IKAN ALM User Guide

Release 5.8

January 2019

IKAN Development N.V.
Kardinaal Mercierplein 2
2800 Mechelen
BELGIUM

© 2006 - 2019 IKAN Development N.V.

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, for any purpose, without the express written permission of IKAN Development N.V.

The IKAN Development and IKAN ALM logos and names and all other IKAN product or service names are trademarks of IKAN Development N.V.
All other trademarks are property of their respective owners.

Table of Contents

PART I - GETTING STARTED

Chapter 1 - Introduction 1

1.1. How to Use This Guide 1

Chapter 2 - Logging in and out 3

2.1. Logging in 3

2.2. Logging out 5

2.3. Installing a New License 5

2.4. Error Messages 6

2.5. Warning Messages 7

Chapter 3 - User Interface Overview 8

3.1. Navigating Through IKAN ALM 8

Main Menu 8

Submenus 8

3.2. The IKAN ALM Screen 9

Selected Menu Indication 9

Search Panels 10

Create Panels 10

Overview Panels 10

3.3. Auto Refresh 12

PART II - DESKTOP

Chapter 4 - Desktop 14

Chapter 5 - Personal Settings 15

5.1. Editing Your Personal Settings 15

5.2. Defining Search Filters 16

Creating a search filter *17*

Selecting a Search Filter *18*

Managing Filter Definitions *18*

Modifying a filter's search criteria *20*

Chapter 6 - Approvals 21

6.1. The Outstanding Approvals 22
6.2. Approving Outstanding Approvals 23
6.3. Rejecting Outstanding Approvals 24
6.4. The Approvals Overview Screen 26

Chapter 7 - Level Requests 29

7.1. Level Request Action Flows 29
(Re)Build Level Requests *30*
(Re)Build and Deploy Level Requests *35*
Deploy Level Requests *42*
7.2. Creating Level Requests 45
The Create Level Request: Select Level Screen *45*
Creating a Build Level Request *48*
Creating a Deliver Build Level Request *52*
Creating a Rollback Build Level Request *55*
7.3. Level Requests Overview 58
The Level Requests Overview Screen *58*
The IKAN ALM RSS Functionality *62*
Generating a Report *63*
7.4. Level Request Detail 65
Summary *66*
Status Header 67
Actions Panel 68
Info Panel 70
Builds and Deploys Panel 71
Approvals Panel 72
Issues Panel 73
Error/Warning Log Panel 75
Phase Logs *77*
Level Parameters 80
Level Phases 80
Build Actions 82
Deploy Actions 85
Results *89*
Approvals *91*
Issues *93*
Editing an Issue 94
Deleting an Issue 95
Synchronizing an Issue 95
Adding an Issue 96
Sources *97*
Modifications *98*

Dependencies *99*
Build Dependencies 100
Build Usage 101
Level Request Usage 103
7.5. Build History Screen 104
7.6. Builds and Deploys Overview 107

Chapter 8 - Packages 110
8.1. Creating a Package 111
8.2. The Package Overview Screen 113
8.3. Viewing Package Settings 115
8.4. Editing a Package 116
8.5. Hiding/Showing a Package 117
8.6. Viewing the Package History 117
8.7. Package Details 117
Summary *118*
Actions Panel 119
Info Panel 119
Latest Level Requests Panel 120
View Contents *120*
Edit Contents *121*
Lifecycle Actions *124*

Chapter 9 - Package Build Groups 125
9.1. Creating a Package Build Group 126
9.2. The Package Build Groups Overview Screen 127
9.3. Editing a Package Build Group 129
Adding a Package to a Package Build Group *130*
Removing a Package from a Package Build Group *133*
9.4. Hiding/Showing a Package Build Group 133
9.5. Viewing the Package Build Group History 134

Chapter 10 - Managing the Desktop 135
10.1. The Desktop Overview Screen 136
10.2. Adding a Desktop Tab Page 139
10.3. Editing the Desktop Tab Page Label 139
10.4. Removing a Desktop Tab Page 140
10.5. Adding Elements to a Desktop Tab Page 140
10.6. Rearranging Desktop Elements 143
10.7. Removing Desktop Elements 143

PART III - PROJECT ADMINISTRATION

Chapter 11 - Project Administration 145

11.1. Accessing the Projects Overview 145
11.2. Accessing the Project Administration Options 145

Chapter 12 - Projects 146

12.1. The Projects Overview Screen 146
12.2. Viewing Project Settings 148
12.3. Editing Project Settings 149
12.4. Deleting a Project 153
12.5. Locking/Unlocking a Project 153
12.6. Hiding/Showing a Project 154
12.7. Viewing the Project History 154

Chapter 13 - Project Administration Options 155

Chapter 14 - [Project Name] 157

Chapter 15 - History Log 158

Chapter 16 - Project Streams 161

16.1. Creating a Branch Project Stream 161
16.2. Project Streams Overview 165
The Project Streams Overview Screen 165
Viewing Project Stream Settings 166
Editing Project Stream Settings 167
Deleting a Project Stream 168
Locking/Unlocking a Project Stream 168
Hiding/Showing a Project Stream 168
Viewing the Project Stream History 169
Project Stream Dependencies 169
Adding a Project Stream Dependency 171
Editing a Project Stream Dependency 173
Deleting a Project Stream Dependency 173

Chapter 17 - Lifecycles 174

17.1. Creating a Lifecycle 175
17.2. Lifecycles Overview Screen 176
The Lifecycles Overview Screen 176
Editing a Lifecycle 177
Defined Levels Panel 178
Creating Additional Levels 180

Inserting an Existing Level 180
Changing the order of the Levels 181
Making a Level optional or required 181
Enabling or Disabling Optional Deploys 182
Deleting a Lifecycle *182*
Viewing the Lifecycle History *184*
Editing a Level *184*
Removing a Level *186*

Chapter 18 - Levels 187

18.1. Creating a Build Level 188
18.2. Creating a Test or Production Level 190
18.3. Levels Overview 192
The Levels Overview Screen *193*
Editing a Level *194*
Level Phases 195
Build and Deploy Environment Settings 203
Level Approvals 205
Deleting a Level *206*
Cloning a Level *208*
Viewing the Level History *208*

Chapter 19 - Build Environments 209

19.1. Creating a Build Environment 210
19.2. The Build Environments Overview Screen 212
Editing a Build Environment *213*
Deleting a Build Environment *214*
Cloning a Build Environment *215*
Build Environment Phases *215*
The Build Environment Phases Overview Screen 216
Inserting a Build Environment Phase 217
Editing a Build Environment Phase 219
Changing the Order of the Build Environment Phases 219
Viewing the Build Environment Phase Parameters 220
Deleting a Build Environment Phase 223
Viewing the Build Environment Parameters *223*
Viewing the Build Environment History *224*

Chapter 20 - Deploy Environments 225

20.1. Creating a Deploy Environment 225
20.2. The Deploy Environments Overview Screen 228
Editing a Deploy Environment *229*
Deleting a Deploy Environment *230*
Cloning a Deploy Environment *230*
Deploy Environment Phases *231*

The Deploy Environment Phases Overview Screen 232
Inserting a Deploy Environment Phase 233
Editing a Deploy Environment Phase 234
Changing the Order of the Deploy Environment Phases 234
Viewing the Deploy Environment Phase Parameters 235
Deleting a Deploy Environment Phase 238
Viewing the Deploy Environment Parameters *239*
Viewing the Deploy Environment History *239*

Chapter 21 - Environment Parameters 240

21.1. The Parameters Overview Screen 241
21.2. Creating Environment Parameters 242
21.3. Viewing the Environment Parameters History 244
21.4. Editing Environment Parameters 244
21.5. Deleting Environment Parameters 245
21.6. Copying Environment Parameters 245

Chapter 22 - Auditing Projects 247

22.1. Auditing a Project 248
22.2. Error and Warning Messages 249
Error Messages *249*
Warning Messages *250*

PART IV - GLOBAL ADMINISTRATION

Chapter 23 - Global Administration 253

Chapter 24 - System 255

24.1. System Settings 255
Local Environment *257*
Security *259*
Mail *259*
Miscellaneous *261*
24.2. Housekeeping 262
24.3. Housekeeping History 265
24.4. Synchronize Build Archive 268

Chapter 25 - History Log 271

Chapter 26 - Users and Groups 275

26.1. Users 275

The Users Overview Screen *276*
Editing User Settings *278*
Disconnecting Users *279*
Viewing the User History *280*
26.2. User Groups 280
Creating a User Group *281*
The User Groups Overview Screen *282*
Editing a User Group *283*
Editing the User Group Members *286*
Deleting a User Group *287*
Viewing the User Group History *288*

Chapter 27 - Machines 289

27.1. Creating a Machine Definition 289
27.2. The Machines Overview Screen 292
27.3. Editing a Machine Definition 294
27.4. Viewing the Machine Parameters 295
27.5. Deleting a Machine Definition 296
27.6. Viewing the Machine Status 297
27.7. The Installed Phases Overview Screen 298
27.8. Viewing the Machine History 300
27.9. Machine Parameters 300
The Machine Parameters Overview Screen *301*
Creating Machine Parameters *303*
Editing Machine Parameters *304*
Deleting Machine Parameters *305*
Copying Machine Parameters *306*
Viewing the Machine Parameter History *306*

Chapter 28 - Transporters 307

28.1. Creating a Transporter Protocol 308
28.2. The Transporter Protocols Overview Screen 309
28.3. FileCopy Transporters 310
Creating a FileCopy Transporter *311*
The FileCopy Transporters Overview Screen *313*
Editing a File Copy Transporter Definition *314*
Deleting a File Copy Transporter Definition *314*
Viewing the File Copy Transporter History *315*
28.4. FTP Transporters 316
Creating an FTP Transporter *316*
The FTP Transporters Overview Screen *319*
Editing an FTP Transporter Definition *320*
Deleting an FTP Transporter Definition *321*
Viewing the FTP Transporter History *322*
28.5. Secure Shell Transporters 323

Creating a Secure Shell Transporter *323*
The Secure Shell Transporters Overview Screen *326*
Editing a Secure Shell Transporter Definition *327*
Deleting a Secure Shell Transporter Definition *328*
Viewing the Secure Shell Transporter History *329*

Chapter 29 - Version Control Repositories 330

29.1. Creating a Repository 330
29.2. The Repository Overview Screen 331
29.3. Subversion Repositories 332
Creating a Subversion Repository *332*
The Subversion Repositories Overview Screen *336*
Editing a Subversion Repository *337*
Deleting a Subversion Repository *338*
Viewing the Subversion Repository History *339*
29.4. Git Repositories 340
Creating a Git Repository *340*
The Git Repositories Overview Screen *343*
Editing a Git Repository *344*
Deleting a Git Repository *345*
Viewing the Git Repository History *346*
29.5. CVS Repositories 346
Creating a CVS Repository *346*
The CVS Repositories Overview Screen *349*
Editing a CVS Repository *350*
Deleting a CVS Repository *351*
Viewing the CVS Repository History *353*
29.6. Visual SourceSafe (VSS) Repositories 353
Creating a VSS Version Control Repository *353*
The VSS Repositories Overview Screen *356*
Editing a VSS Repository *357*
Deleting a VSS Repository *358*
Viewing the VSS Repository History *359*
29.7. ClearCase Repositories 360
Creating a ClearCase Repository *360*
The ClearCase Repositories Overview Screen *363*
Editing a ClearCase Repository *364*
Deleting a ClearCase Repository *365*
Viewing the ClearCase Repository History *366*
29.8. PVCS Repositories 367
Creating a PVCS Repository *367*
The PVCS Repositories Overview Screen *369*
Editing a PVCS Repository *371*
Deleting a PVCS Repository *372*
Viewing the PVCS Repository History *373*
29.9. TFVC Version Control Repositories 373

Creating a TFVC Repository *373*
The TFVC Repositories Overview Screen *376*
Editing a TFVC Repository *378*
Deleting a TFVC Repository *379*
Viewing the TFVC Repository History *380*

Chapter 30 - Issue Tracking 381

30.1. Creating an Issue Tracking System 382
30.2. The Issue Tracking Systems Overview Screen 385
Editing an Issue Tracking System Definition *387*
Deleting an Issue Tracking System Definition *389*
Viewing the Issue Tracking System History *390*
30.3. The Issue Tracking System Properties Overview Panel 390
Creating an Issue Tracking System Property 392
Editing an Issue Tracking System Property 393
Deleting an Issue Tracking System Property 393

Chapter 31 - Scripting Tools 394

31.1. Creating a Scripting Tool 394
31.2. The Scripting Tools Overview Screen 395
31.3. ANT Scripting Tools 396
Creating an ANT Scripting Tool *397*
The ANT Scripting Tools Overview Screen *400*
Editing an ANT Scripting Tool Definition *401*
Deleting an ANT Scripting Tool Definition *402*
Viewing the ANT Scripting Tool History *403*
31.4. Gradle Scripting Tools 403
Creating a Gradle Scripting Tool *404*
The Gradle Scripting Tools Overview Screen *407*
Editing a Gradle Scripting Tool Definition *408*
Deleting a Gradle Scripting Tool Definition *409*
Viewing the Gradle Scripting Tool History *410*
31.5. NANT Scripting Tools 410
Creating a NANT Scripting Tool *411*
The NANT Scripting Tools Overview Screen *413*
Editing a NANT Scripting Tool Definition *414*
Deleting a NANT Scripting Tool Definition *415*
Viewing the NANT Scripting Tool History *416*
31.6. Maven2 Scripting Tools 416
Creating a Maven2 Scripting Tool *417*
The Maven2 Scripting Tools Overview Screen *420*
Editing a Maven2 Scripting Tool Definition *421*
Deleting a Maven2 Scripting Tool Definition *422*
Viewing the Maven2 Scripting Tool History *423*

Chapter 32 - Phases 424

32.1. Creating a Phase Definition 425
32.2. The Phases Overview Screen 428
32.3. Editing a Phase Definition 429
32.4. The Phase Parameters Overview Screen 434
32.5. Creating Phase Parameters 435
32.6. Editing Phase Parameters 437
32.7. Deleting Phase Parameters 440
32.8. Mass Editing Phase Parameters 440
32.9. Mass Replacing Phases 442
32.10. Mass Deleting Phases 445
32.11. Deleting a Phase Definition 447
32.12. Exporting a Phase Definition 447
32.13. Copying a Phase Definition 448
32.14. Viewing the Phase History 449
32.15. Importing a Phase Definition 450

Chapter 33 - Schedules 452

33.1. Creating a Schedule 452
33.2. The Schedules Overview Screen 456
Editing a Schedule Definition 458
Deleting a Schedule Definition 459
Viewing the Schedule History 460

Chapter 34 - Projects 461

34.1. Creating a Project 461
34.2. Cloning an Existing Project 468

Chapter 35 - Miscellaneous 472

35.1. Notifying IKAN ALM Users 472

PART V - COMMAND LINE INTERFACE

Chapter 36 - Command Line Interface (Optional) 475

36.1. Manual Usage of the Command Line interface 475
Creating Level Requests by Specifying Options and Parameters 475
Creating Level Requests Using a Properties File 479
Generating Reports by Specifying Options and Parameters 482
Generating Reports Using a Properties File 488

36.2. Integrating IKAN ALM in an External IDE 490
Integrating IKAN ALM in JDeveloper 490
Integrating IKAN ALM in Eclipse 492
36.3. Automating the Creation of Level Requests Using an ANT script 494

PART VI - APPENDICES

Appendix A - History and Event Logging 497

Appendix B - Predefined Level Parameters 503

Appendix C - Predefined Build Parameters 505

Appendix D - Predefined Deploy Parameters 508

Appendix E - Subversion - General Information 511

E.1. Repository Connection 511
E.2. Repository Definition 511
Specific Fields 512
Trunk Directory 512
Tags Directory 512
Repository Layout 512
Layout Examples 512
Layout 1 (Project-oriented) 513
Layout 2 (Repository-oriented) 514
Layout 3 (One Repository = One Project) 515
Layout 4 (Repository-oriented, no trunk) 516
E.3. Fetching Meta Properties 517

Appendix F - RSS URL Details 518

Appendix G - Phases - General Information 520

G.1. Level Phases 522
Retrieve Code Phase 522
Build Phase 523
Tag Code Phase 523
Deploy Phase 523
Issue Tracking Phase 524
Link File Revisions 524
Cleanup Work Copy Phase 525
Execute Script Phase 525
Custom Level Phase 525

G.2. Build Phases 526
Transport Source Phase *526*
Verify Build Script Phase *526*
Execute Script Phase *527*
Transport Deploy Script Phase *527*
Transport Package Results Phase *527*
Compress Build Phase *527*
Archive Result Phase *528*
Cleanup Source Phase *528*
Cleanup Result Phase *528*
Custom Build Phase *528*
G.3. Deploy Phases 529
Transport Build Result Phase *529*
Decompress Build Result Phase *530*
Verify Deploy Script Phase *530*
Execute Script Phase *530*
Cleanup Build Result Phase *530*
Custom Deploy Phase *531*

Appendix H - Glossary of Terms 532

Index 538

Getting Started

CHAPTER 1
Introduction

IKAN ALM offers a uniquely integrated web-based *Application Lifecycle Management (ALM)* platform for both Agile and traditional software development teams. It combines Continuous Integration and Lifecycle Management, offering a single point of control and delivering support for build and deploy processes (manually generated or automated), approval processes, release management and software lifecycles.

