In order to reinstate it, the content is copied during the deployment process and stored at this location.

Root Directory

The directory path where the deployed artifacts are temporary stored before being officially copied to the target.

The Root Directory is used to temporarily store the artifacts to be deployed before DROPS actually deploys them. The deployable content follows the same structure that is defined in each component. Actions can be taken on the content before it is finally deployed, enabling you to automate steps such as unzipping, testing, transforming a configuration file, etc or to set a requirement for validation.

Define this path manually or enter *AUTO to use the default Local Repository Storage Area path created when the DROPS Agent is installed.

> ***Reference***
> For more information about the Local Repository Storage Area refer to the *DROPS Installation Guide*.

File Separator

Enter the standard file separator for the target system:
\ for windows or / for any other system.

Deployment Schema

To define the list of deployment schemas allowed for an infrastructure item, click the **Add** button. The dialog is prepopulated with the schemas defined for the current server.

Selecting an added schema and clicking the **Edit** button or double-clicking it opens the **Deployment Schema** editor.

To remove an added schema from the infrastructure item, select it and click the **Remove** button.

Reference

For more information about managing schemas, refer to Deployment schemas on page 132.

Save the changes (, Ctrl+S or **File** > **Save**).

22.3 Remote SSH options

The remote SSH options added to the Infrastructure Items view allow you to adjust the agentless execution mode depending on its own context.

Once these options are set with the wanted values, they override the default ones.

Remote home directory

Indicates the path to a home directory on the remote Infrastructure Item.

Remote JAVA path

Indicates the path to a directory where JAVA has been installed on the remote Infrastructure Item.

Remote JAVA version command

Sets the command that needs to be executed to get the java version installed on the remote Infrastructure Item.

Cat command

Sets the CAT Command used to read the content of a file on the remote Infrastructure Item. Note that this command needs to define the substitution variable ${filename} that is then substituted by the name of the file we want to read the content. This command allows the definition of additional CAT options (for z/OS for example).

22.4 Viewing the deployed releases

The read-only **Deployed Releases** tab in the **Infrastructure Item** editor displays the complete list of releases that have already been deployed on the item.

22.5 Viewing the deployment history

The **Deployment History** tab in the **Infrastructure Item** editor displays the complete history of the deployment process instances executed for the current item.

The table displays complete information about the deployment including the final result, the application and release deployed, the date and time the deployment was finished, the package type and the deployment process used to deliver.

22.6 Changing the type of an infrastructure item

To change the Infra. Item Type, either right-click on it in the **Infrastructure Items** search view, then select **Change type**, or select the item then click the change type icon.

The drop-down list to select the new type is prepopulated with the infrastructure item types defined by the administrator in the **Preferences** menu. Click **OK** then close the item editor and refresh the search list. When the infrastructure item is opened again, the change is visible.

> ***Warning!***
>
> If the extended configuration for an infrastructure item has already been defined, changing its type will delete **all** of the information.
>
> To create a different type of item and keep the information defined in the current one, create a new infrastructure item.

> ***Reference***
>
> For more information about entity types, refer to Entity types in the appendix.

22.7 Changing the OS family of an infrastructure item

Changing the OS family of an infrastructure item is useful if you are migrating the DROPS Agent from one OS to another. If you change an agent’s OS by remotely installing it, it is important to change the OS in the infrastructure item and fill in the correct Installation Properties.

To change the *OS Family of an infrastructure item, either right-click on it in the **Infrastructure Items** search view, then select **Change OS family**, or select the item then click the change OS icon.

The OS family cannot be changed in the infrastructure item editor.

22.8 Downloading remote agent logs

Select an item and click the download icon or right-click on an item and select **Download logs** to save a .zip file containing the item’s remote agent’s logs.

22.9 Executing compliance tests

Perform tests on entity to ensure that the target environment complies to specific requirements. Compliance rules are a complementary tool, available independently of the main DROPS process.

> ***Reference***
>
> For more information about compliance rules, refer to Compliance rules on page 140.

22.10 Provisioning infrastructure items

If a provisioning template is associated with an infrastructure item, it will be executed when the item is deployed. You can also provision items outside of the deployment process by executing the process manually.

> **Note**
> It is not possible to execute the provisioning process if the infrastructure item is not instantiated.

To execute the provisioning process manually, either select the parent item and click the Provisioning Process icon or right-click on it and select **Provision**. The log is automatically opened.

The **Provisioning Result** log allows you to filter the log entries using the filter bar.

22.10.1 View provisioning results

The results for every provisioning instance launched for any infrastructure item can be viewed from the **Operational Console Management** perspective.

Step 1 Switch to the Operational Console Management perspective.

Step 2 Connect to and expand the DROPS Server.

Step 3 Double-click on the **Provisioning Execution Instances** node.

Step 4 Either select the instance in the list and click the Show results icon, right-click on it and select **Show provisioning results** or double-click on it to open the results view.

Result The **Overview** tab displays the general information about the process, the **Log** tab displays the execution log entries and the **Attachment** tab displays the attachment list.

22.11 Instantiating infrastructure items

If an instantiation template is associated with an infrastructure item, it will be executed when the item is deployed. You can also instantiate items outside of the deployment process by executing the process manually.

To execute the instantiation process manually, either select the parent item and click the Instantiation Process icon or right-click on it and select **Instantiate**. The following log is automatically opened.

The **Instantiation Result** log allows you to filter the log entries using the filter bar.

22.11.1 View instantiation results

The results for every instantiation instance launched for any infrastructure item can be viewed using the **Operational Console Management** perspective.

Step 1 Switch to the Operational Console Management perspective.

Step 2 Connect to and expand the DROPS Server.

Step 3 Double-click on the **Instantiation Execution Instances** node.

Step 4 Either select the instance in the list and click the Show results icon, right-click on it and select **Show instantiation results** or double-click on it to open the results view.

Result The **Overview** tab displays the general information about the process, the **Log** tab displays the execution log entries and the **Attachment** tab displays the attachment list.

22.12 Managing an infrastructure item’s remote installation properties

An infrastructure item’s Installation Properties are specific to the OS on which the corresponding DROPS Agent is installed.

To access the item’s installation properties click the **Edit remote installation properties...** link in the item's editor. The **Linux**, **Windows** or **IBM i Installation** editor is displayed depending on the OS selected for the item.

⚠ **Important!**
If the agent was initially installed remotely, some of the following values are already defined. If the agent was initially installed manually it is required to create them on the DROPS database and then to define the following values manually before you can update the DROPS Agent remotely. If the infrastructure item was created in an early version of DROPS, it may be required to activate these parameters first by generating them.

Reference
For more information about the remote installation process, refer to the *DROPSInstallation Guide*.

Field	Description
Installation Path	The directory where the agent is installed.
Local Repository Storage Area	The path on the target DROPS Agent that contains both the Root Directory and the Backup Directory. This path must be accessible to the DROPS Agent and is intended to regroup all of the content deployed to an agent in one location. This path is used when the Backup and/or the Root Directory is set to ***AUTO** in an infrastructure item.
User & Password	The user name and password used, or to use, during the remote installation process.

Table 7: DROPS Agent Installation properties

The user required to *install* the agent is not necessarily the same as the user that *runs the agent* after it is installed. To access the agent after it is installed to update it, the user who runs it and the corresponding connection information are also required.

Windows - Define the **Agent User** and **Password** required to access the profile that will run the agent's service.

Linux - Define the **Agent User** required to access the profile that will run the agent's daemon.

IBM i - Define the **Target iASP** and its child **Installation Library** required to hold the agent's objects. Define the **Agent User** required to access the profile that will run the agent's job.

22.12.1 Generating an infrastructure item's remote installation properties

If the Installation Properties link is not displayed for an item created in the DROPS Studio v2.0 or earlier, the properties must first be generated.

> **Note**
> If an infrastructure item was created in the DROPS Studio v2.2 or later, you do not need to generate installation properties. They are already available by default.

To generate the installation properties for an "older" item, either right-click on it in the **Infrastructure Items** search view, then select **Generate installation properties**, or select the item then click the generate installation properties icon.

After they are generated, the link to the **Remote Installation Properties** is displayed in the item's editor.

22.12.2 Creating remote installation information in the DROPS database

> **Note**
> If an agent was initially installed manually the installation details do not exist in the DROPS database. To update an agent remotely, the installation information is required in the database so that DROPS knows the agent has already been installed. If the install information isn't the in database, when DROPS tried to update the agent remotely, it will launch a fresh installation instead of an update.
>
> If the agent was initially installed remotely, it is not required to create the installation information in the database because it was created and stored automatically during the remote installation process.

To create (remote) installation information in the DROPS database, either right-click on the item in the **Infrastructure Items** search view, then select **Reference installation**, or select the item then click the reference installation icon. The **Module Release** dialog is displayed and enables you to select the DROPS Agent and set up an arbitrary remote install.

22.13 Testing the connection to an infrastructure item (agent)

It is recommended to test the connection information defined in an infrastructure item to ensure that DROPS can successfully connect to it before executing the deployment process.

If required, restart the agent before testing it by selecting the item then clicking the restart remote agent button, or right-clicking on it and selecting **Restart agent**.

There are two ways to test the connection DROPS maintains with the agent installed on infrastructure items:

1. Manually test the connection one time from the **Infrastructure Items** search view.
2. Add the item to a list that is regularly checked by the DROPS Server to monitor that the connections continue to be reachable.

> **Note**
> If an infrastructure item is not correctly connected, when it is used in a deployment process, the process will be suspended.

> ***Reference***
> For more information about suspended items in process deployment, refer to Suspended actions on page 294.

22.13.1 Test the connection to an infrastructure item (agent) manually

To test the connection to an agent manually, either right-click on it in the **Infrastructure Items** search view, then select **Test connection**, or select the item then click the test connection icon.

If the connection is broken the **Problem Occurred** dialog is displayed confirming that the connection to the agent could not be successfully established. Click the **Details** button for more information about why the connection failed.

If the connection is stable, the dialog confirms that the connection to the agent was successfully established.

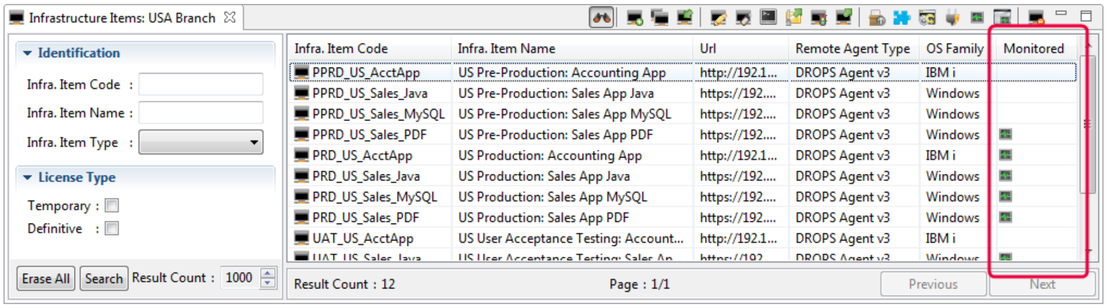
22.13.2 Monitor the connection to an infrastructure item (agent) automatically

To test the connection to one or more agent(s) automatically and periodically, add infrastructure items to the monitoring list. A task is scheduled on the DROPS Server to check if the items that belong to the monitoring list are reachable. If the agent is not reachable, the status of the entry is set to FAILED and a notification can be sent to the user(s) in the Administrator or Contributor role.

The schedule that the monitor list follows to verify the connection(s) to the DROPS Agent(s) is defined in the configuration view by an administrator.

> ***Reference***
> For more information about defining the monitoring schedule, refer to the *DROPS Configuration Guide*.

If an infrastructure item is added to the monitoring list, the monitor icon is displayed for that item in the **Infrastructure Items** search view.



Infrastructure Items: USA Branch

Identification
Infra. Item Code :
Infra. Item Name :
Infra. Item Type :

License Type
Temporary : □
Definitive : □

Erase All | Search | Result Count : 1000

Infra. Item Code	Infra. Item Name	Url	Remote Agent Type	OS Family	Monitored
PPRD_US_AcctApp	US Pre-Production: Accounting App	http://192.1...	DROPS Agent v3	IBM i	
PPRD_US_Sales_Java	US Pre-Production: Sales App Java	https://192....	DROPS Agent v3	Windows	
PPRD_US_Sales_MySQL	US Pre-Production: Sales App MySQL	https://192....	DROPS Agent v3	Windows	
PPRD_US_Sales_PDF	US Pre-Production: Sales App PDF	https://192....	DROPS Agent v3	Windows	🟩
PRD_US_AcctApp	US Production: Accounting App	http://192.1...	DROPS Agent v3	IBM i	🟩
PRD_US_Sales_Java	US Production: Sales App Java	https://192....	DROPS Agent v3	Windows	🟩
PRD_US_Sales_MySQL	US Production: Sales App MySQL	https://192....	DROPS Agent v3	Windows	🟩
PRD_US_Sales_PDF	US Production: Sales App PDF	https://192....	DROPS Agent v3	Windows	🟩
UAT_US_AcctApp	US User Acceptance Testing: Account...	http://192.1...	DROPS Agent v3	IBM i	
UAT_US_Sales_Java	US User Acceptance Testing: Sales Ap	https://192	DROPS Agent v3	Windows	🟩

Result Count : 12 | Page : 1/1 | Previous | Next

Figure 35: Infrastructure agents included in the Monitoring List

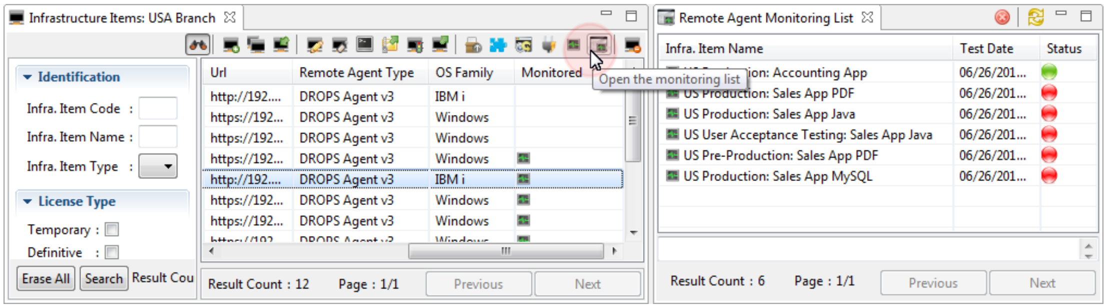
To add one or multiple items to the monitoring list, select it from the **Infrastructure Items** search view then either right-click and select **Add to the monitoring list** or click the monitor icon in the local toolbar.

To remove one or multiple items from the monitoring list, select it from the **Infrastructure Items** search view then either right-click and select **Remove from the monitoring list** or click the monitor icon in the local toolbar.

Selecting multiple items that may or may not be in the list already is possible. All the selected items that are monitored will be removed from the list and all the selected items that are not monitored will be added to the list.

22.13.2.1 Accessing the Remote Agent Monitoring List

To access the **Remote Agent Monitoring List** view, click the monitoring list icon in the **Infrastructure Items** search view toolbar.



Infrastructure Items: USA Branch

Identification
Infra. Item Code :
Infra. Item Name :
Infra. Item Type :

License Type
Temporary : □
Definitive : □

Erase All | Search | Result Cou

Url	Remote Agent Type	OS Family	Monitored
http://192....	DROPS Agent v3	IBM i	
https://192...	DROPS Agent v3	Windows	
https://192...	DROPS Agent v3	Windows	
https://192...	DROPS Agent v3	Windows	🟩
http://192....	DROPS Agent v3	IBM i	🟩
https://192...	DROPS Agent v3	Windows	🟩
https://192...	DROPS Agent v3	Windows	🟩
https://192	DROPS Agent v3	Windows	🟩

Result Count : 12 | Page : 1/1 | Previous | Next

Remote Agent Monitoring List

Infra. Item Name	Test Date	Status
ion: Accounting App	06/26/201...	🟢
US Production: Sales App PDF	06/26/201...	🔴
US Production: Sales App Java	06/26/201...	🔴
US User Acceptance Testing: Sales App Java	06/26/201...	🔴
US Pre-Production: Sales App PDF	06/26/201...	🔴
US Production: Sales App MySQL	06/26/201...	🔴

Result Count : 6 | Page : 1/1 | Previous | Next

Figure 36: The Remote Agent Monitoring List view

The **Remote Agent Monitoring List** displays a list of all the infrastructure items in the list. Nested under each item is the date of the last check and the status of the entry at the last check. Selecting an entry displays the message related to the connection attempt in the lower part of the view.

To refresh the content of this view, click the Refresh icon.

To remove an item from the list select it and either right-click and select **Remove Entry** or click the remove icon.

22.14 Deleting infrastructure items

> ***Warning!***
>
> Deleted infrastructure items cannot be accessed or recovered.
>
> Any deployment process that references the deleted infrastructure item will not be updated and will fail if deployed. Existing deployment processes are never updated automatically with changes to infrastructure items.

To delete an infrastructure item search for it in the **Infrastructure Items** search view. Either right-click on it and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the infrastructure item.

23 SSH keys

Required Roles	n/a
Required Rights (by category)	SSH Key Management
Perspective	Configuration Management
Access	DROPS Server → Configuration → SSH Keys

Chapter Summary

23.1 Create new SSH keys ... 130
23.2 Import existing SSH keys ... 131
23.3 Editing SSH keys ... 131
23.4 Accessing the public key ... 131
23.5 Deleting SSH keys ... 131

DROPS can generate SSH key pairs to facilitate secured connections to external target tools. It is not required to use an SSH key generated by DROPS to connect to external tools but it is an option among others.

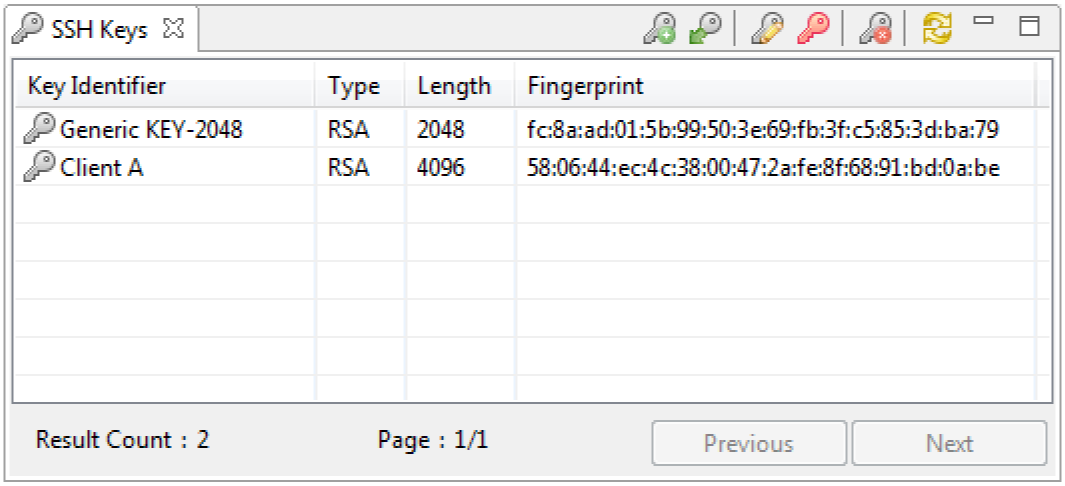
An SSH key pair is comprised of:

- the private key, referenced in DROPS.
- the public key, which must be referenced by the system to which DROPS needs to connect.

The SSH keys generated by DROPS are either EdDSA or RSA keys.

If you are using the agentless mode to connect an infrastructure item to a target (in order to deploy to that target), the SSH keys defined in this section can be used to secure the connection. SSH keys can be generated in the DROPS Studio or imported if you have already generated the key(s) you want to use elsewhere.

> ***Reference***
> For more information about how SSH keys are used in DROPS, refer to Infrastructure items on page 108



Key Identifier	Type	Length	Fingerprint
Generic KEY-2048	RSA	2048	fc:8a:ad:01:5b:99:50:3e:69:fb:3f:c5:85:3d:ba:79
Client A	RSA	4096	58:06:44:ec:4c:38:00:47:2a:fe:8f:68:91:bd:0a:be

Result Count : 2 Page : 1/1

Figure 37: The SSH Keys view

To access the **SSH Keys** view, select the keys node in the **Configuration** menu. The **SSH Keys** view displays all of the keys defined for the current server. Each key's fingerprint is displayed in the Fingerprint column.

23.1 Create new SSH keys

Follow the subsequent steps to create a new SSH key pair.

Step 1 To access the **SSH Key** wizard, either click the create icon in the **SSH Keys** search view or right-click anywhere in the view and select **Create new SSH key**.

Step 2 Enter a recognizable ID in the **Key Identifier** field.

Step 3 Select the **Type** of key to generate. You can choose between an EdDSA key and a RSA key.

Step 4 Enter the passphrase for the SSH key pair.

> **Note**
> The passphrase will not be displayed in order to ensure absolute confidentiality. Be sure to enter the correct passphrase and not to forget it.

Step 5 Click **Finish** to create the new key. The new key appears in the list.

Result SSH keys are saved in the DROPS Server installation directory in the following path: */../ssh/keystore/ks<n>* (where n is the key's internal ID).

There are two files, each representing half of the key pair:

- the private part of the key which DROPS will reference.
- the public part of the key which must be referenced by the target tool.

23.2 Import existing SSH keys

Follow the subsequent steps to import an existing SSH key. Only the private half of the SSH is imported into DROPS.

Step 1 To import an existing private key, either right-click anywhere in the view and select **Import SSH key** or click the import icon.

Step 2 Enter a recognizable ID in the **Key Identifier** field.

Step 3 Browse to the location where the existing private key file is stored and select it in the **Private key** field.

Step 4 Enter the passphrase for the key if one was defined.

Step 5 Click **Finish**.

Result The private key is saved in the DROPS Server installation directory in the following path: */../ssh/keystore/ks<n>* (where n is the key's internal ID).

23.3 Editing SSH keys

To view or edit a key's ID, either right-click on it and select **Edit**, select the key then click the Edit icon or double-click on it. The **SSH Key** editor is opened where its ID can be edited. You can also add comments in the SSH key editor but you cannot edit the key pair.

23.4 Accessing the public key

To display the public key, either right-click on the SSH key and select **Show Public Key** or select the key in the list and click the public key icon. You can copy the public key from the dialog and use it to create the SSH connection in the target tool.

23.5 Deleting SSH keys

> ***Warning!***
> Deleted SSH Keys cannot be accessed or recovered and will no longer be useable. If a referenced SSH key is deleted, the entity referencing it will no long be able to connect to its target.

To delete an SSH key either right-click on it and select **Delete** or select it and click the Delete icon. Click **OK** to confirm or click **Cancel** to keep the key.

24 Deployment schemas

Required Roles	n/a
Required Rights (by category)	Infrastructure Items Deployment Schemas
Perspective	Configuration Management
Access	DROPS Server → Configuration → Deployment Schemas

Chapter Summary

24.1 Accessing deployment schemas 133
24.2 Creating deployment schemas 133
24.3 Editing deployment schemas 134
24.4 Managing deployment rules 134
24.5 Deleting deployment schemas 135

Deployment schemas are sets of deployment rules used to change the destination location for artifacts when they are deployed to an infrastructure item. During regular deployment, the transferred artifacts are copied to the **Root Directory** destination defined in the infrastructure item following the same directory structure defined in the parent component. However, you may need to change this location for a certain group of artifacts.

Deployment rules are used to assign a replacement target path for a set of artifacts defined by a RegEx filter. During the transfer stage of the deployment process, if a deployment schema that has been added to an infrastructure item is selected, all of the artifacts in the component deployed are run through the rules' RegExes. Each artifact that matches a RegEx will be redirected to the corresponding replacement path. Regular expressions can only be applied at the file name-level but they take into account the artifact key. The key is the relative path to the artifact starting from its parent component.

> ***Example***
>
> A Deployment rule defines a RegEx to find .SQL artifacts in the component deployed to an infrastructure item. No matter what the paths defined in the components are, any artifact that matches the .SQL RegEx is redistributed to the **Target Path** defined in the deployment rule. This enables you to regroup artifacts in your target service that come from multiple strategies which may each have a different source directory structure.

The ${arcad.sourcepath} substitution variable can be used in the **Target Path** field. It is replaced by the artifact Key (relative path in the artifact repository).

DROPS Component directory structure	<ArtifactRepository>scripts/myScript.sql

Table 8: Deployment Rule example

No Deployment Rule: deployment path follows component directory structure	<RootDirectory>/scripts/myScript.sql
Deployment Rule: *.sql → db/SQLScripts	<RootDirectory>/db/SQLScripts/myScript.sql
Deployment Rule: *.sql → db/SQLScripts/${arcad.sourcepath}	<RootDirectory>/ db/SQLScripts/scripts/myScript.sql

Table 8: Deployment Rule example

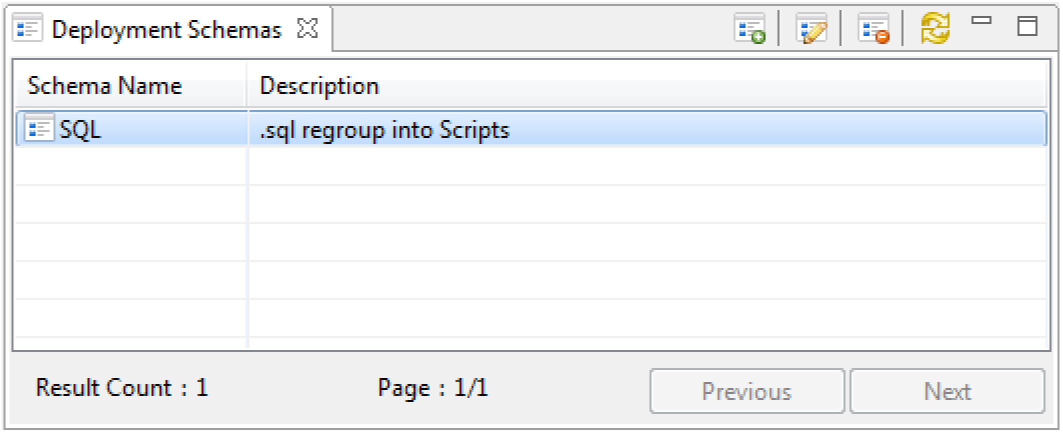
The deployment rules are sorted by a specific order defined by the user. The order of the rules is important because if an artifact matches several Rules, the target file name will be generated by the *first* rule it matches.

⚠ **Important!**
If an infrastructure item contains a deployment schema it will always be used. It is taken into account during the transfer stage of the deployment process.

24.1 Accessing deployment schemas

To access the **Deployment Schemas** view, select the deployment schemas node in the **Configuration Explorer**.

The **Deployment Schemas** view displays all of the schemas defined for the current server.



Schema Name	Description
SQL	.sql regroup into Scripts

Figure 38: The Deployment Schemas view

24.2 Creating deployment schemas

Follow the subsequent steps to create a new deployment schema.

Step 1 To access the **Create a new deployment schema** wizard, either click the create icon in the **Deployment Schemas** view or right-click anywhere in the view and select **Create a new deployment schema**.

Step 2 Enter the mandatory **Name** of the schema.

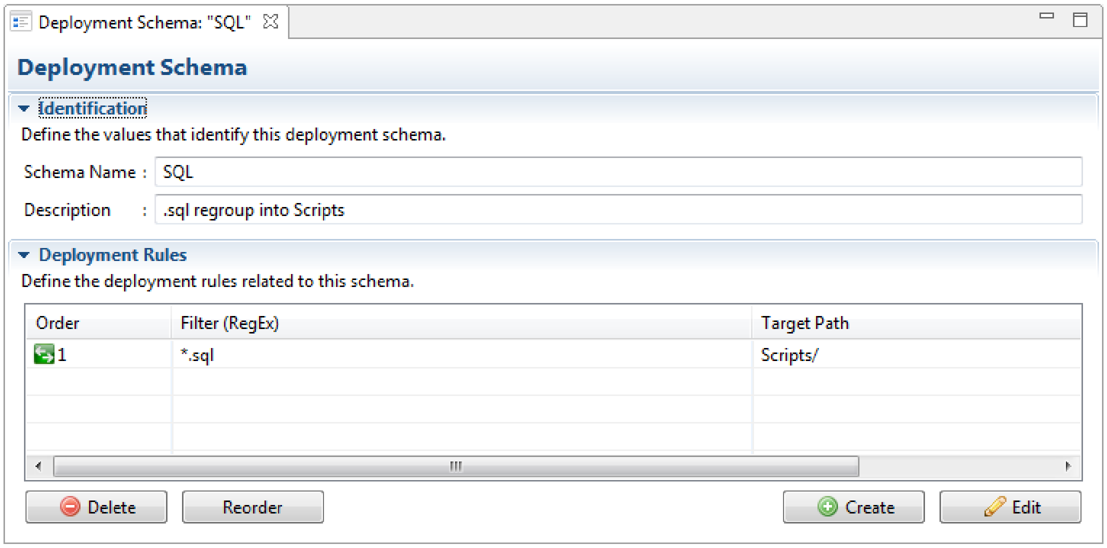
Step 3 Click **Finish**.

Result New deployment schemas are available in the **Deployment Schemas** view.

24.3 Editing deployment schemas

To edit deployment schema details, either right-click on it in the **Deployment Schemas** view then select **Edit**, select the item then click the edit icon or double-click it. The **Deployment Schema** editor is opened where all of the entity's details are managed.

The information that defines schemas can be edited at any time by users with the rights to do so.


Deployment Schema: "SQL"

Deployment Schema

Identification

Define the values that identify this deployment schema.

Schema Name : SQL

Description : .sql regroup into Scripts

Deployment Rules

Define the deployment rules related to this schema.

Order	Filter (RegEx)	Target Path
1	*.sql	Scripts/

Figure 39: The Deployment Schema editor

Enter a short **Description** of the schema to help you describe the RegExes defined in the schema's deployment rules.

The **Deployment Rules** manager displays all of the existing rules defined for the current deployment schema. The **Order** column defines the order that the rules will be run in if there is more than one Rule. If the rules' order numbers are not chronological, click the **Reorder** button and DROPS will automatically renumber each rule following the order in which they are displayed.

24.4 Managing deployment rules

To create a new rule, click the **Create** button in the **Deployment Rules** manager. In the **Deployment Rule** dialog, define:

1. the priority in the **Order** field. This number enables you to initially select the position this rule will have in the order rules are run. Change this number to a higher number than the last rule in the list to move it to the end of the list.
2. the regular expression in the **Rule** field.
3. the relative path on the target infrastructure item in the **Target Path** field.

Double-click an existing rule or select it and click the **Edit** button to change its settings.

To delete a deployment rule, select it then click **Delete**. Click **OK** to confirm or click **Cancel** to keep the rule.

Save the changes (, Ctrl+S or **File** > **Save**).

24.5 Deleting deployment schemas

> ***Warning!***
>
> Deleted deployment schemas cannot be accessed or recovered.
>
> Any deployment rules defined in a deleted schema are also deleted.

To delete a deployment schema, either right-click on it in the **Deployment Schemas** view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the schema.

25 Replacement rules

Required Roles	n/a
Required Rights (by category)	Replacement Rules
Perspective	Configuration Management
Access	DROPS Server → Configuration → Replacement Rules

Chapter Summary

25.1 Accessing replacement rules 137
25.2 Creating replacement rules 138
25.3 Editing replacement rules 138
25.4 Deleting replacement rules 139

Replacement rules are similar to a *Search & Replace* on configuration files linked to an application. Rules enable you to activate specific commented content in artifacts such as .properties files during deployment. This means that commented content can be selected and rendered active by automatically removing the comment tag. The now-active content will be taken into account during the deployment process.

Replacement rules are run at the beginning of the transfer stage of the deployment process if the Use Replacement Rules checkbox is ticked in the **Transfer** tab of a deployment process' **Properties** view.

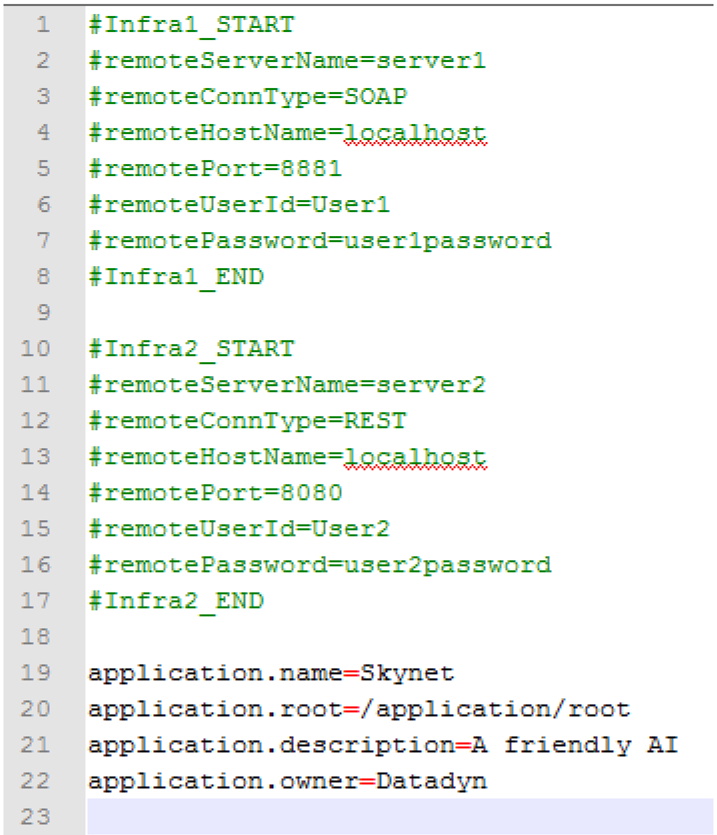
Files to use in relation to replacement rules *must* be prepared in advance. Replacement rules are configured to read and convert specific comment tags which are unique to the file type. Multiple rules can be created but a single rule can only refer to one file type.

Substitution variables are often used in commented content in order to substitute the value of each variable with the correct context-specific content related to the different infrastructure items you deploy to.

Example

You have multiple infrastructure items that are all similar servers. They are so similar that they use the same .properties file, except there are some lines in that common properties file that are specific to each individual server. Use a replacement rule to call out only the server-specific information for each infrastructure item by including all of the information for every server in the .properties file and turning them all into comments. Create a rule for each infrastructure item that calls only the instruction specific to the server it points to and ignores the commented content that refers to the other servers.

A rule for Infra1 would pull the information between #Infra1_START and #Infra1_END into action in order to connect to the Infra1 server but it would not recognize the comments made for Infra2 because the rule only refers to the content between the two delimiters defined.



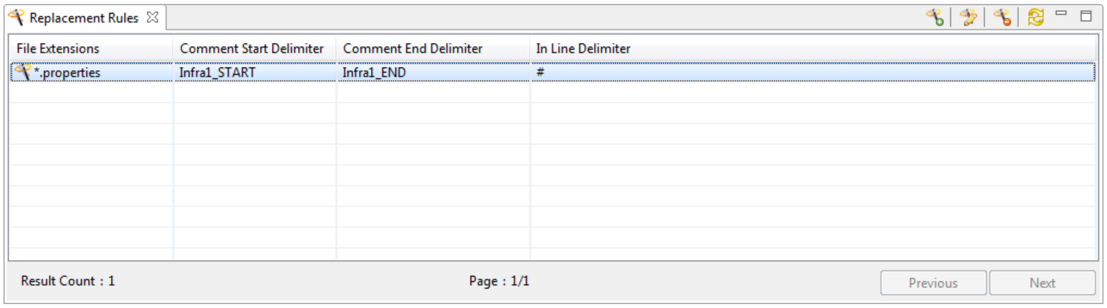
```
#Infra1_START
#remoteServerName=server1
#remoteConnType=SOAP
#remoteHostName=localhost
#remotePort=8881
#remoteUserId=User1
#remotePassword=user1password
#Infra1_END

#Infra2_START
#remoteServerName=server2
#remoteConnType=REST
#remoteHostName=localhost
#remotePort=8080
#remoteUserId=User2
#remotePassword=user2password
#Infra2_END

application.name=Skynet
application.root=/application/root
application.description=A friendly AI
application.owner=Datadyn

```

Figure 40: Example of a .properties file prepared for replacement rules

25.1 Accessing replacement rules

To access the **Replacement Rules** view, select the replacement rules node in the **Configuration Explorer**.

The **Replacement Rules** view displays all of the rules defined for the current server.



File Extensions	Comment Start Delimiter	Comment End Delimiter	In Line Delimiter
*.properties	Infra1_START	Infra1_END	#

Result Count : 1

Page : 1/1

Figure 41: The Replacement Rules view

25.2 Creating replacement rules

> **Note**
>
> Only file types that use an in-line delimiter on every line of a comment can be referenced in replacement rules, such as properties files which use the pound symbol (#) at the beginning of *every line* of commented content.
>
> Do NOT use .xml files. Comments are encased in beginning and ending <!-- ... --> comment tags that are not recognized by replacement rules.

Follow the subsequent steps to create a new replacement rule.

Step 1 To access the **Create a new replacement rule** wizard, either click the create icon in the **Replacement Rules** view or right-click anywhere in the view and select **Create a new replacement rule**.

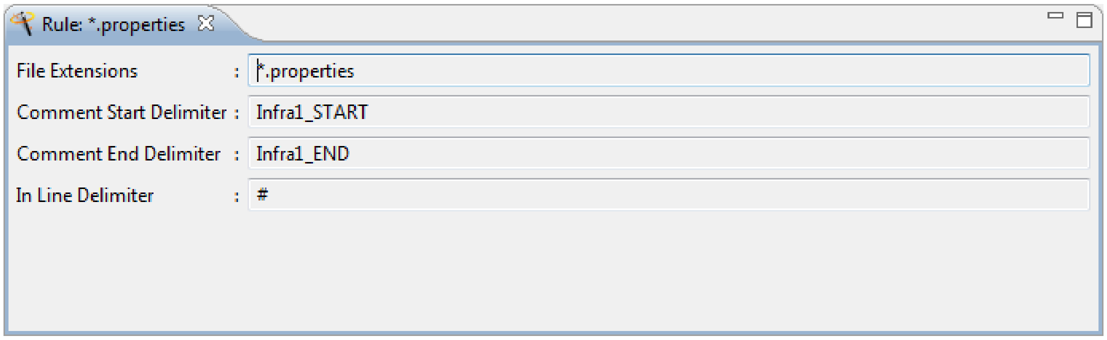
Step 2 Complete all of the fields. These values can be edited later.

Step 3 Click **Finish**.

Result New replacement rules are available in the **Replacement Rules** view.

25.3 Editing replacement rules

To edit replacement rule details, either right-click on it in the **Replacement Rules** view then select **Edit**, select the item then click the edit icon or double-click it. The **Rule** editor is opened where all of the entity's details are managed.



File Extensions	:	*.properties
Comment Start Delimiter	:	Infra1_START
Comment End Delimiter	:	Infra1_END
In Line Delimiter	:	#

Figure 42: The Replacement Rules editor

File Extensions

The file extension to which the rule should be applied. Use a RegEx here to apply the replacement rule to multiple files of the same type.

Comment Start Delimiter

The first line of the commented content to replace.

Comment End Delimiter

The last line of the commented content to replace.

In Line Delimiter

The file type-specific comment symbol (tag) that the rule should recognize. The comment delimiter is different for many file types:

- **#** - .properties files use the pound/hashtag symbol on every line of the comment.
- **rem** - Windows batch files use the string 'rem'.
- **;** - .inf files use the semi-colon on every line of the comment.

Only the symbol or string that represents the commented content for the file type defined in the Field Extensions field should be entered here.

25.4 Deleting replacement rules

> ***Warning!***
> Deleted replacement rules cannot be accessed or recovered.

To delete a replacement rule, either right-click on it in the **Replacement Rules** view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the rule.

26 Compliance rules

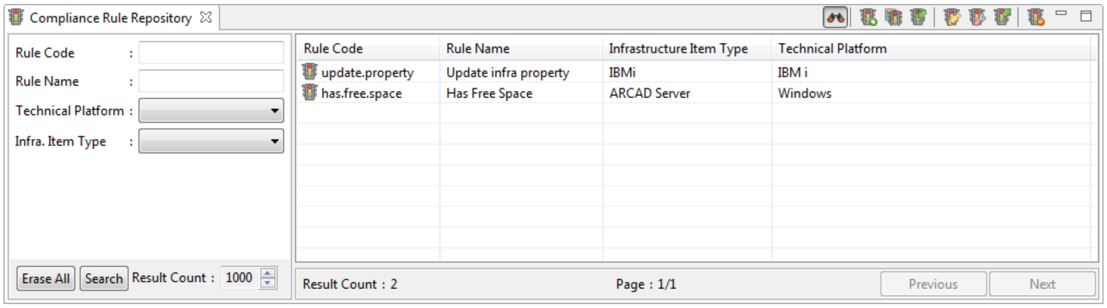
Required Roles	n/a
Required Rights (by category)	Compliance Rules
Perspective	Configuration Management
Access	DROPS Server → Compliance

Chapter Summary

26.1 Creating compliance rules 140
26.2 Editing a compliance rule 141
26.3 Importing and exporting compliance rules 143
26.4 Executing compliance rules 144
26.5 Understanding compliance rule logs and compliance reports 146
26.6 Deleting compliance rules and rule sets 149

Execute **Compliance rules** to check that target environments comply to given requirements. The requirements are specific to your needs and can change from one rule to the next. Users with the rights to do so can create or import compliance rules according to the properties to be tested and group compliance rules into rule sets. Rule sets are an ensemble of related compliance rules aimed to be used together for an environment.

Execute the set of rules on an environment, or one or more infrastructure items before carrying out the deployment process. See the results of the tests in the compliance reports and make the necessary changes to the properties of the target environment if needed.



Compliance Rule Repository

Rule Code :
Rule Name :
Technical Platform :
Infra. Item Type :

Erase All | Search | Result Count : 1000

Rule Code	Rule Name	Infrastructure Item Type	Technical Platform
update.property	Update infra property	IBMi	IBM i
has.free.space	Has Free Space	ARCAD Server	Windows

Result Count : 2 Page : 1/1 Previous Next

Figure 43: The Compliance Rule Repository search view

The **Compliance Rule Repository** search view is accessed from the **Compliance** node in the **Configuration Explorer**.

26.1 Creating compliance rules

There are two ways to create compliance rules:

- Create a new compliance rule using the **Create a new compliance rule** wizard.
- Duplicate an existing compliance rule to quickly create a copy of the rule that can be modified.

Follow the subsequent steps to create a new compliance rule.

Step 1 Click the create icon in the **Compliance Rule Repository** search view or right-click anywhere in the view and select **Create a new compliance rule** to open the **Create a new compliance rule** wizard.

Step 2 Define the Rule Code and Rule Name. The Technical Platform and Infrastructure Item Type are not mandatory but highly recommended. These values can be edited later.

Click **Next >** to continue.

Step 3 Selected the Ant project file from an external source. The Ant project file constitutes the content of the compliance rule and can be edited later.

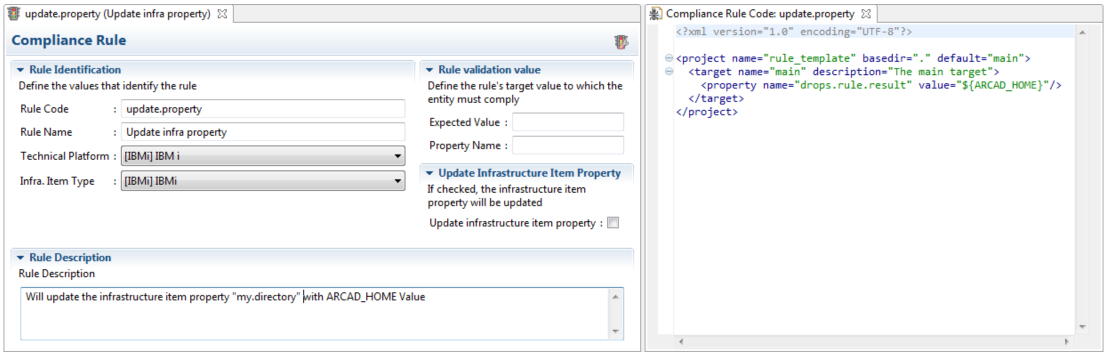
Click **Finish**.

Result The new rules are displayed in the **Compliance Rule Repository** search view.

You can **Duplicate a compliance rule** to quickly create a copy of the rule that can be modified. Select a rule in the list and click the duplicate icon.

26.2 Editing a compliance rule

Compliance rules have two editors. The main editor is where the properties and definitions of the rule can be added or modified. The other editor view is used to write or modify the rule's script content.



update.property (Update infra property)
Compliance Rule
Rule Identification
Rule validation value
Update Infrastructure Item Property
Rule Description
Compliance Rule Code: update.property

Define the values that identify the rule

Rule Code : update.property

Rule Name : Update infra property

Technical Platform : [IBMi] IBM i

Infra. Item Type : [IBMi] IBMi

Define the rule's target value to which the entity must comply

Expected Value :

Property Name :

If checked, the infrastructure item property will be updated

Update infrastructure item property : □

Rule Description

Will update the infrastructure item property "my.directory" with ARCAD_HOME Value

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="rule_template" basedir="." default="main">
  <target name="main" description="The main target">
    <property name="drops.rule.result" value="${ARCAD_HOME}"/>
  </target>
</project>
```

Figure 44: The Compliance Rule editor

26.2.1 Editing a compliance rule's properties

Follow the subsequent steps to edit a compliance rule's properties.

Step 1 Select a compliance rule and click the edit icon in the **Compliance Rule Repository** search view.

Step 2 Edit the compliance rule's identification properties if needed.

Rule Code

A unique string that identifies the rule. This label is used throughout DROPS to select the entity.

Rule Name

A label used to give a descriptive name the rule. This label is used throughout DROPS to select the entity.

Technical Platform

[*Optional*] Select the technical platform type if the rule should or can only be used on a certain OS.

Technical platforms refer to the service to which an infrastructure item points. They are defined by an administrator in the **Preferences** menu.

> ***Reference***
>
> For more information about technical platforms, refer to the *DROPS Configuration Guide*.

Infrastructure Item Type

[*Optional*] Select the type of infrastructure item targeted by this rule. Infrastructure item types are defined by the administrator in the **Preferences** menu.

> ***Reference***
>
> For more information about types, refer to Entity types in the appendix.

Rule Description

[*Optional*] Enter a descriptive text for the compliance rule.

Step 3 Define the compliance rule's functional properties.

Rule Validation

Expected Value: Define a value that is expected when the rule is executed.

Property Name: Define a target property that is expected when the rule is executed. If the element tested complies to the rule’s requirements, it is marked as **succeeded**, otherwise it is marked as **failed**.

Update infrastructure item property

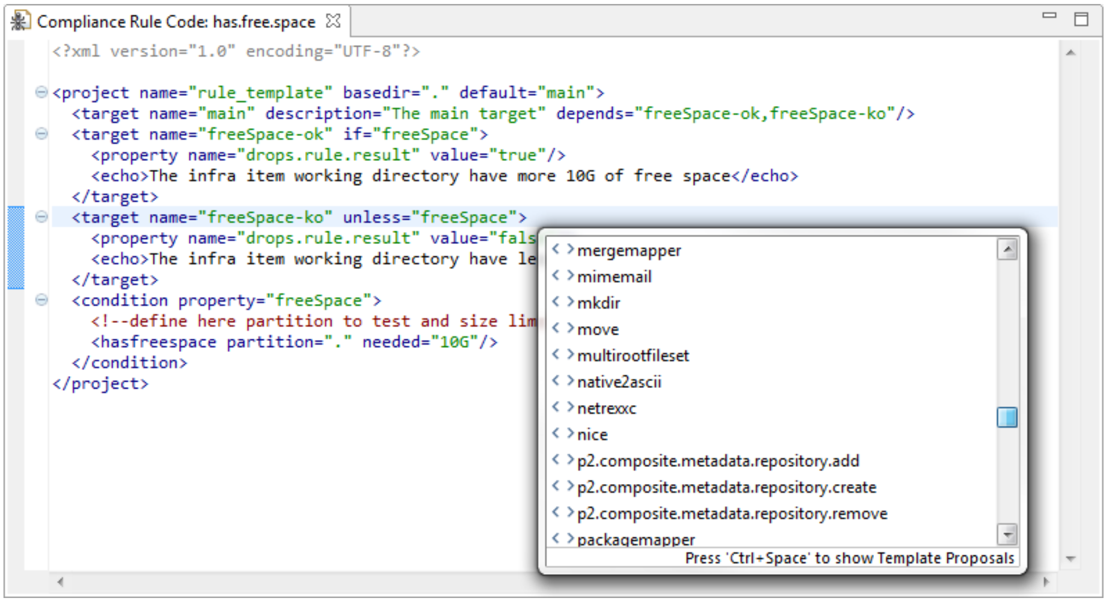
[*Optional*] If active, this option will update the infrastructure item's properties with those defined in the compliance rule.

> ⚠ **Important!**
>
> This option will act on the infrastructure item's properties and must be used with care.

Save the changes (, Ctrl+S or **File** > **Save**).

26.2.2 Editing a compliance rule's content

Editing the content of a compliance rule can be done directly in the DROPS Studio. DROPS uses Eclipse's **Ant Editor** which offers shortcuts (Ctrl+space) to provide easy access to all of the Ant project templates, tasks, parameters and variables available.


Compliance Rule Code: has.free.space
mergemapper
mimemail
mkdir
move
multirootfileset
native2ascii
netrexxc
nice
p2.composite.metadata.repository.add
p2.composite.metadata.repository.create
p2.composite.metadata.repository.remove
packagemapper
Press 'Ctrl+Space' to show Template Proposals

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="rule_template" basedir="." default="main">
  <target name="main" description="The main target" depends="freeSpace-ok,freeSpace-ko"/>
  <target name="freeSpace-ok" if="freeSpace">
    <property name="drops.rule.result" value="true"/>
    <echo>The infra item working directory have more 10G of free space</echo>
  </target>
  <target name="freeSpace-ko" unless="freeSpace">
    <property name="drops.rule.result" value="fals
    <echo>The infra item working directory have le
  </target>
  <condition property="freeSpace">
    <!--define here partition to test and size lim
    <hasfreespace partition="." needed="10G"/>
  </condition>
</project>
```

Figure 45: The Compliance Rule's Ant editor

Select a rule and click the edit content icon or right-click on a rule in the **Compliance Rules Repository** search view and select **Edit Content**. The rule content editor can also be reached by clicking the edit content icon in the **Compliance Rule** editor.

The **Ant Editor** is opened where you can write or edit the content of any rule.

> **Note**
> If a new compliance rule is created without content (no Ant project uploaded), a template is automatically added to the content editor.

Save the changes (, Ctrl+S or **File** > **Save**).

26.3 Importing and exporting compliance rules

Exporting and importing compliance rules is useful when sharing rules between multiple DROPS servers or users.

Compliance rules are exported as (*.xml) files that contain all the rule’s properties and content.

Compliance rules that were previously exported from DROPS can be imported into DROPS again and all of the details that come from the original exported rule are retrieved. DROPS interprets the rule's properties and content, as well as any other, additional information, and populates the corresponding fields in the editor automatically.

> **Note**
> Later modifications to the rule's properties or content applied to the imported rule will override the properties retrieved from the import.

26.3.1 Exporting a compliance rule

Follow the subsequent steps to export compliance rules.

Step 1 Select the rule in the **Compliance Rules Repository** search view then click the export icon, or right-click on the rule then select **Export rule**.

Step 2 Select the location in which to save the rule and change the **File name** if needed. By default, rules are exported with the File name *<RuleName>.xml*.

> ⚠ **Important!**
> Do not change the extension type (*.xml*) that is automatically included in the rule's file name.

Click **Save**.

Result The rules are available externally and can be imported as such for other DROPS servers or users.

26.3.2 Importing a compliance rule

Follow the subsequent steps to import a compliance rule into DROPS.

Step 1 Click the import icon in the **Compliance Rule Repository** search view, or right-click anywhere in the view and select **Import rule**.

Step 2 Click the browse button and select the rule *.xml* file. Click **Finish**.

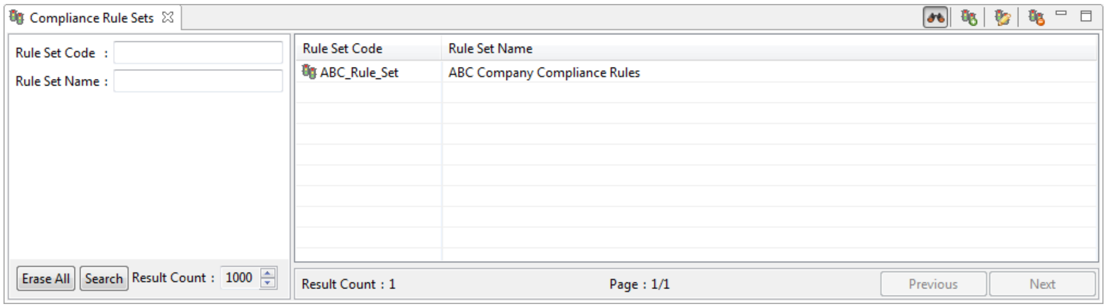
Result Imported rules are displayed in the **Compliance Rule Repository** search view. Refresh the list view if necessary by clicking the **Search** button.

26.4 Executing compliance rules

Even if only one compliance rule will be executed on an environment or an infrastructure item, the rule must be placed into a **rule set** to launch the execution. Individual rules cannot be executed outside sets. Rule sets are an ensemble of related compliance rules that aimed to be used together in an environment.

Execute the set of rules on an environment's infrastructure items or on an individual infrastructure item.

26.4.1 Grouping compliance rules into rule sets



Rule Set Code	Rule Set Name
ABC_Rule_Set	ABC Company Compliance Rules

Result Count : 1 Page : 1/1

Figure 46: The Compliance Rule Sets search view

Follow the subsequent steps to group rules into a **compliance rule set**.

Step 1 Create a **compliance rule set**. Click the create icon in the **Compliance Rule Sets** search view or right-click anywhere in the view and select **Create a new compliance rule set** to open the wizard.

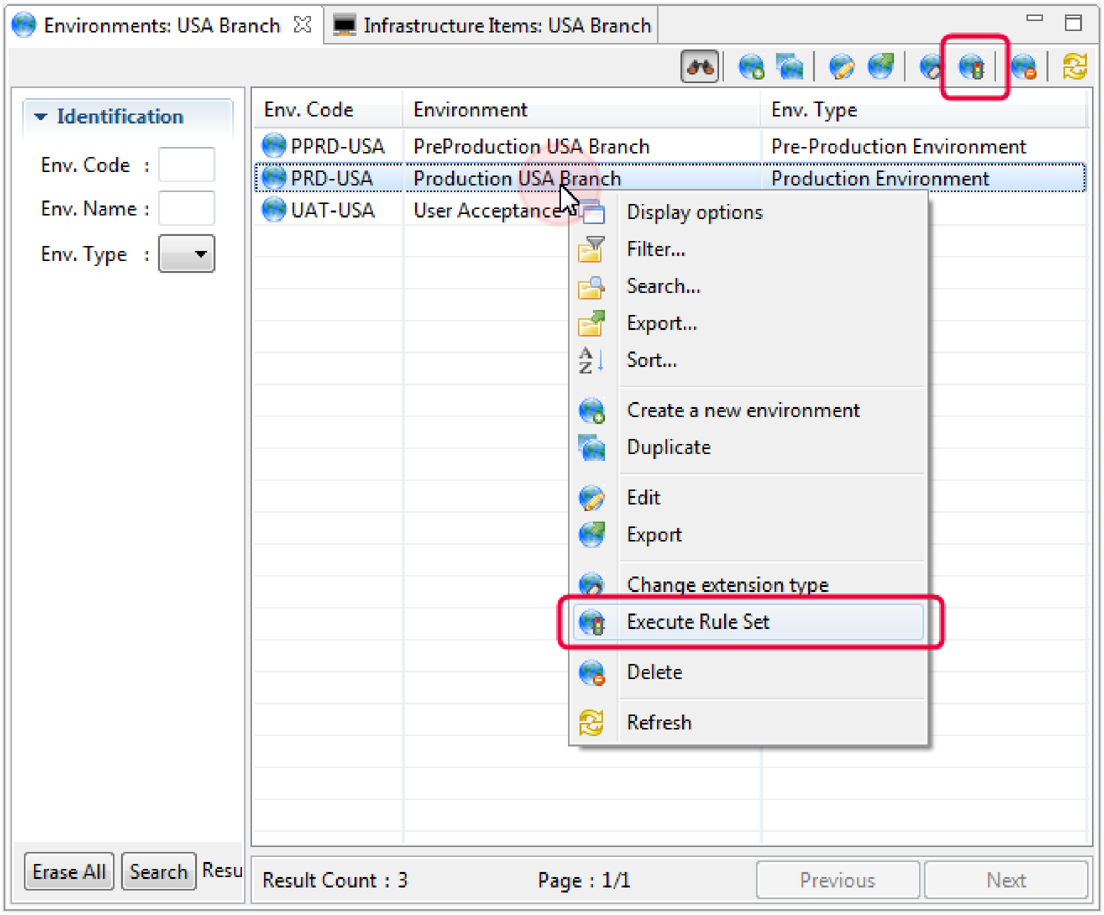
Step 2 Define the rule set's code and name. These values can be edited later. Click **Finish**.

Step 3 Add **compliance rules** to the **rule set**. In the editor, click the **Add Rule** button and select from the list of rules. Ctrl+click to add multiple rules to the set.

Result Rule sets are displayed in the **Compliance Rule Sets** search view.

26.4.2 Executing compliance rule sets

Compliance rule sets can be executed on all of the infrastructure items that belong to an environment or on individual infrastructure items.



Env. Code	Environment	Env. Type
PPRD-USA	PreProduction USA Branch	Pre-Production Environment
PRD-USA	Production USA Branch	Production Environment
UAT-USA	User Acceptance	

Result Count : 3 Page : 1/1

Figure 47: Executing Compliance Rule Cets

Select an environment or infrastructure item in their respective search views and click on the **execute rule set** icon or right-click on one and select **Execute rule set**. Choose the rule set to run from the dialog. Click **OK**.

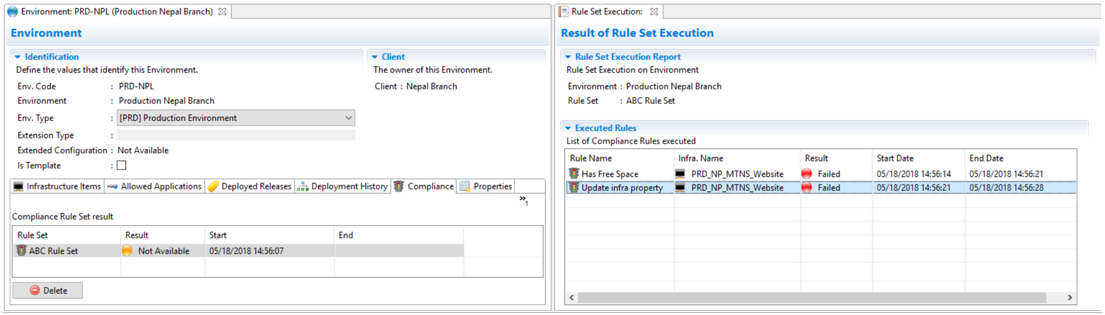
For each infrastructure item, the compliance rule's **Execution Start Date** and **End Date** is specified. Each compliance rule in a set are executed consecutively, one after the other.

> **Note**
>
> When executing a set of rules on an environment, all the rules of the set are applied to all the infrastructure items defined in the environment.
>
> When executing a set of rules on an infrastructure item, all the rules in the set are applied to the item.

26.5 Understanding compliance rule logs and compliance reports

Logs report the details of what happened when testing an infrastructure item against the rule's requirements.

Reports summarize the execution results of a compliance rule set for each infrastructure item in an environment.



Environment

Identification
Define the values that identify this Environment.

Env. Code	:	PRD-NPL
Environment	:	Production Nepal Branch
Env. Type	:	[PRD] Production Environment
Extension Type	:	
Extended Configuration	:	Not Available
Is Template	:	□

Client
The owner of this Environment.
Client : Nepal Branch

Compliance Rule Set result

Rule Set	Result	Start	End
ABC Rule Set	Not Available	05/18/2018 14:56:07	

Result of Rule Set Execution

Rule Set Execution Report
Rule Set Execution on Environment

Environment : Production Nepal Branch
Rule Set : ABC Rule Set

Executed Rules
List of Compliance Rules executed

Rule Name	Infra. Name	Result	Start Date	End Date
Has Free Space	PRD_NP_MTNS_Website	Failed	05/18/2018 14:56:14	05/18/2018 14:56:21
Update infra property	PRD_NP_MTNS_Website	Failed	05/18/2018 14:56:21	05/18/2018 14:56:28

Figure 48: The Compliance Reports view

The report displays all the details of the compliance rule's execution. Each entry is a link to the execution's log file. Double clicking on an entry opens the log file.

🟢 Succeeded	The infrastructure item that complied with the rule's requirement.
🔴 Failed	The infrastructure item that failed to comply with the rule's requirement.
🟠 Not Available	The result is not known yet. The compliance rule's execution is not completed.
⚪ Not Compatible	The compliance rule is not compatible with the infrastructure item type.

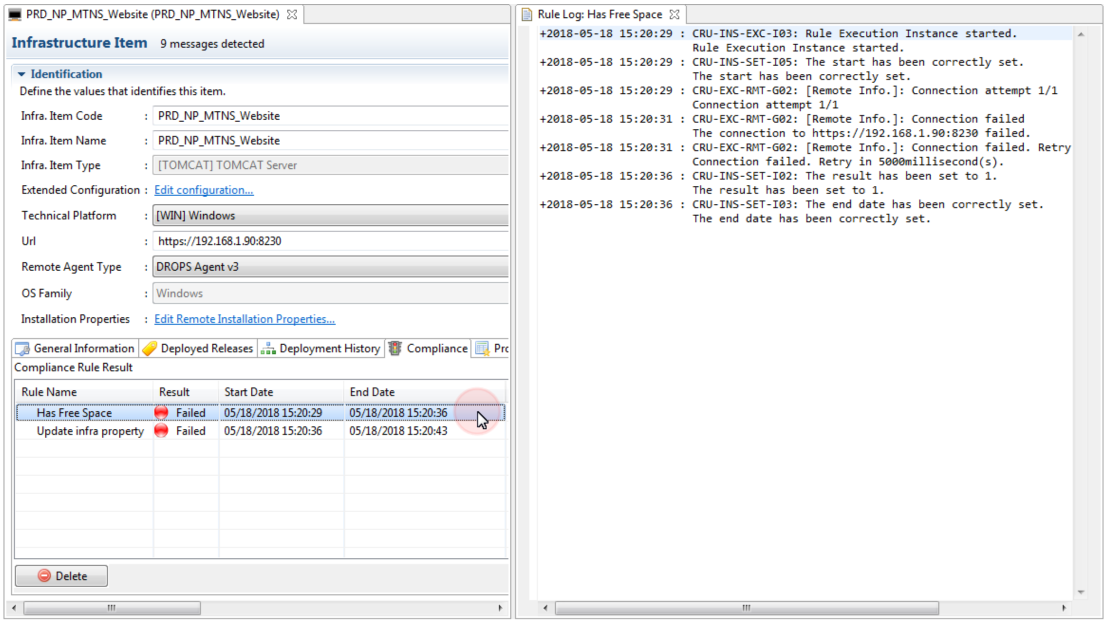
Table 9: Compliance Rule results

26.5.1 Viewing compliance rule execution logs

Compliance rules' execution logs are managed in the **Infrastructure Item** editor. Follow the subsequent steps to view compliance rules' execution logs.

Step 1 Select the **Compliance** tab from the **Infrastructure Item** editor.

Step 2 Click on the result line in the list to view the log file.



PRD_NP_MTNS_Website (PRD_NP_MTNS_Website)

Infrastructure Item 9 messages detected

Identification
Define the values that identifies this item.

Infra. Item Code :	PRD_NP_MTNS_Website
Infra. Item Name :	PRD_NP_MTNS_Website
Infra. Item Type :	[TOMCAT] TOMCAT Server
Extended Configuration :	Edit configuration...
Technical Platform :	[WIN] Windows
Url :	https://192.168.1.90:8230
Remote Agent Type :	DROPS Agent v3
OS Family :	Windows
Installation Properties :	Edit Remote Installation Properties...

Compliance Rule Result

Rule Name	Result	Start Date	End Date
Has Free Space	● Failed	05/18/2018 15:20:29	05/18/2018 15:20:36
Update infra property	● Failed	05/18/2018 15:20:36	05/18/2018 15:20:43

Rule Log: Has Free Space

```
+2018-05-18 15:20:29 : CRU-INS-EXC-I03: Rule Execution Instance started.
                       Rule Execution Instance started.
+2018-05-18 15:20:29 : CRU-INS-SET-I05: The start has been correctly set.
                       The start has been correctly set.
+2018-05-18 15:20:29 : CRU-EXC-RMT-G02: [Remote Info.]: Connection attempt 1/1
                       Connection attempt 1/1
+2018-05-18 15:20:31 : CRU-EXC-RMT-G02: [Remote Info.]: Connection failed
                       The connection to https://192.168.1.90:8230 failed.
+2018-05-18 15:20:31 : CRU-EXC-RMT-G02: [Remote Info.]: Connection failed. Retry
                       Connection failed. Retry in 5000millisecond(s).
+2018-05-18 15:20:36 : CRU-INS-SET-I02: The result has been set to 1.
                       The result has been set to 1.
+2018-05-18 15:20:36 : CRU-INS-SET-I03: The end date has been correctly set.
                       The end date has been correctly set.
```

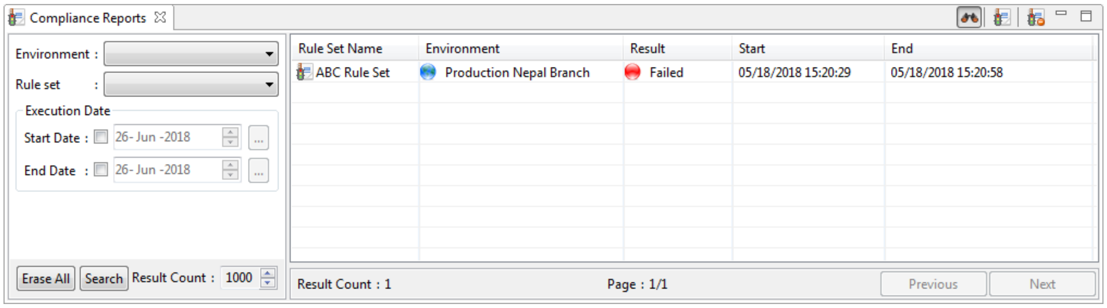
Figure 49: The Infrastructure Item Compliance log tab

26.5.2 Searching for compliance reports

All the compliance reports for any environment are managed in the **Compliance Report** search view. Reports are also available from the **Compliance** tabs in the individual **Environment**.

Follow the subsequent steps to search for compliance reports.

Step 1 Select **Compliance Reports** from the **Compliance** node in the **Configuration Explorer**.



Compliance Reports

Environment :	
Rule set :	

Execution Date

Start Date :	□	26- Jun -2018
End Date :	□	26- Jun -2018

Result Count : 1000

Rule Set Name	Environment	Result	Start	End
ABC Rule Set	Production Nepal Branch	● Failed	05/18/2018 15:20:29	05/18/2018 15:20:58

Result Count : 1 Page : 1/1

Figure 50: The Compliance Reports search view

Step 2 When the search view is opened, the list of compliance reports may be empty. Use the search criteria sections to filter the search.

- To search for a report by its **environment** or **rule set**, enter all or part of the information in the corresponding field.
- To search for a report by the **Execution Start Date** or **End Date**, tick the corresponding box and select the date from the calendar.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.

Result The **Compliance Report** search view displays all of the reports that meet the search criteria.

26.6 Deleting compliance rules and rule sets

> ***Warning!***
>
> Deleted compliance rules cannot be accessed or recovered. They are no longer available for any rule set in which they may have been included.
>
> Deleting a rule set does not delete the individual rules that are included but it can no longer be accessed or used to test.

Select the compliance rule or rule set in its corresponding search view and either click the delete icon or right-click on it and select **Delete**. Click **OK** to confirm or click **Cancel** to keep the compliance rule/set.

27 Scripts

Required Roles	n/a
Required Rights (by category)	Scripts
Perspective	Configuration Management
Access	DROPS Server → Configuration → Scripts

Chapter Summary

27.1 Creating scripts 151
27.2 Editing scripts 152
27.3 Importing scripts 157
27.4 Exporting scripts 157
27.5 Executing scripts 158
27.6 Deleting scripts 159

Scripts contain one or more Ant tasks and provide coded instructions to carry out jobs for DROPS and/or target systems.

Actions in the deployment process diagrams call Ant scripts. These Ant scripts are sets of Ant tasks. These Ant tasks are written in Java and are stored in libraries (*.jar* files). When Ant executes a script, it looks for these libraries to find the tasks build into each script. The path to the directory holding the libraries is called the ANTLIB.

DROPS provides a number of built-in scripts by default which cannot be edited but can be duplicated then modified if you need to make adjustments. The ANTLIB that contains all of these built-in scripts is included in the installation. The default, built-in ANTLIB can be updated when the database is updated to recover any modifications to default scripts or new scripts we create.

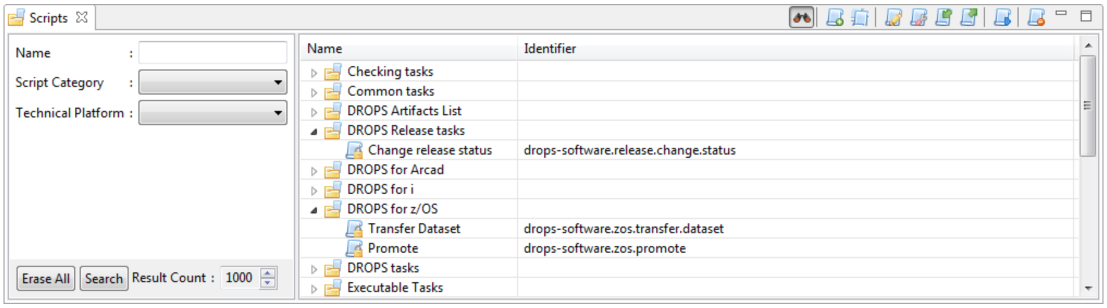
There are two ways to create brand-new, customized scripts. Use external, valid Ant projects to create new scripts that use previously-defined tasks or create new scripts manually from scratch directly in the DROPS Studio. Scripts that you create can be modified at any time. Any kind of script can be exported and shared (imported into another DROPS Server). You can also duplicate built-in scripts to copy information from them and customize a version for yourself.

Version control for DROPS scripts is an important feature that enables you to “close” valid scripts and modify different versions of them as their purpose(s) evolve. Deployment processes also use the same version control techniques. When a process is validated, both it and all of the scripts used inside it are locked to ensure that approved processes and their scripts cannot be edited.

Note
This documentation does not cover the content of Ant projects.

Reference
For more information about how to create a valid Ant project, refer to the Apache Ant documentation.

For detailed information about the parameters for each built-in script from DROPS Software, refer to the *DROPS Script Reference Guide* or the **Documentation** section in each script.



Name	Identifier
Checking tasks	
Common tasks	
DROPS Artifacts List	
DROPS Release tasks	
Change release status	drops-software.release.change.status
DROPS for Arcad	
DROPS for i	
DROPS for z/OS	
Transfer Dataset	drops-software.zos.transfer.dataset
Promote	drops-software.zos.promote
DROPS tasks	
Executable Tasks	

Figure 51: The Scripts search view

The **Scripts** search view displays all of the scripts, ordered by category. The **Identifier** column displays the unique ID of each script when a category is expanded.

27.1 Creating scripts

There are multiple ways to create scripts:

1. Create a brand new script using the **Create a new script** wizard.
2. Duplicate an existing script to quickly create a copy of the script that can be modified.
3. Create a new version of an existing script to edit.

27.1.1 Create a new script

Creating a new script enables you to duplicate it later.

Follow the subsequent steps to create a new script.

Step 1 Click the create icon in the **Scripts** search view or right-click anywhere in the view and select **Create a new script** to open the **Create a new script** wizard.

Step 2 Define the Identifier and Name of the new script. These values can be edited later.

Step 3 Select the Execution Platform and Script Language in the drop-down lists. The technical platform is not mandatory.

Click **Next >** to continue.

Step 4 The new script's version is automatically 1.0. Document the initial version of the script by adding comments.

Click **Next >** to continue.

Step 5 The final page of the dialog enables you to select an Ant file to use as the content of the script.

Step 6 Click **Finish**.

Result New scripts are displayed in the **Scripts** search view. Each script is nested under its corresponding category.

27.1.2 Duplicate a script

Duplicating a script is a quick way to apply existing script details or content to a new script.

Step 1 To open the **Duplicate script** wizard, either select an existing script in the list then click the duplicate icon in the **Scripts** search view, or right-click on it and select **Duplicate**.

Script content can also be duplicated by clicking the duplicate icon in an open script editor.

Step 2 Select the version of the original script to duplicate. By default, the most recent version is selected.

Click **Next >** to continue.

Step 3 Define the Identifier and Name of the new script. These values can be edited later but must be unique.

Step 4 Select the Execution Platform and Script Language in the drop-down lists. The technical platform is not mandatory.

Step 5 Click **Finish**.

Result Duplicated scripts are displayed in the **Scripts** search view. Each script is nested under its corresponding category.

27.1.3 Create a new version of a script

To create a new version of the script click the **Create** button in the Version section of the script editor. The **Create a new script** wizard opens. To delete a version of the script, select it and click **Delete**.

Version Upgrade

Enter a description of the new version of the script.

Select **Major Upgrade** to increase the version number from 1.0 to 2.0 for example.

Select **Minor Upgrade** to increase the version number from 1.0 to 1.1 for example.

ANT Project

The final page of the dialog enables you to select an Ant file to use as the content of the script.

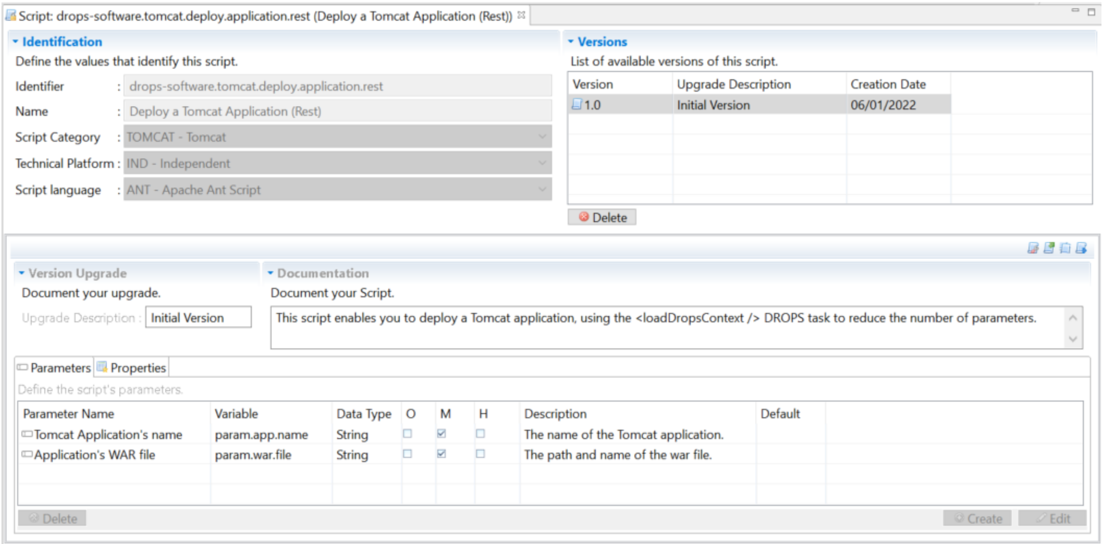
27.2 Editing scripts

The Ant project content of a script can be edited as well as the identification details and the variables it uses.

27.2.1 Edit script details

To edit script identification details, either right-click on it then select **Edit**, select the item then click the edit icon or double-click it. The **Script** editor is opened where all of the entity's details are managed.

> **Note**
>
> If a script is built-in, its details cannot be edited. The **Script** editor is opened in read-only mode.



Script: drops-software.tomcat.deploy.application.rest (Deploy a Tomcat Application (Rest))

Define the values that identify this script.

Identifier	drops-software.tomcat.deploy.application.rest
Name	Deploy a Tomcat Application (Rest)
Script Category	TOMCAT - Tomcat
Technical Platform	IND - Independent
Script language	ANT - Apache Ant Script

List of available versions of this script.

Version	Upgrade Description	Creation Date
1.0	Initial Version	06/01/2022

Document your upgrade.

Upgrade Description : Initial Version

Document your Script.

This script enables you to deploy a Tomcat application, using the <loadDropsContext /> DROPS task to reduce the number of parameters.

Define the script's parameters.

Parameter Name	Variable	Data Type	O	M	H	Description	Default
Tomcat Application's name	param.app.name	String	□	☑	□	The name of the Tomcat application.	
Application's WAR file	param.war.file	String	□	☑	□	The path and name of the war file.	

Figure 52: The Script editor

> ***Reference***
>
> For more information about the different actions managed from the tabs in the **Script** editor that are not described in this chapter, refer to:
>
> - Properties on page 412

Identifier

A unique string that identifies scripts. This label is used throughout DROPS to select the entity.

Name

A label used to give additional information about the script. This label is used throughout DROPS to select the entity.

Script Category

The category under which the script is nested.

Execution Platform

[*Optional*] If the script should or can only be used on a certain platform, select it from the drop-down list.

Script Language

Select the scripting language of the script's content in the drop-down list. Note that this language refers to the language used to write the script's actions. Ant scripts support the use of other languages for its content.

Version Upgrade

Enter a description of the current version of the script.

Documentation

Enter a general description of the script.

If the content of the script contains variables, create references to them in the **Parameters** tab. The values for these variables are defined in the **Script** tab in the deployment diagram's actions' properties.

> ***Reference***
>
> For more information about defining the values for a script's parameters, refer to Edit deployment action properties on page 187.

Variables in the script's content are always referenced using the following syntax: ${*Variable name*}.

Click **Create** to open the **Script Parameter Definition** dialog.

To edit a parameter, select it and click **Edit**.

To delete a parameter, select it and click **Delete**.

Parameter name

A unique string that identifies the parameter.

Variable Name

The exact text used in the ${*Variable name*} variable found in the content of the script.

Data Type

String: The content of the parameter is intended to be a string of characters (text).

Numeric: The content of the parameter is intended to be a string of numbers.

Boolean: The content of the parameter is intended to be 0/1, True/False or Yes/No.

Date: The content of the parameter is intended to be a date.

Default Value

Enter a default value for the parameter. This can be overridden when editing the deployment action parameters in the deployment process diagram.

Output Parameter

Tick this checkbox if the script should produce an output to be sent back in order to create another parameter to be used after the task is over in another part of the deployment process diagram.

Is Mandatory

Tick this checkbox if the parameter is mandatory and a value must be defined.

Is Hidden

Tick this checkbox to hide the value of sensitive script parameters during their logging.
When a script containing hidden values is imported or exported, the hidden parameter appears as such in the XML file.

Description

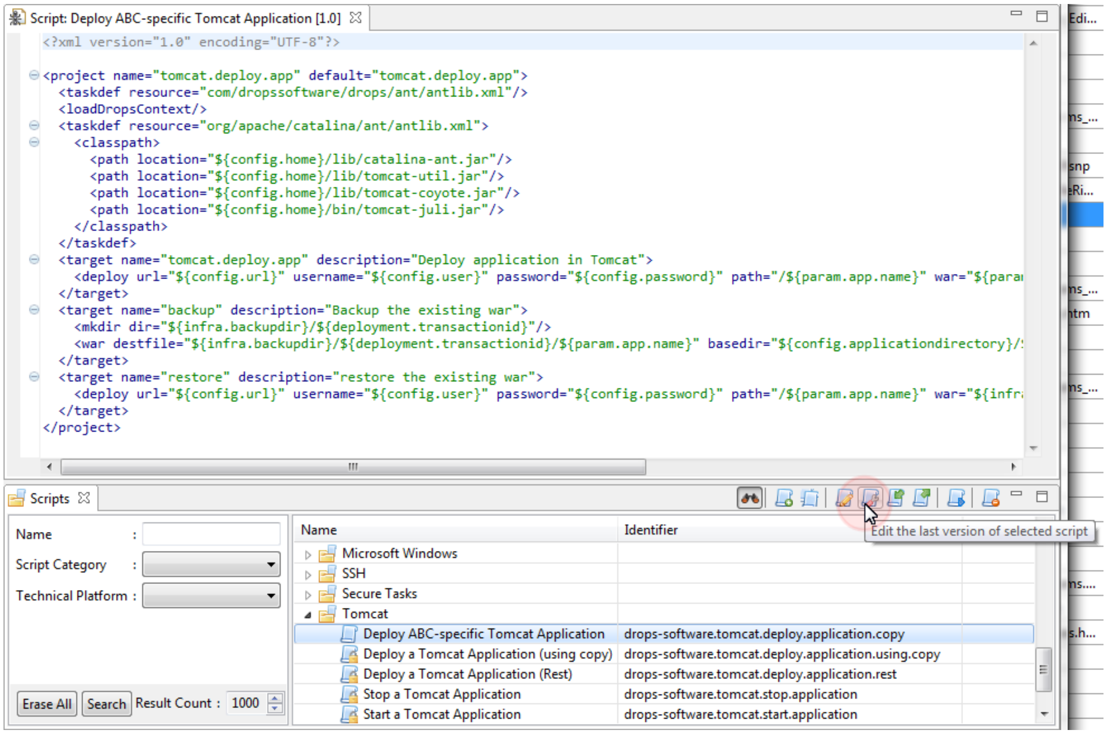
Enter a description of the parameter.

Save the changes (, Ctrl+S or **File** > **Save**).

27.2.2 Edit script content

Select a script from the list and click the edit content icon or right-click on a script and select **Edit Content**. Script content can also be edited by clicking the edit content icon in the **Script** editor.

> **Note**
>
> If a script is “built-in”, its content cannot be edited. The **Script** content editor is opened in read-only mode.



Script: Deploy ABC-specific Tomcat Application [1.0]

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="tomcat.deploy.app" default="tomcat.deploy.app">
  <taskdef resource="com/dropssoftware/drops/ant/antlib.xml"/>
  <loadDropsContext/>
  <taskdef resource="org/apache/catalina/ant/antlib.xml">
    <classpath>
      <path location="${config.home}/lib/catalina-ant.jar"/>
      <path location="${config.home}/lib/tomcat-util.jar"/>
      <path location="${config.home}/lib/tomcat-coyote.jar"/>
      <path location="${config.home}/bin/tomcat-juli.jar"/>
    </classpath>
  </taskdef>
  <target name="tomcat.deploy.app" description="Deploy application in Tomcat">
    <deploy url="${config.url}" username="${config.user}" password="${config.password}" path="/${param.app.name}" war="${para
  </target>
  <target name="backup" description="Backup the existing war">
    <mkdir dir="${infra.backupdir}/${deployment.transactionid}"/>
    <war destfile="${infra.backupdir}/${deployment.transactionid}/${param.app.name}" basedir="${config.applicationdirectory}/
  </target>
  <target name="restore" description="restore the existing war">
    <deploy url="${config.url}" username="${config.user}" password="${config.password}" path="/${param.app.name}" war="${infr
  </target>
</project>
```

Scripts

Name :
Script Category :
Technical Platform :

Erase All | Search | Result Count : 1000

Edit the last version of selected script

Name	Identifier
Microsoft Windows	
SSH	
Secure Tasks	
Tomcat	
Deploy ABC-specific Tomcat Application	drops-software.tomcat.deploy.application.copy
Deploy a Tomcat Application (using copy)	drops-software.tomcat.deploy.application.using.copy
Deploy a Tomcat Application (Rest)	drops-software.tomcat.deploy.application.rest
Stop a Tomcat Application	drops-software.tomcat.stop.application
Start a Tomcat Application	drops-software.tomcat.start.application

Figure 53: Edit script content

The **Script** content editor is opened where you can write or edit the content of the template's script. Editing the content of a script can be done directly in DROPS. Eclipse's Ant Editor offers shortcuts (Ctrl+space) to provide easy access to all of the Ant project templates, tasks, parameters and variables available.

> ⚠ **Important!**
> When creating or modifying scripts, the content must be a valid Ant project.

The **Script** content editor is opened where you can write or edit the content of the template's script. Editing the content of a script can be done directly in DROPS. The script editor offers shortcuts (Ctrl+space) to provide easy access to all of the chosen scripting language's templates, tasks, parameters and variables available.

> **Note**
> If a new script is created without content (no script file uploaded), a template is automatically added to the content editor.

DROPS uses the ANT language for the transport layer of the tasks but other scripting languages can be used to write the script's content. The **Script language** you selected when creating the script determines the template use in the script content.

All supported languages benefit of DROPS' integrated specific editors with syntax coloring and simple syntax checks.

All integrated languages benefit of DROPS' contextual properties and parameters such as project properties, including the *drops.target* which is the current target of the executed action.

The following scripting languages are delivered with Drops without further configuration.

- Apache Ant Script
- Apache Groovy Script
- Apache Groovy Ant Task Script
- JavaScript (Nashorm Implementation)
- JRuby Script
- Jython Script

To integrate the following scripting languages, refer to ARCAD Support.

- Bash Script
- Ansible Playbook Script (yaml)
- Batch

Refer to ARCAD Support

Save the changes (, Ctrl+S or **File** > **Save**).

27.3 Importing scripts

Archive (*.*dxml*) or simple (*.*xml*) script files that were previously exported from DROPS can be imported into DROPS again to create new scripts. This is an easy way to recover scripts from other servers.

When imported, the script's Identifier, Script Category and any of the other prepopulated details that come from the original exported script are retrieved from the XML structure's content, even if the values have changed. This ID and the defined category are what registers the script as unique and is prepopulated automatically. In order to create a valid script in DROPS, the imported script must be a valid Ant project and reference these <script> tags.

Importing a script enables you to modify and duplicate it later.

Follow the subsequent steps to import DROPS script files.

Step 1 Click the import icon in the **Scripts** search view, or right-click anywhere in the view and select **Import a script file**.

Step 2 Select the *.dxml* or *.xml* file(s) by clicking the browse button.

Step 3 Click **Finish**.

Result If the selected script file has the same ID as an existing script, DROPS asks if it is a **major** upgrade for the existing script.

- Clicking **No** imports the script as a minor upgrade for the existing script, changing the version number from 1.0 to 1.1, for example.
- Clicking **Yes** imports the script as a major upgrade for the existing script, changing the version number from 1.0 to 2.0, for example.

If an imported script file references a category that does not exist in DROPS, a new category is created with the name defined in the script.

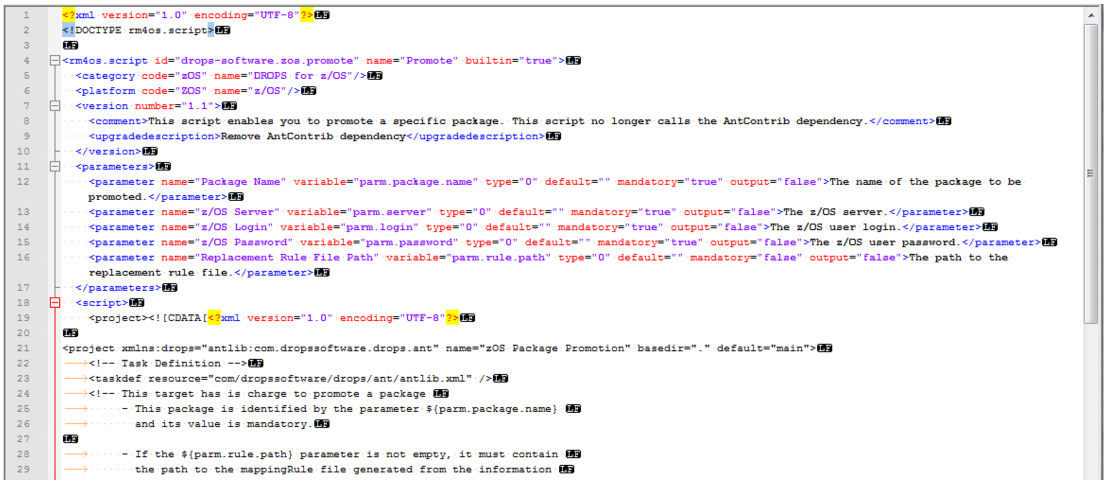
If the script file has a unique ID, it is imported as a new script under the category defined (version 1.0).

27.4 Exporting scripts

Exporting scripts is useful when sharing scripts between multiple DROPS servers or users. When exported, DROPS scripts contain all of the details entered in the **Script** editor. Scripts that are exported from one server can be imported into another. Importing scripts into the new server is made easy because DROPS interprets the ID and category, as well as any version or other information, and prepopulates these fields in the editor automatically.

Note

To use an external script file in DROPS, create a new script and attach the external file using the **Create a new script** wizard.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rm4os.script>

<rm4os.script id="drops-software.zos.promote" name="Promote" builtin="true">
  <category code="zOS" name="DROPS for z/OS"/>
  <platform code="ZOS" name="z/OS"/>
  <version number="1.1">
    <comment>This script enables you to promote a specific package. This script no longer calls the AntContrib dependency.</comment>
    <upgradedescription>Remove AntContrib dependency</upgradedescription>
  </version>
  <parameters>
    <parameter name="Package Name" variable="parm.package.name" type="0" default="" mandatory="true" output="false">The name of the package to be
    promoted.</parameter>
    <parameter name="z/OS Server" variable="parm.server" type="0" default="" mandatory="true" output="false">The z/OS server.</parameter>
    <parameter name="z/OS Login" variable="parm.login" type="0" default="" mandatory="true" output="false">The z/OS user login.</parameter>
    <parameter name="z/OS Password" variable="parm.password" type="0" default="" mandatory="true" output="false">The z/OS user password.</parameter>
    <parameter name="Replacement Rule File Path" variable="parm.rule.path" type="0" default="" mandatory="false" output="false">The path to the
    replacement rule file.</parameter>
  </parameters>
  <script>
    <project><![CDATA[<?xml version="1.0" encoding="UTF-8"?>

<project xmlns:drops="antlib:com.dropssoftware.drops.ant" name="zOS Package Promotion" basedir="." default="main">
	<!-- Task Definition -->
	<taskdef resource="com/dropssoftware/drops/ant/antlib.xml" />
	<!-- This target has is charge to promote a package 
	     - This package is identified by the parameter ${parm.package.name} 
	       and its value is mandatory.

	     - If the ${parm.rule.path} parameter is not empty, it must contain 
	       the path to the mappingRule file generated from the information 
```

Figure 54: Example of an exported script

Scripts are exported in *.xml* format.

Follow the subsequent steps to export scripts.

Step 1 Either right-click on the script in the **Scripts** search view then select **Export a script file**, or select the script then click the export icon.

Scripts can also be exported by clicking the export icon in an open **Script** editor.

Step 2 If you want to import the script into another server and have it be considered “built-in” in the new server, click **OK** in the dialog box. If you want to be able to edit the script in another server (consider it not “built-in”), click **Cancel**.

Step 3 Select the location in which to save the script and change the **File name** if needed. By default, scripts are exported with the file name *<ID>.xml*.

If a script has multiple versions, the name is changed to *<ID>_<VersionNumber>.xml*.

Step 4 Click **Save**.

27.5 Executing scripts

To test that a script functions correctly before using it in a deployment process, execute it. The **Execute Script** wizard recreates the external environment in a test scenario.

Step 1 To open the **Execute Script** wizard, either select a script from the **Scripts** search view and click the execute icon, or right-click on a script and select **Execute Script**.

Script content can also be executed by clicking the execute icon in an open **Script** editor.

Step 2 Modify the script's parameters in the **Value** column. This page recreates the **Script** tab in the deployment diagram's action's properties.

If a parameter is not given a value, it will not be referenced when the script is tested.

Variables can be used as parameter values.

> ***Reference***
> For more information about variables, refer to Substitution variables in the appendix.

Click **Next >** to continue.

Step 3 Select the infrastructure item on which to test the script. It is suggested to test the script on the same item it is intended to be run on in the deployment process.

The list of infrastructure items is prepopulated with all of the Items defined on the current server.

Click **Next >** to continue.

Step 4 Attach an external properties file, if needed. Recreating the access properties required during deployment may be required.

Click the … browse button to select a .properties file.

> ***Example***
> Every deployment is assigned a Transaction ID in order to facilitate rollback. This number will not be generated in a test environment. Creating an external properties file that contains a dummy Transaction ID enables you to test the rollback function in a script.

Step 5 Click **Finish**.

Result The DROPS Agent on the target infrastructure item produces the results log. It is opened automatically in the default text editor enabling you to verify that the script executed correctly.

27.6 Deleting scripts

> ***Warning!***
> Deleted scripts cannot be accessed or recovered.

To delete a script, search for it in the **Scripts** search view. Either right-click on it and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the script.

28 Deployment processes

Required Roles	n/a
Required Rights (by category)	Clients Environments Deployment Process Diagrams Scripts
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) Environments → *click on Environment* → Process List

Chapter Summary

28.1 Accessing deployment processes 160
28.2 Creating deployment processes 161
28.3 Changing a deployment process' status 164
28.4 Defining default deployment processes 164
28.5 Deleting deployment processes 165
28.6 Building deployment process diagrams 165
28.7 Editing deployment process properties 175

All of the entities configured in the artifact and Infrastructure repositories come together during the deployment process. Application(s) and the component(s) to deploy are joined with the target environment (s) and specific infrastructure item(s) in deployment process diagrams. It is in these diagrams, the key elements in your process, that the final location for the content to deploy and the actions to carry out during deployment are defined.

The deployment process is represented by the diagram in order to easily define and display the activities and stages required to successfully deploy releases to each infrastructure item.

28.1 Accessing deployment processes

The **Process List** is displayed automatically when you click on an environment in the **Environments** search view.

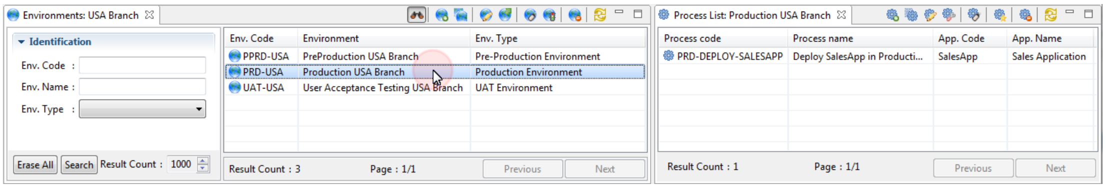
Note

It is recommended to open the **Process List** in a different area than the **Environments** view. If both views are tabs in the same area, it is difficult to access the **Environment** editor.

Reference

For more information about the **Environments** view, refer to Environments on page 96.

The **Process List** displays all of the processes defined for the selected environment.



Environments: USA Branch

Identification

Env. Code :

Env. Name :

Env. Type :

Erase All | Search | Result Count : 1000

Env. Code	Environment	Env. Type
PPRD-USA	PreProduction USA Branch	Pre-Production Environment
PRD-USA	Production USA Branch	Production Environment
UAT-USA	User Acceptance Testing USA Branch	UAT Environment

Result Count : 3 Page : 1/1 Previous Next

Process List: Production USA Branch

Process code	Process name	App. Code	App. Name
PRD-DEPLOY-SALESAPP	Deploy SalesApp in Producti...	SalesApp	Sales Application

Result Count : 1 Page : 1/1 Previous Next

Figure 55: The Process List

28.2 Creating deployment processes

There are two ways to create deployment processes:

1. Create a new deployment process using the **Create a new deployment process** wizard.
2. Duplicate a deployment process. Duplicating creates a copy of an existing process that can be modified.

28.2.1 Create a new deployment process

Creating a new process enables you to duplicate it later.

Follow the subsequent steps to create a new deployment process.

Step 1 Click the create icon in the **Process List** or right-click anywhere in the view and select **Create a new deployment process** to open the **Create a new deployment process** wizard.

Step 2 Define the Process Code and Process Name. These values can be edited later.

Step 3 Select the application to associate with the process. This is the application that contains the artifacts to deploy to the environment.

> **Note**
> If the application only has one component, the component will be automatically assigned to be deployed to (all of) the infrastructure item(s) selected in the next step.

> **Important!**
> Make sure to select the correct application! Once you click **Finish** and create a deployment process, you *cannot change* the application associated with it.

Step 4 Select the infrastructure item(s) to use in the deployment process diagram. The list of items is prepopulated with all of the items defined for the selected environment. If no item is selected, the deployment process diagram will include all of the infrastructure items defined for the environment.

> **Note**
> infrastructure items can be added and removed from the diagram after it is created, also.

Click **Finish**.

Result New deployment processes are available in the **Process List** after refreshing the list. It is very important to continue configuring a process after it's created before using it.

> ***Reference***
> For more information about building and configuring deployment process, refer to Building deployment process diagrams on page 165 and Editing deployment process properties on page 175.

28.2.2 Duplicate a deployment process

Duplicating a deployment process defined for an environment is a quick way to apply an existing method of deploying content to another set of infrastructure items on a different environment.

> ***Example***
> You have similar infrastructure items in two different environments. They both require the same deployment process. Instead of creating the same process twice, create it once then duplicate it and manually define which of the original items' processes should be "copied" and applied to the similar item in the second environment.

> **Important!**
> You cannot change the application associated with a process when you duplicate it. The new process will deploy the same application as the original.

Follow the subsequent steps to duplicate an existing deployment process in order to create a new process.

Step 1 To open the **Duplicate a deployment process** wizard, either select an existing process in the **Process List** then click the duplicate icon, or right-click on it and select **Duplicate**.

Step 2 Define the new process' Process Code and Process Name. These values can be edited later but must be unique.

Click **Next >** to continue.

Step 3 Select the version of the existing process to duplicate.

Click **Next >** to continue.

Step 4 Define the environment for the new process.

> **Note**
> If you do not tick the **Change the target Environment** checkbox, the new process will be copied to the current process' environment and use the original infrastructure items.

1. To change the environment and the infrastructure items for the new process, tick the **Change the target Environment** checkbox, then select the new environment from the drop-down list.

   The list of environments to select is prepopulated with all of the environments defined for the current server that include the original process' application in the **Allowed Applications** manager. You cannot change the application the process is linked to when duplicating it, so it must also be allowed to be deployed to the new environment.

   Select the original environment to duplicate the process using different infrastructure items.

   > ⚠ **Important!**
   > Make sure to select the correct environment. Once you click **Finish** and create the process, you cannot change the associated environment.

2. To define which of the new environment's infrastructure items should use the processes defined, map the original items to the new environment's items. The items from the original process already contain defined stages and actions. In order to ensure their diagrams are matched to similar infrastructure items in the new process, define which of items in the original process should be mapped to each target infrastructure item.

   The infrastructure items that the original process references are listed in the **Source Infrastructure Item** column. Match an original item to an item in the new environment by double-clicking in the corresponding item's row in the **Target Infrastructure Item** column.

   The **Infrastructure Item Selection** dialog is prepopulated with all of the infrastructure items defined in the selected environment that are also the same type as the original infrastructure item. Requiring that the types of both the original and new items are the same when duplicating a process ensures that the actions defined in the process are applicable to both infrastructure items.

   > ⚠ **Important!**
   > Make sure to select the correct infrastructure item in the mapping dialog. Once you click **OK** and map the process, you cannot change the individual infrastructure item. To reset all of the mapped items, reselect the environment from the **Target Environment** drop-down list.

   All of the original items must be mapped to a new infrastructure item to finish the duplication. You can edit, add and remove items from the new process once it is created.

Step 5 Click **Finish**.

Result New deployment processes are available in the **Process List** after refreshing the list. It is very important to continue configuring a process after it’s created before using it.

> ***Reference***
> For more information about building and configuring deployment process, refer to Building deployment process diagrams on

page 165 and Editing deployment process properties on page 175.

28.3 Changing a deployment process' status

To change the deployment process status, either right-click on it in the **Process List** then select **Change Status** or select it, then click the Change Status icon.

The list of available deployment process statuses is prepopulated with the statuses defined by the administrator in the **Preferences** menu.

If a deployment process is marked as **Production Status**, it means that the process has been validated and cannot be modified.

Reference

For more information about defining statuses, refer to the *DROPS Configuration Guide*.

28.4 Defining default deployment processes

Defining a default status for an environment is not required but it is useful when creating multiple deployment process instances at the same time using environment groups. The default deployment process for an environment is always used as the process to deploy when multiple deployment process instances are created using environment groups. Default deployment processes are indicated by the default icon in the **Process List**.

Because environment groups can share more than one common application, default deployment processes are defined per application. An environment can have multiple default processes but only one per application. If a different deployment process with the same application is already defined as the default, selecting another and clicking the Default icon will remove the title from the previous process.

To define the default deployment process for an environment, select it from the **Process List** then either right-click and select **Set As Default** or click the set default icon in the local toolbar.

To remove the default status of a deployment process, select it from the **Process List** then either right-click and select **Remove As Default** or click the set default icon in the local toolbar.

Reference

For more information about the entities mentioned above, refer to:

- Environments on page 96
- Environment groups on page 104
- Create multiple instances to deploy one release to an environment group on page 266

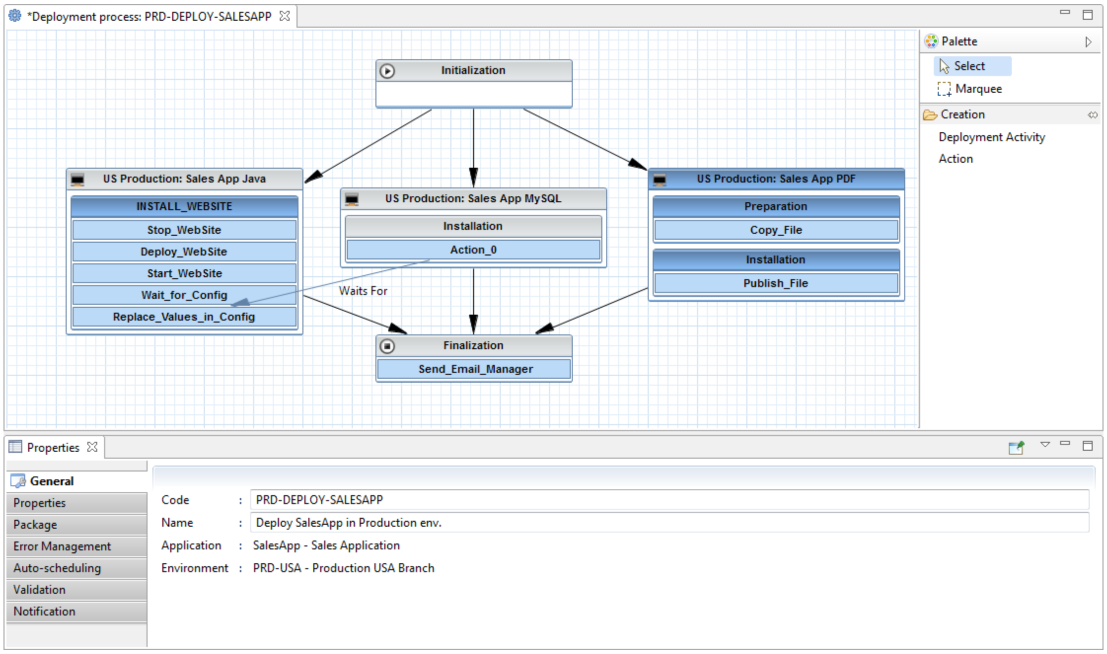
28.5 Deleting deployment processes

Warning!
Deleted deployment processes cannot be accessed or recovered. Any deployment process instances created from the process are deleted as well.

To delete a deployment process, either right-click on it in the **Process List** and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the deployment process.

28.6 Building deployment process diagrams

The deployment process diagram is a visual representation of the deployment process for one specific environment. All of the actions and parameters that define a deployment process are configured in the diagram. Diagrams can be managed manually or can call pre-defined templates that can be used as building blocks to include consistent parameters in multiple diagrams.



Code : PRD-DEPLOY-SALESAPP
Name : Deploy SalesApp in Production env.
Application : SalesApp - Sales Application
Environment : PRD-USA - Production USA Branch

Figure 56: The Deployment Process editor

Reference
For detailed definitions of the different elements in the process diagram, refer to the Glossary.

28.6.1 Manage deployment activities

Activities are the sum of all the actions taken for an infrastructure item or the initialization or finalization phases during the deployment process. They contain the different deployment actions to carry out.

Deployment actions are driven by scripts which each carry out a specific Ant project. Actions are added directly to the ▶ **Initialization** and ■ **Finalization** activities during which no artifacts are managed. These first and last phases' tasks should not be related to actually deploying content. They cannot be removed from the deployment process diagrams but do not require actions.

Infrastructure item activities are for individual items and are unique because each one may have specific requirements or because the components deployed to multiple items may need to be treated differently depending on the target setup. Therefore, the actions needed to properly deploy content to a target service are defined inside each individual infrastructure item activity in the diagram. Artifacts are deployed during these phases and actions specific to deploying content are divided into stages. Stages define where and when actions are carried out.

By default, the infrastructure items associated with the environment, and selected when creating the deployment process, are included in a new deployment process diagram. They are represented by infrastructure item activities. New activities can be added only if new infrastructure items are added to the environment or all of the items were not selected when the process was created.

28.6.1.1 Create new deployment activities

Note
New **Initialization** or **Finalization** Activities cannot be added.

Follow the subsequent steps to create a new infrastructure item activity.

Step 1 Click the **Deployment Activity** node in the right-hand menu.

Step 2 Draw a box in the grid for the new activity, guided by the shaded outline.

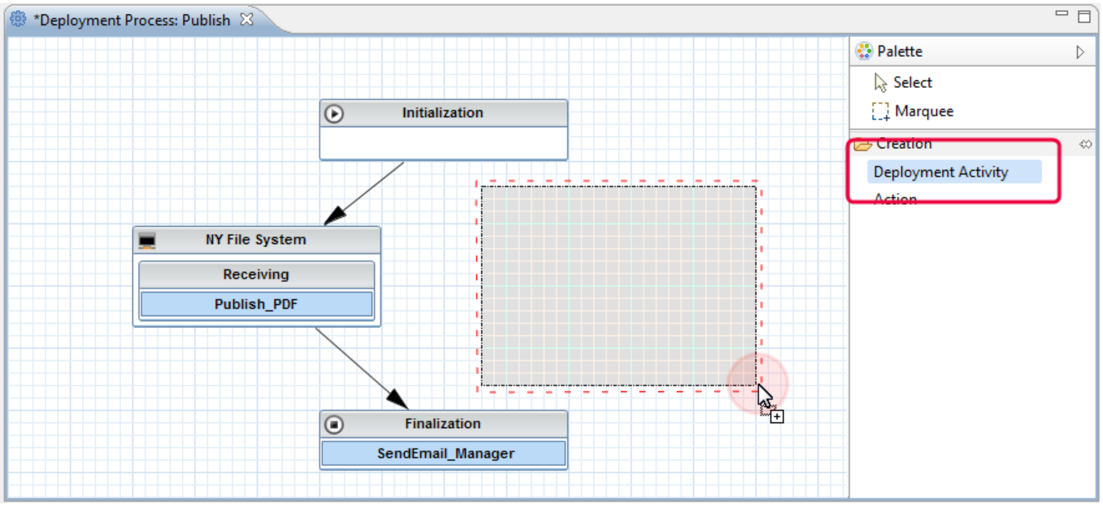


Figure 57: Create a new infrastructure item activity

Step 3 From the dialog box that is automatically displayed, select the corresponding infrastructure item and click **OK**.

> **Note**
> If the dialog box is not displayed, there are no infrastructure items associated with the environment that can be added to the diagram. All of the items must already have an activity in the diagram.

Result The new activity is automatically linked to the infrastructure item and is ready to be edited and linked to its parent and child activities.

Step 4 Click on the activity to modify its properties.

> ***Reference***
> For more information about editing an activity's properties, refer to Edit deployment activity properties on page 182.

Step 5 Right-click on the activity and select **Edit** > **Component** to view and modify the component defined for the infrastructure item if one has already been defined in the activity's properties.

If the application associated with the process only has one component, the component is automatically assigned to be deployed to (all of) the infrastructure item(s). It can be changed if more components have been added to the application since the initial creation of the process.

> ***Warning!***
> Infrastructure item activities *must* be linked to a component in order to save the deployment process. If you do not add a component to every activity and you close the deployment process diagram editor, all changes will be lost.

> ***Reference***
> For more information about editing components, refer to Components on page 67.

Step 6 Right-click on the activity and select **Edit** > **Infrastructure Item** to view and/or modify the infrastructure item.

> ***Reference***
> For more information about editing items, refer to Infrastructure items on page 108.

Step 7 Save the changes (, Ctrl+S or **File** > **Save**).

28.6.1.2 Use deployment activity templates

Pre-defined deployment activity templates can be included in any deployment process diagram. They must include pre-defined stages (with actions included) and its properties must always be defined in the template. The only elements in an activity template that can be modified are the Component to deploy and the Transfer options. When a template is used in a diagram, the activity or stage is dark blue, instead of the default gray. The difference in color makes it easy to see which parts of a diagram are inherited from a template and which are unique.

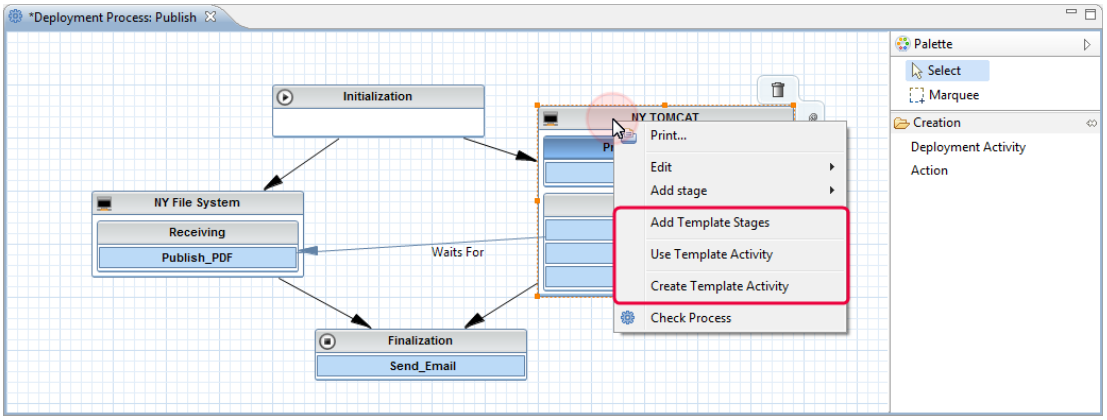


Figure 58: Activity templates in the deployment process diagram

Follow the subsequent steps to use a deployment activity template in a process diagram.

Step 1 Create the box for a new deployment activity and select the infrastructure item to use.

Step 2 Link the activity to its parent and child activities.

> ***Reference***
> For more information about linking activities, refer to Define the order in which to deploy activities below.

Step 3 Right-click on the activity and select **Use Template Activity**.

Step 4 Select the deployment activity template from the **Process Template Selector** dialog. This list is prepopulated with all of the activity templates available on the current DROPS Server.

Step 5 Click **OK**.

Result The activity element is now dark blue which implies that the template is applied.

Step 6 Configure the Component to deploy and the Transfer options in the activities **Properties** view.

> ***Warning!***
> Infrastructure item activities *must* be linked to a component in order to save the deployment process. If you do not add a component to every activity and you close the deployment process diagram editor, all changes will be lost.

Save the changes (, Ctrl+S or **File** > **Save**).

28.6.1.3 Define the order in which to deploy activities

The hierarchical order in which the activities in the deployment process are carried out is defined by the black arrows that connect them. The flow of the deployment process follows the direction of the arrows.

By default, the **Initialization** activity is the first activity carried out in order and the **Finalization** the last. Between the two, infrastructure item activities can be carried out in a defined uni-lateral order or simultaneously by linking two activities to the same parent activity. If a new deployment process contains multiple infrastructure items, they are configured to be deployed simultaneously by default (they are both automatically children of the **Initialization** activity).

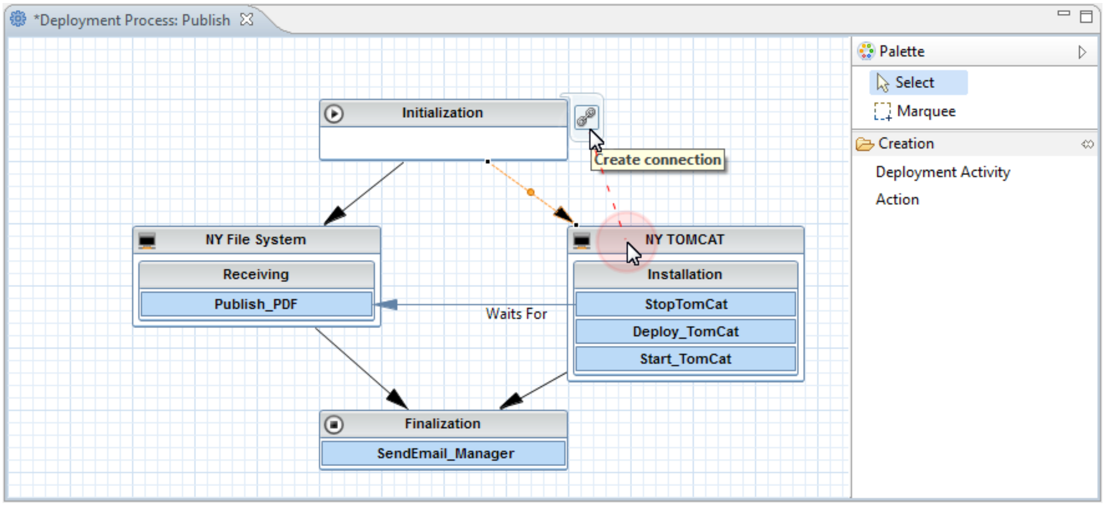


Figure 59: Create ordered connections between deployment activities

Follow the subsequent steps to create connections between deployment activities.

Step 1 Hover over the activity that should be carried out first during deployment.

Step 2 Click the Create connection icon and drag the cursor to the next activity, guided by the orange dotted line.

Step 3 When the connection is made the orange line changes to a black arrow that confirms the source activity will be carried out before the second activity. The second activity will not begin until the first is complete.

Save the changes (, Ctrl+S or **File** > **Save**).

Warning!

Infrastructure item activities *must* have both a parent and a child activity in order to save the deployment process. Most commonly, the parent will be an **Initialization** activity and the child a **Finalization** activity. If you do not define both parent and child activities for every infrastructure item activity and you close the deployment process diagram editor, all changes will be lost.

To delete a connection, select the arrow and press the Delete key.

28.6.1.4 Rearrange deployment activity boxes

The **Select** and **Marquee** tools enable you to select individual or several activities in the diagram to move them around the grid. Use these tools to visually reorder the boxes on the screen.

To select multiple activities, the whole activity box must be included in the square drawn with the **Marquee** tool to include it in the section.

> **Note**
> Moving the activities around the screen does not change the order that activities are carried out. Only the black connection arrows define the order.

28.6.1.5 Delete deployment activities

> **Note**
> **Initialization** and **Finalization** activities cannot be deleted.

> ***Warning!***
> Deleted deployment activities cannot be accessed or recovered. Any associated stages and any actions defined inside them are also deleted.
>
> Deleting an activity template from a deployment process diagram does not delete the template itself.

To delete an activity, hover over its title bar and click the Delete icon or right-click on the bar and select **Delete**. Click **OK** to confirm or click **Cancel** to keep the deployment activity.

28.6.2 Manage deployment process stages

Stages are linked to the mode in which a deployment is executed and are always carried out in a specific location. They are added to activities to regroup deployment actions in order to define where and when they are carried out. There are four types of stages:

1. The **Preparation Stage** executes actions to prepare or transform the necessary artifacts before transferring them to the target infrastructure item. This is the first stage carried out when the deployment process is executed. All actions are performed *on the DROPS Server*. This stage is not mandatory.
2. The **Transfer Stage** is *always performed behind the scenes and cannot be modified*. The artifacts are transferred to the infrastructure item using the DROPS Agent running on the remote service.

   The transfer stage is only activated (meaning files are transferred from the DROPS Server to the target DROPS Agent/infrastructure item) if either a Receiving or an Installation stage is defined in the activity. You can force the transfer of content without defining these stages by ticking the **Force Transfer** checkbox in the activity's properties.

   > **Note**
   > If a deployment schema is included in the infrastructure item, it is applied during this stage. If not, the artifacts are copied directly from the artifact repository to the **Root Directory** defined in the infrastructure item.

Reference

For more information about schemas, refer to Deployment schemas on page 132.

3. The **receiving stage** is intended to enable you to modify the deployed artifacts once they are available in the Root Directory on the target infrastructure item. The deployment process can be suspended after this stage in order to work with artifacts before installing them onto the target item. This stage is not mandatory.

   If you execute a deployment process instance in **transfer-only mode**, the process stops after this stage is complete.

4. The **Installation Stage** executes actions to deploy the artifacts to their final destination on the infrastructure item. This stage is not mandatory.
   If you execute a deployment process instance in **transfer-only mode**, once it's finished, manually execute the instance in **installation-only mode** which executes only the actions contained in the installation stage.

28.6.2.1 Add new stages to deployment activities

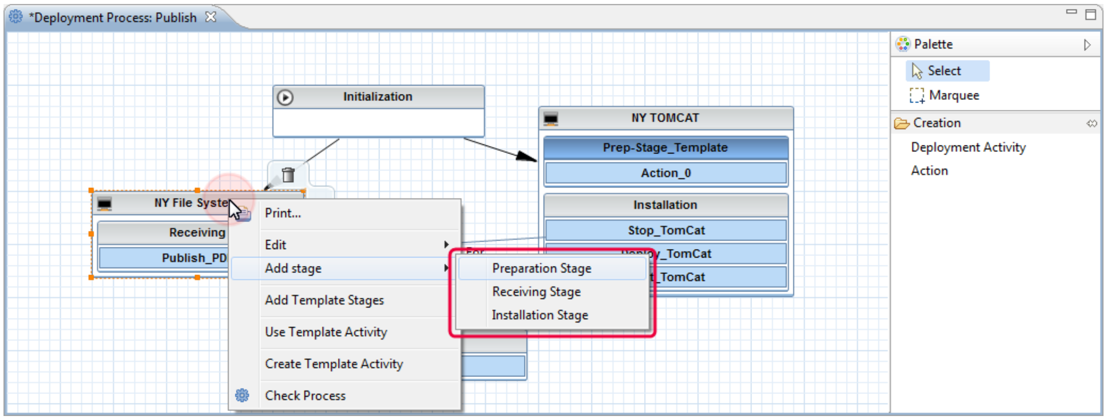


Figure 60: Add stages to activities

To add stages to an activity, right-click on it and hover over **Add stage**. Select the stage to add. The new stage is added to the activity and must be filled with at least one deployment action.

Save the changes (🖫, Ctrl+S or **File** > **Save**).

28.6.2.2 Add template stages to deployment activities

Pre-defined stage templates can be included in any activity in a deployment process diagram except a deployment activity template. They must include pre-defined actions and can only be modified in the template editor. When a template is used in a diagram, the activity or stage is dark blue, instead of the default gray. The difference in color makes it easy to see which parts of a diagram are inherited from a template and which are unique.

Follow the subsequent steps to add a stage template to a deployment activity.

Step 1 Right-click on the activity and select **Add template stages**.

Step 2 Select the stage template from the **Process Template Selector** dialog. This list is prepopulated with all of the stage templates available on the current DROPS Server.

Step 3 Click **OK**.

Result The stage element is now dark blue and the template is applied.

Save the changes (💾, Ctrl+S or **File** > **Save**).

28.6.2.3 Delete stages from deployment activities

> ***Warning!***
>
> Deleted stages cannot be accessed or recovered. Any associated actions are also deleted.
>
> Deleting a stage template from an activity does not delete the template itself.

To delete a stage, hover over its title bar and click the 🗑 Delete icon or right-click on the bar and select **Delete**. Click **OK** to confirm or click **Cancel** to keep the stage.

28.6.3 Manage deployment actions

Actions are carried out sequentially in their parent activity following the uneditable order of their parent stages.

28.6.3.1 Add deployment actions to activities and stages

Follow the subsequent steps to add deployment actions to an Initialization or Finalization activity or to a stage in an infrastructure item activity. Activities cannot be added to template activities/stages.

Drag and drop actions inside their parent containers to reorder them.

Step 1 Click and hold the **Action** node in the right-hand menu.

Step 2 Drag your cursor to the activity or stage.

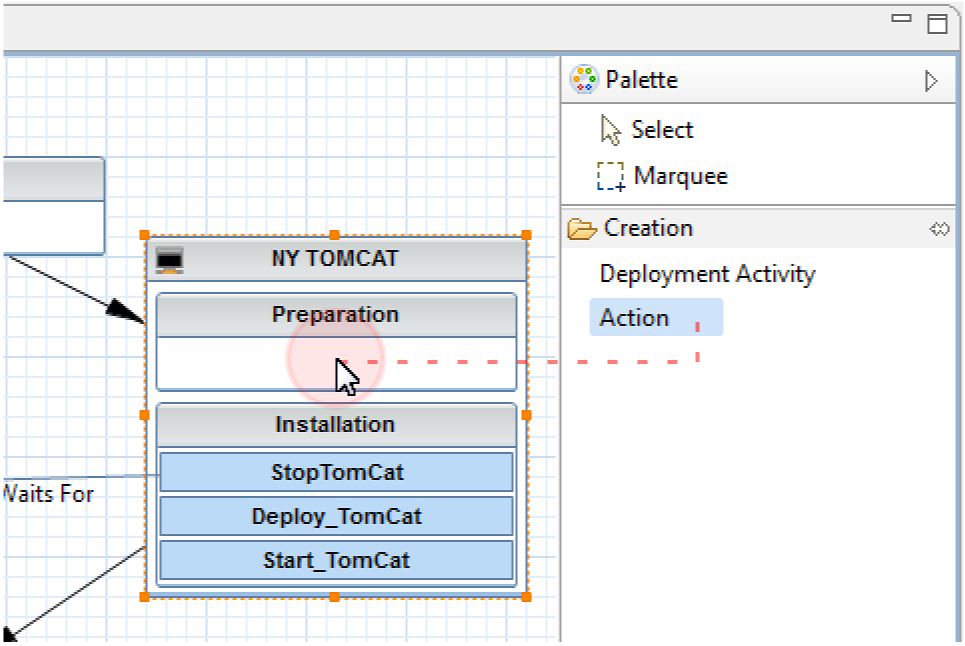


Figure 61: Drag and drop actions into activities or stages

Step 3 Releasing opens the **Script Selection** dialog where you select the action's script.

Use the **Name**, **Script Category** and **Execution Platform** fields to filter the list of scripts displayed on the right. Click the **Search** button to display all of the scripts available.

> **Important!**
> When you have selected a script, be sure to also select the **Version** of the selected script to ensure you have the correct file.

> ***Reference***
> For more information about all the built-in scripts DROPS Software offers, refer to the *DROPS Script Reference Guide*.

Click **OK**.

Result The new Action is automatically created in a blue box and is ready to be configured.

> ***Reference***
> For more information about editing an action's properties, refer to Edit deployment action properties on page 187.

Right-click on the blue Action box and select **Edit** > **Script** to modify the identification details of the script defined for the action.

Right-click on the blue Action box and select **Edit** > **Script Content** to modify the version of the ANT script used in the diagram (defined for the action).

> ***Reference***
> For more information about editing scripts, refer to Edit script details on page 153 and Edit script content on page 155.

Save the changes (, Ctrl+S or **File** > **Save**).

28.6.3.2 Delete deployment actions

> ***Warning!***
> Deleted deployment actions cannot be accessed or recovered.

To delete a deployment action, hover over the blue bar and click the Delete icon or right-click on the bar and select **Delete**. Click **OK** to confirm or click **Cancel** to keep the action.

28.6.4 Create "Wait For" Connections

Creating connections between deployment actions enables you to define which Action(s) should not begin until another Action in the process *defined for another activity* has completed. If an action is linked to another Action it will wait for that Action to finish before it begins.

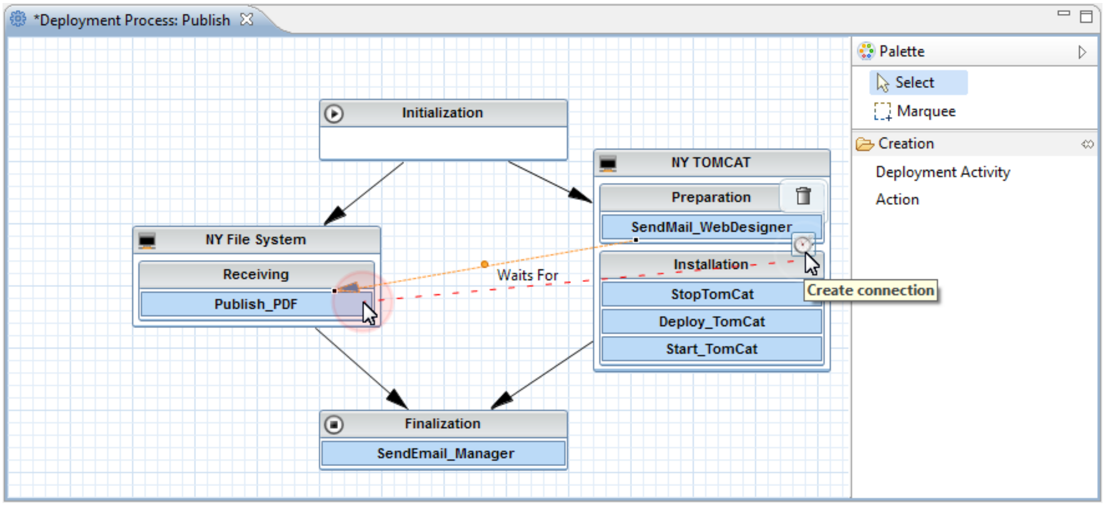


Figure 62: Create "wait for" connections between deployment actions

Follow the subsequent steps to create a "wait for" connection between deployment actions.

Step 1 Hover over the action that should wait.

Step 2 Click the Create "wait for" connection icon and drag the cursor to the deployment action that must complete, guided by the orange dotted line.

Step 3 When the connection is made the orange line changes to a blue arrow that confirms the source Action must "wait for" the second Action to complete before it can begin.

To delete a "wait for" connection, select it and press the Delete key.

Save the changes (, Ctrl+S or **File** > **Save**).

28.6.5 Print deployment process diagrams

Right-click on any activity, stage or Action in a deployment process diagram, or anywhere in the empty grid, then select **Print** to print a section of or the whole deployment process diagram.

28.7 Editing deployment process properties

There are three levels of properties to define for a deployment process:

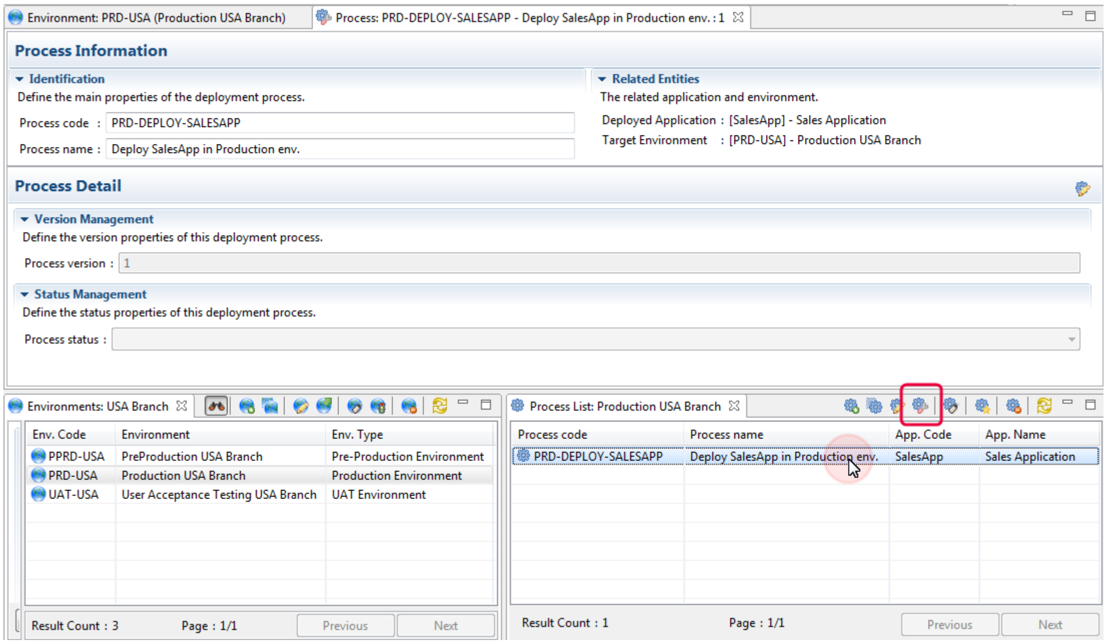
- The global-level, where properties that affect the whole deployment process are defined.
- The activity-level, where properties that affect only the Initialization and infrastructure item activities are defined.
- The action-level, where properties that affect each individual deployment action are defined.

Selecting the element for which to edit the properties in the deployment process diagram opens the context-sensitive **Properties** view.

> **Note**
>
> Click the pin icon in this view to remove the context-sensitive property. Clicking this icon ensures that the current entity's properties will stay in the view, even if you navigate away from that entity in the editor. If the pin in not activated, the options in the **Properties** view will change depending on what entity is currently selected in the editor.

28.7.1 Edit global deployment process details

To edit deployment process details, either right-click on the process in the **Process List** view then select **Edit Details** or select it, then click the Edit Details icon. The **Process** editor is opened.



Environment: PRD-USA (Production USA Branch) | Process: PRD-DEPLOY-SALESAPP - Deploy SalesApp in Production env. : 1

Process Information

Identification
Define the main properties of the deployment process.
Process code : PRD-DEPLOY-SALESAPP
Process name : Deploy SalesApp in Production env.

Related Entities
The related application and environment.
Deployed Application : [SalesApp] - Sales Application
Target Environment : [PRD-USA] - Production USA Branch

Process Detail

Version Management
Define the version properties of this deployment process.
Process version : 1

Status Management
Define the status properties of this deployment process.
Process status :

Environments: USA Branch

Env. Code	Environment	Env. Type
PPRD-USA	PreProduction USA Branch	Pre-Production Environment
PRD-USA	Production USA Branch	Production Environment
UAT-USA	User Acceptance Testing USA Branch	UAT Environment

Result Count : 3 Page : 1/1 Previous Next

Process List: Production USA Branch

Process code	Process name	App. Code	App. Name
PRD-DEPLOY-SALESAPP	Deploy SalesApp in Production env.	SalesApp	Sales Application

Result Count : 1 Page : 1/1 Previous Next

Figure 63: The deployment process details editor

Process Code

A unique string that identifies the deployment process. This label is used throughout DROPS to select the entity.

Process Name

A label used to give additional information about the deployment process. This label is used throughout DROPS to select the entity.

Related Entities

The process' Application and Environment fields are not editable. These values cannot be changed for an established deployment process. Create a new process to change the application deployed to the current Environment. To change the Environment that the current application is deployed to, duplicate the existing process.

Process Version

The current version of the deployment process.

Process Status

The current status of the deployment process. Status are defined in the **Preferences** menu. If a deployment process is marked as **Production Status**, it means that the process has been validated and cannot be modified.

> ***Reference***
> For more information about changing a process' status, refer to Changing a deployment process' status on page 164.

Save the changes (🖫, Ctrl+S or **File** > **Save**).

Warning!

If there are errors in the process diagram, DROPS cannot save it.

A dialog opens to warn you that there are invalid items in process diagram and the **Invalid Items** view displays the problem. If you close the process without fixing the issues listed and successfully saving, any changes will be lost!

To check if a diagram is valid at any time, right-click on any of the elements and select **Check Process**.

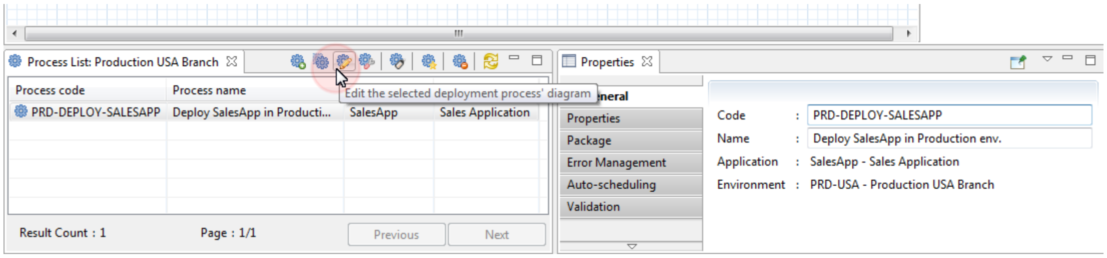
Reference

Refer to the Troubleshooting guide in the appendix for more details on the different reasons a diagram cannot be saved.

28.7.2 Edit global deployment process properties

To edit the global deployment process properties either right-click on the process in the **Process List** view then select **Edit**, select it then click the edit icon or double-click it.

If the **Properties** view is already opened, navigating to the deployment process diagram will open the global properties in that view.


Process List: Production USA Branch
Edit the selected deployment process' diagram
Properties
Previous
Next

Process code	Process name		
PRD-DEPLOY-SALESAPP	Deploy SalesApp in Producti...	SalesApp	Sales Application

Result Count : 1

Page : 1/1

Code : PRD-DEPLOY-SALESAPP

Name : Deploy SalesApp in Production env.

Application : SalesApp - Sales Application

Environment : PRD-USA - Production USA Branch

Figure 64: The deployment process properties editor

The tabs on the left enable you to modify the process' different global properties.

General

Process Code

A unique string that identifies deployment process instance. This label is used throughout DROPS to select the entity.

Process Name

A label used to give additional information about the deployment process.This label is used throughout DROPS to select the entity.

Application & Environment

The process' Application and Environment fields are not editable. These values cannot be changed for an established deployment process. Create a new process to change the application deployed to the current Environment. To change the Environment that the current application is deployed to, duplicate the existing process.

Properties

Refer to Properties and Server properties in the appendix.

Package

Select the **Package Type** for the associated application.

If this field is set to **From Application**, the packaging type defined in the application will be used.

[*default*] **Unitary** packaging deploys only the release selected.

Cumulative packaging deploys the selected release and all of the preceding releases that have been imported into DROPS but have not yet been deployed to the target environment. A ‘cumulative packaging type’ deploys only the most recent artifacts in each release. This ensures that all of the most up-to-date artifacts are deployed and that the deployed application is accurately updated.

> ***Example***
>
> You have imported 4 versions of your release but only deployed the first version to the target environment. The 4th release contains the same version of all the artifacts that the 3rd release contains, except for one file. Now, you select the 4th release to deploy and choose a ‘cumulative packaging type’. Result: The 2nd, 3rd and 4th releases will all be deployed in order, finishing with the most current version of every artifact and release. The artifacts from the 3rd release are all current except the one, which is deployed with the 4th release.

Error Management

If the **Execute Rollback on error** checkbox is ticked, when the process encounters an error, it will automatically roll back *everything* it accomplished up until the moment the error occurred.

The process executes all of the rollback actions for every activity in the deployment process diagram in reverse order.

If this checkbox is not ticked, the process will complete even if it encounters errors. Likely, its final status will be *Failed*. Rollback can then be carried out manually.

Auto-scheduling

If deployment instances are created using environment groups, they can be automatically broken into transfer-only and installation-only pairs. Installation-only instances can be automatically scheduled to execute a certain amount of time after the transfer-only portion is complete. This property defines when automatically-scheduled installation-only deployment process instances are executed.

Delay Value

The delay, in number of days, between the end of a transfer-only execution and when its child installation-only instance will be scheduled to begin automatically.

If this property is set to **-1**, DROPS will automatically retrieve the default value defined in the Server Configuration.

By default the delay is one (**1**) day.

> ***Reference***
>
> For more information about defining the default, refer to the *DROPS Configuration Guide*.

Execute at a fixed time

If ticked, the scheduled time of day is defined by the user. If this option is activated the two options below are available and must be defined. If this option is left inactive, the automatic execution will begin right after the parent instance is finished.

Hour

The hour at which the execution will take place on the day defined by the date the parent instance execution was completed + the delay value.

Minute

The minute of the hour defined at which the execution will take place on the day defined by the date the parent instance execution was completed + the delay value.

Default scheduling

Enable default scheduling for a deployment process, to define a **Default scheduling** which automatically schedules a deployment instance relatively to its creation date (for single deployment instance) or the stage validation date (for deployments that are part of a pipeline). The default schedule can be forced so that it cannot be modified later with the execution console.

> **Note**
>
> The default scheduling is different from the **Auto-scheduling**, which only schedules an installation instance relatively to a transfer deployment instance.

Default scheduling

The date at which the execution will take place. It can be today, next-*D*, next-*X*-*D* with *X* a number and *D* one of the following: monday, tuesday, wednesday, thursday, friday, saturday, sunday, day, week, month.

Hour

The hour at which the execution will take place on the day defined by the date defined in the **Default scheduling** field.

Minute

The minute of the hour defined at which the execution will take place on the date defined in the **Default scheduling** field.

Skip non-working days

When using next-*X-D*, you can check the **Skip non-working days** box so that Saturday and Sunday are not counted as days. So if today is Friday, next-3-day is next Wednesday. It is not used for other scheduling types.

Enforce scheduling

Enforce the schedule rule by checking the **Enforce scheduling** box. In that case, it will not be possible to modify this scheduling with the execution console.

✓ Validation

Validation Required Before Starting

To require validation from a specific user before being able to execute this part of the deployment, tick the checkbox. During deployment, the element is executed as soon as the validator approves the request.

Validation requests display in the ✓ **Validation Requests** node (under the relavent parent node in the ⚙ **Operational Console Management** perspective) for the user(s) assigned the role(s) selected. They also receive an automatic email.

> ***Reference***
> For more information about approving requests, refer to Validation requests on page 296.

For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide.*

Validation Request Message & Comment

The subject line and body of the email to send to the validator. If defined here, the default content of the email, defined in the Preferences menu, is overridden.

Validation comment is required

Tick this checkbox if the user is required to provide a reason why they validate or reject the process.

Validation Roles

Select the role of the user(s) that should receive the validation request. The list of roles is pre-populated with all the Roles that can validate the entity.

✉ Notification

Click the ⊕ **Create** button to create a new notification linked to the deployment process.

Notifications messages are not like validation/suspension/interruption messages in that they are intended to be only informative. They do not require action from the user and simply let them know that certain steps have been carried out in the process. Notifications are sent to the email address configured for each user who is assigned the role(s) selected.

When & Subject

On Start sends the notification email to the user(s) at execution.

On End sends the notification email to the user(s) after execution.

Enter the subject line for the email in the **Subject** field.

Message

The body of the email sent to the user(s).

Custom recipients

Enter a CSV list of email addresses, without spaces, and those addresses will be notified when the process is executed.

Recipient Roles

Select the role of the user(s) that should receive the notification. The list of Roles is prepopulated with the roles that are allowed to validate.

Warning!

If there are errors in the process diagram, DROPS cannot save it.

A dialog opens to warn you that there are invalid items in process diagram and the **Invalid Items** view displays the problem. If you close the process without fixing the issues listed and successfully saving, any changes will be lost!

To check if a diagram is valid at any time, right-click on any of the elements and select **Check Process**.

Reference

Refer to the Troubleshooting guide in the appendix for more details on the different reasons a diagram cannot be saved.

28.7.2.1 Use a properties template

Pre-defined deployment process properties templates can be applied to any deployment process diagram. These templates define the diagram’s global properties. Applying the properties defined in a template at the global level is not permanent. After global-level properties are applied using a deployment process properties template, you can edit the diagram’s properties to continue to personalize the options.

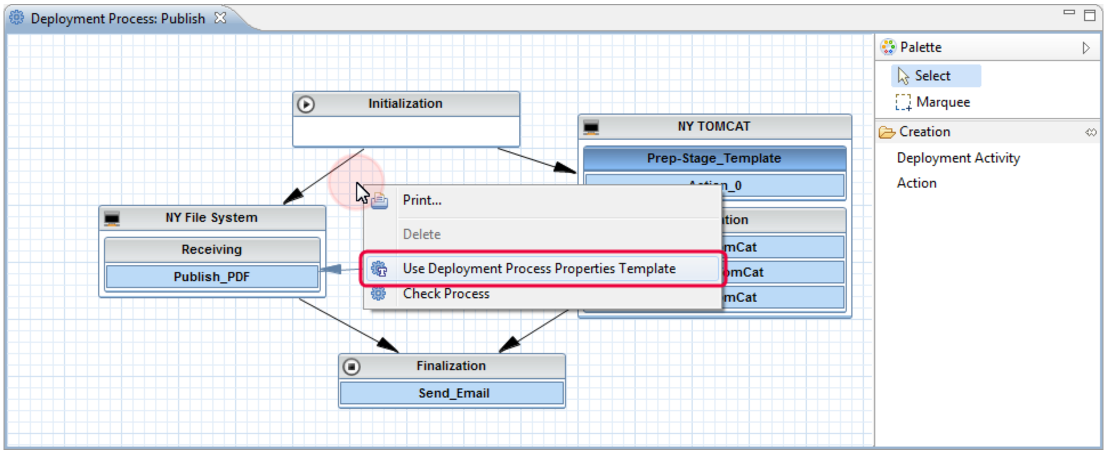


Figure 65: Global-level properties templates in the deployment process diagram

Follow the subsequent steps to apply the predefined global properties to a deployment process.

Step 1 Right-click anywhere in the editor and select **Use Deployment Process Properties Template**.

Step 2 Select the deployment process properties template from the **Process Template Selector** dialog. This list is prepopulated with all of the global process properties templates available on the current DROPS Server.

Step 3 Click **OK**.

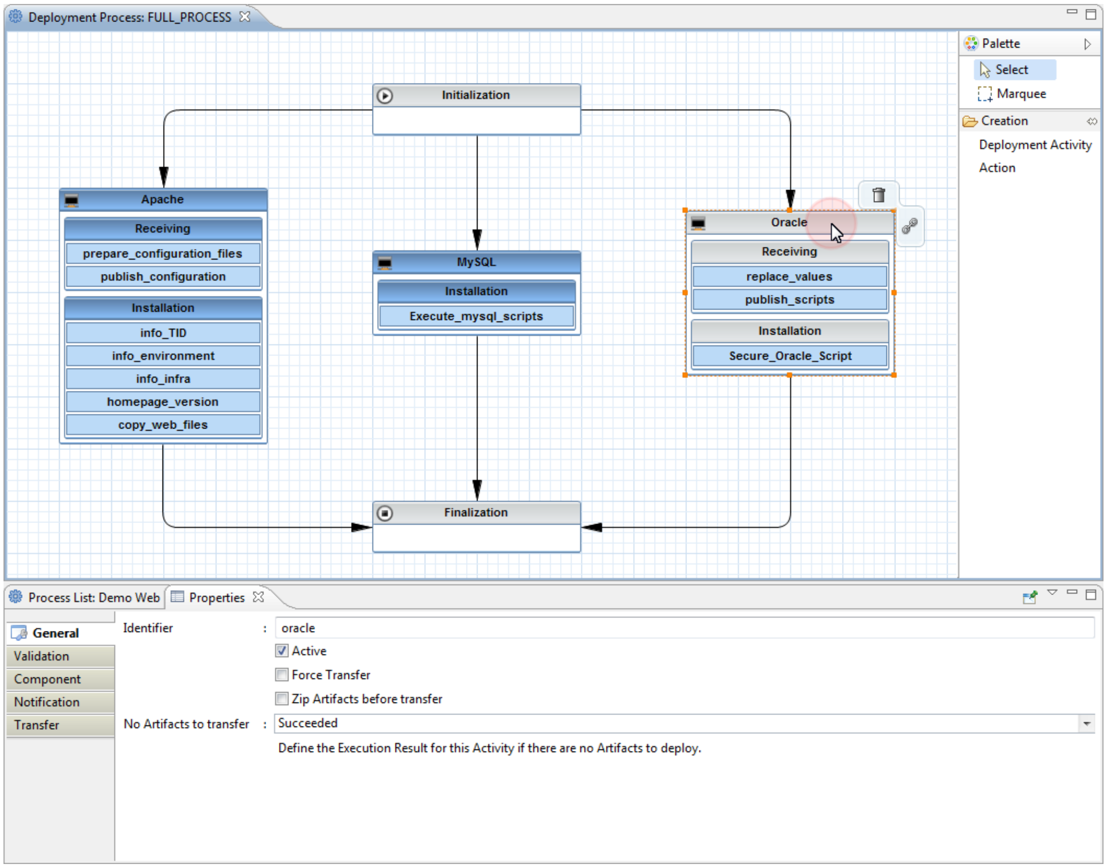
Result The properties defined in the template are applied to the current process but can be modified.

Save the changes (, Ctrl+S or **File** > **Save**).

28.7.3 Edit deployment activity properties

To edit the activity-specific properties in the deployment process diagram, click on the **Initialization** or *<infrastructure item>* activity.

If the **Properties** view is already opened, clicking on any activity will open its properties in that view.



Identifier : oracle
☑ Active
□ Force Transfer
□ Zip Artifacts before transfer
No Artifacts to transfer : Succeeded
Define the Execution Result for this Activity if there are no Artifacts to deploy.

Figure 66: The deployment activity properties editor

Click the ▶ **Initialization** activity to open its **Properties** view. The tabs on the left enable you to modify the activity's properties.

✓ Validation

Validation Required Before Starting

To require validation from a specific user before being able to execute this part of the deployment, tick the checkbox. During deployment, the element is executed as soon as the validator approves the request.

Validation requests display in the ✓ **Validation Requests** node (under the relavent parent node in the ⚙ **Operational Console Management** perspective) for the user(s) assigned the role(s) selected. They also receive an automatic email.

> ***Reference***
> For more information about approving requests, refer to Validation requests on page 296.

For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide*.

Validation Request Message & Comment

The subject line and body of the email to send to the validator. If defined here, the default content of the email, defined in the Preferences menu, is overridden.

Validation comment is required

Tick this checkbox if the user is required to provide a reason why they validate or reject the process.

Validation Roles

Select the role of the user(s) that should receive the validation request. The list of roles is pre-populated with all the Roles that can validate the entity.

Import

Execute Import

If ticked, the import process(es) associated with the releases to deploy are launched. It is recommended to only re-import the release being deployed if you are sure of its contents. This option is helpful when working in a Continuous Integration system.

Fail on Error

If ticked, if the operation fails for any reason at this step, the import will fail and no further actions will be carried out.

Click an infrastructure item activity to open its **Properties** view. The tabs on the left enable you to modify the activity's properties.

General

*Identifier

[*Required*] Enter a unique label for the activity. Only alphanumeric characters are allowed (no spaces).

Active

Tick this checkbox to ensure that inactive infrastructure item activities do not execute when the deployment process is launched.

Force Transfer

If the activity does not contain either a receiving or an installation stage, or both, the transfer stage will not be executed, meaning that the content will not be transferred from the DROPS Server to the target infrastructure item.

Tick this checkbox to force the content to transfer *without* defining receiving or installation stages. If ticked, content will transfer when the activity is executed.

Ticking this checkbox also ensures that content is transferred even if there are no deployment actions defined.

Zip artifacts before transfer

Tick this checkbox to automatically zip all of the artifacts before transferring them and them automatically unzip them after the transfer. This is helpful if you have a lot of small artifacts to transfer because the compression will save space and time.

No artifacts to transfer

In some cases, an activity may never need to transfer any artifacts. In other cases, if an activity fails to detect artifacts to transfer or fails to transfer them, it may cause problems. This option allows you to decide what status to give an activity should no artifacts be transferred by it.

The status can be defined for every activity in each process to manage the **Execution Result** in the deployment log. If it's normal that one activity should not transfer any artifact, set the status to **Succeeded**. If this is abnormal, choose between **Warning** and **Failed**.

✓ Validation

Validation Required Before Starting

To require validation from a specific user before being able to execute this part of the deployment, tick the checkbox. During deployment, the element is executed as soon as the validator approves the request.

Validation requests display in the ✓ **Validation Requests** node (under the relavent parent node in the **Operational Console Management** perspective) for the user(s) assigned the role(s) selected. They also receive an automatic email.

Reference

For more information about approving requests, refer to Validation requests on page 296.

For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide*.

Note

If an activity requires validation but has not been validated before the process is executed, it will not run. Approved activities in the process will run but any unapproved activities will not.

Validation Request Message & Comment

The subject line and body of the email to send to the validator. If defined here, the default content of the email, defined in the Preferences menu, is overridden.

Validation comment is required

Tick this checkbox if the user is required to provide a reason why they validate or reject the process.

Validation Roles

Select the role of the user(s) that should receive the validation request. The list of roles is prepopulated with all the Roles that can validate the entity.

Component

Every activity is required to be linked to a ***Component to deploy**. Click the browse button to select the component to deploy to the corresponding infrastructure item. The **Component Selection** dialog is prepopulated with all of the components defined in the application associated with the current deployment process.

Only one component can be deployed to an infrastructure item per deployment process. To deploy multiple components to one Item, you must create multiple deployment processes.

> **Important!**
> Make sure to select the correct component! Once confirmed, you cannot change it.

To edit the properties for the component defined in an infrastructure item, right-click on its activity and select **Edit** > **Component**.

Notification

Click the **Create** button to create a new notification linked to the activity.

Notifications messages are not like validation/suspension/interruption messages in that they are intended to be only informative. They do not require action from the user and simply let them know that certain steps have been carried out in the process. Notifications are sent to the email address configured for each user who is assigned the role(s) selected.

When & Subject

On Start sends the notification email to the user(s) at execution.

On End sends the notification email to the user(s) after execution.

Enter the subject line for the email in the **Subject** field.

Message

The body of the email sent to the user(s).

Custom recipients

Enter a CSV list of email addresses, without spaces, and those addresses will be notified when the process is executed.

Recipient Roles

Select the role of the user(s) that should receive the notification. The list of Roles is prepopulated with the roles that are allowed to validate.

Transfer

Deployment Schema

Select the deployment schema to run on the artifacts deployed to the corresponding infrastructure item. Click the browse button. **The Deployment Schema** dialog is prepopulated with all of the deployment schemas defined in the infrastructure item.

Only one schema can be used for an infrastructure item per deployment process.

Reference

For more information about adding schemas to infrastructure items, refer to Editing infrastructure items on page 115.

Use Replacement Rules

If ticked, the substitutions defined by the replacement rules are activated at the beginning of the transfer stage for all of the artifacts deployed to the current infrastructure item. This ensures that the replacement rules are taken into account locally before the artifacts reach their final destination.

Important!

This box *must* be ticked in order to run the replacement rules defined in DROPS.

Replacement rules are similar to a *Search & Replace* on configuration files linked to an application. Rules enable you to activate specific commented content in artifacts such as .properties files during deployment. This means that commented content can be selected and rendered active by automatically removing the comment tag. The now-active content will be taken into account during the deployment process.

Reference

For more information about replacement rules, refer to Replacement rules on page 136.

Warning!

If there are errors in the process diagram, DROPS cannot save it.

A dialog opens to warn you that there are invalid items in process diagram and the **Invalid Items** view displays the problem. If you close the process without fixing the issues listed and successfully saving, any changes will be lost!

To check if a diagram is valid at any time, right-click on any of the elements and select **Check Process**.

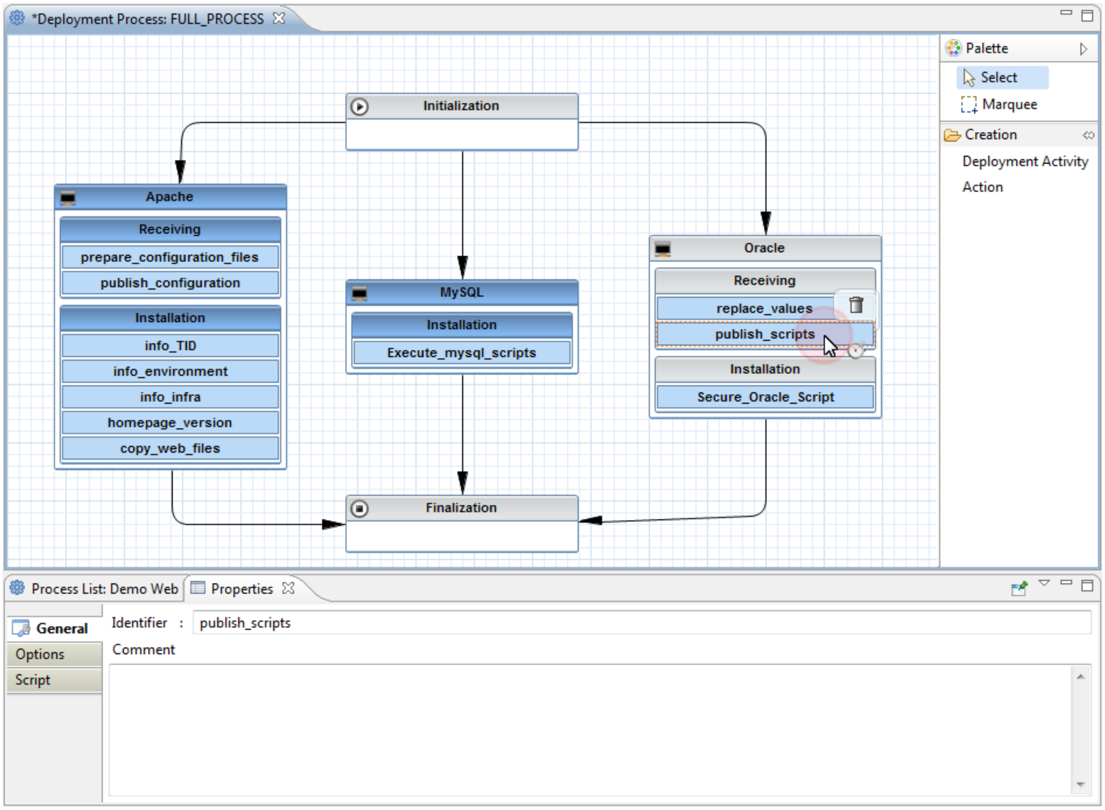
Reference

Refer to the Troubleshooting guide in the appendix for more details on the different reasons a diagram cannot be saved.

28.7.4 Edit deployment action properties

To edit the action-specific properties in the deployment process diagram, click the blue Action box.

If the **Properties** view is already opened, clicking on any Action will open its properties in that view.



Identifier : publish_scripts

Comment

Figure 67: The deployment action properties editor

General

*Identifier

[*Required*] Enter a unique label for the deployment action. Only alphanumeric characters are allowed (no spaces). This label is viewed in the deployment process diagram to easily see what the action accomplishes.

Comment

Enter additional information in this field to clarify the specific details of this Action.

Options

Apply script to the list of artifacts

If ticked, this option ensures that the script will be applied to every artifact deployed.

If not ticked, the script will run just one time.

> ***Example***
> An email script should likely not be applied to all of the artifacts, but 'copy to' scripts specific to a certain file type should be applied to every artifact to ensure that all the artifacts are filtered through the

script.

Important!

If any of the action-level substitution variables are used, you *must* tick this checkbox in order to ensure that the variable is taken into account at the artifact-level. If the script is not applied to each individual artifact in the list, these variables will not be applied because they are specific to individual artifacts.

Reference

For more information about these variables, refer to Substitution variables in the appendix.

File Name Pattern

Enter the pattern to follow to filter the artifacts. Artifacts that meet the pattern defined here will all be used for the current script.

Fails on Error

[*Ticked by default*] If ticked, if the operation fails for any reason at this step, the deployment will fail and no further actions will be carried out.

Script

Script

The name and version number of the action's current script. The associated script is selected when the action is created but it can also be changed here.

Click the … browse button to select a new script to associate with the action.

Filter the search in the Script Selection dialog using the **Name**, **Script Category** and/or **Execution Platform** options. To display all of the scripts available on the current server, click **Search** without defining any search criteria.

Select the script, then select the version of that script to use. Click **OK** to apply the selected script to the deployment action.

Warning!

Replacing a script in an action deletes any parameters that may have been defined in the original script. All changes are lost.

Reference

For more information about all the built-in scripts DROPS Software offers, refer to the *DROPSScript Reference Guide*.

Parameters

If required, modify the parameters for the selected script in the **Value** column. The parameters in this list are prepopulated with all of the parameters defined in the **Script** editor.

Variables can be used as parameter values.

Press Ctrl+Space to view a full list of the variables available for the current field. The list of available variables is prepopulated with the available properties defined in the entities or at server level, and the ready-to-use context-specific variables available by default.

Reference

For more information about variables, refer to Substitution variables in the appendix.

If a parameter is not given a value, it will not be referenced when the script is run during the deployment process.

Warning!

If there are errors in the process diagram, DROPS cannot save it.

A dialog opens to warn you that there are invalid items in process diagram and the **Invalid Items** view displays the problem. If you close the process without fixing the issues listed and successfully saving, any changes will be lost!

To check if a diagram is valid at any time, right-click on any of the elements and select **Check Process**.

Reference

Refer to the Troubleshooting guide in the appendix for more details on the different reasons a diagram cannot be saved.

29 Deployment plans

Required Roles	n/a
Required Rights (by category)	Clients Deployment Plans
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) Deployment Plans

Chapter Summary

29.1 Searching for deployment plans 192
29.2 Creating deployment plans 192
29.3 Duplicating deployment plans 193
29.4 Deleting deployment plans 194
29.5 Building deployment plan diagrams 194
29.6 Editing deployment plan properties 199

While a deployment process delivers one application to one environment, a deployment plan automates the deployment of multiple applications to multiple environments.

Deployment plans describe how, to which environments, and in which order releases of given applications are to be deployed to satisfy a defined need. They bring multiple deployment processes into the same schedule.

Example

Before being deployed to a production environment, an application may need to be deployed to a testing environment first to complete QA, then perhaps to a training environment to ensure that the company's employees know the product before the end-users have access. Deployment plans automate the deployment of the application to each of the three environments and enable you to schedule everything.

Deployment plan diagrams enable you to create ordered plans that include multiple deployment processes.

The target environments are grouped by their type in the diagram creating delivery stages and making it easy to see the different phases of the application's movement through the processes. Deployment plans ensure complete deployment of every application onto every environment in the order defined.

Important!

Prerequisite for deployment plans: A deployment process for every application/environment pair in the plan *must* already be created.

Reference

For more information about creating deployment processes, refer to Deployment processes on page 160.

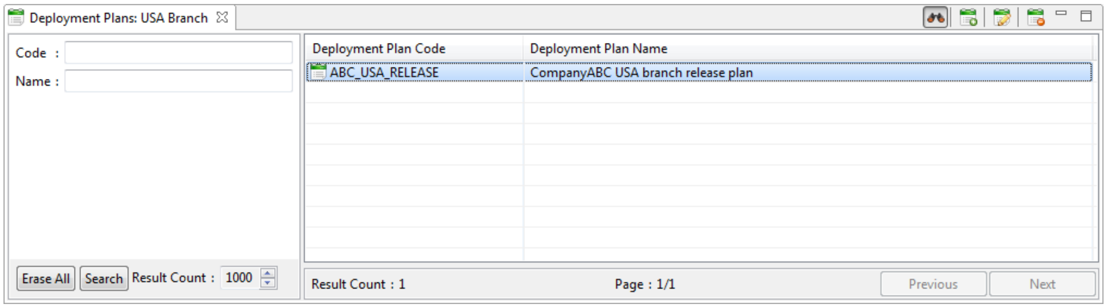
29.1 Searching for deployment plans

Follow the subsequent steps to search for deployment plans.

Step 1 Select **Deployment Plans** from the **Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

> ***Reference***
> For more information about managing multiple clients, refer to Multiple clients on page 94.



Deployment Plans: USA Branch

Deployment Plan Code	Deployment Plan Name
ABC_USA_RELEASE	CompanyABC USA branch release plan

Code :
Name :
Erase All | Search | Result Count : 1000
Result Count : 1 | Page : 1/1 | Previous | Next

Figure 68: The Deployment Plans search view

Step 2 When the search view is opened, the list of plans may be empty. Use the search criteria sections to filter the search.

- To search for an existing plan by its **Code** or **Name**, enter all or part of the information in the corresponding field.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.

Result The **Deployment Plans** search view displays all of the plans defined for the current server (and for the selected client, if applicable) that meet the search criteria.

29.2 Creating deployment plans

Follow the subsequent steps to create a new deployment plan.

Step 1 Click the create icon in the **Deployment Plans** search view or right-click anywhere in the view and select **Create a new deployment plan** to open the **Create a new deployment plan** wizard.

Step 2 Define the plan's **Code** and **Name**. These values can be edited later.

Step 3 Select the application(s) to deploy. The list is prepopulated with all of the applications available for the current client.

> **Tip**
> Ctrl+click enables you to select multiple applications from the list.

Step 4 Select the environment type(s) to which to deploy the content of the selected application(s).

Selecting a type here creates a container in the plan's diagram into which you add the appropriate environment(s).

> **Tip**
> Ctrl+click enables you to select multiple types from the list.

> **Note**
> Applications and environment types chosen in this wizard can be changed later and additional entities can be added even if they are not selected here.

Step 5 Click **Finish**.

Result New deployment plans are displayed in the **Deployment Plans** search view.

> ***Reference***
> For more information about building and configuring deployment plan diagrams, refer to Building deployment plan diagrams on the next page and Editing deployment plan properties on page 199.

29.3 Duplicating deployment plans

Follow the subsequent steps to create a new deployment plan.

Step 1 Click the duplicate icon in the **Deployment Plans** search view or right-click anywhere in the view and select **Duplicate** to open the **Duplicate a deployment plan** wizard.

Step 2 Define the plan's **Code** and **Name**. These values can be edited later.

Step 3 Select the part of the deployment plan number to increment in the **Increment Part** drop-down list. You can increment the *MAJOR*, *MINOR* or *BUILD* part of the deployment plan number.

Step 4 Click **Finish**.

Result The duplicated deployment plans are displayed in the **Deployment Plans** search view.

> ***Reference***
> For more information about building and configuring deployment plan diagrams, refer to Building deployment plan diagrams on the next page and Editing deployment plan properties on page 199.

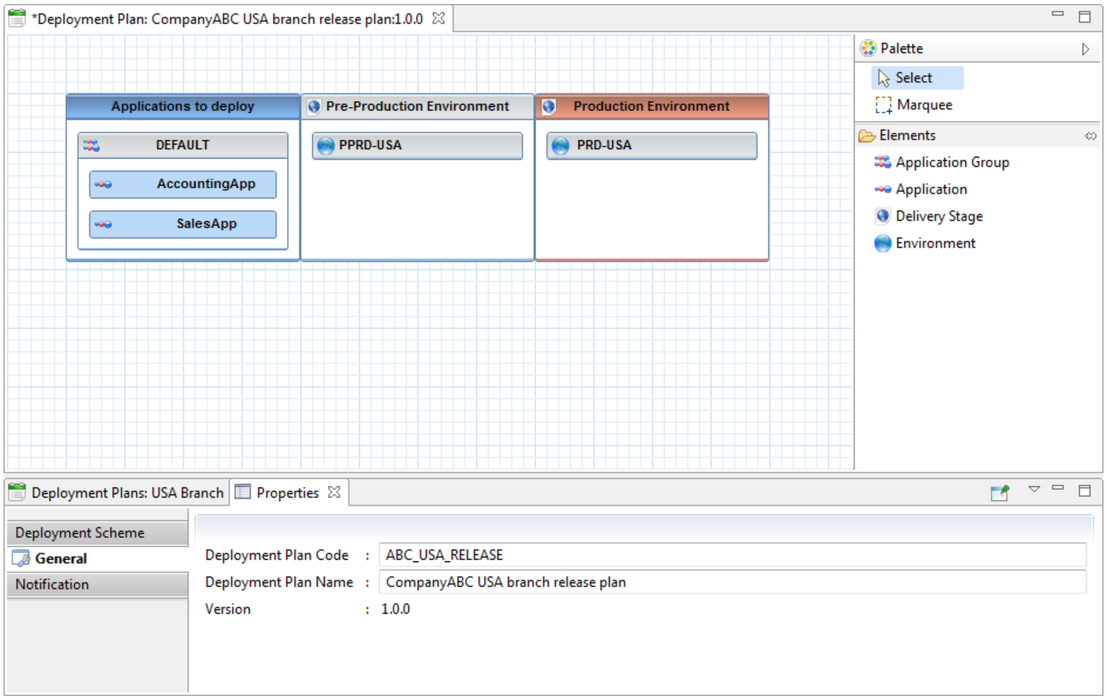
29.4 Deleting deployment plans

> ***Warning!***
> Deleted deployment plans cannot be accessed or recovered. The corresponding configuration is lost as well.

To delete a deployment plan, search for it in the **Deployment Plans** search view. Either right-click on it, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the plan.

29.5 Building deployment plan diagrams

The deployment plan diagram is a visual representation of the deployment process for multiple environments. Like deployment process diagrams, deployment plan diagrams are represented in grids and built by dragging and dropping the elements to be included in the plan into their appropriate containers.



Deployment Plan Code : ABC_USA_RELEASE
Deployment Plan Name : CompanyABC USA branch release plan
Version : 1.0.0

Figure 69: The Deployment Plan editor

> **Important!**
> The order of the applications and environments in their respective containers, from top to bottom, left to right, matters. The deployment plan will deploy the first application to the first environment completely before deploying the second application in the list. If there are multiple environments in one delivery stage, the plan will ensure complete sets of all of the applications are deployed to every environment in the stage, in top-to-bottom order before moving to the next stage and starting at the top of that list.

⚠ Drag and drop applications and environments inside the parent containers to reorder them.

The chronological order of the environment type stages is important, also. Applications are deployed to the environments from left to right. The order of the containers in the deployment plan diagram corresponds to the workflow number assigned to the environment type during configuration.

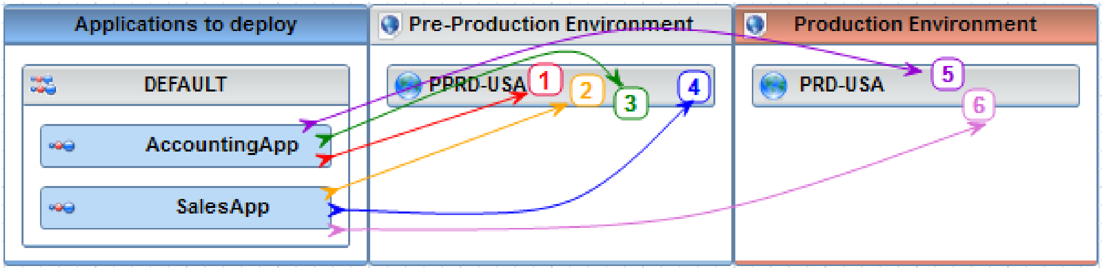


Figure 70: Deployment Plan: Order of deployment

29.5.1 Manage the application(s) to deliver

The blue **Applications to deploy** container is created automatically in a new deployment plan diagram. It is intended to organize the applications to deploy to all of the environments in the plan.

The applications selected in the **Create a new deployment plan** wizard are automatically displayed in this container, within a default application group.

29.5.1.1 Add application groups

Follow the subsequent steps to add application groups to the **Applications to deploy** container.

Step 1 Click and hold the **Application group** element in the right-hand menu.

Step 2 Drag your cursor and release the new group in the **Applications to deploy** container.

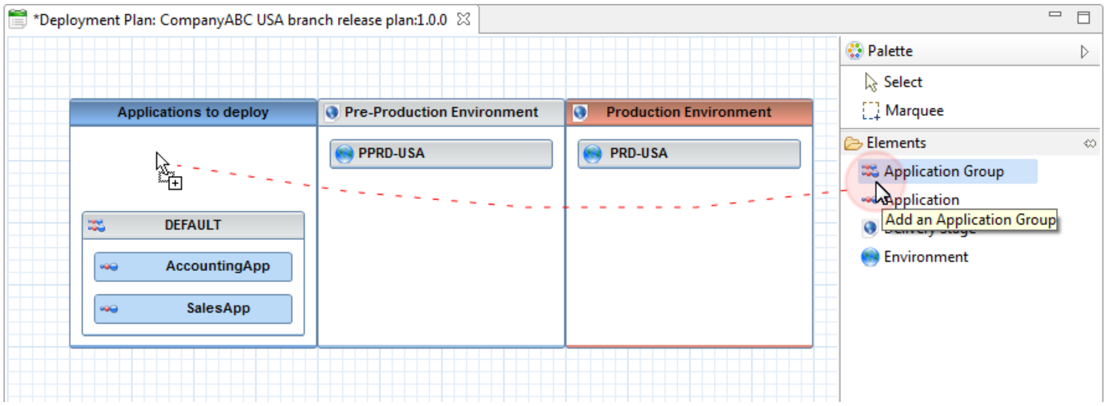


Figure 71: Create new groups of applications to deploy

Step 3 Enter the name of the new application group in the dialog.

Result The new application group is added to the container as a gray container.

Follow the Add applications to groups to deploy procedure to add applications to a group.

> **Note**
> Application cannot be dragged and dropped from one group to another. It must be deleted and then added again to a different group.

Save the changes (, Ctrl+S or **File** > **Save**).

29.5.1.2 Add applications to groups to deploy

Follow the subsequent steps to add applications to an application group in the **Applications to deploy** container.

Step 1 Click and hold the **Application to deliver** node in the right-hand menu.

Step 2 Drag your cursor to the **Applications Group** container.

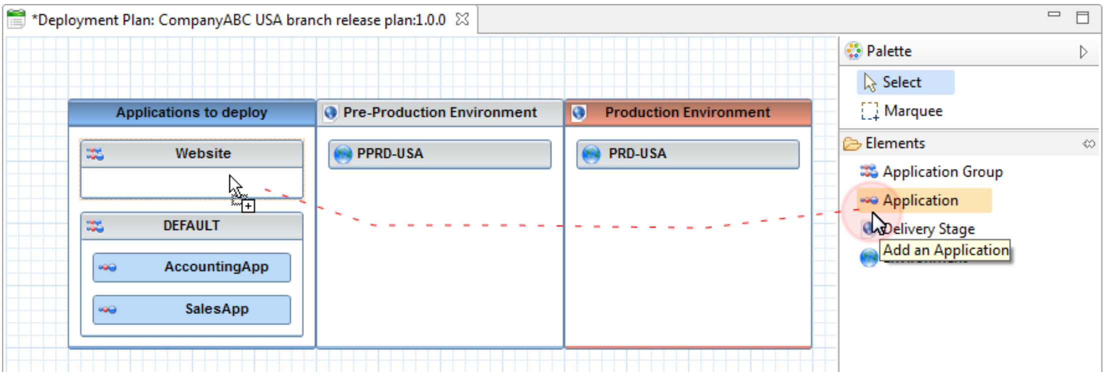


Figure 72: Drag and drop applications into the application group container

Step 3 Releasing opens the **Select an application** dialog where you select the application to deploy. The list is prepopulated with all of the applications defined for the current client.

> **Important!**
> Be careful selecting the applications to use in deployment plans. Each application selected must be allowed for each of the environments in the plan and a deployment process that includes both of these entities must already exist.

Result The new application is added to the container as a blue box.

Drag and drop applications inside the parent container to reorder them.

Right-click on the blue application box and select **Edit** > **Edit Application** to view and modify the application.

Reference
For more information about editing applications, refer to Applications on page 60.

Save the changes (, Ctrl+S or **File** > **Save**).

29.5.1.3 Delete applications from groups to deploy

To delete an application from a group in the applications to deploy container, hover over the blue application bar and click the delete icon or right-click on the bar and select **Delete**. Click **OK** to confirm or click **Cancel** to keep the application.

Note
If deleted, applications can always be added back to the **Applications to deploy** container.

29.5.2 Manage delivery stages

Delivery stages are environment type-specific containers that follow a specific order to define when the applications should be deployed to the different environments in the plan.

The environment types selected in the **Create a new deployment plan** wizard are created automatically. Existing stages can be edited and new stages can be added manually in the diagram.

29.5.2.1 Add delivery stages

Follow the subsequent steps to add additional delivery stages to the deployment plan diagram.

Step 1 Click and hold the **Delivery Stage** node in the right-hand menu.

Step 2 Drag and drop it anywhere into the diagram's grid.

Step 3 Select the environment type that the stage should contain from the **Environment Type** dialog that is displayed when the stage is dropped. The list is prepopulated with all of the types defined.

Important!
Be careful selecting the stage's type because it cannot be changed! Only environments that match the stage's type can be added to it.

Result New delivery stages are automatically created in the grid, always in the chronological order that corresponds to the workflow order defined in the **Preferences** menu. Production environment type stages are red to outline their importance.

Reference
For more information about the predefined workflow, refer to Entity types in the appendix.

Save the changes (, Ctrl+S or **File** > **Save**).

29.5.2.2 Delete delivery stages

> **Warning!**
> If deleted, any environments defined in the delivery stage are also removed from the deployment plan.

To delete a delivery stage, hover over it and click the 🗑 delete icon or right-click on the bar and select **Delete**. Click **OK** to confirm or click **Cancel** to keep the delivery stage.

29.5.3 Manage environment(s)

Environments are grouped by type in delivery stages. Only environments that are the same type as a stage can be added to the stage.

29.5.3.1 Add environments to delivery stages

Follow the subsequent steps to add environments to a delivery stage.

Step 1 Click and hold the **Environment** node in the right-hand menu.

Step 2 Drag your cursor to the delivery stage in which to add an environment.

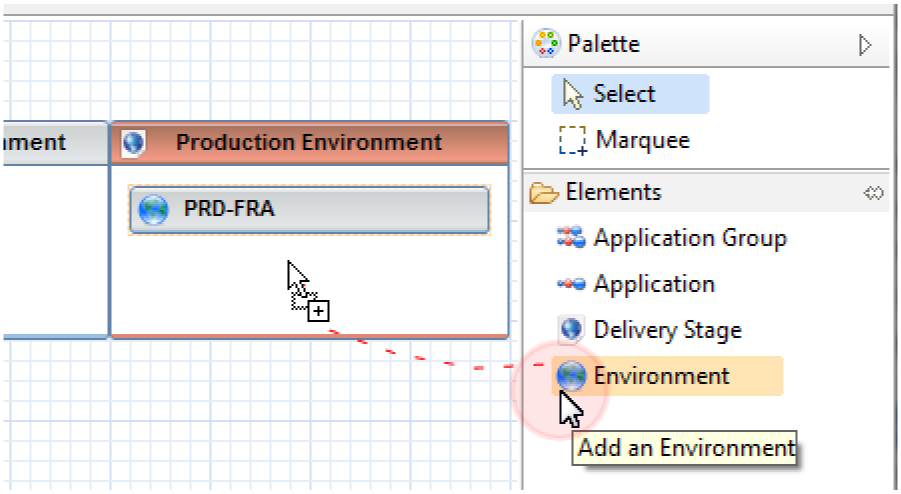


Figure 73: Drag and drop environments into delivery stages

Step 3 Releasing opens the **Environment Selection** dialog. The list is prepopulated with all of the environments defined for the current server/client that are the same type as the delivery stage.

> **Important!**
> Be careful selecting the environments to use in deployment plans. Each environments selected must allow all of the applications in the plan and a deployment process that includes both of these entities must exist.

Result The new environment is added to the delivery stage.

Drag and drop environments inside the parent containers to reorder them.

Right-click on the environment box and select **Edit** > **Edit Environment** to view and modify the environment.

> ***Reference***
> For more information about editing environments, refer to Environments on page 96.

Save the changes (, Ctrl+S or **File** > **Save**).

29.5.3.2 Delete environments from delivery stages

> **Note**
> If deleted, environments can be added back without any affect to their configuration.

To delete an environment from a delivery stage, hover over it and click the Delete icon or right-click on the bar and select **Delete**. Click **OK** to confirm or click **Cancel** to keep the environment.

29.5.4 Print deployment plans

Right-click on any container or entity in a deployment plan, or anywhere in the empty grid, then select **Print** to print a section of, or the whole deployment plan.

29.6 Editing deployment plan properties

There are multiple levels of properties to define for deployment plans:

- The global level, where properties that affect the whole plan are defined.
- The delivery stage level, where properties that affect only the different delivery stages are defined.
- The environment level, where properties that affect each individual environment are defined.
- The application group level, where properties that affect each individual application group are defined.

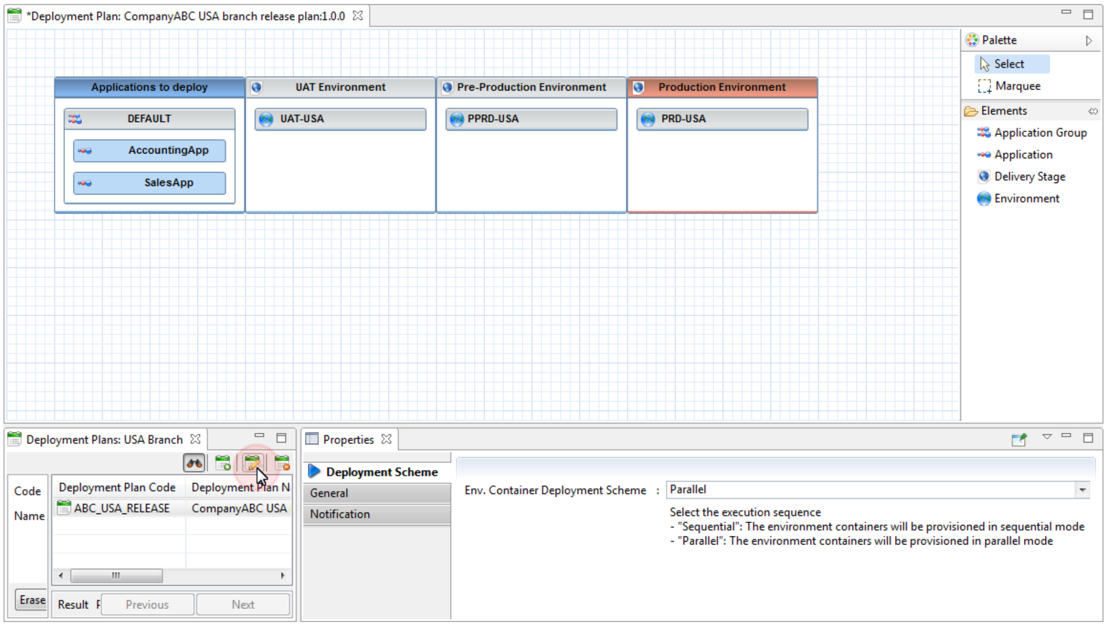
Selecting the element for which to edit the properties in the deployment plan opens the context-sensitive **Properties** view.

> **Note**
> Click the pin icon in this view to remove the context-sensitive property. Clicking this icon ensures that the current entity's properties will stay in the view, even if you navigate away from that entity in the editor. If the pin in not activated, the options in the **Properties** view will change depending on what entity is currently selected in the editor.

29.6.1 Edit global plan properties

To edit the global deployment plan properties either right-click on the plan in the **Deployment Plans** search view then select **Edit**, select it then click the edit icon or double-click on it.

If the **Properties** view is already opened, navigating to the deployment plan will open the global properties in that view.



Deployment Plan Code	Deployment Plan N
ABC_USA_RELEASE	CompanyABC USA

Env. Container Deployment Scheme : Parallel

Select the execution sequence
- "Sequential": The environment containers will be provisioned in sequential mode
- "Parallel": The environment containers will be provisioned in parallel mode

Figure 74: Edit deployment plan properties

The tabs on the left enable you to modify the plan's different global properties.

Deployment Scheme

Env. Container Deployment Scheme

If there are multiple stages in a deployment plan, selecting to execute a **Sequential** deployment scheme will deploy full-transfer deployment process instances for each environment in each stage order, right-to-left, one by one. The next full-transfer instance will not be executed until the previous instance is successful. If an error is encountered, the deployment plan instance will fail.

To first deploy transfer-only deployment process instances for each environment in each stage, in order, then, after they are all completed, deploy each installation-only instance, select **Parallel**.