There are many excellent versioning systems on the market today offering complete versioning control, but lacking an integrated solution for automated builds or deploys, library management and other important ALM tasks. Many of their users would like to evolve to a higher level of ALM practices without having to abandon their favorite versioning tool. IKAN ALM tightly integrates with leading existing third-party versioning solutions (e.g., Subversion, Git, Microsoft Team Version Control, IBM® Rational® ClearCase®, Microsoft® Visual SourceSafe®, Serena® PVCS Version Manager, CVS), build and deploy tools (Make, Ant, Gradle, NAnt, Maven2) and issue tracking software (Atlassian® JIRA®, HP Quality Center, CollabNet TeamForge, Trac, Bugzilla,…) and as a consequence offers a unique cross-platform ALM solution.

Today's short release cycles of both Agile and Traditional software development require the application of techniques like nightly builds, Continuous Integration and unit, regression, performance, … testing in order to assure the quality and stability of the delivered code. By automating these back-end activities, IKAN ALM provides an incrementing refresh frequency which can be incorporated into daily development tasks, eliminating bottlenecks in feedback cycles. A tailored lifecycle process including development, testing, quality assurance and production can be easily defined, implemented and enforced, offering a comprehensive framework across all major platforms including Windows, UNIX, Linux and IBM mainframe systems. IKAN ALM also supports a stream-based project model allowing project managers to easily add lifecycles to each version of a project, which makes it easy to differentiate between maintenance, "urgency fix" or release build and deploy processes.

IKAN ALM facilitates compliance with legislation like the Sarbanes-Oxley Act, DO-178B and numerous similar state-specific acts, and enhances an organization's ability to conform to popular process standards such as the CMM, ITIL, Six SigmaT. It can play an integral part in achieving compliance through automation of the business process, including all software development processes, and by providing full audit trails and reporting capability.

IKAN ALM offers your organization one versatile process and one secure pipeline through which applications are delivered safely and efficiently into production, saving time by eliminating errors while meeting corporate standards, and ensuring Quality Management. IKAN ALM continues where versioning systems stop, improving the entire software engineering process, and saving time and money on radical migration efforts.

1.1. How to Use This Guide

The way you use IKAN ALM is determined by the User Category you belong to. The table below lists the three IKAN ALM User Categories.

It is advised to concentrate on the actions matching your IKAN ALM User Category. Click a link to switch to the detailed description.

User Category	Allowed Actions
Regular User	A Regular User can perform the following tasks (depending on his or her ID's security settings): • Managing the Desktop (page 135) • Checking the Build History Screen (page 104) • Approving or rejecting Approvals (page 21) • Creating or updating Level Requests (page 29) • Creating or updating Packages (page 110) and Package Build Groups (page 125)
Project Administrator	A User with Project Administrator access rights will typically perform the following tasks: • Updating Projects (page 146) • Creating or updating Project Streams (page 161) • Creating or updating Lifecycles (page 174) • Creating or updating Levels (page 187) • Creating or updating Build Environments (page 209)or Deploy Environments (page 225) For a complete overview of the Project Administration activities, refer to the section Project Administration (page 145).
Global Administrator	A User with Global Administrator access rights will typically perform the following tasks: • Creating a Project (page 461) • Cloning an Existing Project (page 468) • Edit System Settings (page 255) • Creating or updating Machines (page 289) • Creating or updating Version Control Repositories (page 330) • Creating or updating Transporters (page 307) • Creating or updating Scripting Tools (page 394) • Creating or importing Phases (page 424) • Creating or updating Schedules (page 452) For a complete overview of the Global Administration activities, refer to the section Global Administration (page 253).

CHAPTER 2
Logging in and out

IKAN ALM uses the JAAS interface to base its User ID/Password authentication on an existing Security System in your organization. As a result, these User IDs and User Groups will most likely be maintained in such an external security system outside IKAN ALM, although the Global Administrator can also set up User IDs and User Groups in a configuration file (e.g., for small organizations or demo purposes). Consequently, end users will probably log in to IKAN ALM using a User ID and Password of an external Security System like Active Directory or LDAP.

Each time a User attempts a login, synchronization of the User Groups for this User takes place. There are two types of User Groups: external and internal. During login, the external User Groups are automatically synchronized with an external security system, whereas the internal User Groups are kept intact. For more information on this issue, refer to the section User Groups (page 280).

A user can only successfully log in if doing so does not violate the restrictions imposed by the IKAN ALM License.

- If a User is defined as a *Reserved User*, he/she can always log in. For more information on defining a User as Reserved, refer to the section Users (page 275).
- If a User is not defined as a *Reserved User*, the User is considered being a *Floating User*, and he/she can only log in if the current number of connected Floating Users is less than the number of Floating Users defined by the IKAN ALM License.

This chapter provides more information on:

- Logging in (page 3)
- Logging out (page 5)
- Error Messages (page 6)
- Warning Messages (page 7)

2.1. Logging in

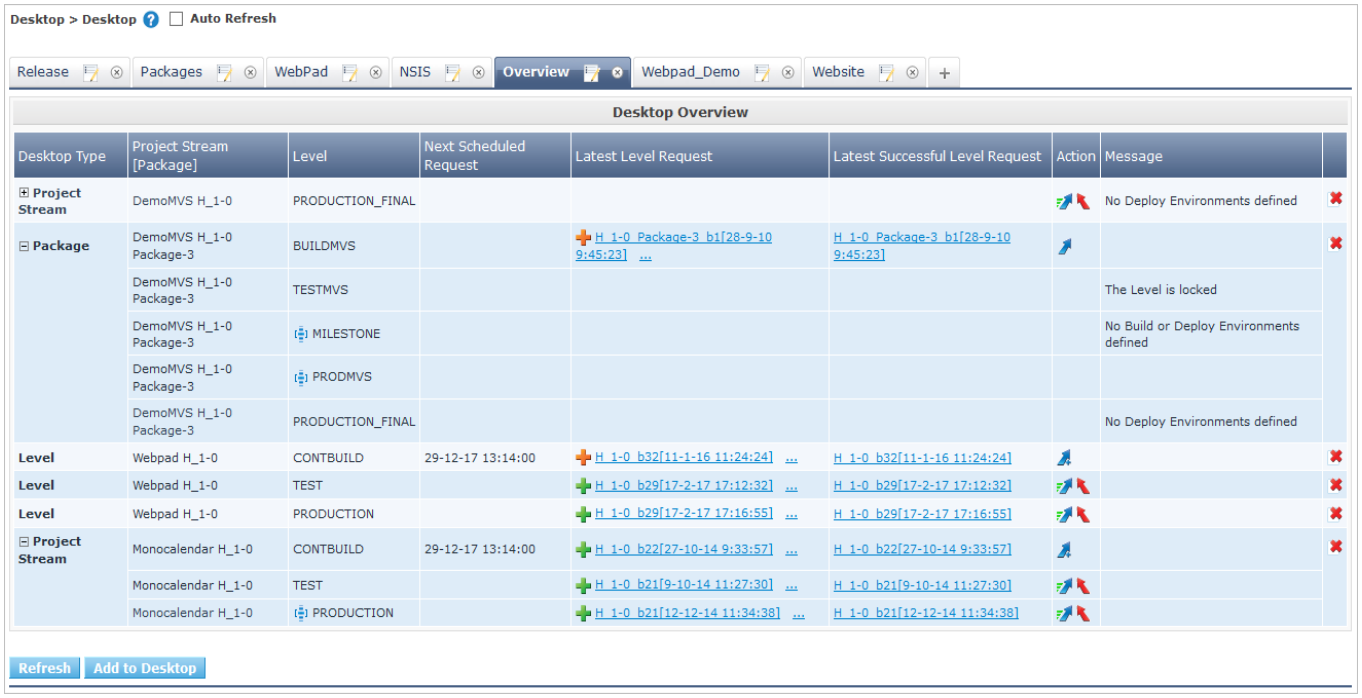
1. Open a browser window and enter the IKAN ALM Login URL.
   Your IKAN ALM Administrator will provide the exact URL.
   This URL is specified on the *Mail* subpanel on the *System Settings* screen. See System on page 255.

The *IKAN ALM Log in* screen is displayed:

Note: Once you know the correct URL, it is useful to save the link in your *Favorites* or *Bookmarks* folder.

2. Enter your IKAN ALM User ID and Password as provided to you by your IKAN ALM Administrator. Then click *Log in*.

   Your personal Desktop will be displayed:

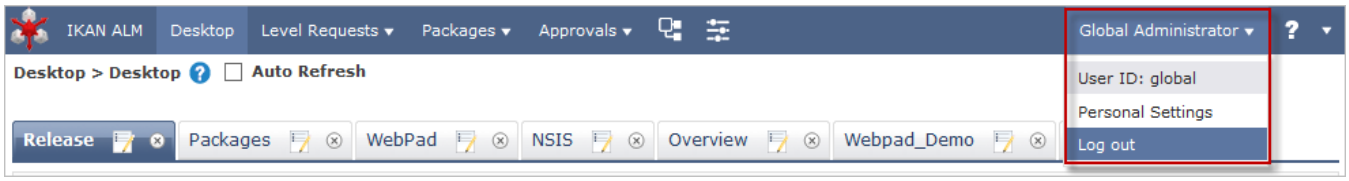


For more information on managing the items displayed on the Desktop, refer to the section Managing the Desktop (page 135).

Note: If you log in to IKAN ALM for the first time, you will be asked to complete your Personal Settings. The *Name* and *E-mail Address* fields are mandatory. For more information, refer to the section Personal Settings (page 15).

2.2. Logging out

1. On the Main Menu, at the right, select *Log out* from the User Settings drop-down menu.



The following screen is displayed:

2. You can now reconnect to IKAN ALM or close your browser window.

Note: If your User is defined as a Floating User (not Reserved User), it is very important that you explicitly log out of IKAN ALM when you have finished using it. If not, your session will remain active after you closed your browser and will continue to count as a connected Floating User, possibly preventing other Users to log in to IKAN ALM.

2.3. Installing a New License

If no valid license is found when you try to log in to IKAN ALM, an error message will be displayed and the link *Install new license* will be displayed on the login page.

Note: For users with Global Administration rights, this installation option is also available on the *About* screen. To access this option, select *About* from the *Help* drop-down menu, on the Main Menu, at the right.

1. Click the *Install New License* link.

The following screen is displayed:

2. Select the license file (alm_license.lic) using the *Browse* button.
   Your IKAN ALM Administrator will provide the exact file name and location.

3. Click the *Submit* button.
   When the License file is accepted, the following screen is displayed:

   If the License file is not accepted, the following error message is displayed:

   Most likely the problem is either a corrupted License file (or a file that doesn't contain an IKAN ALM License), or an expired License in the file. Contact your IKAN ALM Administrator or your IKAN ALM vendor to obtain a new License file.

4. You can now log in to IKAN ALM.
   See Logging in on page 3.

2.4. Error Messages

In case of a problem when trying to log in to IKAN ALM, one of the following error messages could be displayed. Please refer to the table underneath for the corresponding solution.

Error Message	Solution
`No valid License found`	No IKAN ALM License was installed. Install a License as explained in the section Installing a New License (page 5).
`License has expired`.	The IKAN ALM License has expired. Contact your IKAN ALM vendor to obtain a new License.

Error Message	Solution
`Maximum number of connected floating Users (x) reached.` x= number of Floating Users as defined by the license	• Wait and retry until one of the connected Floating Users has logged out from IKAN ALM. or • Contact your IKAN ALM vendor to upgrade to a License that allows more Floating Users.
`There are more reserved active Users (x) found in the database than allowed in the license (y)` x = number of Reserved Users defined in IKAN ALM y= number of Reserved Users as defined by the license	The number of Reserved Users defined in IKAN ALM exceeds the number of allowed Reserved Users in the License. Contact your IKAN ALM vendor.
`There are more machines definitions (x) found in the database than allowed in the license (y)` x = number of Machines defined in IKAN ALM y= number of Named Machines as defined by the license	The number of Machines defined in IKAN ALM exceeds the number of allowed Named Machines in the License. Contact your IKAN ALM vendor.

2.5. Warning Messages

The following warning message could be displayed when logging in to IKAN ALM. Please refer to the table underneath for the corresponding solution.

Warning	Solution
`License will expire within x day(s).` x= between 14 and 0 days	The IKAN ALM License will expire soon, contact your IKAN ALM vendor.

CHAPTER 3
User Interface Overview

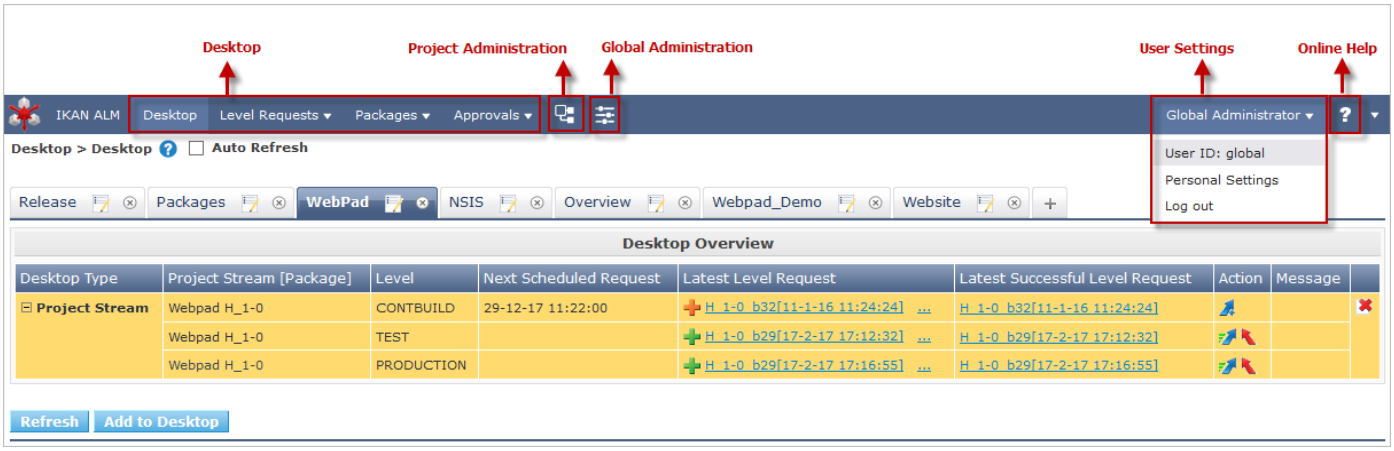
Refer to the following sections for detailed information about:

- Navigating Through IKAN ALM (page 8)
- The IKAN ALM Screen (page 9)
- Auto Refresh (page 12)

3.1. Navigating Through IKAN ALM

Main Menu

The IKAN ALM user interface contains 3 main sections: the Desktop, the Project Administration section and the Global Administration section. Those sections can be accessed from the *Main Menu*, the blue toolbar on top of the screen. At the right, it contains the User Settings of the current user as well as the context-sensitive help pages.