General

Deployment Plan Code

A unique string that identifies the deployment plan. This code is used throughout DROPS.

Deployment Plan name

A label used to give additional information about the deployment plan. This code is used throughout DROPS.

Notification

Click the **Create** button to create a new notification linked to the deployment plan.

Notifications messages are not like validation/suspension/interruption messages in that they are intended to be only informative. They do not require action from the user and simply let them know that certain steps have been carried out in the process. Notifications are sent to the email address configured for each user who is assigned the role(s) selected.

When & Subject

On Start sends the notification email to the user(s) at execution.

On End sends the notification email to the user(s) after execution.

Enter the subject line for the email in the **Subject** field.

Message

The body of the email sent to the user(s).

Custom recipients

Enter a CSV list of email addresses, without spaces, and those addresses will be notified when the process is executed.

Recipient Roles

Select the role of the user(s) that should receive the notification. The list of Roles is prepopulated with the roles that are allowed to validate.

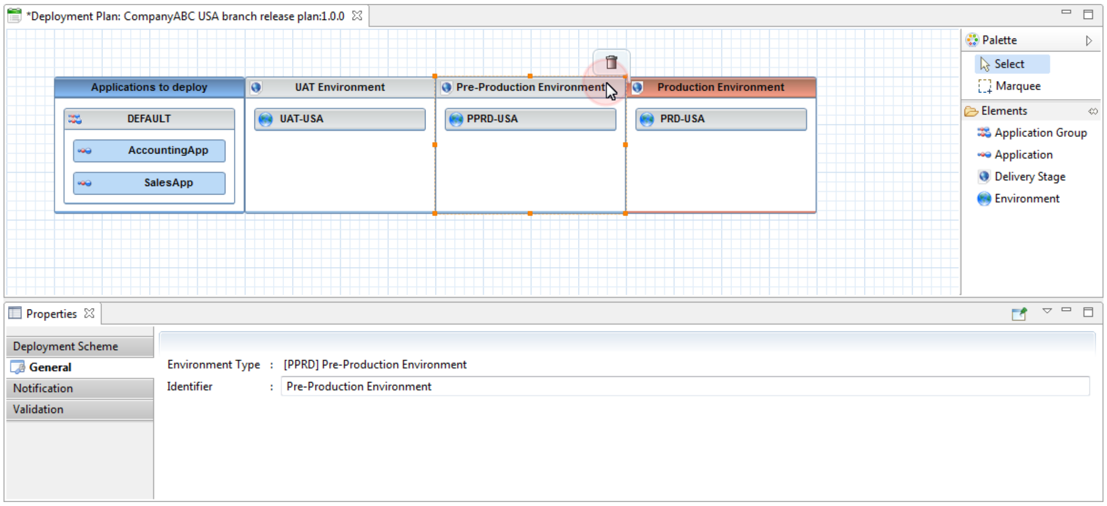
Save the changes (, Ctrl+S or **File** > **Save**).

> ***Warning!***
>
> If there are errors in the plan and DROPS cannot save it, the **Invalid Items** view displays the incomplete elements. If you close the plan without fixing the issues listed and successfully saving, any changes will be lost!

29.6.2 Edit delivery stage properties

To edit the delivery stage-specific properties in the deployment plan, click on the delivery stage.

If the **Properties** view is already opened, clicking on any stage will open its properties in that view.



Environment Type : [PPRD] Pre-Production Environment
Identifier : Pre-Production Environment

Figure 75: Edit delivery stage properties

Note
There are no properties to define for the **Applications to deploy** container.

The tabs on the left enable you to modify the environment stages' different properties.

Deployment Scheme

Env. Deployment Scheme

If there are multiple environments in a delivery stage, selecting to execute a **Sequential** deployment scheme will deploy full-transfer deployment process instances to each environment in order, from top to bottom, one by one. The next full-transfer instance will not be executed until the previous instance is successful. If an error is encountered, the deployment plan instance will fail.

To first deploy transfer-only deployment process instances to each environment in order, then, after they are all completed, deploy each installation-only instance, select **Parallel**.

General

Environment Type

Delivery stages can only contain one type of environment. This uneditable type is listed here..

Identifier

The editable display name of the delivery stage container.

Notification

Click the **Create** button to create a new notification linked to the deployment plan.

Notifications messages are not like validation/suspension/interruption messages in that they are intended to be only informative. They do not require action from the user and simply let them know that certain steps have been carried out in the process. Notifications are sent to the email address configured for each user who is assigned the role(s) selected.

When & Subject

On Start sends the notification email to the user(s) at execution.

On End sends the notification email to the user(s) after execution.

Enter the subject line for the email in the **Subject** field.

Message

The body of the email sent to the user(s).

Custom recipients

Enter a CSV list of email addresses, without spaces, and those addresses will be notified when the process is executed.

Recipient Roles

Select the role of the user(s) that should receive the notification. The list of Roles is prepopulated with the roles that are allowed to validate.

✓ Validation

Validation Required Before Starting

To require validation from a specific user before being able to execute this part of the deployment, tick the checkbox. During deployment, the element is executed as soon as the validator approves the request.

Validation requests display in the ✓ **Validation Requests** node (under the relavent parent node in the **Operational Console Management** perspective) for the user(s) assigned the role(s) selected. They also receive an automatic email.

> ***Reference***
>
> For more information about approving requests, refer to Validation requests on page 296.

For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide*.

Validation Request Message & Comment

The subject line and body of the email to send to the validator. If defined here, the default content of the email, defined in the Preferences menu, is overridden.

Validation comment is required

Tick this checkbox if the user is required to provide a reason why they validate or reject the process.

Validation Roles

Select the role of the user(s) that should receive the validation request. The list of roles is prepopulated with all the Roles that can validate the entity.

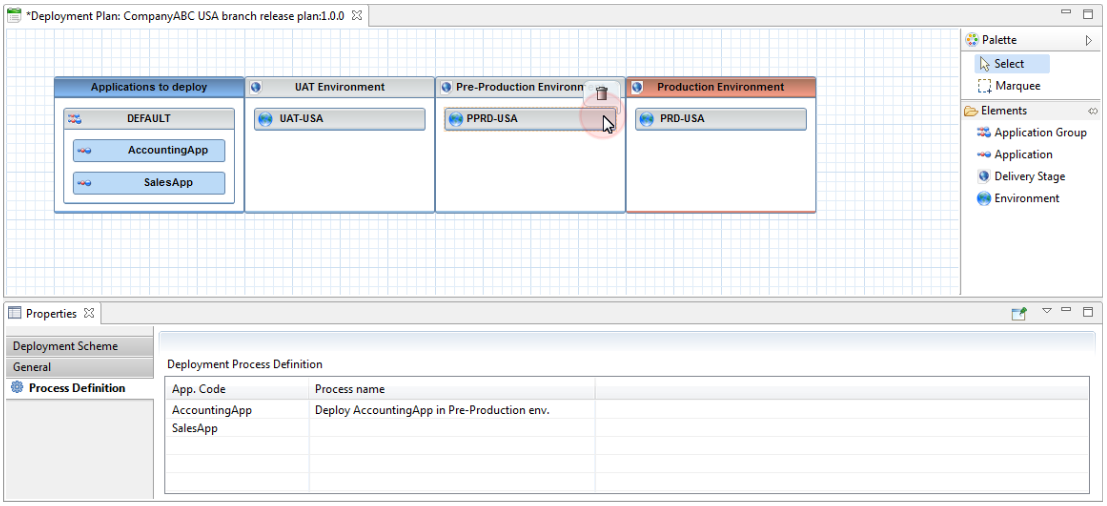
Save the changes (, Ctrl+S or **File** > **Save**).

> ***Warning!***
> If there are errors in the plan and DROPS cannot save it, the **Invalid Items** view displays the incomplete elements. If you close the plan without fixing the issues listed and successfully saving, any changes will be lost!

29.6.3 Edit environment properties

To edit the environment-specific properties in the deployment plan, click the gray or red environment box.

If the **Properties** view is already opened, clicking on any environment will open its properties in that view.



Deployment Process Definition

App. Code	Process name
AccountingApp	Deploy AccountingApp in Pre-Production env.
SalesApp	

Figure 76: Edit deployment plan environment properties

Deployment Scheme

Release Deployment Scheme

If there are multiple deployment process diagrams defined for an application/environment pair in the plan, selecting to execute a **Sequential** deployment scheme will deploy full-transfer deployment process instances for each diagram in order, one by one. The next full-transfer instance will not be executed until the previous instance is successful. If an error is encountered, the deployment plan instance will fail.

To first deploy transfer-only deployment process instances for each process diagram for the pair, in order, then, after they are all completed, deploy each installation-only instance, select **Parallel**.

General

Environment

The read-only code and name of the selected environment.

Process Definition

In order for the deployment plan to know how to deploy each application to each target environment, each pair must already have a deployment process defined for the current client.

Define which process the plan should use for each application+environment pair in this tab.

Double-click on every application and select the deployment process from the prepopulated dialog.

> ***Warning!***
> Every environment in the deployment plan *must* have a deployment process defined for each of the applications in the plan in order to save the plan.

> **Note**
> If the environment or the application have only one process, this process is selected by default.

> ***Reference***
> For more information about creating deployment processes, refer to Deployment processes on page 160.

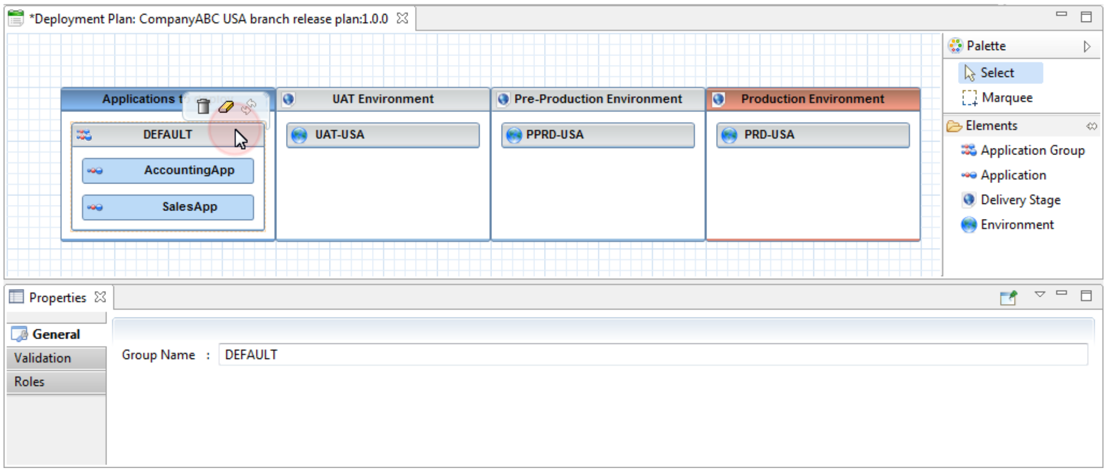
Save the changes (, Ctrl+S or **File** > **Save**).

> ***Warning!***
> If there are errors in the plan and DROPS cannot save it, the **Invalid Items** view displays the incomplete elements. If you close the plan without fixing the issues listed and successfully saving, any changes will be lost!

29.6.4 Edit application group properties

To edit the application group-specific properties in the deployment plan, click the gray application group box.

If the **Properties** view is already opened, clicking on any group will open its properties in that view.



Group Name : DEFAULT

Figure 77: Edit application group properties

> ***Reference***
> For more information about the different actions managed from the editor application group properties that are not described in this chapter, refer to:

- Assigning entity-level roles on page 54

General

Group Name

Edit the name of the group displayed in the diagram.

✓ Validation

Validation Required Before Starting

To require validation from a specific user before being able to execute this part of the deployment, tick the checkbox. During deployment, the element is executed as soon as the validator approves the request.

Validation requests display in the ✓ **Validation Requests** node (under the relavent parent node in the **Operational Console Management** perspective) for the user(s) assigned the role(s) selected. They also receive an automatic email.

> ***Reference***
> For more information about approving requests, refer to Validation requests on page 296.

For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide*.

Validation Request Message & Comment

The subject line and body of the email to send to the validator. If defined here, the default content of the email, defined in the Preferences menu, is overridden.

Validation comment is required

Tick this checkbox if the user is required to provide a reason why they validate or reject the process.

Validation Roles

Select the role of the user(s) that should receive the validation request. The list of roles is pre-populated with all the Roles that can validate the entity.

Save the changes (, Ctrl+S or **File** > **Save**).

> ***Warning!***
> If there are errors in the plan and DROPS cannot save it, the **Invalid Items** view displays the incomplete elements. If you close the plan without fixing the issues listed and successfully saving, any changes will be lost!

Templates

Introduction to configuration templates

Required Roles	n/a
Required Rights (by category)	Application Templates Component Types Strategy Templates Process Templates Provisioning Templates Instantiation Templates
Perspective	Configuration Management
Access	DROPS Server → Templates

Most key entities required to configure the deployment process have associated template features or built-in, predefined entity characterizations, called types.

Types are definitions and properties for the entities, and a set of these default types are built into the DROPS Server. New types can be added and the existing types can be modified according to a given configuration (see Entity types on page 407 and Component types on page 212).

Templates are the association of properties, configuration settings and even other entities to be used when creating a new entity in DROPS.

Templates make the creation of entities easier by automatically pre-populating them with related items. Templates also allow the reuse of a typical configuration for a repeated given situation by creating associations between entities.

The following templates are available to make creating generic entities easier:

- Application templates (see Application templates on page 209).
- Component types (see Component types on page 212),
- Import strategy templates (see Import strategy templates on page 215),
- Deployment process templates (see Deployment process templates on page 218),
- Environment provisioning templates (see Environment provisioning templates on page 223),
- Instantiation templates (see Instantiation templates on page 229).

Environments can also be turned into templates to some extent, in particular cases.

> ***Reference***
> For more information about template environments, refer toEditing environment details on page 98

Templates articulate themselves at different levels in the process. All template features can be used in combination or separately.

30 Application templates

Required Roles	n/a
Required Rights (by category)	Application Templates
Perspective	Configuration Management
Access	DROPS Server → Templates → Application Templates

Chapter Summary

30.1 Creating application templates 210
30.2 Deleting application templates 211

Build an application template and use it to create new applications with a number of predefined sets of entities associated with it. The application template is the root of all the other templates and may be used to apply any combination of templates to new entities in cascading order.

> **Note**
> The procedures presented in this documentation will use all of the features available to build a full **application template** but not all of the template features are required and they can each be used on their own or in any combination.

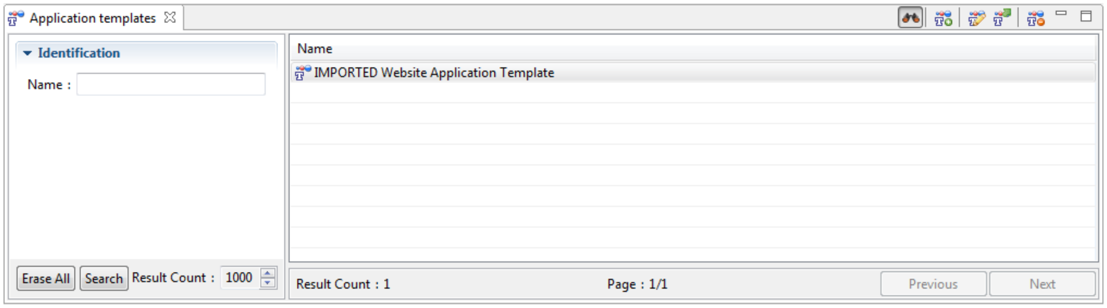
An application template contains a number of component types. These component types are, on one hand, markers to define the types of artifacts that the application contains. On the other hand, they can call other templates to be used in the import and deployment processes. Associate a component type with an import strategy template to automatically define the type of artifacts to import. Associate a component type with a deployment activity template that will be used for infrastructure item types in the target deployment environment to automatically populate infrastructure (of the corresponding type) with the deployment activity templates defined for the component types.

Apply the template when creating a new **application**. The new application will be populated with all of the components whose properties match those of the component type(s) defined in the application template. **Import strategies** will also be created for each of the components, according to any templates declared in the component type.

> **Important!**
> Once the application template has been applied, the application and its components can be edited independently of the template used to create it.

> **Note**
> Templates cannot be applied to an existing application, but component types can be changed, and the associated import strategy templates and deployment activity templates are then applied to the component.

The **Application Templates** search view is accessed from the **Templates** node in the **Configuration Explorer**.



Name
IMPORTED Website Application Template

Result Count : 1 Page : 1/1

Figure 78: The Application templates search view

30.1 Creating application templates

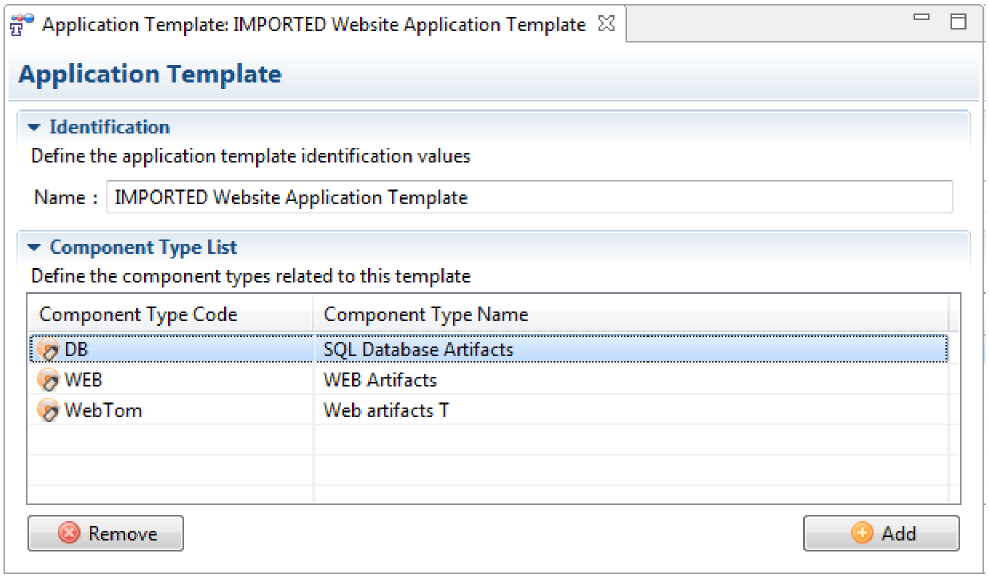
Follow the subsequent steps to create an application template.

Step 1 Create an new application template.

Click the create icon in the **Application Templates** search view or right-click anywhere in the view and select **Create a new application template** to access the **Create a new application template** wizard. Enter a **Name** for the new template.

Click **Finish**.

Step 2 Open the application template editor to define the component type(s) for the template. Either right-click on it then select **Edit**, select it then click the edit icon or double-click on it.


Application Template: IMPORTED Website Application Template
Application Template
Identification
Define the application template identification values
Component Type List
Define the component types related to this template
Remove
Add

Name : IMPORTED Website Application Template

Component Type Code	Component Type Name
DB	SQL Database Artifacts
WEB	WEB Artifacts
WebTom	Web artifacts T

Figure 79: The Application Template editor

Step 3 Manage the template's **component type(s)**.

Click the **Add** button and select the component type(s) from the list [Ctrl+click] to add component types to the application template. The list is prepopulated with all of the component types defined for the current server. Click **OK**.

To remove a component type from an application template select it in the **Component Type List** and click **Remove**.

Result The **application templates** are ready to be used when creating new applications.

30.2 Deleting application templates

Note

Application templates, when used, will populate new applications with component types and all the other cascading template entities. Deleting an application template has no consequence on any application that used it or its content.

Warning!

Deleted application templates cannot be accessed or recovered and cannot be used to create new applications.

To delete an application template, either right-click on it in the **Application Templates** search view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the template.

31 Component types

Required Roles	n/a
Required Rights (by category)	Component Types
Perspective	Configuration Management
Access	DROPS Server → Templates → Component Types

Chapter Summary

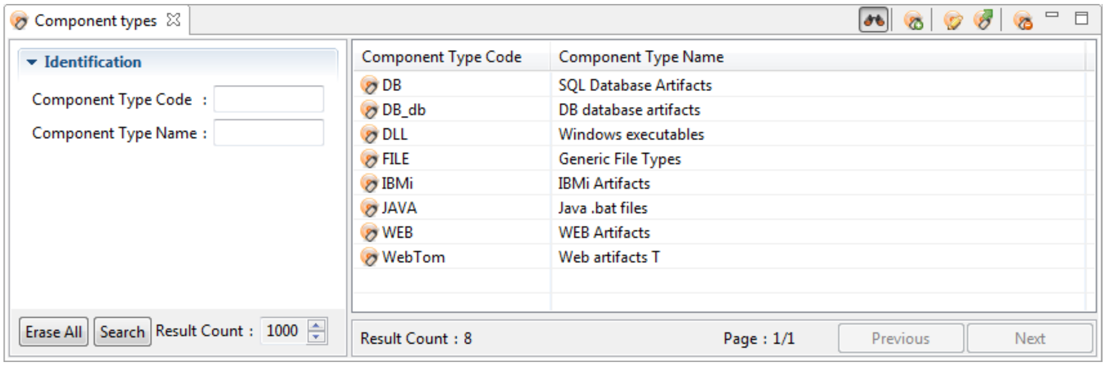
31.1 Creating component types 213
31.2 Deleting component types 214

Component types are used to categorize components in order to easily see what kind of artifacts they contain. Components can contain any type of artifact and any combination of different artifacts. Usually, they contain a group of artifacts that together create an individual part of a larger application/software/website/etc. Categorizing them creates a way to identify these individual parts of the whole application.

> ***Example***
> Component types might be web archives, executables, SQL database artifacts, IBM i save files, etc.

Component types are selected when creating and/or editing a component but selecting a component type is not mandatory.

If strategy or activity templates are not defined for the component type, choosing a type does not change how a component functions. However, if an import strategy template or a deployment activity template have been defined for the component type, choosing a type will associate the templates to the component; and changing the type of a component will override any templates previously associated with the component.



Component Type Code	Component Type Name
DB	SQL Database Artifacts
DB_db	DB database artifacts
DLL	Windows executables
FILE	Generic File Types
IBMi	IBMi Artifacts
JAVA	Java .bat files
WEB	WEB Artifacts
WebTom	Web artifacts T

Figure 80: The Component Types search view

> ***Reference***
> For more information about managing components, refer to Components on page 67.

The **Component Types** search view is accessed from the **Templates** node in the **Configuration Explorer**.

31.1 Creating component types

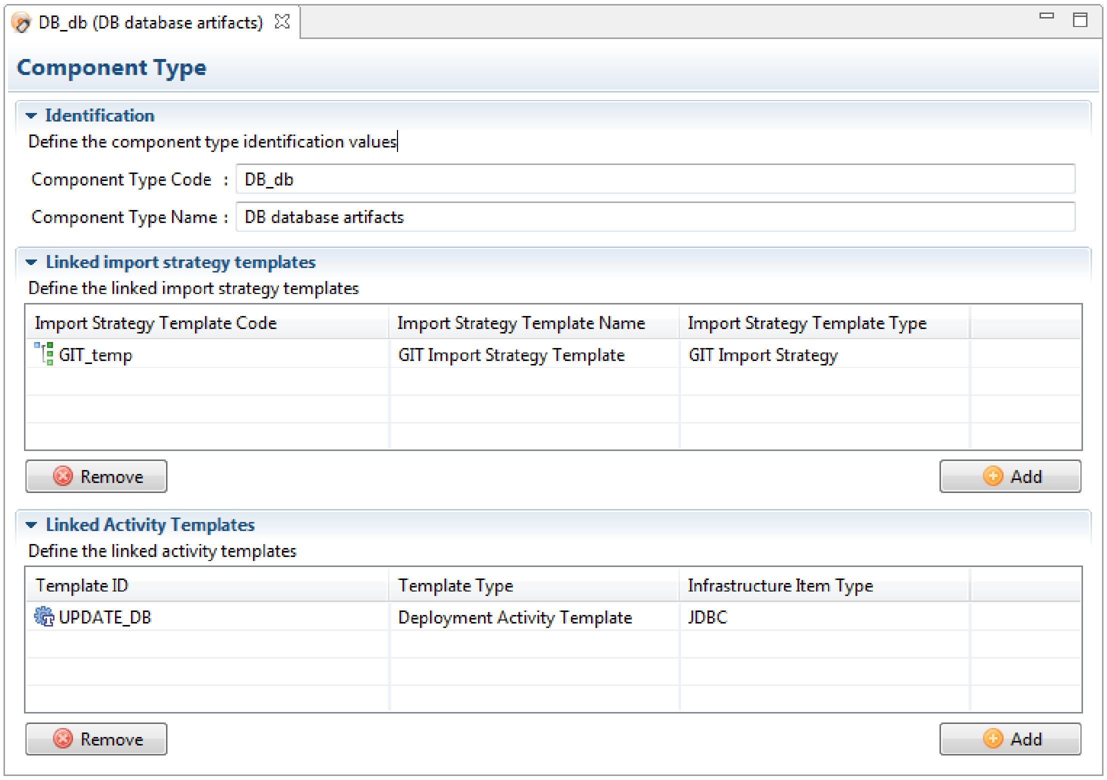
Follow the subsequent steps to create a component type.

Step 1 Create an new component type.

Click the create icon in the **Component Types** view or right-click anywhere in the view and select **Create a new component type** to access the **Create a new component type** wizard. Enter a **Name** and a **Code** for the new component type.

Click **Finish**.

Step 2 Open the **Component Type** editor to define the strategy and activity templates to link. Either right-click on a component type then select **Edit**, select it then click the edit icon or double-click on it.



DB_db (DB database artifacts)

Component Type

Identification

Define the component type identification values

Component Type Code : DB_db

Component Type Name : DB database artifacts

Linked import strategy templates

Define the linked import strategy templates

Import Strategy Template Code	Import Strategy Template Name	Import Strategy Template Type
GIT_temp	GIT Import Strategy Template	GIT Import Strategy

Linked Activity Templates

Define the linked activity templates

Template ID	Template Type	Infrastructure Item Type
UPDATE_DB	Deployment Activity Template	JDBC

Figure 81: The component type editor

Step 3 Associate a component type with an import strategy template to automatically define the type of artifacts to import. Associate a component type with a deployment activity template that will be used for infrastructure item types in the target deployment environment to automatically populate infrastructure (of the corresponding type) with the deployment activity templates defined for the component types.

Click the **Add** button in the **Linked Strategy Templates** section and select one or more import strategy templates from the list. Click **OK**.

Click the **Add** button in the **Linked Activity Templates** section and select one or more import strategy templates from the list. Select the corresponding infrastructure item types from the next list dialog. Click **OK**.

> **Note**
> Component types can only be associated with the **deployment activity template** type of deployment process templates (see Deployment process templates on page 218

Result The component type is ready to be used in an application or an application template.

31.2 Deleting component types

> **Note**
> Component types, when defined in a component, will populate the component with the component type's related properties or template entities. Deleting the component type will remove the type definition from the components that use it but has no consequence on the components' existing content.

> ***Warning!***
> Deleted component types cannot be accessed or recovered.

To delete a component type, either right-click on it in the **Component Type** search view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the component type.

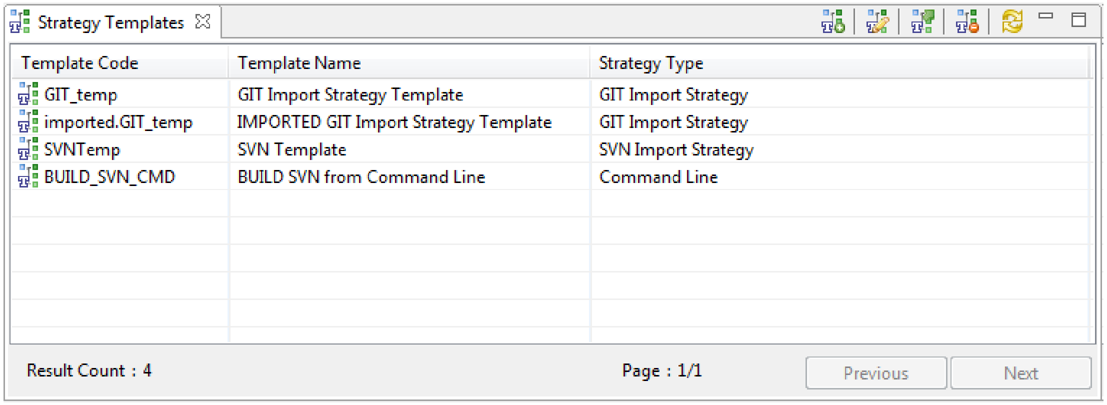
32 Import strategy templates

Required Roles	n/a
Required Rights (by category)	Strategy Templates
Perspective	Configuration Management
Access	DROPS Server → Templates → Import Strategy Templates

Chapter Summary

32.1 Creating import strategy templates 215
32.2 Editing import strategy templates 216
32.3 Deleting import strategy templates 217

Users with the rights to do so can create import strategy templates that can be selected and then modified in order to save time while creating new import strategies. An import strategy template can contain some or all of the details required to use the import strategy, including properties.



Template Code	Template Name	Strategy Type
GIT_temp	GIT Import Strategy Template	GIT Import Strategy
imported.GIT_temp	IMPORTED GIT Import Strategy Template	GIT Import Strategy
SVNTemp	SVN Template	SVN Import Strategy
BUILD_SVN_CMD	BUILD SVN from Command Line	Command Line

Result Count : 4

Page : 1/1

Figure 82: The import strategy templates view

The **Import Strategy Templates** view is accessed from the **Templates** node in the **Configuration Explorer**.

32.1 Creating import strategy templates

Follow the subsequent steps to create a new import strategy template.

Step 1 To access the **Create a new import strategy template** wizard, either click the create icon in the **Import Strategy Templates** view or right-click anywhere in the view and select **Create a new import strategy template**.

Step 2 Define the template's **Code**, **Name** and the Root Directory. These values can be edited later.

Click **Next >** to continue.

Step 3 Select the Strategy Type. The type of strategy to use for a template cannot be changed.

> **Important!**
> Take care when selecting the type to use based on the type of source your artifacts are coming from.

Step 4 Click **Finish**.

Result New import strategy templates are available in the **Import Strategy Templates** view after refreshing the list. It is recommended that you edit the template's details before using it.

> ***Reference***
> For more information about using import strategy templates, refer to Use or copy from strategy templates on page 74.

32.2 Editing import strategy templates

To edit strategy template details, either right-click on it then select **Edit**, select it then click the edit icon or double-click it.

Follow the instructions in Editing import strategy details to modify the details of a strategy template.

> **Important!**
> Any modifications made to import strategy templates are automatically carried over to ALL of the import strategies that *use* it but not to the strategies that were copied from it.

Using properties in import strategy templates can help you automate the import process by calling information defined in applications, components and import strategies.

The following properties/substitution variables can be called into the extended configuration of an import strategy or a import strategy template:

- any user-defined property in an application
- the ${activity.[activityID].component.code} substitution variable
- any user-defined property in an import strategy

> ***Example***
> Using the ${importStrategy.props.env} variable in a template enables you to create an "env" property in the import strategy that *uses* the template which will be called when the extended configuration is taken into account.

> ***Reference***
> For more information about using properties, refer to Properties and Substitution variables in the appendix.

32.3 Deleting import strategy templates

> ***Warning!***
>
> Deleted import strategy templates cannot be accessed and do not appear in the list of templates to use when creating a new import strategy.
>
> If an active import strategy is linked to an import strategy template and the template is deleted, the link to the template is broken. The strategy is then independent from any template but can still be used and still contains the information carried over from the original template.

To delete an import strategy template, either right-click on it in the **Import Strategy Templates** view then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the template.

33 Deployment process templates

Required Roles	n/a
Required Rights (by category)	Process Templates
Perspective	Configuration Management
Access	DROPS Server → Templates → Process Templates

Chapter Summary

33.1 Creating deployment process templates 219
33.2 Editing deployment process templates 221
33.3 Viewing the process diagrams that use a template 221
33.4 Deleting deployment process templates 222

Users with the rights to do so can create deployment process templates that can be selected in order to save time while creating new processes. A deployment process template can contain some or all of the details required to deploy a release, including properties. The levels in a diagram are nested and once a template is used in a diagram it is completely uneditable, so any template must be configured completely.

The type of template defines which level(s) of the deployment process diagram are used. There are five different types of deployment process templates and each one represents a different level and all of the levels it contains.

- **Stage templates** must include actions to predefine the steps to carry out when the stage is executed. Templates can be created for all three of the different stages in a deployment process:
  - preparation stage
  - receiving stage
  - installation stage
- **Activity templates** are only for infrastructure item activities. Activity templates must include either at least one stage (which must also include at least one action) or at least one action outside of a stage. The same requirements for a normal, unique deployment activities apply to activity templates, including most of the properties.

> **Note**
> Initialization and finalization activity templates cannot be created. These activities must always be modified in the process diagram and remain unique to each deployment process.

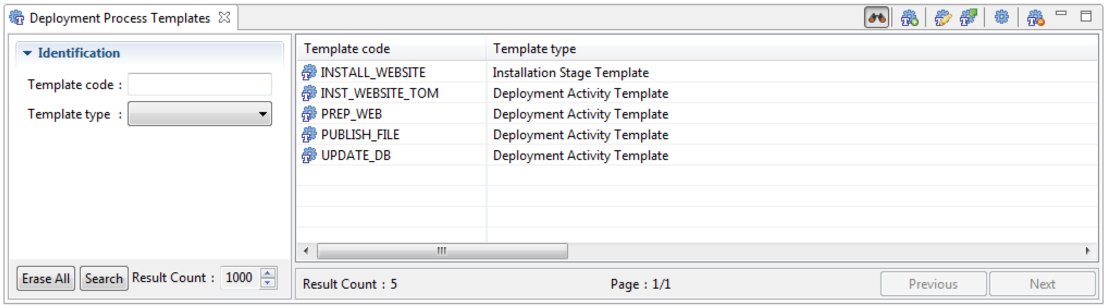
- **Deployment process properties templates** are pre-defined process-level properties and parameters which can be applied to whole deployment process diagrams to automatically define this information without using uneditable elements (activity or stage templates) in the diagram itself.

Templates can be used in any deployment process diagram but can only be modified in the **Deployment Process Templates** editor. Any changes made to a template are carried over to all of the process diagrams using it.

When a template is used in a diagram, the activity or stage is dark blue, instead of the default gray. The difference in color makes it easy to see which parts of a diagram are inherited from a template and which are unique.

> ⚠ **Important!**
> Because activity and stage templates can be used for any infrastructure item, any user-defined properties included in variables that are referenced in the template *must* be defined for every item that may be included in a deployment process that uses the template. User-defined variables always have **.props**. before the property name so you can be sure to know which variables called must be defined in the entities referenced.

> ***Reference***
> For more information about calling user-defined properties as variables, refer to Properties and Substitution variables in the appendix.



Template code	Template type
INSTALL_WEBSITE	Installation Stage Template
INST_WEBSITE_TOM	Deployment Activity Template
PREP_WEB	Deployment Activity Template
PUBLISH_FILE	Deployment Activity Template
UPDATE_DB	Deployment Activity Template

Figure 83: The Deployment Process Templates search view

The **Process Templates** search view is accessed from the **Templates** node in the **Configuration Explorer**.

33.1 Creating deployment process templates

There are two ways to create deployment activity and stage templates:

1. Create a new template manually using the **Create a new deployment process template** wizard.
2. Create a template from an existing element in a deployment process diagram.

Deployment process templates can only be created using the **Create a new process template** wizard.

33.1.1 Create a deployment process template manually

Follow the subsequent steps to create a new process template.

Step 1 To access the **Create a new deployment process template** wizard, either click the create icon in the **Deployment Process Templates** search view or right-click anywhere in the view and select **Create a new template**.

Step 2 Complete all of the fields.

1. Define the template's Code. This value can be edited later in the property tab.
2. Select the type. The **Type** drop-down list is prepopulated with the five types of templates available.

> **Important!**
> Be sure to select the correct type based on your needs. A template's type cannot be modified later.

> ***Reference***
> For more information about template types, refer to the introduction to this chapter or the Glossary on page 475.

Step 3 Click **Finish**.

Result New templates are available in the **Deployment Process Templates** search view and must be configured completely before using them in a diagram. The **Process Template** editor or the **Process Option Template** editor is opened automatically.

33.1.2 Create an activity or a stage template from an existing diagram

Step 1 From the **Deployment Process** editor, right-click on the activity or stage to use to create a template.

> **Note**
> It is recommended, but not required, to use completely-defined diagrams to create templates. They can be edited later in the template editor.
>
> Templates cannot be included in other templates. If the activity chosen already contains a stage made from a template, the stage template will not be included in the new activity template.

Result The **Create a new deployment process template** wizard is displayed.

Step 2 Define the template's Code.

Step 3 Click **Finish**.

Result New templates are available in the **Deployment Process Templates** search view and must be configured completely before using them in a diagram. The **Deployment Process Template** editor is opened automatically.

> ***Reference***
> For more information about using the process diagram to create templates, refer to Building deployment process diagrams on page 165.

33.2 Editing deployment process templates

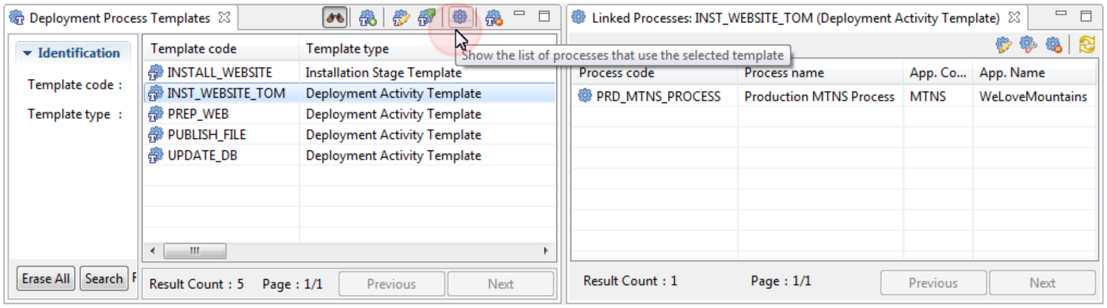
To edit a template either right-click on it in the **Deployment Process Templates** search view then select **Edit Template**, or select it, then click the edit icon or double-click it.

Process templates are edited in the same way as deployment process diagrams. If the **Properties** view is already opened, selecting an activity or a stage template will open the properties in that view. Editing a deployment process template opens the **Process Option Template** editor.

> ***Reference***
>
> For more information about how to build a process template diagram, refer to Building deployment process diagrams on page 165.
>
> For more information about the properties to edit for each entity in a process template, refer to Editing deployment process properties on page 175.

33.3 Viewing the process diagrams that use a template

To view the list of process diagrams that use a process template, select the template in the **Deployment Process templates** search view and either click the show the list of processes icon or right-click on it and select **Show the list of Process that use this template**.


Deployment Process Templates
Identification
Template code :
Template type :
Erase All
Search
Show the list of processes that use the selected template
Linked Processes: INST_WEBSITE_TOM (Deployment Activity Template)
Previous
Next

Template code	Template type
INSTALL_WEBSITE	Installation Stage Template
INST_WEBSITE_TOM	Deployment Activity Template
PREP_WEB	Deployment Activity Template
PUBLISH_FILE	Deployment Activity Template
UPDATE_DB	Deployment Activity Template

Result Count : 5 Page : 1/1

Process code	Process name	App. Co...	App. Name
PRD_MTNS_PROCESS	Production MTNS Process	MTNS	WeLoveMountains

Result Count : 1 Page : 1/1

Figure 84: Viewing deployment process that use process templates

The **Linked Processes** view is displayed.

Click the edit icon or the properties editor icon to modify the diagram or its properties.

Click the delete icon to delete the deployment process.

> ***Warning!***
>
> Deleting deployment processes from the **Linked Processes** view deletes the process completely. It does not simply unbind the process from the template. To remove the template from a deployment process, open the deployment process diagram and delete the template item from the diagram.

Reference

For more information about the options available in this view, refer to Deployment processes on page 160.

33.4 Deleting deployment process templates

Note

If a template is used in any deployment process diagram, it cannot be deleted. Delete the template item in the process diagram first then delete it from the template search view.

Warning!

Deleted process templates cannot be accessed or recovered.

To delete a deployment process template, either right-click on it in the **Deployment Process Templates** search view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the template.

34 Environment provisioning templates

Required Roles	While no role is required for a user to work with provisioning templates, a user must play the role of Manager on any infrastructure item to which the template is associated to launch the provisioning engine.
Required Rights (by category)	Provisioning Instances Provisioning Templates
Perspective	Configuration Management
Access	DROPS Server → Templates → Provisioning Templates

Chapter Summary

34.1 Creating provisioning templates 223
34.2 Editing provisioning templates 224
34.3 Associating provisioning templates with infrastructure items 227
34.4 Deleting provisioning templates 227

Build a provisioning template to prepare infrastructure items to receive an application's installation or update if they are not already set up to do so. These templates allow you to create scripts to provision empty infrastructure items on-the-fly with all the elements required for the artifacts that will later be deployed to it. Provisioning an environment requires DROPS to install a middleware layer (servers, services, etc.).

Provisioning templates can be associated with existing or new infrastructure items. Once associated with an item, the provisioning process can be launched manually or will be automatically launched when an application is deployed to the item. When the deployment activity associated with the item is executed, the provisioning process will be started if the item is associated with a provisioning template and it is not marked as "already provisioned".

> **Note**
> If the provisioning execution is launched during a deployment and fails, the entire deployment process execution will fail.

The **Provisioning Templates** search view is accessed from the **Templates** node in the **Configuration Explorer**.

34.1 Creating provisioning templates

Follow the subsequent steps to create a provisioning template.

Step 1 Click the create icon in the **Provisioning Templates** search view or right-click anywhere in the view and select **Create a new provisioning template** to access the **Create a new provisioning template** wizard. Enter a Code and Name for the new template, then select the Provisioning Type of template to create.

> **Important!**
> The template type cannot be changed once the template is created.

Click **Finish**.

Step 2 Open the provisioning template editor to define the rest of the properties. Either right-click on it then select **Edit**, select it then click the edit icon or double-click on it.

> ***Reference***
>
> Refer to Editing provisioning templates below

Result The provisioning templates are ready to be used when executing the deployment process.

34.2 Editing provisioning templates

A provisioning template's identification and connection details can be edited as well as the ANT project content of a template's script, if it has one. Command line-type provisioning templates do not require ANT project scripts.

34.2.1 Edit provisioning template details

To edit provisioning template identification details, either right-click on it then select **Edit**, select the item then click the edit icon or double-click it. The **Provisioning Template** editor is opened where all of the entity's details are managed.

> ***Reference***
>
> For more information about extended properties, which are not described in this chapter, refer to:
>
> - Properties on page 412

Code

A unique string that identifies the template. This label is used throughout DROPS to select the entity.

Name

A string used to give additional information about the template. This label is used throughout DROPS to select the entity.

Provisioning Type

The template type is selected when creating the template and cannot be changed.

Attachments to publish

Click the add button to add an external file as an attachment to this template.

> **Tip**
>
> Use Substitution variables in the path name.

Force provisioning

If both a provisioning template and an instantiation template are referenced in an infrastructure item, ticking this option will force the provisioning process even if there are errors in the instantiation process.

Working directory

Enter a working directory to use as a variable.

Provision Agent

Infrastructure items can call remote agents to manage installation procedures. If this box is checked, the remote agent defined in the Installation Properties defined in the associated infrastructure item will be called.

> **Note**
> The agent provisioning process is the same as the one executed by the installer module.

Failed if Provisioning Agent failed

Tick this box to mark the provisioning process as failed if the remote agent defined in the item's Installation Properties fails.

User/Password

Both of these values can be used as variables in properties.

SSH Key

A reference to an SSH key defined using the SSH Key Module, if needed.
The absolute path to the public key (the id_rsa.pub file) related to the selected key can be used as a variable - ${this.sshkeypath}.

> ***Reference***
> SSH keys on page 129

ANT Processor

Field	Description
Target	The name of the target to execute. This target has to exist in the script file that represents the ANT Project.
ANT Base directory	The base directory passed to the ANT Executor.
Additional ANT libraries	Enter a semi-colon separated list and the JAR files that declare specific ANT Tasks.
Set up the ANT libraries	Tick this box to set up the libraries.

Table 10: Edit provisioning engine properties: ANT processor

Command Line Processor

Field	Description
Executable path	The path to the external command to execute.
Working directory	The working directory from where the execution will be done.
Command template	The parameters that will be passed to the executable. These values can be used as variables - ${this.workingdir}

Table 11: Edit provisioning engine properties: Command-Line processor

Ansible Playbook processor

Field	Description
Working directory	The path to the working directory from where the execution will be done.
Additional parameters	Enter any additional parameters to use to execute the playbook tasks.

Table 12: Edit provisioning engine properties: Ansible Playbook processor

Microsoft Azure

Field	Description
Subscription ID.	Enter the Azure Subscription ID to access the Azure cloud platform.
Tenant ID.	Enter the Azure Tenant ID associated with the Azure Subscription.
Resource Group	Select the Resource Group you wish to use.
Use an URL	Tick to use URLs for the management of templates files.
Template URL	Provide the URL to the ARM template file.
Parameter URL	Provide the URL to the ARM template's parameter file. Leave blank is the parameters are defined in the ARM template file. For more information, refer to the Azure ARM templates practices.

Table 13: Edit provisioning engine properties: Microsoft Azure

Save the changes (, Ctrl+S or **File** > **Save**).

34.2.2 Edit provisioning script content

Select a template from the list and click the edit content icon or right-click on it and select **Edit Content**.

The **Script** content editor is opened where you can write or edit the content of the template's script. Editing the content of a script can be done directly in DROPS. Eclipse's Ant Editor offers shortcuts

(Ctrl+space) to provide easy access to all of the Ant project templates, tasks, parameters and variables available.

> **Important!**
> When creating or modifying scripts, the content must be a valid Ant project.

> **Note**
> There are no templates for provisioning template scripts. You are responsible for the content of your provisioning scripts.

You can use variables to define provisioning templates that will be replaced by the actual value when the process is launched. A variable can be declared using the following syntax: ${<variable.name}

All of the infrastructure item's properties can be used as variables as well as 2 others, specific to this process:

- the main attributes - ${infra.<attribute.name>}
- the properties - ${infra.props.< property.name>}
- the extended properties - ${infra.config.<extended.property.name>}
- the installation properties - ${infra.install.<install.property.name>}
- the user/password - ${this.user} and ${this.password}.
- the absolute path of the related script - ${provisioning.script}
- the value of the Working Directory defined in the provisioning template - ${provisioning.root}

Save the changes (🖫, Ctrl+S or **File** > **Save**).

34.3 Associating provisioning templates with infrastructure items

To associate a provisioning template with an infrastructure item so that it is automatically launched when the item is used in a deployment process, open the item's editor and select the template in the **Provisioning Template** field.

> ***Reference***
> For more information about editing infrastructure items, refer to Editing infrastructure items on page 115.

34.4 Deleting provisioning templates

> ***Warning!***
> Deleted provisioning templates cannot be accessed or recovered and cannot be used during the deployment process. If a template is deleted, it will be removed from any infrastructure item that referred to it.

To delete a provisioning template, either right-click on it in the **Provisioning Templates** search view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the template.

35 Instantiation templates

Required Roles	While no role is required for a user to work with instantiation templates, a user must play the role of Manager on any infrastructure item to which the template is associated to launch the instantiation engine.
Required Rights (by category)	Instantiation Instances Instantiation Templates
Perspective	Configuration Management
Access	DROPS Server → Templates → Instantiation Templates

Chapter Summary

35.1 Creating instantiation templates 229
35.2 Editing instantiation templates 230
35.3 Associating instantiation templates with infrastructure items 233
35.4 Deleting instantiation templates 233

Build an instantiation template and use it to create new VMs for infrastructure items automatically. This creates the item and recovers the ID after it's created. Instantiating an environment requires DROPS to declare a VM.

Once associated with an item, the instantiation process can be launched manually or will be automatically launched when an application is deployed to the item. When the deployment activity associated with the item is executed, the instantiation process will be started if the item is associated with a instantiation template and it is not marked as "already instantiated".

If an infrastructure item is linked to both an instantiation and a provisioning template, the instantiation process will be run before the provisioning process in order to create the item first, then provision it.

> **Note**
> If the instantiation execution is launched during a deployment and fails, the entire deployment process execution will fail.

The **Instantiation Templates** search view is accessed from the **Templates** node in the **Configuration Explorer**.

35.1 Creating instantiation templates

Follow the subsequent steps to create an instantiation template.

Step 1 Click the create icon in the **Instantiation Templates** search view or right-click anywhere in the view and select **Create a new instantiation template** to access the **Create a new instantiation template** wizard. Enter a Code and Name for the new template, then select the Provisioning Type of template to create.

> **Important!**
> The template type cannot be changed once the template is created.

Click **Finish**.

Step 2 Open the instantiation template editor to define the rest of the properties. Either right-click on it then select **Edit**, select it then click the edit icon or double-click on it.

> ***Reference***
> Refer to Editing instantiation templates below

Result The instantiation templates are ready to be paired with provisioning templates.

35.2 Editing instantiation templates

To edit instantiation template identification details, either right-click on it then select **Edit**, select the item then click the edit icon or double-click it. The **Instantiation Template** editor is opened where all of the entity's details are managed.

> ***Reference***
> For more information about extended properties, which are not described in this chapter, refer to:
>
> - Properties on page 412

Code

A unique string that identifies the template. This label is used throughout DROPS to select the entity.

Name

A string used to give additional information about the template. This label is used throughout DROPS to select the entity.

Provisioning Type

The template type is selected when creating the template and cannot be changed.

Attachments to publish

Click the Add button to add an external file as an attachment to this template.

> **Tip**
> Use Substitution variables in the path name.

Force Execution

Select this option to systematically re-provision an infrastructure item when deploying even if it has already been provisioned.

If both a provisioning template and an instantiation template are referenced in an infrastructure item, selecting this option will force the provisioning process even if there are errors in the instantiation process.

Working directory

Enter a working directory to use as a variable.

User/Password

Both of these values can be used as variables.

SSH Key

A reference to an SSH key defined using the SSH Key Module, if needed.

The absolute path to the public key (the id_rsa.pub file) related to the selected key can be used as a variable - ${this.sshkeypath}.

Reference
SSH keys on page 129

ANT Processor

Field	Description
Target	The name of the target to execute. This target has to exist in the script file that represents the ANT Project.
ANT Base directory	The base directory passed to the ANT Executor.
Additional ANT libraries	Enter a semi-colon separated list and the JAR files that declare specific ANT Tasks.
Set up the ANT libraries	Tick this box to set up the libraries.

Table 14: Edit provisioning engine properties: ANT processor

Command Line Processor

Field	Description
Executable path	The path to the external command to execute.
Working directory	The working directory from where the execution will be done.
Command template	The parameters that will be passed to the executable. These values can be used as variables - ${this.workingdir}

Table 15: Edit provisioning engine properties: Command-Line processor

Ansible Playbook processor

Field	Description
Working directory	The path to the working directory from where the execution will be done.
Additional parameters	Enter any additional parameters to use to execute the playbook tasks.

Table 16: Edit provisioning engine properties: Ansible Playbook processor

Microsoft Azure

Field	Description
Subscription ID.	Enter the Azure Subscription ID to access the Azure cloud platform.
Tenant ID.	Enter the Azure Tenant ID associated with the Azure Subscription.
Resource Group	Select the Resource Group you wish to use.
Use an URL	Tick to use URLs for the management of templates files.
Template URL	Provide the URL to the ARM template file.
Parameter URL	Provide the URL to the ARM template's parameter file. Leave blank is the parameters are defined in the ARM template file. For more information, refer to the Azure ARM templates practices.

Table 17: Edit provisioning engine properties: Microsoft Azure

Save the changes (, Ctrl+S or **File** > **Save**).

35.2.1 Edit instantiation script content

Select a template from the list and click the edit content icon or right-click on it and select **Edit Content**.

The **Script** content editor is opened where you can write or edit the content of the template's script. Editing the content of a script can be done directly in DROPS. Eclipse's Ant Editor offers shortcuts (Ctrl+space) to provide easy access to all of the Ant project templates, tasks, parameters and variables available.

> **Important!**
> When creating or modifying scripts, the content must be a valid Ant project.

> **Note**
> There are no templates for instantiation template scripts. You are responsible for the content of your instantiation scripts.

You can use variables to define provisioning templates that will be replaced by the actual value when the process is launched. A variable can be declared using the following syntax: ${<variable.name}

All of the infrastructure item's properties can be used as variables as well as 2 others, specific to this process:

- the main attributes - ${infra.<attribute.name>}
- the properties - ${infra.props.< property.name>}
- the extended properties - ${infra.config.<extended.property.name>}
- the installation properties - ${infra.install.<install.property.name>}
- the user/password - ${this.user} and ${this.password}.
- the absolute path of the related script - ${provisioning.script}
- the value of the Working Directory defined in the provisioning template - ${provisioning.root}

Save the changes (, Ctrl+S or **File** > **Save**).

35.3 Associating instantiation templates with infrastructure items

To associate an instantiation template with an infrastructure item so that it is automatically launched when the item is used in a deployment process, open the item's editor and select the template in the **Instantiation Template** field.

Reference
For more information about editing infrastructure items, refer to Editing infrastructure items on page 115.

35.4 Deleting instantiation templates

Warning!
Deleted instantiation templates cannot be accessed or recovered and cannot be used during the deployment process. If a template is deleted, it will be removed from any infrastructure item that referred to it.

To delete an instantiation template, either right-click on it in the **Instantiation Templates** search view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the template.

drops

Main / Node Configuration

Introduction to main / node servers configuration

Managing DROPS in a main server and node configuration allows you to define the main DROPS Server that pilots one or several node DROPS servers in order to delegate deployment tasks to them.

This kind of architecture is useful:

- To separate the responsibilities on different parts of the organization.
- To dispatch the execution workload on different DROPS servers.
- To centralize all the deployment history on a unique point even if the real executions have been delegated.

You need to configure your servers settings before using the solution. In the server's **Configuration Categories**, you define if the current server is a node or a main server.

Note
Further configuration is required specifically for the node and for the main server.

Reference
Refer to the Configuration Guide.

When you have set the server as node, the DROPS navigator lets you access the view to register to the main server and to synchronize with it. When you have set the server as main, the DROPS navigator lets you have access to the views to manage registered nodes.

Reference
Configuring main and node DROPS servers on page 236

Once the status of the servers is defined, the main and the node servers are ready to be paired and their data to be synchronized.

Reference
Synchronizing main and node DROPS servers on page 238

When the servers synchronized, the main server can be used to pilot the deployments which are executed by the node server.

Reference
Remote deployments on node servers on page 276

36 Configuring main and node DROPS servers

Required Roles	Environment Release Manager and/or Release Operator
Required Rights (by category)	Node Management
Perspective	Configuration Management
Access	DROPS Server → Main/Node Configuration → Main/Node Configuration

Chapter Summary

36.1 Registering node servers ... 236
36.2 Managing registered nodes ... 236

As a prerequisites, defining the node settings in the node server's configuration options is required. Refer to the Configuration Guide.

With the node identification properties set, you can register the node server to the main server. This operation is done from the node server.

All the registered nodes are listed and managed in a dedicated view from the main server. From this view, you can check the synchronization status before deployments, or remove registered nodes you no longer need.

36.1 Registering node servers

Register to the main server from the node server.

Define the DROPS Server connection information in the server settings of the **Main/Node Configuration** editor. Click the validation icon to send its own information to the main server.

DROPS Main Server URL

Indicates the URL to reach the main server.

DROPS Main Login & Password

Indicates the user used to connect to the main server and the password of the selected user.

36.2 Managing registered nodes

This section describes the features available on a main server concerning the management of the node.

To test the connection with the node server, right-click on it and select **Test connection**, or select it and click the connection icon.

⚠ **Important!**

> The connection test is made using the current user. If the connection test fails, it does not necessarily mean that the selected node is not reachable. It may be because the connected user is not allowed to connect to this node.

To unregister a node server, right-click on it and select **Unregister**, or select it and click the unregister icon.

Node Properties

The **Node Properties** are not editable. These values are defined by the node itself during the node's self registration process.

Identifier

Indicates the node's identifier.

URL

Indicates the node's URL.

Dates

The fields of the **Dates** section are not editable. These values are retrieved during the synchronization between the node and the main servers.

Last Reachable Date

Indicates the last time this node has been reachable when testing the connection.

Registration Date

Indicates when this node has been registered.

Roles

Grant the node administration role to a user. This role is used to define which users will receive a notification if the related node is not reachable.

Synchronization Requests

This tab displays the list of synchronization requests emitted by the selected node.

Request Date

Indicates when the request has been emitted.

Request Result

Indicates if the request was successful or not.

37 Synchronizing main and node DROPS servers

Required Roles	Environment Release Manager and/or Release Operator
Required Rights (by category)	Node Management
Perspective	Configuration Management
Access	DROPS Server → Main/Node Configuration → Main Configuration

Chapter Summary

37.1 Synchronizing servers 238
37.2 Scheduling synchronization 239
37.3 Synchronization reports 240

Synchronization of the node servers with the main server is essential for the main server to be able to delegate the deployment execution to a node server. The configuration of the node server must replicate the configuration of the main server to be able to find all the information required to carry out a remote execution.

When synchronizing servers, only the configuration data is taken into account. The following information is excluded from this synchronization:

- the execution data (the data generated during process execution),
- the built-in data (the data provided with the DROPS server).

Note

Synchronizing the server configuration is not an export / import action. It is a process that replicates the exact same data from a main server to node servers (including low-level data such as database technical identifiers).

There are several ways to trigger the synchronization:

- You can do a single synchronization manually from the node server.
- You can schedule a periodical automatic synchronization from the node server.
- You can set an option to do a synchronization when executing a remote deployment instance, from the main server.

Reference

For the last option, refer to the Configuration Guide.

37.1 Synchronizing servers

To synchronize manually the main and the node DROPS servers, click the synchronization icon in the **Main Configuration** view.

Synchronization process steps

The configuration synchronization is always requested by a node server that pulls information from the main server.

1. The node server emits a synchronization package generation to the main server
2. The main server:
   a. reads all the data included in the configuration synchronization scope
   b. generates a synchronization package
   c. informs the node that the package is ready
3. The node server:
   a. downloads the synchronization package
   b. decompresses it into the user-defined area
   c. reads the list of the synchronized items
   d. browses the exported data
   e. inserts or updates its own database
   f. generates a synchronization report and attaches to it the synchronization report database record
   g. informs the main server about the result of the synchronization

What is synchronized?

All the elements necessary to carry out the import and deployment processes from the node server (strategies, infrastructure items, process, etc.)

> ⚠ **Important!**
> **Synchronization Exceptions:** The configuration synchronization includes all the attribute's values of a synchronized item, except for server properties. Server properties are different from a server to another and their values vary according to a dedicated execution context. All the attributes of server properties are synchronized except the value to avoid overwriting the value defined at the node server level.

> ***Reference***
> Refer to Server properties on page 414.

37.2 Scheduling synchronization

To schedule the periodical execution of the synchronization, activate the option in the **Main Configuration** > **Main Server Settings**.

Activate the Synchronization

Tick the box to activate or deactivate the synchronization scheduler.

Synchronization Delay (min)

Define the amount of time between two successive synchronization executions (in minutes).

37.3 Synchronization reports

A report is generated for every synchronization. The list of synchronization reports are found in the **Main Configuration** > **Main Server Settings** view.

You can define how many synchronization reports you want to keep by modifying the **Max Report Count** value.

Max Report Count

Enter the number of reports you want to keep, from the most recently generated reports. At the end of each synchronization, the DROPS node server deletes all the extra reports to keep only the most recently generated reports.

Double-click on a synchronization report to view it. A synchronization report has 4 sections:

Header

This part shows the date of the synchronization.

Synchronization Reports

This part describes the synchronization result for each item.

Summary Execution

This part gives a summary of the global execution.

Log Entries

This part displays all the execution log entries.

Importing artifacts

Introduction to importing artifacts

There are two ways to import artifacts: initially via initial import strategies, or the "normal" way via regular import strategies. Both use the **Import artifacts** wizard which can be accessed by:

- importing via an application
- importing or reimporting via an existing import process instance

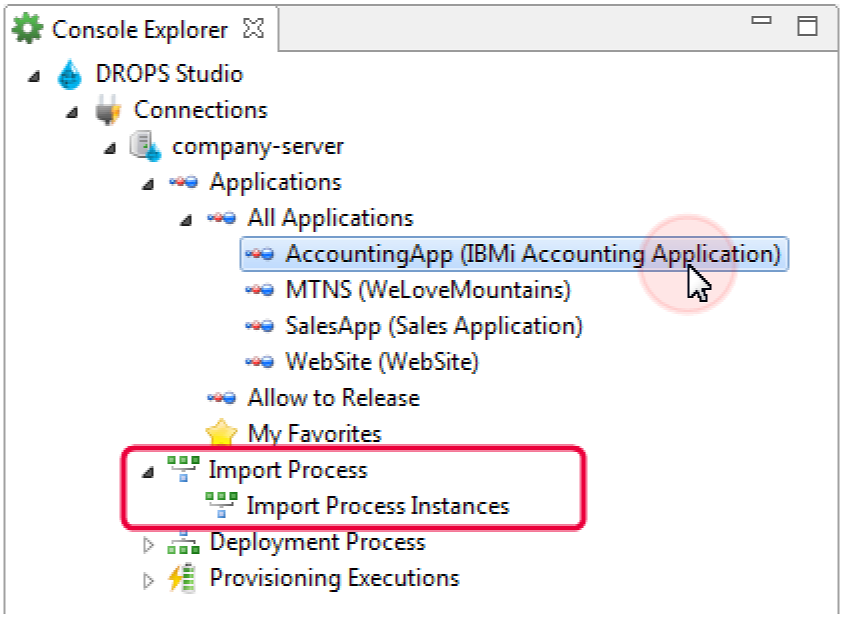


Figure 85: The Operational Console Management perspective - Importing

The following actions, involved in executing the import processes, are completed using the **Operational Console Management** perspective (see Overview of the DROPS Studio on page 34) or in the DROPS web app.

The **Console Explorer** provides access to the import process execution procedures. This perspective is the functional side of DROPS.

- Managing applications (see Managing applications on page 243),
- Importing artifacts (creating releases) (see Importing artifacts immediately by application on page 245 and Import process instances on page 248).

> **Note**
> All of the actions available in this view are also available in the DROPS web app.

38 Managing applications

Required Roles	Application Repository Manager and/or Import Operator
Required Rights (by category)	Import Process Instances
Perspective	Operational Console Management
Access	DROPS Server → Applications

Applications that are defined in the **Configuration Management** perspective are displayed in the **Operational Console Management** perspective under the **Applications** node in the **Console Explorer** menu. If you have the rights to import artifacts, use the applications listed in this menu.

Initially, all of the applications are only listed under **All Applications**.

38.1 Managing Allow-to-Release applications

This list displays all of the applications for which your user has the Import Operator or the Release Manager role. These are all of the applications for which you can carry out the import process.

If you are not the Import Operator or Release Manager for any application, this list will be empty and you cannot import content.

Reference

For more information about assigning roles to applications, refer to Assigning entity-level roles on page 54 or Roles in the appendix.

38.2 Managing favorite applications

Each user can add any defined application to his list of favorites to easily find the application(s) he uses most often.

Right-click on an application in the list and select **Add to favorites**. Refresh the **My Favorites** node to access it from that menu. Any application that is in your favorites is marked by this icon in the Operational Console perspective: . To remove an application from your favorites, right-click on it and select **Remove from favorites**.

38.3 Viewing import history by application

To view the import history for an application, right-click on the application in the **Console Explorer**, then select **Show Import History** or **Show Scheduled Import Processes**.

The **Import Processes** view is displayed giving you access to all of the imports launched for the selected application.

Reference

For more information about this view, refer to Import process instances on page 248.

39 Importing artifacts immediately by application

Required Roles	Application Repository Manager and/or Import Operator
Required Rights (by category)	Import Process Instances
Perspective	Operational Console Management
Access	DROPS Server → Applications → All Applications

When a "normal" import strategy is launched via the import process, you can filter through the component's available import strategies and import only those that correspond to a defined version. This means that if a version defined in the source provider only contains a few artifacts and that the rest of the artifacts have not changed since the last release was created, only the updated artifacts will be re-imported. This ensures that only the artifacts required to complete the new release version are taken into account.

Initial import strategies are all launched together when the **Initial Import** option is selected in the **Console Explorer**. An import strategy is considered 'initial' when the Initialization Strategy checkbox is ticked in the **Import Strategy** editor. Because all of the initial import strategies defined for an application are run at the same time, they do not require external version numbers, which means that all of their artifacts will be imported, regardless of their version.

Follow the subsequent steps to import artifacts via initial import strategies or by running "normal" import strategies immediately.

Step 1 Expand the **All Applications** node of the current server in the **Console Explorer**.

Step 2 Right-click on the application for which to import the artifacts then select either **Import Artifacts** or **Initial Import**.

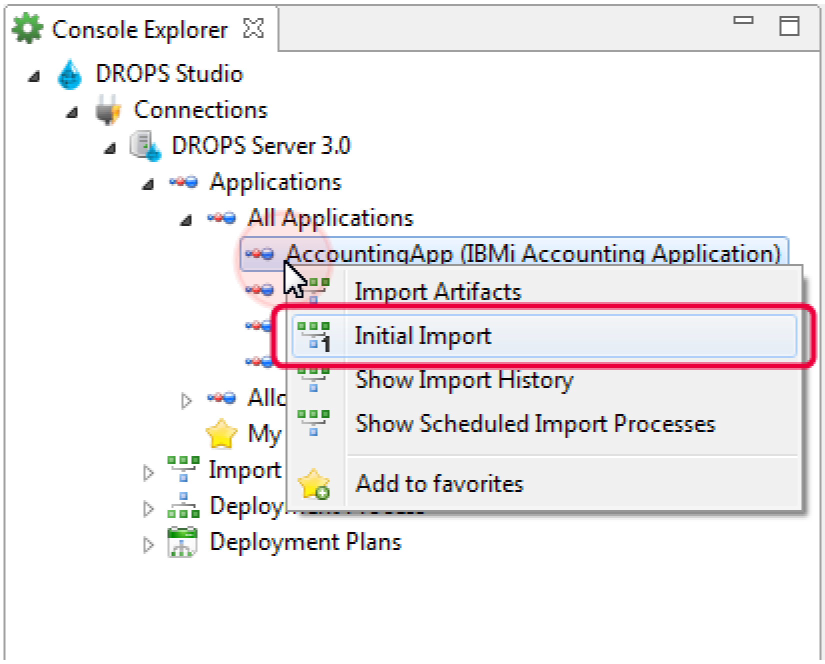


Figure 86: Importing artifacts by application

Step 3 Fill in the required configuration in the **Import Artifacts** wizard.

1. Select the release in which to save the artifacts that will be imported.
   - If **Current Release** is selected, the release defined in the Current Release field in the application is used to import (displayed in parentheses).
   - If **Create a New Release** is selected, a new release will be created using the Release Number Template defined in the application.

     If this option is selected, define the part of a release's version number to increment from the **Increment Part** drop-down menu. Keep in mind that the algorithm will update the other parts automatically. By default, the part is set to the value defined in the the Repository Storage Area settings in the DROPS Server configuration.
   - Selecting an **existing**, *active* release from the prepopulated list will import the artifacts to that release.
2. Define the import options.

   Enter a description for the release to better identify the contents.

   Tick the **Reset release before importing** checkbox to purge the selected release of all of its contents before importing the artifacts. The current import replaces the old content completely.

   Tick the **Use Deployment Process** option to execute a deployment process at the end of the import. Select a deployment process from the drop-down list to execute at the end of the import. The list displays all the processes defined for the selected application. At the end of the import, if it is successful, a deployment process instance is prepared and executed.

   > **Note**
   > The user who executes the import process instance must have the mandatory Rights to prepare and execute deployment process instances for this function to work.
3. If you selected to run a "normal" import strategy, the next page of the wizard enables you to select the import strategy(ies) to use to feed the selected release.

   Filtering the import strategies of the components in the application enables you to select different combinations of artifacts for this release.
   - Ticking **No Filter** will automatically import all of the application's artifacts.
   - To use an external file that defines which component(s) and import strategy(ies) to use, enter its name in the **External File Name** field. This can also be done in the **Application** editor in the Default Imported Version file name field.
   - Select which import strategies to use or not by ⊕ adding or ⊗ removing them from the **User-Defined Filter** section.

     Click the ✎ **Edit** button to edit an import strategy already in the list.

The list displays the import strategies used and their corresponding components as well as the version to import. The **External Version** refers by default to the ${import.version} substitution variable (***RELEASE**).

> **Important!**
> The import strategies selected in this section define the components that will be deployed when this release is launched.

Only "normal" import strategies are available in this list. Initial import strategies can only be launched by selecting to import **Initial Import Strategies**.

> ***Reference***
> For more information about strategies, refer to Import strategies on page 72 and Import strategy type configuration in the appendix.

Step 4 To associate the instance with a workitem, select the provider then the pre-defined workitem.

> ***Reference***
> For more information about JIRA issues and workitems in DROPS, refer to Workitems on page 401.

Step 5 Click **Finish** to launch the import.

Result The import instance is executed in the background.

When an instance's execution status is **Completed** and the result is **Succeeded** its icon changes to show that the import was successful ✓.

> ***Reference***
> For more information about import results, refer to Understanding import process instance results on page 254 and Imported artifacts on page 88.

40 Import process instances

Required Roles	Application Repository Manager and/or Import Operator
Required Rights (by category)	Import Process Instances
Perspective	Operational Console Management
Access	DROPS Server → Import Process → Import Process Instances

Chapter Summary

40.1 Searching for import process instances 248
40.2 Creating import process instances 250
40.3 Executing import process instances 253
40.4 Understanding import process instance results 254
40.5 Viewing associated workitems 255
40.6 Deleting import process instances 255

Import process instances are created automatically when an import is launched from the **All Applications** node. Instances can also be created from the **Import Processes** search view and either executed manually or scheduled in order to execute them at a later date or execute them regularly. Jenkins can also create import process instances if they are launched outside of the DROPS Studio.

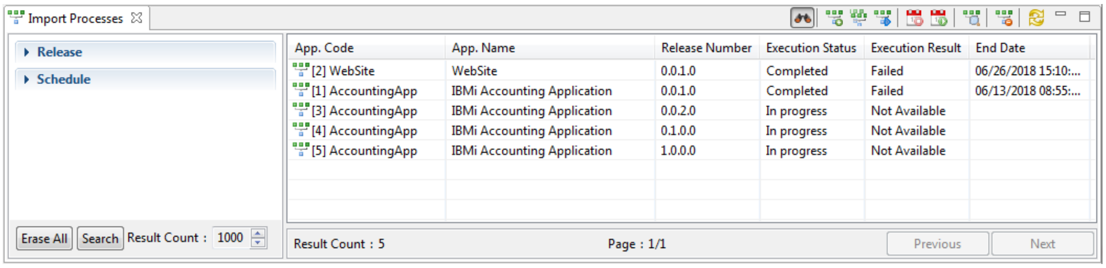
> ***Reference***
>
> For more information about importing with Jenkins, refer to DROPS for Jenkins on page 362.

Different import process instances can be scheduled to match your development team's deliveries. To manage when import strategies are executed, schedule when an import process instance should be carried out. Scheduling the import process is useful if you need to continuously import content that is regularly updated or if you know in advance when the next batch of content to import will be ready.

A dedicated job runs on the DROPS Server that regularly scans for scheduled import process instances, then triggers import strategies according to the schedule(s).

40.1 Searching for import process instances

Follow the subsequent steps to search for import process instances.



Result Count : 1000

App. Code	App. Name	Release Number	Execution Status	Execution Result	End Date
[2] WebSite	WebSite	0.0.1.0	Completed	Failed	06/26/2018 15:10:...
[1] AccountingApp	IBMi Accounting Application	0.0.1.0	Completed	Failed	06/13/2018 08:55:...
[3] AccountingApp	IBMi Accounting Application	0.0.2.0	In progress	Not Available	
[4] AccountingApp	IBMi Accounting Application	0.1.0.0	In progress	Not Available	
[5] AccountingApp	IBMi Accounting Application	1.0.0.0	In progress	Not Available	

Result Count : 5   Page : 1/1

Figure 87: The Import Process Instance search view

Step 1 Select **Import Process Instances** from the **Import Process** node in the **Console Explorer** or right-click on an application and select **Show Import History**.

Step 2 When the search view is opened, the list of instances is empty. Use the search criteria sections to filter the search.

- To search for the processes by **Application** or **Release**, select it by clicking the corresponding browse button.
- To search only for scheduled import process instances, tick the **Scheduled Only** checkbox.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.

Result The **Import Process** search view displays all of the finished and scheduled import process instances that meet the search criteria.

The **App. Code** column displays the instance's Technical ID in [brackets] as well as the name of the application that was used to import and one of the following icons:

- Prepared/Completed.
- Active scheduled instance.
- Paused scheduled instance.

The **App. Name** column displays the name of the application as defined.

The **Release Number** column displays the official version number of the release created by the import process instance.

The **Execution Status** and **Execution Result** columns display the current statuses of each instance and change while the instance is being processed.

Prepared	Prepared instances are ready to be executed.
Completed	Completed instances have already been executed.
In Progress	The progress bar is displayed while the instance is executing.
Waiting	If the import process is stuck and requires user action to complete. This possibility is only active for ARCAD imports and is displayed when the source IBM i is in **MSGW** status. Action must be taken on the IBM i side before the import process can complete.
Succeeded	If an instance executed successfully.
Failed	If any step in the import process failed to successfully execute.

Table 18: Import process instance statuses & results

The **End Date** column displays the date the instance finished importing.

40.2 Creating import process instances

There are three ways to create import process instances:

1. Launch an Import from an application to automatically create an instance that is displayed in the **Import Process** search view.

> ***Reference***
> For more information about launching the import process from an application, refer to Importing artifacts immediately by application on page 245.

2. Create new import process instances using the **Import Artifacts** wizard.
3. Duplicate an existing import process instance to quickly create an instance with the same parameters as another instance. Duplicating creates a copy of an existing instance that can be modified.

40.2.1 Create new import process instances

Follow the subsequent steps to create a new Import process manually.

Step 1 To open the **Import Artifacts** wizard, either click the Create icon in the **Import Process** search view or right-click anywhere in the view and select **Create a new import process instance**.

Step 2 Select the application that includes the artifacts to import.

Click **Next >** to continue.

Step 3 Select the release in which to save the artifacts that will be imported.

- If **Current Release** is selected, the release defined in the Current Release field in the application is used to import (displayed in parentheses).
- If **Create a New Release** is selected, a new release will be created using the Release Number Template defined in the application.

  If this option is selected, define the part of a release's version number to increment from the **Increment Part** drop-down menu. Keep in mind that the algorithm will update the other parts automatically. By default, the part is set to the value defined in the the Repository Storage Area settings in the DROPS Server configuration.

- Selecting an **existing**, *active* release from the prepopulated list will import the artifacts to that release.

Click **Next >** to continue.

Step 4 Define the import options.

Enter a description for the release to better identify the contents.

Tick the **Reset release before importing** checkbox to purge the selected release of all of its contents before importing the artifacts. The current import replaces the old content completely.

Tick the **Use Deployment Process** option to execute a deployment process at the end of the import. Select a deployment process from the drop-down list to execute at the end of the import. The list displays all the processes defined for the selected application. At the end of the import, if it is successful, a deployment process instance is prepared and executed.

> **Note**
> The user who executes the import process instance must have the mandatory Rights to prepare and execute deployment process instances for this function to work.

Click **Next >** to continue.

Step 5 To associate the instance with a workitem, select the provider then the pre-defined workitem.

> ***Reference***
> For more information about JIRA issues and workitems in DROPS, refer to Workitems on page 401.

Step 6 Select the import strategy(ies) to use to feed the selected release.

Filtering the import strategies of the components in the application enables you to select different combinations of artifacts for this release.

- Ticking **No Filter** will automatically import all of the application's artifacts.
- To use an external file that defines which component(s) and import strategy(ies) to use, enter its name in the **External File Name** field. This can also be done in the **Application** editor in the Default Imported Version file name field.
- Select which import strategies to use or not by ⊕ adding or ⊗ removing them from the **User-Defined Filter** section.

  Click the ✎ **Edit** button to edit an import strategy already in the list.

  The list displays the import strategies used and their corresponding components as well as the version to import. The **External Version** refers by default to the ${import.version} substitution variable (***RELEASE**).

  > **Important!**
  > The import strategies selected in this section define the components that will be deployed when this release is launched.

  Only "normal" import strategies are available in this list. Initial import strategies can only be launched by selecting to import **Initial Import Strategies**.

> ***Reference***
> For more information about strategies, refer to Import strategies on page 72 and Import strategy type configuration in the appendix.

Step 7 Schedule the import process.

- A schedule set to execute **Once** executes the import one time at a given time. The **Time** must be set to define the time to run the execution.
- A **Periodic** schedule executes the import process at regular time intervals. The **Interval** must be set, and the day(s) of the week as well, to define the time elapsed (in minutes) between two consecutive executions. The process will run every day that is selected every week.
- **Daily** import process instances will be run once on the **Day(s) of the week** selected, every week.
- **Monthly** import process instances will be run once on day **Day of the Month** defined, every month.

Step 8 Click **Finish**.

Result If the instance was scheduled to run immediately or wasn't scheduled at all, the import is launched.

If the instance was scheduled for a later date or time, it appears in the **Import Process** search view as a scheduled process and will launch automatically at the defined time.

When an instance's execution status is **Completed** and the result is **Succeeded** its icon changes to show that the import was successful .

> ***Reference***
> For more information about import results, refer to Understanding import process instance results on page 254 and Imported artifacts on page 88.

40.2.2 Duplicate an import process instance

Duplicating an instance is a quick way to create an identical instance that you can later modify or schedule differently.

To duplicate an existing import process instance, select it then either click the Duplicate icon, or right-click on it and select **Duplicate an import process instance**. The **Duplicate Import Process** wizard is prepopulated with most of the values defined for the original instance. When an import process instance is duplicated, the release will be reset by default before importing, even if this option was inactive in the original instance. You can modify any settings before clicking **Finish** to create the duplication instance.

> ⚠ **Important!**
> You cannot change the application of a duplicated import instance.

The duplication instance is automatically available in the **Import Process** search view and has its own unique Technical ID.

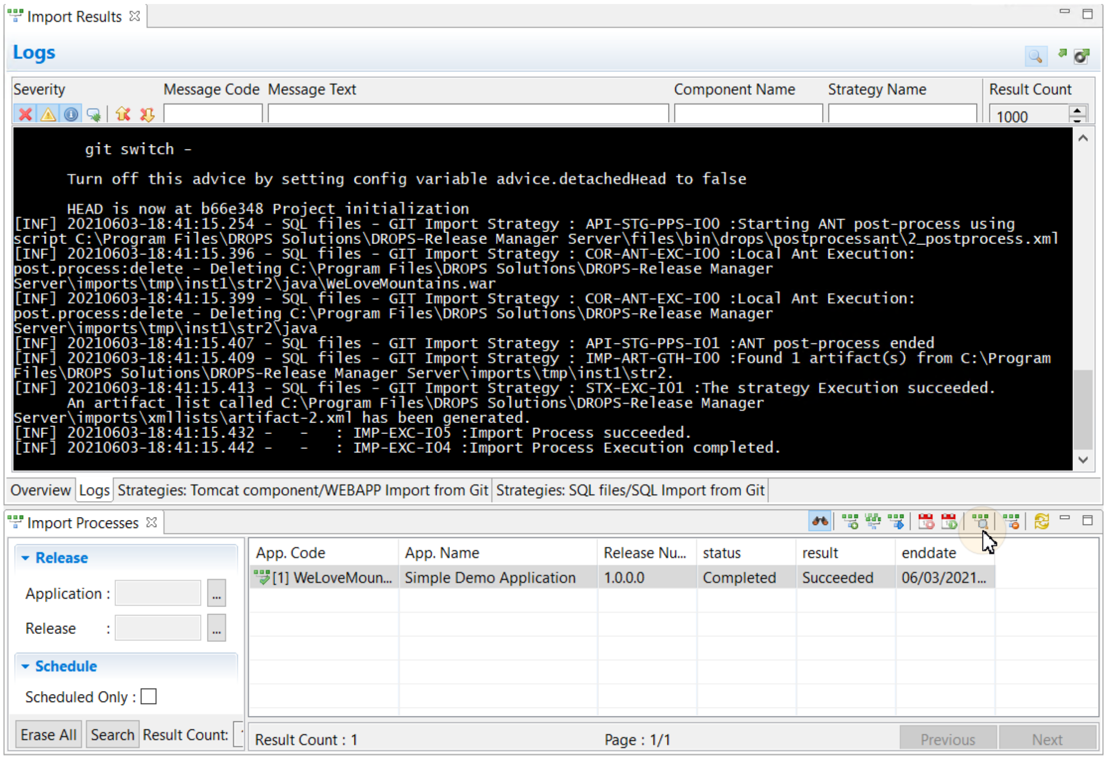
40.3 Executing import process instances

Executing an existing import process instance creates a new instance.

To execute an existing import process instance, either right-click then select **Execute the import process instance**, or click the execute import process instance icon in the view.

> ***Warning!***
> Executing a scheduled import process instance will unschedule it and import it immediately.

To view what happens during the execution in real-time, open the import instance's **Results** view and watch the import actions take place from the **Console** tab.



```
      git switch -

     Turn off this advice by setting config variable advice.detachedHead to false

     HEAD is now at b66e348 Project initialization
[INF] 20210603-18:41:15.254 - SQL files - GIT Import Strategy : API-STG-PPS-I00 :Starting ANT post-process using
script C:\Program Files\DROPS Solutions\DROPS-Release Manager Server\files\bin\drops\postprocessant\2_postprocess.xml
[INF] 20210603-18:41:15.396 - SQL files - GIT Import Strategy : COR-ANT-EXC-I00 :Local Ant Execution:
post.process:delete - Deleting C:\Program Files\DROPS Solutions\DROPS-Release Manager
Server\imports\tmp\inst1\str2\java\WeLoveMountains.war
[INF] 20210603-18:41:15.399 - SQL files - GIT Import Strategy : COR-ANT-EXC-I00 :Local Ant Execution:
post.process:delete - Deleting C:\Program Files\DROPS Solutions\DROPS-Release Manager
Server\imports\tmp\inst1\str2\java
[INF] 20210603-18:41:15.407 - SQL files - GIT Import Strategy : API-STG-PPS-I01 :ANT post-process ended
[INF] 20210603-18:41:15.409 - SQL files - GIT Import Strategy : IMP-ART-GTH-I00 :Found 1 artifact(s) from C:\Program
Files\DROPS Solutions\DROPS-Release Manager Server\imports\tmp\inst1\str2.
[INF] 20210603-18:41:15.413 - SQL files - GIT Import Strategy : STX-EXC-I01 :The strategy Execution succeeded.
     An artifact list called C:\Program Files\DROPS Solutions\DROPS-Release Manager
Server\imports\xmllists\artifact-2.xml has been generated.
[INF] 20210603-18:41:15.432 -    -    : IMP-EXC-I05 :Import Process succeeded.
[INF] 20210603-18:41:15.442 -    -    : IMP-EXC-I04 :Import Process Execution completed.
```

App. Code	App. Name	Release Nu...	status	result	enddate
[1] WeLoveMoun...	Simple Demo Application	1.0.0.0	Completed	Succeeded	06/03/2021...

Result Count : 1 Page : 1/1

Figure 88: Consulting Import execution results in real-time (Console)

> **Note**
> The Console tab is only available in **Prepared** instances or those which are currently In Progress. If an instance has finished executing the Console is no longer available but the log can be accessed in the **Logs** tab.

40.3.1 Pause scheduled import process instances

Scheduled import process instances will launch automatically when the defined day and time arrive.

- To pause a scheduled import process instance, select it and either right-click then select **Stop Schedule**, or click the stop schedule icon in the toolbar.

The Icon in the search view changes from an active scheduled instance, to paused.

- Re restart a paused scheduled import process instance, select it and either right-click then select **Restart Schedule**, or click the restart schedule icon in the toolbar.

40.4 Understanding import process instance results

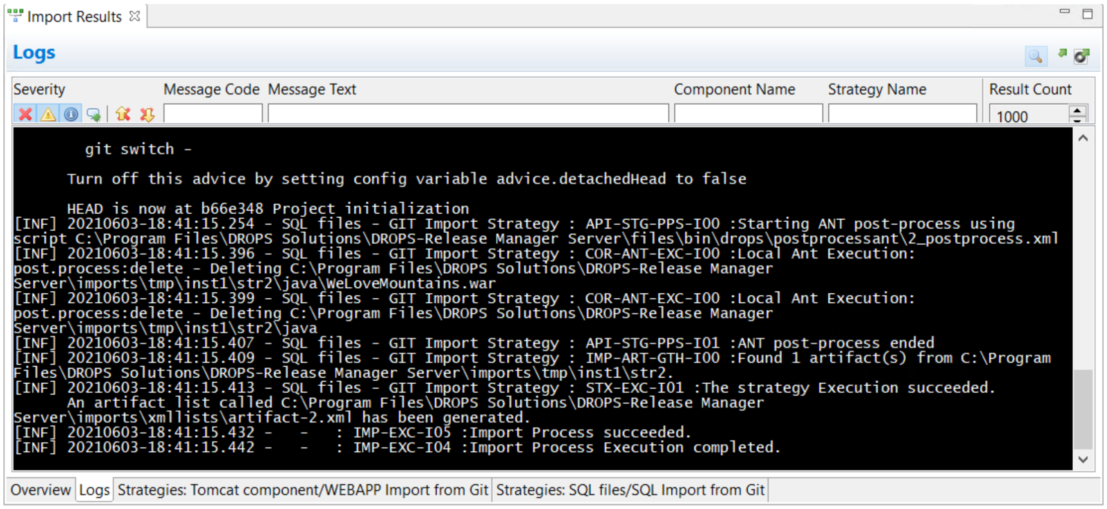
The **Import Process Instances** search view provides some details of processed import process instances. Access more information by either double-clicking on instance in the list, selecting it then clicking the Results icon in the view or right-clicking then selecting **Show Import Results**.

The **Import Results** view contains multiple tabs that each contain different elements of the import.

The **Overview** tab displays the uneditable general information about the instance. The scheduling information for the instance can be edited here. The **End Date** is left blank until the instance is executed.

40.4.1 View the Logs

All of the activity from the import is displayed in the **Logs** tab. Every action carried out during a deployment is logged.



```
        git switch -

      Turn off this advice by setting config variable advice.detachedHead to false

      HEAD is now at b66e348 Project initialization
[INF] 20210603-18:41:15.254 - SQL files - GIT Import Strategy : API-STG-PPS-I00 :Starting ANT post-process using
script C:\Program Files\DROPS Solutions\DROPS-Release Manager Server\files\bin\drops\postprocessant\2_postprocess.xml
[INF] 20210603-18:41:15.396 - SQL files - GIT Import Strategy : COR-ANT-EXC-I00 :Local Ant Execution:
post.process:delete - Deleting C:\Program Files\DROPS Solutions\DROPS-Release Manager
Server\imports\tmp\inst1\str2\java\WeLoveMountains.war
[INF] 20210603-18:41:15.399 - SQL files - GIT Import Strategy : COR-ANT-EXC-I00 :Local Ant Execution:
post.process:delete - Deleting C:\Program Files\DROPS Solutions\DROPS-Release Manager
Server\imports\tmp\inst1\str2\java
[INF] 20210603-18:41:15.407 - SQL files - GIT Import Strategy : API-STG-PPS-I01 :ANT post-process ended
[INF] 20210603-18:41:15.409 - SQL files - GIT Import Strategy : IMP-ART-GTH-I00 :Found 1 artifact(s) from C:\Program
Files\DROPS Solutions\DROPS-Release Manager Server\imports\tmp\inst1\str2.
[INF] 20210603-18:41:15.413 - SQL files - GIT Import Strategy : STX-EXC-I01 :The strategy Execution succeeded.
      An artifact list called C:\Program Files\DROPS Solutions\DROPS-Release Manager
Server\imports\xmllists\artifact-2.xml has been generated.
[INF] 20210603-18:41:15.432 -    -    : IMP-EXC-I05 :Import Process succeeded.
[INF] 20210603-18:41:15.442 -    -    : IMP-EXC-I04 :Import Process Execution completed.
```

Figure 89: The Import Results view - Logs

Filter results by **Message Code**, **Message Text**, **Component Name**, or **Strategy Name**. Enter any combination of search criteria then click the search icon to display the results. To display the complete log, click Search without entering any criteria.

Filter the severity of the results displayed by selecting one or more of the **Severity** icons

- ✕ **Fatal**: only fatal errors are displayed.
- ⚠ **Warning**: only Warnings errors are displayed.

- **Info**: only Important information is displayed.
- **Verbose**: only details are displayed.

Use the Show Previous and Next icons to quickly move through a long list of entries to find and jump to the fatal errors.

Click the export icon to open the selected log(s) in an external text editor. By default, the editor will open automatically.

Click the export to JSON file icon to open the selected log(s) in .json format. You will be prompted to select where to save the file.

40.4.2 View the results for each import strategy

Each import strategy carried out during the import process is represented by its own tab. Each of these tabs describe the strategy's status and enables you to see the list of imported artifacts (the artifact list) by clicking the **Display the artifact list** link.

Clicking this link opens the **List Editor** which lists each artifact included in the import process instance. All of the relavent information is listed for each artifact in the table, including the source path and the path to the artifact on the DROPS Server.

40.5 Viewing associated workitems

If the import process was associated with a workitem, a link to the JIRA issue is included in the **Workitem** section. Click the link to open the issue in your default browser.

40.6 Deleting import process instances

> ***Warning!***
> Deleted import process instances cannot be accessed or recovered. Deleting an instance does not delete the artifacts that have already been imported by it. Only the trace of the import is removed.

To delete an import process instance search for it in the **Import Process** search view. Either right-click on it, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the instance.

Deploying releases

Introduction to deploying releases

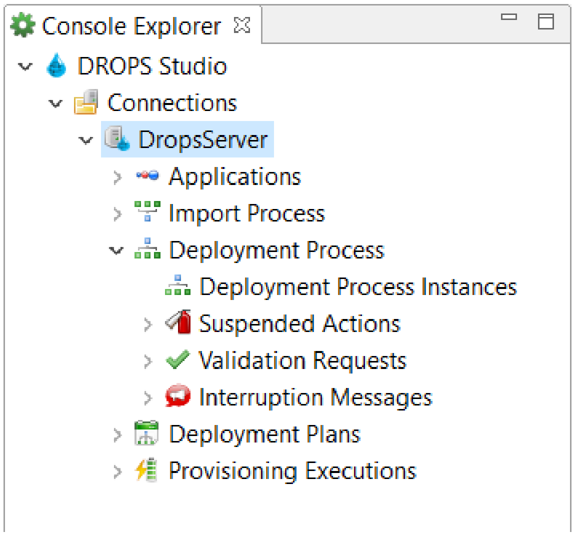
Releases are deployed to your target environments by executing **deployment process instances** to run the actions defined in deployment process diagrams.

There are two ways to deploy releases to your target environments:

1. by executing **deployment process instances** to run the actions defined in deployment process diagrams
2. by executing **deployment plan instances** to run through the stages defined in deployment plans, which include multiple deployment processes.

There are two safeguards built into this process:

1. All actions can be suspended if an error occurs and users can rectify the issues before re-executing the process.
2. When authorization is required users can manage validation requests to approve actions/stages/processes.



Console Explorer
- DROPS Studio
  - Connections
    - DropsServer
      - Applications
      - Import Process
      - Deployment Process
        - Deployment Process Instances
        - Suspended Actions
        - Validation Requests
        - Interruption Messages
      - Deployment Plans
      - Provisioning Executions

Figure 90: The Operational Console Management perspective - Deploying

The following actions, involved in executing the deployment processes, are all completed using the **Operational Console Management** perspective (see the Overview of the DROPS Studio on page 34) or in the DROPS web app.

The **Console Explorer** provides access to the deployment process execution procedures. This perspective is the functional side of DROPS.

- Deploying releases (see Deployment process instances on page 259),
- Addressing suspended actions (see Suspended actions on page 294),
- Addressing validation requests (see Validation requests on page 296),
- Addressing interruption messages (see Interruption messages on page 291),
- Deployment plan instances (see Deployment plan instances on page 284).

Note

All of the actions available in this view are also available in the DROPS web app.

41 Deployment process instances

Required Roles	Environment Release Manager and/or Release Operator
Required Rights (by category)	Deployment process Instances
Perspective	Operational Console Management
Access	DROPS Server → Deployment Process → Deployment Process Instances

Chapter Summary

41.1 Searching for deployment process instances 259
41.2 Creating deployment process instances 262
41.3 Changing the release to deploy 268
41.4 Managing deployment tags 269
41.5 Defining an instance's deployment mode 270
41.6 Executing deployment process instances manually 272
41.7 Scheduling deployment process instances 273
41.8 Remote deployments on node servers 276
41.9 Understanding deployment process instance results 277
41.10 Viewing associated workitems 281
41.11 Re-executing failed deployments 281
41.12 Rolling back a deployed deployment process instance 282
41.13 Canceling deployment process instances 283
41.14 Deleting deployment process instances 283

Deployment process instances enable you to deploy one application to one target environment. Instances can be created from the **Deployment Process Instances** search view and either executed manually or scheduled in order to execute them at a later date or even to execute them regularly.

> ***Warning!***
> If the DROPS Server is stopped during a deployment, the deployment process instance is canceled. You cannot continue the deployment once the server is started again. In order to finish deploying, duplicate the instance and execute again.

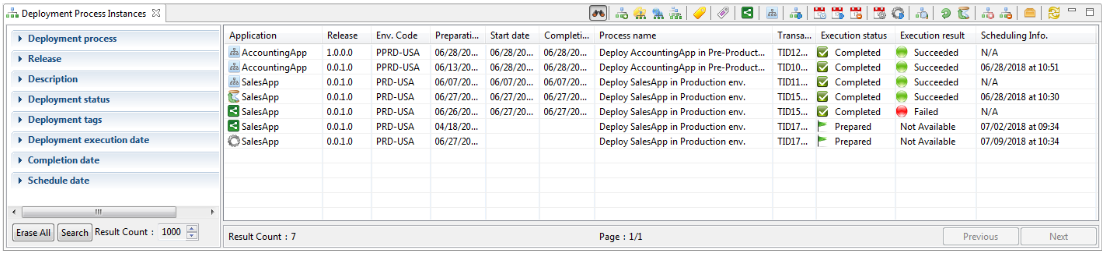
Jenkins can also create deployment process instances if they are launched outside of the DROPS Studio.

> ***Reference***
> For more information about deploying with Jenkins, refer to DROPS for Jenkins on page 362.

41.1 Searching for deployment process instances

Follow the subsequent steps to search for deployment process instances in the instances view.

Step 1 Select **Deployment Process Instances** from the **Deployment Process** node in the **Console Explorer**.



Deployment Process Instances

Result Count : 1000

Application	Release	Env. Code	Preparati...	Start date	Completi...	Process name	Transa...	Execution status	Execution result	Scheduling Info.
AccountingApp	1.0.0.0	PPRD-USA	06/28/20...	06/28/20...	06/28/20...	Deploy AccountingApp in Pre-Product...	TID12...	Completed	Succeeded	N/A
AccountingApp	0.0.1.0	PPRD-USA	06/13/20...	06/28/20...	06/28/20...	Deploy AccountingApp in Pre-Product...	TID10...	Completed	Succeeded	06/28/2018 at 10:51
SalesApp	0.0.1.0	PRD-USA	06/07/20...	06/07/20...	06/07/20...	Deploy SalesApp in Production env.	TID11...	Completed	Succeeded	N/A
SalesApp	0.0.1.0	PRD-USA	06/27/20...	06/27/20...	06/27/20...	Deploy SalesApp in Production env.	TID15...	Completed	Succeeded	06/28/2018 at 10:30
SalesApp	0.0.1.0	PRD-USA	06/26/20...	06/27/20...	06/27/20...	Deploy SalesApp in Production env.	TID15...	Completed	Failed	N/A
SalesApp	0.0.1.0	PRD-USA	04/18/20...			Deploy SalesApp in Production env.	TID17...	Prepared	Not Available	07/02/2018 at 09:34
SalesApp	0.0.1.0	PRD-USA	06/27/20...			Deploy SalesApp in Production env.	TID17...	Prepared	Not Available	07/09/2018 at 10:34

Result Count : 7

Page : 1/1

Figure 91: The Deployment Process Instances search view

Step 2 When the search view is opened, the list of instances is empty. Use the search criteria sections to filter the search.

- To search for an existing instance by the deployment process it uses to deploy, expand the **Deployment Process** section.
  Click the corresponding … browse button to select the associated client, environment and/or deployment process.

> **Important!**
> If the **Not completed** checkbox is ticked [*default*], only deployment instances that are not completed or not canceled are displayed in the search results.

> **Important!**
> If the **Not linked to a plan** checkbox is ticked [*default*], only deployment instances created manually are displayed in the search results.

- To search for an existing instance by the release it deploys, expand the **Release** section. Click the corresponding … browse button to select the associated application and/or release.
- To search for an existing instance by the description of either its environment or release, expand the **Description** section then enter part of the corresponding description.
- To search for an existing instance by its deployment or release status, expand the **Deployment Status** section then select a predefined status.
- To search for an existing instance by the deployment instance tag it belongs to, expand the **Deployment Tag** section then select the tag from the drop-down list.

> ***Reference***
> For more information about tags, refer to Managing deployment tags on page 269

- To search for an existing instance by the date it was or should be executed, expand the **Deployment Execution Date** section. Tick the Start and/or End Date, then select the range using the calendar.

- To search for an existing instance by the date it finished, expand the **Completion Date** section. Tick the Start and/or End Date, then select the range using the calendar.
- To search for an existing instance by its scheduled deployment date, expand the **Schedule Date** section. Tick the Start and/or End Date, then select the range using the calendar.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.

Result The **Deployment Process Instances** search view displays all of the finished and scheduled instances that meet the search criteria.

The **Application** column displays the name of the application deployed as well as one of the following icons which describes the mode in which the instance is set to deploy.

Full-transfer	Defining an instance's deployment mode on page 270
Transfer-only	
Installation-only	
Rolled back	Rolling back a deployed deployment process instance on page 282
Re-execute	Re-executing failed deployments on page 281

Table 19: Deployment process instance modes

The **Release** column displays the official version number of the release deployed.

The **Env Code** column displays the name of the environment to which the content was deployed. If an instance is linked to a deleted deployment process, this information is not available (**N/A**).

The **Preparation**, **Start** and **Completion Date** columns display these respective dates.

The **Process Name** column displays the name of the deployment process used to deploy.

The **Transaction ID** column displays the automatically-generated ID of the deployment.

The **Execution Status** and **Execution Result** columns display the current deployment status and change while the instance is being processed.

Prepared	Prepared instances are ready to be executed or scheduled.
Completed	Completed instances have already been executed and cannot be scheduled.
In Progress	The progress bar is displayed while the instance is executing.

Table 20: Deployment process instance statuses & results

Suspended	Suspended deployment process instances are highlighted in red to bring attention to the fact that an action is required before the instance can continue to deploy. Deployment process instances are suspended when either a ✓ validation request is required, the execution was suspended (likely because an infrastructure item cannot be reached), or the execution was interrupted (an interruption message was sent by the DROPS Agent).
✖ Canceled	If an instance is canceled.
● Succeeded	If an instance executed all of its actions successfully.
● Succeeded with Warnings	If an action specified *not* to **Fail on Error** encountered a problem during the execution.
● Failed	If any action in the deployment process failed to successfully execute.

Table 20: Deployment process instance statuses & results

The **Scheduling Info.** column displays the date and time the instance is scheduled to deploy, if configured.

If an installation-only instance has been auto-scheduled, the label will be in one of the following formats:

- *AUTO [_+<delay_value>_] if no fixed time has been chosen
- *AUTO [_+<delay_value>_] [_<hour>:<minute>_] if a fixed time has been chosen

41.2 Creating deployment process instances

There are multiple ways to create deployment process instances in the instances view:

1. Create a deployment process instance using the **Create a new deployment process instance** wizard.
2. Create one instance to use to deploy a set of releases to the same environment using the **Create a new deployment process instance for a Release Set** wizard.
3. Create multiple instances to deploy one release to a set of environments using the **Create a new deployment process instance from environment group** wizard.
4. Duplicate an existing instance to quickly create an instance with the same parameters as the original. Duplicating creates a copy of an existing instance that can be modified.

Note

By default, new instances are always marked to be deployed in full-transfer mode, however they are *not* executed automatically and must be executed manually or scheduled to execute automatically.

Reference

For more information about changing the mode in which to deploy an instance, refer toDefining an instance's deployment mode on page 270.

For more information about executing instances, refer toExecuting deployment process instances manually on page 272.

41.2.1 Create a deployment process instance

A single deployment process instance allows you to deploy one release to a single environment.

Note
Creating a new single instance enables you to duplicate it later.

Follow the subsequent steps to create a new deployment process instance in the instances view.

Important!
Be careful selecting the properties of a deployment instance! Most of them cannot be changed once the instance is created.

Step 1 To open the **Create a new deployment process instance** wizard, either click the create icon in the **Deployment Process Instances** search view or right-click anywhere in the view and select **Create a new deployment process instance**.

Step 2 Define the instance by clicking the browse button for each of the following entities.

1. If multiple clients are available, select the **client** to which to deploy the release. Changing the client alters the options available throughout the wizard. Only the entities available for the selected client will be accessible.
2. Select the **environment** to which to deploy the release. The list is prepopulated with the environments available for the selected client. Changing the environment alters the options available throughout the wizard. Only the entities available for the selected environment will be accessible.
3. Select the **deployment process** to use. The list is prepopulated with the processes available for the selected environment.
4. Select the **Release** to deploy. The list is prepopulated with the releases available for the selected client.
5. Modify the template for the **Transaction ID**, if required.

   The ID's syntax follows the template managed in the **Preferences** menu.

   Reference
   For more information about the default Transaction ID template, refer to the *DROPS Configuration Guide*.

Click **Next >** to continue.

Step 3 If required, or useful, select an existing deployment tag to regroup the instance with other similar instances.

Note
Tags can be managed here or in the **Preferences** menu.

Warning!
Deleting tags removes them from all of the instances to which they were linked.

Reference
For more information about deployment tags, refer to Managing deployment tags on page 269.

Step 4 To associate the instance with a workitem, select the provider then the pre-defined workitem.

Reference
For more information about JIRA issues and workitems in DROPS, refer to Workitems on page 401.

Click **Finish**.

Result The new deployment process instance is available in the **Deployment Process Instances** search view.

41.2.2 Create an instance to deploy a release set to an environment

This option creates a single deployment process instance that includes multiple releases to be deployed to a single environment.

During the deployment, the release set is processed just like a CUMULATIVE deployment: DROPS will first go through each release of the set and keep the latest version of each artifacts to build its artifacts list.

Sets of releases are parsed to only include the most recent version of an artifact, if the same artifact is included in multiple releases. The release with the most recent version of the artifact will include it during deployment and the releases with outdated versions of that artifacts will not.

Follow the subsequent steps to create an instance that will deploy a set of multiple releases.

Important!
Be careful selecting the properties of a deployment instance! Most of them cannot be changed once the instance is created.

Step 1 To open the **Create a new deployment process instance for a release set** wizard, either click the create for release set icon in the **Deployment Process Instances** search view or right-click anywhere in the view and select **Create for release set**.

Step 2 Define the main properties of all the instances by clicking the browse button for each of the following entities.

1. If multiple clients are available, select the **client** to which to deploy the releases. Changing the client alters the options available throughout the wizard. Only the entities available for the selected client will be accessible.
2. Select the **environment** to which to deploy the releases. The list is prepopulated with the environments available for the selected client. Changing the environment alters the options available throughout the wizard. Only the entities available for the selected environment will be accessible.
3. Select the **deployment process** to use. The list is prepopulated with the processes available for the selected environment.
4. Modify the template for the **Transaction ID**, if required.

   The ID's syntax follows the template managed in the **Preferences** menu.

   Reference
   For more information about the default Transaction ID template, refer to the *DROPS Configuration Guide*.

Click **Next >** to continue.

Step 3 Ctrl+click to select multiple releases to deploy. The list is prepopulated with the releases available for the selected client.

Click **Next >** to continue.

Step 4 If required, or useful, select an existing deployment tag to regroup the instance with other similar instances.

Note
Tags can be managed here or in the **Preferences** menu.

Warning!
Deleting tags removes them from all of the instances to which they were linked.

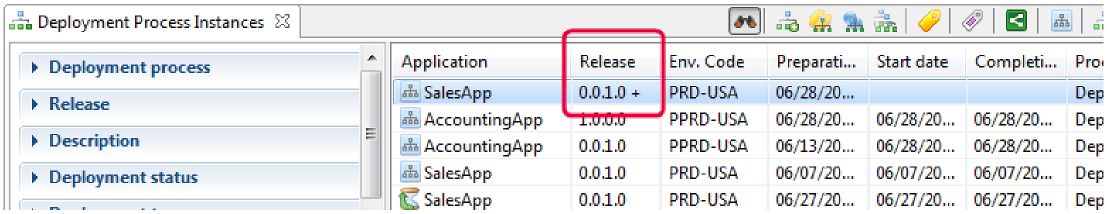
Reference
For more information about deployment tags, refer to Managing deployment tags on page 269.

Step 5 To associate the instance with a workitem, select the provider then the pre-defined workitem.

Reference
For more information about JIRA issues and workitems in DROPS, refer to Workitems on page 401.

Click **Finish**.

Result The new deployment process instance is available in the **Deployment Process Instances** search view. Instances created for release sets are indicated by a plus sign in the **Release** column indicating that more than one release is to be included in the deployment.



Application	Release	Env. Code	Preparati...	Start date	Completi...	Pro
SalesApp	0.0.1.0 +	PRD-USA	06/28/20...			Dep
AccountingApp	1.0.0.0	PPRD-USA	06/28/20...	06/28/20...	06/28/20...	Dep
AccountingApp	0.0.1.0	PPRD-USA	06/13/20...	06/28/20...	06/28/20...	Dep
SalesApp	0.0.1.0	PRD-USA	06/07/20...	06/07/20...	06/07/20...	Dep
SalesApp	0.0.1.0	PRD-USA	06/27/20...	06/27/20...	06/27/20...	Dep

Figure 92: Deployment process instances for release sets

41.2.3 Create multiple instances to deploy one release to an environment group

Contrary to the release set option above, this option creates multiple individual deployment process instances that each deploy the same release to multiple environments that are defined in an environment group.

Follow the subsequent steps to create multiple instances using an environment group.

> ⚠ **Important!**
> Be careful selecting the properties of a deployment instance! Most of them cannot be changed once the instance is created.

Step 1 To open the **Create a new deployment process instance from environment group** wizard, either click the create from environment group icon in the **Deployment Process Instances** search view or right-click anywhere in the view and select **Create from environment group**.

Step 2 Define the main properties of all the instances by clicking the … browse button for each of the following entities.

1. If multiple clients are available, select the **client** to which to deploy the release. Changing the client alters the options available throughout the wizard. Only the entities available for the selected client will be accessible.
2. Select the **environment group** that includes the environments to which to deploy the release. The list is prepopulated with the environment groups available for the selected client. Changing the environment group alters the options available throughout the wizard. Only the entities available for the selected group will be accessible.
3. Select the **application** to deploy. This list is prepopulated with all of the applications that the environments in the group share.

> ℹ **Note**
> The deployment processes used to generate multiple instances are the process(es) that concern the selected application/release defined as default for the environment(s) in the environment group. If no default process is defined for the environment/application pair, the option to choose the

> deployment process to use is displayed.
>
> The list of processes to select is prepopulated with the processes available for each environment/application pair in the Group.

4. Select the **Release** to deploy. The list is prepopulated with the releases available for the selected client.
5. Modify the template for the **Transaction ID**, if required.

   The ID's syntax follows the template managed in the **Preferences** menu.

   > ***Reference***
   > For more information about the default Transaction ID template, refer to the *DROPS Configuration Guide*.

Click **Next >** to continue.

Step 3 To **Generate transfer-only and installation-only instances**, leave the corresponding checkbox ticked [*default*]. If the check box is not selected, full-transfer instances are created.

> **Note**
> If both transfer- and installation-only instances are created, the installation-only instance can be **automatically scheduled** to deploy.

> ***Reference***
> For more information about setting the automatic schedule for installation instances, refer to the *DROPS Configuration Guide*.

Step 4 If required, or useful, select an existing deployment tag to regroup the instance with other similar instances.

> **Note**
> Tags can be managed here or in the **Preferences** menu.

> ***Warning!***
> Deleting tags removes them from all of the instances to which they were linked.

> ***Reference***
> For more information about deployment tags, refer to Managing deployment tags on page 269.

Click **Finish**.

Step 5 If no default deployment process is defined for an environment included in the environment group selected, the **Process Selection** dialog is displayed. Select the deployment process to use to deploy the selected release. The list is prepopulated with the deployment processes available for the selected environment/application pair(s).

> ***Reference***
> For more information about default processes, refer toDefining default deployment processes on page 164.

Step 6 To associate the instance with a workitem, select the provider then the pre-defined workitem.

> ***Reference***
> For more information about JIRA issues and workitems in DROPS, refer to Workitems on page 401.

Result The new deployment process instances are available in the **Deployment Process Instances** search view.

41.2.4 Duplicate a deployment process instance

Duplicating an instance is a quick way to create an identical instance, with the same tags, that can later be scheduled differently or to which you can add different attachments.

To duplicate an existing deployment process instance, select it then either click the duplicate icon, or right-click on it and select **Duplicate a deployment process instance**. The duplicated instance is automatically available in the **Deployment process Instances** search view and has its own Transaction ID.

> **Note**
> Duplicating a transfer-only or an installation-only instance creates a full-transfer instance.

41.3 Changing the release to deploy

Scheduling future deployments is often very important but the exact release to deploy at a scheduled time isn't always defined until shortly before the deployment is to be executed. Defining an arbitrary release in order to schedule a deployment instance, then changing the release before deployment can be helpful to schedule multiple future releases well in advance.

> **Note**
> Changing the release that a deployment instance deploys is only possible if the instance's execution status is **Prepared** and if it is in full- or transfer-only mode.
>
> If the release is changed for a parent instance, the same change is also carried over to all of its scheduled child installation-only and rollback instances.

Follow the subsequent steps to change a deployment process instance's release in the instances view.

Step 1 To open the **Select a release** dialog, select the instance in the **Deployment process Instances** search view then either click the change release icon, or right-click on it and select **Change release**.

Step 2 Select the new release to deploy then click **OK** or double-click it.

Result The **Release** column in the **Deployment Process Instances** search view is updated with the new release to deploy.

41.4 Managing deployment tags

Deployment tags are intended to regroup similar deployment process instances and search for these instances by their tag. If an instance is duplicated, its tag(s) are carried over to the new instance, too.

Tags may be required for every deployment instance, or optional. If they are required, deployment process instances must be tagged when they are created. If they are optional, instances can be tagged when they are created or afterwords.

Tags can always be removed after an instance is created.

The **Deployment Tags** dialog is used to tag one or more instances and manage the tags themselves after instances are created. Tags can be edited at any time. To access the **Deployment Tags** dialog, select one or more deployment process instance(s) in the search view then either click the deployment tags icon or right-click then select **Manage Deployment Tags**.

To tag the selected instance(s), select the tag(s) from the list and click **OK**. When an instance is tagged, that tag is highlighted.

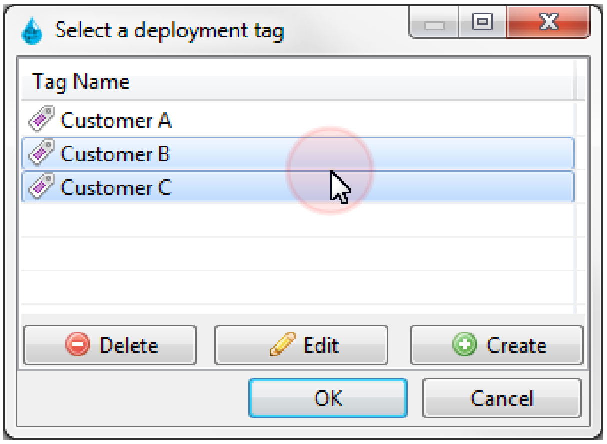


Figure 93: Tagged deployment process instances

To remove a tag from the selected instance(s), unselect the highlighted tag(s) [Ctrl+click], then click **OK**.

Note
Tags can be managed in the Preference menu and in each tag selection dialog when working with deployment process instances.

Reference

For more information about managing tags from the Preferences menu and about making tags required, refer to the *DROPS Configuration Guide*.

To create a new tag click the **Create** button. Enter the name of the new tag then click **OK**.

To edit the name of an existing tag, double-click on it or select it then click the **Edit** button. Modify the name of the tag, then click **OK**.

To delete an existing tag, select it then click the **Delete** button.

Warning!

Deleted deployment tags cannot be accessed or recovered.

Deleting a tag does not delete any of the instances that may be linked to it. The link between the tag and the instance is broken but the instance's content is not modified.

41.5 Defining an instance's deployment mode

Deployment instances can be carried out in one of three modes: full-transfer, transfer-only or installation-only mode. If an instance has not been executed, you can change the mode to deploy it in at any time.

41.5.1 Mark an instance to deploy in full-transfer mode

Deployment process instances that are marked to be deployed in full-transfer mode deploy all of the stages included in the deployment process diagram.

Note

By default, all new deployment process instances are set to deploy in full-transfer mode.

Note

If a transfer-only instance already has an installation-only child instance, the child instance is not deleted. It is recommended to delete the child instance so that the installation does not run twice.

To mark a transfer-only or installation-only deployment process instance to run in full-transfer mode, either right-click on the instance in the **Deployment process Instances** search view then select **Mark as full-transfer** or select it, then click the full-transfer icon.

Result The instance's icon changes to in the **Application** column.

Reference

For more information about the different Stages of the deployment process, refer to Manage deployment process stages on page 170.

For more information about executing a full-transfer deployment, refer to Executing deployment process instances manually on the next page or Scheduling deployment process instances on page 273.

41.5.2 Mark an instance to deploy in transfer-only mode

Deployment process instances that are marked to be deployed in transfer-only mode stop deploying after the **receiving stage** is complete. If an **installation stage** is included in the deployment process it is not carried out.

To mark a deployment process instance to run in transfer-only mode, either right-click on the instance in the **Deployment process Instances** search view then select **Mark as transfer-only** or select it, then click the transfer-only icon.

Result The instance's icon changes to in the **Application** column.

Reference

For more information about the different Stages of the deployment process, refer to Manage deployment process stages on page 170.

For more information about executing a transfer-only deployment, refer to Executing deployment process instances manually on the next page or Scheduling deployment process instances on page 273.

41.5.3 Create installation-only instances

Installation-only deployment process instances are created from transfer-only instances that contain an installation stage. An instance in installation-only mode is linked to its transfer-only parent and has the exact same Transaction ID.

Example

If the first stages in a deployment process have already been carried out in transfer-only mode and any actions required for the transferred artifacts are complete, the deployment is ready to be finalized manually which can only be done by creating an installation-only instance.

Reference

For more information about the different Stages of the deployment process, refer to Manage deployment process stages on page 170.

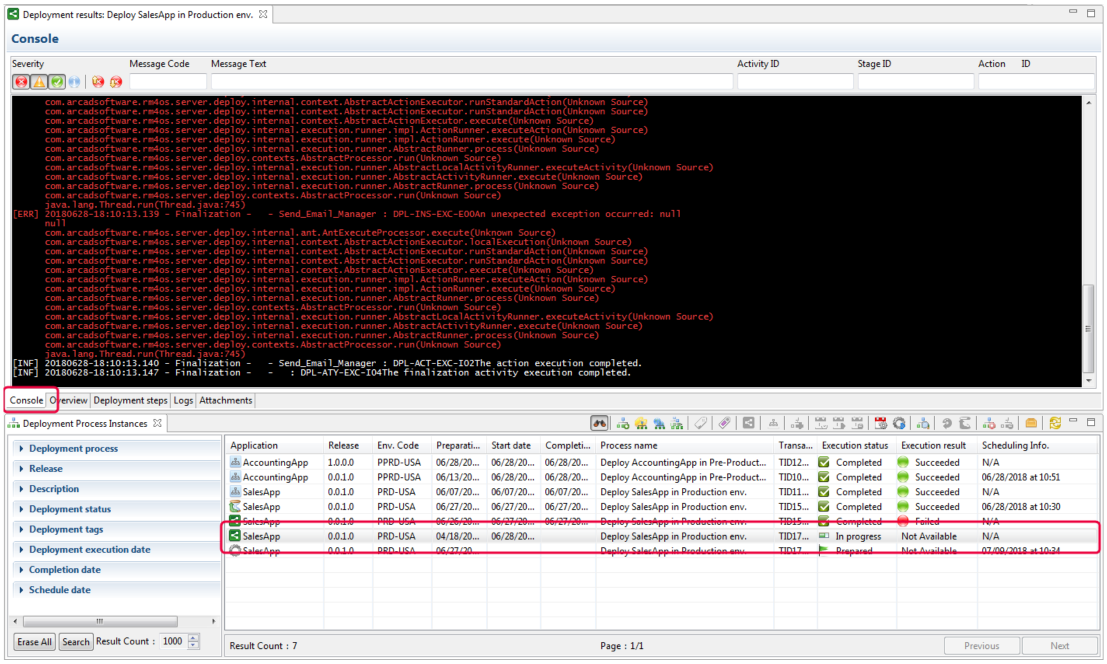
There are two ways to manually create installation-only instances:

1. To ⚙ create and immediately execute an installation-only instance, refer to Executing deployment process instances manually below.
2. To 📅 create and schedule an installation-only instance to execute later, refer to Scheduling deployment process instances on the facing page.

41.6 Executing deployment process instances manually

Deployment process instances can either be executed manually or scheduled to be executed automatically in the future. If an instance is scheduled, override the schedule by executing the instance manually.

To view what happens during the execution in real-time, open the deployment instance's **Results** view and watch the deployment actions take place from the **Console** tab.



```
     com.arcadsoftware.rm4os.server.deploy.internal.context.AbstractActionExecutor.runStandardAction(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.context.AbstractActionExecutor.runStandardAction(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.context.AbstractActionExecutor.execute(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.impl.ActionRunner.executeAction(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.impl.ActionRunner.execute(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractRunner.process(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.contexts.AbstractProcessor.run(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractLocalActivityRunner.executeActivity(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractActivityRunner.execute(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractRunner.process(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.contexts.AbstractProcessor.run(Unknown Source)
     java.lang.Thread.run(Thread.java:745)
[ERR] 20180628-18:10:13.139 - Finalization -   - Send_Email_Manager : DPL-INS-EXC-E00An unexpected exception occurred: null
     null
     com.arcadsoftware.rm4os.server.deploy.internal.ant.AntExecuteProcessor.execute(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.context.AbstractActionExecutor.localExecution(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.context.AbstractActionExecutor.runStandardAction(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.context.AbstractActionExecutor.runStandardAction(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.context.AbstractActionExecutor.execute(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.impl.ActionRunner.executeAction(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.impl.ActionRunner.execute(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractRunner.process(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.contexts.AbstractProcessor.run(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractLocalActivityRunner.executeActivity(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractActivityRunner.execute(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.internal.execution.runner.AbstractRunner.process(Unknown Source)
     com.arcadsoftware.rm4os.server.deploy.contexts.AbstractProcessor.run(Unknown Source)
     java.lang.Thread.run(Thread.java:745)
[INF] 20180628-18:10:13.140 - Finalization -   - Send_Email_Manager : DPL-ACT-EXC-I02The action execution completed.
[INF] 20180628-18:10:13.147 - Finalization -   -   : DPL-ATY-EXC-I04The finalization activity execution completed.
```

Application	Release	Env. Code	Preparati...	Start date	Completi...	Process name	Transa...	Execution status	Execution result	Scheduling Info.
AccountingApp	1.0.0.0	PPRD-USA	06/28/20...	06/28/20...	06/28/20...	Deploy AccountingApp in Pre-Product...	TID12...	✓ Completed	● Succeeded	N/A
AccountingApp	0.0.1.0	PPRD-USA	06/13/20...	06/28/20...	06/28/20...	Deploy AccountingApp in Pre-Product...	TID10...	✓ Completed	● Succeeded	06/28/2018 at 10:51
SalesApp	0.0.1.0	PRD-USA	06/07/20...	06/07/20...	06/07/20...	Deploy SalesApp in Production env.	TID11...	✓ Completed	● Succeeded	N/A
SalesApp	0.0.1.0	PRD-USA	06/27/20...	06/27/20...	06/27/20...	Deploy SalesApp in Production env.	TID15...	✓ Completed	● Succeeded	06/28/2018 at 10:30
SalesApp	0.0.1.0	PRD-USA	06/26/20...	06/27/20...	06/27/20...	Deploy SalesApp in Production env.	TID15...	✓ Completed	● Failed	N/A
SalesApp	0.0.1.0	PRD-USA	04/18/20...	06/28/20...		Deploy SalesApp in Production env.	TID17...	In progress	Not Available	N/A
SalesApp	0.0.1.0	PRD-USA	06/27/20...			Deploy SalesApp in Production env.	TID17...	Prepared	Not Available	07/09/2018 at 10:34

Result Count : 7 Page : 1/1

Figure 94: Consulting Deployment execution results in real-time (Console)

> **Note**
> The **Console** tab is only available in **Prepared** instances or those which are currently In Progress. If an instance has finished executing the Console is no longer available but the log can be accessed in the **Logs** tab.

41.6.1 Execute full-transfer and transfer-only instances

Full-transfer and transfer-only instances are not executed immediately when they are created.

To execute a full-transfer or transfer-only deployment process instance immediately, select it in the **Deployment process Instances** search view then either click the Execute icon or right-click then select

Execute. The deployment begins right away and any scheduling information for the instance is overridden.

41.6.2 Execute installation-only instances

Installation-only deployment process instances can be created and then executed immediately.

Follow the subsequent steps to create and execute an installation-only deployment process instance immediately from the **Deployment process Instances** search view.

Step 1 Select the transfer-only instance in the **Deployment process Instances** search view.

Step 2 Either click the execute installation-only icon, or right-click then select **Execute installation-only**.

Step 3 Click OK in the dialog to confirm.

Note

If the transfer-only stage of the deployment process has not been executed yet or was unsuccessful, a second dialog is displayed. You can either cancel the installation or proceed anyway and the actions in the deployment diagram's installation stage(s) will begin.

A transfer stage can "fail" when any action in the stage cannot be carried out. This means that the stage can "fail" after content is transferred by one action if the following action cannot be completed. If an action post-transfer fails, the transferred files can still be installed.

Example

If your transfer stage includes an action to send an email after a transfer is complete and the email cannot be sent (the final Action fails) the whole transfer stage will have a failed status even though the transfer action completed successfully.

Reference

To define when an action should be considered as complete or failed, modify to the action’s Fails on Error property.

Result The installation stage of the deployment process diagram begins automatically and the installation-only instance is created in the **Deployment process Instances** search view.

41.7 Scheduling deployment process instances

Deployment process instances can either be executed manually or scheduled to be executed by DROPS in the future.

> **Note**
> All instances in any mode can be scheduled but they *must* have a **Prepared** status. Instances that have already been executed once cannot be scheduled to deploy again.

41.7.1 Schedule full-transfer and transfer-only instances

Full-transfer and transfer-only instances are not executed immediately when they are created. Any deployment process instance's schedule can be modified up until the moment it executes.

Follow the subsequent steps to schedule a deployment process instance to execute in the future from the **Deployment process Instances** search view. These steps can also be used at any time to change the schedule defined for any instance.

Step 1 To access the **Scheduling Information** dialog, select the instance then either click the Change the scheduling information icon or right-click, then select **Change the scheduling information**.

Step 2 Tick the **Scheduled** checkbox to activate the schedule for the current instance.

Step 3 Select the **Date**, **Hour** and **Minute** that the instance should be executed.

Step 4 If the deployment should be carried out in transfer-only mode, tick the **transfer-only** checkbox. If this checkbox is not ticked, the deployment process instance will be carried out in full-transfer mode.

> **Note**
> If a full-transfer instance is changed to a transfer-only, the transfer-only's child installation-only instance is automatically created and scheduled to deploy at the same time as its parent transfer-only instance.
>
> If a transfer-only instance is changed to a full-transfer, the transfer-only's child installation-only instance is deleted.

Click **OK**.

Result The **Scheduling Info** column for the selected instance is filled in with the date and time of the scheduled execution.

41.7.2 Schedule installation-only instances

Installation-only deployment process instances can be created and scheduled to be executed by DROPS at a later date.

Follow the subsequent steps to create and schedule an installation-only deployment process instance from the **Deployment process Instances** search view.

Note
Only transfer-only instances can be used to schedule installation-only instances.

Step 1 To access the **Scheduling Information** dialog, select the transfer-only instance then either click the Schedule installation-only icon or right-click then select **Schedule installation-only**.

Step 2 Tick the **Scheduled** checkbox to activate the schedule for the Installation operation.

Step 3 Select the **Date**, **Hour** and **Minute** that the installation-only instance should be executed.

Note
The **Install Only** checkbox is not available because the parent instance is transfer-only by default.

Click **OK**.

Result The installation-only instance is created and the installation stage of the deployment process diagram will begin at the time defined (displayed in the **Scheduling Info** column).

41.7.2.1 Managing an installation-only instance's automatic schedule

If an installation-only instance was automatically scheduled but needs to be modified, toggle between automatic schedule and manual schedule (or no schedule at all - N/A) to deactivate or modify it as needed.

If an installation-only instance has been auto-scheduled, the label will be in one of the following formats:

- *AUTO [+<*delay_value*>] if no fixed time has been chosen
- *AUTO [+<*delay_value*>] [<*hour*>:<*minute*>] if a fixed time has been chosen

To toggle between the auto-schedule defined in the deployment process and a user-defined schedule follow the steps inModify any instance's schedule on the next page. Changing the schedule from *AUTO to the schedule defined by the user overrides the auto-schedule.

To toggle between the auto-schedule defined in the deployment process and no schedule at all, right-click on an instance and either select **Auto-Schedule** (to apply the auto-schedule) or **Remove Auto-Scheduling Mode** (to remove any schedule).

Note
The auto-schedule for installation-only instances can only be toggled on/off if the instance has a **Prepared** status and its transfer-only parent instance is not ☑ **Completed**.

Important!
Installation-only instances must always have a scheduled time to deploy. An installation-only instance with **N/A** in the **Scheduling Info** column will never execute. You cannot manually execute this kind of instance after it is created and if its schedule is toggled off.

41.7.3 Modify any instance's schedule

Follow the subsequent steps to modify any deployment process instance's schedule from the **Deployment process Instances** search view.

Step 1 Either right-click on the instance in the **Deployment process Instances** search view then select **Change the scheduling information** or select it, then click the Change the scheduling information icon to open the **Scheduling Information** dialog.

Step 2 Tick the **Scheduled** checkbox to activate the schedule for the current instance.

Step 3 Select the **Date**, **Hour** and **Minute** that the instance should be executed.

Click **OK**.

Result The **Scheduling Info** column for the selected instance is filled in with the date and time of the scheduled execution.

41.8 Remote deployments on node servers

Remote deployments are similar to deployments done in a simple single server configuration. The only difference is that you select a node server to execute the deployment instead of the main server.

A remote execution is an import and deploy action of the server. When executing the remote deployment instance, the node server executes an import instance to provision the node server's artifact repository, and then execute the deployment instance.

> **Important!**
> To delegate the execution of the import and deployment activities to node servers, a specific right must be enabled for the user role in charge of deployment: *Substitute another user when accessing web-services* in the **Server general administrative rights**.

> ***Reference***
> For more information about the main and node server configuration, refer to Introduction to main / node servers configuration on page 235.

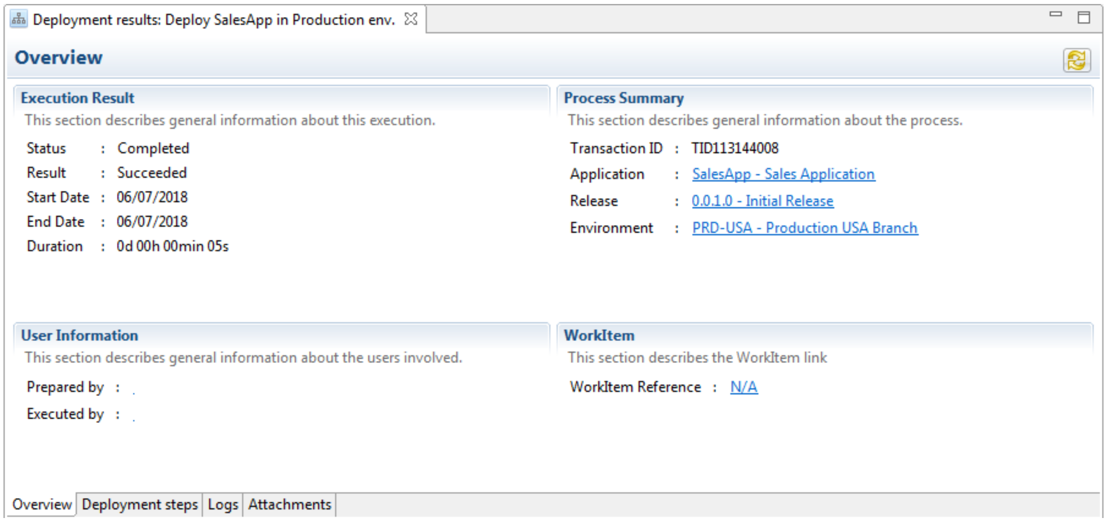
Main and node servers can be synchronized automatically prior the execution of the deployment process instance.

> ***Reference***
> To configure the automatic synchronization of the main and node servers before the execution of a deployment instance, refer to the Configuration Guide.

41.9 Understanding deployment process instance results

The **Deployment process Instances** search view provides some details of processed deployment process instances. Access more information by either double-clicking on an instance in the list, selecting it then clicking the results icon in the view or right-clicking then selecting **Show deployment results**.

The **Deployment Results** view contains four tabs that each contain different elements of the deployment.



Deployment results: Deploy SalesApp in Production env.

Overview

Execution Result
This section describes general information about this execution.

Status : Completed
Result : Succeeded
Start Date : 06/07/2018
End Date : 06/07/2018
Duration : 0d 00h 00min 05s

Process Summary
This section describes general information about the process.

Transaction ID : TID113144008
Application : SalesApp - Sales Application
Release : 0.0.1.0 - Initial Release
Environment : PRD-USA - Production USA Branch

User Information
This section describes general information about the users involved.

Prepared by : .
Executed by : .

WorkItem
This section describes the WorkItem link

WorkItem Reference : N/A

Figure 95: Overview of the deployment process instance results

The **Overview** tab displays the uneditable general information about the instance. The **End Date** and **Executed by** fields are blank until the instance is executed.

41.9.1 View the steps carried out

All of the activities and stages included in the deployment process, along with their current **Execution Status** and **Result**, are displayed in the **Deployment Steps** tab.

If a deployment process instance is suspended, check this tab to see why. Often, the deployment is waiting for approval which means that a validation request is pending.

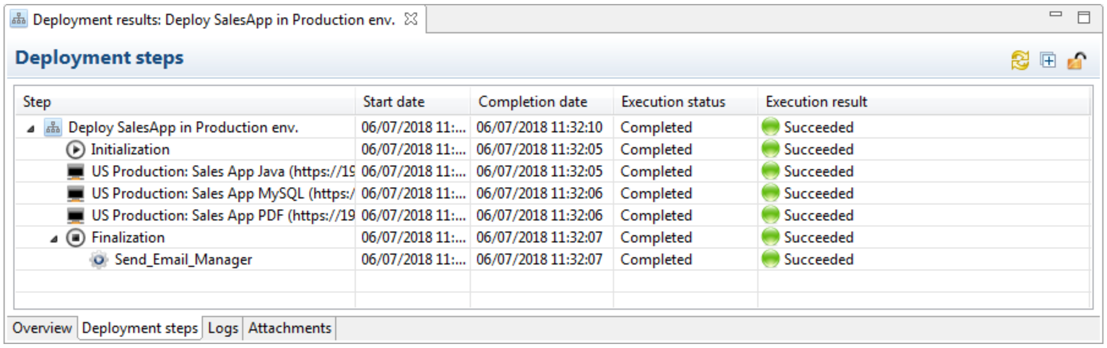
Reference
For more information about resolving suspended deployment instances, refer to Validation requests on page 296 or Suspended actions on page 294.

If a file transfer fails, DROPS will retry the transfer a number of times. This number of attempts is defined in the server configuration (the default value is 3). If all attempts fail, the process is aborted and the **Execution Result** is set to ● **Failed**.

Before trying to re-transfer the file, DROPS will check if the remote agent is reachable. If not, it will retry to reconnect a number of times, this number of attempts is also defined in the server configuration (the default value is 3).

If all the reconnection attempts fail, the related file transfer attempt is marked as failed.

Reference
For more information about defining the defaults mentioned above, refer to the DROPS *Configuration Guide*


Deployment results: Deploy SalesApp in Production env.

Deployment steps

Step	Start date	Completion date	Execution status	Execution result
Deploy SalesApp in Production env.	06/07/2018 11:...	06/07/2018 11:32:10	Completed	Succeeded
Initialization	06/07/2018 11:...	06/07/2018 11:32:05	Completed	Succeeded
US Production: Sales App Java (https://19	06/07/2018 11:...	06/07/2018 11:32:05	Completed	Succeeded
US Production: Sales App MySQL (https:/	06/07/2018 11:...	06/07/2018 11:32:06	Completed	Succeeded
US Production: Sales App PDF (https://19	06/07/2018 11:...	06/07/2018 11:32:06	Completed	Succeeded
Finalization	06/07/2018 11:...	06/07/2018 11:32:07	Completed	Succeeded
Send_Email_Manager	06/07/2018 11:...	06/07/2018 11:32:07	Completed	Succeeded

Figure 96: The deployment process result view - steps

41.9.2 View the Logs

All of the activity from the deployment is displayed in the **Logs** tab. Every action carried out during a deployment is logged.

To filter the records at the source and exclude specific entries that are unnecessary, create a regex and enter it in the ANT Message Filter field found in the **Local Execution settings** of the DROPS Server. If no regex is defined, all entries are logged in the deployment log. Refer to the *DROPS Configuration Guide*.

Filter results by **Message Code**, **Message Text**, **Activity ID**, **Stage ID** or **Action ID**. Enter any combination of search criteria then click the search icon to display the results. To display the complete log, click search without entering any criteria.

Filter the severity of the results displayed by selecting one or more of the **Severity** icons

- ✗ **Fatal**: only fatal errors are displayed.
- ⚠ **Warning**: only Warnings errors are displayed.
- ℹ **Info**: only Important information is displayed.
- **Verbose**: only details are displayed.

Use the Show Previous and Next icons to quickly move through a long list of entries to find and jump to the fatal errors.

Click the export icon to open the selected log(s) in an external text editor. By default, the editor will open automatically.

Click the export to JSON file icon to open the selected log(s) in .json format. You will be prompted to select where to save the file.

If no log entries are displayed, the log may be archived. Logs can be automatically or manually archived to create space in the database. If no logs are available in this view, check the **Attachments** tab to access the archived .txt log file.

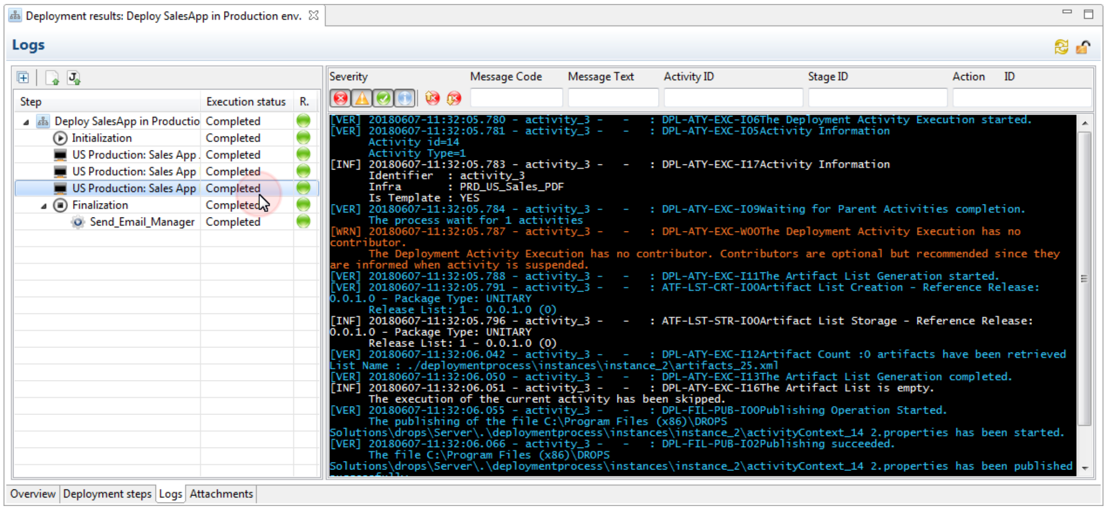
While only successfully deployed instance's logs can be automatically archived, both successful and unsuccessful deployment logs can be manually archived. To manually archive a deployment process instance's log, search for it in the **Deployment process Instances** search view. Either right-click on it then select **Archive Log** or select it, then click the archive log icon. When archived, logs are removed from the database which liberates space.

> ***Reference***
>
> For more information about activating the auto-archive option for deployment logs and how to define the location to store archived logs, refer to the *DROPS Configuration Guide*.

Execution instances' logs are stored in separate databases so they do not share a common base. This helps to limit the space required to store deployment results. Logs are written once the execution is complete which helps speed up the amount of time it takes to complete the deployment and record the results.

The steps carried out during deployment are displayed on the left in the new logs, just as they are in the **Deployment Steps** tab. Each step's individual result (failed or successful) are displayed in order to quickly address the step that may be causing problems in your deployment. To view the log information for any part of the process, click on the step on the left. The log for the step selected, as well as any sub-steps/actions/scripts included in the parent selected, are displayed on the right.

> ***Example***
>
> To view the entire process' log, click on the highest-level entry in the steps. The full log is displayed on the right.
>
> To view the log for a certain infrastructure item or action carried out, click on the step to filter the log as much as required to find the exact action in the process that failed.
>
> In the example below, the deployment to the Apache infrastructure item failed in general but we can see that only one stage in that item's process failed. Furthermore we can see that only one action in the installation stage failed but not all of them. If we click on the action that failed, we can see its unique log and quickly find the problem.



Deployment results: Deploy SalesApp in Production env.

Logs

Step	Execution status	R.
Deploy SalesApp in Productio	Completed	●
Initialization	Completed	●
US Production: Sales App .	Completed	●
US Production: Sales App	Completed	●
US Production: Sales App	Completed	●
Finalization	Completed	●
Send_Email_Manager	Completed	●

Severity | Message Code | Message Text | Activity ID | Stage ID | Action | ID

```
[VER] 20180607-11:32:05.780 - activity_3 -   -   : DPL-ATY-EXC-I06The Deployment Activity Execution started.
[VER] 20180607-11:32:05.781 - activity_3 -   -   : DPL-ATY-EXC-I05Activity Information
      Activity id=14
      Activity Type=1
[INF] 20180607-11:32:05.783 - activity_3 -   -   : DPL-ATY-EXC-I17Activity Information
      Identifier  : activity_3
      Infra       : PRD_US_Sales_PDF
      Is Template : YES
[VER] 20180607-11:32:05.784 - activity_3 -   -   : DPL-ATY-EXC-I09Waiting for Parent Activities completion.
      The process wait for 1 activities
[WRN] 20180607-11:32:05.787 - activity_3 -   -   : DPL-ATY-EXC-W00The Deployment Activity Execution has no
contributor.
      The Deployment Activity Execution has no contributor. Contributors are optional but recommended since they
are informed when activity is suspended.
[VER] 20180607-11:32:05.788 - activity_3 -   -   : DPL-ATY-EXC-I11The Artifact List Generation started.
[VER] 20180607-11:32:05.791 - activity_3 -   -   : ATF-LST-CRT-I00Artifact List Creation - Reference Release:
0.0.1.0 - Package Type: UNITARY
      Release List: 1 - 0.0.1.0 (0)
[INF] 20180607-11:32:05.796 - activity_3 -   -   : ATF-LST-STR-I00Artifact List Storage - Reference Release:
0.0.1.0 - Package Type: UNITARY
      Release List: 1 - 0.0.1.0 (0)
[VER] 20180607-11:32:06.042 - activity_3 -   -   : DPL-ATY-EXC-I12Artifact Count :0 artifacts have been retrieved
List Name : ./deploymentprocess\instances\instance_2\artifacts_25.xml
[VER] 20180607-11:32:06.050 - activity_3 -   -   : DPL-ATY-EXC-I13The Artifact List Generation completed.
[INF] 20180607-11:32:06.051 - activity_3 -   -   : DPL-ATY-EXC-I16The Artifact List is empty.
      The execution of the current activity has been skipped.
[VER] 20180607-11:32:06.055 - activity_3 -   -   : DPL-FIL-PUB-I00Publishing Operation Started.
      The publishing of the file C:\Program Files (x86)\DROPS
Solutions\drops\Server\.\deploymentprocess\instances\instance_2\activityContext_14 2.properties has been started.
[VER] 20180607-11:32:06.066 - activity_3 -   -   : DPL-FIL-PUB-I02Publishing succeeded.
      The file C:\Program Files (x86)\DROPS
Solutions\drops\Server\.\deploymentprocess\instances\instance_2\activityContext_14 2.properties has been published
```

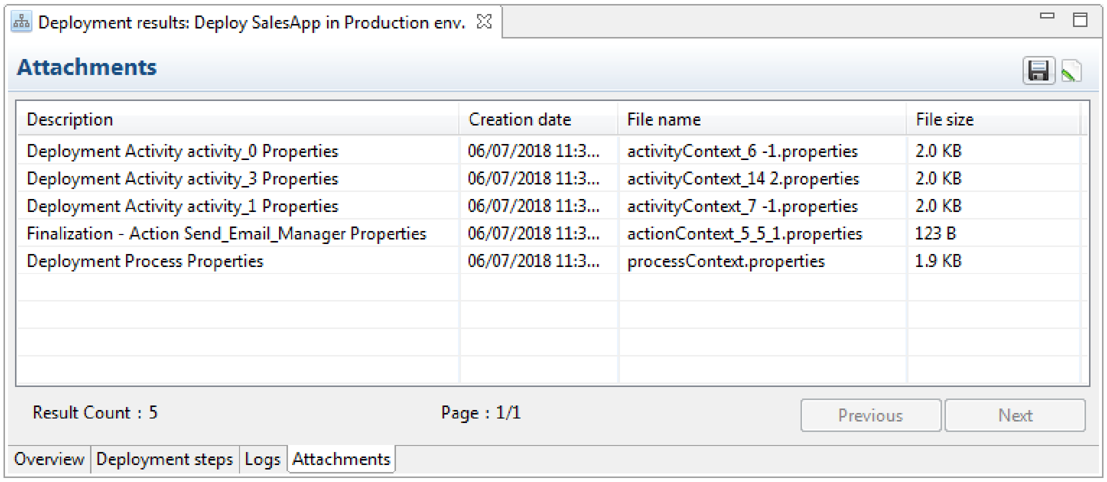
Figure 97: The Deployment Results view - Logs

> ***Reference***
>
> For more information about changing the syntax displayed for each line in the log, refer to the *DROPS Configuration Guide*.

41.9.3 View the attachments

Attachments that are automatically created represent the deployed artifact list and *.properties* files. Attachments can also be added to deployment process instances by the **Publish File** script (action). Archived deployment logs are automatically added to an instance's attachments and can be downloaded as a .txt file.

View any attachments published during the deployment action in the **attachments** tab.



Deployment results: Deploy SalesApp in Production env.

Attachments

Description	Creation date	File name	File size
Deployment Activity activity_0 Properties	06/07/2018 11:3...	activityContext_6 -1.properties	2.0 KB
Deployment Activity activity_3 Properties	06/07/2018 11:3...	activityContext_14 2.properties	2.0 KB
Deployment Activity activity_1 Properties	06/07/2018 11:3...	activityContext_7 -1.properties	2.0 KB
Finalization - Action Send_Email_Manager Properties	06/07/2018 11:3...	actionContext_5_5_1.properties	123 B
Deployment Process Properties	06/07/2018 11:3...	processContext.properties	1.9 KB

Result Count : 5   Page : 1/1

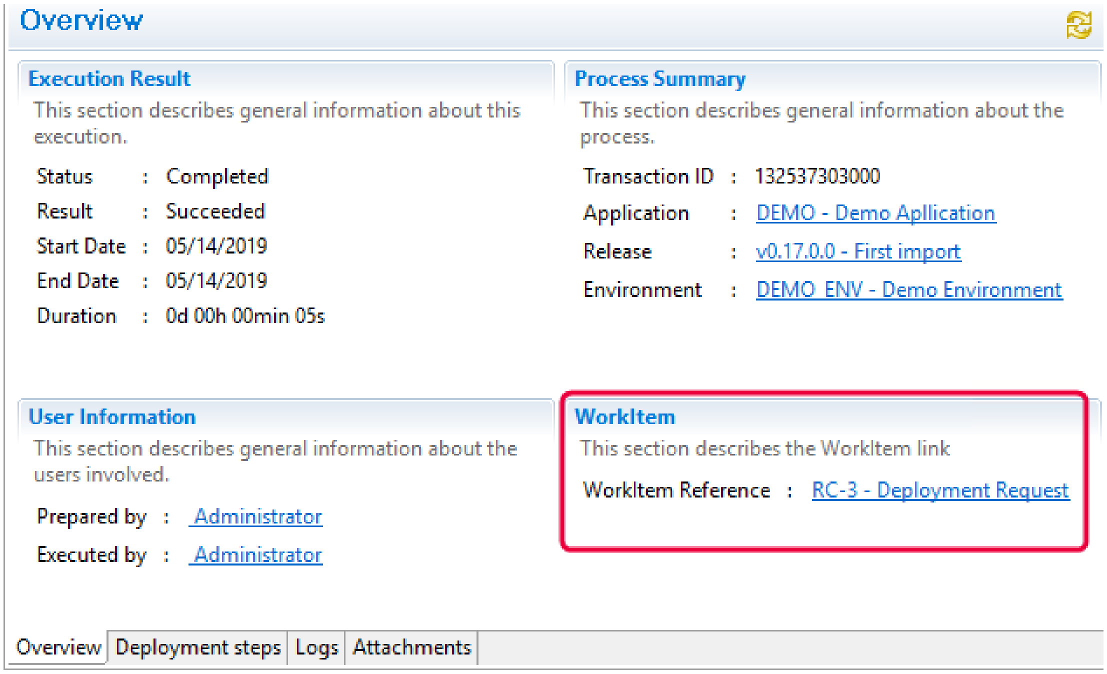
Figure 98: The Deployment Result view - attachments

Download an attachment by selecting it and clicking the 💾 Save icon.

Open the attachment in a local viewer (if possible) by selecting it and clicking the 📄 Open icon.

41.10 Viewing associated workitems

If the deployment process was associated with a workitem, a link to the JIRA issue is included in the **Workitem** section. Click the link to open the issue in your default browser.



Overview

Execution Result

This section describes general information about this execution.

Status	:	Completed
Result	:	Succeeded
Start Date	:	05/14/2019
End Date	:	05/14/2019
Duration	:	0d 00h 00min 05s

Process Summary

This section describes general information about the process.

Transaction ID	:	132537303000
Application	:	DEMO - Demo Apllication
Release	:	v0.17.0.0 - First import
Environment	:	DEMO_ENV - Demo Environment

User Information

This section describes general information about the users involved.

Prepared by	:	Administrator
Executed by	:	Administrator

WorkItem

This section describes the WorkItem link

WorkItem Reference	:	RC-3 - Deployment Request

Figure 99: The Deployment Result view - workitems

41.11 Re-executing failed deployments

If, for any reason, a deployment failed this option allows you to resume where it left off after fixing the issue (s) that caused the failure. All of the actions performed correctly before the failure was encountered are still valid and will not be affected by re-executing the deployment. Any transfers already made will not be made again since the process only restarts from the first action after the failure. A process can fail, be corrected and resumed as many time as needed to correct all the issues it may encounter. A log is created each time a deployment is resumed.

Follow the subsequent steps to re-execute a failed deployment.

Step 1 Select the failed instance to re-execute in the **Deployment process Instances** search view then either click the Re-execute icon or right-click, then select **Re-execute**.

Step 2 Click **OK** in the dialog to confirm that the instance should be re-executed.

Result All of the stages and actions in the deployment process not yet executed by the instance are launched.

41.12 Rolling back a deployed deployment process instance

Any deployment process instance can be rolled back in order to reinstate the previously deployed content that the current deployment overrode. Rollback for deployment process instances can either be executed manually or scheduled to be executed automatically in the future.

41.12.1 Roll back a deployment manually

Follow the subsequent steps to create and execute a rollback instance immediately from the **Deployment process Instances** search view.

Step 1 Select the instance to roll back in the **Deployment process Instances** search view then either click the Rollback icon or right-click, then select **Rollback**.

Step 2 Click **OK** in the dialog to confirm that the instance should be rolled back.

Step 3 Click **No** in the dialog to schedule the rollback for another time.

Result All of the stages and actions in the deployment process executed by the instance are recalled in reverse order and the new rollback instance is available in the **Deployment process Instances** search view.

41.12.2 Schedule a deployment's rollback

Follow the subsequent steps to schedule a rollback instance from the **Deployment process Instances** search view.

Step 1 Select the instance to roll back in the **Deployment process Instances** search view then either click the rollback icon or right-click, then select **Rollback**.

Step 2 Click **OK** in the dialog to confirm that the instance should be rolled back.

Step 3 Click **Yes** in the dialog to schedule the rollback for another time.

Step 4 Define the **Date**, **Hour** and **Minute** when the rollback should be executed.

> **Note**
>
> If the instance to roll back is already ✓ Completed, the scheduled date/time to roll back must be after the source's Completion Date.
>
> If the instance to roll back is scheduled, the scheduled date/time to roll back must be after the source's Scheduled Date.
>
> If the instance to roll back is not completed and not scheduled, the date/time to roll back must be after the source's Start Date.

Result The new rollback instance is available in the **Deployment process Instances** search view.

Schedules for rollback instances are carried over to the scheduler and any changes made via the scheduler are carried over to the **Deployment process Instances** search view.

41.13 Canceling deployment process instances

> ***Warning!***
> Canceled deployment process instances cannot be restarted.

To cancel a deployment process instance search for it in the **Deployment process Instances** search view. Either right-click on it, then select **Cancel Execution** or select it, then click the cancel icon. Click **OK** to confirm or click **Cancel** to keep the instance.

41.14 Deleting deployment process instances

> **Note**
> Instances that are In Progress cannot be deleted. To delete an instance that is stuck in progress, cancel it first, then delete it.

> ***Warning!***
> Deleted deployment process instances cannot be accessed or recovered. Deleting an instance removes the trace of the deployment.

To delete a deployment process instance search for it in the **Deployment process Instances** search view. Either right-click on it, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the instance.

42 Deployment plan instances

Required Roles	Environment Release Manager and/or Release Operator
Required Rights (by category)	Deployment Plan Execution Instance
Perspective	Operational Console Management
Access	DROPS Server → Deployment Plans → Deployment Plan Instances

Chapter Summary

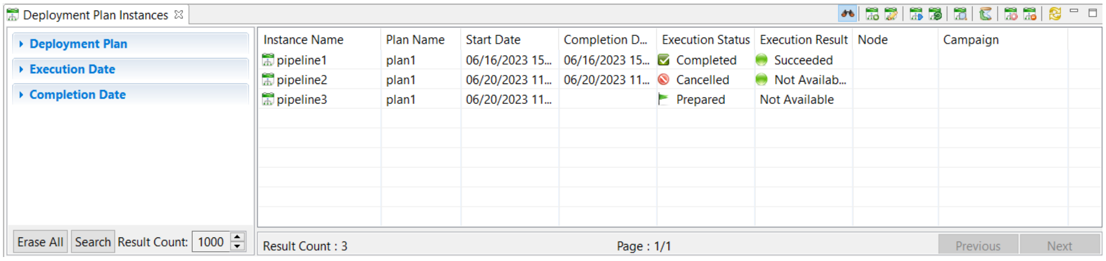
42.1 Creating deployment plan instances 285
42.2 Executing deployment plan instances 286
42.3 Understanding deployment plan instance results 287
42.4 Re-executing failed deployments 288
42.5 Rolling back a deployed deployment plan instance 289
42.6 Canceling deployment plan instances 289
42.7 Deleting deployment plan instances 289

Deployment plans enable you to deploy one or more applications to multiple environments.

Deployment plan instances are created automatically when a deployment plan is executed. Instances can also be created from the **Deployment Plan Instances** search view and either executed manually, or scheduled to execute them at a later date or regularly.

Multiple deployment plan instances can be scheduled to match your team's needs. To manage when the first the stages and actions included in the deployment plan diagram are executed, schedule when a deployment plan instance should be carried out. Scheduling deployment plans is useful if you need to continuously deliver content that is regularly updated or if you know in advance when the next batch of content to deliver will be ready.

A dedicated job runs on the DROPS Server that regularly scans for scheduled deployment plan instances, then triggers them according to the schedule(s).



Instance Name	Plan Name	Start Date	Completion D...	Execution Status	Execution Result	Node	Campaign
pipeline1	plan1	06/16/2023 15...	06/16/2023 15...	✓ Completed	● Succeeded		
pipeline2	plan1	06/20/2023 11...	06/20/2023 11...	⊘ Cancelled	● Not Availab...		
pipeline3	plan1	06/20/2023 11...		Prepared	Not Available		

Result Count : 3   Page : 1/1

Figure 100: The Deployment Plan Instance search view

The **Deployment Plan Instances** search view is accessed from the **Deployment Plans** node in the **Console Explorer**.

42.1 Creating deployment plan instances

Follow the subsequent steps to create new deployment plan instances.

Step 1 To access the **Create a new deployment plan instance** wizard, either click the create icon in the **Deployment Plan Instances** search view or right-click anywhere in the view and select **Create a new deployment plan instance**.

Step 2 Define the main properties by clicking the browse button for each of the following entities.

> **Important!**
> Select the following properties carefully. They cannot be changed once the instance is created.

1. If multiple clients are available, select the **Client** to which to deploy the release(s). Changing the client will alter the options available throughout the wizard. Only the entities available for the selected client will be accessible.
2. Enter a name for the specific plan execution instance.
3. Select the **Deployment Plan** to use. The list is prepopulated with the plans available for the selected client.
4. Select a remote execution node from the **Node** drop-down list.

Click **Next >** to continue.

Step 3 Select the release to deploy for each application in the plan.

Select each application listed and click the **Edit** button to select the release. The list is prepopulated with all of the releases available for the application.

Click **Next >** to continue.

Step 4 To **Generate transfer-only and installation-only instances**, leave the corresponding checkbox ticked [*default*]. If the check box is not selected, full-transfer instances are created.

Transfer- and installation-only instances will be executed in order. First, all of the transfers for all of the diagrams in the plan will be executed. If they are all successful, then all of the installation instances will be executed, in order. Full-transfer instances for plans will also be carried out in order, but each process will fully transfer before moving on to the next process in the plan.

> ***Reference***
> For more information about how deployment process instances are deployed, refer to Defining an instance's deployment mode on page 270

Click **Finish**.

Result New deployment plan instances are displayed in the **Deployment Plan instances** search view.

If the plan instance was scheduled to execute immediately, clicking Finish executes the deployment.

42.2 Executing deployment plan instances

Deployment plan instances are, by default, scheduled to be executed automatically. Instances can also be executed manually or scheduled to be executed at a specific time and date in the future. Manually executing an instance will override the schedule.

To execute a plan instance manually, either select it in the **Deployment Plan instances** search view and click the Execute icon or right-click on it and select **Execute**.

Double-click on a plan to see the plan progress diagram. This diagram displays the plan in a different manner and enables you to follow the progress of each stage and each deployment process instance created by the plan.

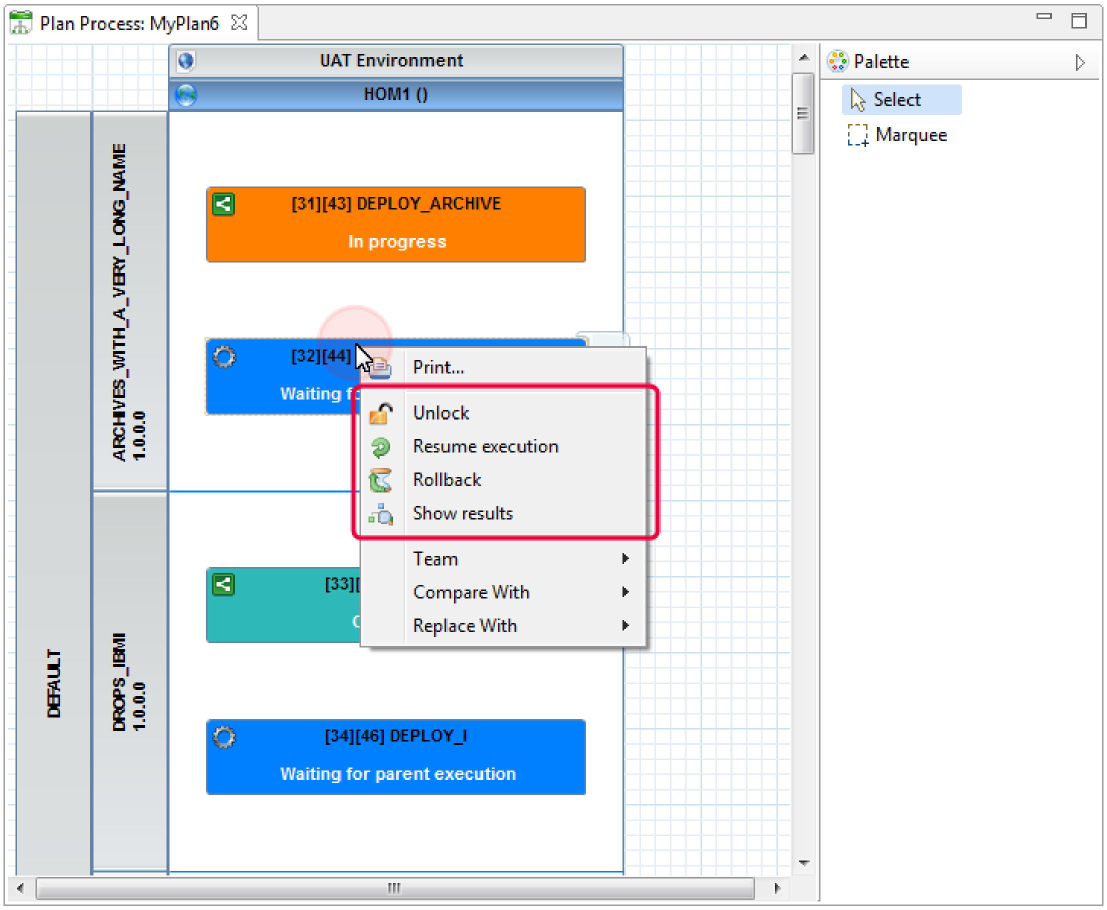


Figure 101: Follow the progress of a deployment plan

Right-click on a deployment instance to manage it individually. You can unlock a dependent instance from its parent to launch it without waiting for the parent, resume instances that have been paused, suspended or validated, roll instances back and display the deployment results.

> ***Reference***
>
> For more information about these actions, refer toRe-executing failed deployments on page 281Rolling back a deployed deployment process instance on page 282, andUnderstanding deployment process instance

results on page 277.

42.3 Understanding deployment plan instance results

The **Deployment Plan instances** search view provides some details of processed deployment plan instances. Access more information by either double-clicking on an instance in the list, selecting it then clicking the results icon in the view or right-clicking then selecting **Show deployment plan results**.

The **Deployment Plan Instance Results** view contains three tabs that each contain different elements of the deployment.

The **Overview** tab displays the uneditable general information about the instance. The **End Date** and **Executed by** fields are blank until the instance is executed.

42.3.1 View the instance status

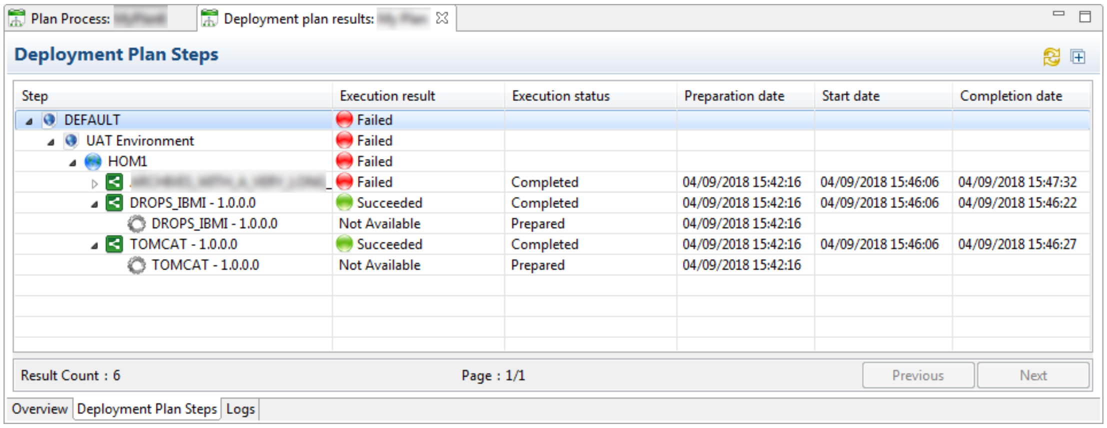
From the **Deployment Plan Instances** search view, the **Execution Status** and **Execution Result** columns display the current status of instances and change while the instance is being processed.

Prepared	Prepared instances are ready to be executed.
☑ Completed	Completed instances have already been executed.
In Progress	The progress bar is displayed while the instance is executing.
Suspended	Suspended deployment process instances are highlighted in red to bring attention to the fact that an action is required before the instance can continue to deploy. Deployment process instances are suspended when either a ✓ validation request is required.
☒ Canceled	If an instance is canceled.
🟢 Succeeded	If an instance executed all of its actions successfully.
🔴 Failed	If any step in deployment process failed to successfully execute.

Table 21: Deployment plan instance statuses & results

42.3.2 View the steps carried out

All of the stages included in the deployment plan process, along with their current **Execution Status** and **Result**, are displayed in the **Deployment Instance** tab.



Deployment Plan Steps

Step	Execution result	Execution status	Preparation date	Start date	Completion date
DEFAULT	🔴 Failed				
UAT Environment	🔴 Failed				
HOM1	🔴 Failed				
[redacted]	🔴 Failed	Completed	04/09/2018 15:42:16	04/09/2018 15:46:06	04/09/2018 15:47:32
DROPS_IBMI - 1.0.0.0	🟢 Succeeded	Completed	04/09/2018 15:42:16	04/09/2018 15:46:06	04/09/2018 15:46:22
DROPS_IBMI - 1.0.0.0	Not Available	Prepared	04/09/2018 15:42:16		
TOMCAT - 1.0.0.0	🟢 Succeeded	Completed	04/09/2018 15:42:16	04/09/2018 15:46:06	04/09/2018 15:46:27
TOMCAT - 1.0.0.0	Not Available	Prepared	04/09/2018 15:42:16		

Result Count : 6   Page : 1/1

Figure 102: The deployment plan result view - steps

42.3.3 View the Logs

All of the activity from the deployment is displayed in the **Logs** tab. Every action carried out during a deployment is logged.

Filter results by **Code** or **Text**. Enter any combination of search criteria then click the **Search** button to display the results. To display the complete log, click **Search** without entering any criteria.

Filter the severity of the results displayed by selecting one or more of the **Severity** icons

- ✖ **Fatal**: only fatal errors are displayed.
- ⚠ **Warning**: only Warnings errors are displayed.
- ℹ **Info**: only Important information is displayed.
- **Verbose**: only details are displayed.

42.4 Re-executing failed deployments

If, for any reason, a deployment failed this option allows you to resume where it left off after fixing the issue (s) that caused the failure. All of the actions performed correctly before the failure was encountered are still valid and will not be affected by re-executing the deployment. Any transfers already made will not be made again since the process only restarts from the first action after the failure. A process can fail, be corrected and resumed as many time as needed to correct all the issues it may encounter. A log is created each time a deployment is resumed.

Follow the subsequent steps to re-execute a failed deployment.

Step 1 Select the 🔴 failed instance to re-execute in the **Deployment Plan Instances** search view then either click the Re-execute icon or right-click, then select **Re-execute**.

Step 2 Click **OK** in the dialog to confirm that the instance should be re-executed.

Result All of the stages and actions in the deployment plan not yet executed by the instance are launched.

42.5 Rolling back a deployed deployment plan instance

Follow the subsequent steps to create and execute a rollback instance immediately from the **Deployment Plan Instances** search view.

Step 1 Select the instance to roll back in the **Deployment Plan Instances** search view then either click the rollback icon or right-click, then select **Rollback**.

Step 2 Click **OK** in the dialog to confirm that the instance should be rolled back.

Step 3 Click **No** in the dialog to schedule the rollback for another time.

Result All of the stages and actions in the deployment plan executed by the instance are recalled in reverse order and the new rollback instance is available in the **Deployment Process Instances** search view.

42.6 Canceling deployment plan instances

> ***Warning!***
> Canceled deployment plan instances cannot be restarted.

To be able to cancel a deployment plan instance, you must have the Release Manager or Release Operator role attributed on each environment of the deployment plan.

Deployment plans can only be canceled if its status is:

- In progress
- Waiting for validation
- Suspended (the infrastructure item is not reachable)

To cancel a deployment plan instance, search for it in the **Deployment Plan instances** search view. Either right-click on it, then select **Cancel** or select it, then click the cancel icon. Click **OK** to confirm or click **Cancel** to keep the instance.

42.7 Deleting deployment plan instances

> ***Warning!***
> Deleted deployment plan instances cannot be accessed or recovered.
> Deleting a deployment plan instance removes all the related items (plan data, validation requests, etc.).

A deployment plan instance can only be deleted if its status is:

- Prepared
- Completed
- Canceled

To delete an deployment plan instance, search for it in the **Deployment Plan Instances** search view. Either right-click on it, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the instance.

43 Interruption messages

Required Roles	Environment Release Operator Infrastructure Item Manager and/or Contributor
Required Rights (by category)	Suspended Actions
Perspective	Operational Console Management
Access	DROPS Server → Deployment Process → Interruption Messages

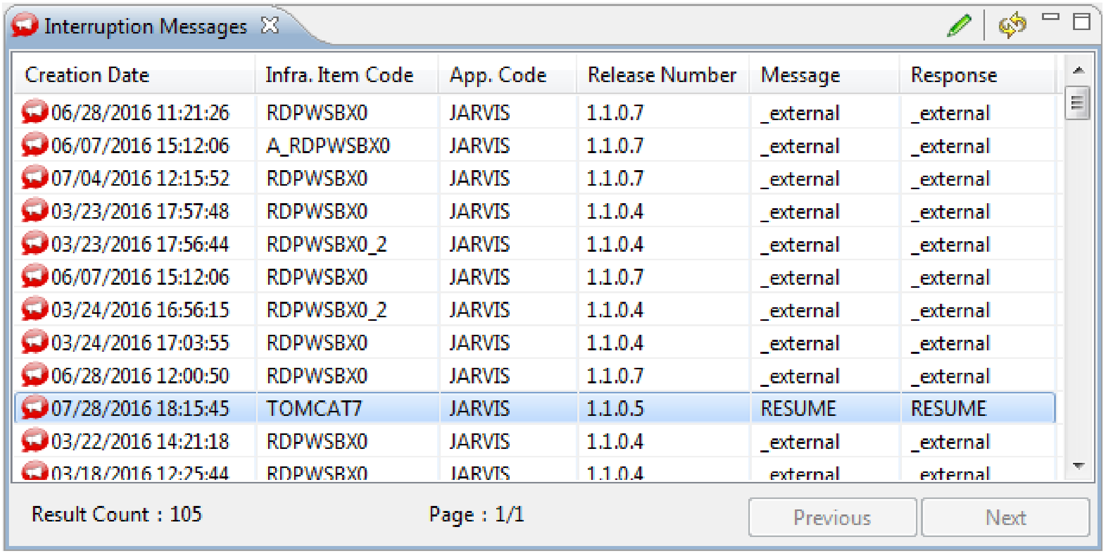
Chapter Summary

43.1 Responding to interruption messages 292
43.2 Reviewing processed interruption messages 293

Some ANT tasks, which are called during the deployment process, can launch external processes on target systems. If an external process becomes suspended, the DROPS Agent relays the problem and the possible solutions back to the DROPS Server as an interruption message. Interruption messages must be attended to before deployment can continue.

If an external process is suspended, a generic notification email is automatically sent to the user(s) in the designated role(s). Users with the role to do so choose the correct response to a interruption message which pilots the action to take on the suspended external process.

> **Note**
> To work with ARCAD as an external process, you must have ARCAD ≥v10.06.xx installed.



Creation Date	Infra. Item Code	App. Code	Release Number	Message	Response
06/28/2016 11:21:26	RDPWSBX0	JARVIS	1.1.0.7	_external	_external
06/07/2016 15:12:06	A_RDPWSBX0	JARVIS	1.1.0.7	_external	_external
07/04/2016 12:15:52	RDPWSBX0	JARVIS	1.1.0.7	_external	_external
03/23/2016 17:57:48	RDPWSBX0	JARVIS	1.1.0.4	_external	_external
03/23/2016 17:56:44	RDPWSBX0_2	JARVIS	1.1.0.4	_external	_external
06/07/2016 15:12:06	RDPWSBX0	JARVIS	1.1.0.7	_external	_external
03/24/2016 16:56:15	RDPWSBX0_2	JARVIS	1.1.0.4	_external	_external
03/24/2016 17:03:55	RDPWSBX0	JARVIS	1.1.0.4	_external	_external
06/28/2016 12:00:50	RDPWSBX0	JARVIS	1.1.0.7	_external	_external
07/28/2016 18:15:45	TOMCAT7	JARVIS	1.1.0.5	RESUME	RESUME
03/22/2016 14:21:18	RDPWSBX0	JARVIS	1.1.0.4	_external	_external
03/18/2016 12:25:44	RDPWSBX0	JARVIS	1.1.0.4	_external	_external

Result Count : 105   Page : 1/1

Figure 103: The Interruption Messages view

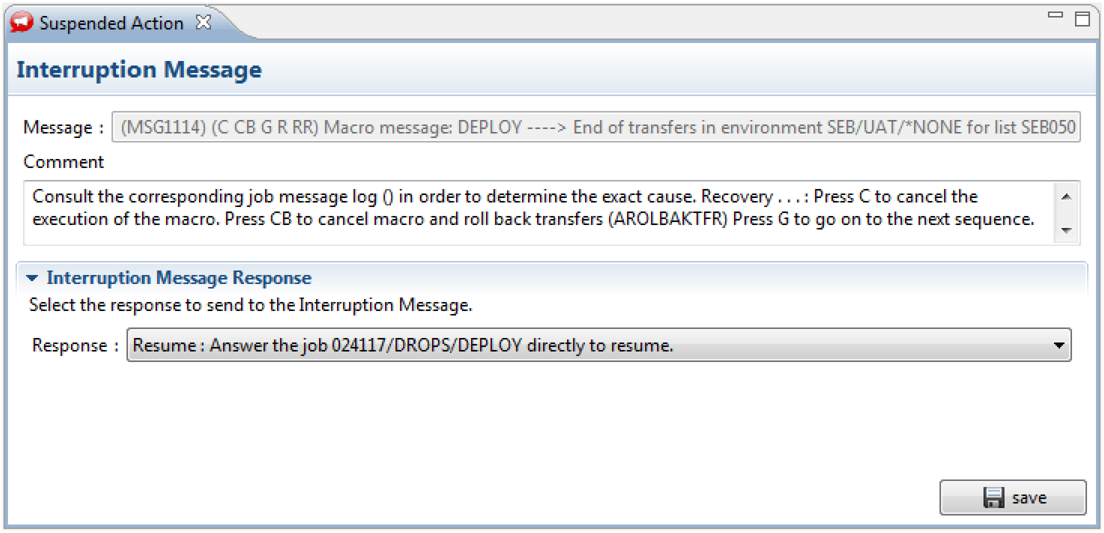
The **Interruption Messages** view displays all of the pending messages that you must reply to in order for the associated deployment process to continue forward.

43.1 Responding to interruption messages

A notification email is sent to the users who have the correct role to respond to interruption messages.

> ***Reference***
> For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide*.

To respond to the interruption message you must fill out the response manually. Select the item in the **Interruption Messages** view and either click the Respond icon or right-click on it and select **Respond** to open the **Interruption Messages** editor.



Message : (MSG1114) (C CB G R RR) Macro message: DEPLOY ----> End of transfers in environment SEB/UAT/*NONE for list SEB050

Comment

Consult the corresponding job message log () in order to determine the exact cause. Recovery . . . : Press C to cancel the execution of the macro. Press CB to cancel macro and roll back transfers (AROLBAKTFR) Press G to go on to the next sequence.

Interruption Message Response

Select the response to send to the Interruption Message.

Response : Resume : Answer the job 024117/DROPS/DEPLOY directly to resume.

Figure 104: The Interruption Messages editor

The **Message** and **Comment** fields display information about why the external process was interrupted.

The options from which to choose to resolve the problem depend on the actual type of problem encountered. Every interruption will have different options to resolve it. Select the corresponding response from the **Response** drop-down menu.

If there is only one response to the problem, it is automatically selected. There will be nothing to change but you must save to confirm that the option is confirmed.

To save changes made, click the **Save** button.

> **Important!**
> You can only respond to an interruption messages once. After the selected option is saved, the **Response** field is read-only.

43.2 Reviewing processed interruption messages

To view all of the processed interruption messages, select the **Processed interruption messages** node from **Interruption Messages** in the **Console Explorer**.

The **Interruption Messages** view displays all of the messages that you have addressed. Double-click on a message to open the **Interruption Messages** editor.

44 Suspended actions

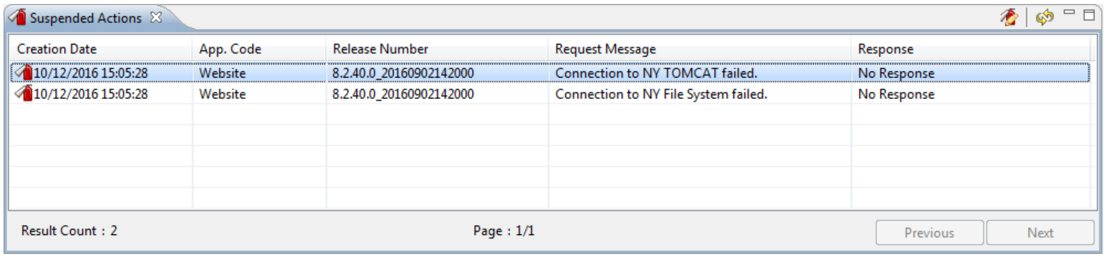
Required Roles	Environment Release Operator Infrastructure Item Manager and/or Contributor for the infrastructure item
Required Rights (by category)	Suspended Actions
Perspective	Operational Console Management
Access	DROPS Server → Deployment Process → Suspended Actions

Chapter Summary

44.1 Responding to suspended actions 294
44.2 Reviewing processed suspended actions 295

Deployment processes are suspended when one of two things occurs. Either an infrastructure item in the process cannot be accessed (the connection is incorrect) or the license key is restricting the item from being used.

If a deployment process is suspended, a generic notification email is automatically sent to the user(s) in the designated role(s). Suspended actions must be attended to before deployment can continue.



Creation Date	App. Code	Release Number	Request Message	Response
10/12/2016 15:05:28	Website	8.2.40.0_20160902142000	Connection to NY TOMCAT failed.	No Response
10/12/2016 15:05:28	Website	8.2.40.0_20160902142000	Connection to NY File System failed.	No Response

Result Count : 2

Page : 1/1

Figure 105: The pending Suspended Actions view

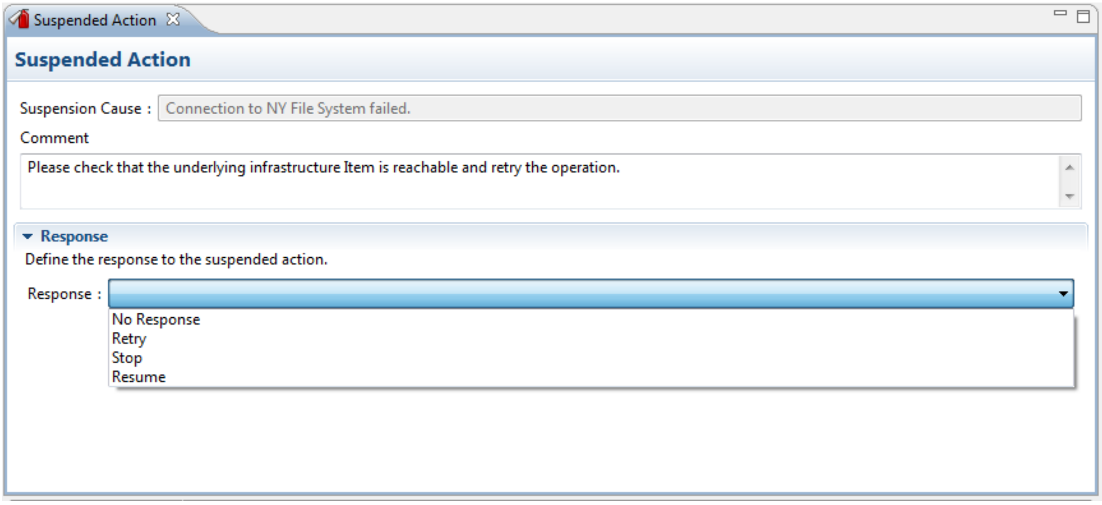
The **Suspended Actions** view displays all of the pending requests that you must reply to in order for the associated deployment process to continue forward.

44.1 Responding to suspended actions

A notification email is sent to the users who have the correct role to respond to suspended actions.

> ***Reference***
>
> For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide*.

To respond to the suspended action you must fill out the response manually. Select the request in the **Suspended Actions** view and either click the respond icon or right-click on it and select **Respond** to open the **Suspended Action** editor.



Suspended Action

Suspension Cause : Connection to NY File System failed.

Comment

Please check that the underlying infrastructure Item is reachable and retry the operation.

Response

Define the response to the suspended action.

Response :

No Response
Retry
Stop
Resume

Figure 106: The Suspended Action editor

The **Suspension Cause** and **Comment** fields display information about why the process was interrupted. If this is not enough information, check the deployment process instance log for more details.

There are four options to choose from when selecting the response to resolve the problem. Each have a different effect on the processing deployment process instance. Select the corresponding response from the **Response** drop-down list.

- **No Response** maintains the current status of the request and it will remain "pending". The process will not continue forward.
- **Retry** launches the failed action only, enabling you to test it (retry the connection to an infrastructure item. The suspended action is closed if the retry is successful. It is recommended to retry only if the root problem has been addressed.
- **Stop** abandons the failed Action and enables it to cause its parent activity to fail. This leads to a failed deployment process instance.
- **Resume** resumes the process after the failed action. It does not retry the suspended action and continues forward.

To save changes made, click the **Save** button.

44.2 Reviewing processed suspended actions

To view all of the processed suspended actions and their responses, select the **Processed suspended actions** node from **Suspended Actions** in the **Console Explorer**.

The **Suspended Actions** view displays all of the requests that you have replied to. Double-click on a request to open the **Suspended Action** editor and change the response.