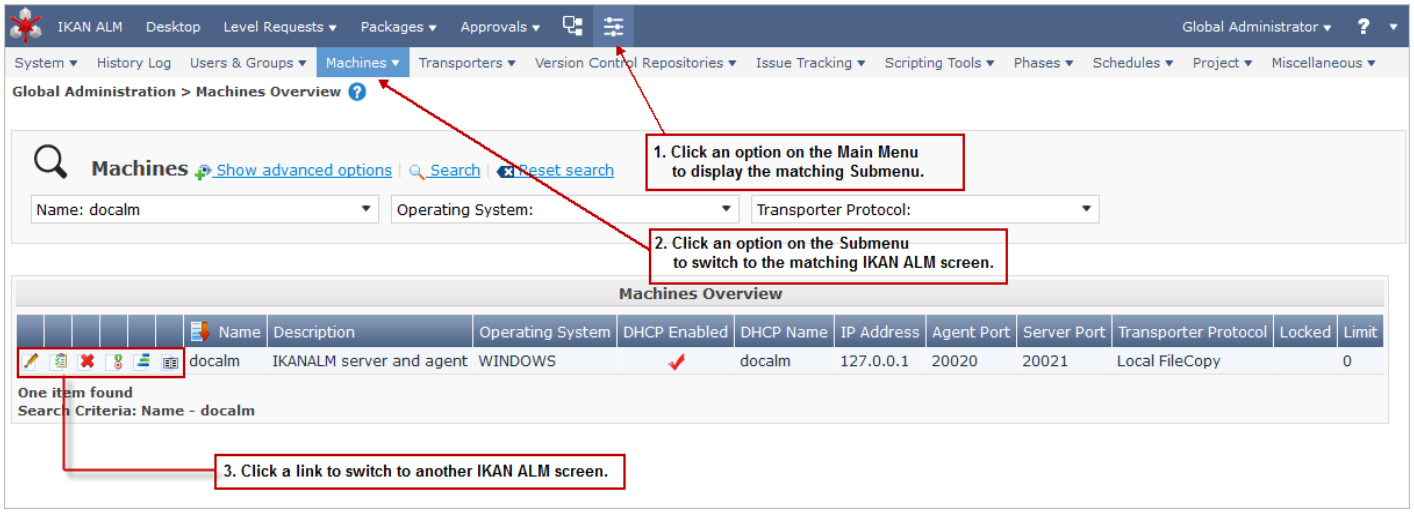


Submenus

When you select the Global or Project Administration option from the Main Menu, the dependent Submenu is displayed in the gray banner below the Main Menu.

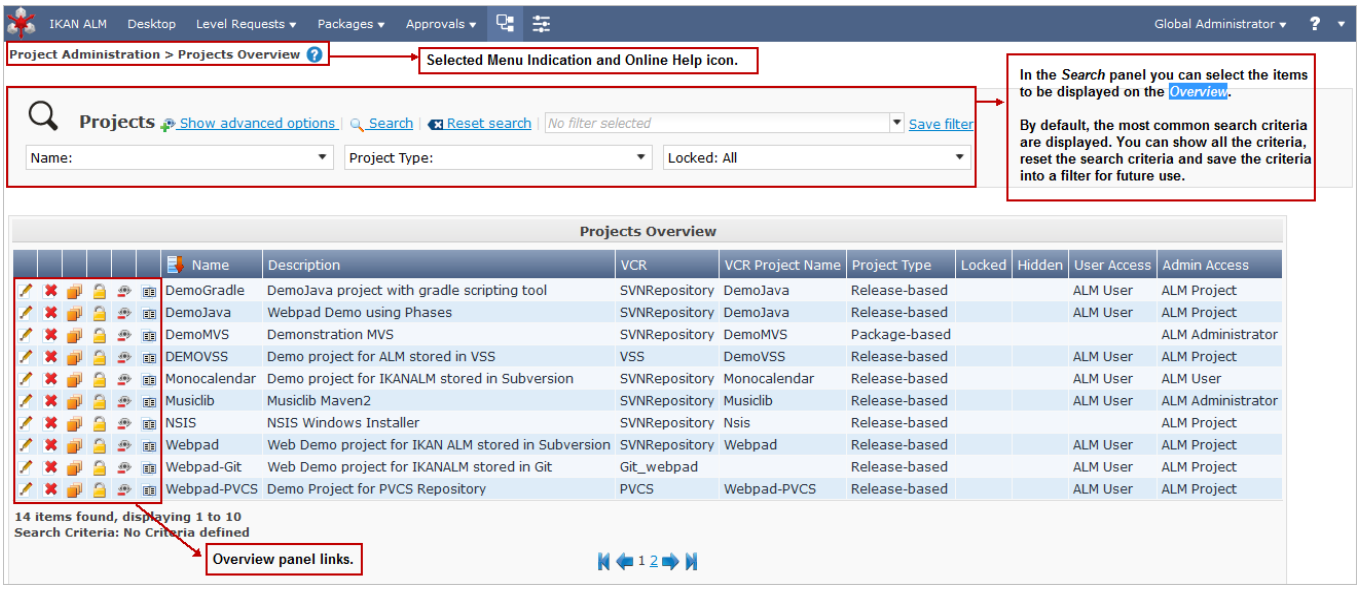
Submenu options followed by a small triangle contain a drop-down menu with extra selection options.

Links appear on or below the Overview panels of the IKAN ALM screens. When you select them, another IKAN ALM screen is displayed.



3.2. The IKAN ALM Screen

An IKAN ALM screen contains the following main items.



Selected Menu Indication

The *Selected Menu Indication* below the Submenu shows how the user switched to the displayed IKAN ALM Screen. The following format is used: *Main Menu Option* > *Submenu Option*

Search Panels

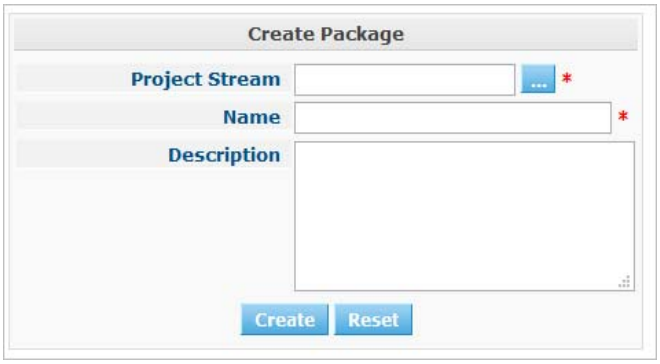
Search panels allow specifying search criteria, which restrict the list of items displayed in the *Overview* panel. While defining those search criteria in comboboxes and text fields, you may use the standard * and ? wild card characters. The asterisk can represent 0, 1 or more characters. The question mark can represent exactly 1 character.

By default, only the most common search criteria are displayed on the search panel. You can display all available search criteria by clicking the *Show advanced options* link.

Using the *Reset search* link you can clear all search criteria and display the full list of items on the Overview.

On some of the more complex search panels, you can save the search criteria as a filter, to be able to reuse them afterwards. If a filter already exists, you can select it from the drop-down menu. For more info on using filters, refer to the section Defining Search Filters (page 16).

Create Panels



Create panels allow defining new IKAN ALM Objects. Mandatory fields are marked with a red asterisk.

Clicking the *Create* button will create the object. The *Reset* button will, depending on the object you are creating, clear the fields and/or restore the initial values.

Overview Panels

Overview panels are displayed below the *Search* or *Create* panel. If no search criteria have been defined, the overview lists all IKAN ALM Objects of the selected type. If search criteria have been defined, only IKAN ALM Objects matching those search criteria are displayed.

Note: All applied search criteria will be indicated in the footer of the results list.

The *Overview* panel may list one or more links to the left of the listed IKAN ALM Objects. The options offered depend on the displayed IKAN ALM Objects and on the User's Access Rights.

If you have User Access Rights, the following links are available:

Option	Description
View	When you select the *View* option, a *View* panel will be displayed below the *Selected Menu Indication*. You will not be able to edit any values.
History	This option is available on the Overview screens in Global Administration and Project Administration. It is also available in some detailed Global and Project Administration interfaces, e.g on the System Settings screen, or the Edit Project screen. All operations on objects and components related to those sections are logged in the IKAN ALM database. The *History* link allows you to display their history of events.
Status	This option is available on the *Machine Overview* screen. Use this link to display the activity status of the Machine and the last lines of output of the Agent Daemon process running on this Machine.

If you have Project Administrator or Global Administrator Access Rights, the following additional options may be available:

Option	Description
Edit	When you select the *Edit* option, an *Edit* panel will be displayed below the *Selected Menu Indication*. You will be able to edit the required values and save the changes.
Delete	When you select the *Delete* option, a *Delete Confirmation* panel will be displayed below the *Selected Menu Indication*. You will be able to confirm or cancel the deletion.
Lock Unlock	This option is specific for the *Project* panel and the *Project Streams Overview* panel.
Hide Show	These options are specific for the *Projects Overview* and the *Project Streams Overview* panels. They allow you to hide specific Projects or Project Streams on the search results Overview.
Set Optional Set Required	These options are specific for the *Lifecycles Overview* screen. They allow you to make Levels optional.
Edit Properties	This option is specific for the *Issue Tracking Systems Overview* screen. It allows to modify the value of the property to ensure a correct usage of the Issue Tracking System.
Disconnect User	This option is specific for the *Users Overview* screen. It allows disconnecting Users before their session has timed out.

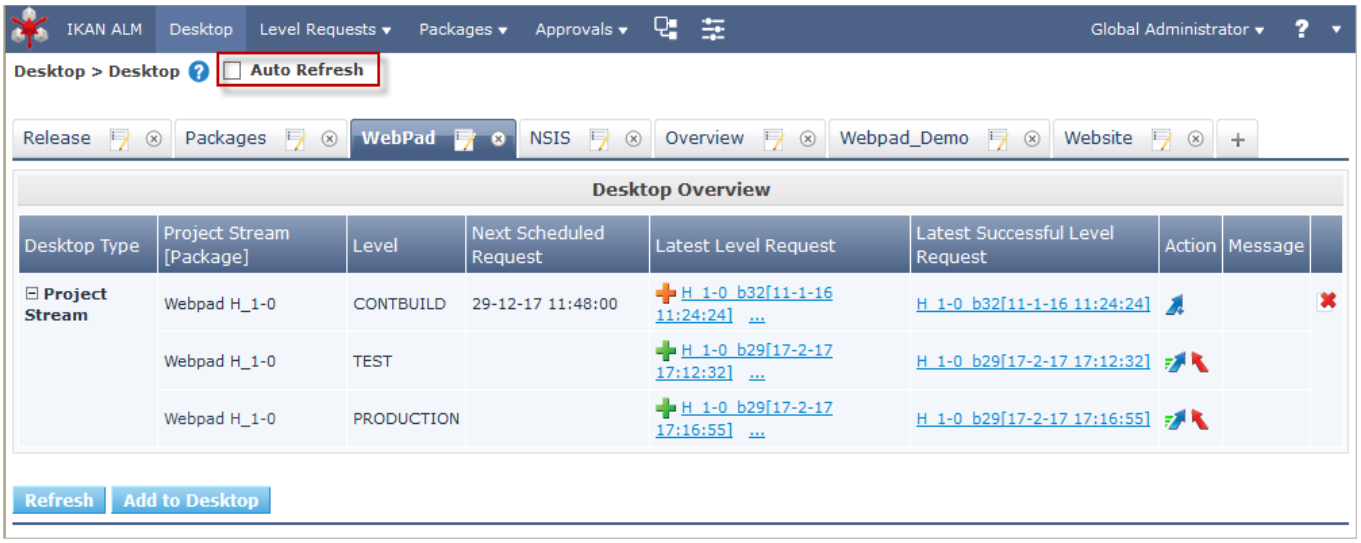
Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

3.3. Auto Refresh

If your IKAN ALM Global Manager has defined an *Automatic Refresh Rate* on the *Miscellaneous* panel in the System Settings (page 255), the *Auto Refresh* option will be available on the Desktop, and on the Level Requests and Builds and Deploys overviews.

If the *Auto Refresh* option is activated, the screen will be refreshed following the interval specified by the Auto Refresh Rate specified in the System Settings.

Select the option to toggle its status.



Desktop

CHAPTER 4
Desktop

The Desktop makes up the work environment for the regular IKAN ALM end user.
It gives access to common actions as:

- creating or verifying a build (by a developer)
- verifying the project status (by a project leader)
- delivering a build to the QA environment (by a software tester)
- approving a deploy to the production environment (by a release manager)
- ...

This section describes the following Desktop Administration tasks:

- Level Requests (page 29)
- Packages (page 110) and Package Build Groups (page 125)
- Approvals (page 21)
- Managing the Desktop (page 135)
- Auto Refresh (page 12)
- Personal Settings (page 15)

CHAPTER 5
Personal Settings

The Personal Settings section focuses on settings directly related to your account, such as your name, e-mail address, phone number, ...

It is also the place where you store your personal search filters which you can use to specify which objects need to be displayed on the different overview lists.

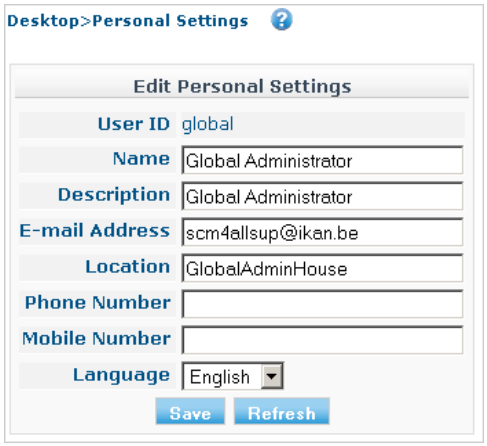
Refer to the following procedures for more information:

- Editing Your Personal Settings (page 15)
- Defining Search Filters (page 16)

5.1. Editing Your Personal Settings

This option allows you to edit your personal settings. These are the settings you (or the Global Administrator using your User ID) defined when you logged on to IKAN ALM for the first time.

1. Select *Desktop > Personal Settings*.
   The *Personal Settings* screen is displayed:



2. Update the fields as required.
   The following fields are available:

Field	Meaning
User ID	This read-only field contains your IKAN ALM User ID. As User IDs are administered outside IKAN ALM, you cannot change it.
Name	In this field, you may edit your complete name. This field must be filled out.