45 Validation requests

Required Roles	Environment Release Operator and/or Manager Release Project Manager Application Database Administrator
Required Rights (by category)	Validation Requests
Perspective	Operational Console Management
Access	DROPS Server → Deployment Process → ✓ Validation Requests DROPS Server → Deployment Plans → ✓ Validation Requests

Chapter Summary

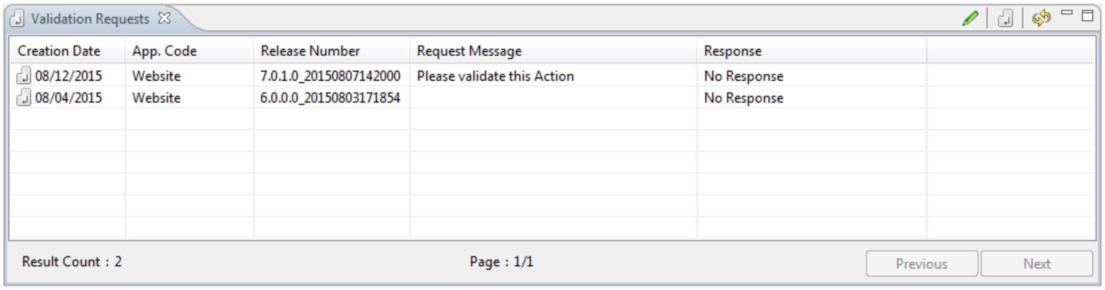
45.1 Responding to requests 297
45.2 Reviewing processed requests 298

If a deployment plan or a deployment process, or any entity included in one, requires validation before it is allowed to move forward, a validation request notification is sent to the user(s) in the designated role(s).

Validation requests are generated during the deployment process if the ✓ **Validation** parameter is activated. They are activated at the highest process-level of the deployment process diagram and at the activity-level. They can be executed before any processes can begin or after they're completed. Required requests must be attended to before deployment can continue and the validator may also be required to provide a reason for either accepting or rejecting the request.

> ⚠ **Important!**
> To view and approve/reject validation requests, a user must have the Release Operator role for the environment in question.

> ***Reference***
> For more information about defining when validation requests are generated, refer to Editing deployment process properties on page 175.and Editing deployment plan properties on page 199.



Creation Date	App. Code	Release Number	Request Message	Response
08/12/2015	Website	7.0.1.0_20150807142000	Please validate this Action	No Response
08/04/2015	Website	6.0.0.0_20150803171854		No Response

Result Count : 2

Page : 1/1

Figure 107: The pending Validation Requests view

The ✓ **Validation Requests** view displays all of the pending requests that you must reply to in order for the associated deployment plan or deployment process to continue forward.

⚠ **Important!**

When a user, different than the one that started the execution of deployment plan, gets involved in the validation process, this user becomes the new executor of all the following deployment plan instances.

45.1 Responding to requests

The validation notification email is sent to the users who have the correct role to respond to validation requests.

Reference

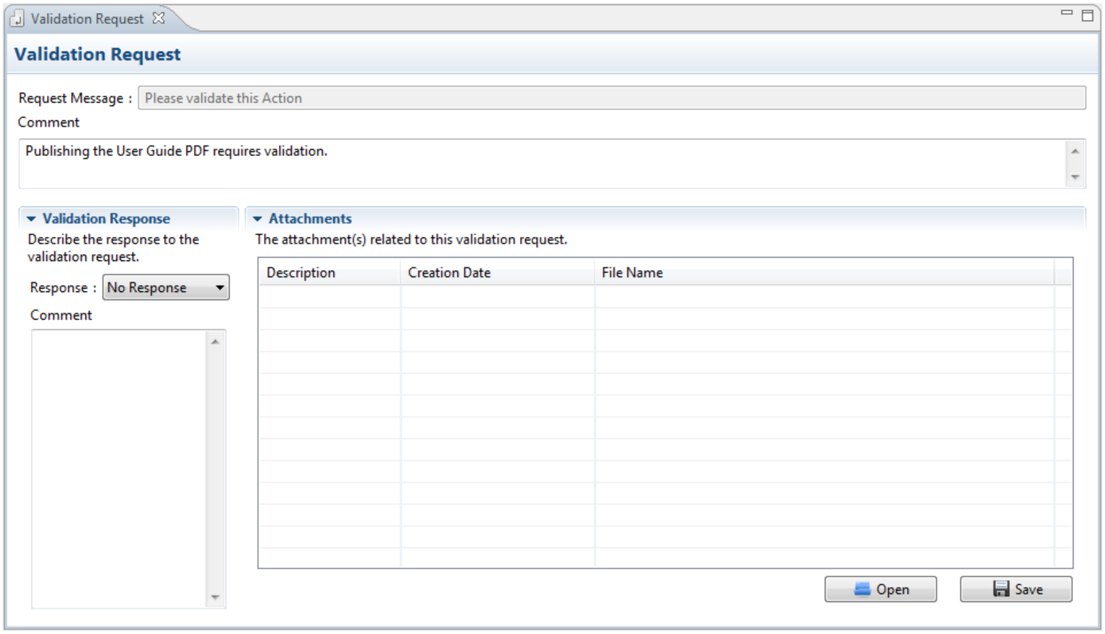
For more information about editing the default content of automatic emails and defining the mail configuration, refer to the *DROPS Configuration Guide*.

There are two ways to respond to a pending validation request:

- Fill in the validation request manually,
- Validate immediately.

45.1.1 Fill out the validation request editor

To fill out the validation request manually, select the request in the ✓ **Validation Requests** view and either click the ✓ respond icon or right-click on it and select ✓ **Respond** to open the ✓ **Validation Request** editor.



Validation Request

Request Message : Please validate this Action

Comment

Publishing the User Guide PDF requires validation.

Validation Response

Describe the response to the validation request.

Response : No Response

Comment

Attachments

The attachment(s) related to this validation request.

Description	Creation Date	File Name

Figure 108: The Validation Request editor

To validate or reject the request to carry out the corresponding step, activity, action, stage etc. select the response from the **Response** drop-down list.

- A **Rejected** request will stop the action and the process or plan will not continue forward.
- An **Approved** request will execute the action and enable the process or plan to continue.

Enter details about the decision made in the **Comment** field. This step may be required to save the approved or rejected request if the user who created it ticked the **Validation Reason is Mandatory** checkbox.

To save changes made, click the **Save** button.

> ⚠ **Important!**
> You can only respond to a validation request once. After the selected option is saved, the **Response** field is read-only.

45.1.2 Validate immediately

To validate a request immediately from the **Validation Requests** view, select it and either click the validate icon or right-click on it and select **Validate immediately**.

Click **OK**.

If the **Validation Reason is Mandatory** checkbox is ticked in the original request, enter details about the decision made in the dialog.

Result The request is approved and the process or plan is allowed to continue.

45.2 Reviewing processed requests

To view all of the processed validation requests, select the **Processed validation requests** node from **Validation Requests** in the **Console Explorer**.

The **Validation Requests** view displays all of the requests that you have replied to. Double-click on a request to open the **Validation Request** editor.

Managing campaigns

Introduction to Campaigns

Campaigns allow you to automate the creation and deployment of a release across one or multiple deployment plans, either executed locally or on a remote node, and in a single step.

Using the DROPS campaigns, you can precisely configure how to deploy your application(s) across one or multiple environments, in parallel or in sequence.

The campaign creation is defined by:

- The import process options set up for a given application,
- The selection or creation of a target release number, and
- The selection of pipeline instances to be executed during the deployment.

Campaigns can be prepared and waiting for a delayed execution, or they can be immediately launched after their creation.

The Campaign module, also called One-Click Deployment, is available in the Operational Console Management perspective and accessed in the Console Explorer node in the Navigator, under the currently connected DROPS Server.

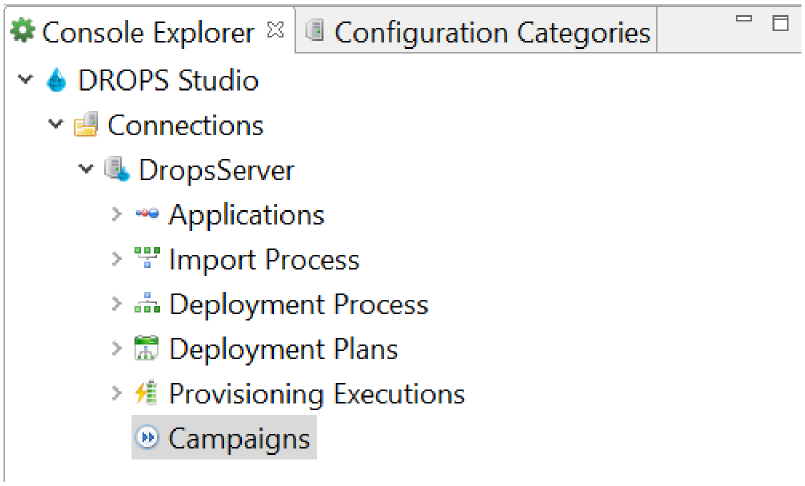


Figure 109: The Operational Console Management Perspective - Campaigns

Once the campaign execution is launched, the import process starts. If the artifacts are successfully imported, the pipeline instances are then started. The campaign then waits for the completion of all the pipeline instances. The execution process, status, errors and results are available and accessible at all times in the Logs tab.

Check the menu on the right for more information about the creation and management of DROPS Campaigns.

46 Campaigns

Required Roles	Environment Release Manager and/or Release Operator Application Repository Manager
Required Rights (by category)	Import Process Instances Deployment Plan Execution Instance
Perspective	Operational Console Management
Access	DROPS Server → Campaigns

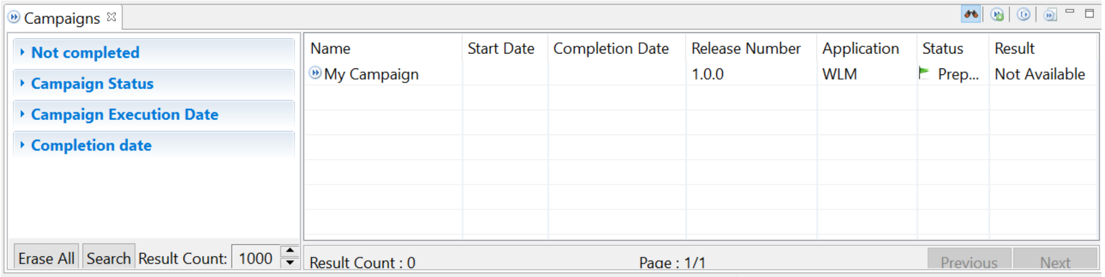
Chapter Summary

46.1 Creating campaigns 301
46.2 Executing campaigns 304
46.3 Understanding campaign execution results 305

A campaign is an identified execution unit that allows to put the following processes together:

- The import of a release,
- The execution of deployment process instances on the newly created release.

Campaigns allow you to execute all these operations at once, in a "One Click" deployment.



Name	Start Date	Completion Date	Release Number	Application	Status	Result
My Campaign			1.0.0	WLM	Prep...	Not Available

Result Count : 0 Page : 1/1

Figure 110: The Campaigns search view

The **Campaigns** search view is accessed from the **Campaigns** node in the **Console Explorer**.

46.1 Creating campaigns

The creation of a campaign in DROPS consists in defining the Import Process options for a given application and the list of deployment instances that are executed to deploy the release created.

Follow the subsequent steps to create a new campaign.

Step 1 To access the Create a new Campaign wizard, either click the Create a new campaign button in the **Campaign** search view or right-click anywhere in the view and select **Create a new campaign**.

Step 2 Define the campaign main properties by setting a **Name** for the campaign. Tick the **Execute the campaign immediately** checkbox to launch the campaign right after its creation.

> ⚠ **Important!**
> Set the campaign main properties carefully. They cannot be changed once the campaign is created.

Click **Next >** to continue.

Step 3 Select the application(s) containing the artifacts to import. At least one application must be selected and it is possible to select multiple applications.

Step 4 Select the release in which to save the artifacts that will be imported.

- If **Current Release** is selected, the release defined in the Current Release field in the application is used to import (displayed in parentheses).
- If **Create a New Release** is selected, a new release will be created using the Release Number Template defined in the application.

  If this option is selected, define the part of a release's version number to increment from the **Increment Part** drop-down menu. Keep in mind that the algorithm will update the other parts automatically. By default, the part is set to the value defined in the the Repository Storage Area settings in the DROPS Server configuration.

- Selecting an **existing**, *active* release from the prepopulated list will import the artifacts to that release.

Click **Next >** to continue.

Step 5 Define the Pipeline selection, which is a list of pipeline instances that are executed in the current campaign.

- Enter a **Pipeline Instance Name**.
- Select a **Pipeline** in the ones available in the drop-down list.

> ℹ **Note**
> The content of the **Pipeline** drop-down list is conditioned by the selected Application. The available pipelines are linked to the related client, they have only one environment group and this group contains only the selected application.

- *[Optional]* Select a target **Execution Node** in the ones available in the drop-down-list.
- Click the **Add** button to include the defined pipeline in the Pipeline Instance list.

> ℹ **Note**
> To remove pipelines from the ones added in the list, select the pipeline(s) and click the **Remove** button.

Click **Next >** to continue.

Step 6 *[Optional]* Enter a description in the **Release Description** field. Tick the **Reset release before importing** checkbox to force the Import Process to clean the content of the release before importing any new artifacts.

Click **Next >** to continue.

Step 7 To associate the instance with a workitem, select the provider then the pre-defined workitem.

> ***Reference***
> For more information about JIRA issues and workitems in DROPS, refer to Workitems on page 401.

Click **Next >** to continue.

Step 8 Select the import strategies to add to populate the selected release.

Filtering the import strategies of the components in the application enables you to select different combinations of artifacts for this release.

- Ticking **No Filter** will automatically import all of the application's artifacts.
- To use an external file that defines which component(s) and import strategy(ies) to use, enter its name in the **External File Name** field. This can also be done in the **Application** editor in the Default Imported Version file name field.
- Select which import strategies to use or not by ⊕ adding or ⊗ removing them from the **User-Defined Filter** section.

  Click the ✎ **Edit** button to edit an import strategy already in the list.

  The list displays the import strategies used and their corresponding components as well as the version to import. The **External Version** refers by default to the ${import.version} substitution variable (***RELEASE**).

  > **Important!**
  > The import strategies selected in this section define the components that will be deployed when this release is launched.

  Only "normal" import strategies are available in this list. Initial import strategies can only be launched by selecting to import **Initial Import Strategies**.

  > ***Reference***
  > For more information about strategies, refer to Import strategies on page 72 and Import strategy type configuration in the appendix.

Click **Finish**.

Result A dialog appears to display the result details of the creation of the campaign and the import, click **OK**. If the campaign was defined to be executed immediately after the creation, the execution starts when you close the dialog. The **Status** and **Results** columns are updated according to the process of the execution.

46.2 Executing campaigns

If the campaign was defined to not be automatically executed after the creation, it appears in the **Campaigns** search view and has to be executed manually.

Executing an existing campaign process instance creates a new instance in the **Import Process Instances** and **Deployment Plan Instances** views.

To execute a campaign, either right-click on a campaign in the **Campaigns** view and select **Execute**, or select a campaign and click the Execute button from the toolbar.

Note

To view the details of the execution while it happens, right-click on a campaign and select **Show Result**, or click the Show Result button to open the campaign results view and watch the steps as they are performed from the Log tab.

The current **Status** of campaigns changes while the campaign is being processed.

Status	Description
Prepared	The campaign was successfully prepared.
Initialization in Progress	The execution started and the initialization stage is in progress.
Release creation in progress	The initialization stage was successfully completed, the Import Process started ans is in progress.
Deployment in progress	The Import Process stage was successfully completed, the Deployment Process started and is in progress.
Completed	The campaign execution is completed.

Table 22: Campaign execution statuses

The current **Result** of campaigns changes while the campaign is being processed.

Result	Description
Not Available	The campaign execution has not started yet or was just launched.
Initialization Failed	The campaign initialization process failed.
Release Creation Failed	The campaign Import Process failed.
Deployment Failed	The Deployment Instance execution failed.
Succeeded	All operations of the campaign execution succeeded.

Table 23: Campaign execution results

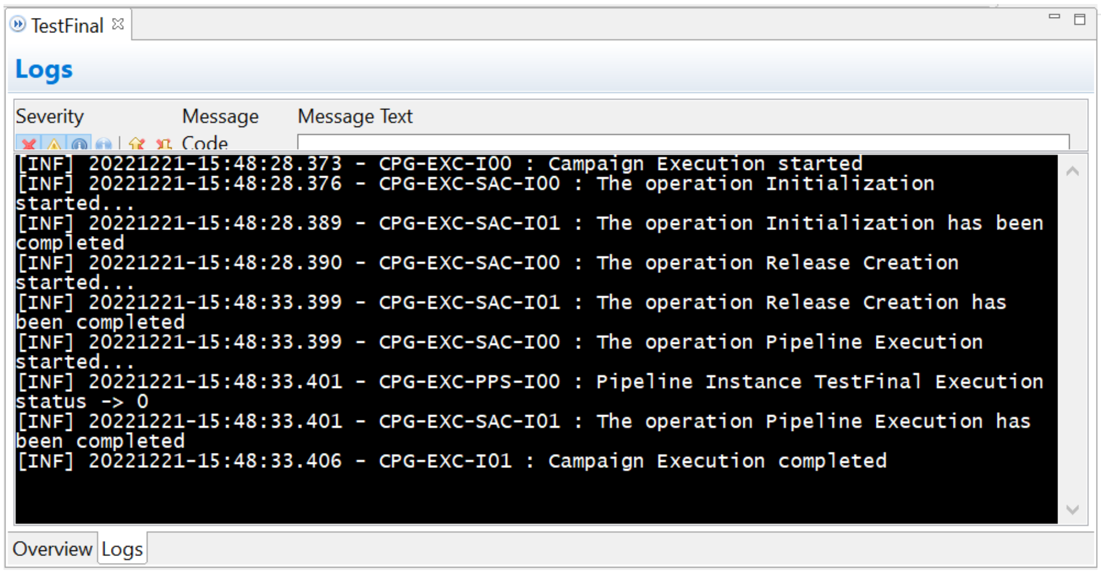
46.3 Understanding campaign execution results

The **Campaign Results** view contains two tabs that each contain different elements and details of the campaign.

- The **Overview** tab displays the uneditable general information about the instance. This tab contains several sub-tabs:
  - The **Campaign Properties** section displays the name of the campaign as defined during its creation.
  - The **Execution Result** section displays the details of the campaign execution, such as the Status, the Result and the Start/End date.
  - The **Release Creation/Update** section displays the details of the release process, weather it was just created or updated. Click on the **Release Creation/Update Process** link to open the campaign Import Results view.
  - The **Pipeline Instance List** section displays the list of deployment instances that were created for this campaign. For each deployment in the list, you can right-click and select **Edit** to open the pipeline instance diagram, or **Show deployment plan results** to open the pipeline Instance Result view.
- The **Log** tab displays the Campaign execution log entries.

View the logs

All the steps and details that occurred during the campaign execution are displayed in the **Logs** tab.



```
[INF] 20221221-15:48:28.373 - CPG-EXC-I00 : Campaign Execution started
[INF] 20221221-15:48:28.376 - CPG-EXC-SAC-I00 : The operation Initialization
started...
[INF] 20221221-15:48:28.389 - CPG-EXC-SAC-I01 : The operation Initialization has been
completed
[INF] 20221221-15:48:28.390 - CPG-EXC-SAC-I00 : The operation Release Creation
started...
[INF] 20221221-15:48:33.399 - CPG-EXC-SAC-I01 : The operation Release Creation has
been completed
[INF] 20221221-15:48:33.399 - CPG-EXC-SAC-I00 : The operation Pipeline Execution
started...
[INF] 20221221-15:48:33.401 - CPG-EXC-PPS-I00 : Pipeline Instance TestFinal Execution
status -> 0
[INF] 20221221-15:48:33.401 - CPG-EXC-SAC-I01 : The operation Pipeline Execution has
been completed
[INF] 20221221-15:48:33.406 - CPG-EXC-I01 : Campaign Execution completed
```

Figure 111: The Campaign results view - Logs

Filter results by **Message Code** or **Message Text**. Enter any combination of search criteria then click the search icon to display the results. To display the complete log, click Search without entering any criteria.

Filter the severity of the results displayed by selecting one or more of the **Severity** icons

- ✖ **Fatal**: only fatal errors are displayed.
- ⚠ **Warning**: only Warnings errors are displayed.
- ℹ **Info**: only Important information is displayed.
- **Verbose**: only details are displayed.

Use the Show Previous and Next icons to quickly move through a long list of entries to find and jump to the fatal errors.

DROPS for ARCAD

Introduction to DROPS for ARCAD

DROPS can import and deploy artifacts that come from and are intended to be deployed to an ARCAD Server.

The **Configuration Explorer** provides access to the entities required to set up the import process. This perspective is the configuration side of DROPS.

Configuring DROPS to import and deploy ARCAD applications can be accomplished in two ways:

1. Manually create the various import and/or deployment elements by selecting the ARCAD-specific options in every element.

2. Automatically create the various import and deployment elements via one streamlined process. The automated creation of the DROPS elements associated with your ARCAD application deployment process will save you time by bypassing the requirement to create and link the individual elements manually.

   - Define the connection information to the source development server and to the target server(s) to which to deploy.
     Configuring ARCAD connections on page 311

   - Generate the ARCAD-type elements to deploy via DROPS.
     Generating ARCAD elements on page 316

You can also manage the target ARCAD layout remotely from DROPS.

- Managing ARCAD remote templates on page 319

To update the content of any menu list, right-click then select **Refresh**.

Reference

Once the required elements are created, no matter how they are created, ARCAD artifacts can be imported into DROPS and deployed using the standard processes.

Introduction to importing artifacts on page 242
Introduction to deploying releases on page 257

If you choose to create all of the elements manually, refer to the following sections to see where ARCAD information needs to be defined throughout the process:

- When defining an import strategy to use to import ARCAD applications, select the **ARCAD strategy** type.
  Import strategy type configuration on page 433

- When defining an infrastructure item to which to deploy, select **ARCAD Server** for the type, which requires specific extended configuration.
  Infrastructure item type extended configuration on page 455

How ARCAD applications are mapped and deployed in DROPS

ARCAD applications vs DROPS applications

Because the concept of an application exists in both ARCAD and DROPS, it's important to understand the difference and describe how they are mapped.

- In ARCAD, an application is a set of libraries, referred as application operational libraries. It is the versioned, hierarchical structure of the libraries (layout) and their contents (objects) that will be imported into DROPS as artifacts and deployed.
- In DROPS, an application is a container that regroups different technical sub-parts to which together make up a functional user-oriented application. For example, a DROPS application can be made up of a native IBM i part that provides back-office features, and by a Web server that provides the front-end. The union of these two parts is called an application in DROPS.

Because of this difference, it is not (always) possible to create a 1-to-1 relationship between an ARCAD application and a DROPS application. An ARCAD application is only a sub-part of a DROPS application.

> ⚠ **Important!**
> When imported, an ARCAD application (version) is mapped to a DROPS component, which is the technical sub-part of a DROPS application.

ARCAD runtime vs DROPS infrastructure items

The technical location in which ARCAD applications are stored, and how they are arranged, must be mapped to DROPS infrastructure items.

- On IBM i, the layout of ARCAD applications is critical. The position of the objects must remain intact when deployed. This layout is managed by the ARCAD Runtime.

  When an ARCAD version is ready to be deployed using DROPS, a SAVF is generated that contains the objects to be deployed and some additional information in ARCAD lists. These lists identify which objects have been updated, added or deleted. They also contain information about the target location (the libraries) to where these artifacts are to be transferred to maintain the integrity of the hierarchical structure of the libraries (layout).

- In DROPS, the logical targets are called infrastructure items. An infrastructure item is the top-level technical location where the artifacts in a component will be deployed. This is the entity that must receive the layout information from the SAVF to reproduce the hierarchy system on the target so that the ARCAD application remains intact.

> ⚠ **Important!**
> To deploy an ARCAD application that requires a specific layout, a DROPS infrastructure item must point to an ARCAD remote application.

Because an ARCAD application can be deployed to multiple infrastructure items, on multiple IBM i machines, the template ARCAD structure to respect is centralized so that it can be reproduced on multiple target systems.

Tips for working with multiple ARCAD instances

No ARCAD libraries can be in system LIBL, check QSYSLIBL.

Ensure that all subsystems involved, typically ARCAD and DROPS, do NOT have any ARCAD libraries in the LIBL. Check the subsystem description using DSPSBSD ARCAD_SYS/ARCAD (option 1).

You need multiple macro libraries if you are using the DROPS4ARC library, since the ARCAD library is embedded in the macro. To switch the library, go into the specific instance of ARCAD and use AWRKMACCMD MACRO(DROPS4ARC/*ALL) on the appropriate library name and select option 14 on all macros, then hit enter.

Note

If you're using the newer A4DEPLOY and A4EXPORT macros that are in ARCAD itself you do not have to worry about the macros.

47 Configuring ARCAD connections

Required Roles	n/a
Required Rights (by category)	Applications
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) ARCAD Configuration

Chapter Summary

47.1 Connecting to the ARCAD development server 311
47.2 Defining connections to target ARCAD-Servers 313

The ARCAD configuration process allows you to automatically create DROPS elements, which will be pre-populated and linked to one another, successfully creating a streamlined DROPS setup in one quick operation. The elements can then be used to import ARCAD applications and deploy them on IBM i.

The prerequisites for this process are:

1. a connection to a predefined development ARCAD Server from which to import the ARCAD application,
2. a connection to at least one predefined target ARCAD Server to which to link the infrastructure item to use to deploy is also required *if you choose to automatically create the target environment*.

> **Important!**
> DROPS can only import ARCAD applications from one development server. Only one source server can be defined, however multiple target systems, to which to deploy the application, can be defined.

> **Note**
> The process to generate ARCAD elements can begin only after these connections are defined.

47.1 Connecting to the ARCAD development server

When generating ARCAD elements, you can automatically create the DROPS component which will retrieve versions and import the ARCAD artifacts required to deploy your application.

To automatically populate the source IBM i system in the new component, you must define the connection to the development IBM i system. This connection information will be carried over to the new component's ARCAD extended configuration.

> **Important!**
> Connecting to the ARCAD development server is required to generate ARCAD elements. This connection will be automatically pre-populated in the ARCAD-type import strategy, from which you will import your ARCAD

⚠ application, when it is generated.

Reference
Components on page 67
Component type extended configuration on page 432

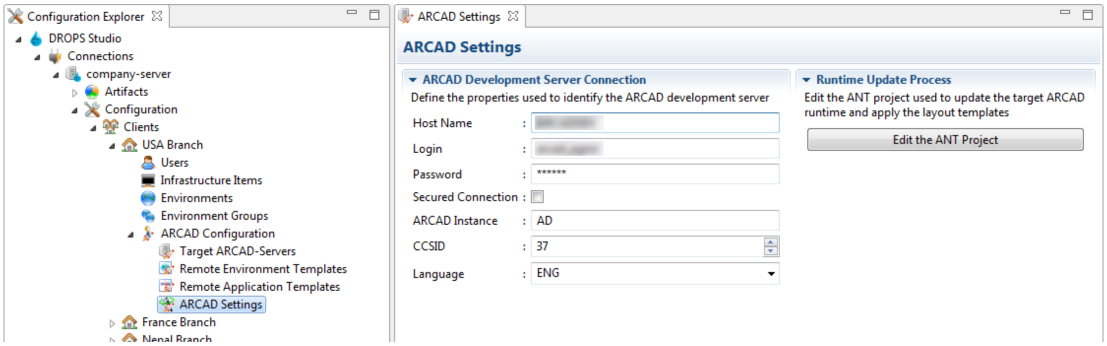
Follow the subsequent steps to configure or edit the connection to the source development server.

Step 1 Select **ARCAD Settings** from the **ARCAD Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

Reference
For more information about managing multiple clients, refer to Multiple clients on page 94.

Step 2 Define the connection information.



ARCAD Settings

ARCAD Development Server Connection
Define the properties used to identify the ARCAD development server

Host Name : [redacted]
Login : [redacted]
Password : ******
Secured Connection : □
ARCAD Instance : AD
CCSID : 37
Language : ENG

Runtime Update Process
Edit the ANT project used to update the target ARCAD runtime and apply the layout templates

Figure 112: Connect to the ARCAD development server

Host Name

Enter the name or IP address of the IBM i partition hosting the server.

Login/Password

Enter a valid user login and password. This user must be declared on the server and will be the user that will execute any operations on the IBM i. The password should be encoded in Base64.

Secured Connection

Tick this checkbox if the server requires a secured TLS/SSL connection.For more information about ARCAD secure connections, refer to the *ARCAD SSL Configuration Guide*.

ARCAD Instance

Enter the ID of the instance to reach. If you are using ARCAD the instance is **AD** by default.

CCSID

Enter a valid CCSID to connect to the server.

Language

Select the language to use on the server (FRA or ENG).

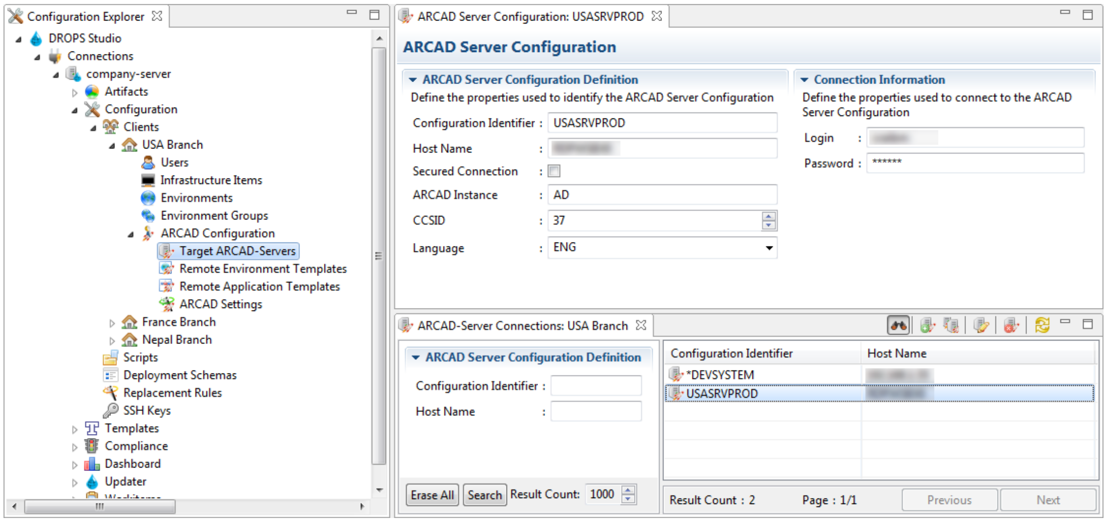
Step 3 Save the changes (, Ctrl+S or **File** > **Save**).

47.2 Defining connections to target ARCAD-Servers

> **Note**
>
> Creating connections to target ARCAD-Servers is not required to generate ARCAD elements but it is required if you want to automatically pre-populate the ARCAD-type infrastructure item.

When generating ARCAD elements, you can automatically create the infrastructure item to which the ARCAD application will be distributed.

To automatically populate a target IBM i system in the new infrastructure item, you must define the connection to that system in the **ARCAD-Server Connections** view.



ARCAD Server Configuration: USASRVPROD

ARCAD Server Configuration

ARCAD Server Configuration Definition

Define the properties used to identify the ARCAD Server Configuration

Configuration Identifier : USASRVPROD
Host Name : [redacted]
Secured Connection : □
ARCAD Instance : AD
CCSID : 37
Language : ENG

Connection Information

Define the properties used to connect to the ARCAD Server Configuration

Login : [redacted]
Password : ******

ARCAD-Server Connections: USA Branch

ARCAD Server Configuration Definition

Configuration Identifier :
Host Name :

Configuration Identifier	Host Name
*DEVSYSTEM	[redacted]
USASRVPROD	[redacted]

Figure 113: Connect to target ARCAD servers

47.2.1 Creating connections to target ARCAD-Servers

There are two ways to create connections to target ARCAD-Servers:

1. Create a brand new connection using the **Create a new ARCAD-Server Connection** wizard.
2. Duplicate an existing connection. Duplicating creates a copy of an existing connection that can be modified.

47.2.1.1 Creating a new connection

Follow the subsequent steps to configure or edit a connection to a target ARCAD server.

Step 1 Select **Target ARCAD Servers** from the **ARCAD Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

> ***Reference***
> For more information about managing multiple clients, refer to Multiple clients on page 94.

Step 2 To access the **Create a new ARCAD-Server Connection** wizard, either click the create icon in the search view or right-click anywhere in the view and select **Create a new ARCAD-Server connection**.

Step 3 Define the connection information.

Configuration ID

Enter a unique name to identify this connection to the server.

This is the name that will appear when you are selecting the target to which to deploy.

Host Name

Enter the name or IP address of the IBM i partition hosting the server.

Login/Password

Enter a valid user login and password. This user must be declared on the server and will be the user that will execute any operations on the IBM i. The password should be encoded in Base64.

Step 4 Click **Finish**.

Result The new connection is created and the editor opens automatically.

> **Important!**
> It is important to continue editing the new connection right away to complete the configuration.

Step 5 Verify that the connection information is complete.

Secured Connection

Tick this checkbox if the server requires a secured TLS/SSL connection.For more information about ARCAD secure connections, refer to the *ARCAD SSL Configuration Guide*.

ARCAD Instance

Enter the ID of the instance to reach. If you are using ARCAD the instance is **AD** by default.

CCSID

Enter a valid CCSID to connect to the server.

Language

Select the language to use on the server (FRA or ENG).

Step 6 Save the changes (, Ctrl+S or **File** > **Save**).

Result New connections are available in the **Target ARCAD-Servers** view and can be used to generate new ARCAD elements.

47.2.1.2 Duplicate a connection

Duplicating a connection to a target ARCAD Server is a quick way to create a new connection to use when generating ARCAD elements with similar properties as the original connection.

Follow the subsequent steps to duplicate an existing connection to a target ARCAD Server.

Step 1 To open the **Duplicate an ARCAD Server connection** wizard, either select an existing connection in the **Target ARCAD-Servers** view then click the duplicate icon, or right-click on it and select **Duplicate**.

Step 2 Enter a new Configuration ID for the connection. This ID must be unique.

Step 3 Click **Finish**.

Result New connections are available in the **ARCAD Target Servers** view and can be used to generate new ARCAD elements.

47.2.2 Deleting connections to target ARCAD servers

Warning!
Deleted connections cannot be accessed, recovered, or used to automatically populate new environments created by generating ARCAD applications. Deleting a connection does not effect environments that already used it when they were generated.

To delete a connection, search for it in the search view. Either right-click on it and select **Delete** or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the connection.

48 Generating ARCAD elements

Required Roles	n/a
Required Rights (by category)	Applications Clients Environments Infrastructure Items
Perspective	Configuration Management
Access	DROPS Server → Artifacts → *right-click on Applications* → Generate an ARCAD application

Important!
Before you can generate ARCAD elements automatically, you must configure the connection to the source ARCAD Server.

Reference
Connecting to the ARCAD development server on page 311

The ARCAD elements created automatically by generating an ARCAD application are:

- an ARCAD-type application which contains:
  - An ARCAD-type component, which will house all of the ARCAD application's artifacts, pre-configured to connect to the development system and import the ARCAD application.
  - An ARCAD-type import strategy for the component, predefined to import all of the elements required for the ARCAD application.
- [*Optional*] an ARCAD-type infrastructure item, to which you will deploy your ARCAD application, pre-configured to connect to the target system.
- [*Optional*] an ARCAD-type environment linked to the infrastructure item and the application, for which you can create a deployment process.

Note
The ARCAD-specific extended configuration for each of these entities will also be pre-configured.

This process can also automatically create and provision remote applications and environments on the fly. Provisioning remotely eliminates the need to connect to the target machine(s) to configure the target environment to receive the deployed application. Everything needed to reproduce the layout required for the ARCAD application on the target system, including the operational libraries and library lists are included in these entities.

- [*Optional*] a remote application template based on which DROPS will provision your target runtime with the appropriate resources and layout required to receive the ARCAD application.
- [*Optional*] a remote environment template based on which DROPS will provision your target runtime with the appropriate resources and layout required to receive the ARCAD application.

Follow the subsequent steps to automatically generate ARCAD elements to use to import and deploy ARCAD applications.

Step 1 Right-click the **Applications** node in the **Configuration Explorer**, then select **Generate an ARCAD application**.

Step 2 Select the ARCAD application to import. Click the browse button to browse for and select the application. The list is pre-populated with all of the applications available on the ARCAD development server.

The extended configuration of the ARCAD-type component that this process will automatically generate will be pre-populated with the connection information for the source system and the application you select here.

Click **Next >** to continue.

Step 3 If you want to automatically recreate the layout of the source application on the target system, to avoid manually configuring it to receive the application once it is deployed, tick the **Create an ARCAD remote application template** checkbox.

Step 4 [*Optional*] Tick the **Create an ARCAD remote application template** checkbox to automatically recreate the layout of the source application on the target system. This option is useful to avoid manually configuring the target system to receive the application once it is deployed.

The ARCAD remote application template will contain the runtime layout of the selected ARCAD application.

> **Note**
> You can also define the layout of the target application manually and not use templates at all. If you do not want to create the remote application template automatically, click **Finish**.

Click **Next >** to continue.

Step 5 [*Optional*] Tick the **Create an ARCAD remote environment template** checkbox to automatically recreate the layout of the source environment on the target system.

Click the browse button to browse for and select the environment linked to the selected ARCAD application. The list is pre-populated with all of the environments available for the application selected before.

The ARCAD remote environment template will contain the runtime layout of the selected ARCAD environment.

> **Note**
> You can also define the layout of the target environment manually and not use templates at all. If you do not want to create the remote environment automatically, click **Finish**.

Click **Next >** to continue.

Step 6 [*Optional*] Tick the **Create an infrastructure item** checkbox to automatically create a dedicated infrastructure item.

An infrastructure item is required to deploy your application but you can create them manually. If you create it automatically here, the item will be pre-populated with the connection information for the **Target System** you select. All of the target systems defined in the **Target ARCAD-Servers** configuration view are available here.

> ***Reference***
> For more information about connecting to the target servers, refer to Defining connections to target ARCAD-Servers on page 313.

> **Note**
> If you do not want to create the infrastructure item, click **Finish**.

Click **Next >** to continue.

Step 7 [*Optional*] Define the environment to use during the deployment process. The environment defined here will be automatically linked to the new infrastructure item and application.

You can either create a new environment or use an existing environment. In the **Create new or use existing** field select:

- ***CREATE** to create a new environment, then define the **Type** (ref.: Entity types).
- ***USE** to use an existing environment, then click the browse button to select the environment. The list is populated with all of the environments defined in DROPS.

> **Note**
> If you do not want to define the environment, you must select ***NONE** in the **Create new or use existing field** and click **Finish**.

Step 8 Click **Finish**.

Result The process is launched and all the selected elements are created. When finished, the log is displayed. The log lists all of the elements that have been created.

Close the log file and navigate to the various elements to work with them individually as you would any other DROPS element.

> ***Reference***
> Applications on page 60
> Components on page 67
> ARCAD component extended configuration
> Import strategies on page 72
> The ARCAD Enhanced import strategy
> Infrastructure items on page 108
> ARCAD Runtime extended configuration
> Environments on page 96
> ARCAD environment extended configuration
> Managing ARCAD remote templates on page 319

49 Managing ARCAD remote templates

Required Roles	n/a
Required Rights (by category)	n/a
Perspective	Configuration Management
Access	DROPS Server → Configuration → (*Client* →) ARCAD Configuration

Chapter Summary

49.1 Working with remote application templates 319
49.2 Working with remote environment templates 324
49.3 Working with the target ARCAD runtime updater 328
49.4 Updating an ARCAD target's runtime 331

To deploy an ARCAD application runtime you must define the target operational libraries to which the ARCAD objects will be deployed on the target system. Previously, this pre-configuration had to be done manually on your target system prior to deploying ARCAD applications. Now, this process can be automated using templates that can either be manually applied to the target or automatically during the deployment process. ARCAD remote templates reproduce the original layout of the objects deployed on the target so that all of the objects are deployed to the location where they are required.

These templates are referenced in ARCAD-type infrastructure items' extended configuration.

49.1 Working with remote application templates

ARCAD applications are the highest-level organizational entities in which development objects are organized. They are broken down into a set of libraries, referred as application operational libraries, in which all of the functional objects for the application are managed. The layout of these libraries is essential. Applications also include library lists. The second organizational layer in the layout of an application are its child environments.

To preserve the integrity of an application's source layout when deploying an ARCAD application, you can create remote application templates which will reproduce the structure of the application on the target server.

> **Note**
> Remote application templates must reference at least one remote environment template.

49.1.1 Searching for remote application templates

Follow the subsequent steps to search for remote application templates.

Step 1 Select **Remote Application Templates** from the **ARCAD Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

> ***Reference***
> For more information about managing multiple clients, refer to Multiple clients on page 94.

Step 2 When the search view is opened, the list of templates is empty. Use the search criteria sections to filter the search.

- To search for an existing template by its **Reference ID**, enter all or part of the information in the corresponding field.
- To search for an existing template by the **Application Code** of the ARCAD application it will create, enter all or part of the information in the corresponding field.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.

Result The **ARCAD Remote Application Templates** view displays all of the templates defined for the current server (and for the selected client, if applicable) that meet the search criteria.

49.1.2 Creating remote application templates

There are two ways to create remote application templates:

1. Generate the template automatically using the process defined in Generating ARCAD elements on page 316
2. Create a new remote application template using the **Create a new ARCAD remote application** template wizard.
3. Import an application's layout to create a template.

49.1.2.1 Create a new remote application template

Follow the subsequent steps to create a new remote application template.

Step 1 To access the **Create a new ARCAD remote application template** wizard, either click the create icon in the **ARCAD Remote Application Templates** view or right-click anywhere in the view and select **Create a new ARCAD remote application template**.

Step 2 Define the new template’s *Reference ID.

From the *Application Code field, select the ARCAD application to reproduce. The list is pre-populated with all of the applications available in the ARCAD development system, identified in the **ARCAD Settings**.

> ***Reference***
> For more information about this machine, refer toConnecting to the ARCAD development server on page 311).

These values can be edited later.

Step 3 Click **Finish**.

Result New remote application templates are available in the **ARCAD Remote Application Templates** view after refreshing the list.

> **Important!**
> You must continue to configure the template by opening the editor and defining the libraries, before the template is finished.

49.1.2.2 Import an application's layout to create a template

Follow the subsequent steps to import the configuration of an existing ARCAD application to create a new template. The remote application template's properties will be automatically set up using the properties of the selected ARCAD application including the operational library list and the library list.

Step 1 To open the **Import an ARCAD application** wizard, click the import icon in the **ARCAD Remote Application Template** view or right-click anywhere in the view and select **Import from an ARCAD application**.

Step 2 Define the new template's *Reference ID.

From the *Application Code field, select the ARCAD application to reproduce. The list is pre-populated with all of the applications available in the ARCAD development system, identified in the **ARCAD Settings**.

> ***Reference***
> For more information about this machine, refer toConnecting to the ARCAD development server on page 311).

These values can be edited later.

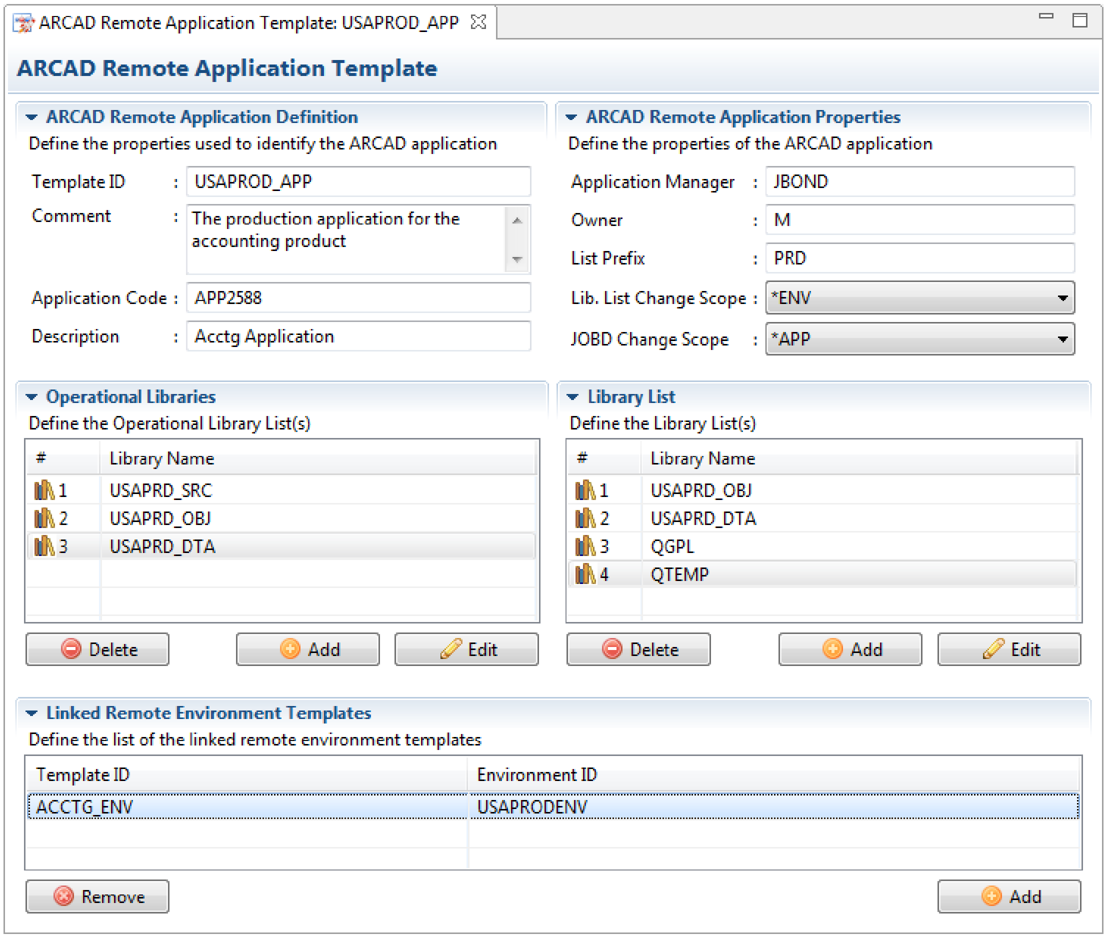
Step 3 Click **Finish**.

Result New remote application templates are available in the **ARCAD Remote Application Templates** view after refreshing the list.

The source application's properties and libraries have been imported and can be viewed and modified in the editor.

49.1.3 Editing remote application templates

To edit remote application template details, either right-click on it in the **ARCAD Remote Application Templates** view then select **Edit**, select the item then click the edit icon or double-click it. The **ARCAD Remote Application Template** editor is opened where all of the entity's details are managed.



ARCAD Remote Application Template: USAPROD_APP

ARCAD Remote Application Template

ARCAD Remote Application Definition

Define the properties used to identify the ARCAD application

Template ID : USAPROD_APP

Comment : The production application for the accounting product

Application Code : APP2588

Description : Acctg Application

ARCAD Remote Application Properties

Define the properties of the ARCAD application

Application Manager : JBOND

Owner : M

List Prefix : PRD

Lib. List Change Scope : *ENV

JOBD Change Scope : *APP

Operational Libraries

Define the Operational Library List(s)

#	Library Name
1	USAPRD_SRC
2	USAPRD_OBJ
3	USAPRD_DTA

Library List

Define the Library List(s)

#	Library Name
1	USAPRD_OBJ
2	USAPRD_DTA
3	QGPL
4	QTEMP

Linked Remote Environment Templates

Define the list of the linked remote environment templates

Template ID	Environment ID
ACCTG_ENV	USAPRODENV

Figure 114: The Remote Application Template editor

The information that defines remote application templates can be edited at any time.

Field	Description
***Reference ID**	A unique string that identifies applications templates. This label is used throughout DROPS to select the entity.
Comment	A comment used to describe this application.
***Application Code**	The ARCAD application code, as declared in the source runtime.

Table 24: Edit application templates

Field	Description
Description	The application text displayed in the source runtime. This label is used throughout DROPS to select the entity.
Application Manager	The manager of this application as declared in the runtime.
Owner	The owner of the application as declared in the runtime.
List Prefix	The prefix used for the associated version libraries.
Lib. List Change Scope	Select the Library List Change scope. {*NONE|*APP|*ENV|*VERS}
JOBD Change Scope	Select the JOBD Library List Change scope. {*NONE|*APP|*ENV}

Table 24: Edit application templates

The operational libraries and the library lists are managed the same way. Each section allows you to create, delete and edit libraries or lists. When there are multiple entries, you can drag and drop them to reorder them.

> **Note**
> - The **Library Name** is a max 10-characters length value.
> - All library names are automatically converted into upper case.
> - Each library name must be unique.

> ***Reference***
> For more information about operational libraries and library lists, refer to the ARCAD documentation.

Save the changes (💾, Ctrl+S or **File** > **Save**).

49.1.4 Linking remote environment templates

To link a remote environment template to a remote application template, click on the **Add** button in the **Linked Remote Environment Templates** section. The list of all the available ARCAD remote environment templates is displayed. *Ctrl+click* to select multiple environment templates to associate with the application template.

> **Note**
> If a remote application template and a remote environment template were generated automatically, the environment template will automatically be linked to the parent application template.

To remove an environment template, select it and click **Remove**.

49.1.5 Linking remote application templates to infrastructure items

Link an ARCAD remote application template to an infrastructure item to recreate the layout of the ARCAD application on its target system.

> **Note**
> ARCAD remote application templates can only be linked to ARCAD Runtime-type infrastructure items.

In the **Infrastructure Item** editor, click the **Edit configuration** link in the Extended Configuration field. In the item's extended configuration editor, select the ARCAD remote application template. Click the browse button to access the dialog that is pre-populated with all of the templates available on the current server.

> ***Reference***
> For more information about ARCAD extended configuration, refer to Infrastructure item type extended configuration on page 455.

49.1.6 Deleting remote application templates

> ***Warning!***
> Deleted remote application templates cannot be accessed or recovered. Deleting application templates does not affect any remote environment template that may have been referenced, but those templates will no longer have a parent application template.

To delete a remote application template, either right-click on it in the **ARCAD Remote Application Templates** view, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the template.

49.2 Working with remote environment templates

Each ARCAD application contains one or more dedicated workspaces used for specific needs in its maintenance cycle. These workspaces are called ARCAD environments. Just like their parent applications, ARCAD environments are broken down into operational libraries and include library lists.

Remote environment templates are linked to parent remote application templates in the template's editor.

49.2.1 Searching for remote environment templates

Follow the subsequent steps to search for remote environment templates.

Step 1 Select **Remote Environment Templates** from the **ARCAD Configuration** node in the **Configuration Explorer**.

If you are in multi-client mode, select the client first.

> ***Reference***
> For more information about managing multiple clients, refer to Multiple clients on page 94.

Step 2 When the search view is opened, the list of templates is empty. Use the search criteria sections to filter the search.

- To search for an existing template by its **Reference ID**, enter all or part of the information in the corresponding field.
- To search for an existing template by the **Environment ID** of the ARCAD environment it will create, enter all or part of the information in the corresponding field.

Enter any combination of the above search criteria, then click the **Search** button to display the results. To display the complete list, click the **Search** button without entering any search criteria.

To display all of the items by default each time the search view is opened, select the auto search icon.

Result The **ARCAD Remote Environment Templates** view displays all of the templates defined for the current server (and for the selected client, if applicable) that meet the search criteria.

49.2.2 Creating remote environment templates

There are two ways to create remote environment templates:

1. Generate the template automatically using the process defined in Generating ARCAD elements on page 316
2. Create a new remote environment template using the **Create a new ARCAD remote environment** wizard.
3. Import an environment's layout to create a template.

49.2.2.1 Create a new remote environment template

Follow the subsequent steps to create a new remote environment template.

Step 1 To access the **Create a new ARCAD remote environment template** wizard, either click the create icon in the **ARCAD Remote Environment Templates** view or right-click anywhere in the view and select **Create a new ARCAD remote environment template**.

Step 2 Define the *Reference ID and *Environment ID. These values can be edited later.

Step 3 Click **Finish**.

Result New remote environment templates are available in the **ARCAD Remote Environment Templates** view after refreshing the list and can be referenced in remote application templates.

> ⚠ **Important!**
> You must continue to configure the template by opening the editor and defining the libraries, before the template is finished.

49.2.2.2 Import an environment's layout to create a template

Follow the subsequent steps to import the configuration of an existing ARCAD environment to create a new template. The remote environment template's properties will be automatically set up using the properties of the selected ARCAD environment including the operational library list and the library list.

Step 1 To open the **Import an ARCAD environment** wizard, click the import icon in the **ARCAD Remote Environment Templates** view or right-click anywhere in the view and select **Import from an ARCAD environment**.

Step 2 Define the new template’s *Reference ID and *Environment ID. These values can be edited later.

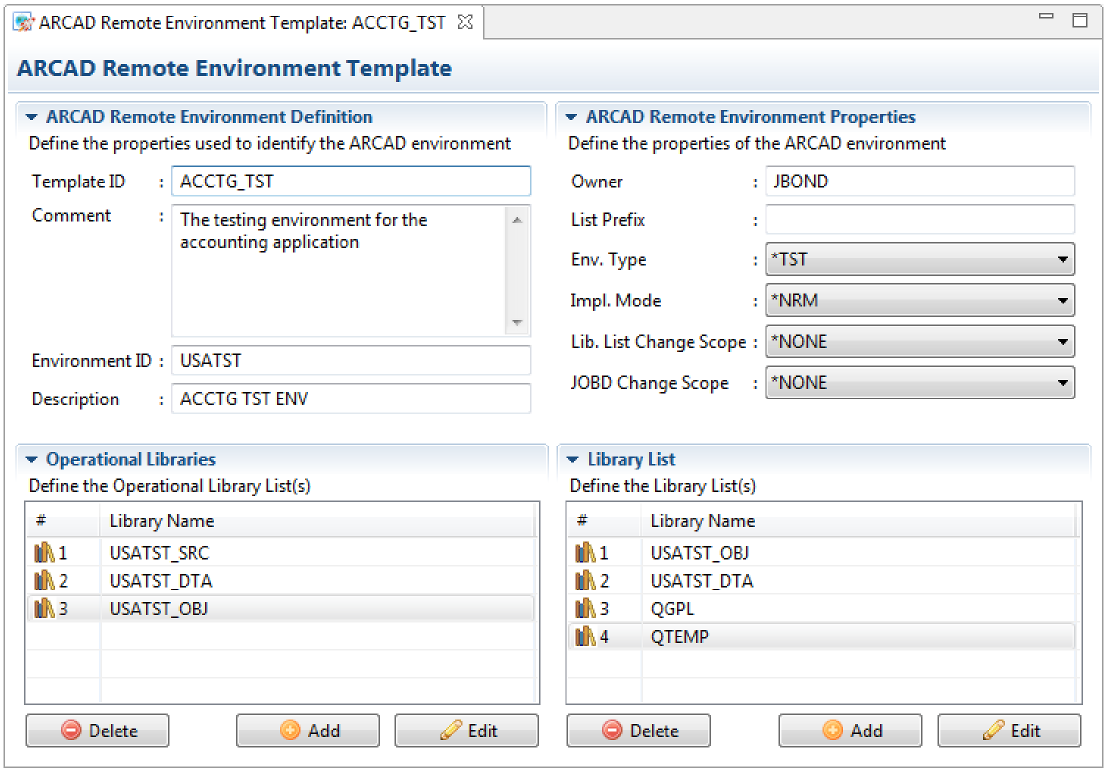
Step 3 Click **Finish**.

Result New remote environment templates are available in the **ARCAD Remote Environment Templates** view after refreshing the list and can be referenced in remote application templates.

The source environment’s properties and libraries have been imported and can be viewed and modified in the editor.

49.2.3 Editing remote environment templates

To edit remote environment template details, either right-click on it in the **ARCAD Remote Environment Templates** view then select **Edit**, select the item then click the edit icon or double-click it. The **ARCAD Remote Environment Templates** editor is opened where all of the entity's details are managed.



ARCAD Remote Environment Template: ACCTG_TST

ARCAD Remote Environment Template

ARCAD Remote Environment Definition

Define the properties used to identify the ARCAD environment

Template ID : ACCTG_TST

Comment : The testing environment for the accounting application

Environment ID : USATST

Description : ACCTG TST ENV

ARCAD Remote Environment Properties

Define the properties of the ARCAD environment

Owner : JBOND

List Prefix :

Env. Type : *TST

Impl. Mode : *NRM

Lib. List Change Scope : *NONE

JOBD Change Scope : *NONE

Operational Libraries

Define the Operational Library List(s)

#	Library Name
1	USATST_SRC
2	USATST_DTA
3	USATST_OBJ

Library List

Define the Library List(s)

#	Library Name
1	USATST_OBJ
2	USATST_DTA
3	QGPL
4	QTEMP

Figure 115: The Remote Environment Template editor

The information that defines remote environment templates can be edited at any time.

Field	Description
***Reference ID**	A unique string that identifies environment templates. This label is used throughout DROPS to select the entity.
Comment	A comment used to describe this environment.
***Environment ID**	The ARCAD environment ID, as declared in the source runtime.
Description	The environment text displayed in the source runtime. This label is used throughout DROPS to select the entity.
Owner	The owner of the environment as declared in the runtime.
List Prefix	The prefix used for the associated version libraries.
Env. Type	Select the environment type. {*DEV|*TST|*PRD}
Impl. Mode	Select the implementation mode. {*NRM|*EXT|*DTA}
Lib. List Change Scope	Select the Library List Change scope. {*NONE|*APP|*ENV|*VERS}

Table 25: Edit environment templates

Field	Description
JOBD Change Scope	Select the JOBD Library List Change scope. {*NONE|*APP|*ENV}

Table 25: Edit environment templates

The operational libraries and the library lists are managed the same way. Each section allows you to ⊕ create, ✖ delete and ✎ edit libraries or lists. When there are multiple entries, you can drag and drop them to reorder them.

Note

- The **Library Name** is a max 10-characters length value.
- All library names are automatically converted into upper case.
- Each library name must be unique.

Reference

For more information about operational libraries and library lists, refer to the ARCAD documentation.

Save the changes (, Ctrl+S or **File** > **Save**).

49.2.4 Deleting remote environment templates

Warning!

Deleted remote environment templates cannot be accessed or recovered. Deleted templates will also be deleted from any remote application template that referenced them.

To delete a remote environment template, either right-click on it in the **ARCAD Remote Environment Templates** view, then select **Delete** or select it, then click the delete icon. Click **OK** to confirm or click **Cancel** to keep the template.

49.3 Working with the target ARCAD runtime updater

The process in charge of applying the information from ARCAD remote application and environment templates to the target systems is carried out by dedicated ANT tasks. The process is the same for every infrastructure item and every template.

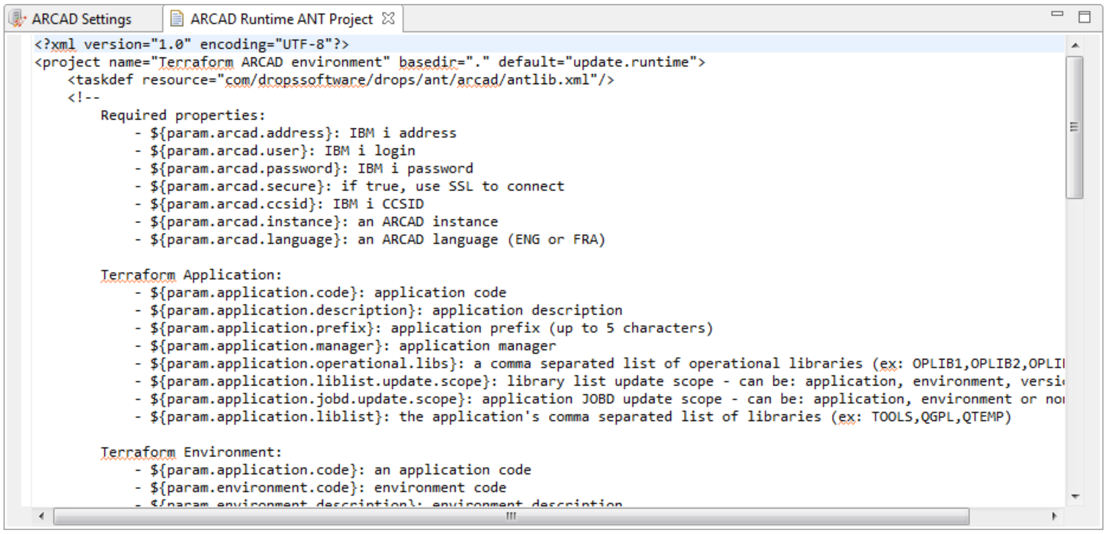
Because each technical target is only reachable using the DROPS Agent, it is important to use a standard Remote Execution Process to guarantee that all the tasks are executed the same way. Thanks to this centralized process manager, all the secured execution mechanisms are ensured.

The setup and maintenance of target ARCAD runtimes is done using a single ANT project that calls these dedicated ANT Tasks.

Important!
You must have advanced notions in ANT coding to successfully modify this ANT project.

From the **ARCAD Settings** view, click the **Edit the ANT Project** button to modify the ANT project in charge of maintaining the ARCAD layout via templates on target systems.

The ANT project is edited in a local editor.



```
<?xml version="1.0" encoding="UTF-8"?>
<project name="Terraform ARCAD environment" basedir="." default="update.runtime">
    <taskdef resource="com/dropssoftware/drops/ant/arcad/antlib.xml"/>
    <!--
        Required properties:
            - ${param.arcad.address}: IBM i address
            - ${param.arcad.user}: IBM i login
            - ${param.arcad.password}: IBM i password
            - ${param.arcad.secure}: if true, use SSL to connect
            - ${param.arcad.ccsid}: IBM i CCSID
            - ${param.arcad.instance}: an ARCAD instance
            - ${param.arcad.language}: an ARCAD language (ENG or FRA)

        Terraform Application:
            - ${param.application.code}: application code
            - ${param.application.description}: application description
            - ${param.application.prefix}: application prefix (up to 5 characters)
            - ${param.application.manager}: application manager
            - ${param.application.operational.libs}: a comma separated list of operational libraries (ex: OPLIB1,OPLIB2,OPLI
            - ${param.application.liblist.update.scope}: library list update scope - can be: application, environment, versi
            - ${param.application.jobd.update.scope}: application JOBD update scope - can be: application, environment or no
            - ${param.application.liblist}: the application's comma separated list of libraries (ex: TOOLS,QGPL,QTEMP)

        Terraform Environment:
            - ${param.application.code}: an application code
            - ${param.environment.code}: environment code
            - ${param.environment.description}: environment description
```

Figure 116: The ARCAD runtime ANT project editor

Before execution, the properties defined in the infrastructure item's extended configuration will be injected into the ANT Execution Context to be usable by the ANT project via variables:

- the target ARCAD-Server
- the ARCAD remote application template (and any linked remote environment template) to use to update the target server

Additionally, the following properties may be used in the project.

Note
All the properties' values are already formatted to be directly used by the dedicated ANT Tasks.

ARCAD Server Config.Properties

Property	Description
${config.server.host}	The IP Address or DNS of the server
${config.server.login}	A valid ARCAD user that will be used to execute all the operation
${config.server.password}	The related password
${config.server.secured}	Indicates if the connection use TLS or not
${config.server.ccsid}	The CCSID of the connection
${config.server.instance}	The target ARCAD instance
${config.server.language}	The language of the ARCAD Server {ENG |FRA}

Remote Application (template) Properties

Refer to Table 24: Edit application templates

Property	Description
${config.application.code}	The ARCAD Application Code
${config.application.text}	The Application Description
${config.application.prefix}	Prefix used for the associated version libraries
${config.application.manager}	The Manager of this application
${config.application.updateliblistscopeValue}	Library List Change scope
${config.application.updatejobdscopeValue}	JOBD Library List Change scope
${config.application.operationalLibraries}	The comma-separated list of the operational libraries
${config.application.libraries}	The comma-separated list of the library list

Remote Environment (template) Properties

Refer to Table 25: Edit environment templates

Property	Description
${config.environment.code}	The ARCAD Environment ID.
${config.environment.text}	The environment Description
${config.environment.prefix}	The Prefix used for the associated version libraries
${config.environment.owner}	The owner of the environment
${config.environment.environmenttypeValue}	The Environment Type

Property	Description
${config.environment.implementationmodeValue}	The Implementation Mode
${config.environment.updateliblistscopeValue}	Library List Change scope
${config.environment.updatejobdscopeValue}	JOBD Library List Change scope
${config.environment.operationalLibraries}	The comma-separated list of the operational libraries
${config.environment.libraries}	The comma-separated list of the library list

Save the changes (💾, Ctrl+S or **File** > **Save**).

49.4 Updating an ARCAD target's runtime

The Arcad Runtime Deployment script (provided by default) is in charge of launching the ANT project to update target ARCAD runtimes before deploying an ARCAD application. There are two ways to launch this process:

- automatically during the deployment process that includes an ARCAD RUNTIME-type infrastructure item and ARCAD runtime scripts (ref.:Add deployment actions to activities and stages), or
- manually at any time from the ARCAD RUNTIME-type infrastructure item.

To manually push the layout template to an ARCAD runtime without executing a deployment process, right-click on the infrastructure item in the **Infrastructure Items** search view and select **Apply target ARCAD runtime templates**.

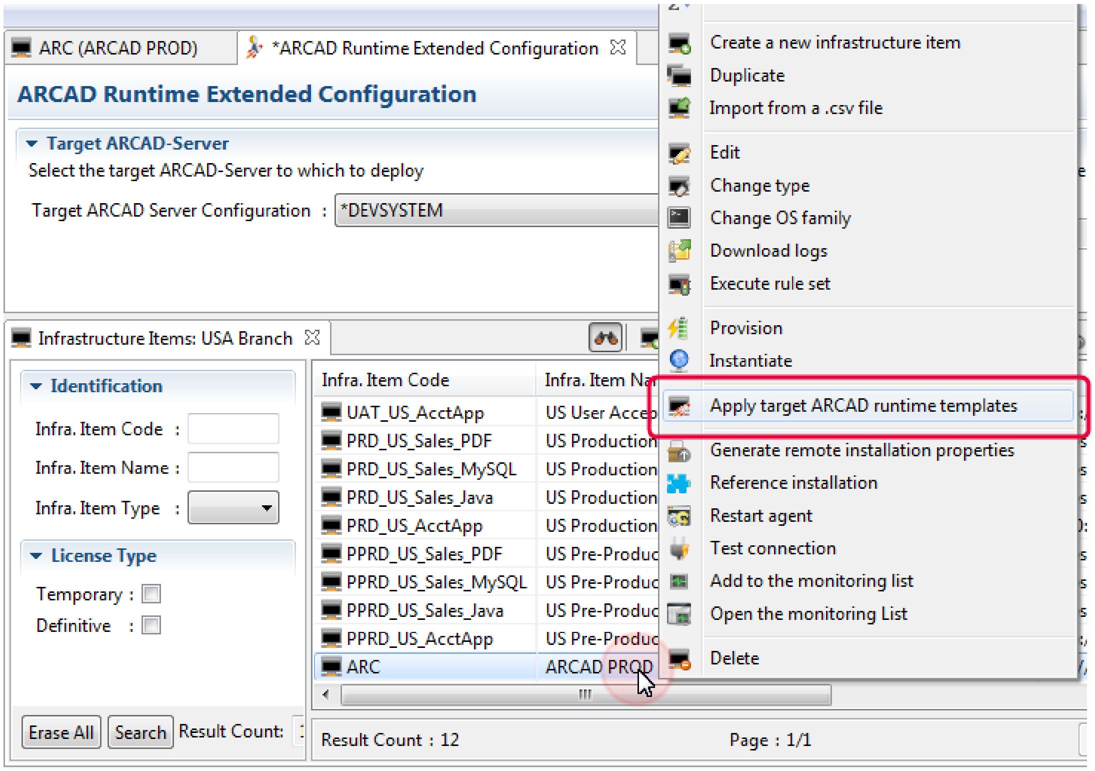


Figure 117: Manually updating the application layout on a target ARCAD runtime

Note

Only ARCAD RUNTIME-type items will display this option.

When the update is complete, the execution log will open automatically allowing you to confirm the changes have taken place on the target system.

Plug-ins & APIs

Introduction to the DROPS plug-ins & APIs

The import and deployment processes can be carried out from the DROPS Studio, however these processes can also be defined and launched by various other interfaces and tools, via plug-ins. The DROPS workflow can also be linked to a JIRA via a plug-in.

These plug-ins are quick to install and often offer additional features. All of the following plug-ins are accessed and managed inside their respective platforms.

- Ant tasks on page 335
- DROPS' Public REST API on page 343
- DROPS for Jenkins on page 362
- DROPS for JIRA on page 370
- DROPS for JIRA Cloud on page 391
- DROPS for Azure DevOps on page 396
- Workitems on page 401

50 Ant tasks

DROPS comes with a set of Ant tasks designed to allow the creation and execution of import and deployment processes.

The JAR file containing the ANT task client API can be found in the directory where DROPS is installed: *<DROPS Server installation directory>/clientAPI directory.*

An exemplary Ant build file can also be found in this directory.

Client APIs
drops.instance task below
drops.import.release task on the next page
drops.deploy task on page 339
Tips on page 341
Full script example on page 342

Table 26: ANT Task APIs: Configuration

ANT Task Elements

drops.import.filter

The drops.import.filter element is an embedded element that can be used with the drops.import.release task. If the *filterType attribute set to *MANUAL, then this element is used to specify the external version(s) to use for each import strategy run by that task.

Attribute	Description
*componentCode	The code of the import strategy's parent component.
*strategyCode	The code of the import strategy to run.
*externalVersion	The external version used by the import strategy.

Table 27: The drops.import.filter element's attributes

Example

```
<drops.import.filter componentCode="SQL" strategyCode="svn"
externalVersion="head" />
```

drops.instance task

This task creates a reference to a DROPS Server and saves it in an ANT property. This property can be used later on by the drops.import.release task and the drops.deploy task to refer to a connection automatically.

Attribute	Description
*propertyName	Create an ANT property to reference the connection to a DROPS Server that will include the URL, user and password defined below. Enter the name of the property to call later to refer to a connection automatically.

Table 28: The drops.instance task attributes

Attribute	Description
*url	The remote DROPS Server URL. http://*<server IP or network name>*:*<port>*
*user	A valid DROPS user.
*password	The user's password.

Table 28: The drops.instance task attributes

Example

```
<drops.instance propertyName="drops.server" url="http://###.###.#.###:5254"
user="James" password="Bond" />
```

drops.import.release task

This task creates and executes an import process instance for an application. The task supports the same range of options as the ones found in the **New import process instance** dialog, as described in the table below.

Reference

For more information about the parameters for an import process instance, refer to Create new import process instances on page 250.

Attribute	Description
*dropsServerProperty	The Ant tasks of the reference to the DROPS Server to use, defined by the Ant tasks.
*applicationCode	The application that includes the artifacts to import.
*releaseNumber	The number of the release that will hold the imported artifacts. This value can be any release number that complies with the application's Release Number Template or ***GEN**.
releaseText	The description of the release.
incrementPart	This attribute is required if the *releaseNumber is set to *GEN. The part of the release number that will be increased to create the release. Can be **MAJOR**, **MINOR**, **BUILD** or **REV**.

Table 29: The drops.import.release task's attributes

Attribute	Description
*filterType	The import strategy to use to import the artifacts. Like in the DROPS Studio, three methods are available: • ***NOFILTER**: all import strategies are run using the release number as the external version number. • ***MANUAL**: import strategies listed using embedded drops.import.filter elements or listed in a local file are executed. • ***FILE**: import strategies listed in an external .properties file located on the DROPS Server are executed.
filterFilename	The absolute path on the DROPS Server of the .properties file that contains a list of import strategy codes and the external version number used to run each of them. This attribute is only required if the *filterType is set to *FILE.
fromLocalFile	The absolute path of a local .properties file that contains a list of import strategy codes and the external version number used to run each of them. This attribute is only used if the *filterType is set to *MANUAL.
resetRelease	If set to **true** and if the Release already exists, its content is cleared before artifacts are imported. By default the value is **false**.
autoStart	If set to **true**, the import process instance created by this task is immediately executed. If set to **false**, it will have to be manually executed later. By default the value is **true**.
waitForCompletion	If set to **true** and if autoStart is true as well, this task will block the execution until the import process is complete. By default the value is **true**.
releaseNumberProperty	Create an ANT property to reference the release created when this task is run. Enter the name of the property to call. This property can be used later as the *releaseNumber in the drops.deploy task to create and launch a deployment process for this release.
importResultProperty	The name of an ANT property that will be set with the result of the import process instance. Import result: 0: succeeded 1: failed 2: unknown

Table 29: The drops.import.release task's attributes

Supported nested elements

- drops.import.filter: if the task's *filterType attribute is set to ***MANUAL**, multiple elements can be used to describe the import strategies to be run and the external version numbers to use.

> ***Example***
>
> The following example will create and execute a new import process for the DROPS_DEMO application. The release's number will be generated by incrementing its revision digit. Every component's import strategies will be run using the release Number as their external version number. The import process instance reference will be put in the drops.release property and the release number will be put in the drops.import.instance property.
>
> This import process instance will be created and executed on the DROPS Server referenced by the drops.server property.
>
> ```
> <drops.import.release
>     dropsServerProperty="drops.server"
>     applicationCode="DROPS_DEMO"
>     releaseNumber="*GEN"
>     releaseText="ANT task importation"
>     incrementPart="REV"
>     filterType="*NOFILTER"
>     releaseNumberProperty="drops.release"
>     instanceProperty="drops.import.instance" />
> ```

> ***Example***
>
> Same as above, but this time the filterType attribute is set to *MANUAL. Consequently, two nested drops.import.filter elements are used to list the import strategies to be run: the SQL component’s svn strategy, and the IBMI component’s ARCAD strategy.
>
> ```
> <drops.import.release
>     dropsServerProperty="drops.server"
>     applicationCode="DROPS_DEMO"
>     releaseNumber="*GEN"
>     releaseText="ANT task importation"
>     incrementPart="REV"
>     filterType="*MANUAL"
>     releaseNumberProperty="drops.release"
>     instanceProperty="drops.import.instance">
>     <drops.import.filter componentCode="SQL" strategyCode="svn"
> externalVersion="head" />
>     <drops.import.filter componentCode="IBMI" strategyCode="arcad"
> externalVersion="10.06.00" />
>     </drops.import.release>
> ```

> ***Example***
>
> Still the same as above, but the filterType attribute is set to *FILE. In this case, the filterFilename attribute is set to point to the location of a .properties file on the DROPS Server, that contains a list of import strategy codes and the external version number to use for each of them.
>
> ```
> <drops.import.release
>     dropsServerProperty="drops.server"
>     applicationCode="DROPS_DEMO"
> ```

```
releaseNumber="*GEN"
releaseText="ANT task importation"
incrementPart="REV"
filterType="*FILE"
filterFilename="/tools/release12547856.properties"
releaseNumberProperty="drops.release"
instanceProperty="drops.import.instance" />
```

drops.deploy task

This task creates and executes a deployment process instance for a given process and release number.

Attribute	Description
*dropsServerProperty	The Ant tasks of the reference to the DROPS Server to use, defined by the Ant tasks.
*environmentCode	The code of the environment holding the deployment process to run.
*processCode	The code of the deployment process to run.
*releaseNumber	The release number of the release to deploy. Enter either a release number or the value of the releaseNumberProperty from the drops.import.release task.
transactionId	The Transaction Id to use for this deployment process instance.
autoStart	If set to **true**, the deployment process instance created by this task is immediately executed. If set to **false**, it will have to be manually executed later on. By default the value is **true**.
waitForCompletion	If set to **true** and if autoStart is true as well, this task will block the execution until the deployment process instance is done. By default the value is **true**.
deploymentInstanceProperty	Create an ANT property to reference the deployment process instance created by this task.
deploymentResultProperty	Create an ANT property to reference the result of the execution of the deployment process instance created and executed by this task. Deployment result: 0: succeeded 1: failed 2: unknown 5: succeeded with warnings 6: canceled

Table 30: The release.deploy task's attributes

Example

Following the previous example, the previously imported release will be deployed using the process DEPLOY_DEMO from the UAT environment. The release number was retrieved from the releaseNumberProperty in the drops.import.release task and its value is now passed to the *releaseNumber attribute.

```
<drops.deploy
     dropsServerProperty="drops.server"
     environmentCode="UAT"
     processCode="DEPLOY_DEMO"
     releaseNumber="${drops.release}"
     deploymentResultProperty="drops.deployment.result" />
```

Tips

Classify your Server Credentials

As stated above, the drops.instance task element has a password attribute which is supposed to be confidential. ANT itself doesn't offer a way to hide this kind of information. However, because this information should classified and is used multiple times in an ANT script, it is advised to create an external .properties file with restricted permissions and allow the ANT script to load it using the <property> standard ANT tasks.

This method offers more flexibility and security over the credentials used to connect to the DROPS Server.

Example

A file named *drops.credentials.properties* is created under the */opt/tools/scripts* directory. This file contains the following set of properties:

```
drops.server.url=http://###.###.#.###:5254
drops.port=5254
drops.user=James
drops.password=Bond
```

It is then possible to load these properties in an ANT script using the loadProperties ANT task:

```
<property file="/opt/tools/scripts/drops.credentials.properties" />
```

Then it is possible to use the properties retrieved from the credentials' file with the drops.instance task:

```
<drops.instance propertyName="drops.server" url="${drops.server.url}"
login="${drops.user}" password="${drops.password}" />
```

Full script example

Bellow is a sample ANT script that imports artifacts in a new release and deploys it on the go.

```
<project default="main">
    <taskdef resource="com/dropssoftware/client/ant/antlib.xml"/>

    <!-- Required properties -->
    <property name="in.jenkins.build.number" value="" />
    <property name="in.drops.application" value="" />
    <property name="in.drops.environment" value="" />
    <property name="in.drops.process" value="" />
    <property name="in.drops.import.filter.file" value="" />
    <property file="${in.drops.server.properties}" />

    <!-- Optional properties -->
    <property name="in.drops.increment.part" value="REV" />
    <property name="in.drops.release.number" value="*GEN" />

    <!-- Global properties -->
    <property name="drops.server.url" value="http://${drops.server}:${drops.port}"/>
    <target name="main" depends="connect, importation, deployment" />
    <target name="connect">
        <drops.instance propertyName="drops.server" url="${drops.server.url}" login="${drops.user}"
password="${drops.password}" />
    </target>
    <target name="importation">
        <drops.import.release
        dropsServerProperty="drops.server"
        applicationCode="${in.drops.application}"
        releaseNumber="${in.drops.release.number}"
        releaseText="Automatic import from Jenkins Build #${in.jenkins.build.number}"
        incrementPart="${in.drops.increment.part}"
        filterType="*MANUAL"
        fromLocalFile="${in.drops.import.filter.file}"
        releaseNumberProperty="drops.release"/>
        <echo>Release ${drops.release} imported successfully.</echo>
    </target>
    <target name="deployment" if="drops.release">
        <drops.deploy
        dropsServerProperty="drops.server"
        environmentCode="${in.drops.environment}"
        processCode="${in.drops.process}"
        releaseNumber="${drops.release}"/>
    <echo>Deployment of release ${drops.release} of application ${in.drops.application} on environment
${in.drops.environment} is done.</echo>
    </target>
</project>
```

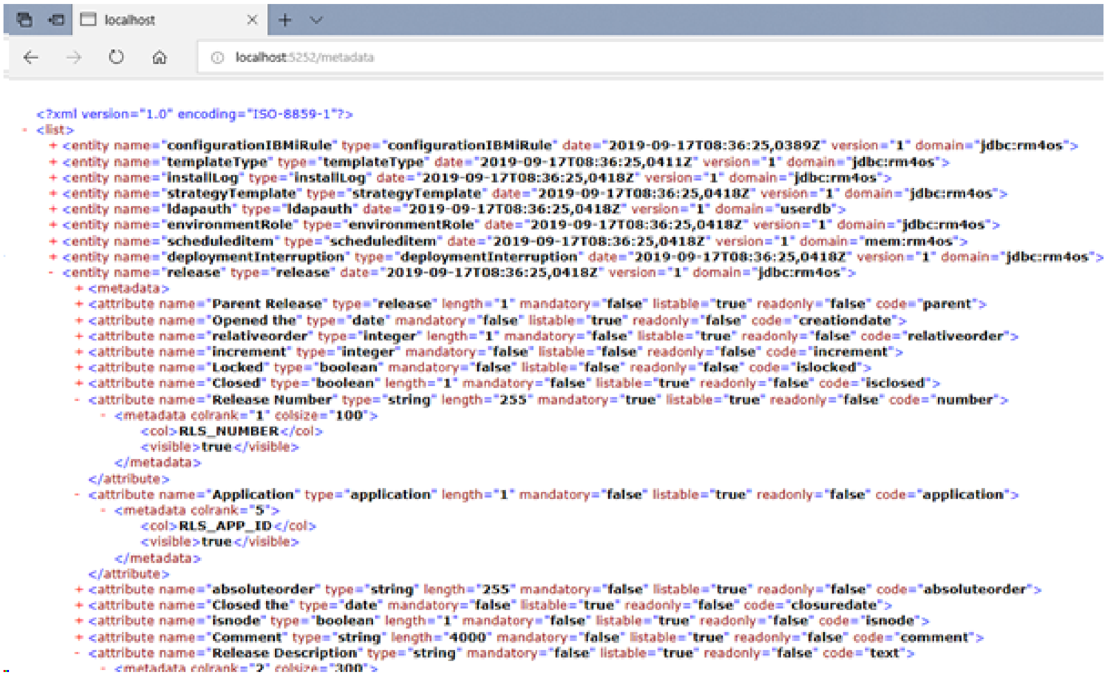
51 DROPS' Public REST API

51.1 Basic principles

DROPS' Public REST API relies on conventional CRUD methods for persistence of data. By default, authentication on the API is done using basic HTTP authentication. The API is available in XML ("Accept: application/xml ") or JSON ("Accept: application/json ") and can generate HTML when browsing with a browser.

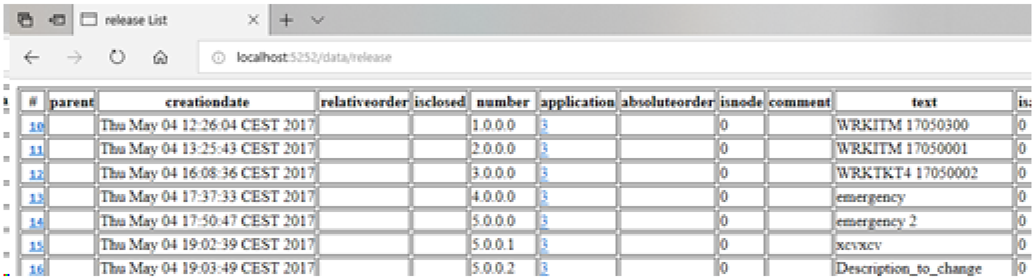
Third-party systems that want to automate DROPS will interact primarily during DROPS' operation phases, namely the creation of releases and the deployment of releases.

All the entities and their metadata managed by the API can be consulted via the URL *http://<server>:<port>/metadata* and the GET method.



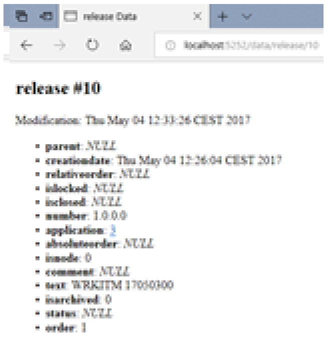
```
  <?xml version="1.0" encoding="ISO-8859-1"?>
- <list>
  + <entity name="configurationIBMiRule" type="configurationIBMiRule" date="2019-09-17T08:36:25,0389Z" version="1" domain="jdbc:rm4os">
  + <entity name="templateType" type="templateType" date="2019-09-17T08:36:25,0411Z" version="1" domain="jdbc:rm4os">
  + <entity name="installLog" type="installLog" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os">
  + <entity name="strategyTemplate" type="strategyTemplate" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os">
  + <entity name="ldapauth" type="ldapauth" date="2019-09-17T08:36:25,0418Z" version="1" domain="userdb">
  + <entity name="environmentRole" type="environmentRole" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os">
  + <entity name="scheduleditem" type="scheduleditem" date="2019-09-17T08:36:25,0418Z" version="1" domain="mem:rm4os">
  + <entity name="deploymentInterruption" type="deploymentInterruption" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os">
  - <entity name="release" type="release" date="2019-09-17T08:36:25,0418Z" version="1" domain="jdbc:rm4os">
    + <metadata>
    + <attribute name="Parent Release" type="release" length="1" mandatory="false" listable="true" readonly="false" code="parent">
    + <attribute name="Opened the" type="date" mandatory="false" listable="true" readonly="false" code="creationdate">
    + <attribute name="relativeorder" type="integer" length="1" mandatory="false" listable="true" readonly="false" code="relativeorder">
    + <attribute name="increment" type="integer" mandatory="false" listable="false" readonly="false" code="increment">
    + <attribute name="Locked" type="boolean" mandatory="false" listable="false" readonly="false" code="islocked">
    + <attribute name="Closed" type="boolean" length="1" mandatory="false" listable="true" readonly="false" code="isclosed">
    - <attribute name="Release Number" type="string" length="255" mandatory="true" listable="true" readonly="false" code="number">
      - <metadata colrank="1" colsize="100">
          <col>RLS_NUMBER</col>
          <visible>true</visible>
        </metadata>
      </attribute>
    - <attribute name="Application" type="application" length="1" mandatory="false" listable="true" readonly="false" code="application">
      - <metadata colrank="5">
          <col>RLS_APP_ID</col>
          <visible>true</visible>
        </metadata>
      </attribute>
    + <attribute name="absoluteorder" type="string" length="255" mandatory="false" listable="true" readonly="false" code="absoluteorder">
    + <attribute name="Closed the" type="date" mandatory="false" listable="true" readonly="false" code="closuredate">
    + <attribute name="isnode" type="boolean" length="1" mandatory="false" listable="true" readonly="false" code="isnode">
    + <attribute name="Comment" type="string" length="4000" mandatory="false" listable="true" readonly="false" code="comment">
    - <attribute name="Release Description" type="string" mandatory="false" listable="true" readonly="false" code="text">
      - <metadata colrank="2" colsize="300">
```

Each entity list can be consulted via the URL *http://<server>:<port>/data/<entity>*



#	parent	creationdate	relativeorder	isclosed	number	application	absoluteorder	isnode	comment	text	is:
10		Thu May 04 12:26:04 CEST 2017			1.0.0.0	3		0		WRKITM 17050300	0
11		Thu May 04 13:25:43 CEST 2017			2.0.0.0	3		0		WRKITM 17050001	0
12		Thu May 04 16:08:36 CEST 2017			3.0.0.0	3		0		WRKTKT4 17050002	0
13		Thu May 04 17:37:33 CEST 2017			4.0.0.0	3		0		emergency	0
14		Thu May 04 17:50:47 CEST 2017			5.0.0.0	3		0		emergency 2	0
15		Thu May 04 19:02:39 CEST 2017			5.0.0.1	3		0		xcvxcv	0
16		Thu May 04 19:03:49 CEST 2017			5.0.0.2	3		0		Description_to_change	0

or the URL *http://<server>:<port>/data/<entity>/<id>* for a particular entity ID.



release #10

Modification: Thu May 04 12:33:26 CEST 2017

- **parent**: *NULL*
- **creationdate**: Thu May 04 12:26:04 CEST 2017
- **relativeorder**: *NULL*
- **islocked**: *NULL*
- **isclosed**: *NULL*
- **number**: 1.0.0.0
- **application**: 3
- **absoluteorder**: *NULL*
- **isnode**: 0
- **comment**: *NULL*
- **text**: WRKITM 17050300
- **isarchived**: 0
- **status**: *NULL*
- **order**: 1

Access to a complete documentation that gives a description of all the CRUD operations available on DROPS' Public REST API. To do so, add */drops/api/doc* to your DROPS Server URL. The schema of each entity is also dynamically documented and referenced when they are used.

If you created customized web services, you can also add specifications for them. To do so, add in the **drops.restapi** bundle and in the *specs/custom* folder, a dedicated JSON file that contains only the **paths** node for each one of them.

```
{
  "paths":
  {
    "/mycustompath": {
      "get": {
        ...
        "security": [ { "basicAuth": [] } ] //necessary if the endpoint needs
authentication
      }
    }
  }
```

Example of JSON file content for a customized web service

A Swagger web interface is also served by the DROPS Server, that needs authentication. To access the interface, add */swaggerui* to your DROPS Server URL.

> **Note**
> It is necessary to install Maven to use this feature, as the Swagger UI is fetched by a Maven dependency.

The classic CRUD methods apply to entities:

- **GET**: fetch the entity list or a particular entity instance
  - *GET http://server:5252/data/release* - reading the list of entities releases
  - *GET http://server:5252/data/release/10* - read release entity with identifier 10

- **POST**: create a new entity. The creation parameters are in the body of the request
    - *POST http://server:5252/data/release*

      *body: increment=0&text=Automated+release&application=3&number=*GEN* – creation of a new empty release in the application with ID 3, the description of the release will be "Automated release", the release number will be generated automatically by incrementing the major number of the current counter

      > **Note**
      > The new generated ID is available in the answer.