Field	Meaning
Description	In this field, you may enter or edit a free description.
E-mail Address	In this field, you may edit your e-mail address. Notifications generated by IKAN ALM (for example, on failed or succeeded Level Requests, on Approvals, ...) will be sent to this e-mail address. This field is mandatory.
Location	In this field, you may enter or edit the description of your working location.
Phone Number	In this field, you may enter or edit your phone number.
Mobile Number	In this field, you may enter or edit your mobile number.
Language	From the drop-down list, you can select the language in which the IKAN ALM User Interface must be displayed, when you log on to IKAN ALM with this User ID. You can choose between English, French and German.

3. Click *Save* to save your changes.

   You may also click *Refresh* to retrieve the settings from the database.

5.2. Defining Search Filters

On the Overview screens, the improved search options allow to specify which objects will be displayed on the overview list.

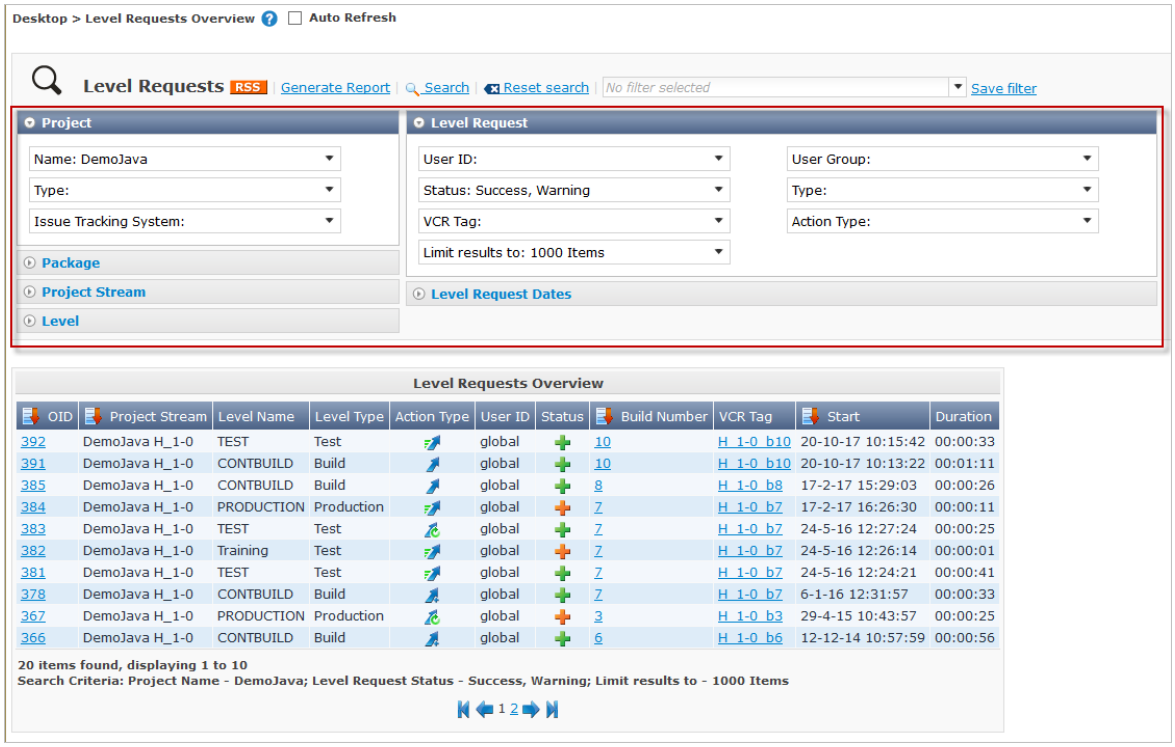
On some of the overview screens, after having specified the search criteria, you can save the search specification using the *Save filter* link. This way, you can reuse the same criteria for future search actions.

Refer to the following procedures for more information:

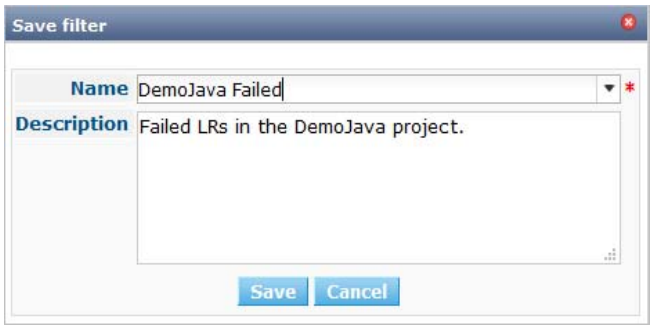
- Creating a search filter (page 17)
- Selecting a Search Filter (page 18)
- Managing Filter Definitions (page 18)
- Modifying a filter's search criteria (page 20)

Creating a search filter

1. Specify the required (advanced) search criteria.



2. Click the *Save filter* link.
   The following pop-up window is displayed.



3. Enter the name and a description for the new filter and click the *Save* button.
   The filter will now become available in the filter drop-down menu.

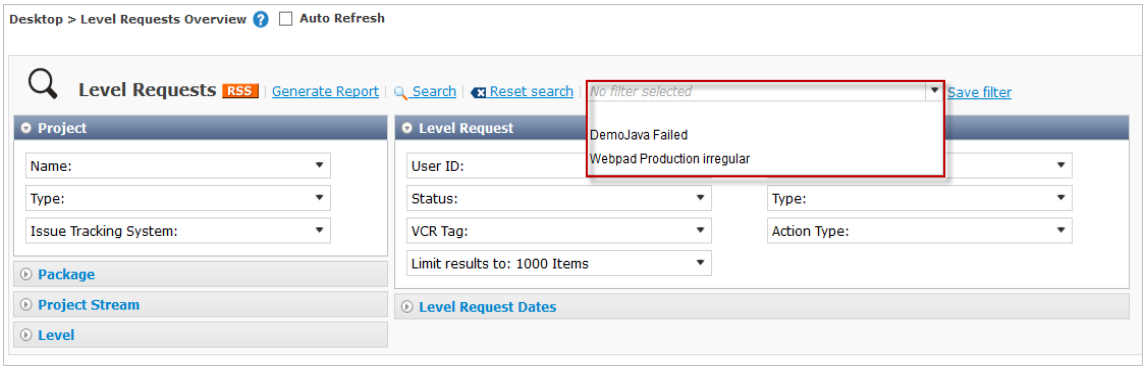
4. On this screen you can also manage the filter definitions.
   For more information, refer to the section Managing Filter Definitions (page 18).

Note: Editing the name and description of the filter, can be done on the *Personal Settings* screen (Managing Filter Definitions (page 18)).

Selecting a Search Filter

On the Overview screens, search filters can be used instead of manually specifying search criteria, you can select an existing filter.

1. To display the list of existing filters, click the down arrow of the *No filter selected* drop-down box.



2. Select the required filter from the list.

   The search criteria will be automatically filled in and the filtered result will be displayed on the overview.

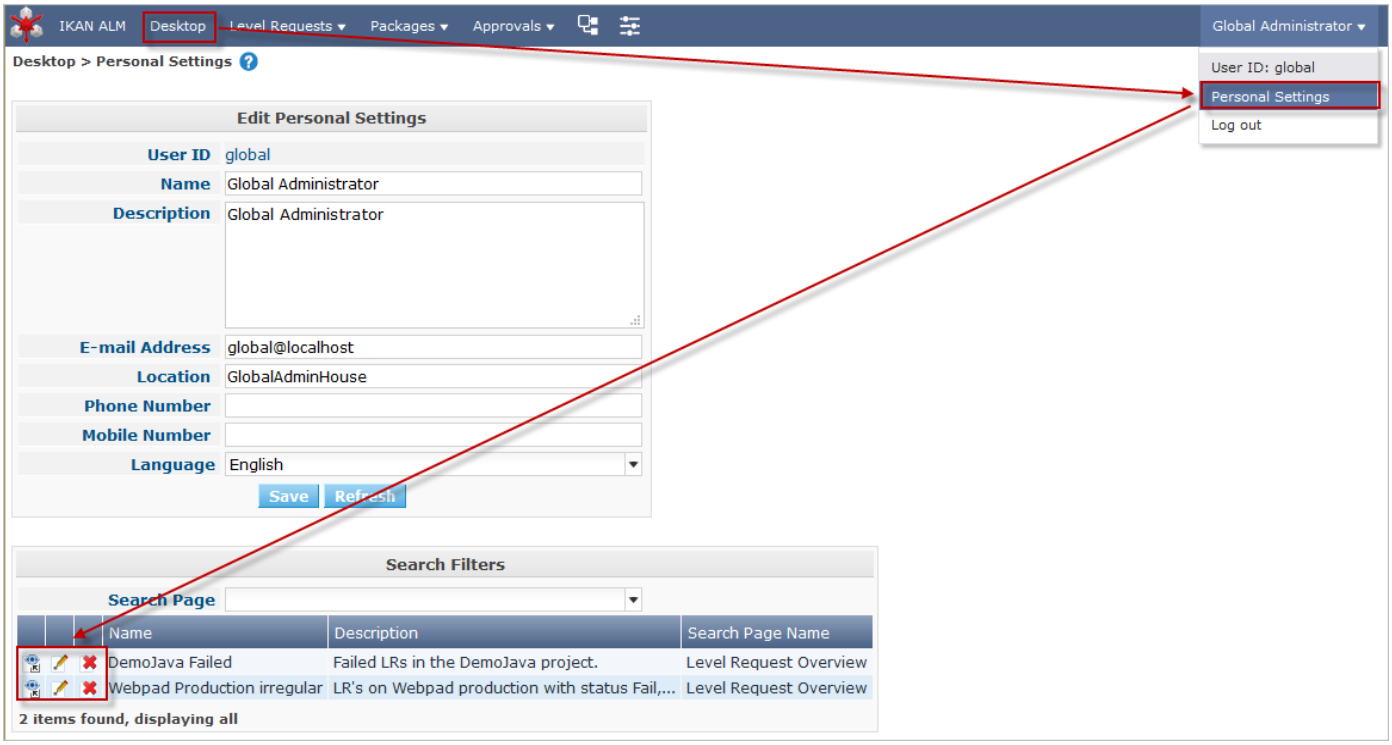
Managing Filter Definitions

The Search Filters panel on the Personal Settings screen allows you to view the current filter definitions, to modify their name and description or to delete a filter.

1. Select Desktop > Personal Settings.

   The *Search Filters* panel displays the list of all filters defined for the current user.

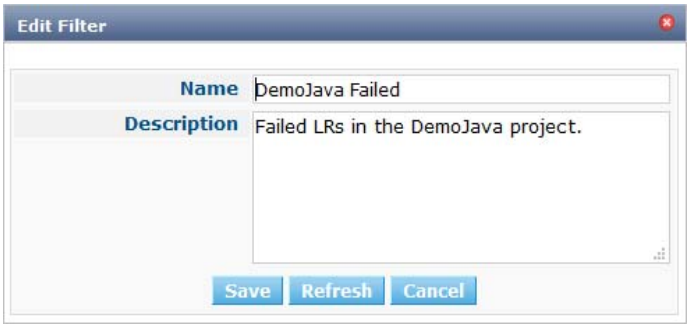
If required, you can limit the list of displayed filters by selecting the required *Search Page* (Package Overview, Projects Overview, Level Request Overview of Build and Deploy Overview) from the drop-down list.



2. Click the *View* icon in front of the filter, to view the specified search criteria.
You will be forwarded to the related Overview screen and the search criteria will be automatically applied.

Note: If required, you can now modify the search criteria and save them to the filter definition. See also Modifying a filter's search criteria (page 20).

3. Click the *Edit* icon in front of the filter, to modify the name or the description.



Confirm the modification, by clicking the *Save* button.

Note: Modifying the search criteria is only possible on the Overview screens themselves. See Modifying a filter's search criteria on page 20.

4. Click the *Delete* icon in front of the filter, to delete a specific filter.

Confirm the deletion, by clicking the *Delete* button.

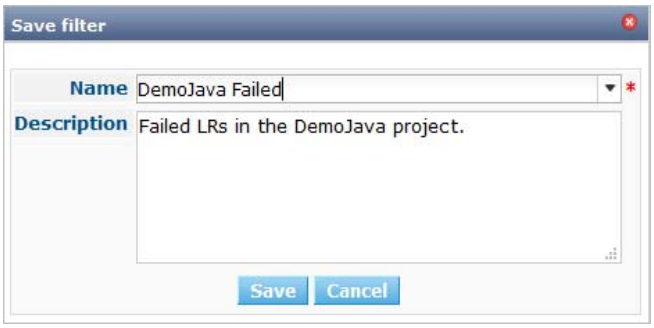
Modifying a filter's search criteria

Modifying the search criteria of a filter is only possible on the Overview screens themselves.

1. Select the filter you want to modify.
   There are two possible ways to do so:
   - via the *Edit* icon on the Search Filters panel on the Personal Settings screen (Managing Filter Definitions (page 18)), or
   - directly on the concerned Overview, by selecting the filter from the drop-down list.

2. Adapt the search criteria.

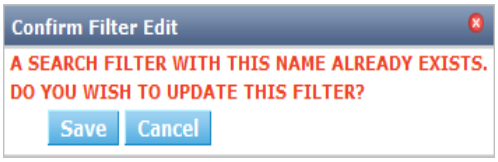
3. Click the *Save filter* option.
   The *Save Filter* pop-up window is displayed.



4. If required, you can also at the same time adapt the *Name* and *Description*.

5. Click *Save*.

6. As the filter already existed, you need to confirm the update of an existing filter by clicking once again *Save*.



CHAPTER 6
Approvals

There are 2 types of Approvals which may be set on Test or Production Levels (Level Approvals (page 205)).

- A *Pre-Approval* is created when delivering a Level Request to the Level, i.e., before the Level Request is executed.
- A *Post-Approval* is created after the Level Request has been executed, enabling to verify if the actions on the Level and/or the Environments were successful.

A Pre-Approval enables a verification moment before executing a Level Request on a Test or Production Level. It adds an extra condition to the execution of the Level Request: not only the Requested Date/Time must be expired, but the Pre-Approval must also be granted by an IKAN ALM User. As long as both conditions are not fulfilled, the Level Request will not be handled. When a Pre-Approval is rejected, the Level Request will never be handled.

A Post-Approval enables a verification moment after the execution of the Level Request on a Test or Production Level. It adds an extra condition to the determination of the end status of a Level Request: not only must all Build(s) or Deploy(s) have been ended successfully, but the Post-Approval must also be granted by an IKAN ALM User. As long as the Post-Approval is not handled, the Level Request status will not be set to warning or success. When a Post-Approval is rejected, the Level Request can never be delivered to the next Level in the Lifecycle.

Every approval is protected by a User Group: only a User belonging to that User Group may grant or reject the Approval.

For example, assume a Lifecycle with one Build, Test and Production Level: a Post-Approval on the Test Level will generate a verification moment after the Level Request is executed on the Test Level. This enables the people from Test to judge whether the Level Request is qualified to deliver to the Production Level. Additionally, a Pre-Approval is set on the Production Level. This generates a verification moment before the Level Request is executed on the Production Level, enabling people from Production or management to judge whether everything is OK before “going into Production”.

It is also possible to have a hierarchy of Approvals on a Level, e.g., with one or more Pre- and one or more Post-Approvals. Every Approval will result in a notification (mail or net send) to the Users belonging to the protecting User Group, asking them to grant or reject the Approval via the Outstanding Approvals interface.

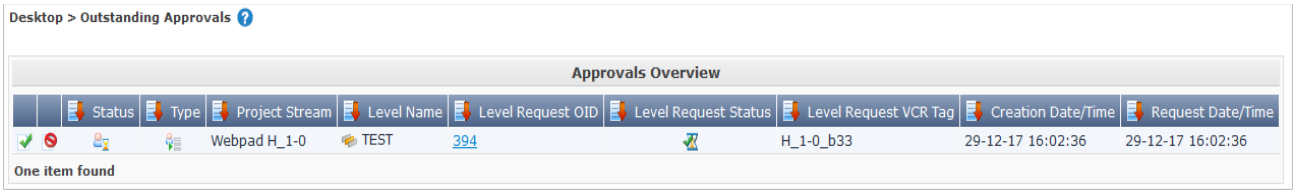
Refer to the following sections for more information about:

- The Outstanding Approvals (page 22)
- Approving Outstanding Approvals (page 23)
- Rejecting Outstanding Approvals (page 24)

6.1. The Outstanding Approvals

1. Select *Approvals* > *Outstanding Approvals* on the Main Menu.

   The *Outstanding Approvals* screen displays all currently outstanding Approvals which you (or another member of your User Group) can approve or reject.



2. Verify the information.

   The Overview only lists the Level Requests you are entitled to approve or reject. For each displayed Outstanding Approval, the following information is available:

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

Field	Meaning
Status	This field displays the *Awaiting Approval* status. This status means that you (or another member of your User Group) can approve or reject this Approval.
Type	This field contains the Approval Type. There are two possibilities: • *(Pre-Approval)* This Approval must be given before the Sources or Build Result are delivered to the requested Level. • *(Post-Approval)* This Approval must be given at the end of the Level Request, after the Build(s) or Deploy(s) linked to the Level Request have been executed.
Project Stream	This name in this field is composed as follows: • Project name • Project Stream type: H for Head and B for Branch. • Prefix and, optionally, suffix of the Project Stream Example: Webpad H_1-0 With: H = Head and 1-0 = Prefix
Level Name	This field contains the name and type of the Level to which a Build will be delivered as a result of this Level Request. There are two possible types: • *Test*: The Approval is associated with a TEST Level. • *Production*: The Approval is associated with a PRODUCTION Level. See Levels on page 187.
Level Request OID	This field contains the sequential number of the Level Request. All Level Requests defined for a Project are numbered sequentially.
Level Request Status	This field contains the Level Request status icon. There are 2 possibilities: • Awaiting Pre-Approval • Awaiting Post-Approval

Field	Meaning
Level Request VCR Tag	This field contains the VCR Tag of the Level Request. This Tag matches a Build with its source code in the VCR. The format of the VCR Tag normally matches the Tag Template defined for the Stream. See Project Streams on page 161. However, the user can override the default VCR Tag while creating a Level Request, so that the Tag Format can be completely different.
Creation Date/Time	This field contains the Date and Time when the Level Request was created.
Request Date/Time	This field contains the Date and Time when the execution of the Level Request was asked. The Level Request will not be handled before this point in time is reached.

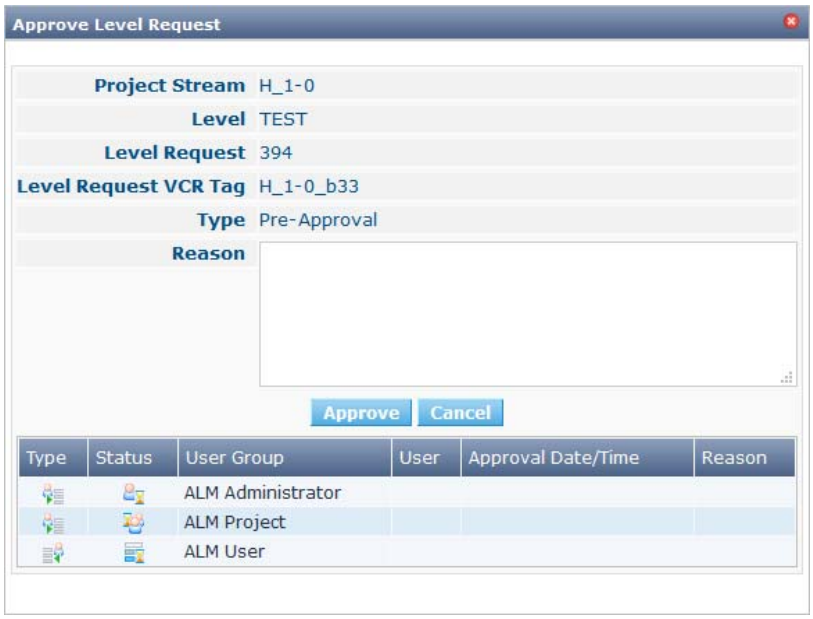
3. Use the appropriate link for approving or rejecting an Approval.
   The following links are available in front of each Outstanding Approval:
   - *Approve*. See Approving Outstanding Approvals on page 23.
   - *Reject*. See Rejecting Outstanding Approvals on page 24.

6.2. Approving Outstanding Approvals

1. Select *Approvals* > *Outstanding Approvals* on the Main Menu.

Note: The *Approve* and *Reject* links are also available on the *Approvals Overview* screen.

2. Click the *Approve* link to approve the Level Request.
   The *Approve Level Request* pop-up window is displayed:



3. Verify the information on the *Level Approval List* panel.

Field	Meaning
Type	This field contains the Approval Type. There are two possibilities: • *Pre-Approval* • *Post-Approval*
Status	This field contains the approval status. The following status icons are available: • *Awaiting Approval*: you (or another member of your User Group) can grant or reject this Approval. • *Awaiting Predecessor Approval*: a higher-level Approval (having a lower sequential number) must be granted, before you (or another member of your User Group) can grant or reject this Approval. • *Approved*: the Approval has been granted • *Rejected*: the Approval has been rejected • *Canceled*: a preceding Approval has been rejected or, in case of a canceled Post-Approval, the Level Request has been aborted or canceled in the meantime. • *Awaiting Level Request Finish*: this status indicates that the Level Request has not been executed yet.
User Group	This field contains the name of the IKAN ALM User Group of which a member must grant or reject the approval
User	This field contains the ID of the User, who granted or rejected the approval. This field is empty for outstanding approvals.
Approval Date/ Time	This field contains the date and time, when the approval was granted or rejected. This field is empty for outstanding approvals.
Reason	This field contains the Reason of approval or rejection, as defined by the user. This field is empty for outstanding approvals.

4. Optionally, enter the reason for approving the Level Request on the *Approve Level Request* panel.

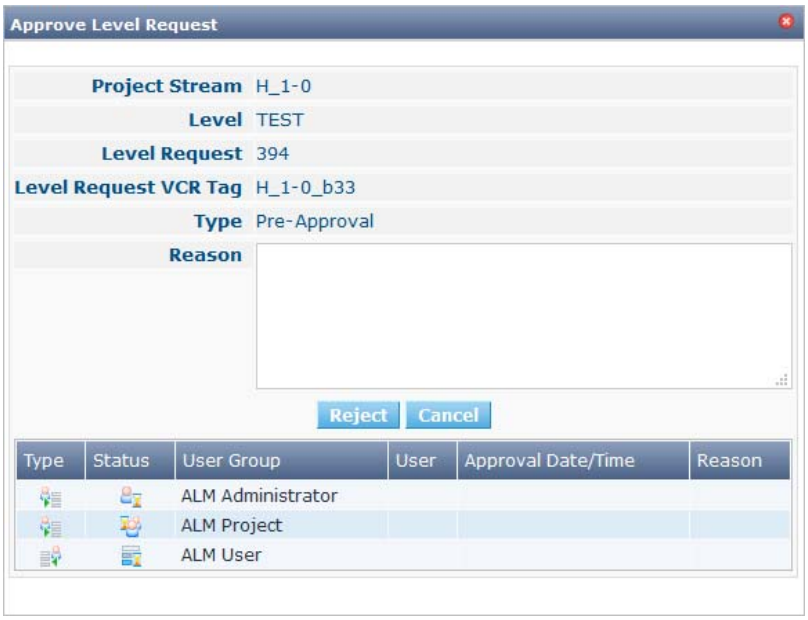
5. Click *Approve* to confirm the action.
You can also click *Cancel* to close the pop-up window without approving the Level Request.

6.3. Rejecting Outstanding Approvals

1. Select *Approvals* > *Outstanding Approvals* on the Main Menu.

Note: The *Approve* and *Reject* links are also available on the *Approvals Overview* screen.

2. Click the *Reject* link to reject the Level Request.
   The *Reject Level Request* pop-up window is displayed.



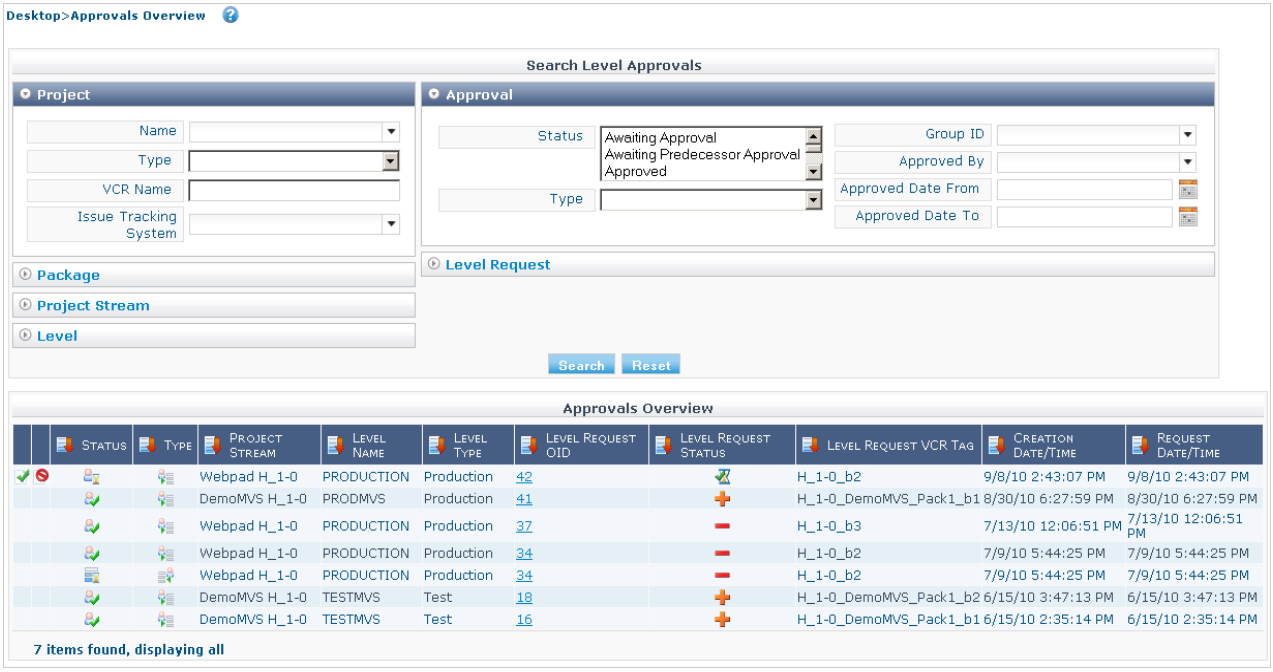
3. Verify the information on the *Level Approval List* panel.
   The *Level Approval List* displays additional information concerning the Approvals. See Approving Outstanding Approvals on page 23.

4. Enter the reason for rejecting the Level Request on the *Approve Level Request* panel.

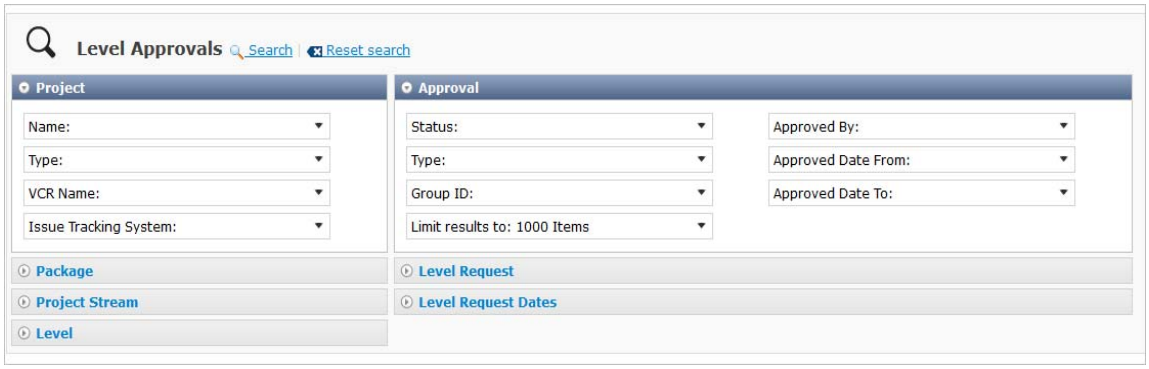
5. Click *Reject* to confirm the action.
   You can also click *Cancel* to close the pop-up window without rejecting the Level Request.

6.4. The Approvals Overview Screen

1. Select *Approvals > Overview Approvals* on the Main Menu.
   The following screen is displayed:



2. Use the search criteria on the *Search Level Approvals* panel to only display the Approvals you are looking for.



The following options are available:

- Search: in principle it is not necessary to click the *Search* option. The results on the overview will be automatically synchronized in function of the selected criteria.
- Reset search: to clear all search criteria and display the full list of items.

3. Verify the Approval information fields.

Note: Outstanding Approvals for you (or other members of your User Group) are preceded by the *Approve* and *Reject* icons.
For more information, refer to Approving Outstanding Approvals (page 23)and Rejecting Outstanding Approvals (page 24).

Field	Meaning
Status	This field contains the approval status. The following status icons are available: • *Awaiting Approval*: you (or another member of your User Group) can grant or reject this Approval. • *Awaiting Predecessor Approval*: a higher-level Approval (having a lower sequential number) must be granted, before you (or another member of your User Group) can grant or reject this Approval. • *Approved*: the Approval has been granted • *Rejected*: the Approval has been rejected • *Canceled*: a preceding Approval has been rejected or, in case of a canceled Post-Approval, the Level Request has been aborted or canceled in the meantime. • *Awaiting Level Request Finish*: this status indicates that the Level Request has not been executed yet.
Type	This field contains the approval type. There are two possibilities: • *(Pre-Approval)* This Approval must be granted before the Source code or the Build result are delivered to the Level. • *(Post-Approval)* This approval must be granted at the end of the Level Request, after the Builds or Deploys associated with the Level Request have been executed.
Project Stream	This name in this field is composed as follows: • Project name • Project Stream type: H for Head and B for Branch. • Prefix and, optionally, suffix of the Project Stream Example: Webpad H_1-0 With: H = Head and 1-0 = Prefix
Level Name	This field contains the name and type of the Level, to which a Build will be delivered as a result of this Level Request. See Levels on page 187. There are two possible types: • *Test*: The Approval is associated with a TEST Level. • *Production*: The Approval is associated with a PRODUCTION Level. See Levels on page 187.
Level Request OID	This field contains the sequential number of the Level Request. All Level Requests defined for a Project are numbered sequentially. Select this link to switch to the *Level Request Detail* screen.