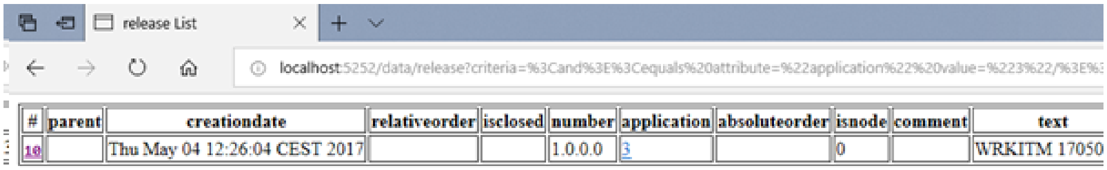
- **PUT**: update an entity
    - *PUT http://server:5252/data/release/10?text=New+description* - update description of the release with ID 10

- **DELETE**: delete an entity
    - **DELETE http://server:5252/data/release/10** - delete the release entity with ID 10

It is possible to filter the result lists with lists of criteria:

- *GET http://localhost:5252/data/release?**criteria=<and><equals attribute="application" value="3"/><equals attribute="number" value="1.0.0.0"/></and>***

  get the list of releases whose number is 1.0.0.0 and for the linked application with ID 3



localhost:5252/data/release?criteria=%3Cand%3E%3Cequals%20attribute=%22application%22%20value=%223%22/%3E%3

#	parent	creationdate	relativeorder	isclosed	number	application	absoluteorder	isnode	comment	text
10		Thu May 04 12:26:04 CEST 2017			1.0.0.0	3		0		WRKITM 17050

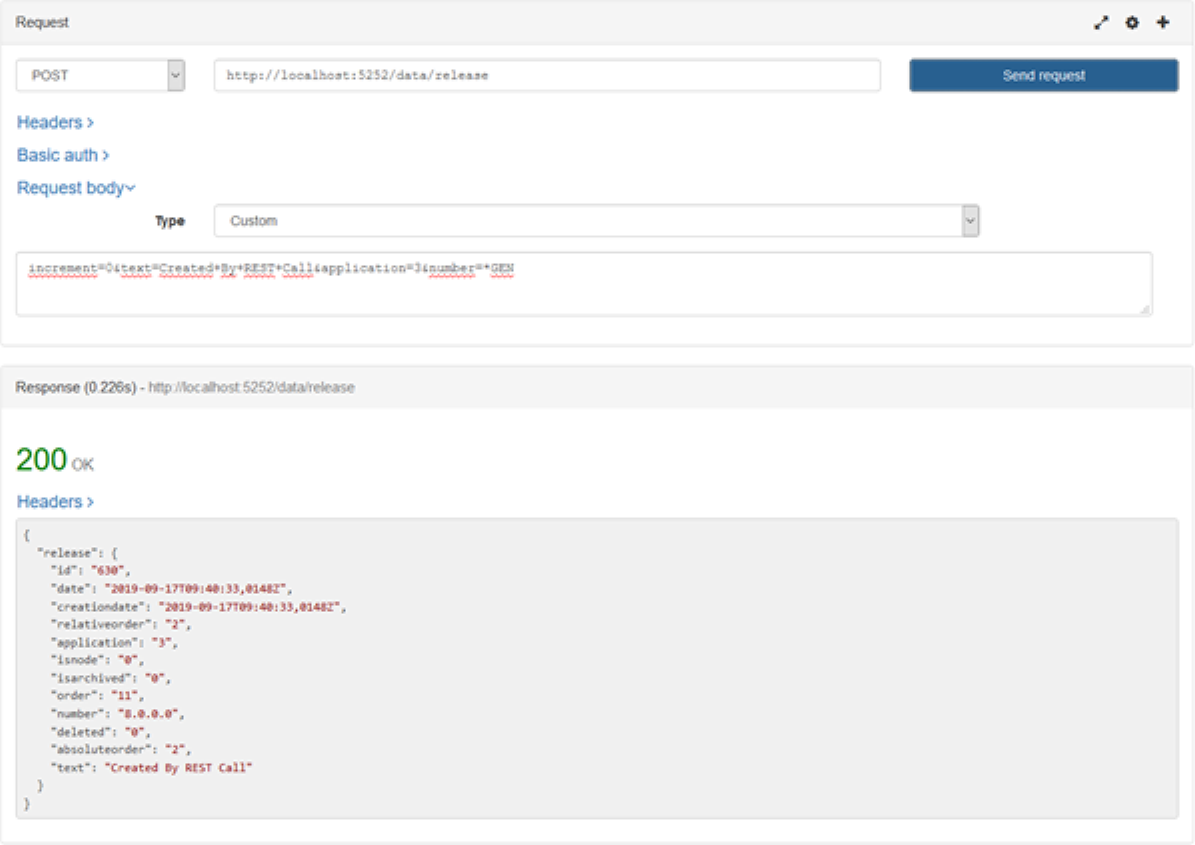
51.2 Using DROPS' Public REST API to create releases

The creation of a release is done in several phases:

- Opening/Creating the release
- Creating the import instance and add import strategies
- Running the import instance
- Retrieving the result of the import instance

51.2.1 Opening the release

If the release already exists then the ID can be retrieved using a GET request, but if the release doesn't exist yet, it must be created. This is a POST request that includes the necessary parameters in the body of the release entity.



POST http://localhost:5252/data/release

Type: Custom

```
increment=0&text=Created+By+REST+Call&application=3&number=*GEN
```

Response (0.226s) - http://localhost:5252/data/release

200 OK

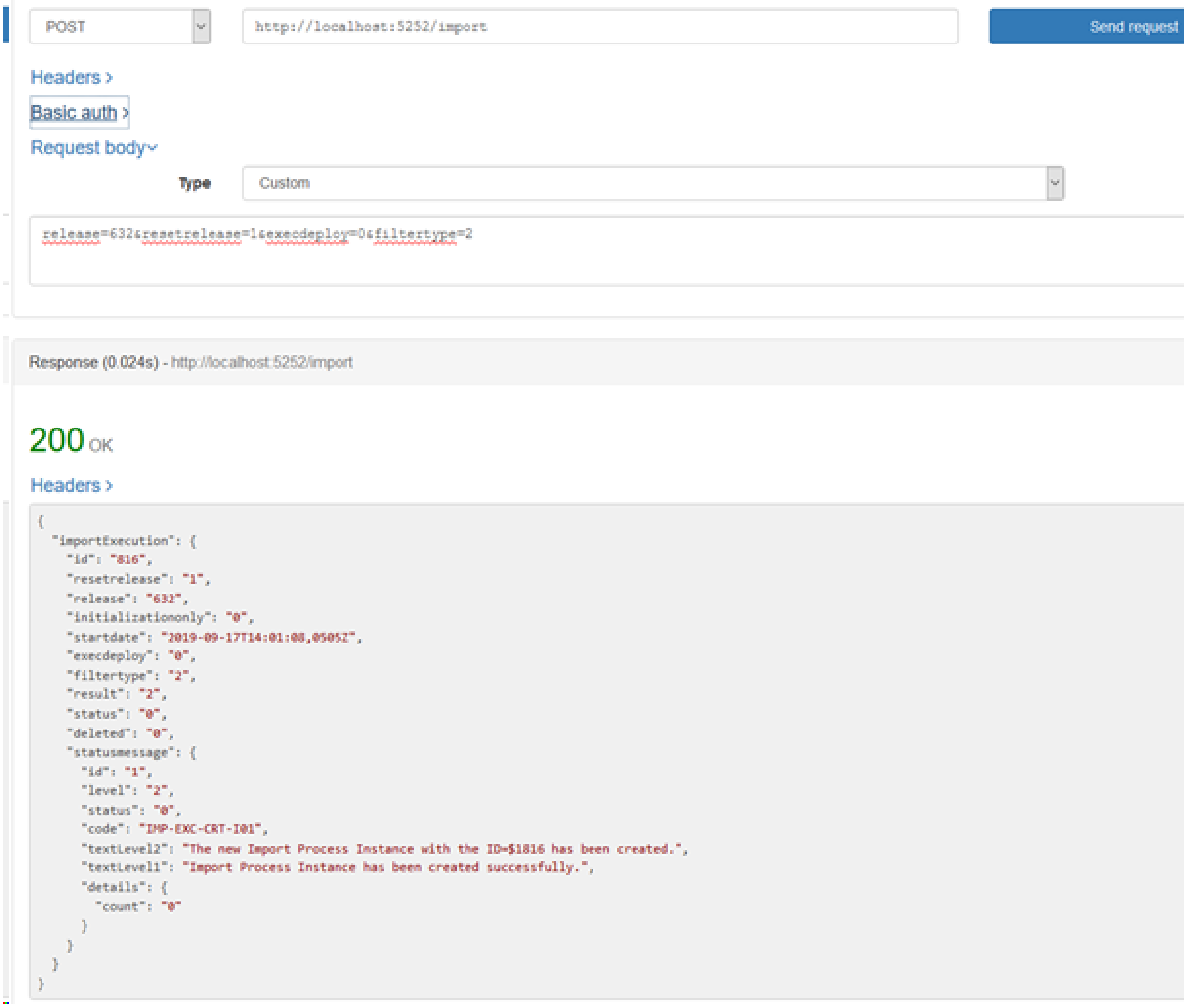
```
{
  "release": {
    "id": "630",
    "date": "2019-09-17T09:40:33,0148Z",
    "creationdate": "2019-09-17T09:40:33,0148Z",
    "relativeorder": "2",
    "application": "3",
    "isnode": "0",
    "isarchived": "0",
    "order": "11",
    "number": "8.0.0.0",
    "deleted": "0",
    "absoluteorder": "2",
    "text": "Created By REST Call"
  }
}
```

- **increment**: the part of the release number to increment when the release number is automatically generated
- **text**: description of the release
- **release**: the ID of the application this release relates to
- **number**: number of the release to be created or *GEN for automatic generation using the increment part

The HTTP response 200 confirms that the release was created successfully. The ID of the new release is available in the response body.

51.2.2 Creating the import instance

First of all, you have to create the import instance on the previously-created release using a POST request on the **importExecution** entity or by using the **/import** endpoint.



```
http://localhost:5252/import
```

```
release=632&resetrelease=1&execdeploy=0&filtertype=2
```

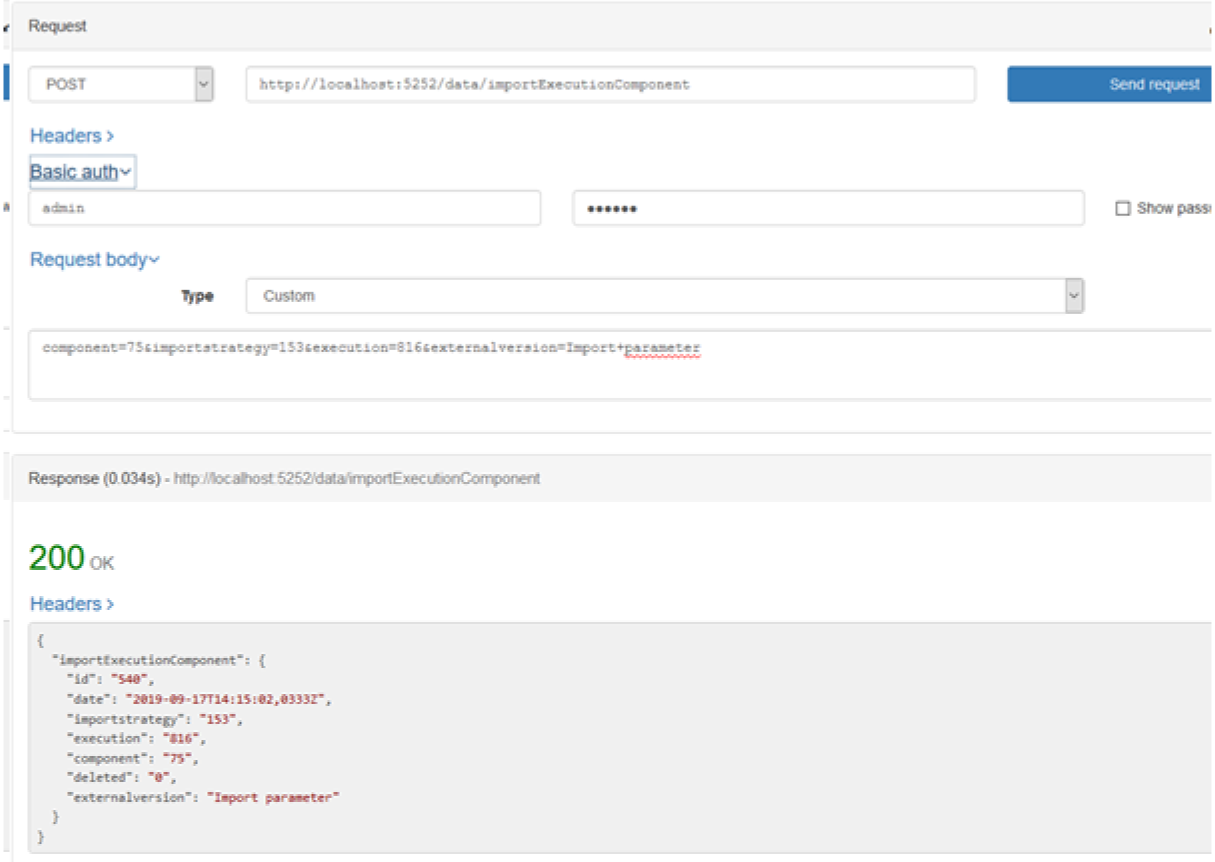
Response (0.024s) - http://localhost:5252/import

200 OK

```
{
  "importExecution": {
    "id": "816",
    "resetrelease": "1",
    "release": "632",
    "initializationonly": "0",
    "startdate": "2019-09-17T14:01:08,0505Z",
    "execdeploy": "0",
    "filtertype": "2",
    "result": "2",
    "status": "0",
    "deleted": "0",
    "statusmessage": {
      "id": "1",
      "level": "2",
      "status": "0",
      "code": "IMP-EXC-CRT-I01",
      "textLevel2": "The new Import Process Instance with the ID=$1816 has been created.",
      "textLevel1": "Import Process Instance has been created successfully.",
      "details": {
        "count": "0"
      }
    }
  }
}
```

- **release**: ID of the release this import instance relates to
- **resetrelease**: 0 or 1 to empty the release content before executing the import
- **execdeploy**: automatically execute a deployment after executing the import. (extra parameters may be needed)
- **filtertype**: filter to apply to this import (extra parameters may be needed)
  - 0: do not apply filter
  - 1: file filter
  - 2: custom filter

Most of the time filters are not needed or custom filters are used. To use custom filters, we must add the import strategies of the different components and the possible import parameters.



POST http://localhost:5252/data/importExecutionComponent

admin ••••••

□ Show pass

Type: Custom

```
component=75&importstrategy=153&execution=816&externalversion=Import+parameter
```

Response (0.034s) - http://localhost:5252/data/importExecutionComponent

200 OK

```
{
  "importExecutionComponent": {
    "id": "540",
    "date": "2019-09-17T14:15:02,0333Z",
    "importstrategy": "153",
    "execution": "816",
    "component": "75",
    "deleted": "0",
    "externalversion": "Import parameter"
  }
}
```

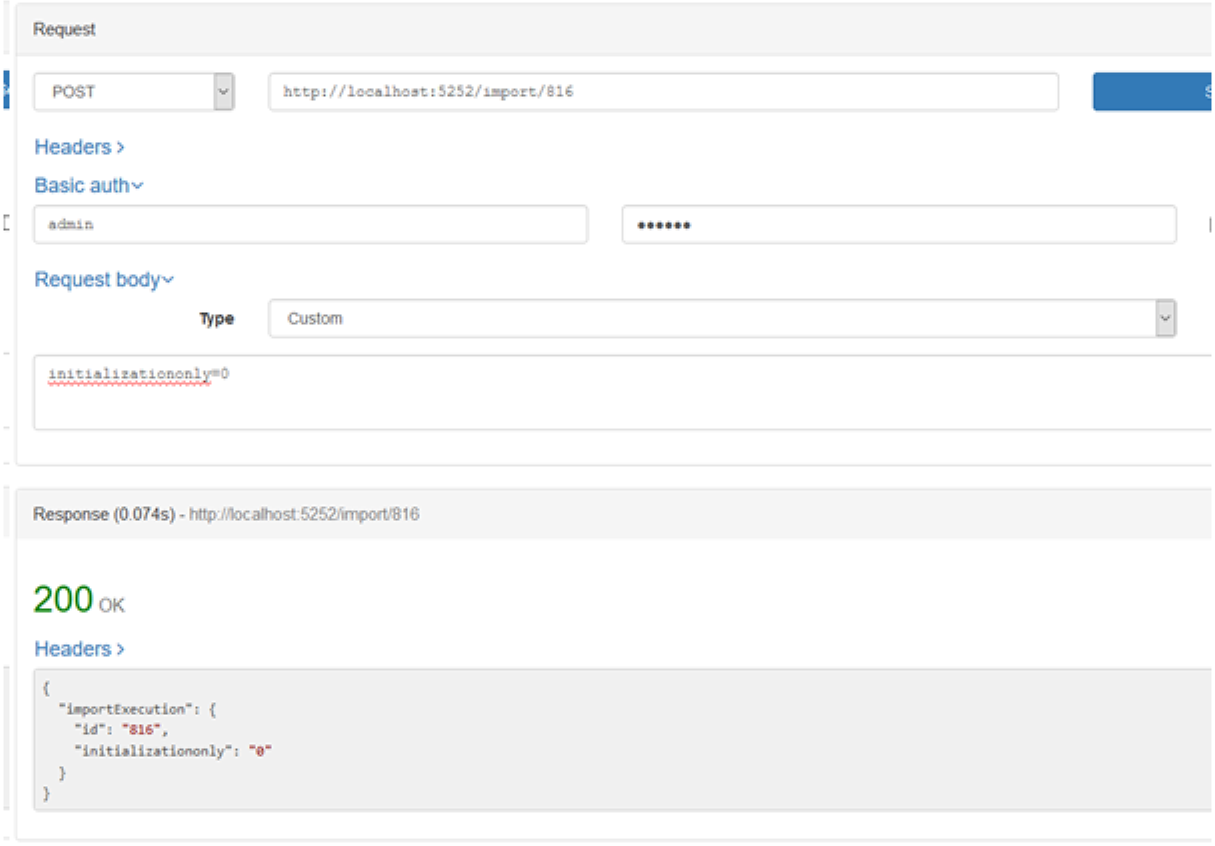
- **component**: ID of the component this filter relates to
- **importstrategy**: ID of the import strategy to use for this component
- **execution**: ID of the import instance
- **externalversion**: parameter for the import strategy

The HTTP response 200 confirms that the import instance was created successfully. The ID of the new import execution component is available in the response body.

At this stage the import instance is now configured and ready to run.

51.2.3 Running the import instance

The import instance execution is launched using a POST request on the entity or by using the **/import** endpoint.



Request

POST `http://localhost:5252/import/816`

Headers >

Basic auth

admin ••••••

Request body

Type: Custom

```
initializationonly=0
```

Response (0.074s) - http://localhost:5252/import/816

200 OK

Headers >

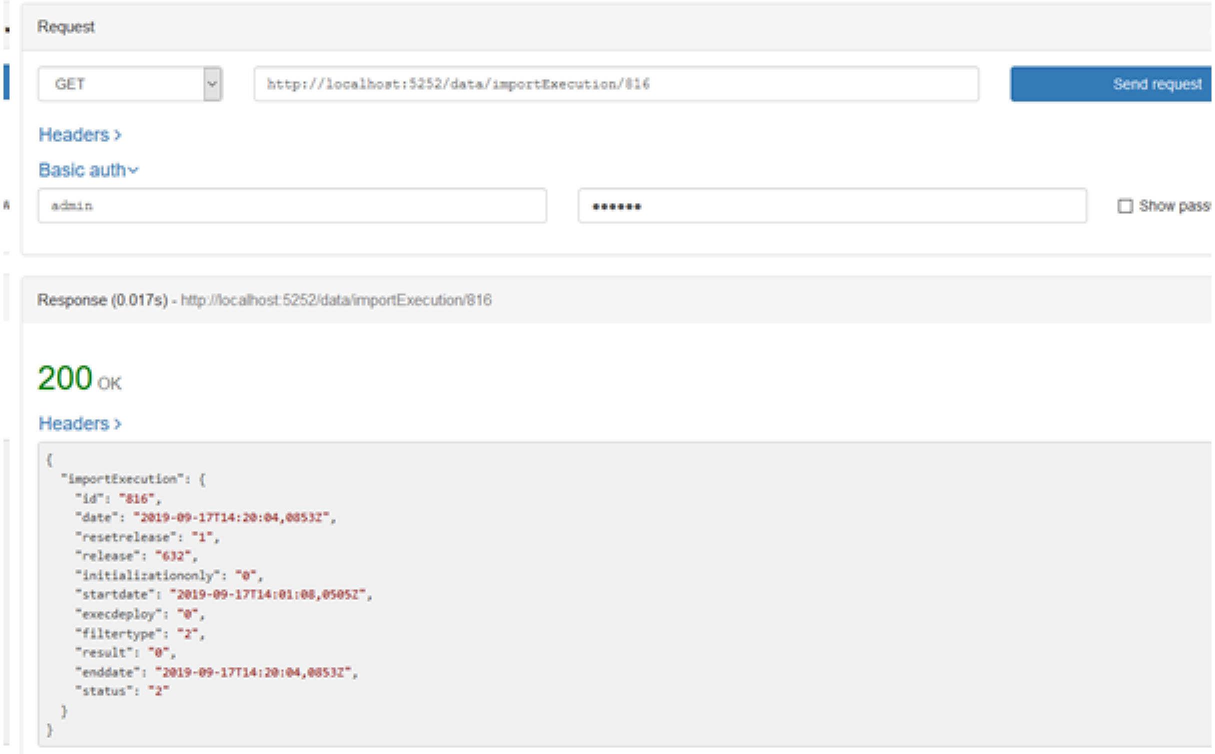
```
{
  "importExecution": {
    "id": "816",
    "initializationonly": "0"
  }
}
```

- **initializationonly**: optional parameter in the body to tell DROPS to launch only import strategies typed as initialization one.

The HTTP response 200 confirms that the import instance execution was launched successfully.

51.2.4 Reading the result of the import instance execution

The execution status of the import instance is available for consultation via a GET request.


Request
GET
http://localhost:5252/data/importExecution/816
Send request
Headers ›
Basic auth⌄
admin
••••••
□ Show pass
Response (0.017s) - http://localhost:5252/data/importExecution/816
200 OK
Headers ›

```
{
  "importExecution": {
    "id": "816",
    "date": "2019-09-17T14:20:04,0853Z",
    "resetrelease": "1",
    "release": "632",
    "initializationonly": "0",
    "startdate": "2019-09-17T14:01:08,0505Z",
    "execdeploy": "0",
    "filtertype": "2",
    "result": "0",
    "enddate": "2019-09-17T14:20:04,0853Z",
    "status": "2"
  }
}
```

The HTTP response 200 returns the status and result of the import instance execution.

The different statuses are:

- 0: Prepared
- 1: In progress
- 2: Completed

The different types of results are:

- 0: Succeeded
- 1: Failed
- 2: Not available

> **Note**
> The import execution status is prepared or running until the status is completed. When the status is completed an import execution result is set.

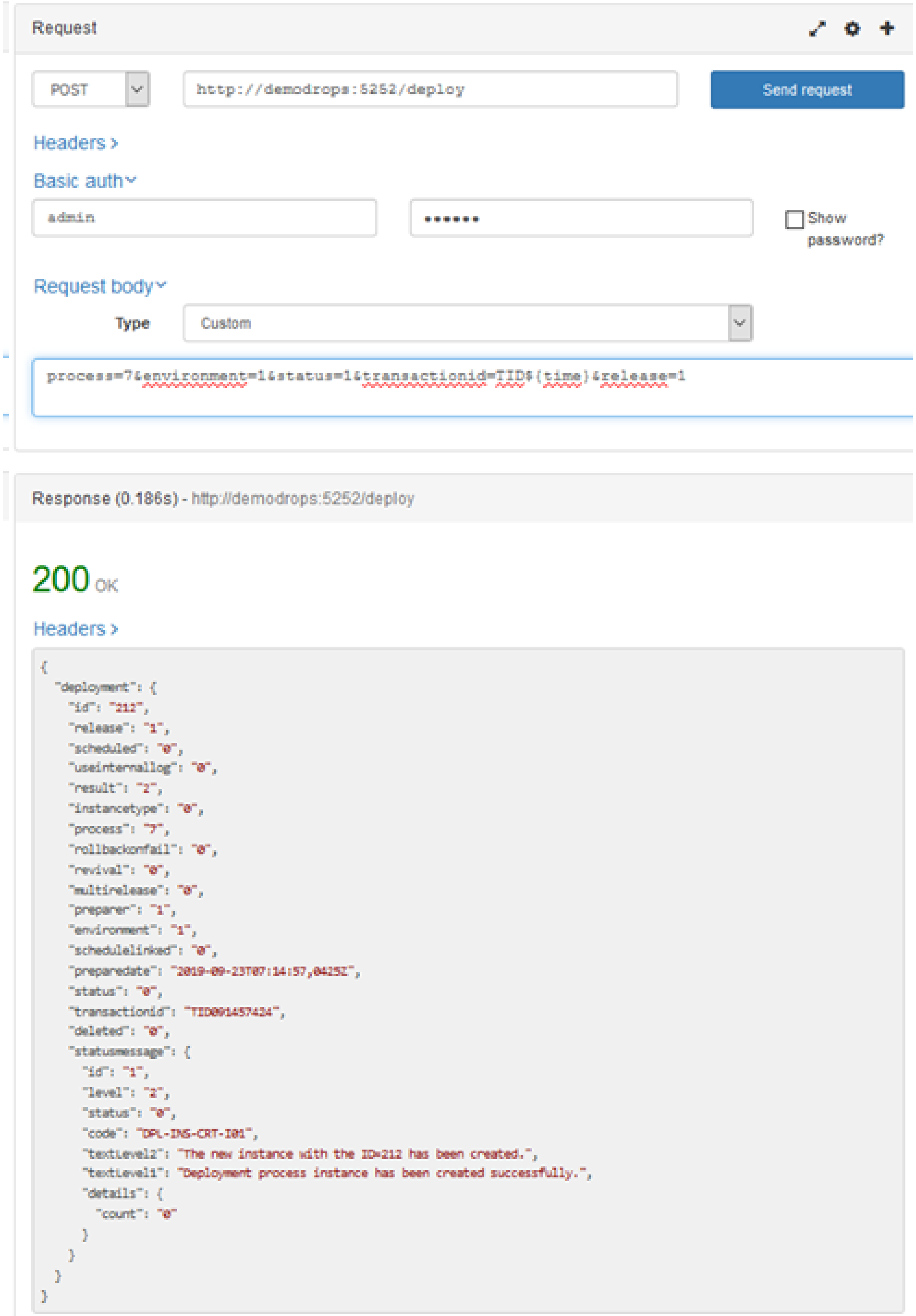
51.3 Using DROPS' Public REST API to deploy

Deploying a release can be done in two ways:

- Transfer of deliverables and deployment at once
- Provisioning deliverables and launching the deployment instance asynchronously

51.3.1 Creating the deployment instance

The deployment instance is created by a POST request to create the deployment instance.



http://demodrops:5252/deploy

admin

••••••

□ Show password?

Type: Custom

```
process=7&environment=1&status=1&transactionid=TID${time}&release=1
```

200 OK

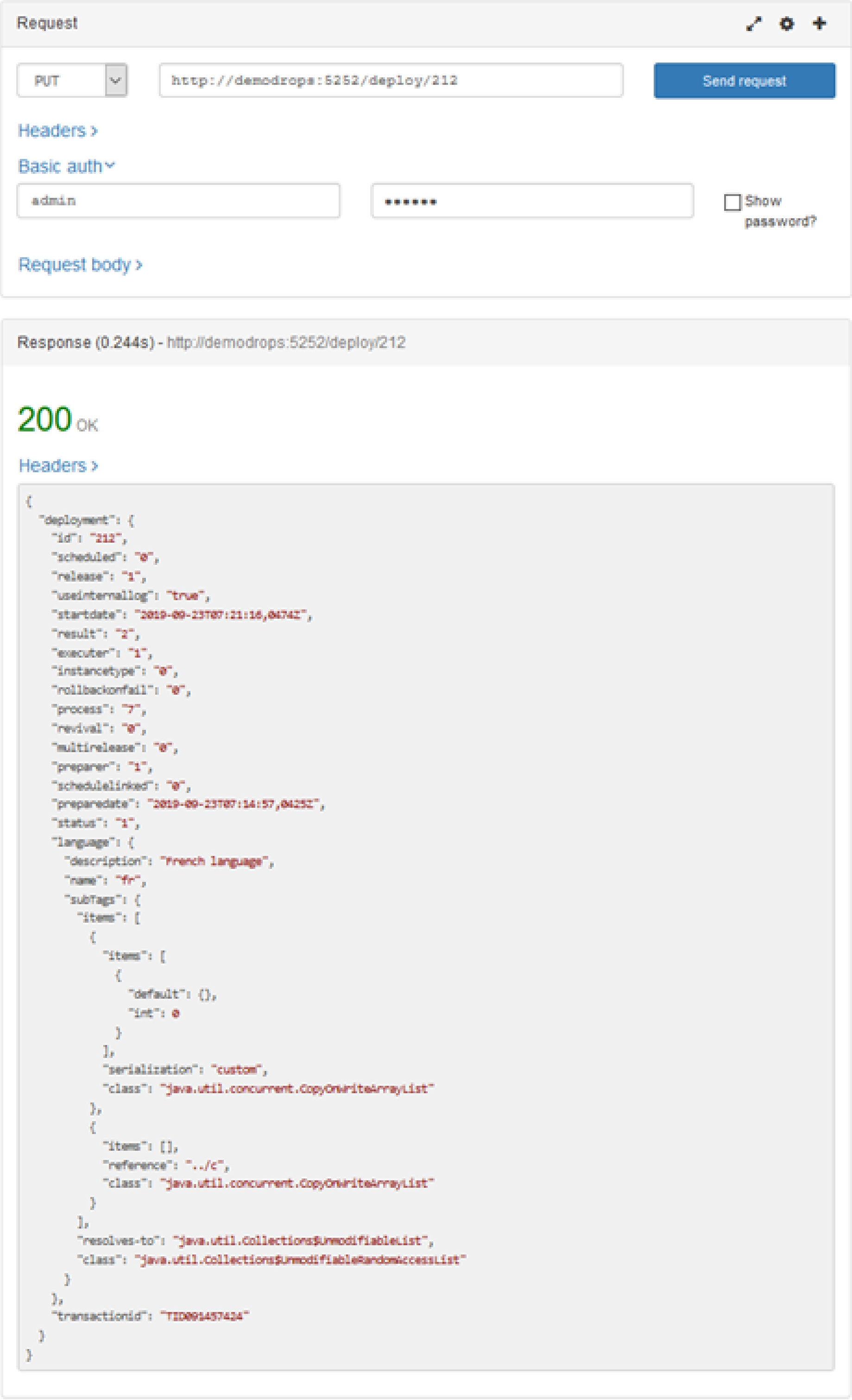
```
{
  "deployment": {
    "id": "212",
    "release": "1",
    "scheduled": "0",
    "useinternallog": "0",
    "result": "2",
    "instancetype": "0",
    "process": "7",
    "rollbackonfail": "0",
    "revival": "0",
    "multirelease": "0",
    "preparer": "1",
    "environment": "1",
    "schedulelinked": "0",
    "preparedate": "2019-09-23T07:14:57,0425Z",
    "status": "0",
    "transactionid": "TID091457424",
    "deleted": "0",
    "statusmessage": {
      "id": "1",
      "level": "2",
      "status": "0",
      "code": "DPL-INS-CRT-I01",
      "textLevel2": "The new instance with the ID=212 has been created.",
      "textLevel1": "Deployment process instance has been created successfully.",
      "details": {
        "count": "0"
      }
    }
  }
}
```

- **process**: ID of the process to apply
- **environment**: ID of the target environment

- **transactionid**: template to apply for generation
- **release**: ID of the release to deploy

The HTTP response 200 confirms that the deployment instance was created successfully. The ID of the new deployment instance is available in the response body.

Once the instance is created and the instance ID read, the instance is launched via a POST or PUT request on the entity or by using the **/deploy** endpoint.



`http://demodrops:5252/deploy/212`

admin

••••••

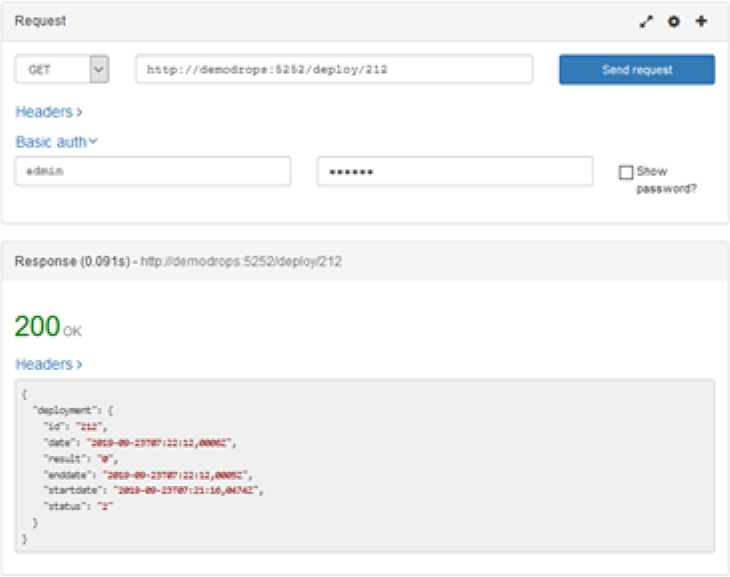
□ Show password?

Response (0.244s) - http://demodrops:5252/deploy/212

200 OK

```
{
  "deployment": {
    "id": "212",
    "scheduled": "0",
    "release": "1",
    "useinternallog": "true",
    "startdate": "2019-09-23T07:21:16,04742",
    "result": "2",
    "executer": "1",
    "instancetype": "0",
    "rollbackonfail": "0",
    "process": "7",
    "revival": "0",
    "multirelease": "0",
    "preparer": "1",
    "schedulelinked": "0",
    "preparedate": "2019-09-23T07:14:57,04252",
    "status": "1",
    "language": {
      "description": "French language",
      "name": "fr",
      "subTags": {
        "items": [
          {
            "items": [
              {
                "default": {},
                "int": 0
              }
            ],
            "serialization": "custom",
            "class": "java.util.concurrent.CopyOnWriteArrayList"
          },
          {
            "items": [],
            "reference": "../c",
            "class": "java.util.concurrent.CopyOnWriteArrayList"
          }
        ],
        "resolves-to": "java.util.Collections$UnmodifiableList",
        "class": "java.util.Collections$UnmodifiableRandomAccessList"
      }
    },
    "transactionid": "TID091457424"
  }
}
```

The result and monitoring of the deployment instance is done with a GET request on this same instance.



```
{
  "deployment": {
    "id": "212",
    "date": "2019-09-23T07:22:12,00062",
    "result": "0",
    "enddate": "2019-09-23T07:22:12,00052",
    "startdate": "2019-09-23T07:21:16,04742",
    "status": "2"
  }
}
```

The deployment execution status is prepared or running until it is completed or canceled. When the status is completed a deployment execution result is set.

The different statuses are:

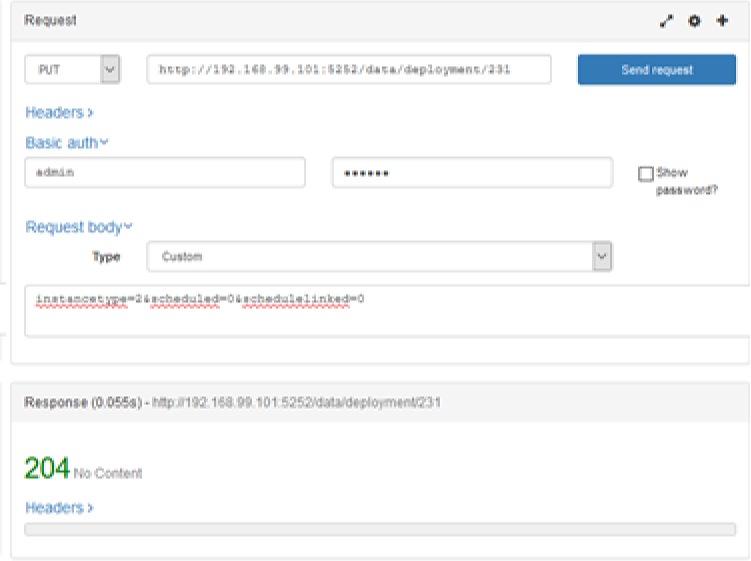
- 0: Prepared
- 1: In progress
- 2: Completed
- 3: Canceled
- 8: Suspended

The different types of results are:

- 0: Succeeded
- 1: Failed
- 5: Succeeded with warnings
- 6: Not available

51.3.2 Running the transfer instance, then the deployment instance

This mode starts with creating a deployment instance just like in the previous deployment mode. This deployment instance must be "transformed" into a Transfer instance (**transfer-only** in DROPS). This operation is done using a PUT request on the new instance.



Request

PUT http://192.168.99.101:5252/data/deployment/231 Send request

Headers ›

Basic auth ˅

admin •••••• □ Show password?

Request body ˅

Type: Custom

```
instancetype=2&scheduled=0&schedulelinked=0
```

Response (0.055s) - http://192.168.99.101:5252/data/deployment/231

204 No Content

Headers ›

The parameters are set in the request body:

- **instancetype**:
  - 2: transfer only
  - 3: execute only
- **scheduled**: 0 or 1 if schedule is used

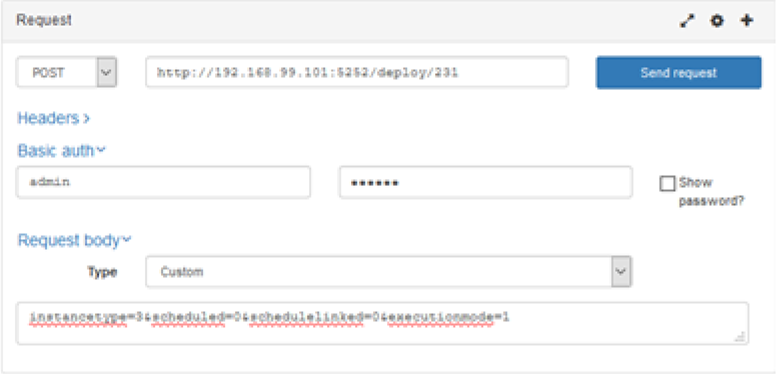
> **Note**
> This query does not return content.

The instance can now be executed using POST or PUT. Only the provisioning (the transfer of deliverables) will be executed.

The execution of the transfer instance can be followed via a GET method, as usual, on the deployment instance. Statuses and results are available in the response body.

Now, you have to create the deployment instance that will execute the installation phase of the deployment. As a reminder, the previous deployment instance only corresponds to the provisioning phase, that is to say, the transfer of the artifacts.

This creation is done using a new POST request on the deployment instance where the value of the instance type field is 3 and the execution mode field is set to 1.

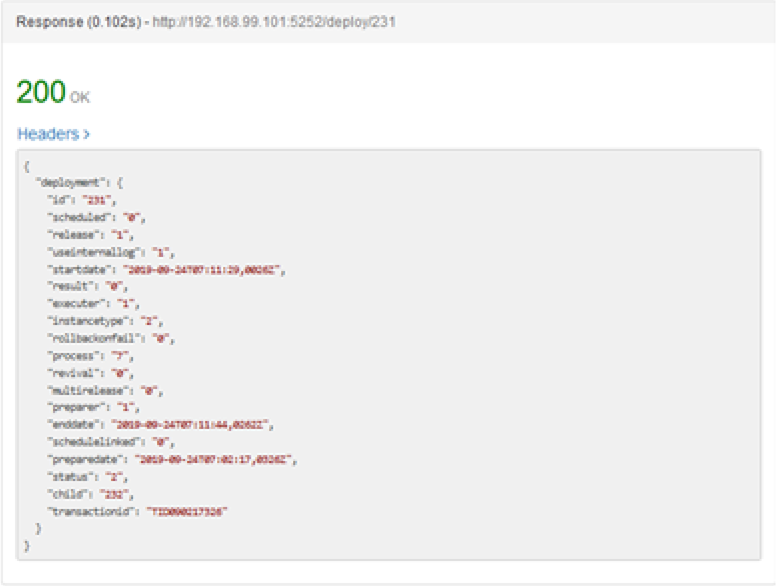


```
http://192.168.99.101:5252/deploy/231
```

admin / ••••••

Type: Custom

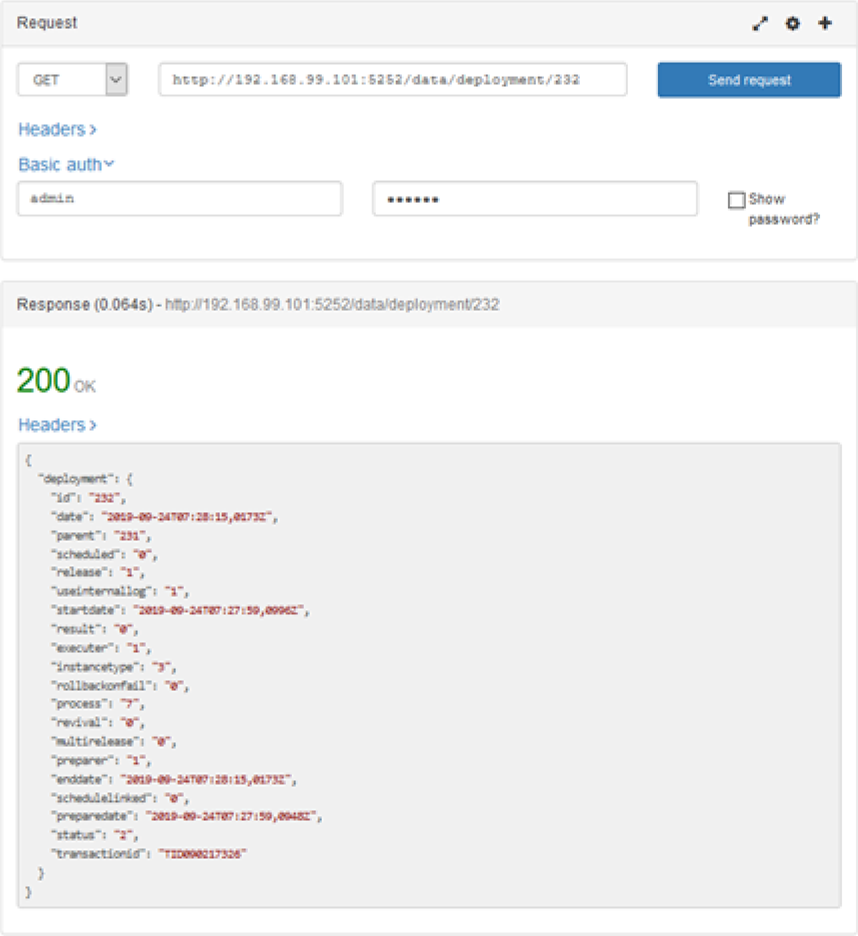
```
instancetype=3&scheduled=0&schedulelinked=0&executionmode=1
```



```
{
  "deployment": {
    "id": "231",
    "scheduled": "0",
    "release": "1",
    "useinternallog": "1",
    "startdate": "2019-09-24T07:11:29,00262",
    "result": "0",
    "executer": "1",
    "instancetype": "2",
    "rollbackonfail": "0",
    "process": "7",
    "revival": "0",
    "multirelease": "0",
    "preparer": "1",
    "enddate": "2019-09-24T07:11:44,02622",
    "schedulelinked": "0",
    "preparedate": "2019-09-24T07:02:17,03262",
    "status": "2",
    "child": "232",
    "transactionid": "TID000217326"
  }
}
```

The response contains a **child** field, which is the ID of the new linked deployment instance that corresponds to the installation phase of the deployment.

The execution of this new instance can be followed, as usual, with a GET request.



Request

GET http://192.168.99.101:5252/data/deployment/232 Send request

Headers >

Basic auth

admin ******

□ Show password?

Response (0.064s) - http://192.168.99.101:5252/data/deployment/232

200 OK

Headers >

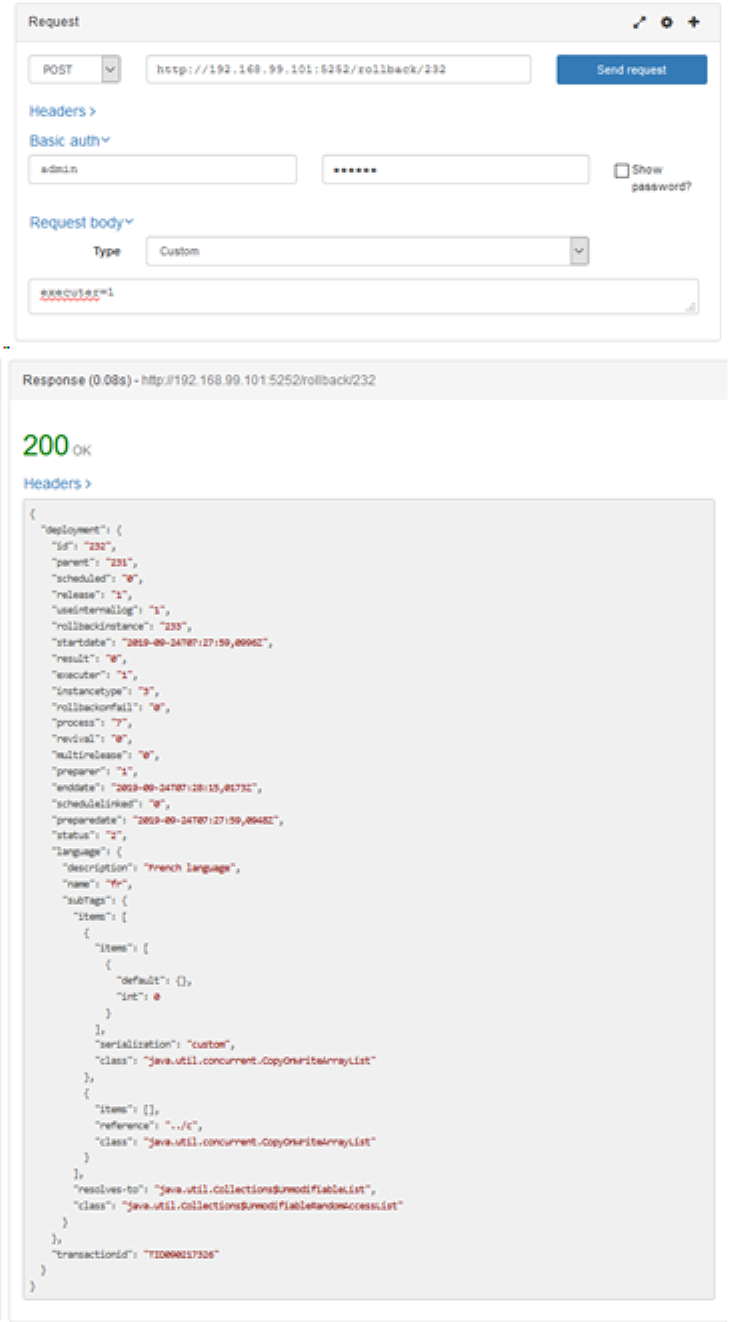
```
{
  "deployment": {
    "id": "232",
    "date": "2019-09-24T07:28:15,0173Z",
    "parent": "231",
    "scheduled": "0",
    "release": "1",
    "useinternallog": "1",
    "startdate": "2019-09-24T07:27:59,0996Z",
    "result": "0",
    "executer": "1",
    "instancetype": "3",
    "rollbackonfail": "0",
    "process": "7",
    "revival": "0",
    "multirelease": "0",
    "preparer": "1",
    "enddate": "2019-09-24T07:28:15,0173Z",
    "schedulelinked": "0",
    "preparedate": "2019-09-24T07:27:59,0948Z",
    "status": "2",
    "transactionid": "TID090217326"
  }
}
```

The status and result fields have the same meaning as before.

51.3.3 Rolling back

Rollback is only possible on a deployment instance that has already been executed. To implement the rollback, you need the ID of the deployment instance to revert. In the case of a two-part deployment, transfer + installation, you also need the deployment instance ID corresponding to the ID of the child install instance.

To implement the rollback, you must create a deployment instance using the **/rollback** endpoint and the POST method.



http://192.168.99.101:5252/rollback/232

admin

••••••

Type: Custom

executer=1

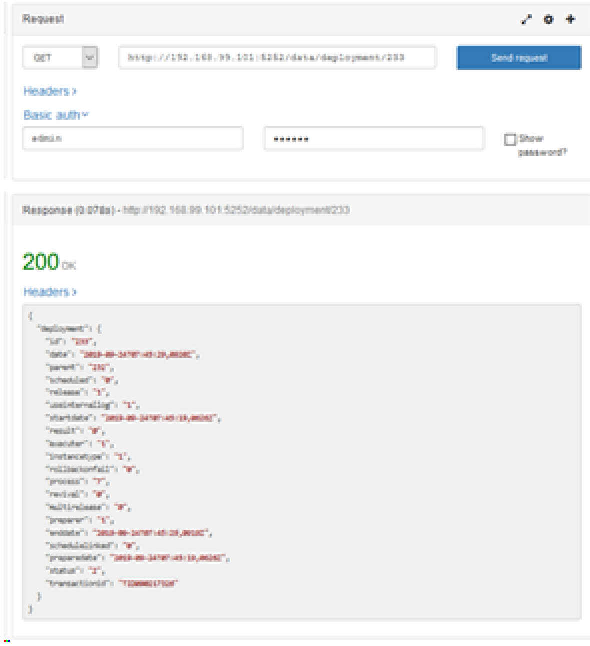
Response (0.08s) - http://192.168.99.101:5252/rollback/232

200 OK

```
{
  "deployment": {
    "id": "232",
    "parent": "231",
    "scheduled": "0",
    "release": "1",
    "useinternallog": "1",
    "rollbackinstance": "233",
    "startdate": "2019-09-24T07:27:59,09962",
    "result": "0",
    "executer": "1",
    "instancetype": "3",
    "rollbackonfail": "0",
    "process": "7",
    "revival": "0",
    "multirelease": "0",
    "preparer": "1",
    "enddate": "2019-09-24T07:28:15,01731",
    "schedulelinked": "0",
    "preparedate": "2019-09-24T07:27:59,09481",
    "status": "2",
    "language": {
      "description": "French language",
      "name": "fr",
      "subTags": {
        "items": [
          {
            "items": [
              {
                "default": {},
                "int": 0
              }
            ],
            "serialization": "custom",
            "class": "java.util.concurrent.CopyOnWriteArrayList"
          },
          {
            "items": [],
            "reference": "../c",
            "class": "java.util.concurrent.CopyOnWriteArrayList"
          }
        ],
        "resolves-to": "java.util.Collections$UnmodifiableList",
        "class": "java.util.Collections$UnmodifiableRandomAccessList"
      }
    },
    "transactionid": "TID000217326"
  }
}
```

The response contains a **rollbackinstance** field whose value is the ID of the new deployment instance created for the rollback part of the deployment.

This ID is used to follow the progress of the rollback instance using a GET request on the deployment entity.



http://192.168.99.101:5252/data/deployment/233

admin ●●●●●● □ Show password?

Response (0.078s) - http://192.168.99.101:5252/data/deployment/233

200 OK

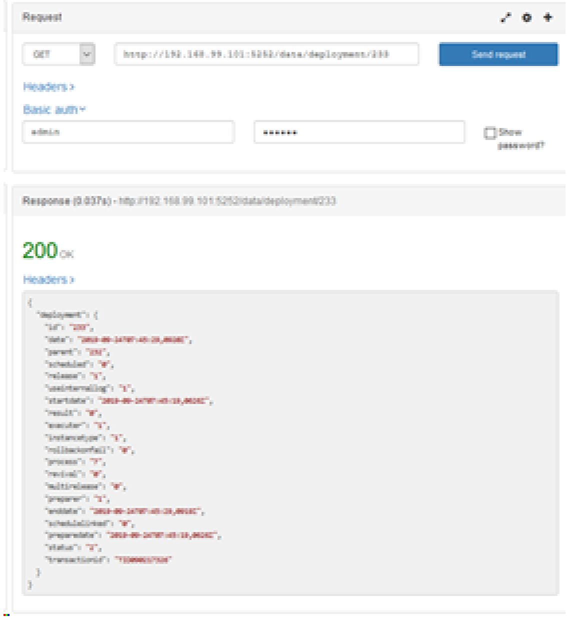
```
{
  "deployment": {
    "id": "233",
    "date": "[illegible]",
    "parent": "232",
    "scheduled": "0",
    "release": "1",
    "useinternallog": "1",
    "startdate": "[illegible]",
    "result": "0",
    "executor": "1",
    "instancetype": "1",
    "rollbackonfail": "0",
    "process": "7",
    "revival": "0",
    "multirelease": "0",
    "preparer": "1",
    "enddate": "[illegible]",
    "schedulelinked": "0",
    "preparedate": "[illegible]",
    "status": "2",
    "transactionid": "[illegible]"
  }
}
```

The status and result fields have the same meaning as before.

51.3.4 Tracking execution instances

Monitoring the execution of a deployment instance is done using a GET request:

- either on the deployment instance entity in order to have the complete information about the instance,



http://192.168.99.101:5252/data/deployment/233

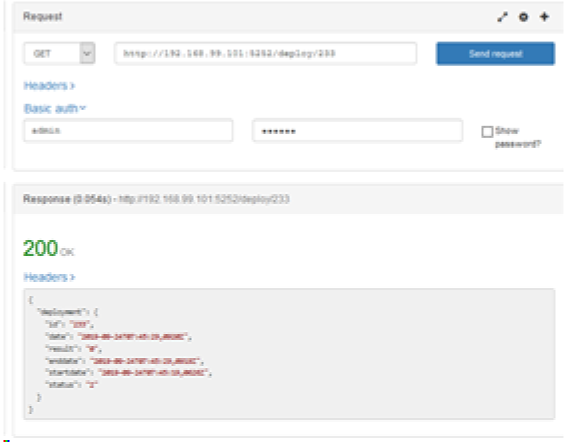
admin ●●●●●● □ Show password?

Response (0.037s) - http://192.168.99.101:5252/data/deployment/233

200 OK

```
{
  "deployment": {
    "id": "233",
    "date": "[illegible]",
    "parent": "232",
    "scheduled": "0",
    "release": "1",
    "useinternallog": "1",
    "startdate": "[illegible]",
    "result": "0",
    "executor": "1",
    "instancetype": "1",
    "rollbackonfail": "0",
    "process": "7",
    "revival": "0",
    "multirelease": "0",
    "preparer": "1",
    "enddate": "[illegible]",
    "schedulelinked": "0",
    "preparedate": "[illegible]",
    "status": "2",
    "transactionid": "[illegible]"
  }
}
```

- or on the endpoint /deploy to have only the follow-up of execution.



GET http://192.168.99.101:5252/deploy/233

admin ******

Response (0.054s) - http://192.168.99.101:5252/deploy/233

200 OK

```
{
  "deployment": {
    "id": "233",
    "date": "[illegible]",
    "result": "0",
    "enddate": "[illegible]",
    "startdate": "[illegible]",
    "status": "1"
  }
}
```

The same logic applies to the import process instances using the /import endpoint.

52 DROPS for Jenkins

Chapter Summary

52.1 Prerequisites 362
52.2 Configuring the DROPS for Jenkins plug-in 362
52.3 Adding DROPS steps to a project 364
52.4 Understanding results in the DROPS for Jenkins plug-in 368

The DROPS plug-in for Jenkins is an open-source tool for continuous integration.

The DROPS for Jenkins plug-in is intended for release managers and is compatible with both ways to use Jenkins (normal and pipeline, which is scriptable). It enables you to trigger the execution of import and deployment processes via Jenkins. You can add as many steps in a Jenkins job as required, and each can call DROPS to execute a different task.

> **Note**
> Only one DROPS Server can be piloted by Jenkins. The DROPS Server linked to Jenkins will be the only one for which you can select pre-defined import and deployment processes to launch.

This chapter describes how to install, configure and use the DROPS for Jenkins plug-in. There is no configuration required in DROPS.

52.1 Prerequisites

1. Jenkins must be installed. The plug-in is compatible with Jenkins ≥v**2.176.3**.

   > ***Reference***
   > For more information about installing Jenkins, refer to the Jenkins Installation Guide.

2. The ARCAD-Commons plug-in must be installed on Jenkins. This plug-in, common to many ARCAD products on Jenkins, only needs to be installed once, no matter how many ARCAD product plug-ins are installed on Jenkins. To install the ARCAD-Commons plug-in, click **Manage Jenkins** in the left menu of the dashboard, then click **Manage Plugins** and open the **Advanced** tab. In the **Upload Plugin** section, click **Choose File...** and navigate to the location where the file is stored. Select the *arcad-commons.hpi* file and click **Open**. Click **Upload**.

The DROPS for Jenkins plug-in is a simplified version of the full DROPS tool. This guide assumes that you are already familiar with the basics of DROPS and that the processes to execute have already been configured. This plug-in does not offer any DROPS configuration features.

52.2 Configuring the DROPS for Jenkins plug-in

Add the DROPS for Jenkins plug-in to Jenkins and set the server connection details.

52.2.1 Adding the plug-in to Jenkins

Follow the subsequent steps to install an ARCAD plug-in on Jenkins.

Step 1 Open Jenkins in a navigator.

Step 2 To access the plug-in install page, click **Manage Jenkins** in the left menu of the dashboard, then click **Manage Plugins** and open the **Advanced** tab.

Step 3 In the **Upload Plugin** section, click **Choose File...** and navigate to the location where the installation file is stored. Select the *.hpi* file that corresponds to the product and click **Open**.

Step 4 Click **Upload**.

Step 5 To complete the installation, check **Restart Jenkins when installation is complete and no jobs are running**, and wait for Jenkins to restart.

Result The plug-in is installed and available for use.

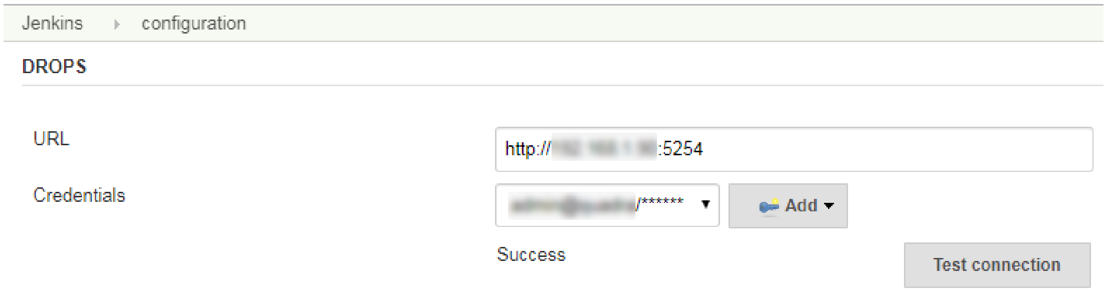
52.2.2 Setting the connection details

> ***Warning!***
> The following configuration is required.

Follow the subsequent steps to configure the DROPS for Jenkins plug-in.

Step 1 To access the plug-in configuration page, click **Manage Jenkins** in the left menu of the dashboard, then click **Configure System** and find the **DROPS** section.

Step 2 Define in the following fields:



URL: http://[redacted]:5254

Credentials: [redacted]/******

Success

Figure 118: Configuring the DROPS for Jenkins plug-in

Field	Description
URL	Define the URL address where the DROPS Server is located. This URL is usually made up of a prefix, *http://* or *https://*, then followed by the server's host name (DNS name or IP address) and the server's port. By default, the DROPS Server uses the 5254 port number or the 52540 port number for the secured connection, but these numbers may have been changed during configuration.
Credentials	Enter a valid DROPS Server user and the corresponding password. Click the **Add** button to add any user defined on the DROPS Server.

Table 31: Configure the DROPS for Jenkins plug-in

Step 3 Use the **Test connection** button to ensure that the connection is successful.

Step 4 To save any changes made, click **Save**.

Result The DROPS for Jenkins plug-in is connected to the DROPS Server.

52.3 Adding DROPS steps to a project

Create DROPS steps for Jenkins projects from the **Build Environment** configuring settings.

Step 1 In the home page, select the desired project.

Step 2 On the left menu, click **Configuration** then select the **Build** tab or scroll down to the **Build** section.

Step 3 In the **Build** section, select one of the DROPS steps from the **Add build step** drop-down list.

Result The step is added and ready to be defined.

52.3.1 Defining DROPS: Import steps

Define the following parameters for each DROPS: Import step added to a Jenkins project. These parameters define the release to import. They are the same parameters required to define an import process instance in the DROPS Studio.

The import is launched via Jenkins in the DROPS Studio.

Reference
For more information about defining import process instances, refer to Introduction to importing artifacts on page 242.

Field	Description
Applicatio n	Select the application for which to import the artifacts. The list is pre-populated with all of the applications defined in the DROPS Server.

Table 32: Configure DROPS: Import steps in Jenkins

Field	Description

Release	Select the release in which to save the artifacts that will be imported.
	• If **Current Release** is selected, the release defined in the Current Release field in the application is used to import (displayed in parentheses).
	• If **Create a New Release** is selected, a new release will be created using the Release Number Template defined in the application.
	If this option is selected, define the part of a release's version number to increment from the **Increment Part** drop-down menu. Keep in mind that the algorithm will update the other parts automatically. By default, the part is set to the value defined in the the Repository Storage Area settings in the DROPS Server configuration.
	• Selecting an **existing**, *active* release from the prepopulated list will import the artifacts to that release.
	You can create a new release by adding a script via the pipeline. To create a release by providing a release number (or updating an existing release if the number already exists) you must use the Jenkins pipeline. The following is the syntax to follow:
	`*pipeline {`
	` agent any`
	
	` stages {`
	`  stage('DEMO') {`
	`   steps {`
	`    dropsImport application: [code: 'DEMO'],`
	`      executions: [[component: [code: 'WEBAPP'], externalVersion: '1.0.0.0', strategy: [code: 'WEBAPP_GIT_IMPORT']]],`
	`      {color:#ff0000}release: [$class: 'NamedRelease', number: 'BUILD_${BUILD_NUMBER}']{color},`
	`      releaseDescription: 'my test from pipeline',`
	`      resetRelease: true,`
	`      startAutomatically: true`
	`   }`
	`  }`
	` }`
	`}*`
	The *NamedRelease* parameter must be paired with one of the following parameters in order to find the correct release. If the corresponding release exists, the task will update it, if it doesn't exist, Jenkins will create a new release.
	1. number: 'the release number' (The number parameter is a String type and can use Jenkins environment variables).
	2. releaseID: 'the ID of the existing release'.

Table 32: Configure DROPS: Import steps in Jenkins

Field	Description
Release Description	If a description is added to the step in Jenkins, any existing description of the release in DROPS will be replaced when the build is launched.
Reset release	Tick the **Reset release before importing** checkbox to purge the selected release of all of its contents before importing the artifacts. The current import replaces the old content completely.
Automatically start	Tick this checkbox to automatically start the import process automatically. If unselected, the import execution instance is created but not launched.
Executions	Define the component/import strategy pair(s) to use to import the artifacts. You can add as many pre-defined pairs as needed.
	Select the **Component**. The list is pre-populated with all of the components defined for the selected application.
	Select the **Import strategy**. The list is pre-populated with all of the strategies defined for the selected component.
	Enter the version number to import. Enter ***RELEASE** to refer to the default release defined in the DROPS Studio.

Table 32: Configure DROPS: Import steps in Jenkins

Save the job’s configuration.

52.3.2 Defining DROPS: Deploy steps

Define the following parameters for each DROPS: Deploy step added to a Jenkins project. These parameters define the process to deploy. They are the same parameters required to define a deployment process instance in the DROPS Studio.

The deployment is launched via Jenkins in the DROPS Studio.

> ***Reference***
> For more information about defining import process instances, refer to Introduction to deploying releases on page 257.

Field	Description
Application	Select the application for which to deploy the artifacts. The list is pre-populated with all of the applications defined in the DROPS Server.
Release	Select the release to deploy. If you select an existing release, the list of releases to select is pre-populated with the releases that have already been imported.
Environment	Select the environment to which to deploy the release.
Process	Select the deployment process to use. The list is prepopulated with the processes available for the selected environment.

Table 33: Configure DROPS: Deploy steps in Jenkins

Save the job’s configuration.

52.3.3 Adding scripted steps to a pipeline project

If your Jenkins project is a pipeline, generate the script for each DROPS step using the Pipeline Syntax generator in the **Pipeline** section of the project's confirmation view.

Step 1 From a pipeline project's Pipeline section, click the **Pipeline Syntax** link.

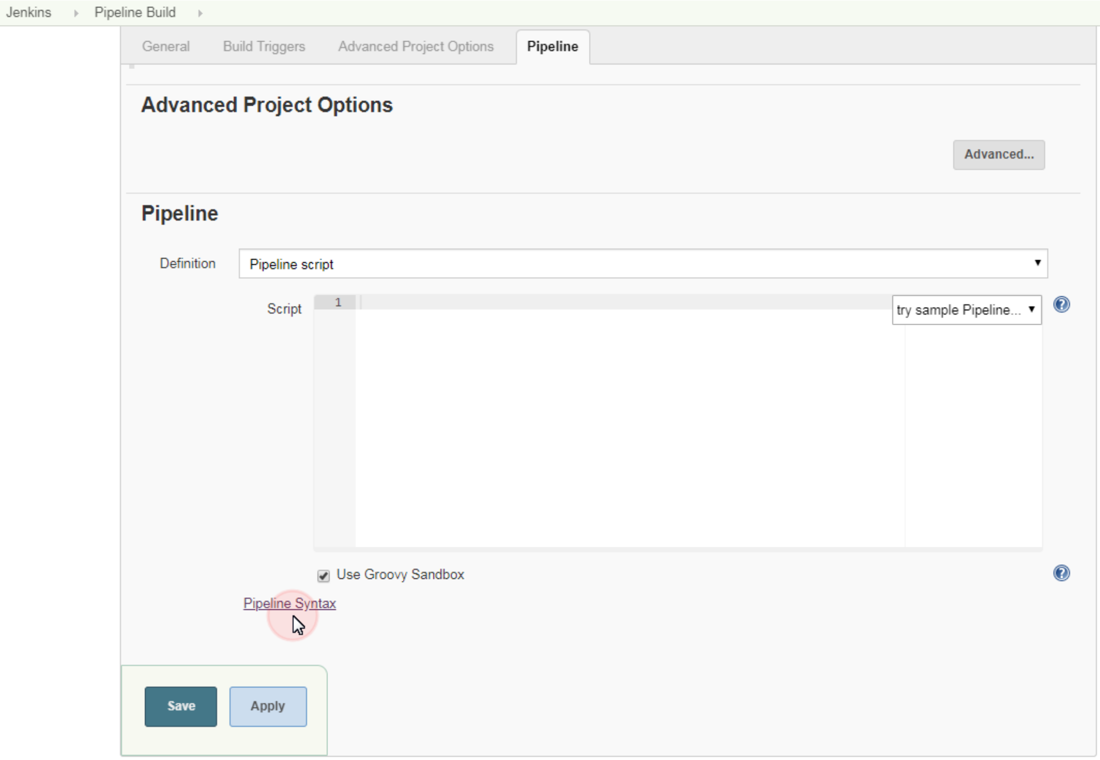
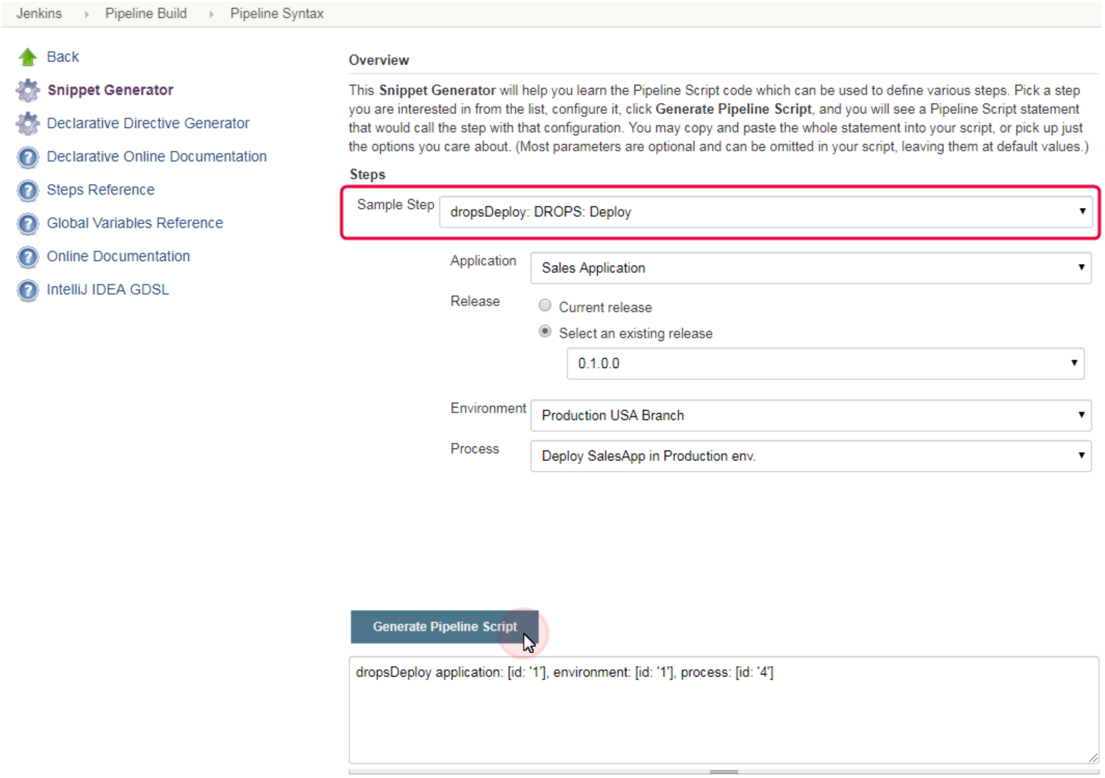


Figure 119: Jenkins pipeline project: add DROPS steps

Step 2 Select a DROPS import or deployment step from the **Step** menu and then define the parameters required. The parameters for each type of step are described above.



Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step: dropsDeploy: DROPS: Deploy

Application: Sales Application

Release: ○ Current release ● Select an existing release: 0.1.0.0

Environment: Production USA Branch

Process: Deploy SalesApp in Production env.

```
dropsDeploy application: [id: '1'], environment: [id: '1'], process: [id: '4']
```

Figure 120: Jenkins pipeline syntax for DROPS steps

Step 3 Click the **Generate Pipeline Script** button.

Note

The technical IDs are used by default when generating the code in the pipeline script. To change the IDs to the entity's code or release number (to better understand the written script) you can do so manually.

Example

dropsDeploy application: [id: '1'], environment: [id: '1'], process: [id: '4'], release: [$class: 'ExistingRelease', id: '2']

is the same as the following, once the IDs are changed to the entities' names:

dropsDeploy application: [code: 'SalesApp'], environment: [code: 'PRD-USA'], process: [code: 'PRD-DEPLOY-SALESAPP'], release: [$class: 'ExistingRelease', number: '0.1.0.0']

Step 4 Copy the script and paste it into your project's **Script** section.

52.4 Understanding results in the DROPS for Jenkins plug-in

DROPS Server logs are available in Jenkins after an import or deployment step has been carried out.

A process can have one of three statuses in Jenkins, each which correspond to an execution status in the DROPS Studio.

DROPS Studio	Jenkins
▭ In Progress	Tasks stay "in progress" until any and all issues are handled in the DROPS Studio.
🧯 Suspended	Unstable
☒ Canceled	Unstable
🟢 Succeeded	Successful
🟠 Succeeded with Warnings	Successful
🔴 Failed	Failure

Table 34: Deployment process instance statuses in Jenkins

53 DROPS for JIRA

Chapter Summary

53.1 Main concepts 370
53.2 Functional constraints 371
53.3 Prerequisites 371
53.4 Installing the DROPS for JIRA add-on 372
53.5 Configuring the DROPS for JIRA add-on 374
53.6 Adding the request to deploy to your development project workflow 379
53.7 Define the request/requested by link between issues 384
53.8 Adding the instruction to launch the deployment to your DROPS project workflow 385
53.9 Adding custom fields to deployment request issues in your DROPS project 387

The DROPS for JIRA add-on allows you to add fields to your issues and actions to your workflows in order to trigger the execution of import and deployment processes via JIRA. When a DROPS process is launched via JIRA, the parent JIRA issue is linked to every part of the process and can be opened via the DROPS Studio.

The DROPS for JIRA add-on offers a post-function to add a request for deployment to your JIRA workflow. When the request is launched, a deployment request issue is created in JIRA and the process is launched in the DROPS Studio.

This chapter describes how to install, configure and use the DROPS for JIRA add-on.

> **Note**
> Only one DROPS Server can be piloted by JIRA. The DROPS Server linked to JIRA will be the only one for which you can select pre-defined import and deployment processes to launch.

> ***Reference***
> For more information about how to configure the required elements in the DROPS Studio in order for the add-on to function, refer to Workitems on page 401.

53.1 Main concepts

The DROPS for JIRA add-on is an extension of the DROPS tool. It allows you to create deployment requests via your JIRA workflow and to execute the import and deployment processes via a new workflow dedicated to a DROPS project in JIRA.

An action is added to your JIRA workflow which allows you to request a deployment. Selecting this action will create an issue (linked to your project's issue that sends the request) in a different project dedicated to the DROPS workflow.

This linked deployment request issue has its individual workflow so that the deployment process can be managed outside of your development project's workflow. It is this issue that will be linked to all of your entities in DROPS and that includes the button to execute the import and deployment processes via JIRA.

The import and deployment are managed in the DROPS Server, as usual, and the results are relayed to the JIRA execution issue.

The JIRA issue is updated and the parent request issue can transition to the next step and advance depending on the deployment results.

53.2 Functional constraints

1. DROPS Software assumes that you are already familiar with the basics of DROPS and JIRA. This documentation does not cover how to use JIRA.
2. When an import process instance is executed, all of the import strategies defined in every component included in the application is carried out. This is the equivalent to creating an import process instance with "no filter".

> ***Reference***
> For more information about the different filter options for import execution instances, refer to Step 6   Select the import strategy(ies) to use to feed the selected release. inCreate new import process instances on page 250.

3. When the import process is executed via JIRA, the current release defined in the application is always used to receive the imported artifacts. You cannot select a different release or create new releases via JIRA. The current release must already exist and be clearly defined in the application.

> ***Reference***
> For more information about how the current release is overridden when used during the import process, refer to Step 3   Select the release in which to save the artifacts that will be imported. inCreate new import process instances on page 250.

4. When the deployment process is executed via JIRA, the default deployment process in the target environment is always used. The default deployment process must already exist and be clearly defined in the environment. If no process is defined as default and there are more than one processes defined in the environment, the process will fail. If there is only one process defined in the environment it will be used to execute, even if it is not defined as the default process.
5. JIRA does not support special characters when defining field and transitions.

53.3 Prerequisites

System requirements:

1. JRE/JDK v8
2. JIRA must be installed on a server. The add-on is compatible with JIRA ≥v**7.2.2**. For lower versions, the interface may be a little different.

Reference

For more information about installing JIRA, refer to the JIRA installation instructions and JIRA requirements.

DROPS-specific prerequisites:

1. This plug-in does not offer any DROPS configuration features. To execute import and deployment processes in DROPS via JIRA, they must already be completely configured in the DROPS Studio.
2. Every application associated with a JIRA project must already have an existing current release.
3. You must create at least one JIRA project dedicated to the DROPS deployment execution workflow.

53.4 Installing the DROPS for JIRA add-on

Follow the subsequent steps to install an ARCAD add-on in JIRA.

Step 1 Open JIRA in a navigator and login as the administrator.

Step 2 To access the add-on install page, select the settings menu, then **Add-ons**.

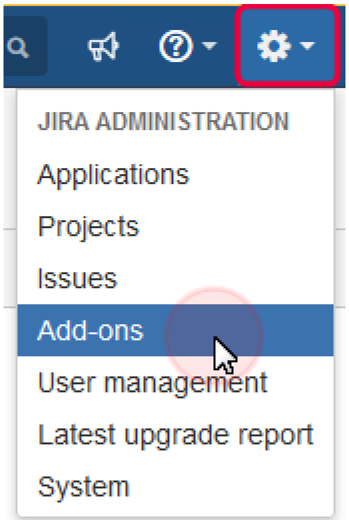
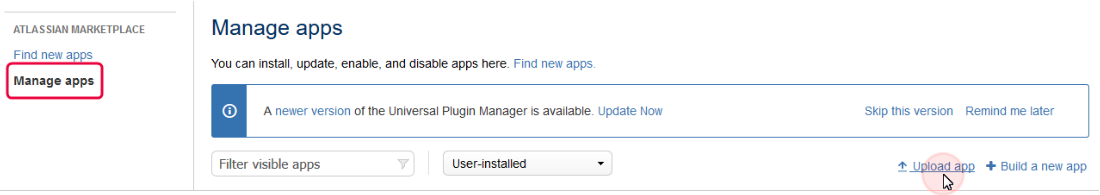


Figure 121: The JIRA settings menu: Add-ons

Step 3 Click on **Manage apps** then upload an external plug-in.



Manage apps

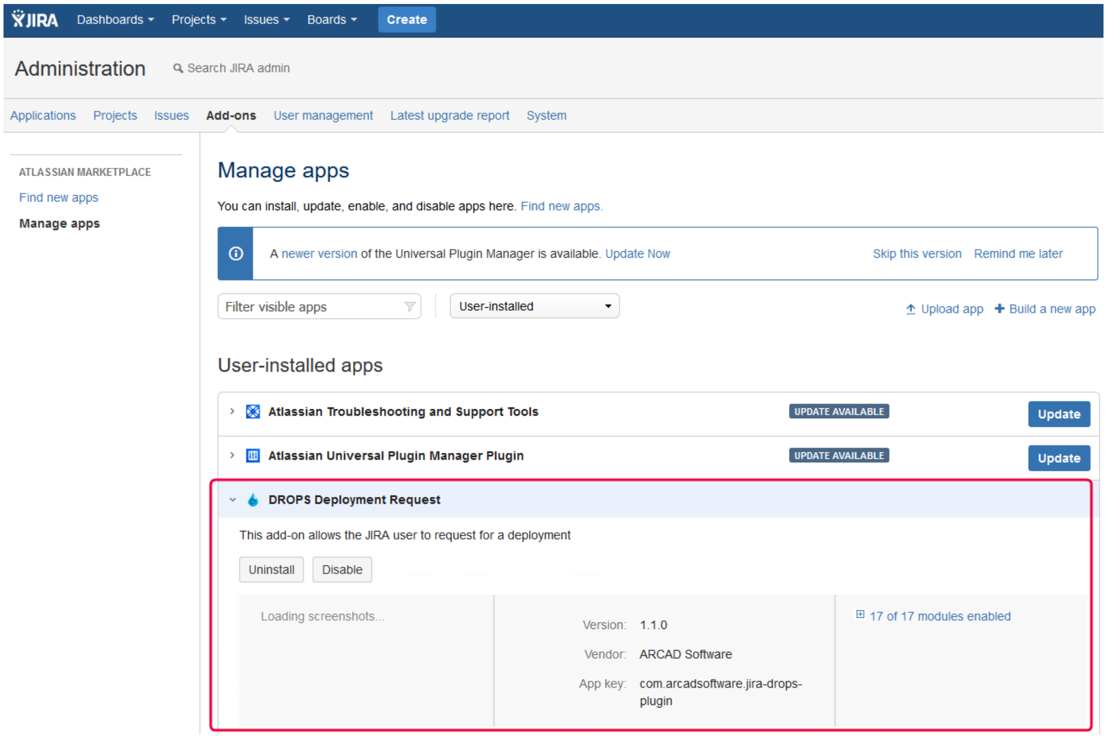
You can install, update, enable, and disable apps here. Find new apps.

A newer version of the Universal Plugin Manager is available. Update Now Skip this version Remind me later

Figure 122: Upload external app to JIRA

Step 4 Navigate to the location where the installation file is stored. Select the *.jar* file that corresponds to the product and click **Open**. This installation file can be found in the root of the installation materials.

Result Refresh the page or your connection to see that the add-on is installed and ready to be configured.



Manage apps

You can install, update, enable, and disable apps here. Find new apps.

A newer version of the Universal Plugin Manager is available. Update Now Skip this version Remind me later

User-installed apps

- Atlassian Troubleshooting and Support Tools
- Atlassian Universal Plugin Manager Plugin
- DROPS Deployment Request

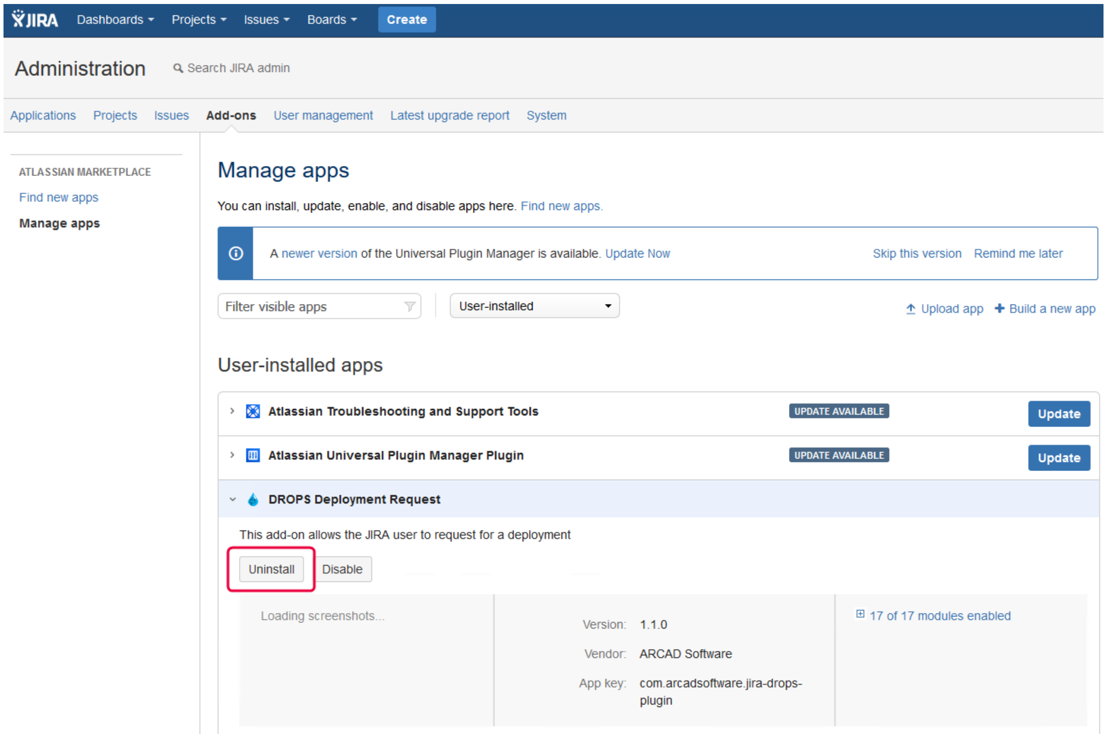
This add-on allows the JIRA user to request for a deployment

Version: 1.1.0
Vendor: ARCAD Software
App key: com.arcadsoftware.jira-drops-plugin

Figure 123: The DROPS for JIRA add-on installed and ready

53.4.1 Uninstall the DROPS for JIRA add-on

To uninstall an add-on from JIRA, access the product's section in the **Manage apps** menu and click on **Uninstall**.



Manage apps

You can install, update, enable, and disable apps here. Find new apps.

A newer version of the Universal Plugin Manager is available. Update Now Skip this version Remind me later

User-installed apps

- Atlassian Troubleshooting and Support Tools
- Atlassian Universal Plugin Manager Plugin
- DROPS Deployment Request

This add-on allows the JIRA user to request for a deployment

Version: 1.1.0
Vendor: ARCAD Software
App key: com.arcadsoftware.jira-drops-plugin

Figure 124: Uninstall the DROPS for JIRA plug-in

53.5 Configuring the DROPS for JIRA add-on

There are four configuration categories to complete in order to fully configure the DROPS for JIRA add-on.

Step 1 To access the DROPS for JIRA configuration categories, select the settings menu, then **Add-ons**.

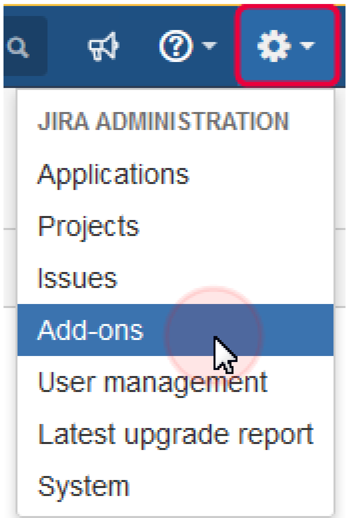
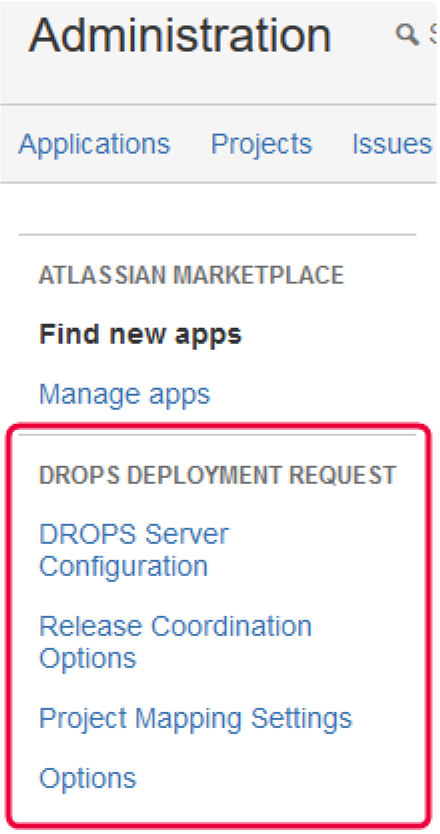


Figure 125: The JIRA settings menu

Step 2 The **DROPS Deployment Request** menu on the left contains the four configuration categories.



Administration

Applications Projects Issues

ATLASSIAN MARKETPLACE

Find new apps

Manage apps

DROPS DEPLOYMENT REQUEST

DROPS Server Configuration

Release Coordination Options

Project Mapping Settings

Options

Figure 126: The DROPS Deployment Request configuration categories

53.5.1 Configure the DROPS Server connection

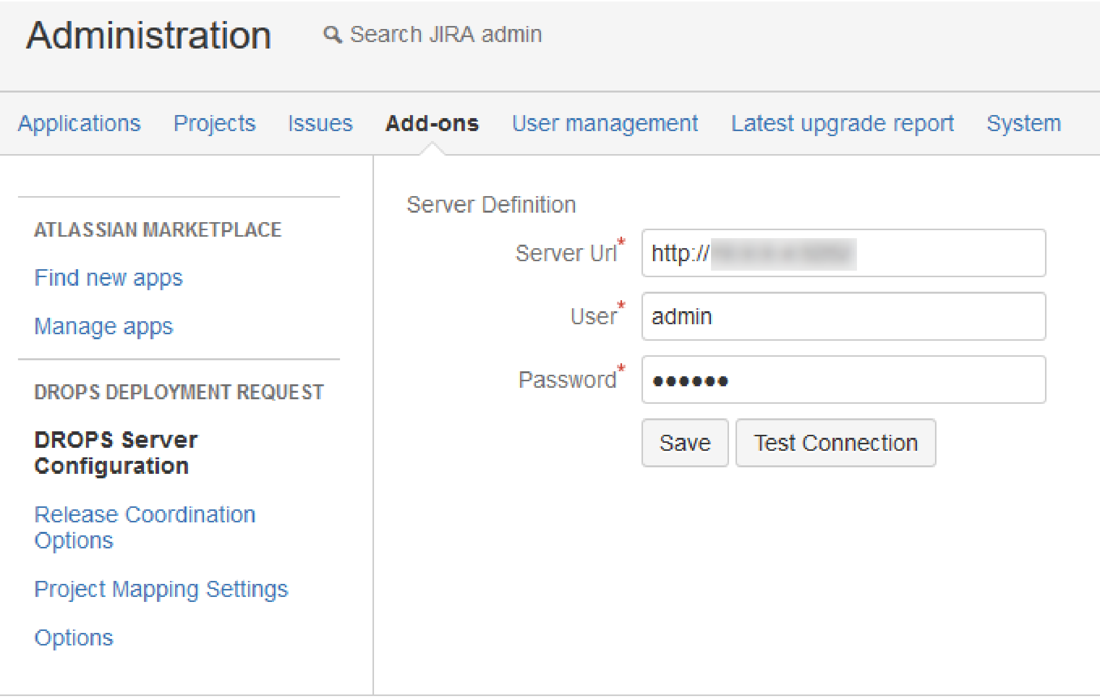
In order for JIRA to communicate with the DROPS Server, the connection must be established.

Step 1 To configure the connection between JIRA and the DROPS Server, click the **DROPS Server Configuration** menu.

Step 2 Enter the full URL to the DROPS Server and the user/password for the DROPS user to use to launch the import and deployment requests.

> **Note**
> The user defined here must already exist on the DROPS Server and have all the corresponding rights to create, edit and execute import and deployment instances.

Step 3 Click the **Test Connection** button to ensure the connection is established correctly.



Server Definition

Server Url* http://[redacted]

User* admin

Password* ●●●●●●

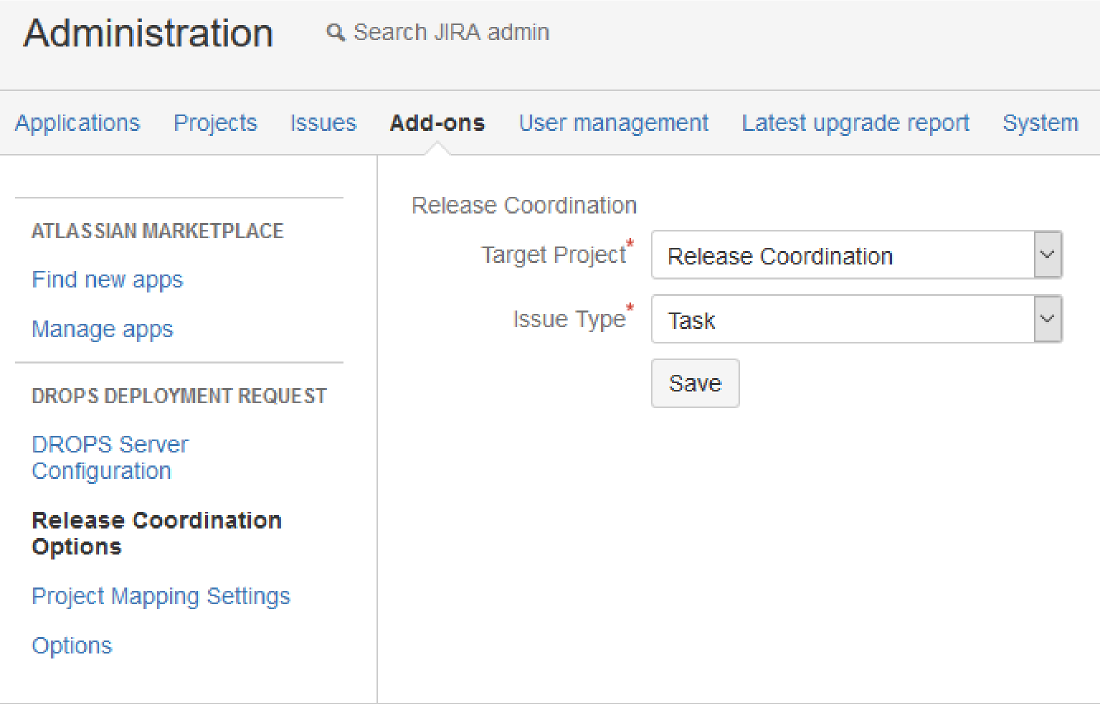
Figure 127: Define the connection to the DROPS Server

53.5.2 Define the type of issue to create in the DROPS project

One prerequisite to using the DROPS for JIRA add-on is that there must be at least one JIRA project in your system dedicated to the workflow that includes actions performed by the DROPS Server. It is in this project that all of the deployment request issues will be launched so that the deploy workflow is a detour and doesn't affect your development projects' workflows. The Release Coordination configuration options allow you to define what type of issue to create in this project.

Step 1 To define the issue types for your DROPS project, click the **Release Coordination Options** menu.

Step 2 Select the project dedicated to the DROPS workflow, then select the type of issue to create for this project.



Release Coordination

Target Project*: Release Coordination

Issue Type*: Task

Figure 128: Define the issue type for project dedicated to DROPS

53.5.3 Map your JIRA projects to your DROPS applications

Every project in which you wish to add a step to launch a DROPS process must be mapped to one pre-configured application on your DROPS Server. Mapping your JIRA project(s) to your application(s) ensures that the processes launched via JIRA correspond to the correct components to deploy and the correct environment to which to deploy.

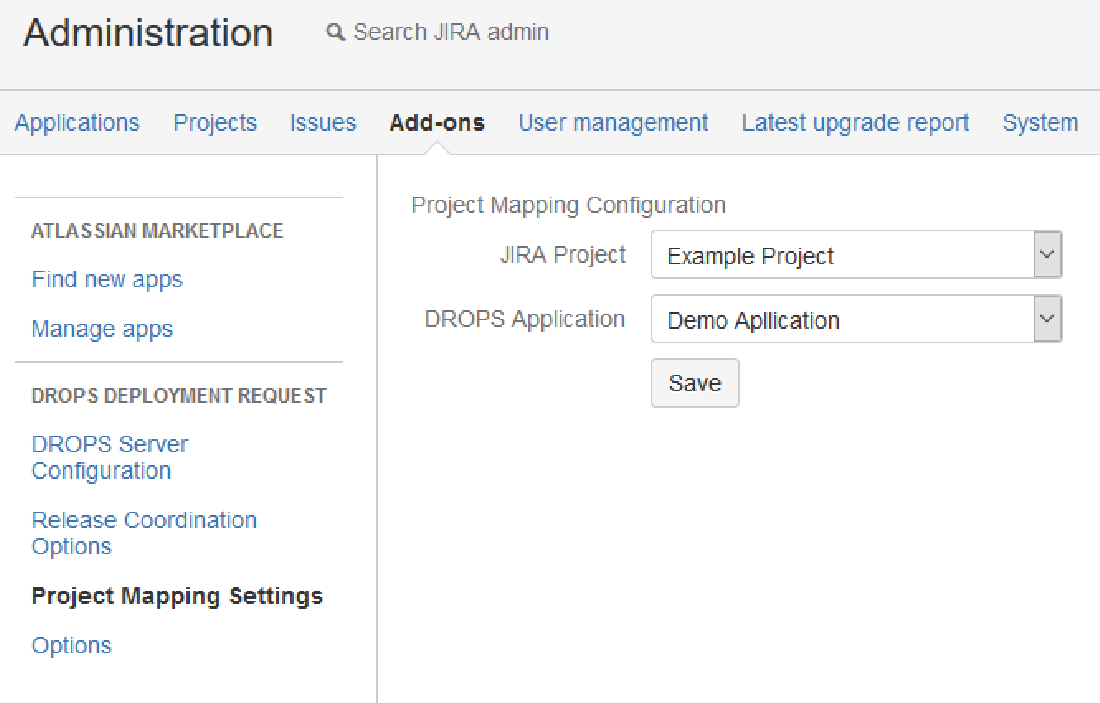
> ***Reference***
> For more information about applications in DROPS, refer to Applications on page 60.

Step 1 To map your JIRA project to a DROPS application, click the **Project Mapping Settings** menu.

Step 2 Click the **Add a new Entry** button.

Step 3 Select the project you will use to launch the deployment processes.

Step 4 Select the application to which to link the selected project. The list of applications is pre-populated with all of the applications currently available on the DROPS Server to which you are connected.



Project Mapping Configuration

JIRA Project: Example Project

DROPS Application: Demo Apllication

Figure 129: Map your JIRA project to a DROPS application

Step 5 Click the **Save** button.

Result The map is configured. You can map as many projects in JIRA to as many applications as you have in DROPS. The only requirements is that each project must be mapped to one application.

53.5.4 Map your DROPS workitem provider to JIRA

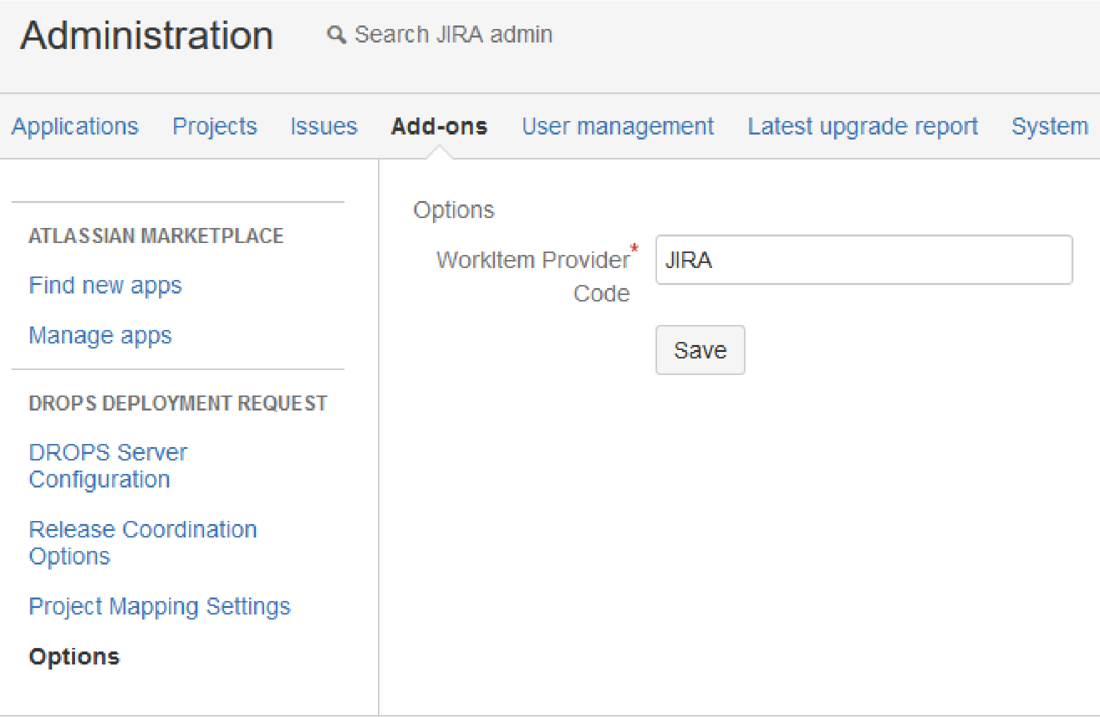
One prerequisite to using the DROPS for JIRA add-on is that you must configure a workitem provider in the DROPS Studio. The workitem provider configuration allows you to map the provider to JIRA so that the DROPS Server and JIRA can communicate.

> ⚠ **Important!**
> Be sure to map the correct workitem provider because the fields defined in the provider must be mapped exactly to the fields in your JIRA projects.

> ***Reference***
> For more information about defining providers in DROPS, refer to Creating workitem providers on page 401.
>
> For more information about the relationship between providers and your JIRA projects refer to Adding custom fields to deployment request issues in your DROPS project on page 387 and Editing JIRA workitems on page 403.

Step 1 To map your workitem provider to JIRA, click the **Options** menu.

Step 2 Enter the **Name** of the workitem provider that you have pre-defined in the DROPS Studio.



Options

WorkItem Provider Code*: JIRA

Figure 130: Map your workitem provider to JIRA

53.6 Adding the request to deploy to your development project workflow

In order to create the DROPS deployment release request via your development project's workflow, you must add a transition to the workflow. This transition must include a screen in which you select the environment to which to deploy the content in the application linked to the project.

Step 1 To add the deployment request transition to your development workflow, select the settings menu, then **Issues**.

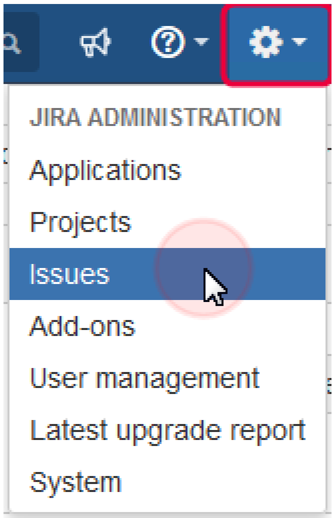


Figure 131: The JIRA settings menu: Issues

Step 2 Create the field from which you will select the target environment when requesting a deployment. This built-in field will display all of the environments in your DROPS Server.

1. Select **Custom Fields** from the left-hand menu.
2. Click the **Add custom field** button.
3. Click **Advanced** in the dialog to access the advanced field options.
4. Select the **DROPS Environment drop-down list** to create a field in which you select the target environment to which to deploy. Click **Next >** to continue.
5. Enter a name and description for the field to help your team understand what they are selecting.
6. Click **Create**.

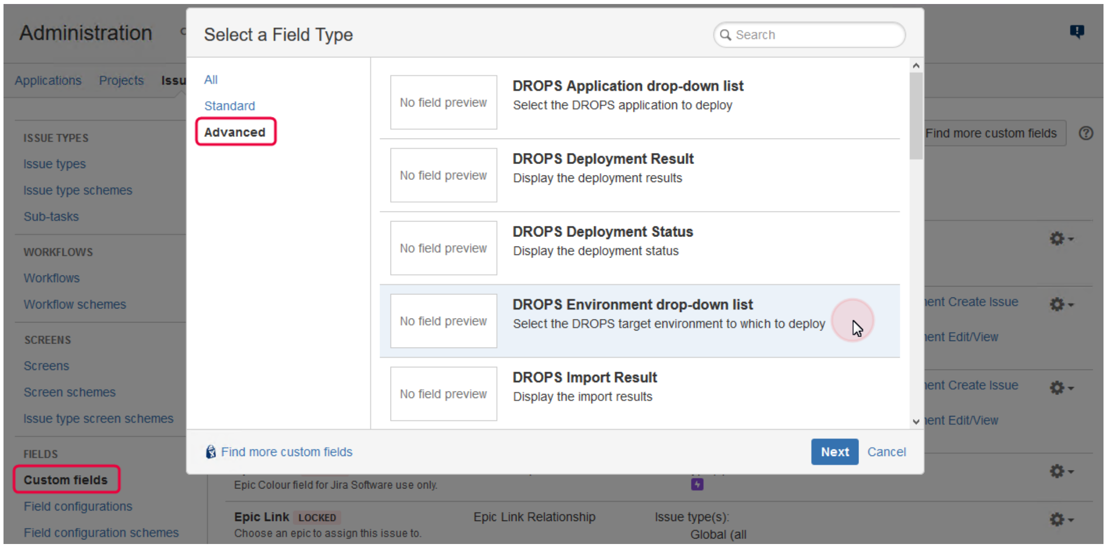
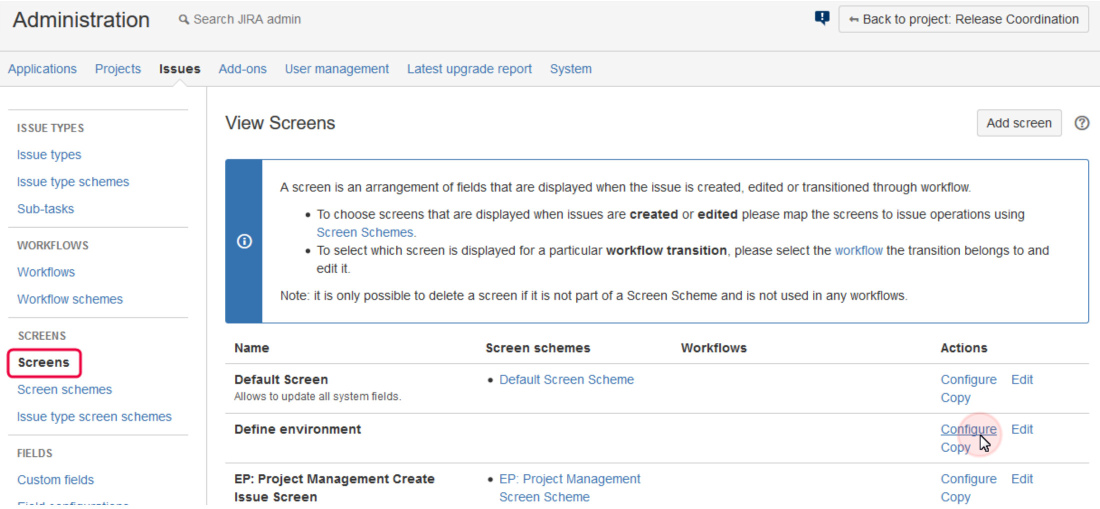


Figure 132: Create the environment selection field

Step 3 Create the screen in which the target environment field will be displayed when requesting a deployment.

1. Select **Screens** from the left-hand menu.
2. Click the **Add screen** button.
3. Enter a name and description for the screen.
4. Click **Add** to create the screen. You will be redirected to the **Screens** page where the new screen appears in the list.
5. Click the **Configure** link to add the field that you created in the previous step to the screen.
6. From the **Field name** list, select the **DROPS Environment** field, then click **Add**.



A screen is an arrangement of fields that are displayed when the issue is created, edited or transitioned through workflow.

- To choose screens that are displayed when issues are **created** or **edited** please map the screens to issue operations using Screen Schemes.
- To select which screen is displayed for a particular **workflow transition**, please select the workflow the transition belongs to and edit it.

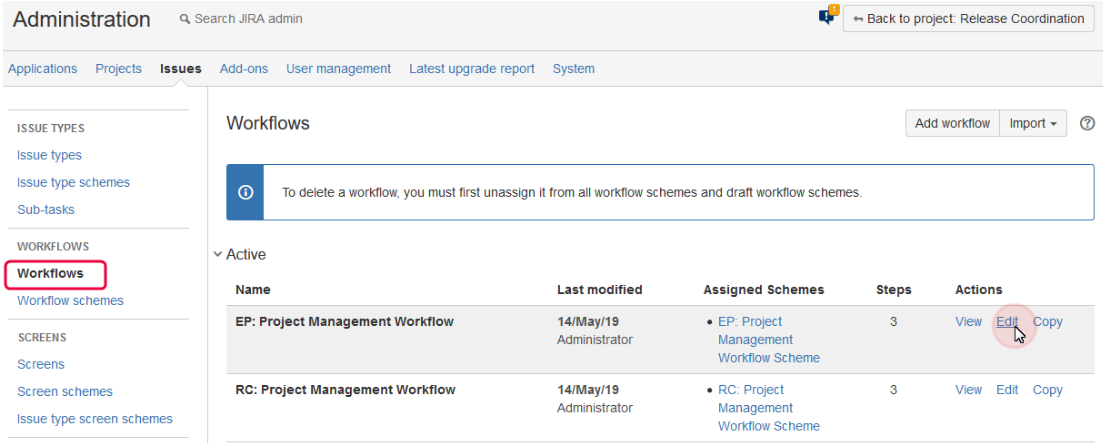
Note: it is only possible to delete a screen if it is not part of a Screen Scheme and is not used in any workflows.

Name	Screen schemes	Workflows	Actions
Default Screen Allows to update all system fields.	• Default Screen Scheme		Configure Edit Copy
Define environment			Configure Edit Copy
EP: Project Management Create Issue Screen	• EP: Project Management Screen Scheme		Configure Edit Copy

Figure 133: Create the environment selection screen

Step 4 Modify your development project's workflow to include the deployment request transition.

1. Select **Workflows** from the left-hand menu.
2. Select your development project or any project in which you want to add a transition to request a deployment.
3. Click **Edit** to modify the project's workflow.
4. Click the **Add transition** button.



Workflows

To delete a workflow, you must first unassign it from all workflow schemes and draft workflow schemes.

Active

Name	Last modified	Assigned Schemes	Steps	Actions
EP: Project Management Workflow	14/May/19 Administrator	• EP: Project Management Workflow Scheme	3	View Edit Copy
RC: Project Management Workflow	14/May/19 Administrator	• RC: Project Management Workflow Scheme	3	View Edit Copy

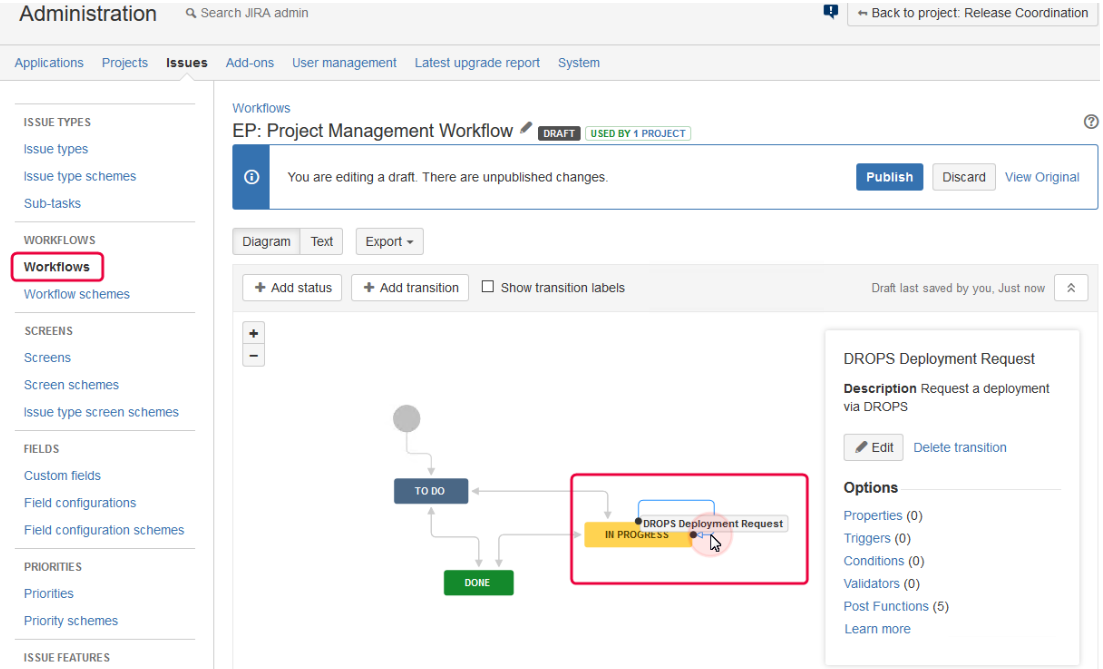
Figure 134: Modify your development project's workflow

Step 5 Define the transition.

1. Select the statuses between which you want to add the deployment request.
2. Define a name for the transition.
3. Select the environment selection screen you created in the previous steps in this procedure to include the screen in the transition.

Step 6 Click **Save**.

> ***Example***
>
> In the following example, we have created a loop transition in the "In progress" status because the parent issue should not close automatically after the deployment is completed. You can add the transition anywhere you need to in your project workflow.



You are editing a draft. There are unpublished changes.

☐ Show transition labels

Draft last saved by you, Just now

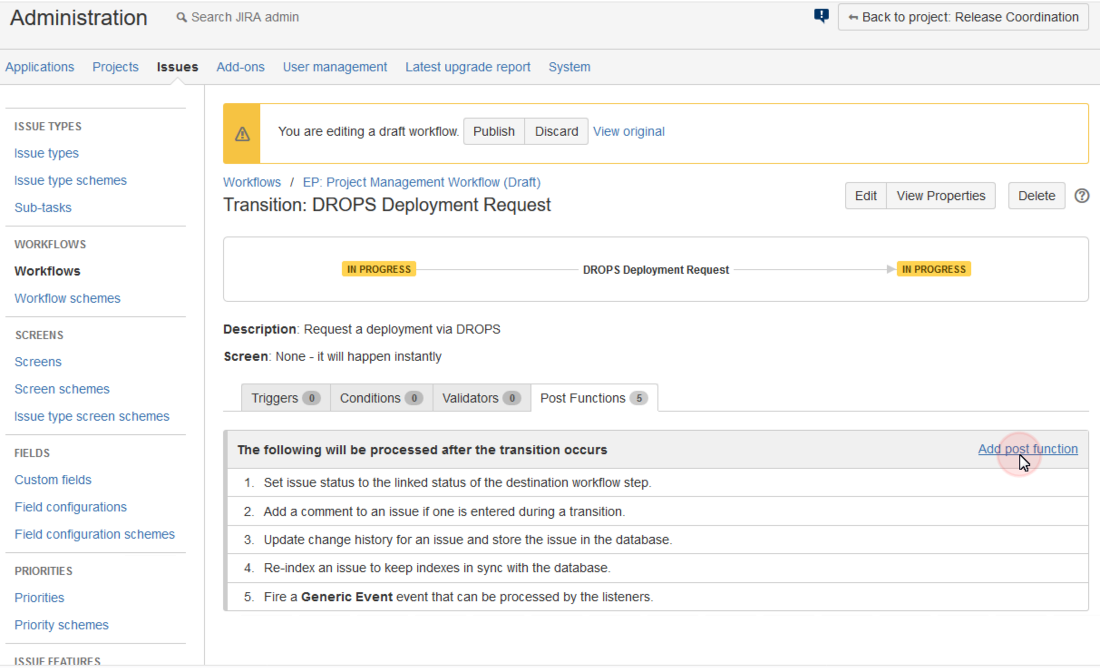
DROPS Deployment Request

Description Request a deployment via DROPS

Figure 135: Add a deployment request transition to your development workflow

Step 7 Add the **DROPS Deployment Request** post function to the transition. This built-in function is the key to creating the new issue in the project dedicated to DROPS that allows you to execute the deployment.

1. In the transition's menu, click on **Post Functions** to add a function to the transition.
2. Click **Add post function**.
3. Select the **DROPS Deployment Request** post function and click the **Add** button.



You are editing a draft workflow.

Workflows / EP: Project Management Workflow (Draft)

Transition: DROPS Deployment Request

Description: Request a deployment via DROPS

Screen: None - it will happen instantly

The following will be processed after the transition occurs

1. Set issue status to the linked status of the destination workflow step.
2. Add a comment to an issue if one is entered during a transition.
3. Update change history for an issue and store the issue in the database.
4. Re-index an issue to keep indexes in sync with the database.
5. Fire a **Generic Event** event that can be processed by the listeners.

Figure 136: Add a post function to the deployment transition

Step 8 Click the **Publish** button at the top of the screen to save all the changes made to the workflow.

Result The workflow has been modified to include the transition for requesting a deployment.

53.7 Define the request/requested by link between issues

Define the process to automatically link deployment request issues in the DROPS project to the parent issue in your development projects. This link ensures that every issue created from the deployment request transition is related to the development issue that requested it.

> **Note**
> The link defined here is added automatically to issues in your development projects and the DROPS project by the **DROPS Deployment Request Post Function** that you added to your development workflow.

Step 1 Select **Issue linking** from the left-hand menu.

Step 2 At the bottom of the page, define the Name of the link in the **Add a new link type** section.

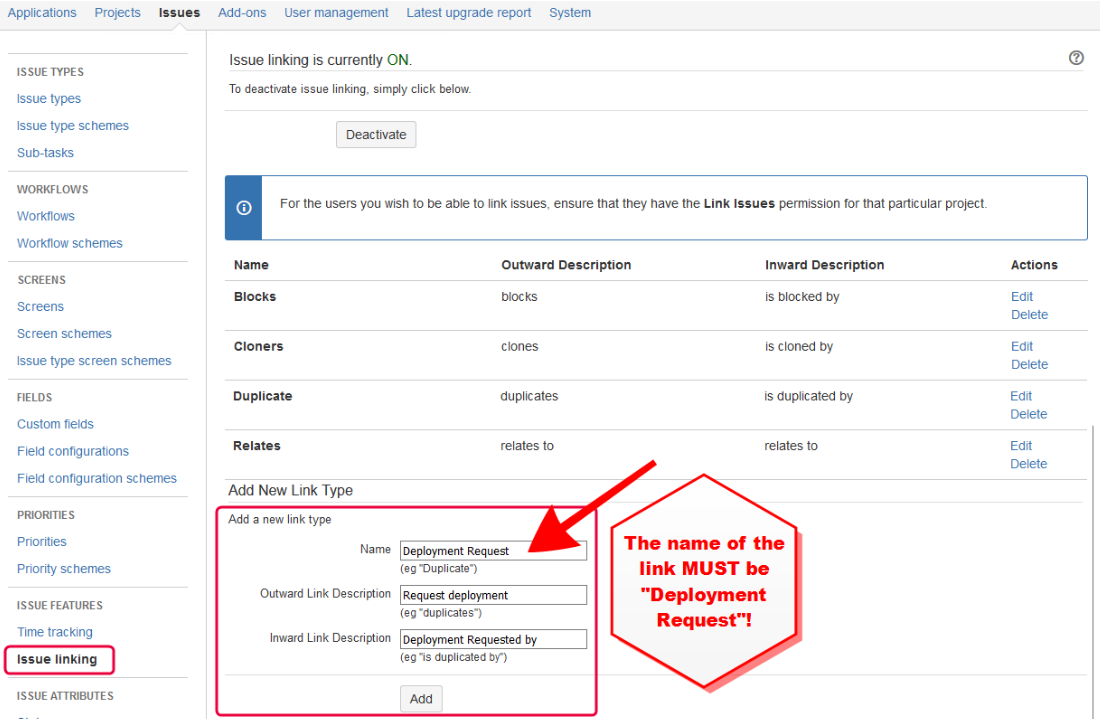
The name must be "Deployment Request".

> **Important!**
> The name of the link between development and deployment-request issues *MUST* be "Deployment Request" in order for the **DROPS Deployment Request Post Function** to add it to each

⚠ issue automatically.

Define the description of the outward and inward link descriptions. This text will appear in each issue next to the link to describe from/to where the link is pointing (the relationship between the issues).

Step 3 Click the **Add** button.



Issue linking is currently ON.

To deactivate issue linking, simply click below.

Deactivate

For the users you wish to be able to link issues, ensure that they have the **Link Issues** permission for that particular project.

Name	Outward Description	Inward Description	Actions
Blocks	blocks	is blocked by	Edit Delete
Cloners	clones	is cloned by	Edit Delete
Duplicate	duplicates	is duplicated by	Edit Delete
Relates	relates to	relates to	Edit Delete

Add New Link Type

Add a new link type

Name: Deployment Request
(eg "Duplicate")

Outward Link Description: Request deployment
(eg "duplicates")

Inward Link Description: Deployment Requested by
(eg "is duplicated by")

Add

The name of the link MUST be "Deployment Request"!

Figure 137: Create the link type to add to the issue pairs

53.8 Adding the instruction to launch the deployment to your DROPS project workflow

In order to execute the deployment process via your DROPS project's workflow, you must add a transition to the workflow.

Step 1 Create the execute import & deployment screen to attach to the transition you will add to the DROPS workflow.

1. Select **Screens** from the left-hand menu.
2. Click the **Add screen** button.
3. Enter a name and description for the screen.
4. Click **Add** to create the screen. You will be redirected to the **Screens** page where the new screen appears in the list.

This screen does not require any fields.

Step 2 Modify the DROPS project's workflow to include the import and deployment execution transition.

1. Select **Workflows** from the left-hand menu.
2. Select the DROPS project.
3. Click **Edit** to modify the project's workflow.
4. Click the **Add transition** button.

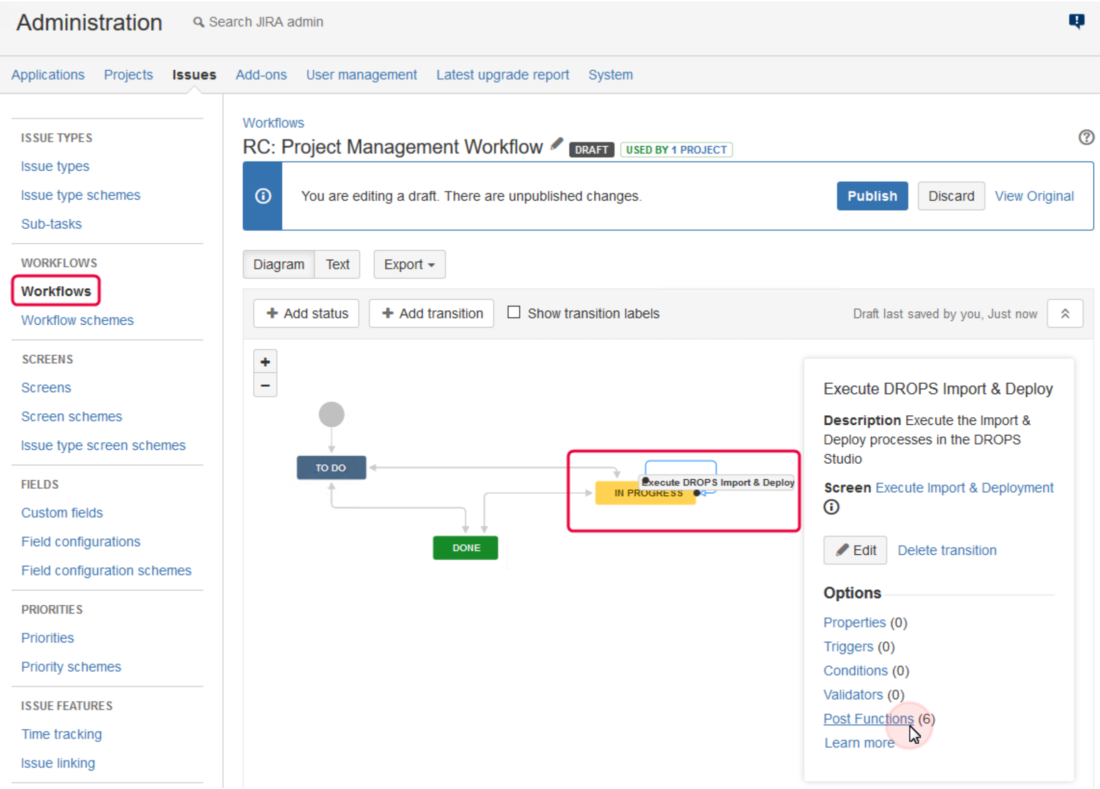
Step 3 Define the Import and Deployment Execution transition.

1. Select the statuses between which you want to add the import and deployment execution.
2. Define a name for the transition.
3. Select the execute import & deployment screen you created in the previous steps in this procedure to include the screen in the transition.

Step 4 Click **Add**.

Step 5 Add the **DROPS Execute Import and Deploy** post function to the execution transition. This built-in function is the key to launching the import and deployment execution instances in DROPS.

1. In the transition's menu, click on **Post Functions** to add a function to the transition.
2. Click **Add post function**.
3. Select the **DROPS Execute Import and Deploy** post function and click the **Add** button.



You are editing a draft. There are unpublished changes.

Draft last saved by you, Just now

Execute DROPS Import & Deploy

Description Execute the Import & Deploy processes in the DROPS Studio

Screen Execute Import & Deployment

Figure 138: Add an import and deployment execution transition to your DROPS workflow

Step 6 Click the **Publish** button at the top of the screen to save all the changes made to the workflow.

Result The workflow has been modified to include the transition to execute import and deployment processes.

53.9 Adding custom fields to deployment request issues in your DROPS project

The deployment request issues in your DROPS project are created automatically when a deployment is requested via the new workflow in your development project (see Adding the request to deploy to your development project workflow on page 379).

This section describes how to add custom fields to these issues in order to display the statuses and results of the import and deployment processes launched in DROPS. These fields correspond to the fields of the same name that you must define in the Workitem provider editor in the DROPS Studio.

The **Target environment** field displays the environment to which the deployment is made.

There are four built-in fields that relay information concerning the import and deployment processes sent by DROPS to JIRA.

- **DROPS Deployment Result**
- **DROPS Deployment Status**

- **DROPS Import Result**
- **DROPS Import Status**

> **Note**
>
> All four of the status/result field types must be defined to fully configure JIRA to work with DROPS. The target environment field is optional but very helpful.

> ***Reference***
>
> For more information about configuring the workitem provider to populate the status/result fields in your JIRA deployment request issue, refer to Editing JIRA workitems on page 403.

Step 1 Select the settings menu, then **Issues**.

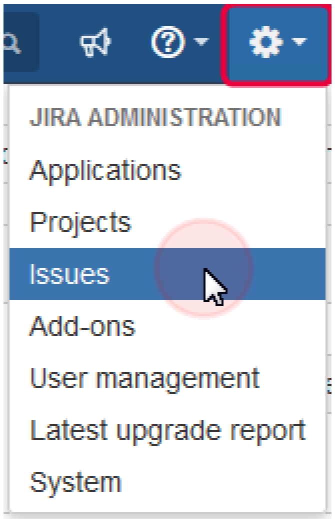


Figure 139: The JIRA settings menu: Issues

Step 2 Create the five fields, one-by-one.

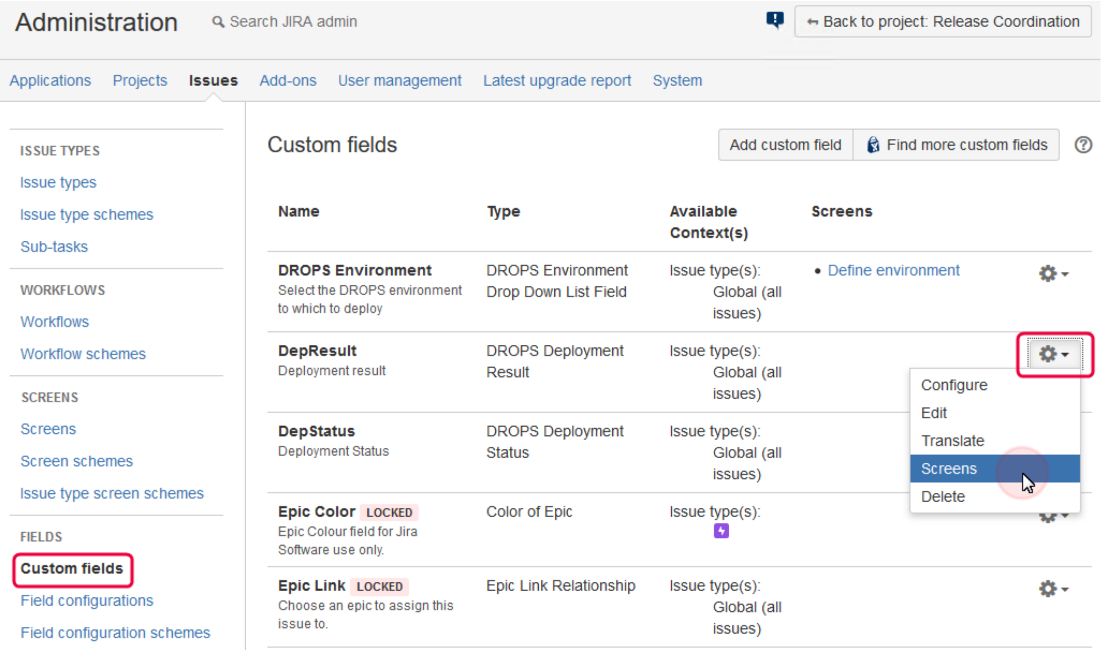
1. Select **Custom Fields** from the left-hand menu.
2. Click the **Add custom field** button.
3. Click **Advanced** in the dialog to access the advanced field options.
4. Select the field to create. Click **Next >** to continue.
5. Enter a name and description for the field to help your team understand what they are selecting.

> **Important!**
>
> The name of each status/result field **MUST** correspond exactly to the four fields defined in the JIRA workitem provider in the DROPS Studio!

6. Click **Create**.

Step 3 Add the custom fields to the **Create issue** and to the **Edit/View Issue** screens in the DROPS project.

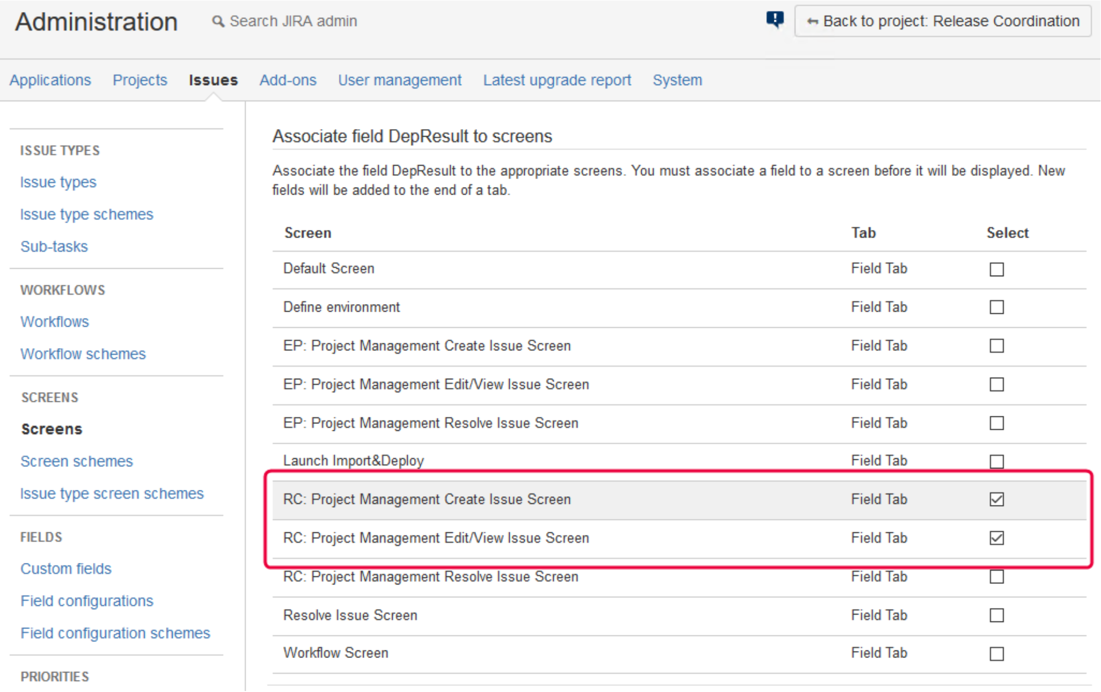
1. For each new field, click the settings icon and select **Screens**.



Custom fields

Name	Type	Available Context(s)	Screens
DROPS Environment Select the DROPS environment to which to deploy	DROPS Environment Drop Down List Field	Issue type(s): Global (all issues)	• Define environment
DepResult Deployment result	DROPS Deployment Result	Issue type(s): Global (all issues)	
DepStatus Deployment Status	DROPS Deployment Status	Issue type(s): Global (all issues)	
Epic Color LOCKED Epic Colour field for Jira Software use only.	Color of Epic	Issue type(s):	
Epic Link LOCKED Choose an epic to assign this issue to.	Epic Link Relationship	Issue type(s): Global (all issues)	

2. Check the boxes for both the [DROPS] **Project Management Create Issue Screen** and the **Project Management Edit/View Issue Screen**.



Associate field DepResult to screens

Associate the field DepResult to the appropriate screens. You must associate a field to a screen before it will be displayed. New fields will be added to the end of a tab.

Screen	Tab	Select
Default Screen	Field Tab	□
Define environment	Field Tab	□
EP: Project Management Create Issue Screen	Field Tab	□
EP: Project Management Edit/View Issue Screen	Field Tab	□
EP: Project Management Resolve Issue Screen	Field Tab	□
Launch Import&Deploy	Field Tab	□
RC: Project Management Create Issue Screen	Field Tab	☑
RC: Project Management Edit/View Issue Screen	Field Tab	☑
RC: Project Management Resolve Issue Screen	Field Tab	□
Resolve Issue Screen	Field Tab	□
Workflow Screen	Field Tab	□

Figure 140: Add the DROPS custom fields to the deployment request issue's screens

3. Click the **Update** button at the bottom of the page.

54 DROPS for JIRA Cloud

Chapter Summary

54.1 Installation 391
54.2 Configuring the DROPS Server connection 391
54.3 Configuring the JIRA user's roles and rights (permissions) 392
54.4 Mapping DROPS applications 393
54.5 Mapping DROPS environments 393
54.6 Importing 393
54.7 Deploying 394

The DROPS for JIRA Cloud add-on allows you to create and execute import and deployment requests via your JIRA Cloud projects and issues.

54.1 Installation

In the Apps management section, click the **Upload app** link and enter the following app descriptor URL to upload the app and install:

https://drops-jira-cloud.arcadsoftware.com

Click **Upload**.

> ***Reference***
> For more information about installing add-ons with the UMP, refer to the Atlassian Universal Plugin Manager documentation.

54.2 Configuring the DROPS Server connection

The server connection form in the App management section is used to ensure that the connectivity with the DROPS Server is successful. This task is an administrative task to finalize the installation of the add-on.

- To configure the connection to a DROPS Server, go to **Apps** menu > **DROPS Release Manager** > **DROPS Server Configuration**.
- Define the **DROPS Server URL**, the **User** and the **Password** to connect to the server. Click **Save**.
- Click **Test connection** to ensure the app is correctly connected to the DROPS Server.

The access to the DROPS Server is defined for each user in the user preferences. The DROPS users access the server with their own credentials.

- To configure the access to the DROPS Server go to *[your user profile]* > **DROPS Credentials**.
- Define the **DROPS Server URL**, the **User** and the **Password** to connect to the server. Click **Save**.
- Click **Test connection** to ensure the app is correctly connected to the DROPS Server.

54.3 Configuring the JIRA user's roles and rights (permissions)

You must have project administrator privileges to configure the permissions for the DROPS app.

The DROPS for JIRA Cloud app has the following project permissions:

DROPS application mapping

Configuring DROPS application mapping in project configuration.

DROPS environment mapping

Configuring DROPS environment mapping for deployment in project configuration.

DROPS release deployment from project

Creating DROPS deployments from the project.

DROPS release imports from project

Creating DROPS imports from the project.

The DROPS for JIRA app has the following issue permissions:

DROPS release deployment from issues

Creating DROPS deployment from issues.

DROPS release import from issues

Creating DROPS import from issues.

Configure these permissions according to the permissions schema you use for your projects. The application and environment mapping permissions allow to link projects to DROPS applications and environments and the other permissions allow the creation of import instances and deployment instances at project or issue levels.

Example

For a development project which creates versions of an application:

- project administrators can map to a DROPS Application,
- project manager can create import instances from JIRA versions at project level,
- no grants to deployments permissions/environment mapping.

Example

For a project dedicated to deployments:

- project administrators can map to DROPS environments,
- users can create deployment instances from DROPS Releases at issue level,
- no grants to imports permissions/application mapping.

The new JIRA permissions schemas coupled to DROPS existing permissions gives to administrators the flexibility to configure applications access and roles/rights to applications/environments.

54.4 Mapping DROPS applications

This procedure maps a JIRA project to a DROPS application and is mandatory for import actions. When configured, it also limits deployments processes to this application.

To configure the application mapping, go to **Project settings** > **DROPS Import**.

Select a **DROPS Application** in the list and click **Save**.

54.5 Mapping DROPS environments

This procedure maps a JIRA project to DROPS environments and is mandatory for deployment actions. When configured, it limits where deployments processes can be launched.

To configure the application mapping, go to **Project settings** > **DROPS Deploy**.

Select one or several **DROPS Environments** in the list and click **Save**.

54.6 Importing

54.6.1 Importing at project level

Go to **Project** > **DROPS Import**.

The DROPS import view shows all the import instances for the mapped DROPS Application.

Each line represents an import instance with relevant information like statuses, dates, releases number and description. The colors reflect the different statuses of the instances. Click the Log icon to download the full formatted log of the instance when the status is completed.

To create a new import instance, click the [...] Action button and select the **Create DROPS Import** action.

Define the necessary elements to create a new import instance and click **Create**.

Version

The JIRA version for which you can create a DROPS Release. The JIRA version name must be the same as the DROPS Release number.

Description

A description of the release you are about to create or modify it already exists.

Reset release

Tick the Reset release option to empty an already existing release before the import instance execution (used only if the release already exists).

Strategy

All the import strategies for all the application components you can activate/deactivate and give the external version if needed (default value is *RELEASE like in the DROPS Studio).

The import instance is executed as soon as it is created.

54.6.2 Importing at issue level

For a JIRA issue, the DROPS tasks are available in the right-hand panel.

Click **DROPS imports** to show the past imports linked to the JIRA issue.

The list is filtered by the Application mapping and the fix versions of the issue and show the status and date of each import instance. Clicking the Log icon allows to download the full formatted logs.

To create a new import instance, click the [...] Action button and select the **Create DROPS Import** action.

Define the necessary elements to create a new import instance and click **Create**.

Version

The JIRA version for which you can create a DROPS Release. The JIRA version name must be the same as the DROPS Release number.

Description

A description of the release you are about to create or modify it already exists.

Reset release

Tick the Reset release option to empty an already existing release before the import instance execution (used only if the release already exists).

Strategy

All the import strategies for all the application components you can activate/deactivate and give the external version if needed (default value is *RELEASE like in the DROPS Studio).

The import instance is executed as soon as it is created.

54.7 Deploying

54.7.1 Deploying at project level

Go to **Project** > **DROPS Deployments**.

The DROPS Deployments view shows all the deployments instances for the mapped DROPS environments. This list can be filtered (limited) to one application if an Application Mapping is configured.

Each line represents a deployment instance with relevant information like statuses, dates, applications and releases number, target environments. The colors reflect the different statuses of the instances. Click the Log icon to download the full formatted log of the instance when the status is completed.

To create a new deployment instance, click the [...] Action button and select the **Create DROPS Deployment** action.

Define the necessary elements to create a new deployment instance and click **Create**.

Environment

The list shows all the available DROPS environments.

Processes

The list shows all the processes for each DROPS environment.

Release

The list shows all the release to deploy for each deployment process.

Transaction ID.

The transaction ID to use (like for the deployments from the DROPS Studio). Use a static format in addition to variables to create IDs your team will recognize.

The deployment instance is executed as soon as it is created.

54.7.2 Deploying at issue level

For a JIRA issue, the DROPS tasks are available in the right-hand panel.

Click **DROPS deployment** to show the past deployments for the mapped environments of the project.

The list is filtered by the Application mapping and the fix versions of the issue and show the status and date of each deployment instance. Clicking the Log icon allows to download the full formatted logs.

To create a new deployment instance, click the [...] Action button and select the **Create DROPS Deployment** action.

Define the necessary elements to create a new deployment instance and click **Create**.

Environment

The list shows all the available DROPS environments.

Processes

The list shows all the processes for each DROPS environment.

Release

The list shows all the release to deploy for each deployment process.

Transaction ID.

The transaction ID to use (like for the deployments from the DROPS Studio). Use a static format in addition to variables to create IDs your team will recognize.

The deployment instance is executed as soon as it is created.

55 DROPS for Azure DevOps

Chapter Summary

55.1 Compatibility ... 396
55.2 DROPS Server connection ... 396
55.3 Configuring DROPS's pipeline tasks ... 397

The DROPS plug-in for Azure DevOps is an open-source tool for continuous integration. The plug-in is installed from the Azure marketplace. It is found with the name **DROPS Integration**.

The DROPS for Azure DevOps plug-in is intended for release managers. It enables you to trigger the execution of import and deployment processes via Azure DevOps pipelines. You can add as many tasks in a pipeline's job as required, and each can call DROPS to execute a different task.

55.1 Compatibility

The DROPS plug-in for Azure DevOps is compatible with:

- TFS 2017 Update 2+
- TFS 2018
- Azure DevOps Server 2019
- Azure DevOps Server 2020
- Azure DevOps Services

55.2 DROPS Server connection

Follow the subsequent steps to set up the connection to the DROPS Server.

Step 1 In the Project Settings > Pipelines > Service Connections, click **Create service connection**.

Step 2 In the **New service connection** dialog, select ARCAD DROPS Server Connections. Click **Next >** to continue.

Step 3 Define the properties of the server connection. Click **Save**.

Server URL (required)

Enter the DROPS **Server URL**, including the port number.

> **Note**
> The server reference must be the one used in the certificate.

Server Key

[Optional] Enter the server key specified in the DROPS Server Certificate in .pem format.

Accept untrusted SSL Certificates

Check this box if you want to allow the DROPS clients to accept self-signed SSL server certificates without installing them into the Azure DevOps service role and/or Build Agent computers.

Authentication

Enter your **Username** and **Password** to access the server.

Details

[Optional] Enter a **Service connection name** and a description for the server connection.

Result The server connection is created as an item in the Azure DevOps plateform.

55.3 Configuring DROPS's pipeline tasks

Configure the following DROPS tasks settings in an Azure DevOps pipeline.

Step 1 Click **+** to add a task to the agent job.

Step 2 Select DROPS import and / or DROPS deploy in the Add task list, and click the **Add** button.

Step 3 Define the task's settings and click **Save & queue**.

Step 4 When all tasks settings are defined, click **Save & queue**.

To run the pipeline manually, select the parameters you need and click the **Save and run** button.

Result The import and deployment tasks are executed in the order defined in the pipeline. Logs are available at the end of the execution.

DROPS Import task settings

Task version

The version number of the Azure DevOps pipeline task.

Display name

The name you wish to display for the Azure DevOps pipeline task.

DROPS Server

Select the server connection you have created in *Service Connections* in the drop-down list.

Application code

The code of the application that owns the target version.

Create Release: Tick to create a release if it does not exist.

Increment part

If the **Create Release** option is selected, define which part of the Release Number is incremented in the Increment Part dropdown list.

Release number

Available only if the **Create Release** box is not ticked. The mumber of the release in which the artifacts must be imported.

Release description

[*Optional*] Enter a description of the release.

Display execution log

Check this option to display the import process log in the task execution result view.

Component options

Select a **Filter Type** in the drop down list.

- **No filter**

  In that case no additional information is required

- **File**

  Specify the path to the filter file.

- **Azure DevOps**

  Enter the external development version number in the **External Release** field.

- **Filtered**

  Define the components to filter using the YAML syntax.

```
components:

- code: <The Component Code #1>

strategy: <The strategy code of the component #i>

externalVersion: <The source external version>

...

- code: <The Component Code #n>

strategy: <The strategy code of the component #j>

externalVersion: <The source external version>
```

Example

```
---

components:

- code: RM-CMP-FS
```

```
strategy: FS-TEMPLATE

externalVersion: "*RELEASE"
```

DROPS Deploy task settings

The DROPS Deploy task is used to deploy Application's Releases on the organization environments

Task version

The version number of the Azure DevOps pipeline task.

Display name

The name you wish to display for the Azure DevOps pipeline task.

DROPS Server

Select the server connection you have created in *Service Connections* in the drop-down list.

Application code

The code of the application that owns the target version.

Release number

The number of the release to be deployed.

Environment Code

The code of the environment on which the release is to be deployed.

Deployment Process

The code of the process used to deploy the release on the environment.

Transaction ID

The ID of the transaction linked to the deployment process instance.

Display execution log

Check this option to display the deploy process log in the task execution result view.

Deployment Process Options

Deployment Process Type

Select the type of deployment process you want to use. There are 2 available values:

- **Existing Process**: will execute the deployment process identified by its code. Enter the code of the process used to deploy the release on the environment in the **Deployment Process Code** field.

- **File**: will execute a deployment process whose definition is defined in the specified file. Enter the path to the file used to deploy the release on the environment for the related application in the **Repository File Name** field.

DROPS Entities Property Update

Properties Provider Type

The type of property provider you want to use. There are 2 available values:

- **Manual**: The properties have to be defined manually.

  Define all the properties in the **Property Definition List**. Declare each property on a unique line using the following syntax:

  [entity].[property_key]=[property_value]

  where [entity] could be environment or application.

- **File**: The properties have to be defined into a property file.

  Define the path to the file that contains the property list. In this file, declare each property on a unique line using the following syntax:

  [entity].[property_key]=[property_value]

  where [entity] could be environment or application.

56 Workitems

Required Roles	n/a
Required Rights (by category)	Workitem Workitem providers
Perspective	Configuration Management
Access	DROPS Server → Configuration → Workitems

Chapter Summary

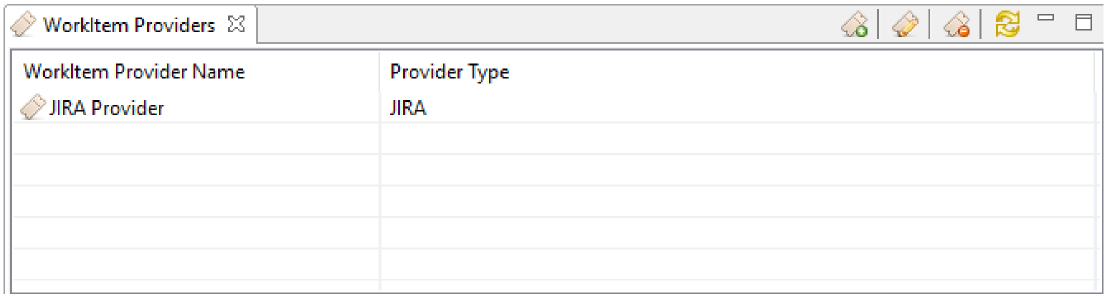
56.1 Workitem providers 401
56.2 Workitems 404

This chapter describes how to configure DROPS to communicate with 3rd party tools via plug-ins and how to access information shared between two tools. Workitems are not required for every 3rd party tool that can pilot DROPS.

56.1 Workitem providers

Workitem providers are the link between DROPS and the 3rd party tool.

The **Workitem providers** are accessed from the **Workitems** node in the **Configuration Explorer**.



WorkItem Provider Name	Provider Type
JIRA Provider	JIRA

Figure 141: The Workitem Providers view

56.1.1 Creating workitem providers

Follow the subsequent steps to create a new workitem provider.

Step 1 To access the **Create a new workitem provider** wizard, either click the create icon in the **Workitem Providers** view or right-click anywhere in the view and select **Create a new workitem provider**.

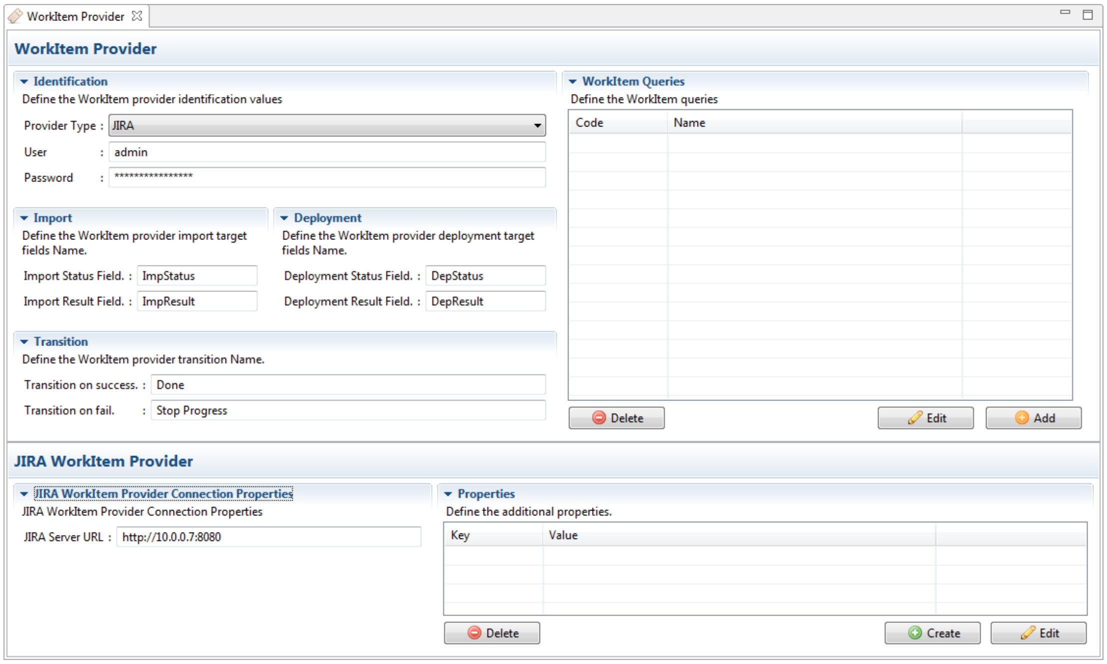
Step 2 Define theProvider Name and Provider Type. Click **Finish**.

Result New workitem providers are available in the **Workitem Providers** view and must be further configured in order to be used to connect to 3rd party applications.

> ***Reference***
> For more information about configuring workitem providers, refer to the section(s) below that corresponds to your 3rd party application.

56.1.2 Editing general workitem provider details

To view or edit a workitem provider, either right-click on it and select **Edit**, select it then click the Edit icon or double-click it to open the **Workitem Provider** editor.



WorkItem Provider

Identification
Define the WorkItem provider identification values
Provider Type : JIRA
User : admin
Password : ****************

Import
Define the WorkItem provider import target fields Name.
Import Status Field. : ImpStatus
Import Result Field. : ImpResult

Deployment
Define the WorkItem provider deployment target fields Name.
Deployment Status Field. : DepStatus
Deployment Result Field. : DepResult

Transition
Define the WorkItem provider transition Name.
Transition on success. : Done
Transition on fail. : Stop Progress

WorkItem Queries
Define the WorkItem queries

Code	Name

JIRA WorkItem Provider

JIRA WorkItem Provider Connection Properties
JIRA WorkItem Provider Connection Properties
JIRA Server URL : http://10.0.0.7:8080

Properties
Define the additional properties.

Key	Value

Figure 142: The Workitem provider editor

> ***Reference***
> For more information about the different actions managed in the **Workitem Provider** editor that are not described in this chapter, refer to:
>
> - Properties on page 412

Provider Name

The name given to the provider.

> **Note**
> This name may be required in the 3rd party application to connect to the DROPS Server.

Provider Type

The type of provider. The configuration options will change depending on the type of provider selected here.

> ***Warning!***
> Changing the type of a workitem provider will delete and/or replace any preexisting configuration already defined for the provider.

User/Password

Enter the user/password required to access the 3rd party application.

Save the changes (🖫, Ctrl+S or **File** > **Save**).

56.1.3 Editing JIRA workitems

The DROPS for JIRA add-on allows you to add fields to your issues and actions to your workflows in order to trigger the execution of import and deployment processes via JIRA. When a DROPS process is launched via JIRA, the parent JIRA issue is linked to every part of the process and can be opened via the DROPS Studio.

When connected to JIRA, issues created in JIRA that execute the import and deployment processes are automatically linked to the corresponding execution instances and the component and release editors.

To share data between DROPS and JIRA, a workitem provider must be defined. This provider provides the connection required between the two tools and allows you to define the fields that will be shared between the issues created in JIRA and the data provided by DROPS.

> ***Reference***
> For more information about how to install, configure and use the DROPS for JIRA plug-in, refer to DROPS for JIRA on page 370.

Edit the following JIRA connection properties:

Import Status field, Import Result field

Enter the exact text to display in your JIRA issue for the field that will be updated with the status and results of an import process execution instance.

Deployment Result field, Deployment Status field

Enter the exact text to display in your JIRA issue for the field that will be updated with the status and results of a deployment process execution instance.

The text in these fields must be referenced word-for-word when configuring the custom fields for your JIRA issues.

> ***Reference***
> For more information about configuring these fields in JIRA, refer to Adding custom fields to deployment request issues in your DROPS project on page 387.

Transition on success, Transition on fail

The **DROPS Execute Import and Deploy** post function requires instructions for how to transition after the import and deployment process instances are complete. Enter the transitions, as defined in your DROPS project in JIRA, here.

> ***Reference***
>
> For more information about the DROPS project in JIRA, refer to Adding the instruction to launch the deployment to your DROPS project workflow on page 385.

JIRA Server URL

Enter the full URL and port to access JIRA.

Save the changes (, Ctrl+S or **File** > **Save**).

56.1.4 Deleting workitem providers

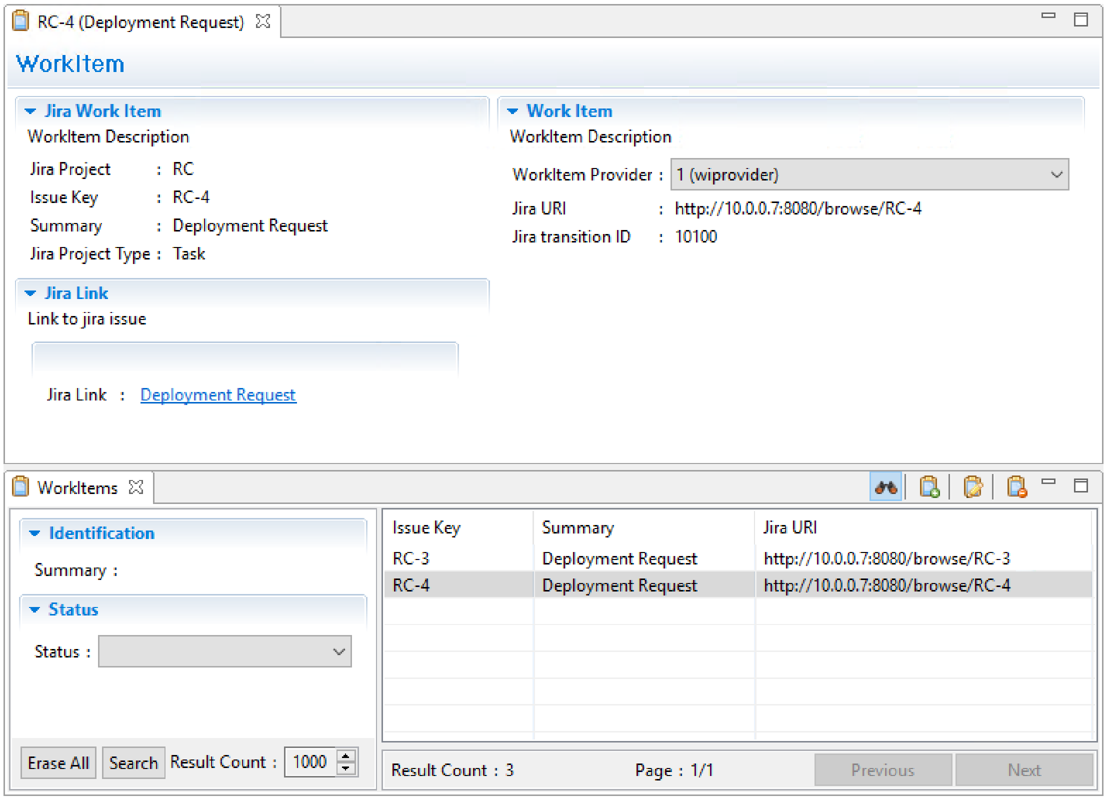
> ***Warning!***
>
> Deleted workitem providers cannot be accessed or recovered.

To delete a workitem provider, either right-click on it in the **Workitem providers** view and select **Delete**, or select it and click the delete icon. Click **OK** to confirm or click **Cancel** to keep the provider.

56.2 Workitems

Workitems are the issues created in the 3rd party tool that pilot DROPS executions.

When working with JIRA, workitems are created automatically by the DROPS Execute Import and Deploy post functions which is launched when a JIRA user executes a deployment process via the DROPS project. JIRA issues are automatically linked to the import and deployment execution instances they create and can be found in the **Workitems** search view. Links to existing JIRA issues can also be added manually to execution instances that were not created by JIRA.



RC-4 (Deployment Request)

WorkItem

Jira Work Item

WorkItem Description

Jira Project : RC
Issue Key : RC-4
Summary : Deployment Request
Jira Project Type : Task

Jira Link

Link to jira issue

Jira Link : Deployment Request

Work Item

WorkItem Description

WorkItem Provider : 1 (wiprovider)
Jira URI : http://10.0.0.7:8080/browse/RC-4
Jira transition ID : 10100

WorkItems

Identification

Summary :

Status

Status :

Erase All | Search | Result Count : 1000

Issue Key	Summary	Jira URI
RC-3	Deployment Request	http://10.0.0.7:8080/browse/RC-3
RC-4	Deployment Request	http://10.0.0.7:8080/browse/RC-4

Result Count : 3 Page : 1/1

Figure 143: The workitem editor

Reference

For more information about accessing links to JIRA issues from execution instances, refer toViewing associated workitems on page 255 (import processes) andViewing associated workitems on page 281 (deployment processes).

drops

Appendices

Entity types

Entity types are used throughout the deployment process and some affect how DROPS operations are carried out. Types are categories for components, environments and infrastructure items. The different types of each entity are defined by an admin in the **Preferences** menu under **Artifacts** and **Configuration**.

Entity types
Component types below
Environment types on the next page
Infrastructure item types on page 410

Table 35: Entity types: Configuration

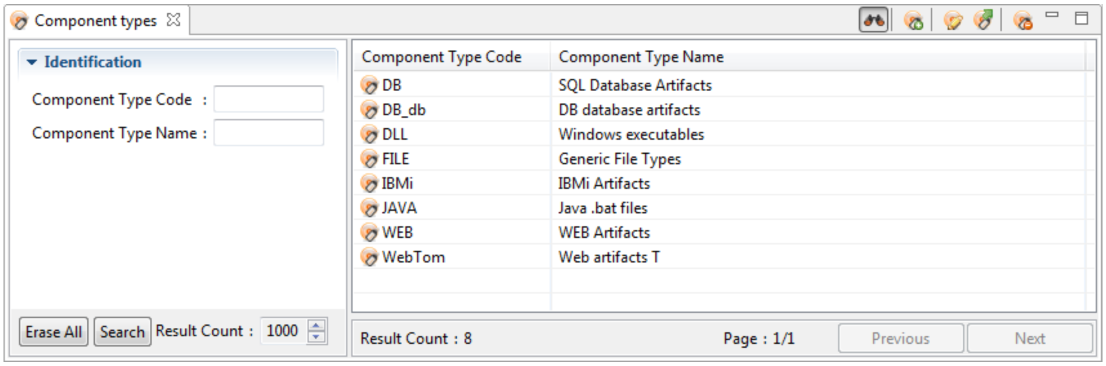
Component types

Component types are used to categorize components in order to easily see what kind of artifacts they contain. Components can contain any type of artifact and any combination of different artifacts. Usually, they contain a group of artifacts that together create an individual part of a larger application/software/website/etc. Categorizing them creates a way to identify these individual parts of the whole application.

> ***Example***
> Component types might be web archives, executables, SQL database artifacts, IBM i save files, etc.

Component types are selected when creating and/or editing a component but selecting a component type is not mandatory.

If strategy or activity templates are not defined for the component type, choosing a type does not change how a component functions. However, if an import strategy template or a deployment activity template have been defined for the component type, choosing a type will associate the templates to the component; and changing the type of a component will override any templates previously associated with the component.



Component Type Code :
Component Type Name :
Result Count : 1000

Component Type Code	Component Type Name
DB	SQL Database Artifacts
DB_db	DB database artifacts
DLL	Windows executables
FILE	Generic File Types
IBMi	IBMi Artifacts
JAVA	Java .bat files
WEB	WEB Artifacts
WebTom	Web artifacts T

Result Count : 8 Page : 1/1

Figure 144: The Component Types search view

Reference
For more information about managing components, refer to Components on page 67.

Reference
For more information about managing component types, refer to Component types on page 212.

Environment types

Environment types are used to categorize the nature of your environments and to manage user roles and rights according to how you use the destination site.

There are a number of environment types available by default which enable you to organize different deployment targets by their intended function.

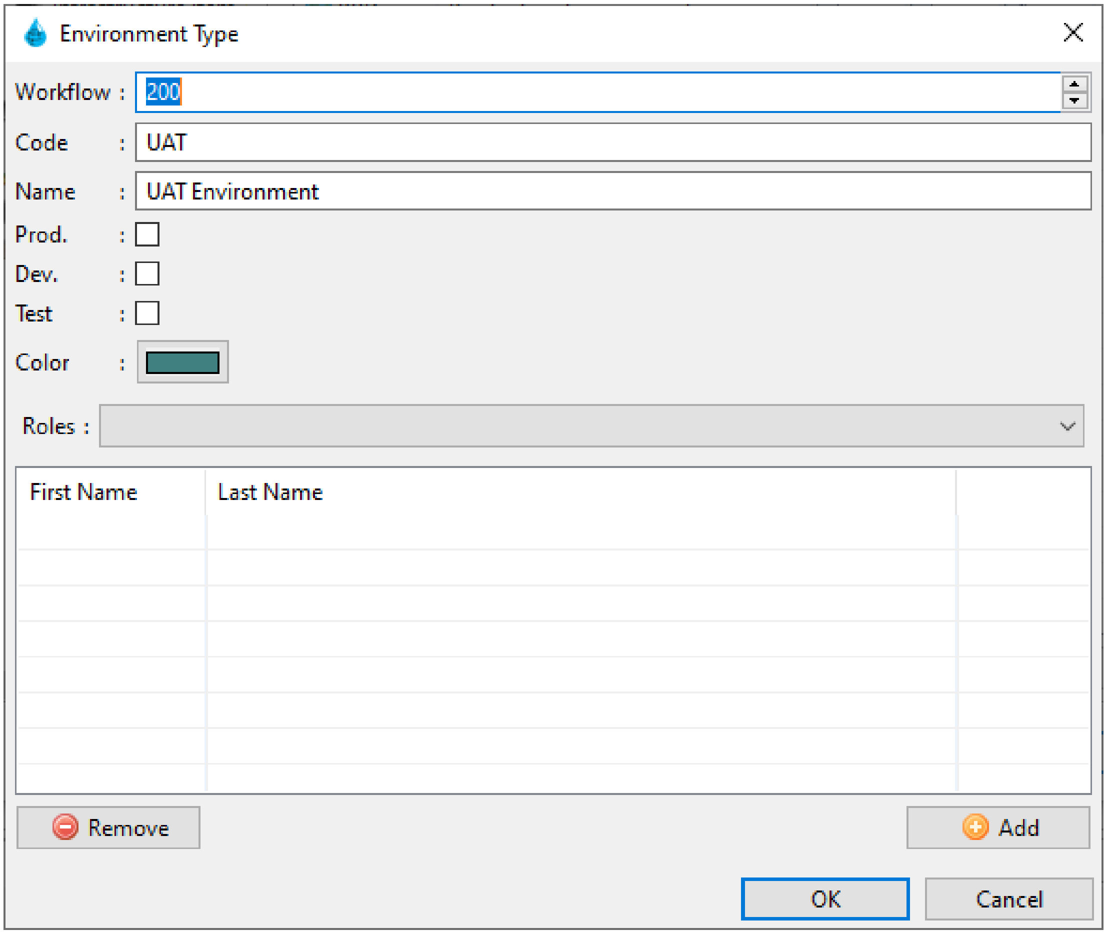
Example
You may need to deploy the same content to a test or training environment as well as a production environment.

An environment type is a set of properties that characterizes the operational need related to the creation of an environment. The properties are:

- the nature of the operational need,
- the workflow order,
- the production flag that indicates if the operational need occurs on an in-production environment.

Choosing an environment type may change a user's access authorization. Global roles can be applied to environments based on their type which means that a user may have the correct role to access/edit/deploy to testing environments but not production environments.

Environment types are mandatory and are assigned to environments when creating and/or editing them.



Workflow : 200
Code : UAT
Name : UAT Environment
Prod. : □
Dev. : □
Test : □
Color : 🟩
Roles :

First Name	Last Name

Figure 145: Create environment types

The development (**Dev**), test (**Test**) and production (**Prod**) check boxes mark the environment type according to how it is used.

The workflow order defines the order in which applications should be deployed to environments. The workflow considers the type with the lowest workflow number the first environment type to deliver to and the highest number the last.

The **Color** option makes it possible to associate a color to any **Environment Type** to help differentiate them. Click on the **Color** field to display the Color Picker box, select a color then click **OK**.

The Roles associated with the management of environment types are defined here.

> **Important!**
> Environment types should not be confused with environment extensions which may require extended configuration.

> **Note**
> Any numbers can be used to define the workflow, except decimals. However, using single-digit numbers may limit use if you have more than 10 types of environments. It is suggested to use three-digit numbers so that you can

insert new types between existing types later.

Example

A deployment plan contains multiple environments so that the developing application can be tested on a test environment, then delivered to a training environment and finally to the final production environment. Testing environment types should be first in the workflow (100), training types second (200) and production types last (300). Later, you could add a UAT testing environment type between testing and training by giving it a workflow number of 150.

Reference

For more information about managing environments, refer to Environments on page 96.

For more information about environment extensions, refer to Environment type extended configuration on page 458.

Infrastructure item types

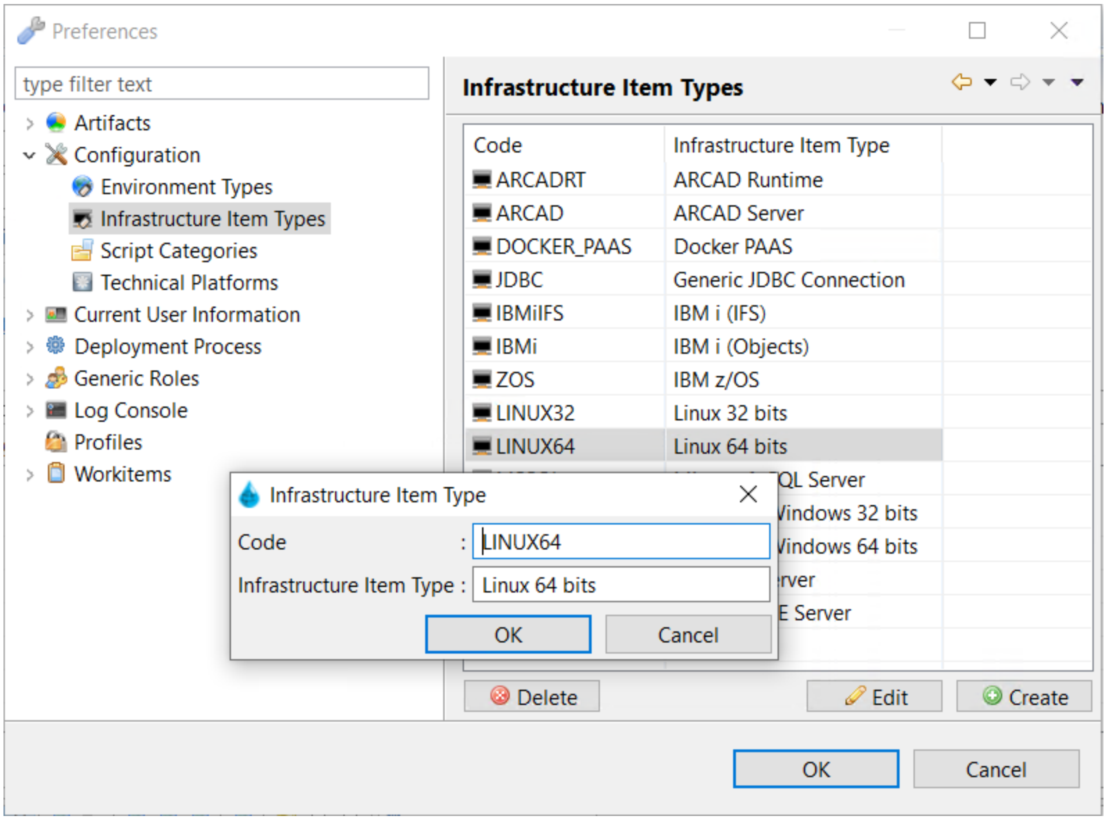
Infrastructure item types are used to categorize the services found on target systems and to manage their specific extended configuration. You can search for infrastructure items based on type. Choosing an infrastructure item type is very important because the associated extended configuration options provide connection details on a type-to-type-basis. Only default types will contain extended configuration but you can create new types of your own to categorize your infrastructure items however you need to.

Example

If an infrastructure item is intended to create the link between DROPS and a TOMCAT, IBM i, ARCAD server, a JDBC or Docker, there is necessarily more information to configure in order for the connection to be authorized than there is for a "normal" type destination (ex: Windows). These types are provided by default in the DROPS Studio.

The extended configuration for an infrastructure item is managed in the **Infrastructure Item** editor after the type is selected.

Infrastructure item types are mandatory and selected when creating and/or editing an infrastructure item.



Code	Infrastructure Item Type
ARCADRT	ARCAD Runtime
ARCAD	ARCAD Server
DOCKER_PAAS	Docker PAAS
JDBC	Generic JDBC Connection
IBMiIFS	IBM i (IFS)
IBMi	IBM i (Objects)
ZOS	IBM z/OS
LINUX32	Linux 32 bits
LINUX64	Linux 64 bits
	QL Server
	Vindows 32 bits
	Vindows 64 bits
	rver
	E Server

Infrastructure Item Type

Code : LINUX64

Infrastructure Item Type : Linux 64 bits

Figure 146: Create infrastructure item types

Reference

For more information about managing infrastructure items, refer to Infrastructure items on page 108.

Properties

Properties are pairs of keys and values that together create variables which can be used when defining actions inside deployment processes. If you enter a property's key in a field when defining an action, the value defined in the same property will be used. Properties refer to specific entity-related content and can only be referenced if the entity in which they are defined is used in the current deployment process. They are intended to help reuse details concerning your entities.

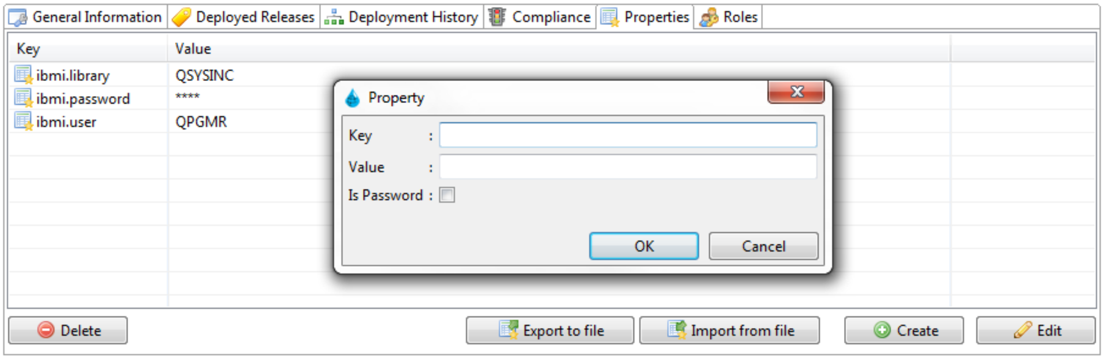
User-defined properties are available in the following DROPS entities:

- applications
- import strategies
- releases
- environments
- infrastructure items
- deployment processes

> ***Example***
> Set a password property in an infrastructure item in order to use a variable when asked for a login instead of entering it manually.

> ***Reference***
> For more information about how to refer to defined properties, refer to Substitution variables in the appendix.

To access and manage the properties defined in an entity, select the **Properties** tab in the entity's editor.



Key	Value
ibmi.library	QSYSINC
ibmi.password	****
ibmi.user	QPGMR

Property

Key :

Value :

Is Password : □

Figure 147: The Properties manager

To create a new property manually, click the **Create** button. Both the **Key** and **Value** fields are mandatory.

If the property is a password, check the **Is Password** checkbox. When the property is saved, the **Value** will be hidden and display as asterisks (*).

To create multiple properties automatically for most entities, click the **Import from File** button to and select a predefined *.properties* file to upload. The properties to import must follow this syntax: key = value.

Note

The .properties file can contain more information than the values to import but only the content in the key = value syntax will be imported into the list of properties.

The imported properties are only available for the current entity. To reuse imported properties in other entities, import the file again in the second entity's editor.

To export the properties defined for any entity as a .properties file, click the **Export to file** button. All of the properties defined in the tab are exported.

To edit an existing property, double-click it, or select it, then click the **Edit** button.

To delete a property, select it, then click the **Delete** button.

Warning!

Deleted properties are no longer associated with any variable and cannot be accessed or recovered.

Server properties

Properties are pairs of keys and values that together create variables which can be used during the DROPS import processes executions. If you enter a property's key in a field when defining an item, the value defined in the same property will be used.

The server properties are properties defined at the DROPS Server level. They are not linked with a sub-entity of the server.

Use server properties to customize the import strategy on a dedicated server and to change the execution behavior when the related process is executed on another server. It is particularly useful when DROPS servers are organized in a main / node management context.

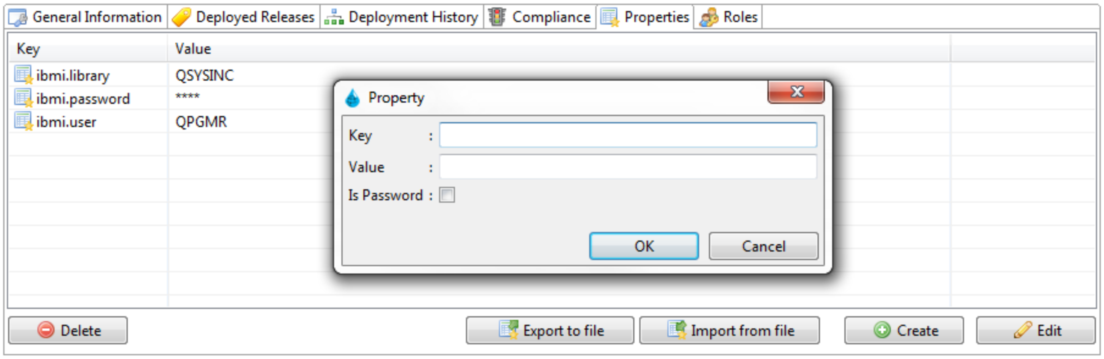
To reference a server property, use the following syntax:

```
${serverProperty.<server_property_key>}
```

Where *<server_property_key>* is the value of the selected server property.

> ***Reference***
>
> For more information about how to refer to defined properties, refer to Substitution variables in the appendix.

To access and manage the server properties, open the **Server Properties** view.



Key	Value
ibmi.library	QSYSINC
ibmi.password	****
ibmi.user	QPGMR

Key :
Value :
Is Password : □

Figure 148: The Properties manager

To create a new property manually, click the **Create** button. Both the **Key** and **Value** fields are mandatory.

If the value of the property must be hidden, check the **Encrypted** checkbox. When the property is saved, the **Value** will be hidden and displayed as asterisks (*).

To create multiple properties automatically for most entities, click the **Import from File** button and select a predefined *.properties* file to upload. The properties to import must follow this syntax: key = value.

> **Note**
>
> The *.properties* file can contain more information than the values to import but only the content in the key = value syntax will be imported into the list of properties.

To export the server properties as a *.properties* file, click the **Export to file** button. All of the properties defined are exported. If you want a property to be encrypted, the value of the property in the *.properties* file must start with the prefix *CRYPTO_DROPS:*.

To edit an existing property, double-click it, or select it, then click the **Edit** button.

> **Note**
> The **Encrypted** attribute cannot be edited. If you want to change this value, you must delete and recreate the property.

To delete a property, select it, then click the **Delete** button.

> ***Warning!***
> Deleted properties are no longer associated with any variable and cannot be accessed or recovered.

Substitution variables

Substitution variables are used to reference parameters throughout DROPS by referring to various entities' properties. The properties can either be the *fields* used to define the entity's details, or the *properties* defined manually in each entity.

> ***Reference***
>
> For more information about defining properties in entities, refer to Properties.

These values can either be entered manually or come from property values defined in different entities linked to the entity. Variables enable you to change the value of parameters retroactively and consistently without changing the overall definition of your entities and guarantee that the deployment process will work correctly.

To declare a substitution variable, use one of the following syntaxes:

- Point to a specific field or property when the entity does not need to be defined.
  ${field-name} or ${[property-key]}

  > ***Example***
  >
  > ${user}, ${password}, ${application.code}, ${local.root} are replaced by the value defined in the corresponding field.

- Point to attribute information defined in an entity.
  ${[entity-name].[attribute]}

  > ***Example***
  >
  > ${importStrategy.localroot} is replaced by the path entered in the import strategy's Root Directory field.

- Point to a user-defined property in a specific entity.
  ${[entity-name].props.[property-key]}

  > ***Example***
  >
  > ${application.props.myProperty} is replaced by the value of the property called *myProperty* declared in the application.

Substitution Variables
Import process substitution variables on the facing page
Deployment process substitution variables on page 420
Activity-level substitution variables on page 423
Artifact-specific substitution variables on page 425

Table 36: Substitution variables

Import process substitution variables

The variables available to use during the import process represent entities that always remains the same, even if they are used in multiple import process instances. The variables available for import processes are all the attributes and properties defined for:

- applications
- import strategies
- components

Variable	Description
${import.version}	This variable is replaced by the value of the External Version defined by the user when launching an import process instance. It is a special substitution variable that does not rely on an entity and is used to point to the version number of an external artifact provider's release. This value is used to define which of the SCM's versions the artifacts should come from when they are imported. The version to import is defined in the External Version field when importing. ***Reference*** For more information about defining the external version to import, refer to Importing artifacts immediately by application on page 245 or Creating import process instances on page 250.
${release.number}	The Release Number of the imported release. **Important!** The release number can be called by many variables. This one is only for the import process.
${application.code}	The App. Code.
${application.text}	The App. Name.
${application.packagetype}	The Packaging Type.
${application.importfile}	The Default Imported Version file name.
${application.releasetemplate}	The Release Number Template.
${application.qualifier}	The Qualifier.
${application.majoritem} ${application.minoritem} ${application.builditem} ${application.revisionitem}	The application's Current Major, Minor, Build and Revisions Items fields. The biggest number of the most recently imported application.

Table 37: Import process substitution variables

Variable	Description
${component.code}	The Component Code.
${component.name}	The Component Name.
${importStrategy.code}	The Import Strategy Code.
${importStrategy.comment}	The Import Strategy Name.
${importStrategy.localroot}	The value defined in the Root Directory field. ⚠ **Important!** The ${importStrategy.localroot} and ${local.root} variables are not the same! If you use the value **AUTO** as the Local Root: • ${importStrategy.localroot} is a static variable. It will be replaced with AUTO. • ${local.root} is a dynamic variable. It will be replaced with the *actual* root directory (i.e. the one that DROPS generates and uses if you enter ***AUTO**)
${importStrategy.usetemplate}	Equals 1 if the Strategy uses a template, 0 if it does not.
${importStrategy.isinitialization}	Equals 1 if the Strategy is an Initialization Strategy, 0 if it is not
${importStrategy.order}	The order of execution. Values start at 0. Import strategies are executed in the order in which they are defined. If a Strategy was the third to be created, it will be executed after the first two Strategies and the variables will be 3.

Table 37: Import process substitution variables

Export process substitution variables

These variables available to use while executing a deployment process based on a release set are used to generate notification messages. You can access the list of releases that belong to this release set, use this information in a notification message and share it with other users that do not have access to the DROPS Studio.

During the execution of a process based on a Release Set and for each release that belongs to the selected release set, some additional properties are generated.

Variable	Description
${releaseset.loop.begin}	This variable identifies the beginning of a text block that contains references to a particular release set.
${releaseset.count}	This value contained the count of releases contained in a particular release set.
${releaseset.number}	This value corresponds to the number of the release.
${releaseset.<release_attribute_i>}	This value is substituted by the value of the attribute generated by the release.
${releaseset.loop.end}	This variable identifies the end of a text block.

Table 38: Export process substitution variables

The DROPS processor captures the text between the ${releaseset.loop.begin} and ${releaseset.loop.end} properties to create the notification.

Deployment process substitution variables

The variables available to use during the deployment process represent entities that always remains the same, even if they are used in multiple deployment process instances. The variables available for deployment processes are all the attributes and properties defined for:

- applications
- releases
- environments and their possible extended configuration
- deployment process

Variable	Description
${drops.home}	The value of the variable called "DROPS_HOME".
${process.code}	The Process Code field.
${process.name}	The Process Name field.
${process.lastrelease.id}	Releases can be tracked chronologically. This refers to the latest deployed release's Release Description field.
${process.lastrelease.order}	Releases can be tracked chronologically. This refers to the latest deployed release.
${process.lastrelease.number}	Releases can be tracked chronologically. This refers to the latest deployed Release Number. ⚠ **Important!** The release number can be called by many variables. This one is only for the latest deployed release.
${application.code}	The App. Code.
${application.text}	The App. Name.
${application.packagetype}	The Packaging Type.
${application.importfile}	The Default Imported Version file name.
${application.releasetemplate}	The Release Number Template.
${application.qualifier}	The Qualifier.
${application.majoritem} ${application.minoritem} ${application.builditem} ${application.revisionitem}	The application's Current Major, Minor, Build and Revisions Items fields. The biggest number of the most recently imported application.