Field	Meaning
Level Request Status	This field contains the Level Request Status. The following status indication icons are possible: • (Success): the Level Request is executed successfully. • (Warning): the Level Request has been successfully executed, but at least one non-critical Level, Build or Deploy Phase failed. • (Fail): the execution of the Level Request as a whole failed. This is due to the failure of one or more critical Level, Build or Deploy Phases. • (Run): the Level Request is being executed at this moment. • (Aborting): the Level Request is being aborted at this moment. • (Awaiting requested Date/Time): the requested execution is in the future. • (Awaiting Pre-Approval or Awaiting Post-Approval): the Level Request is awaiting a Pre- or Post-Approval. • (Rejected): An Approval associated with the Level Request was rejected. The Level Request will never be executed. • (Canceled): the Level Request has been canceled before it was run. It will never be executed. • (Aborted): the Level Request has been aborted during execution. The results (such as Build Results) that were already available at the time of the abort have been cleaned up and cannot be used.
Level Request VCR Tag	This field contains the VCR Tag of the Level Request. This Tag matches a Build with its source code in the VCR. The format of the VCR Tag normally matches the Tag Template defined for the Stream. See Project Streams Overview on page 165. However, the user can override the default VCR Tag while creating a Level Request (and is obliged to do so for a Build Level Request in a Tag Based Project Stream), so that the Tag Format can be completely different.
Creation Date/ Time	This field contains the Date and Time when the Level Request was created.
Request Date/ Time	This field contains the Date and Time when the execution of the Level Request was asked. The Level Request will not be handled before this point in time is reached.

4. Verify the details of a specific Level Request.

   Click the Level Request's *OID* link in front of the required Level Request.

   For more information, refer to the section Level Request Detail (page 65).

5. Use the appropriate link for approving or rejecting an Approval.

   The following links are available in front of each Outstanding Approval:

   - *Approve*. See Approving Outstanding Approvals on page 23.
   - *Reject*. See Rejecting Outstanding Approvals on page 24.

CHAPTER 7
Level Requests

Level Requests are action requests in relation to a Level.

From a logical point of view, they are requests to *create* a Build on a Build Level, to *deliver* a Build to the next Level (Test or Production) or to *rollback* a Build on a Level (Test or Production) defined in the applicable Lifecycle.

From a physical point of view, Level Requests can match a Build with or without a Deploy, a Rebuild with or without a Deploy or just a Deploy.

The following sections describe these Level Request types in detail:

- (Re)Build Level Requests (page 30)
- (Re)Build and Deploy Level Requests (page 35)
- Deploy Level Requests (page 42)

The following sections deal with the procedures involved when creating new Level Requests:

- The Create Level Request: Select Level Screen (page 45)
- Creating a Build Level Request (page 48)
- Creating a Deliver Build Level Request (page 52)
- Creating a Rollback Build Level Request (page 55)

The following sections deal with the procedures involved when using the *Level Requests Overview*:

- Level Requests Overview (page 58)
- Level Request Detail (page 65)
- Builds and Deploys Overview (page 107)

The following section deals with the Build History:

- Build History Screen (page 104)

7.1. Level Request Action Flows

The following sections describe how Level Requests are handled once they have been created, with the default behavior as starting point. When adapting the Level Phases (page 197), the Build Environment Phases (page 219) or the Deploy Environment Phases (page 234), this will result in an action flow differing from this default flow.

The following different action flows are possible, depending on the composition of the Level (the Level Type and the related Build and/or Deploy Environment(s)):

- (Re)Build Level Requests (page 30)
- (Re)Build and Deploy Level Requests (page 35)
- Deploy Level Requests (page 42)

(Re)Build Level Requests

The following graphic displays the action flow of a Build Level Request or a Rebuild Level Request. If the _ Level Phases (page 197) or the _ Build Environment Phases (page 219) have been modified, the sequence of actions may be different.

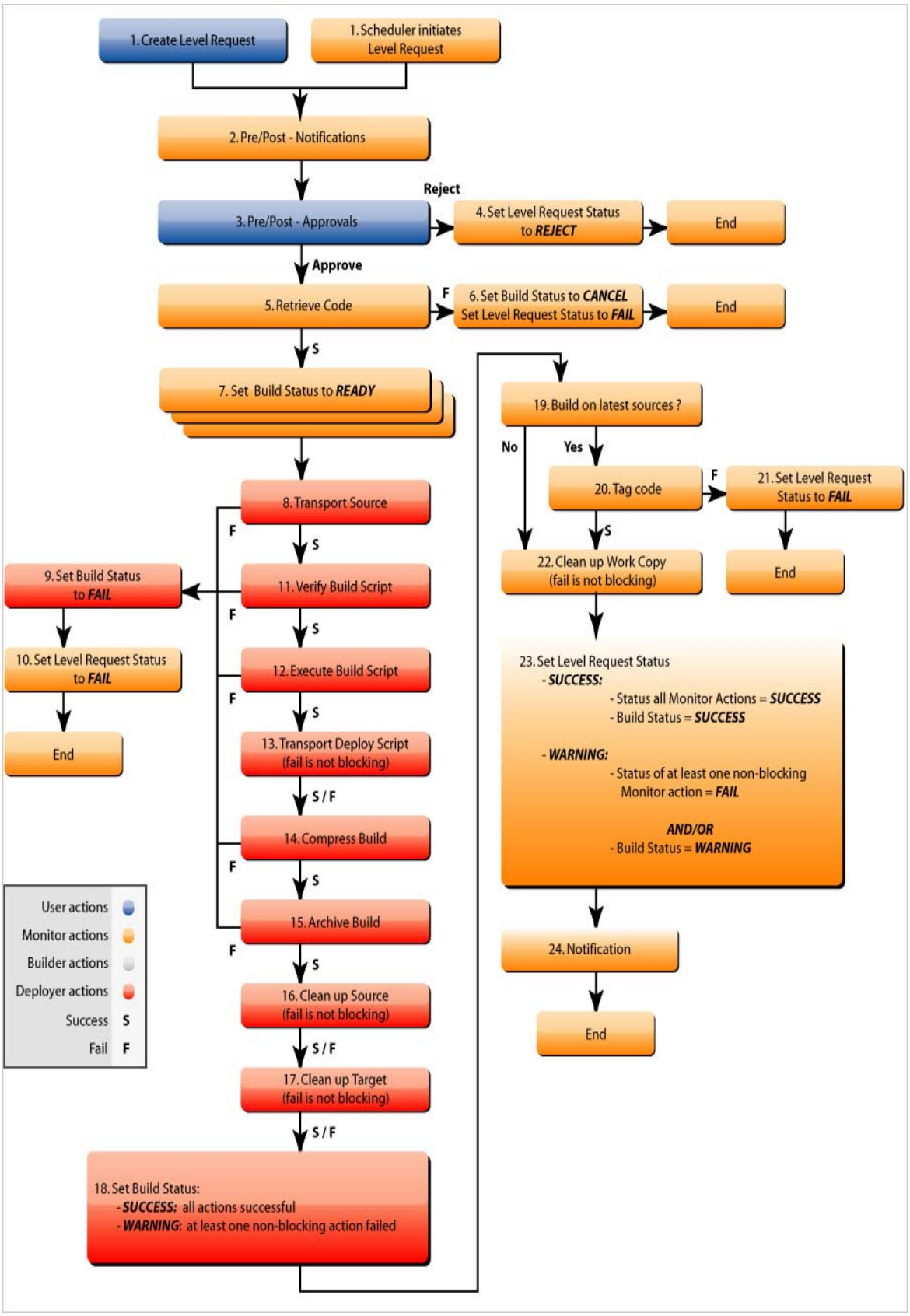
The Build Level Request is generated for the first Level in a Lifecycle, e.g., a Level on which a Nightly Build (or even a Continuous Build) is executed to verify the stability of the latest sources in the VCR. Usually, such a Level has one Build Environment.

The Rebuild Level Request on a Level later on in the Lifecycle will not be executed on the latest sources, but on code that has already been tagged during the Build Level Request on a previous Level.

Note: The main difference between a Build and a Rebuild is step 19.

If the answer is *No*, the Build was not done with the latest sources, but with previous tagged sources and the Level Request is a Rebuild. If the answer is *Yes*, the build was done with the latest sources from the VCR and the Level Request is a Build.

Other differences between Build and Rebuild are indicated in the table describing the action flow.


1. Create Level Request
1. Scheduler initiates Level Request
2. Pre/Post - Notifications
3. Pre/Post - Approvals
Reject
4. Set Level Request Status to REJECT
End
Approve
5. Retrieve Code
F
6. Set Build Status to CANCEL
Set Level Request Status to FAIL
End
S
7. Set Build Status to READY
8. Transport Source
F
S
9. Set Build Status to FAIL
11. Verify Build Script
F
S
10. Set Level Request Status to FAIL
12. Execute Build Script
F
S
End
13. Transport Deploy Script (fail is not blocking)
S / F
14. Compress Build
F
S
15. Archive Build
F
S
16. Clean up Source (fail is not blocking)
S / F
17. Clean up Target (fail is not blocking)
S / F
18. Set Build Status:
- SUCCESS: all actions successful
- WARNING: at least one non-blocking action failed
User actions
Monitor actions
Builder actions
Deployer actions
Success S
Fail F
19. Build on latest sources ?
No
Yes
20. Tag code
F
21. Set Level Request Status to FAIL
S
End
22. Clean up Work Copy (fail is not blocking)
23. Set Level Request Status
- SUCCESS:
- Status all Monitor Actions = SUCCESS
- Build Status = SUCCESS
- WARNING:
- Status of at least one non-blocking Monitor action = FAIL
AND/OR
- Build Status = WARNING
24. Notification
End

Step	Description
1.	A Level Request is created manually by the User (via the Web Interface (page 45) or the Command Line (page 475)) or automatically by the Scheduler. A Build Level Request directly goes on to step 5; a Rebuild Level Request may pass steps 2, 3 and/or 4.
2.	The Monitor Process on the IKAN ALM Server picks up the created Level Request and sends the required Pre- and Post-Notifications. This is an optional step in case of a Rebuild, since there are no Pre- or Post Notifications on a Build Level. If required, the Pre- and Post-Approval groups are defined on the Level Settings screen (page 205).
3.	The Monitor generates the required Pre- and Post-Approvals. This is an optional step in case of a Rebuild, since there are no Pre- or Post Notifications on a Build Level. If required, the Pre- and Post-Approval groups are defined on the Level Settings screen (page 205).
4.	As soon as one of the Approvals is rejected (page 24), the Monitor sets the Level Request status to *Reject* and the action flow is terminated. This step is optional in case of a Rebuild.
5.	If all Approvals of a Rebuild with Pre- or Post-Approval(s) are granted (page 23), and if the requested Date/Time is reached, the Monitor sets the Level Request status to *Run* and retrieves the Source Code from the VCR to a subdirectory of the Work Copy location on the IKAN ALM Server. This location is defined in the System Settings (page 255). If the Project Stream in which this Build is done, is Master for one or more Child Project Streams, the Sources or the Build Result (depending on the Dependency Type) of these Child Project Streams will also be retrieved from the VCR, respectively from the Build Archive, to the Work Copy location.
6.	If the Retrieval process fails, the Monitor sets the Build Status of all Builds related to the Level Request to *Cancel*, the Level Request Status to *Fail* and the action flow is terminated.
7.	If the Retrieval process succeeds, the Monitor sets the Build Status of all Builds related to the Level Request to *Ready*. Since a (Re)Build Level Request may have more than one Build, steps 8 till 18 will be performed for each Build related to the Level Request.
8.	As the Build Status is set to *Ready*, the Builder Agent on the Machine(s) where a Build must be executed, picks up this *Ready* status and transports the Source Code from the Work Copy Location on the IKAN ALM Server to the Build Environment Source Location, defined on this Machine and for this Build Level. Which transport action (local FileCopy, remote FileCopy, SecureCopy or FTP) will be used depends on the type of Transporter that is linked to the Machine containing the Build Environment. Refer to the section Build Environments (page 209). If this process fails, steps 9 and 10 are performed. If this process succeeds, step 11 is performed.
9.	The Builder sets the Build Status to *Fail*.
10.	If the Monitor picks up the *Fail* Build Status, it sets the Level Request status to *Fail* as well and the action flow is terminated.

Step	Description
11.	If the Source Transport process succeeds, the Builder Agent verifies the Build Script. This process comprises two phases. **In the first phase**, the Builder Agent determines which Build Script must be used. If a specific Build Script was defined for the Build Environment (page 209), the Builder assumes it must locate and use this Build Script. If no specific Build Script was defined for the Build Environment, the Builder assumes it must locate and use the Build Script defined in the Project Settings (page 148). **In the second phase**, the Builder tries to locate the Build Script it has determined it must use during the first phase. First the Builder searches checked-out source code available in the Source Location of the Build Environment. If the Build Script is found, the Verify Build Script process succeeds and step 12 will be performed. If the Build Script is not found in the checked-out source code, the Builder searches the default IKAN ALM Script Location as defined in the System Settings (page 255). If the Build Script is found, it will be transported to the Build Environment Source Location defined on this Machine and for this Build Level. The same transport mechanism will be used as for the Source Code. The Verify Build Script process succeeds and step 12 will be performed. If the Build Script is not found here either, or if the transport process from the IKAN ALM Script Location does not succeed, the Verify Build Script action fails and steps 9 and 10 are performed.
12.	If the Build Script Verification process succeeds, the Builder Agent executes the Build Script. First, the Build Script is provided with the following parameters: Build Script Location, Source Location, Target Location, standard IKAN ALM parameters and user-defined Build Parameters. Then, the defined Build Tool for the Build Environment (Ant, Gradle, NAnt or Maven2) generates the Build artifacts (e.g., executables, libraries, …). The Build Script should include a copy mechanism that transfers minimum one Build artifact to the Target Location of the Build Environment. Only the Build artifacts in the Target Location will be available if the Build Result must be deployed later on. If this operation fails, steps 9 and 10 are performed. If this operation succeeds, step 13 is performed.
13.	If the Build Script is executed successfully, the Builder Agent checks if the Build result will be deployed on Levels higher up in the Lifecycle. This is the case when a Deploy Environment of such a Level is linked to the Build Environment on which this Build is executed. If that is the case, the Builder Agent will try to add the Deploy Script to the Build result. If a Deploy Script is available on the Build Source Location (as retrieved together with the Source Code from the VCR), this Deploy Script is copied to the Build Target Location. As the failure of this step is not blocking, step 14 is performed next, whether the operation was successful or not.
14.	The Builder Agent compresses the Build artifacts on the Build Environment Target Location. Depending on the Operating System of the IKAN ALM server holding the Build Archive, a *.zip or *.tar.tgz will be created. If this operation fails, steps 9 and 10 are performed. If this operation succeeds, step 15 is performed.
15.	The Builder Agent archives the Compressed Build to the Build Archive on the IKAN ALM Server. The Build Archive Location on the IKAN ALM Server is defined in the System Settings (page 255). Which transport action (local FileCopy, remote FileCopy, SecureCopy or FTP) will be used depends on the type of Transporter that is linked to the Machine containing the Build Environment. If this operation fails, steps 9 and 10 are performed. If this operation succeeds, step 16 is performed.

Step	Description
16.	The Builder Agent cleans up the Source Location on the Build Environment. This means that all files used to create the Build Result (Source files, Build Script and possibly Deploy Script) are deleted. If the Debug option is activated for a Build Environment, the Source Cleanup action will not be performed, so that the User may use the available sources to run the Build Script manually for testing purposes As the failure of this step is not blocking, step 17 is performed next, whether the operation was successful or not.
17.	The Builder Agent cleans up the Target Location on the Build Environment. This means that all available files (the uncompressed and compressed Build Result as well as the Deploy Script) are deleted. If Debug is activated for a Build Environment, the Target Cleanup action will not be performed, so that the user may inspect the Build Result on the Build Environment. As the failure of this step is not blocking, step 18 is performed next, whether the operation was successful or not.
18.	The Build Agent sets the Build Status. If all Builder actions (steps 8, 11, 12, 13, 14, 15, 16 and 17) were executed successfully, the Build Status will be set to *Success*. If Builder actions 13, 16 and/or 17 failed, the Build will be set to *Warning*. (If another action failed, the Build Status will be set to Fail as indicated by steps 9 and 10)
19.	The Monitor checks if the status of all Builds related to the Level Request have been set to *Success* or *Warning*. Then it verifies if the Build has been executed on the latest sources from the VCR, which is typical for the first Build Level in the Lifecycle of a Project Stream. In this case step 20 will be performed next. Otherwise, the Build has been executed on Code that was tagged before, and the Build is most likely a *Rebuild based on tagged code*, generated on the Build Environment belonging to a Test or Production Level. An exception is the Build on a Build Level in a Tag-based Project Stream: although it is not a Rebuild, this Build will always be executed on sources that have been tagged by the user before. In this case step 20 will also be skipped and the next step will be step 22.
20.	The Monitor tags the code in the VCR if the Build was executed on the latest sources.
21.	If the Tagging Process fails, the Monitor will set the Level Request status to *Fail* and the action flow is terminated.
22.	If the Tagging Process is successful or if it was skipped because the Code was already tagged, the Monitor cleans up the used subdirectories in the Work Copy Location on the IKAN ALM Server. This means that all files retrieved from the VCR or from the Build Archive are deleted. As the failure of this step is not blocking, step 23 is performed next, whether the operation was successful or not.
23.	The Monitor Process on the IKAN ALM Server determines the final Level Request status. The final Level Request Status is set to *Success*, if all Monitor actions (in yellow) were executed successfully and the Build Status has been set to *Success*. The final Level Request Status is set to *Warning*, if at least one non-blocking Monitor Action failed and/or the Build Status has been set to *Warning*.
24.	The required Notifications are sent. All Users belonging to the User Group with User Access Rights or the User Group with Admin Access Rights (both defined on the Project (page 461) screen) receive the required notifications, as well as the Users having Request Rights on the Level. The Notification type (mail, netsend) and the Notification criteria (if Level Request Status is SUCCESS, FAIL, WARNING or ALWAYS) are defined in the sections Creating a Build Level (page 188), Creating a Test or Production Level (page 190) or Editing a Level (page 194).

(Re)Build and Deploy Level Requests

The following graphic displays the action flow of a Build and Deploy Level Request or a Rebuild and Deploy Level Request.

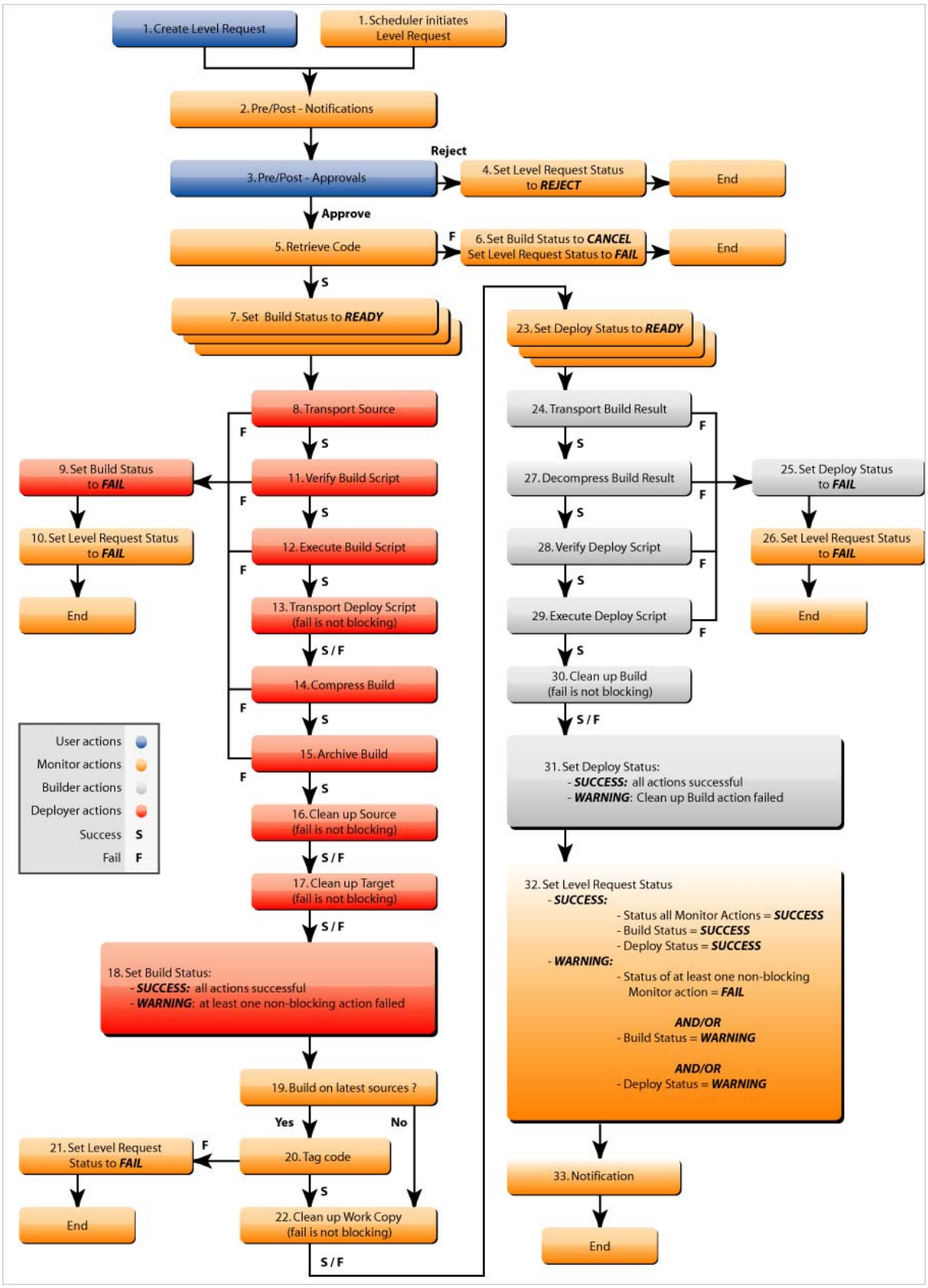
The (Re)Build and Deploy Level Request may be created on any Level in the Lifecycle, e.g., as a Build and Deploy Level Request on the (first) Build Level that has a Deploy Environment to directly deploy the Build Result of the latest sources for integration testing, or on a QA Test Level which is very similar to the Production Level, and where a Rebuild is done so that the Build Result may be deployed later on to a Production Level.

The following section describes the default Action Flow. If the Level Phases (page 197), the Build Environment Phases (page 219) or the Deploy Environment Phases (page 234) have been modified, the sequence of actions may be different.

Note: The main difference between a Build and a Rebuild is step 19.

If the answer is *No*, the build was not done with the latest sources, but with previous tagged sources and the Level Request is a Rebuild. If the answer is *Yes*, the build was done with the latest sources from the VCR and the Level Request is a Build.

Other differences between Build and Rebuild are indicated in the table describing the action flow.


1. Create Level Request
1. Scheduler initiates Level Request
2. Pre/Post - Notifications
3. Pre/Post - Approvals
Reject
4. Set Level Request Status to REJECT
End
Approve
5. Retrieve Code
F
6. Set Build Status to CANCEL
Set Level Request Status to FAIL
End
S
7. Set Build Status to READY
8. Transport Source
F
S
9. Set Build Status to FAIL
11. Verify Build Script
F
S
10. Set Level Request Status to FAIL
12. Execute Build Script
F
S
End
13. Transport Deploy Script (fail is not blocking)
S / F
14. Compress Build
F
S
15. Archive Build
F
S
16. Clean up Source (fail is not blocking)
S / F
17. Clean up Target (fail is not blocking)
S / F
18. Set Build Status:
- SUCCESS: all actions successful
- WARNING: at least one non-blocking action failed
19. Build on latest sources ?
Yes
No
21. Set Level Request Status to FAIL
F
20. Tag code
S
End
22. Clean up Work Copy (fail is not blocking)
S / F
User actions
Monitor actions
Builder actions
Deployer actions
Success S
Fail F
23. Set Deploy Status to READY
24. Transport Build Result
F
S
27. Decompress Build Result
F
S
25. Set Deploy Status to FAIL
28. Verify Deploy Script
F
S
26. Set Level Request Status to FAIL
29. Execute Deploy Script
F
S
End
30. Clean up Build (fail is not blocking)
S / F
31. Set Deploy Status:
- SUCCESS: all actions successful
- WARNING: Clean up Build action failed
32. Set Level Request Status
- SUCCESS:
- Status all Monitor Actions = SUCCESS
- Build Status = SUCCESS
- Deploy Status = SUCCESS
- WARNING:
- Status of at least one non-blocking Monitor action = FAIL
AND/OR
- Build Status = WARNING
AND/OR
- Deploy Status = WARNING
33. Notification
End

Step	Description
1.	A Level Request is created manually by the User (via the Web Interface (page 45) or the Command Line (page 475)) or automatically by the Scheduler. A Build Level Request directly goes on to step 5, a Rebuild Level Request may pass steps 2,3 and/or 4.
2.	The Monitor Process on the IKAN ALM Server picks up the created Level Request and sends the required Pre- and Post-Notifications. This is an optional step in case of a Rebuild, since there are no Pre- or Post Notifications on a Build Level. If required, the Pre- and Post-Notification groups are defined on the Level Settings screen (page 205).
3.	The Monitor generates the required Pre- and Post-Approvals. This is an optional step in case of a Rebuild, since there are no Pre- or Post Notifications on a Build Level. If required, the Pre- and Post-Approval groups are defined on the Level Settings screen (page 205)).
4.	As soon as one of the Approvals is rejected (page 24), the Monitor sets the Level Request status to *Reject* and the action flow is terminated. This step is optional in case of a Rebuild
5.	If all Approvals of a Rebuild with Pre- or Post-Approval(s) are granted (page 23) and if the requested Date/Time is reached, the Monitor sets the Level Request status to *Run* and retrieves the Source Code from the VCR to a subdirectory of the Work Copy location on the IKAN ALM Server. This location is defined in the System Settings (page 255). If the Project Stream in which this Build is done, is Master for one or more Child Project Streams, the Sources or the Build Result (depending on the Dependency Type) of these Child Project Streams will also be retrieved from the VCR, respectively from the Build Archive, to the Work Copy location.
6.	If the Retrieval process fails, the Monitor sets the Build and Deploy Status of all Builds and Deploys related to the Level Request to *Cancel*, the Level Request Status to *Fail* and the action flow is terminated.
7.	If the Retrieval process succeeds, the Monitor sets the Build Status of all Builds related to the Level Request to *Ready*. Since a (Re)Build and Deploy Level Request may have more than one Build, steps 8 till 18 will be performed for each Build related to the Level Request.
8.	As the Build Status is set to *Ready*, the Builder Agent on the Machine(s) where a Build must be executed, picks up this *Ready* status and transports the Source code from the Work Copy Location on the IKAN ALM Server to the Build Environment Source Location, defined on this Machine and for this Build Level. Which transport action (local FileCopy, remote FileCopy, SecureCopy or FTP) will be used depends on the type of Transporter that is linked to the Machine containing the Build Environment. Refer to the section Build Environments (page 209). If this process fails, steps 9 and 10 are performed. If this process succeeds, step 11 is performed.
9.	The Builder sets the Build Status to *Fail*.
10.	If the Monitor picks up the *Fail* Build Status, it sets the Level Request status to *Fail* as well, and the action flow is terminated.

Step	Description
11.	If the Source Transport process succeeds, the Builder Agent verifies the Build Script. This process comprises two phases. **In the first phase**, the Builder Agent determines which Build Script must be used. If a specific Build Script was defined for the _Build Environment (page 209). The Builder assumes it must locate and use this Build Script. If no specific Build Script was defined for the Build Environment, the Builder assumes it must locate and use the Build Script defined on the _Project Settings (page 146) screen. **In the second phase**, the Builder tries to locate the Build Script it has determined it must use in the first phase. First it searches in the checked out source code available in the Source Location of the Build Environment. If the Build Script is found, the Verify Build Script process succeeds and step 12 will be performed. If the Build Script is not found in the checked out source code, the Builder searches the default IKAN ALM Script Location as defined in the _System Settings (page 255). If the Build Script is found, it will be transported to the Build Environment Source Location defined on this Machine and for this Build Level. The same transport mechanism will be used as for the Source Code. The Verify Build Script process succeeds and step 12 will be performed. If the Build Script is not found here either, or if the transport process from the IKAN ALM Script Location does not succeed, the Verify Build Script action fails and steps 9 and 10 are performed.
12.	If the Build Script Verification process succeeds, the Builder Agent executes the Build Script. First, the Build Script is provided with the following parameters: Build Script Location, Source Location, Target Location, standard IKAN ALM parameters and user-defined Build Parameters. Then the defined Build Tool for the Build Environment (Ant, Gradle, NAnt or Maven2) generates the Build (e.g., executables, libraries,…). The Build Script should include a copy mechanism that transfers minimum one Build artifact to the Target Location of the Build Environment. Only the Build artifacts in the Target Location will be available if the Build Result must be deployed later on. If this operation fails, steps 9 and 10 are performed. If this operation succeeds, step 13 is performed.
13.	If the Build Script execution is successful, the Builder Agent checks if the Build result will be deployed in this Level (most likely, since it is a (Re)Build and Deploy Level Request), or on Levels higher up in the Lifecycle. This is the case when a Deploy Environment of this Level or of a higher Level is linked to the Build Environment on which this Build is executed. If that is the case, the Builder Agent will try to add the Deploy Script to the Build result. If a Deploy Script is available on the Build Source Location (as retrieved together with the Source Code from the VCR), this Deploy Script is copied to the Build Target Location. As the failure of this step is not blocking, step 14 is performed next, whether the operation was successful or not
14.	The Builder Agent compresses the Build artifacts on the Build Environment Target Location. Depending on the Operating System of the IKAN ALM Server holding the Build Archive, a *.zip or *.tar.tgz file will be created. If this operation fails, steps 9 and 10 are performed. If this operation succeeds, step 15 is performed.
15.	The Builder Agent archives the compressed Build to the Build Archive on the IKAN ALM Server. The Build Archive Location on the IKAN ALM Server is defined in the _System Settings (page 255). Which transport action (local FileCopy, remote FileCopy, SecureCopy or FTP) will be used depends on the type of Transporter that is linked to the Machine containing the Build Environment. If this operation fails, steps 9 and 10 are performed. If this operation succeeds, step 16 is performed.