Table 39: Deployment process substitution variables

Variable	Description
${release.number}	The Release Number. ⚠ **Important!** The release number can be called by many variables. This one is only for the deployment process.
${release.text}	The release's Release Description field.
${release.creationdate}	The release's uneditable **Opened the** field.
${release.closuredate}	The release's uneditable **Closed the** field.
${release.isclosed}	Points to closed releases.
${release.islocked}	Points to locked releases.
${release.order}	Releases can be tracked chronologically. This refers to the Release Description field of the last release created.
${release.comment}	The content defined in the **Comments** tab of the **Release** editor.
${environment.code}	The *Env. Code.
${environment.name}	The *Env. Name.
${environment.config.application}	The Application Code defined in the environment’s extension.
${environment.config.identifier}	The Environment ID defined in the environment’s extension.
${deployment.executer}	The user who executed the deployment process instance.
${deployment.instancetype}	The deployment mode of the deployment process instance (Full, Transfer-Only, installation-only). ***Reference*** For more information about modes, refer toDefining an instance's deployment mode on page 270.
${deployment.preparedate}	The value defined in the **Preparation Date** column of the deployment process instance.
${deployment.preparer}	The user who created the deployment process instance.
${deployment.result}	The value defined in the **Execution Result** column of the deployment process instance.
${deployment.rollback}	This substitution variable makes it possible to differentiate if the process is a rollback process or a standard deployment process. The value of this variable is **true** if it is a rollback process, **false** if it is a deployment process.

Table 39: Deployment process substitution variables

Variable	Description
${deployment.scheduledate}	The date the schedule was created for a deployment process instance.
${deployment.scheduled}	The date a deployment process instance is scheduled to execute.
${deployment.schedulehour}	The hour the deployment process instance is scheduled to execute.
${deployment.scheduleminute}	The minute the deployment process instance is scheduled to execute.
${deployment.startdate}	The date the deployment process instance executed.
${deployment.status}	The value defined in the **Execution Status** column of the deployment process instance.
${deployment.transactionid}	The deployment process instance's Transaction ID.
${deployment.deploymentorder}	The order number of the deployment plan instance created when the plan is executed, multiplied by 100. ***Example*** If the instance is the 4th to be executed in the plan (following three instances created from the same plan) the value is 400.

Table 39: Deployment process substitution variables

Activity-level substitution variables

An activity-level substitution variable is a combination of multiple values found in the entities that are reunited in the activity (the infrastructure item and the component). The variables available at the activity-level are:

- all the attributes defined for
  - The artifact List
  - components
  - infrastructure items
- all the properties defined for infrastructure items.
- the optional extended configuration of the infrastructure item.

To declare an activity-level substitution variable, use one of the following syntaxes:

- Point to an attribute of the activity.
  ${activity.[activityID].[attribute]}

  Example
  ${activity.tomcat.identifier}

- Point to a related entity.
  ${activity.[activityID].[entity.name].[attribute]}

  Example
  ${activity.tomcat.infra.url}

- Point to a property defined for a related entity.
  ${activity.[activityID].[entity.name].props.[property_name]}

  Example
  ${activity.tomcat.infra.props.connectionurl}

- Point to an extended configuration attribute.
  ${activity.[activityID].config.[attribute]}

  Example
  ${activity.tomcat.config.user}

Example

Artifact Key: **/database/scripts/install/scripts.sql**

1. ${drop.artifact.key} = **/database/scripts/install/scripts.sql**
2. ${drop.file.name} = **scripts.sql**
3. ${drop.file.path} = **/database/scripts/install/**

Note
The slash will be replaced by the File Separator defined for the given infrastructure item.

Variable	Description
${activity.[activityID].artifact.list}	The path to the artifact list xml file located on the DROPS Server.
${activity.[activityID].remote.artifact.list}	The path to the artifact list xml file located on the remote infrastructure item.
${activity.[activityID].component.code}	The Component Code.
${activity.[activityID].component.name}	The Component Name.
${activity.[activityID].component.uri}	The Directory Path where components/artifacts are imported.
activity. [activityID].component.externalversion	The External Version for a component and its release.
${activity.[activityID].identifier}	The defined [activity's] *Identifier.
${activity.[activityID].infra.backupdir}	The path defined in the Backup Directory field.
${activity.[activityID].infra.rootdir}	The path defined in the Root Directory field.
${activity.[activityID].infra.fileseparator}	The path defined in the item's File Separator field.
${activity.[activityID].infra.url}	The infrastructure item's URL which is made up of the Protocol, Host name & Port and (sometimes) the Host name & Port.
${activity.[activityID].infra.code}	The item's *Infra. Item Code.
${activity.[activityID].infra.name}	The item's *Infra. Item Name.
${activity.[activityID].infra.login}	The login name used to connect to the DROPS Agent. The item's Login & Password field.
${activity.[activityID].infra.password}	The password used to connect to the DROPS Agent. The item's Login & Password field.
${activity.[activityID].infra.store}	Points to the directory into which the related component's artifact (s) will be transferred.

Table 40: Activity-level substitution variables

Artifact-specific substitution variables

There are no variables defined for individual artifacts however, if you execute an action specific to an artifact, or multiple artifacts in the list, the following variables are available.

⚠ **Important!**

If any of the following variables are used, you *must* tick the Apply script to the list of artifacts checkbox in order to ensure that the variable is taken into account at the artifact-level. If the script is not applied to each individual artifact in the list, these variables will not be applied because they are specific to individual artifacts.

Variable	Description
${drop.file.name}	The artifact's file name.
${drop.file.path}	The relative path (without the file name) of the artifact.
${drop.artifact.key}	The artifact Key.
${drop.artifact.source.file}	The absolute name of the artifact in the artifact repository.
${drop.artifact.target.file}	The absolute name of the artifact in the infrastructure item storage area.
${drop.file.basename}	The base name (the final name without extension) of the artifact.
${drop.file.extension}	The file extension (without the leading ".").

Table 41: Action-level substitution variables

Roles

Roles are only used during process execution (importing artifacts and deploying). They are intended to manage a user's authorization to execute certain procedures during an import or deployment process or assign a task. users with roles can be kept informed via notifications during process execution in order to be able to make decisions correctly.

Having a certain role does not automatically mean that a user is required to do something during process execution because multiple users can be assigned the same role. However in order to carry out certain actions, like executing a deployment process, a user must be assigned the role that authorizes him to do so.

The DROPS entities involved in process execution and therefore those to which roles must be assigned are:

- applications
- releases
- clients (if multi-client mode is activated)
- environments, environment groups and environment types
- infrastructure items
- deployment plans

Each individual instance of an entity can have different users assigned to it depending on the needs of your organization.

> ***Example***
>
> Two different environments may be managed by two different people because deploying to a production environment might only be done by a manager, while deploying to a test environment could be done by a developer.

However, roles can also be defined globally. Assigning a role to *all* instances of an entity instead of just one or several enables a user to play the same role for every entity.

> ***Example***
>
> A manager with a team of 3 people may be granted global access to all applications in order to oversee all the individual applications his team creates. Each of the three teammates may only need to import artifacts for the application(s) they are in charge of, but not the others. The manager's role is assigned on the global level by an admin using the **Generic Roles** menu in the **Preferences** menu while the team's roles are assigned in the individual application(s) from the **Roles** view.

In terms of authorization, there is no overlap between roles and rights. However, it is important to carefully assign rights that correspond to an entity to a user’s profile so that when he is assigned a role his access is unrestricted.

> ⚠ **Important!**
>
> Roles are used to authorize users to perform actions involved in the import or deployment processes while rights are used to grant authorization to create, access and/or modify the entities involved in the processes.

Example

If a user has the Repository Manager role for an application and has the right to *view* applications but does not have the right to *edit* them, all applications will be read-only. The user can still carry out the processes associated with the application because of their Repository Manager role, such as importing artifacts, but he cannot manage the release number version for the application because his application editing rights are restricted.

The following table lists the entities to which roles are available for each entity and for what a user with that role is responsible. The rights required for each role are also indicated by category.

Role	Authorization(s) Granted	Required Rights (by category)
Application Administrator	Can assign roles in applications.	Applications Validation Requests
Application Repository Manager	Responsible for the content in the artifact repository and the correct configuration of applications, components and import strategies. Can execute the import process.	Applications Releases Import Process Instances Strategy Templates Server Properties Validation Requests
Application Import Operator	Responsible for executing the import process.	Import Process Instances Releases Validation Requests
Application Database Administrator	Responsible for the application's source/target database(s). Can validate the deployment process.	Validation Requests
Application Workflow Manager	Responsible for the correct configuration of the deployment workflow.	Clients Deployment Plans Validation Requests

Table 42: Definitions of user roles and the required rights

Role	Authorization(s) Granted	Required Rights (by category)
Environment Administrator	Can assign roles in environments.	Clients Environments Environment Groups User Management
Environment Release Manager	Responsible for the correct configuration of the elements involved in the deployment process. Can execute and validate the deployment process.	Clients Deployment Plans Deployment Process Diagrams Deployment Process Instances Deployment Process Templates Deployment Schemas Environments Environment Groups Infrastructure Items SSH Key Management Replacement Rules Scripts Validation Requests Node Management
Environment Release Operator	Responsible for executing the deployment processes. Can be selected to validate the deployment process.	Clients Deployment Process Instances Validation Requests Node Management
Environment Group Administrator	Responsible for creating and managing environment groups.	Clients Environments Environment Groups User Management
Environment Type Administrator	Responsible for creating and managing environment types.	Environments Environment Groups Deployment Plans
Environment Type Release Manager	Responsible for the correct configuration of the elements involved in the deployment process.	Environments Environment Groups Deployment Plans
Infrastructure Item Administrator	Can assign roles in infrastructure items.	Clients Infrastructure Items User Management
Infrastructure Item Manager	Responsible for the correct configuration of infrastructure items.	Clients Infrastructure Items Compliance Rules Compliance Rule Sets Instantiation Templates Provisioning Templates

Table 42: Definitions of user roles and the required rights

Role	Authorization(s) Granted	Required Rights (by category)
Infrastructure Item Contributor	In charge of solving problems concerning infrastructure items. Is notified via notification email when an issue arises. Responsible for plan of action concerning suspended actions and/or interruption messages.	Clients Suspended Actions Instantiation Instances Provisioning Instances
Node Administrator	Responsible for main / node configuration and administration.	Main Settings Management Main / Node Mode Node Management
Plan Group Release Manager	Responsible for the correct configuration of deployment plans.	Deployment Plans
Release Administrator	Can assign roles in releases.	Releases User Management
Release Project Manager	Responsible for the correct configuration of releases. Can validate the deployment process.	Applications Releases Validation Requests

Table 42: Definitions of user roles and the required rights

Reference

For more information about assigning roles at the entity-level, refer to Assigning entity-level roles on page 54.

For more information about assigning roles at the global-level, refer to the *DROPS Configuration Guide*.

For more information about creating and managing profiles and rights, refer to the *DROPS Configuration Guide*.

For more information about creating DROPS users, refer to User accounts on page 44.

Rights

Rights grant users the possibility to view, create and modify various DROPS entities. If the proper access rights are not granted to a profile, access to the corresponding entity (ex: applications) or action (ex: editing applications) is restricted.

Rights are granted to users via profiles. A profile is packed with rights that determine which items a user should be able to access throughout DROPS and if they can edit them or not. Profiles are assigned to individual users. When combined with roles, users have complete or restricted access to all of the processes included in the DROPS Studio.

The DROPS Studio has one default profile that contains all of the rights available. This **All Rights** profile is automatically assigned to the default administrator so that he can manage new profiles for all future users.

There is no limit on how many profiles can be created. Depending on the organization of your team, the definition of more specific profiles, corresponding exactly to the attributions of each user, is essential for a well-secured system. It may be helpful to create profiles that correspond to the different roles throughout DROPS to ensure that users are assigned profiles that contain all the rights necessary for their role(s).

In terms of authorization, there is no overlap between roles and rights. However, it is important to carefully assign rights that correspond to an entity to a user's profile so that when he is assigned a role his access is unrestricted.

⚠ **Important!**

Roles are used to authorize users to perform actions involved in the import or deployment processes while rights are used to grant authorization to create, access and/or modify the entities involved in the processes.

Example

If a user has the Repository Manager role for an application and has the right to *view* applications but does not have the right to *edit* them, all applications will be read-only. The user can still carry out the processes associated with the application because of their Repository Manager role, such as importing artifacts, but he cannot manage the release number version for the application because his application editing rights are restricted.

The following table lists the rights that do not require a paired role but are necessary for server administrators or other users that may not be involved in the configuration and execution of the deployment process.

Task	Required Rights (by category)
Creating and editing metadata	Administrative Rights
Managing DROPS modules	Module Updater
Installing DROPS modules remotely	Remote Installation
Accessing server preferences and configuration	Server Administration
Managing server configuration	Server General Administrative Rights

Table 43: Server Administrator and other high-profile rights

Reference

For more information about the other rights available in the DROPS Studio, refer to Table 42: Definitions of user roles and the required rights which outlines the relationship between roles and rights for each entity in the DROPS Studio.

Component type extended configuration

Certain component types require extended configuration to further define connection information.

If the type of component extension chosen requires extended configuration, click the **Edit configuration...** link in the **Identification** section in the editor to open the corresponding **Extended Configuration** editor.

> **Note**
> Like any property defined in an entity, the fields in any extended configuration editor can be referenced by Substitution variables to reuse the details in the deployment process diagram.

Component extension types
ARCAD component extended configuration

Table 44: Component extension types

ARCAD component extended configuration

To connect to an ARCAD application, select the **ARCAD Component Properties** extension type and enter the following extended configuration information.

Field	Description
ARCAD Server Config.	The source ARCAD Server from which to import the application.
Application Code	The application to import.

Table 45: ARCAD component extended configuration

Import strategy type configuration

All of the import strategies follow a standard process for populating the artifact repository after collecting the artifacts to import. The artifacts are copied to the Root Directory, then DROPS scans the content and creates an identical directory structure inside the Artifact Repository. When the duplicate directory structure is finished, DROPS copies the artifacts into the correct locations in the repository. Because the directory structure is redefined in the repository, when the artifacts are deployed, they will be deployed following the same structure.

Configuring an import strategy is mandatory in order to collect the artifacts to import. To quickly communicate with the most commonly used artifact providers and eventually incorporate their specific delivery types, select the corresponding type of strategy from the drop-down list in the **Import Strategy** editor.

Example

Configure an import strategy to retrieve artifacts based on a SCM's "available" versions by choosing the strategy type for that SCM.

Note

If a template was *used* to create a new import strategy, the strategy type configuration described below cannot be modified and the entire section is grayed out. To modify this content, either change the template (which will carry over to all of the strategies using it) or create a new strategy that is not linked to the template.

The following properties/substitution variables can be called into the extended configuration of an import strategy or a import strategy template:

- any user-defined property in an application
- the ${activity.[activityID].component.code} substitution variable
- any user-defined property in an import strategy

Example

Using the ${importStrategy.props.env} variable in a template enables you to create an "env" property in the import strategy that *uses* the template which will be called when the extended configuration is taken into account.

Reference

For more information about using properties, refer to Properties and Substitution variables in the appendix.

The different types of import strategies require different configuration. This section describes each type and how to configure each type.

Import strategy types
The ARCAD import strategy on page 435

Table 46: Import strategy types: Configuration

Import strategy types
The ARCAD Enhanced import strategy on page 438
The Azure DevOps Piplines Builds strategy on page 440
The Command Line strategy on page 441
The CVS import strategy on page 443
The Docker Simple import strategy on page 445
The File System strategy on page 446
The GIT import strategy on page 448
The IBM i import strategy on page 450
The Maven import strategy on page 451
The SFTP import strategy on page 452
The SVN import strategy on page 453

Table 46: Import strategy types: Configuration

The ARCAD import strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by exporting data from an ARCAD version.

This strategy executes an ARCAD macro-command to prepare a SAVF to import into the artifact repository. This command must be executed on an ARCAD Server that runs on an IBM i platform.

First, the system connects to the ARCAD Server using the connection information defined in the import strategy. Then, the defined macro-command is executed:

1. the version number is resolved,
2. the variables are substituted,
3. the execution of the macro-command is submitted using the defined submission parameters.

The system waits for the end of the submitted job and returns the execution result. If the execution is OK, a stream file is created from the target SAVF file into the temporary IFS directory. This stream file is downloaded into the Root Directory defined by the strategy and the content of the Root Directory is imported as artifacts.

> ⚠ **Important!**
> The macro-command must create the related version artifacts in a SAVF that must be named according to the value defined by the user and be created in the library defined by the user.

> **Note**
> The content of Save Files imported using the IBM i and Arcad Import Strategies are saved as .txt attachments by default. This makes it easy to refer to the content of the Save File outside of IBM i after it has been imported into the DROPS Studio.

Field	Description
ARCAD Server	Enter the name or IP address of the IBM i partition hosting the server.
Login/Password	Enter a valid user login and password. This user must be declared on the server and will be the user that will execute any operations on the IBM i. The password should be encoded in Base64.
Instance Code	Enter the ID of the instance to reach. If you are using ARCAD the instance is **AD** by default.
Secure Connection	Tick this checkbox if the server requires a secured TLS/SSL connection. ***Reference*** For more information about ARCAD secure connections, refer to the *ARCAD SSL Configuration Guide*.
Language	Select the language to use on the server (FRA or ENG).

Table 47: ARCAD strategy extended configuration

Field	Description
CCSID	Enter a valid CCSID to connect to the server.
Application Code	The code of the target ARCAD application.
Env. Identifier	The ID of the environment containing the versions.
Target Version Number	The number of the version from which to get the artifacts. The possible value are: 1. a fixed version number. 2. (*currently unavailable*) ***LASTPRD** the last version that has been promoted into reference. 3. (*currently unavailable*) ***LAST** the last version opened on the defined environment. By default, the value is **${import.version}**. Using this substitution variable enables the strategy to use the external version number set using the Import Artifacts wizard.
Library	The library where the macro-command is located. By default, the library is set to ***LIB**.
Name	The name of the macro-command to execute. The following macro-command parameters are available: - PREPARE APP(${arcad.application}) - EXPORT APP(${arcad.application}) - ENV(${arcad.envid}) - VERSION('${arcad.version}') - SAVLIB(${arcad.savlib}) - SAVFILE(${arcad.savfile}) The following substitution variables are available: - ${arcad.application}: the App code - ${arcad.envid}: the environment Identifier - ${arcad.version}: the Version Number - ${arcad.savlib}: the Save Library - ${arcad.savfile}: the Save File - ${arcad.instance}: the ARCAD instance's code - ${arcad.language}: the ARCAD instance's language code By default, the value is **EXPORT APP(${arcad.application}) ENV(${arcad.envid}) VERSION('${arcad.version}' SAVF(${arcad.savfile} TMPLIB(${arcad.savlib}))**.

Table 47: ARCAD strategy extended configuration

Field	Description
Submission Parameters	The additional parameters that will be used to submit the macro-command execution. These must be valid SBMJOB command's parameters. By default, the value is **JOBQ(DROPS/DROPS) JOB(EXPORT)**.
Temporary Library	The Temporary Library where the SAVF file will be generated. By default, the value is **ARCAD_NET**.
Temporary SAVF	The name of the SAVF file that must be imported into the artifact repository. By default, the value is ***GEN**.
Temporary IFS Directory	The name of the IFS directory where the generated SAVF will be copied before importing it into the repository. By default the value is ***AUTO**.

Table 47: ARCAD strategy extended configuration

Save the changes (💾, Ctrl+S or **File** > **Save**).

The ARCAD Enhanced import strategy

This strategy includes most of the same properties as the generic ARCAD import strategy, however you do not need to define the ARCAD Application property because this value is now defined at the component level in the extended configuration. This strategy should, therefore, be used in ARCAD-type components.

The Target Version Number can now be defined using a version selector to avoid making mistakes. The retrieved version list is based on:

- The ARCAD development server connection defined in the **ARCAD Settings** (Connecting to the ARCAD development server on page 311)
- The ARCAD Application Code defined in the parent component

Field	Description
Login/Password	Enter a valid user login and password. This user must be declared on the server and will be the user that will execute any operations on the IBM i. The password should be encoded in Base64.
Instance Code	Enter the ID of the instance to reach. If you are using ARCAD the instance is **AD** by default.
Secure Connection	Tick this checkbox if the server requires a secured TLS/SSL connection. ***Reference*** For more information about ARCAD secure connections, refer to the *ARCAD SSL Configuration Guide.*
Language	Select the language to use on the server (FRA or ENG).
CCSID	Enter a valid CCSID to connect to the server.
Application Code	The code of the target ARCAD application.
Env. ID	The ID of the environment containing the versions.
Target Version Number	The number of the version from which to get the artifacts. The possible value are: 1. a fixed version number - click the browse button to search for and select the version to import 2. (*currently unavailable*) ***LASTPRD** the last version that has been promoted into reference. 3. (*currently unavailable*) ***LAST** the last version opened on the defined environment. By default, the value is **${import.version}**. Using this substitution variable enables the strategy to use the external version number set using the Import Artifacts wizard.
Library	The library where the macro-command is located. By default, the library is set to ***LIB**.

Table 48: ARCAD Enhanced strategy extended configuration

Field	Description
Name	The name of the macro-command to execute. The following macro-command parameters are available: • PREPARE APP(${arcad.application}) • EXPORT APP(${arcad.application}) • ENV(${arcad.envid}) • VERSION('${arcad.version}') • SAVLIB(${arcad.savlib}) • SAVFILE(${arcad.savfile}) The following substitution variables are available: • ${arcad.application}: the App code • ${arcad.envid}: the environment Identifier • ${arcad.version}: the Version Number • ${arcad.savlib}: the Save Library • ${arcad.savfile}: the Save File • ${arcad.instance}: the ARCAD instance’s code • ${arcad.language}: the ARCAD instance’s language code By default, the value is **EXPORT APP(${arcad.application}) ENV(${arcad.envid}) VERSION('${arcad.version}' SAVF(${arcad.savfile} TMPLIB(${arcad.savlib}))**.
Submission Parameters	The additional parameters that will be used to submit the macro-command execution. These must be valid SBMJOB command's parameters. By default, the value is **JOBQ(DROPS/DROPS) JOB(EXPORT)**.
Temporary Library	The Temporary Library where the SAVF file will be generated. By default, the value is **ARCAD_NET**.
Temporary SAVF	The name of the SAVF file that must be imported into the artifact repository. By default, the value is ***GEN**.
Temporary IFS Directory	The name of the IFS directory where the generated SAVF will be copied before importing it into the repository. By default the value is ***AUTO**.

Table 48: ARCAD Enhanced strategy extended configuration

Save the changes (💾, Ctrl+S or **File** > **Save**).

The Azure DevOps Piplines Builds strategy

This strategy populates the artifact repository using artifacts collected from an Azure Pipeline Builds.

The Build ID field is a filter that matches build numbers. It is allowed to use ***** as a prefix search (ex: "#202102*"). The import strategy will import artifacts from all runs that match the build ID filter.

Field	Description
Organization	The Azure Devops Organization where the Azure Devops Project is stored.
Project	The project which hosts the artifacts to imports.
Build ID	The a pattern used to select builds ID from pipelines. By default, a build ID name starts with a # and is a run of the pipeline process.
User & **Password**	The user and password to connect to the Azure DevOps Services.

Table 49: Azure DevOps strategy extended configuration

The following properties/substitution variables can be called into the extended configuration of an import strategy or a import strategy template:

- any user-defined property in an application
- the ${activity.[activityID].component.code} substitution variable
- any user-defined property in an import strategy

> ***Example***
> Using the ${importStrategy.props.env} variable in a template enables you to create an "env" property in the import strategy that *uses* the template which will be called when the extended configuration is taken into account.

> ***Reference***
> For more information about using properties, refer to Properties and Substitution variables in the appendix.

To define a property, click the **Create** button in the **Extended Properties**. Define the **Key** and **Value** of the property in the dialog. To edit an existing property, double-click it, or select it, then click the **Edit** button.

To delete a property, select it, then click the **Delete** button.

> ***Warning!***
> Deleted properties are no longer associated with the import strategy and cannot be accessed, recovered or called by a substitution variable.

Save the changes (, Ctrl+S or **File** > **Save**).

The Command Line strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by executing a user-defined command.

This strategy scans defined directories and retrieves their contents according to the command defined.

To define the command to execute, configure the name and parameters of the executable command and the connection information required to access a 3rd party artifact provider via the command line.

Field	Description
User	The user name that can be referenced as a ${user} variable in the command if a connection is needed.
Password	The password that can be referenced as a ${password} variable in the command if a connection is needed.
Executable	The name of the file to execute.
Options Template	The blank-separated list of parameters to pass to the executable. The template can include substitution variables. ***Reference*** For more information about variables refer to Substitution variables in the appendix.

Table 50: Command Line strategy extended configuration

The following properties/substitution variables can be called into the extended configuration of an import strategy or a import strategy template:

- any user-defined property in an application
- the ${activity.[activityID].component.code} substitution variable
- any user-defined property in an import strategy

> ***Example***
>
> Using the ${importStrategy.props.env} variable in a template enables you to create an "env" property in the import strategy that *uses* the template which will be called when the extended configuration is taken into account.

> ***Reference***
>
> For more information about using properties, refer to Properties and Substitution variables in the appendix.

To define a property, click the **Create** button in the **Extended Properties**. Define the **Key** and **Value** of the property in the dialog. To edit an existing property, double-click it, or select it, then click the **Edit** button.

To delete a property, select it, then click the **Delete** button.

Warning!
Deleted properties are no longer associated with the import strategy and cannot be accessed, recovered or called by a substitution variable.

Save the changes (🖫, Ctrl+S or **File** > **Save**).

The CVS import strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by exporting data from a CVS repository.

Important!

To correctly invoke the CVS command line processor, the DROPS Server must know its location. This location is defined by a DROPS administrator in the **CVS strategy Settings** section of the **DROPS - Import Process** category of the DROPS Server settings. The path set there will be used for all of the CVS strategies created for the current server, so the URL that points to your artifacts must correspond to it.

Reference

For more information about setting the path to the CVS executable, refer to the *DROPS Configuration Guide*.

Define connection properties specific to this strategy and the external CVS tool in the **CVS Repository Connection Properties** section. These properties are used to pass connection information to the CVS command line when the import process begins.

Field	Description
Protocol	The protocol used to connect to the CVS server.
CVS Server	The CVS server.
Port	The CVS server's port.
User	The user to connect to the CVS repository.
Password	The user password used to connect to the CVS repository.
Repository	The path to the repository to import.
Tag	The tag number used to filter the export.
Module	The module to import.

Table 51: CVS Repository import strategy extended configuration

Note

The parameters can use the properties defined in the parent import strategy by referencing them using substitution variables.

Reference

For more information about this process, refer to Substitution variables in the appendix.

The following properties/substitution variables can be called into the extended configuration of an import strategy or a import strategy template:

- any user-defined property in an application
- the ${activity.[activityID].component.code} substitution variable
- any user-defined property in an import strategy

> ***Example***
>
> Using the ${importStrategy.props.env} variable in a template enables you to create an "env" property in the import strategy that *uses* the template which will be called when the extended configuration is taken into account.

> ***Reference***
>
> For more information about using properties, refer to Properties and Substitution variables in the appendix.

To define a property, click the **Create** button in the **Extended Properties**. Define the **Key** and **Value** of the property in the dialog. To edit an existing property, double-click it, or select it, then click the **Edit** button.

To delete a property, select it, then click the **Delete** button.

> ***Warning!***
>
> Deleted properties are no longer associated with the import strategy and cannot be accessed, recovered or called by a substitution variable.

Save the changes (, Ctrl+S or **File** > **Save**).

The Docker Simple import strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by exporting a Docker Image from a reference repository in Docker.

The Docker daemon DROPS uses to manage the image(s) to import from Docker must carry out a transfer before DROPS can access the image(s) to import. First, the daemon duplicates the “source” image in the development repository then transfers the copy to the reference repository. DROPS can then import the transferred “target” image from the reference repository.

Field	Description
Docker Daemon Host	The connection path required to connect to the Docker development repository (the daemon).
Use TLS	Tick this checkbox if the connection is secure.
Certificates Path	If the connection is secure, define the path to the certificates required to access the secure Docker daemon.
Reference Repository Host	The path to reference repository where the target image to import is stored.
User/Password	The login and password required to access the reference repository.
Registry	Define the sub-folder in which the target image is stored, if required.
Source Image Name	The name of the source image as it appears in the development repository.
Target Image Name	The name of the duplicated copy of the image in the reference repository. You can change the original source name of images this way.
Target Tag	The name of the target image tag.

Table 52: Docker strategy extended configuration

The File System strategy

This strategy populates the artifact repository using content from one or several folders found on the same machine as the DROPS Server. It scans the directory(ies) and retrieves content according to the filters defined.

In the dialog to create or edit a **File System strategy**, the root path to the directory in which your artifacts are stored is defined. To filter the defined path and tell DROPS to import only certain artifacts instead of everything found in the root path, define a regular expression that filters which files to import.

There are two ways to filter by the last date artifacts were modified:

1. define a range of dates: all the files that have a *last modification date* in this range will be imported.
2. define a date limit: all the files that have a *last modification date* before this date will be imported. This date limit is computed by subtracting time from the execution date. To define this date, enter the amount of time to subtract and also the time (day, minute or hour).

Field	Description
Directory	The root path of the directory to scan. **Important!** This directory must be a sub-directory of the Root Directory defined in the parent import strategy.
Filter (regExp)	The regular expression to filter the artifacts in the **Directory**. By default this value is **.***.
Use a Date Set	Select this option to use a *last modification date filter* based on a date range.
Start Date	[*Optional*] The lower limit of the date range. Any artifacts modified after this date/time will be imported.
End Date	[*Optional*] The upper limit of the date range. Any artifacts modified before this date/time will be imported.
Use Execution Date	Select this option to use a *last modification date filter* based on a date limit.
Amount to subtract	The amount of time to subtract from the date/time of the import. Any artifacts modified between when the import is launched and the time difference defined here will be imported. Set the number in this field and use the **Part of impact** field to set the value.
Part to impact	The unit of time to use to subtract the value defined in the **Amount to subtract** field from the date/time the import is launched. ***Example*** If you set the **Amount** to **20** and the **Part** to **Minute**, any artifacts modified in the twenty minutes prior to launching the import will be imported.

Table 53: File System strategy extended configuration

To create a File System strategy filter, click the ⊕ **Create** button in the **File System strategy** manager and define the directory to scan and the filters to apply.

To edit an existing filter, double-click it, or select it, then click the ✎ **Edit** button.

To delete a filter, select it, then click the ⊗ **Delete** button.

> ❗ ***Warning!***
> Deleted filters are no longer associated with the import strategy and cannot be accessed or recovered.

Save the changes (💾, Ctrl+S or **File** > **Save**).

The GIT import strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by exporting data from a Git repository.

This strategy executes the GIT checkout command to reach a GIT repository.

> **Important!**
>
> To correctly invoke the GIT command line processor, the DROPS Server must know its location. This location is defined by a DROPS administrator in the **Git strategy Settings** section of the **DROPS - Import Process** category of the DROPS Server settings. The path set there will be used for all of the GIT strategies created for the current server, so the URL that points to your artifacts must correspond to it.

> ***Reference***
>
> For more information about setting the path to the Git executable, refer to the *DROPS Configuration Guide*.

Artifacts are exported from a GIT repository when DROPS executes the following command. When the necessary **connection information** is defined, DROPS uses the information to create and send the fully-defined command to the GIT.

```
#Local Git Repository Preparation
git clone https://username:password@git-repository-url target-export-path
git --git-dir="target-export-path.git"
  --work-tree="target-export-path"
  fetch
  --tag

#Modification retrieval for a given tag
git --git-dir="target-export-path.git"
  --work-tree="target-export-path"
  checkout
  tags/particular-tag
```

> **Note**
>
> The *username*, *password*, *git-repository-URL* and *particular-tag* parameters (in **bold** in the example above) are either defined in the **GIT Repository Connection Properties** section or can be replaced by properties defined either in the parent import strategy or the **Extended Properties** section by referencing them using substitution variables.
>
> The *target-export-path* is assigned by the Root Directory.

> ***Reference***
>
> For more information about this process, refer to Substitution variables in the appendix.

Define connection properties specific to this strategy and the external GIT tool in the **GIT Repository Connection Properties** section. These properties are used to pass connection information to the GIT command line when the import process begins.

Field	Description	Substitution variables
Repository URL	The GIT repository URL. For example: https://github.com/...	N/A
User	The user to connect to the GIT repository.	${git.user}
Password	The user password used to connect to the GIT repository.	${git.password}
Tag	The **tag** number used to filter the export.	${git.tag}

Table 54: GIT import strategy extended configuration

The following properties/substitution variables can be called into the extended configuration of an import strategy or a import strategy template:

- any user-defined property in an application
- the ${activity.[activityID].component.code} substitution variable
- any user-defined property in an import strategy

Example

Using the ${importStrategy.props.env} variable in a template enables you to create an "env" property in the import strategy that *uses* the template which will be called when the extended configuration is taken into account.

Reference

For more information about using properties, refer to Properties and Substitution variables in the appendix.

To define a property, click the **Create** button in the **Extended Properties**. Define the **Key** and **Value** of the property in the dialog. To edit an existing property, double-click it, or select it, then click the **Edit** button.

To delete a property, select it, then click the **Delete** button.

Warning!

Deleted properties are no longer associated with the import strategy and cannot be accessed, recovered or called by a substitution variable.

Save the changes (, Ctrl+S or **File** > **Save**).

The IBM i import strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by exporting data from an IBM i repository.

This strategy is intended to retrieve and import existing SAVF files from an IBM i that match the provided **Save File Name Pattern** defined in the **List of Save Files** section. Regex can be entered in this section.

DROPS connects to the target IBM i using the provided system, login and password. It will eventually set the iASP for the connection if one is defined (and different from *SYSBAS).

DROPS will first transfer each SAVF file found to the temporary IFS directory defined (using the CPYTOSTMF IBM i command), and then copy them to the artifact repository.

> **Note**
> The content of Save Files imported using the IBM i and Arcad Import Strategies are saved as .txt attachments by default. This makes it easy to refer to the content of the Save File outside of IBM i after it has been imported into the DROPS Studio.

Field	Description
IFS Temporary Directory	The directory where DROPS will put the files listed in the **List of Save Files** section.
iASP	The independent auxiliary storage pool.
IBM i System, Login & Password	The connection information for the target IBM i.
Library & Command	An IBM i command to run before DROPS locates the files listed in the **List of Save Files** section.

Table 55: IBM i strategy extended configuration

To define a SAVF file, click the **Add** button in the **List of Save Files** manager.

To edit an existing SAVF file, double-click it, or select it, then click the **Edit** button.

To delete a SAVF file from the list, select it, then click the **Remove** button.

> ***Warning!***
> Deleted file patterns are no longer associated with the import strategy and cannot be accessed or recovered.

Save the changes (, Ctrl+S or **File** > **Save**).

The Maven import strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by exporting data from a unique Maven repository.

No maven client is required.

Define connection properties specific to this strategy and the external Maven server in the **Repository Information** section. All maven repositories respect the same path syntax (Maven Coordinates) which must be defined in the **Artifact Information** section.

Field	Description
Repository URL	The depot root repository URL.
Repository user	The user to connect to the Maven repository, if required.
Repository password	The user password used to connect to the Maven repository, if required.
Group ID	The Maven group ID(s), separated by periods.
Artifact ID	The Maven artifact ID(s), separated by periods.
Version	The version of the artifact(s) to import.
Extension	If there are other types included in the version that have different extensions, enter them here.
Classifier	[*Optional*] Enter any complementary artifacts to deliver, separated by periods.

Table 56: Maven import strategy extended configuration

The SFTP import strategy

This import strategy based on SFTP allows you to retrieve artifacts form a remote host, using SFTP.

It populates the artifact repository by using the content set in the **Root Directory** field, by downloading data from a remote machine reachable via SSH. It scans all the remote directories, including their subdirectories, and retrieves their content according to the defined filters. This import strategy preserves the directory structure and references these files as artifacts.

In the dialog to create or edit an SFTP import strategy, set the **Directory** field with the root path to the directory in which your artifacts are stored. To filter the defined path and tell DROPS to import only certain artifacts instead of everything found in the root path, define a regular expression that filters which files to import in the **Filter (regEx)** field.

Parameter	Description
OS Family	Indicates the OS Type of the target host. This information defines the path separator.
Host	Indicates the IP Address or DNS of the target machine.
Port	Sets the SSH port.
User	Sets the username on the remote host to which you are connecting.
SSH Key	Sets the DROPS SSH Key that is used by the SSH communication.
Remote base directory	Sets the remote parent directory of the artifacts that will be downloaded.

Table 57: SFTP import strategy extended configuration

> **Note**
>
> The **Host**, **Remote Base Directory** and **User** parameters can be set with substitution variables that are substituted at the beginning of the execution.

The SVN import strategy

This strategy populates the artifact repository using the content of the Root Directory after filling it by exporting data from an SVN repository then following the standard process.

This strategy executes the SVN export command to reach an SVN repository.

> **Important!**
>
> To correctly invoke the SVN command line processor, the DROPS Server must know its location. This location is defined by a DROPS administrator in the **SVN Strategy Settings** section of the **DROPS - Import Process** category of the DROPS Server settings. The path set there will be used for all of the SVN strategies created for the current server, so the URL that points to your artifacts *must* correspond to it.

> ***Reference***
>
> For more information about setting the path to the SVN executable, refer to the *DROPS Configuration Guide*.

Artifacts are exported from an SVN repository when DROPS executes the following command. When the necessary **connection information** is defined, DROPS uses the information to create and send the fully-defined command to the SVN.

```
svn export [-r revision]
      [--ignore-externals]
      [--depth ARG]
      [--force]
      [--ignore-keywords]
      [--native-eol ARG]
      [--quiet]
      svn-repository-url
      target-export-path
      --username username
      --password password
      --non-interactive
```

Define connection properties specific to this strategy and the external SVN tool in the **Connection Information** section. These properties are used to pass connection information to the SVN command line when the import process begins.

> **Note**
>
> The parameters to define (in bold in the example above) can use the properties defined in the parent import strategy or the **Extended Properties** section by referencing them using substitution variables.

> ***Reference***
>
> For more information about this process, refer to Substitution variables in the appendix.

Field	Description
Repository URL	The SVN repository URL. For example: *SVN//ip number/repository*
User	The user to connect to the SVN repository.
Password	The user password used to connect to the SVN repository.
Revision	[*Optional*] The SCM's revision to export. This value is used to define which of the SCM's versions the artifacts should come from when they are imported. • Entering a fixed value here will configure the default version that the import strategy will always deploy. It is static. • If this field is left blank, DROPS will always import the most recent version from your SCM. • Enter the special substitution variable ${import.version} in this field to communicate with the **External Version** field accessed when launching an import. You can manually decide which of the SCM's versions to import every time the import strategy is used. ***Reference*** For more information about defining the external version to import, refer to the Section Importing artifacts on page 241.

Table 58: SVN import strategy extended configuration

Define properties specific to this strategy in the **Extended Properties** section. These properties can be used to add options to the connection command.

1. If a property contains a key called **--force** and the value is blank, the option *--force* will be added to the command parameter.
2. If a property contains a key called **--native-eol** and the value is **CRLF**, the option *--native-eol CRLF* will be added to the command parameter.

To define a property, click the ⊕ **Create** button in the **Extended Properties**. Define the **Key** and **Value** of the property in the dialog. To edit an existing property, double-click it, or select it, then click the ✎ **Edit** button.

To delete a property, select it, then click the ⊗ **Delete** button.

> ***Warning!***
> Deleted properties are no longer associated with the import strategy and cannot be accessed, recovered or called by a substitution variable.

Save the changes (💾, Ctrl+S or **File** > **Save**).

Infrastructure item type extended configuration

Certain infrastructure item types require extended configuration to further define connection information.

If the type of infrastructure item chosen requires extended configuration, click the **Edit configuration...** link in the **Identification** section to open the corresponding **Extended Configuration** editor.

> **Note**
> Like any property defined in an entity, the fields in any extended configuration editor can be referenced by Substitution variables to reuse the details in the deployment process diagram.

Infrastructure item types
ARCAD Runtime extended configuration below
ARCAD Server extended configuration on the next page
Docker extended configuration on the next page
IBM i extended configuration on the next page
Generic JDBC Connection extended configuration on page 457
Tomcat extended configuration on page 457

Table 59: Infrastructure item types: Configuration

ARCAD Runtime extended configuration

To deploy to an ARCAD runtime, select the **ARCAD Runtime** infrastructure item type and enter the following extended configuration information.

Field	Description
Target ARCAD Server Config.	Select an existing ARCAD Server configuration. This list is prepopulated with the pre-defined target servers (refer to Defining connections to target ARCAD-Servers on page 313).
Runtime Deployment Target	Select the linked remote application template. This list is prepopulated with the defined templates. If an infrastructure item was generated automatically, and an ARCAD remote application template was created at the same time, that application template will automatically be linked to the item. ***Reference*** Managing ARCAD remote templates on page 319

Table 60: ARCAD Runtime infrastructure item extended configuration

ARCAD Server extended configuration

To connect to an ARCAD server, select the **ARCAD Server** infrastructure item type and enter the following extended configuration information.

Field	Description
ARCAD Instance	Enter the ID of the instance to reach. If you are using ARCAD the instance is **AD** by default.
Language	Select the language to use on the server (FRA or ENG).
CCSID	Enter a valid CCSID to connect to the server.

Table 61: ARCAD Server infrastructure item extended configuration

Docker extended configuration

To connect to a DOCKER daemon, select the **Docker PAAS** infrastructure item type and enter the following extended configuration information.

Field	Description
Docker Daemon Host	The connection path required to connect to the Docker development repository (the daemon) to which to deploy the image(s).
Use TLS	Tick this checkbox if the connection is secure.
Certificates Path	If the connection is secure, define the path to the certificates required to access the secure Docker daemon.

Table 62: Docker infrastructure item extended configuration

IBM i extended configuration

To connect to an IBM i server, select the **IBM i** infrastructure item type and enter the following extended configuration information.

Field	Description
IFS Temporary Directory	The preliminary location to which to copy the SAVf artifacts. When this file is copied, DROPS uses the Copy From Stream File command to restore it in the target IBM i as an object.
Restoration Target Library	The directory in which to save the restored SAVf object(s).
Login & Password	The connection information for the target IBM i.

Table 63: IBM i infrastructure item extended configuration

Field	Description
The jobs library list	The libraries to access via the online job.
Mapping rules	The rules to distribute the deployed objects. Define which libraries should receive certain objects.

Table 63: IBM i infrastructure item extended configuration

Generic JDBC Connection extended configuration

To connect to a JDBC server, select the **Generic JDBC Connection** infrastructure item type and enter the following extended configuration information.

Field	Description
JDBC Driver	The unique target JDBC driver.
Classpath	The unique target JDBC classpath.
URL	The target address and port in URL format.
User & Password	The connection information for the target JDBC.

Table 64: JDBC infrastructure item extended configuration

Tomcat extended configuration

To connect to a Tomcat server, select the **Tomcat Server** infrastructure item type and enter the following extended configuration information.

Field	Description
Tomcat Home Directory	The root path to the target tomcat.
Application Directory	The path to the tomcat's webapps directory.
URL	The target address and port in URL format.
Login & Password	The connection information for the target Tomcat.

Table 65: Tomcat infrastructure item extended configuration

Environment type extended configuration

Certain environment types require extended configuration to further define connection information.

If the type of environment extension chosen requires extended configuration, click the **Edit configuration...** link in the **Identification** section in the editor to open the corresponding **Extended Configuration** editor.

> **Note**
> Like any property defined in an entity, the fields in any extended configuration editor can be referenced by Substitution variables to reuse the details in the deployment process diagram.

Environment extension types
ARCAD environment extended configuration

Table 66: Environment extension types

ARCAD environment extended configuration

To connect to an ARCAD application and environment, select the **ARCAD environment** extension type and enter the following extended configuration information. These values can be used as a variables in the deployment diagram to ensure that the correct target ARCAD environment is reached.

Field	Description
Application Code	The ARCAD application to which to connect.
Environment ID	The ARCAD environment to which to connect.

Table 67: ARCAD environment extended configuration

Rollback

In order for DROPS to carry out the rollback process, certain systems and/or databases must conform to the following administrative prerequisites.

SQL Server

SQL Server rollback has been developed with SQL Server 2014 Express Edition version 12.0. The database to roll back must be in FULL recovery model. To perform the rollback, DROPS connects to the SQL Server database with the same user login as the application login to roll back.

Features not supported

- Data types: hierarchyid, spatial types, text, ntext, image, varchar(max), nvarchar(max), varbinary(max), sql_variant, table, xml, cursor, udt.
- Row > 8060 bytes
- Rollback of sequences
- Clustered table
- Sparse columns
- Partitioned tables

Oracle

Oracle rollback has been developed in version 11g Release 2, more precisely 11.2.0.4. It runs both on Standard and Enterprise versions.

When rolling back LOB data, the data must be accessible in UNDO, limiting the elapsed time the rollback can be performed after the end of the client SQL batches.

Configuration

- The database must be in ARCHIVELOG
- Minimal supplemental logging must be enabled.
- Archive Log Files must be accessible when DROPS executes the rollback.
- When rolling back LOB data, UNDO must contain data to be rolled back.

Rights

To perform the rollback, DROPS connects to the Oracle database with a user login which must have the following rights:

- GRANT CREATE SESSION
- GRANT CREATE TYPE
- GRANT FLASHBACK ANY TABLE
- GRANT SELECT_CATALOG_ROLE
- GRANT EXECUTE_CATALOG_ROLE
- GRANT EXECUTE ON DBMS_LOGMNR
- GRANT EXECUTE ON DBMS_AUDIT_MGMT
- GRANT EXECUTE ON DBMS_UTILITY

- GRANT EXECUTE ON DBMS_LOCK
- GRANT EXECUTE ON DBMS_SESSION
- GRANT SELECT ANY TRANSACTION
- EXECUTE ON DBMS_FLASHBACK
- SELECT ON V_$LOGFILE
- SELECT ON V_$LOGMNR_CONTENTS
- SELECT ON DBA_AUDIT_TRAIL
- SELECT ON DBA_AUDIT_OBJECT
- SELECT ON DBA_OBJECTS
- SELECT ON DBA_LOBS
- SELECT ON DBA_CONSTRAINTS
- SELECT ON DBA_STMT_AUDIT_OPTS
- SELECT ON DBA_OBJ_AUDIT_OPTS
- SELECT ON V_$SESSION
- SELECT ON DBA_TAB_COLUMNS
- SELECT ON DBA_TABLES
- SELECT ON DBA_CONS_COLUMNS
- SELECT ON DBA_SEQUENCES
- SELECT ON DBA_DEPENDENCIES
- ALTER ANY TRIGGER
- SELECT ANY TABLE

Features not supported

- DDL operation is not supported, except TRUNCATE when the table is empty at the start of the capture.
- IOT tables
- BFILE datatype
- Simple and nested abstract datatypes (ADTs)
- Collections (nested tables and VARRAYs)
- Object refs

The Command Line Interface (CLI)

The DROPS Command Line Interface is used to prepare and execute Import and Deployment Processes. Use the CLI to:

- prepare and execute an import Process Instance,
- prepare and execute a Deployment Process Instance.

(missing or bad snippet)
To call any action using the CLI, the following syntax is used: drops *<subcommand> <parameters>*.

Each subcommand requires one or more parameters. To display the list of supported parameters for any subcommand, run the subcommand without any parameters. If nothing is provided, the system prompts you to fill in the blanks required to accomplish the subcommands tasks.

Parameters between [brackets] are optional. Each parameter has a long and a short form (example: --project, -p) and are separated from their value by the equal sign = or a space.

- executeImport below
- executeDeployment on page 463
- processAsCode on page 464

Common parameters

The following parameters are common to all CLI commands.

`-H, --help`

Display the help message for the command.

`-S, --server`

[Required] The URL of the DROPS Server to which to connect.

`-U, --user`

[Required] The user used to connect to the DROPS Server.

`-P, --password`

[Required] The password used to connect to the DROPS Server.

`-C, --crypted`

Required if the password is Base64 encrypted.

`-L, --log-file`

The path to the file where the logs will be written.

`-V, --verbose`

Enable debug and trace log levels.

executeImport

This command allows the user to prepare and execute an Import Process Instance. With the parameters available in this command, you can manage:

- The release creation and/or update,
- the Component filter definition,
- the execution of a deployment process based on the release created and/or updated,
- The relationship with the Work Items Provider.

Usage: [-CHV] [--create[=<createRelease>]] [-l[=<manageLog>]] [-s[=<resetRelease>]] [-t [=<componentFilterType>]] [-x[=<executeImport>]] [-y[=<executeDeploy>]] -P[=<password>][-P [=<password>]]... --application=<applicationCode>[-d=<releaseDescription>] [-e=<deployEnvironmentCode>] [--filterfile=<componentFilterFileName>][--increment=<increment>] [-k=<workitemKey>] [-L=<logFile>][--outdir=<outputDirectory>] [--outfile=<outputFileName>][-p=<deployProcessCode>] [-r=<releaseNumber>]-S=<serverURL> -U=<user>[w=<workitemProviderKey>][-f=<componentFilterDefinitions>]....

The available parameters for the executeImport subcommand are the following:

`--outfile`

Sets the name of the output file.

`--outdir`

Sets the name of the output directory.

`--application`

Sets the parent application code.

`--create`

Indicates whether the target release has to be created. If a release number is indicated and matches with an existing release, the release creation is skipped.

`--increment`

Indicates the Version part number to increment. If the version number is not indicated or invalid, the value defined at server level is used.

`-d, --description`

Sets a description for the new Release created. If a release has been passed, its description is updated (Only if passed and not blank).

`-r, --release`

Indicates the target release number. The release number is required if the --create option is not set or set to false.

`-s, --reset`

Indicates if the release content has to be reset before the import.

`-t, --filtertype`

Indicates the type of filter used for the component. Set this parameter with one of these three options:

- 0: No filter,
- 1: File,
- 2: User-defined.

`--filterfile`

The absolute name of the file that contains the filter information. This value is required if the option --filtertype equals 1.

`-f, --filter`

Indicates the user-defined filter. This parameter can be defined several times. The definition syntax is <ComponentCode>;<StrategyCode>;<ExternalRelease>.

If no component filters are defined, the entire list is used.

`-y, --deploy`

Indicates if a deployment has to be executed at the end of a successful import.

`-p, --process`

Indicates the code of the process to be executed following a successful import. This process has to be linked to the given application. If a process is not defined, no deployment is executed.

`-e, --proenv`

Sets the code of the target environment.

If no process code is defined, this environment is used with the application code to get the default process defined for the related application and the related environment.

`-w, --wiprovider`

Sets the Work Item provider code used to link the Work Item to the import process.

`-k, --wikey`

Sets the Work Item external key.

`-x, --execute`

Indicates if the prepared Import process must be executed.

`-l, --log`

Indicates if the Import log entries appear in the output results.

executeDeployment

This command allows the user to prepare and execute an Import Process Instance.

With this subcommand, a release is deployed on the target environment using a deployment process. The release is identified by its parent application code and by its release number, the target environment is identified by its code and the deployment process is identified by its code. If no process code is indicated, the processor will try to retrieve the default process defined for the application and the environment.

Usage: [-CHV] [-h[=<manageAttachements>]] [-l[=<manageLogEntries>]] -P[=<password>] [-P [=<password>]]... -a=<applicationCode>-e=<environmentCode> [-L=<logFile>][--outdir=<outputDirectory>][--

outfile=<outputFileName>][-p=<deploymentProcessCode>] -r=<releaseNumber>-S=<serverURL> [-t=<transactionIdPattern>] -U=<user>.

The available parameters for the executeImport subcommand are the following:

`--outfile`

Sets the name of the output file.

`--outdir`

Sets the name of the output directory.

`-a, --application`

Indicates the code of the parent application of the release.

`-r, --release`

Indicates the number of the Release to be deployed. The release is retrieved from its number. This number must be defined for a release declared for the parent application.

`-e, --environment`

Indicates the target Environment Code.

`-p, --process`

Indicates the Deployment process Code. If no value is indicated, the service uses the default process. An error occurs if no process can be found or if no default process is defined.

`-t, --transactionid`

Indicates the transaction ID of the Deployment process. If no transaction ID is indicated, the default transaction ID pattern is retrieved from the server configuration.

`-h, --attachment`

Indicates that the attachments are downloaded in the directory defined by the --outdir value and that the list of attachments appears in the output results.

`-l, --log`

Indicates if the Deployment log entries appear in the output results.

processAsCode

Usage: drops processAsCode [-CHV] [--showInfo[=<showInfo>]] [--showLog[=<showLog>]] -P [=<password>] [-P[=<password>]]... --file=<filename> [-L=<logFile>] [--outdir=<outputDirectory>] [--outfile=<outputFileName>] -S=<serverURL> -U=<user>

`--outfile`

Sets the name of the output file.

`--outdir`

Sets the name of the output directory.

Indicates if the Deployment log entries appear in the output results.

`--file`

The path of a file that contains a Deployment Process Definition using YAML Format.

`--showLog`

Indicates if the log entries must be displayed in the standard output.

`--showInfo`

Indicates if the deployment process ID. must be displayed in the standard output

In that case, a line appears in the standard output using this syntax:

```
16:54:04.641 [INFO] Process ID.: <created/updated process identifier>
```

Automatic Release Resolver

In some cases, the release number generated during the import process must be fixed automatically without any manual input. the automatic release resolver is a script executed at the beginning of the import process to extract the release number value.

Getting started

To use the automatic release resolver:

- Activate the option in the Settings.
- Define a resolver ANT script for the Applications.
- Select a release with a number equals to *AUTO.

The resolver is executed at the beginning of the import process, to retrieve a release number. If a release with the retrieved release number already exists for the parent application, then this release will be used as the target release. If not, ea new release will be created using a number provided by the script and this release will be used as the target release.

In the Configuration Explorer, go to **Artifacts** > **Applications**. Right-click on your application and choose **Edit the Release Resolver Script** in the contextual menu. Define the release resolver script in the ANT script editor.

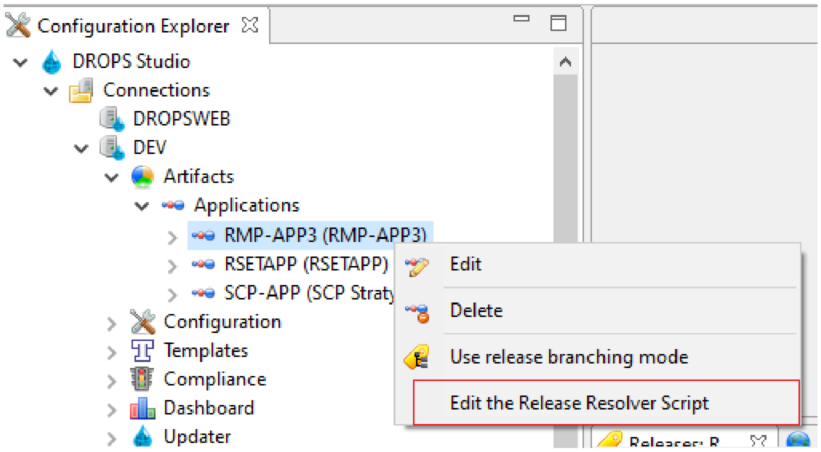
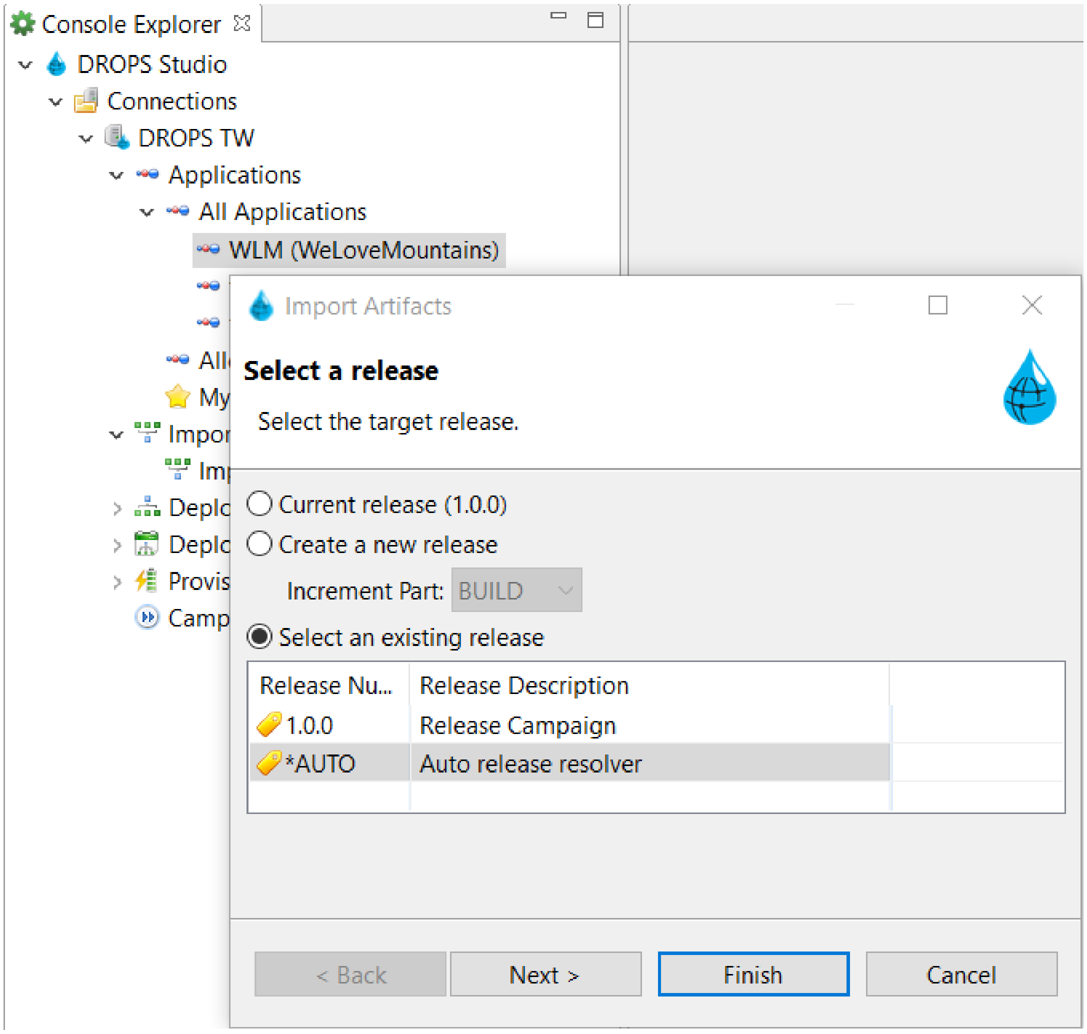


Figure 149: Edit the release resolver script

Create a release with the ***AUTO** release number. This option is only available if the resolver is activated in the settings. When executing the import, select this release to trigger the release resolver mechanism.



Select a release

Select the target release.

○ Current release (1.0.0)

○ Create a new release

Increment Part: BUILD

● Select an existing release

Release Nu...	Release Description
1.0.0	Release Campaign
*AUTO	Auto release resolver

Figure 150: Import with the release resolver

How to write a release resolver script

The release resolver script is an ANT script.

- The ANT Libraries that can be used are located in the directory defined by the value of the **Resolver ANT Libraries Directory** field of the DROPS - Import Process | Release Resolver Settings Section
- The ANT Script base directory is the directory defined by the value of the **Resolver Execution Directory** field of the DROPS - Import Process | Release Resolver Settings Section

The script is executed on the target called *main*. This target must exist in your system.

Assign the generated release number to a the property **resolved.release**.

Reach the parent application values using the **${application.<fieldname>}** ANT properties.

Sample script to set a fixed release number

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="script_template"
         basedir="."
         default="main">

      <target name="main"
            description="The main target">
        <property name="resolved.release" value="2.0.1.3"/>

    </target>
</project>
```

Sample script based on a properties file which location depends on the application code

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="script_template"
         basedir="."
         default="main">

      <target name="main"
            description="The main target">

          <!-- Define a strategy root directory -->
          <property name="importDirectory"
                    value="c:/md/test/strategy"/>

          <!-- Get a property file based on the parent application -->
          <property
file="${importDirectory}/${application.code}/import.properties"/>

          <!-- Use the value of the property called releasenumber located into the
               c:/md/test/strategy/RMP-APP1/import.properties file -->
        <property name="resolved.release" value="${releasenumber}"/>

    </target>
</project>
```

OpenSSH Agentless connection for Windows OS

You can now connect in Agentless mode to Windows machines using OpenSSH. To be able to use SSH to connect to a Windows machine, OpenSSH Server must be enabled.

Enable OpenSSH in Windows OS

To install the OpenSSH components:

Step 1 Open **Settings**, select **Apps**, then select **Optional Features**.

Step 2 Scan the list to see if the OpenSSH is already installed. If not, at the top of the page, select Add a feature, then:

- Find **OpenSSH Clien**t, then select Install
- Find **OpenSSH Server**, then select Install

Step 3 Once setup completes, return to Apps and Optional Features and confirm OpenSSH is listed.

Step 4 Open the **Services** desktop app. (Select Start, type services.msc in the search box, and then select the Service app or press ENTER.)

Step 5 In the details pane, double-click **OpenSSH SSH Server**.

Step 6 On the General tab, from the Startup type drop-down menu, select **Automatic**.

Step 7 To start the service, select **Start**.

Make Powershell the SSH shell by default

Execute the following script in an elevated powershell:

```
New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -
Force
```

Result Once these steps are done, simply configure your infrastructure item by selecting the SSH protocol and specify your username, password or SSH key. If using SSH Key, the public key should be added to the authorized keys on the target Windows machine.

Troubleshooting

I can't find/access the entity I need to create/edit.	I can access an entity, but I can't make any changes.
I can't perform a task because I don't have the correct role.	I can't save the deployment process diagram.
My deployment process is suspended and I don't know why.	My deployment process is suspended but there are no messages in the suspended actions view.
My deployment process is suspended and awaiting approval but there are no messages in the Validation Requests view.	My deployment/import process failed.
My database is full.	The DROPS Server is running but I can’t access it.

Table 68: Troubleshooting

I can't find/access the entity I need to create/edit.

If you see the message "*You are not allowed to use this feature*" when trying to access an entity, verify that your user's profile has been granted the right(s) necessary to access the list of items and/or search for certain items.

Reference

For more information about profiles and rights refer to:

- Accessing your profile and rights on page 56
- The DROPS *Configuration Guide*

I can access an entity, but I can't make any changes.

Verify that your user's profile has been granted the right(s) necessary to edit, delete, create or otherwise modify the entity.

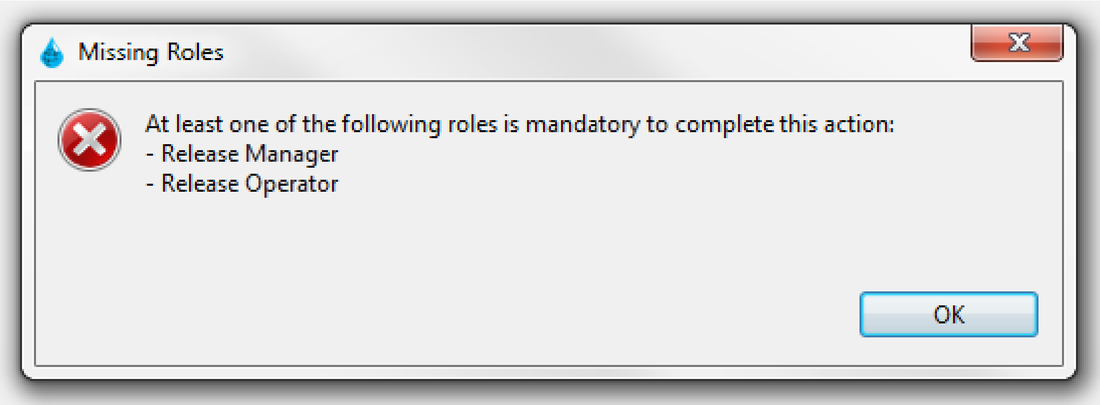
Reference

For more information about profiles and rights refer to:

- Accessing your profile and rights on page 56
- The *DROPS Configuration Guide*

I can't perform a task because I don't have the correct role.

If you see a message similar to the screen shot below, verify that your user has been assigned the role necessary to carry out the task.



At least one of the following roles is mandatory to complete this action:
- Release Manager
- Release Operator

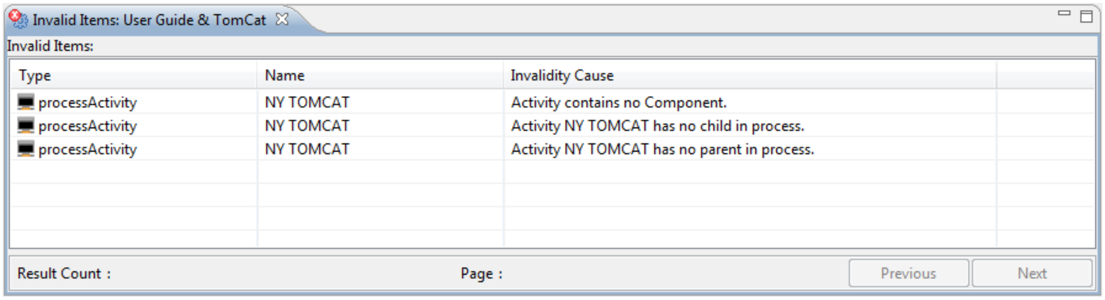
Reference

For more information about roles refer to:

- Roles on page 426
- Assigning entity-level roles on page 54
- The *DROPS Configuration Guide*

I can't save the deployment process diagram.

If there are errors in the deployment process diagram and DROPS cannot save it, the **Invalid Items** view informs you of what the problem is. Any changes will be lost if you close the process without fixing the issues listen and successfully saving.



Invalid Items:

Type	Name	Invalidity Cause
processActivity	NY TOMCAT	Activity contains no Component.
processActivity	NY TOMCAT	Activity NY TOMCAT has no child in process.
processActivity	NY TOMCAT	Activity NY TOMCAT has no parent in process.

Result Count : Page :

Figure 151: The Invalid Items view

In order to save the deployment process diagram, the following actions *must* be taken. If you close the deployment process diagram editor without doing them, all changes will be lost.

- Deployment activities *must* be linked to a component. See Edit deployment activity properties on page 182.
- Deployment activities *must* have both a parent and a child activity. Most commonly the parent will be an **Initialization** activity and the child a **Finalization** activity. See Define the order in which to deploy activities on page 168.
- Deployment activities *must* have an identifier. See Edit deployment activity properties on page 182.

- Deployment actions *must* define a script. See Edit deployment action properties on page 187.
- Deployment actions *must* have an identifier. See Edit deployment action properties on page 187.

My deployment process is suspended and I don't know why.

If a deployment process instance is suspended, check the **Deployment Steps** tab to see why. Often, the deployment is waiting for approval which means that a ✓ validation request is pending.

> ***Reference***
>
> For more information about viewing the deployment steps tab, refer to Understanding deployment process instance results on page 277.
>
> For more information about resolving suspended deployment instances, refer toValidation requests on page 296 or Suspended actions on page 294.

My deployment process is suspended but there are no messages in the suspended actions view.

To view suspended actions and resolve issues, a user must have the Contributor role for the infrastructure item in question.

> ***Reference***
>
> For more information about roles refer to:
>
> - Roles on page 426
> - Assigning entity-level roles on page 54
> - The *DROPS Configuration Guide*

My deployment process is suspended and awaiting approval but there are no messages in the Validation Requests view.

To view and approve/reject ✓ validation requests, a user must have the role defined in the✓ **Validation** tab in the deployment process diagram's properties.

> ***Reference***
>
> Edit global deployment process properties on page 177

My deployment/import process failed.

Check the process' log file to determine at which stage it failed. Common failures are:

- incorrect or missing import strategies (see Introduction to importing artifacts on page 242 and Import strategies on page 72).
- incorrectly configured infrastructure items (see Infrastructure items on page 108).

- incorrectly built deployment process diagram (see Building deployment process diagrams on page 165 and Editing deployment process properties on page 175).

My database is full.

Deployment logs tend to take up a lot of room in the DROPS database.

Limit the size of your deployment logs by filtering out specific unnecessary entries. If your DB still becomes too crowded, archive the deployment logs to create more space.

> ***Reference***
>
> For more information about filtering entries in deployment logs, refer to the *DROPS Configuration Guide*.
>
> For more information about activating the auto-archive option for deployment logs, refer to the *DROPS Configuration Guide*.
>
> For more information about manually archiving logs, refer to View the Logs on page 278.

The DROPS Server is running but I can’t access it.

The DROPS Server has been correctly installed and the service is up and running. However, it is not possible to connect to it from the DROPS Studio or to call the web services.

Symptoms:

- The studio can't connect and says that the server is not responding.
- Accessing the about, shell, or console web service from a web browser fails with a timeout or an "Unauthorized connection" error
- The server's service is actually running and no fatal error shows in the logs or when starting the service from the command line.

The main reason for this is that the port on which the server is listening (5254 by default) is not reachable on the network.

This can be easily diagnosed by trying to access the server locally. Call an netstat-like command on the server and see if it's actually listening on the port. If it's not, proceed with one of the following:

- If a graphical environment and a web browser are available on the machine hosting the server, open the web browser and access the "about" web service using the address *http://###.#.###:port/about*. It should display the server's plugins and their state.
- Use a command line tool to call the "about" web service (like wget, curl...) on *http://###.#.#.#:port/about* and download the result.

If it appears that the DROPS Server is accessible locally, then the network's configuration should be checked as well as the firewall rules, if any.

I can't access the web console

I can't use the operational web console for my deployments, even if I have upgraded my server to the version that supports it.

To use the operational web console, you need to add the following lines to the configuration file *<ServerRoot>/configuration/config.ini*:

```
com.arcadsoftware.csp.style.sources='self' 'unsafe-inline'

com.arcadsoftware.csp.image.sources='self' data:
```

Glossary

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

action

See deployment action.

activation key

A variable-length string value registered in a DROPS Server to enable the creation of a given amount of infrastructure items. It contains the total count of available license units for each license type.

An activation key is encrypted and only valid for one specific DROPS Server.

See license unit.

activity

See deployment activity.

administrator

User: In charge of high-level configuration.

Role: In charge of assigning roles to users at the entity-level.

agent

The tool that the DROPS Server can use to communicate with a target system. Unless you are running the agentless solution, this module must be installed on every target system to which you intend to deploy. The DROPS Server connects to the DROPS Agent on your target machine via HTTP(s).

ant project

A valid set of Ant tasks that makes up the content of a script.

ant task

Individual tasks used in scripts to define the course of action for a deployment action. They produce the same result(s) on every platform they run on.

DROPS provides tasks to perform calls to the DROPS Server to carry out tasks like publishing files from a deployment process, for example, and actions on an IBM i and/or an ARCAD instance.

Prerequisites: the remote runtime must have the **ant-drops.jar** in its classpath.

application

A set of artifacts that creates a specific structure of elements that, ideally, make up and complete the system you deploy.

Applications house artifacts (grouped by components and release) and are required to deploy a given set of functional elements to a target environment.

See artifact and component.

application template

A template feature for applications, containing component types, to be used when creating an application.

Using an application template automatically populates the new application with the types of components defined in the template and their related import strategies, if defined.

See component type and other configuration templates.

archived files storage area

The path to the *root* location where archived deployment logs should be stored. By default, this repository is automatically created in the DROPS Server.

artifact

A unique element of an operational application.

Artifacts are combined in components to create a set of elements to be deployed.

See component.

artifact key

An artifact's relative path starting from its parent component. The relative path is the path defined in the **Default Component URI Template** field during global configuration.

By default the relative path to an artifact is:
file://${root}/app${application}/cmp${component}/rls${release}/...

artifact list

An .xml file created when artifacts are imported which can be referenced in substitution variables to locate artifacts.

artifact repository

A centralized directory that regroups defined applications (components and artifacts) and releases.

Loading the artifact repository with imported artifacts is a process that relies on predefined strategies and ANT tasks. Any type of source provider can be connected to gather artifacts for your components.

See application and release.

B

backup directory

The directory path on the target system used to stock a copy of the content that is currently defined in and active in the deployment process.

C

change package

Specific to SYSchange. A package that contains only a few changed components created in order to update the identically configured libraries of a target system.

compliance rule

A tool that performs tests on infrastructure items to ensure that the target environment complies to specific requirements specific to your setup.

component

A collection of artifacts housed in an application grouped by release.

Components group artifacts based on their function and/or destination. The hierarchy respected in a ready-to-deploy component is dependent on the target release.

component type

Component types are characterization tags used to categorize components in order to easily see what kind of artifacts they contain.

Enhanced component types may contain related templates that will be used in different stages of the DROPS import and deployment processes.

See application template, import strategy template , and process template.

Configuration Management perspective

The eclipse perspective that enables you to manage the artifact and infrastructure repositories.

configuration templates

Most key entities of the DROPS setup process have associated template features. Templates associate properties, configuration settings and related entities which can all be used in unison or separately when creating a new entity.

More information about the different DROPS configuration templates:

- application template
- component type
- import strategy template
- process template

contributor

Role: In charge of solving problems concerning the infrastructure item. User is notified via notification email when an issue arises.

D

database administrator

Role: Responsible for the application's source/target database(s).

delivery stage

An environment type-specific container in a deployment plan, to which environments are added, that enables you to automatically deploy releases to multiple environments. They are necessarily ordered by their type so that the automated deployment starts by delivering artifacts to environments of one type, then to another type, and so on.

deployment action

Specific to deployment processes. Represents one script. Each script defines a task to carry out during the deployment process.

Actions are added to certain activities or stages in the deployment process.

See stage.

deployment activity

Specific to deployment processes. An ensemble of stages that defines the order in which to carry out the deployment process.

The three types of activities in the deployment process diagram are:

1. initialization activity
2. infrastructure item activity
3. finalization activity

deployment activity template

A pre-defined ensemble of stages and/or deployment activities that can be reused in any number of deployment process diagrams.

Activity templates must include at least one stage or one deployment activity. Only infrastructure item activities can be used to create templates.

deployment infrastructure repository

Ensemble of all the entities required to deploy.

deployment plan

While a deployment process delivers one application to one environment, a deployment plan automates the deployment of multiple applications to multiple environments.Deployment plans describe how, to which environments, and in which order releases of given applications are to be deployed to satisfy a defined need. They bring multiple deployment processes into the same schedule.

Like processes, plans cater to an operational need.

deployment process

The ensemble of the deployment process diagram and associated parameters used to define and automate the deployment of one application to one environment.

The process is broken into activities, stages and actions. Actions can be applied to activities and stages.

Step one (initialization activity) = all of the actions defined are carried out.

Step two (infrastructure item activity) is broken into four stages, each of which is carried out in turn:

1. preparation stage
2. transfer stage
3. receiving stage
4. installation stage

Step three (finalization activity) = all of the actions defined are carried out.

See stage, deployment action, deployment activity and deployment process diagram.

deployment process diagram

Specific to deployment processes. An editable visual representation of the activities, stages and actions defined to automate the deployment process.

Use the deployment process diagram to drag and drop the actions to carry out during the deployment process into the activities and stages required for all the infrastructure items for a single environment.

deployment process instance

An execution instance of a deployment process where a particular release has been defined.

They enable users to:

- get feedback on the progression of the execution;
- get logs on the different operations made during the execution;
- access information about what has been deployed;
- validate the different deployment steps;
- interact, if needed, with the process to solve critical situations.

deployment process scheduler

An interactive Gantt chart that displays all of the scheduled deployment process instances in a time line.

The Scheduler can be used to create and modify deployment process instances and their schedules.

deployment process properties template

Pre-defined process-level properties and parameters that can be reused in any number of deployment process diagrams without affecting the diagram itself.

deployment rule

The combination of a RegEx and a specific path on an infrastructure item intended to filter deployed artifacts. Artifacts matching the RegEx are deployed to the rule's destination path instead of the destination defined in the infrastructure item.

deployment schema

A set of deployment rules intended to modify the destination of a filtered group of artifacts on an infrastructure item at the time of deployment.

deployment tag

A tool to mark an ensemble of deployment process instances. You can use tags to quickly search for instances.

deployment XML List Storage Area

When releases are deployed, DROPS creates a deployment list and stores it in a predefined path. By default this path leads to a relative path where the DROPS Server is installed.

DROPS

A licensed software application that brings development and deployment teams together by providing continuous deployment of your company's technology all the way to production. Software, updates, installs, websites and much more can be automatically deployed on a scheduled basis which reduces development time while maintaining team responsibility and respecting your company's security, role distribution and source platform(s).

E

entity

Any item that requires configuration and can be referred to by other items to create a chain of content in order to either import or deploy artifacts.

> ***Example***
>
> application, component, import strategy, release, environment and infrastructure item.

environment

An ensemble of infrastructure items that may represent an easily identifiable location or system that defines where deployed content is intended to go.

Environments are linked to applications and their releases in order to deploy a given set of functional elements to a target.

See application and infrastructure item.

environment group

An ensemble of environments that share one or more applications used to create multiple deployment process instances at the same time.

environment template

Template environments cannot be used to deploy content but may be duplicated to create other active environments that contain the pre-defined elements in the template.

F

finalization activity

The final step in the deployment process. Only one of these types of activities can be defined per deployment process diagram. No artifacts are managed during this step. Only certain actions can be defined in these types of activities and they are always carried out on the DROPS Server. The parent activity must be an infrastructure item activity.

See deployment activity and deployment action.

full package

Specific to SYSchange. A package that contains all of the library members from an environment, including the changed and unchanged components.

G

H

I

import operator

Role: In charge of executing the import process.

import root directory

If an Import Strategy uses the default ***AUTO** as its , the path defined here will be used. By default, this path leads to a relative **tmp** path where the DROPS Server is installed.

import strategy

An engine that communicates with an external artifact provider (source) to find and import the defined artifacts. The imported artifacts are classified according to the directory structure defined in the strategy and can be filtered by version.

import strategy template

A template of an import strategy, containing all the strategy's definitions and parameters, that can either be copied or used as such when creating a component.

See configuration templates.

import XML list storage area

When artifacts are imported, DROPS creates an artifact list and stores it in a predefined path. By default this path leads to a relative path where the DROPS Server is installed.

infrastructure item

A unique entity that defines a Technical Platform target to which content is deployed. It contains connection information and deployment instructions.

Infrastructure items are added to environments and are matched on a one-to-one basis with Components in the deployment process diagram.

See environment and component.

infrastructure item activity

Intermediary steps in the deployment process. There can only be as many infrastructure item activities in a deployment process diagram as the associated environment has infrastructure items. One activity is linked to one infrastructure item. Stages are added to these activities to define the deployment actions to carry out for each item.

> ⚠ **Important!**
>
> These activities must have both a parent and a child activity in order to save the deployment process.
>
> These activities must be linked to at least one component in order to save the deployment process.

See deployment activity, stage, and deployment action.

infrastructure repository

A centralized directory that regroups users, environments, infrastructure items, strategy templates, scripts, deployment schemas, replacement rules, process templates and clients if applicable. The parent menu item for this repository is the **Configuration** node in the **Configuration Explorer**.

Defining the entities included in this directory enables you to configure the entire deployment environment.

Unlike the artifact repository, the deployment Infrastructure Repository is only a menu in the DROPS Studio RCP, not an actual database.

initial import strategy

A type of import strategy that retrieves artifacts to create a release that guarantees that an entire application will be imported into one release. It may be useful to ensure that the first release imported contains every artifact of the application to be able to deploy the full application to a new system.

Contrary to "regular" import strategies, which may repeatably import only the most recent file(s) in a version (partial imports), initial import strategies are usually used once in the beginning to create the complete foundation of your application in the artifact repository.

initialization activity

The first step in the deployment process. Only one of these types of activities can be defined per deployment process diagram. No artifacts are managed during this step. Only certain actions can be defined in these types of activities and they are always carried out on the DROPS Server. The client activity must be an infrastructure item activity.

See deployment activity and deployment action.

installation path

The root directory on the DROPS Agent that contains the local repository storage area.

installation stage

The final phase of the deployment process carried out on the DROPS Agent (the target infrastructure item). The actions defined for this stage are completed after the receiving stage or if a transfer-only deployment instance is performed.

J

Jenkins

Jenkins is a 3rd party application. The DROPS for Jenkins plug-in allows you to pilot DROPS via Jenkins.

JIRA

JIRA is a 3rd party application. The DROPS for JIRA plug-in allows you to pilot DROPS via JIRA.

JMV

Java Virtual Machine

K

L

license unit

Equivalent to one infrastructure item, this is the price paid to use DROPS.

License Units depend on the type of infrastructure item created (the license type). DROPS provides some extensions like DROPS for IBM i that allow you to deploy to special targets (IBM i). The pricing model takes the value of such a feature into account.

To view the number of permanent and temporary License Units available, click the **Configuration** node in the **Preferences** menu (*right-click on the* DROPS Server → Preferences → Configuration).

See also: temporary license unit.

license unit pack

The total number of different types of License Units purchased represented by an activation key that, when registered in the DROPS Server, enables you to use DROPS.

license type

The different kinds of license units available for purchase that enable you to create different types of infrastructure items and therefore use DROPS.

- IBM i
- Open (all the other targets)
- Temporary

local repository storage area

The path on the target DROPS Agent that contains both the Root Directory and the Backup Directory. This path must be accessible to the DROPS Agent and is intended to regroup all of the content deployed to an agent in one location. This path is used when the Backup and/or the Root Directory is set to ***AUTO** in an infrastructure item.

This path may also be called the storage path or the storage root.

M

manager

Role: Responsible for the correct configuration of the infrastructure item(s) concerned.

Maven

Maven is a 3rd party application from which you can import artifacts via a dedicated import strategy.

monitoring List

A list of infrastructure items that is periodically checked by the DROPS Server to confirm that the connection to each corresponding agents is active.

N

O

OS family

The OS on which the DROPS Agent is running. The processes required to remotely install and uninstall a DROPS Agent are defined by this value.

Operational Console Management perspective

The eclipse perspective that enables you to execute the import and deployment processes. The actions available in this perspective are also available in the DROPS web app.

operational need

A particular need that can be satisfied by the execution of a set of processes done by actors that have the required knowledge to correctly understand how and why multiple applications depend on each other.

P

package

Specific to SYSchange. An archive that contains data that comes from multiple libraries. Its internal format is defined by SYSchange.

See promotion package.

packaging

Specific to SYSchange. The process by which all the members in the libraries belonging to a software package are copied into a single file referred to as a SYSchange promotion package.

Packaging of application software with library types of SRC, MAC, PANELS, MSGS, SKELS and LINKLIB involve copying all library members into a physical sequential file. If the software package above was pre-tokenized, optionally only the changed components are stored in the SYSchange promotion package. Thus, only the changed components are packaged and then promoted to update the target system(s).

pattern library list

Specific to SYSchange.and IBM i. A string pattern that will be used by SYSchange to search for the name of the library that will be scanned to detect modifications.

platform

An operating system.

Platforms available are defined by a DROPS admin (high-level configuration) and used in infrastructure items to define the type of OS to which your applications are deployed.

preparation stage

The first phase of the deployment process carried out on the DROPS Server intended to define actions to prepare artifacts for deployment.

profile

A set of rights assigned to a user or user group.

project manager

Role: Responsible for the correct configuration of the release(s) concerned.

promotion

Specific to SYSchange. The process by which a SYSchange promotion package is used to update libraries in the target environment. At the time of promotion, a restore package may be requested.

promotion package

Specific to SYSchange. An archive that contains data that comes from multiple libraries. Its internal format is defined by SYSchange.

See package.

process

See deployment process.

process template

There are five different types of process templates and each one represents a different level and all of the levels it contains.

- A stage template must include at least one deployment action to predefine the steps to carry out when the stage is executed. Templates can be created for every stage in a deployment process:
  - preparation stage
  - receiving stage

    - transfer stage on page 490
    - installation stage
- A deployment activity template is only for infrastructure item activities. Activity templates must include either at least one stage (which must also include at least one action) or at least one action outside of a stage. The same requirements for a normal, unique deployment activities apply to activity templates, including most of the properties.
- A deployment process properties template is an ensemble of pre-defined process-level properties and parameters which can be applied to a deployment process diagram to automatically define this information without using uneditable elements (activity or stage templates) in the diagram itself.

purge

The two methods of permanently removing excess or outdated content from DROPS.

Purging logs, temporary directories and import instances from the DROPS Server is performed in the DROPS Server Configuration menus.

Purging outdated or ineffective releases is performed in the DROPS Studio.

Q

R

receiving stage

The third phase of the deployment process performed on the DROPS Agent (the target infrastructure item). This phase is intended to allow you time to modify transferred artifacts before installing them completely on the infrastructure item. If a deployment process is executed in **transfer-only mode**, the deployment stops after this stage is complete.

release

A consistent set of updated or new artifacts created during the import process and nested under components and applications in the artifact repository.

Releases are the packages that you deploy and can follow your specific version system.

release manager

Role: Responsible for the correct configuration of the environment's deployment process instance and deployment plans.

release number

The unique code applied to releases automatically when they are created.

The template for release numbers is defined in the **Preferences** menu and/or applications.

release operator

Role: In charge of executing deployment processes

release set

A group of releases selected one-by-one to be deployed to the same environment. Sets cannot be stored since they are defined when deployment process instances are created. Every release set instance can have a different combination of releases.

replacement rule

Replacement rules are similar to a *Search & Replace* on configuration files linked to an application. Rules enable you to activate specific commented content in artifacts such as .properties files during deployment. This means that commented content can be selected and rendered active by automatically removing the comment tag. The now-active content will be taken into account during the deployment process.

repository

See artifact repository.

repository manager

Role: In charge of the content in the artifact repository

repository storage area

The path to the *root* location of the Artifact Repository. By default, this repository is automatically created in the DROPS Server.

restore package

Specific to SYSchange. A package created during the promotion (deployment) process which contains a copy of the target environment's original library members prior to being replaced by the data from the promotion package.

To UNDO the effect of a promotion (deployment) operation, the restore package may be used to revert the system to its state just prior to the last promotion (see rollback).

rights

Define a user's access to entities and items and grant authority to create, view and modify them.

Rights are granted to profiles which are assigned to users.

See profile.

role

Defines a user's responsibility concerning an entity involved in the deployment process.

Roles are assigned for applications, releases, clients, environments and infrastructure items. Having a role involving an entity means that the user can and may be required to perform certain actions during the import and/or deployment process.

rollback

Restoring previous content after a deployment is considered 'unsuccessful'.

If a deployment fails or the deployed content is incorrect, reverse the deployment actions by recalling the new content and reinstating the previous content that was saved before the copy.

root directory

The directory path where the deployed artifacts are temporary stored before being officially copied to the target. It is used to temporarily store the artifacts to be deployed before DROPS actually deploys them.

S

SCM

Software Configuration Manager. A.K.A.: VCS

script

A sequence of valid Ant tasks (XML tags) linked to a deployment action that defines what the action should accomplish.

schema

See deployment schema.

step

See deployment activity.

stage

A set of deployment actions in a deployment process that represents a functional part of the parent deployment activity's execution and involves different parts of the system. Each stage runs in a different location - the DROPS Server or the target infrastructure item.

There are four different stages:

1. preparation stage
2. transfer stage
3. receiving stage
4. installation stage

stage template

A pre-defined set of deployment actions in a deployment process that represents a functional part of the parent deployment activity's execution and involves different parts of the system.

Stage templates must include at least one deployment action.

There are three different stage templates:

1. preparation stage
2. receiving stage
3. installation stage

strategy

See import strategy.

T

technical ID

A unique number automatically assigned to each import process instance. Also called the Internal ID.

technical platform

The service to which an infrastructure item deploys releases.

temporary infrastructure item

An infrastructure item that is only active for a fixed amount of time and that consumes one temporary license unit.

Temporary infrastructure items can only be created if every license unit in the current activation key's is consumed. The number of temporary license units available depends on the number of purchased License Units that the current activation key contained. If no activation key has ever been registered for the current DROPS Server, only five (5) temporary infrastructure items can be created and used for a limited period of time (up to one month). This enables you to use DROPS while your activation key is being generated.A temporary infrastructure item becomes definitive as soon as a new license unit is available (as soon as a new license unit pack is registered).

temporary license unit

An automatically-activated License Unit that enable you to create temporary infrastructure items (available for one week). These units are made available before you register your purchased activation key and after all of the purchased License Units in your current Activation Key are consumed.

See license unit.

tokenization

Specific to SYSchange. A Hash Sum value representing the content of a file.

The SYSchange tokenization process is used to determine whether or not a file has changed. When a library is tokenized, each library member along with its token (a 4-byte hex value) is stored in the SYSchange Control file. At a later time, when the same library is re-tokenized, the current member tokens are compared against the ones stored in the Control File and the members with different tokens are determine to have changed. When a file is tokenized any minute change in the file content produces a different token value, i.e. a change has been introduced in the file.

See change package, full package and packaging.

transaction ID

When a deployment process instance is created this identifier is generated which facilitates the rollback process by enabling DROPS to find the actions carried out.

The Transaction ID is not necessarily unique for each instance. If an instance is broken into two parts (transfer-only and installation-only) the two instances will share the same ID in order to maintain the link between the two halves of the deployment process and ensure that if rollback is required, both halves are rolled back.

The ID's syntax follows the template managed in the **Preferences** menu.

transfer stage

The second phase of the deployment process performed behind the scenes which cannot be modified. Artifacts are transferred to infrastructure items during this phase.

U

update repository root path

The path to the *root* location of the new repository in which the server will store files downloaded from the Update Server. By default, this repository is automatically created in the DROPS Server.

user

Password protected accounts that define the access rights for the different members of your team that use DROPS. Each account is associated with one email address and has a unique login.

Users are assigned roles to play in the deployment process and given rights to access certain functions.

user group

A combination of users that are all granted the same rights via profiles and to which roles can be assigned.

Adding users to groups enables you to quickly assign the same roles and rights to multiple users at the same time and always ensure the settings are uniform.

V

validation request

The message sent to a defined user when a deployment process requires his or her approval before carrying out a specific activity or action.

VCS

Version Control System. a.k.a.: SCM

W

workflow order

The predefined order of environment types to which content can/should be deployed. Environment types classify what kind of environment a target system is and according to this classification, determine delivery order.

In DROPS, the workflow order is:

1. Training environment
2. UAT environment
3. Pre-Production environment
4. Pilot environment

5. Production environment

workitem

Workitems are the issues created in the 3rd party tool that pilot DROPS executions.

workitem provider

Workitem providers are the link between DROPS and the 3rd party tool.

X

Y

Z