Step	Description
16.	The Builder Agent cleans up the Source Location on the Build Environment. This means that all files used to create the Build result (Source files, Build Script and possibly Deploy Script) are deleted. If the Debug option is activated for a Build Environment, the Source Cleanup action will not be performed, so that the User may use the available sources to run the Build Script manually for testing purposes. As the failure of this step is not blocking, step 17 is performed next, whether the operation was successful or not.
17.	The Builder Agent cleans up the Target Location on the Build Environment. This means that all available files (uncompressed and compressed Build Result as well as the Deploy Script) are deleted. If Debug is activated for a Build Environment, the Target Cleanup action will not be performed, so that the user may inspect the Build Result on the Build Environment. As the failure of this step is not blocking, step 18 is performed next, whether the operation was successful or not.
18.	The Build Agent sets the Build Status. If all Builder actions (steps 8, 11, 12, 13, 14, 15, 16 and 17) were executed successfully, the Build Status will be set to *Success*. If Builder actions 13, 16 and/or 17 failed, the Build Status will be set to *Warning*.(If another action failed, the Build Status will be set to *Fail* as indicated by steps 9 and 10).
19.	The Monitor checks if the status of all Builds related to the Level Request have been set to *Success* or *Warning*. Then it verifies if the Build has been executed on the latest sources from the VCR, which is typical for the first Build Level in the Lifecycle of a Project Stream. In this case, step 20 will be performed next. If the Build has been executed on Code that was ALREADY tagged, the Build is most likely a Rebuild based on tagged code, generated on the Build Environment belonging to a Test or Production Level. An exception is the Build on a Build Level in a Tag-based Project Stream: although it is not a Rebuild, this Build will always be executed on sources that have been tagged by the user before. In this case step 20 will also be skipped and the next step will be step 22.
20.	If the Build was executed on the latest sources, the Monitor tags the code in the VCR.
21.	If the Tagging Process fails, the Monitor will set the Level Request status to Fail and the action flow is terminated.
22.	If the Tagging Process is successful or if it was skipped because the Code was already tagged, the Monitor cleans up the used subdirectories of the Work Copy on the IKAN ALM Server. This means that all files retrieved from the VCR are deleted. As failure of this step is not blocking, step 23 is performed next, whether the operation was successful or not.
23.	The Monitor sets the Deploy Status of all Deploy actions to *Ready*.
24.	As the Deploy Status is set to *Ready*, the Deploy Agent on the Machine(s) where a Build must be deployed, picks up this *Ready* status. It then transports the compressed Build Result from the Build Archive to the Deploy Environment Source Location for this Level. Which transport action (local FileCopy, remote FileCopy, SecureCopy or FTP) will be used depends on the type of Transporter that is linked to the Machine containing the Deploy Environment. Refer to the section Deploy Environments (page 225). If this process fails, steps 25 and 26 are performed. If this process succeeds, step 27 is performed.
25.	The Deployer sets the Deploy Status to *Fail*.
26.	If the Monitor picks up one Deploy with *Fail* Status, it sets the Level Request status to *Fail* as well and the action flow is terminated.

Step	Description
27.	If the Transport Build Result process succeeds, the Deployer Agent decompresses the Build file (containing the result from a preceding Build action and the Deploy script) into the Deploy Environment Source Location. (This is the same location as to which the compressed Build was transported.) If this process fails, steps 25 and 26 are performed. If this process succeeds, step 28 is performed.
28.	If the Decompress process succeeds, the Deployer Agent verifies the Deploy Script. This process comprises two phases. **In the first phase**, the Deployer Agent determines which Deploy Script it must use. First, it verifies if a specific Deploy Script was defined for the Deploy Environment (Deploy Environments (page 225)). If this is the case, the Deployer assumes it must locate and use this Deploy Script. If no specific Deploy Script was defined for the Deploy Environment, the Deployer assumes it must locate and use the Deploy Script defined on the Project Settings (page 146) screen. If no Deploy Script was defined there either, the Verify Deploy Script process fails and steps 25 and 26 are performed. **In the second phase**, the Deployer tries to locate the Deploy Script it has determined it must use in the first step. First it searches the decompressed Build Result in the Source location of the Deploy Environment (Deploy Environments (page 225)). If the Deploy Script is found, the Verify Deploy Script process succeeds and step 29 will be performed. If the Deploy Script is not found in the decompressed Build result, the Deployer searches the default IKAN ALM Script Location as defined in the System Settings (page 255). If the Deploy Script is found, it will be transported to the Deploy Environment Source Location, defined on this Machine and for this Level. The same transport mechanism will be used as for the Build Result. The Verify Deploy Script process succeeds and step 29 will be performed. If the Build Script is not found here either or the transport from the IKAN ALM Script Location does not succeed, the Verify Deploy Script action fails and steps 25 and 26 are performed.
29.	If the Verify Deploy Script process succeeds, the Deployer Agent executes the Deploy Script. The Deploy Script is provided with the following parameters: Source Location, Target Location, standard IKAN ALM Parameters and user-defined Deploy Parameters. The defined Deploy Tool (Ant, Gradle, NAnt or Maven2) for the Deploy Environment deploys the Build to the Target Location. If this operation fails, steps 25 and 26 are performed. If this operation succeeds, step 30 is performed.
30.	If the Deploy is successful, the Deploy Agent cleans up the Build Result on the Deploy Environment Source Location for this Level. If the Debug option is activated for a Deploy Environment, the Clean-up Build Result action will not be performed, so that the user may use the available Build Result to run the deploy script manually for testing purposes. As failure of this step is not blocking, step 31 is performed next, whether the operation was successful or not.
31.	The Deploy Agent sets the Deploy Status. If all Deployer actions (steps 24, 27, 28, 29 and 30) were executed successfully, the Deploy Status will be set to *Success*. If Deployer action 30 (Clean up Build) failed, the Deploy Status will be set to *Warning*. (If another action failed, the Deploy Status will be set to *Fail* as indicated by steps 25 and 26).

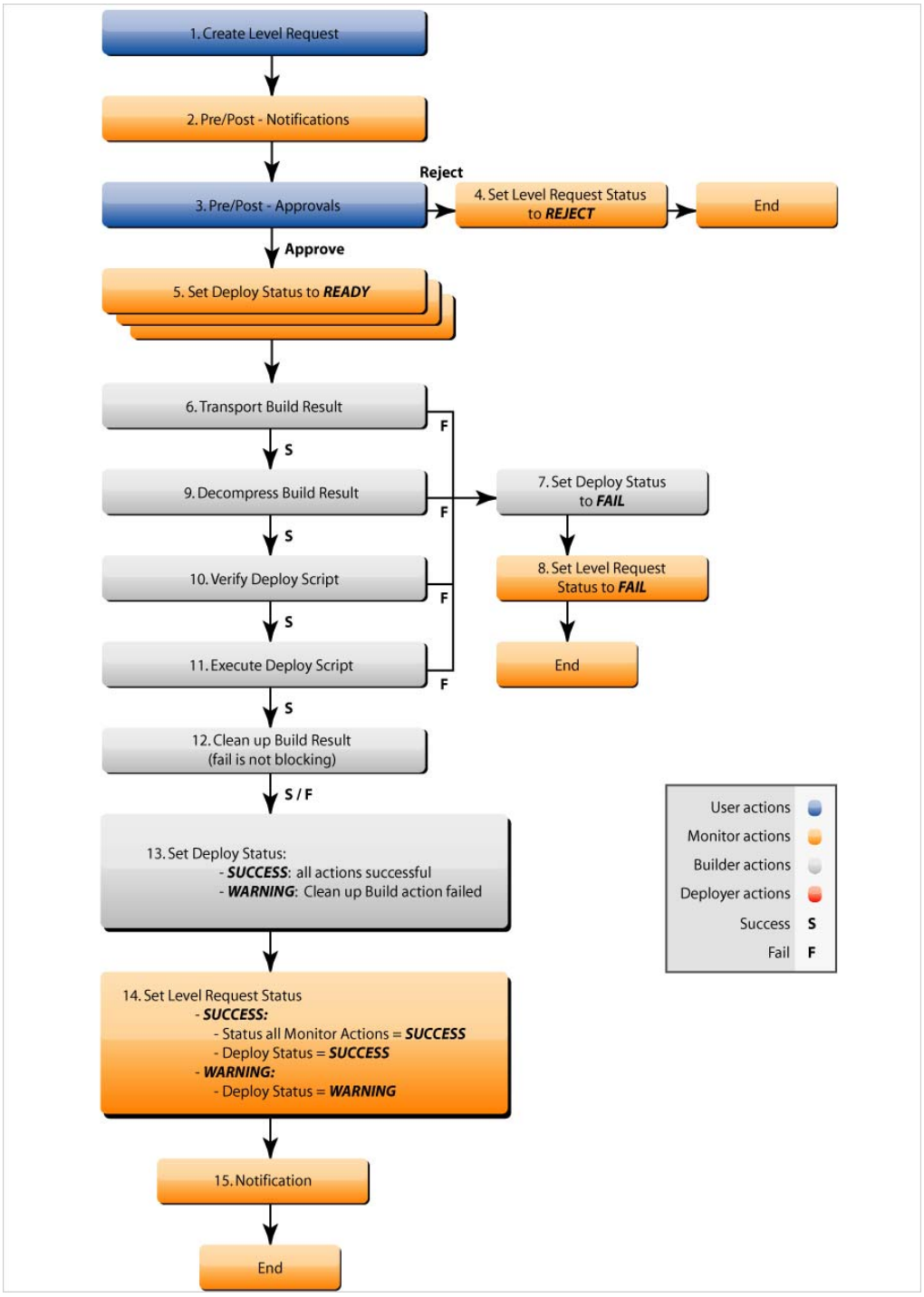
Step	Description
32.	As soon as the Monitor Process on the IKAN ALM Server finds that all Deploy actions connected to a Level Request have the *Success* or *Warning* Status, it determines the final Level Request status. The final Level Request Status is set to *Success*, if all Monitor actions (in yellow) were executed successfully and both the Build Statuses and the Deploy Statuses have been set to *Success*. The final Level Request Status is set to *Warning*, if at least one non-blocking Monitor action *failed* and/or the Build Statuses and/or the Deploy Statuses have been set to *Warning*.
33.	The required Notifications are sent. All Users belonging to the User Group with User Access Rights or the User Group with Admin Access Rights (both defined on the _Project (page 461) screen) receive the required notifications, together with the users that have Request Rights on the Level. The Notification type (mail, netsend) and the Notification criteria (if Level Request Status is SUCCESS, FAIL, WARNING or ALWAYS) are defined in the sections Creating a Build Level (page 188), Creating a Test or Production Level (page 190) or Editing a Level (page 194).

Deploy Level Requests

The following graphic displays the action flow of a Deploy Level Request.

The Deploy Level Request is always executed for a Level after the Build Level in a Lifecycle, e.g., a QA Test Level which is very similar to the Production Level, or the Production Level itself. Most often, such a Level has one or more Deploy Environments, and will reuse the Build Result that has been created on Levels with a Build Environment earlier in the Lifecycle.

Note: The following section describes the default Action Flow.
If the Level Phases (page 197) or the Deploy Environment Phases (page 234) have been modified, the sequence of actions may be different.



Step	Description
1.	A Level Request is created manually by the User (via the _Web Interface (page 45) or the _Command Line (page 475)) or automatically by the Scheduler.
2.	The Monitor Process on the IKAN ALM Server picks up the Level Request and optionally sends the required Pre- and Post-Notifications. The Pre- and Post-Notification groups are optionally defined on the _Level Settings screen (page 205).
3.	The Monitor optionally generates the required Pre- and Post-Approvals. The Pre- and Post-Approval groups are optionally defined on the _Level Settings screen (page 205).
4.	As soon as one of the Approvals is _rejected (page 24), the Monitor sets the Level Request status to *Reject* and the action flow is terminated.
5.	When the Level Request is _approved (page 23), and the requested Date/Time is reached, the Monitor sets the Level Request status to *Run*. Then the Monitor sets the Deploy Status of all Deploys related to the Level Request to *Ready*. Since a Deploy Level Request may have more than one Deploy, steps 6 till 10 will be performed for each Deploy related to the Level Request.
6.	As the Deploy Status is set to *Ready*, the Deploy Agent on the Machine(s) where a Build must be deployed, picks up this *Ready* status. It then transports the compressed Build Result from the Work Copy Location to the Deploy Environment Source Location defined on this Machine and for this Level. Which transport action (local FileCopy, remote FileCopy, SecureCopy or FTP) will be used depends on the type of Transporter that is linked to the Machine containing the Deploy Environment. Refer to the section Deploy Environments (page 225). If this process fails, steps 7 and 8 are performed. If this process succeeds, step 9 is performed.
7.	The Deployer sets the Deploy Status to *Fail*.
8.	If the Monitor picks up the Fail Deploy Status, it sets the Level Request status to *Fail* as well and the action flow is terminated.
9.	If the Transport Build Result succeeds, the Deployer Agent decompresses the Build File into the Deploy Environment Source Location. (This is the same location as the location to which the compressed Build was transported.) If this process fails, steps 7 and 8 are performed. If this process succeeds, step 10 is performed.

Step	Description
10.	If the Decompress process succeeds, the Deployer Agent verifies the Deploy Script. This process comprises two phases. **In the first phase**, the Deployer Agent determines which Deploy Script it must use. First, it verifies if a specific Deploy Script was defined for the Deploy Environment (page 225). If this is the case, the Deployer assumes it must locate and use this Deploy Script. If no specific Deploy Script was defined for the Deploy Environment, the Deployer assumes it must locate and use the Deploy Script defined on the Project Settings (page 146) screen. **In the second phase**, the Deployer tries to locate the Deploy Script it has determined it must use during the first phase. First it searches the decompressed Build Result in the Source location of the Deploy Environment. If the Deploy Script is found, the Verify Deploy Script process succeeds and step 11 will be performed. If the Deploy Script is not found in the decompressed Build result, the Deployer searches the default IKAN ALM Script Location as defined in the System Settings (page 255). If the Deploy Script is found it will be transported to the Deploy Environment Source Location, defined on this Machine and for this Level. The same transport mechanism will be used as for the Build Result. The Verify Deploy Script process succeeds and step 11 will be performed. If the Deploy Script is not found here either or the transport from the IKAN ALM Script Location does not succeed, the Verify Deploy Script action fails and steps 7 and 8 are performed
11.	If the Verify Deploy Script process succeeds, the Deployer Agent executes the Deploy Script. First the Deploy Script is provided with the following parameters: Source Location, Target Location, standard IKAN ALM Parameters and user-defined Deploy Parameters. The defined Deploy Tool for the Deploy Environment (Ant, Gradle, NAnt or Maven2) deploys the Build to the Target Location. If this operation fails, steps 7 and 8 are performed. If this operation succeeds, step 12 is performed.
12.	If the Deploy is successful, the Deploy Agent cleans up the Build Result on the Deploy Environment Source Location for this Level. If the Debug option is activated for a Deploy Environment, the Clean-up Build Result action will not be performed, so that the user may use the available Build Result to run the deploy script manually for testing purposes. As the failure of this step is not blocking, step 13 is performed next, whether the operation was successful or not.
13.	The Deploy Agent sets the Deploy Status. If all Deployer actions (steps 6, 9, 10, 11 and 12) were executed successfully, the Deploy Status is set to Success. If Deployer action 12 (Clean up Build) failed, the Deploy Status will be set to Warning. (If another action failed, the Deploy Status will be set to Fail as indicated by steps 7 and 8).
14.	As soon as the Monitor Process on the IKAN ALM Server detects a Deploy with Deploy Status *Success* or *Warning*, it determines the final Level Request status. The final Level Request Status is set to *Success*, if all Monitor actions (in yellow) were executed successfully and the Deploy Status has been set to *Success*. The final Level Request Status is set to *Warning*, if the Deploy Status has been set to *Warning*.
15.	The required Notifications are sent. All Users belonging to the User Group with User Access Rights or the User Group with Admin Access Rights (both defined on the Project (page 461) screen) receive the required notifications, together with the users that have Request Rights on the Level. The Notification type (mail, netsend) and the Notification criteria (if Level Request Status is SUCCESS, FAIL, WARNING or ALWAYS) are defined in the sections Creating a Build Level (page 188), Creating a Test or Production Level (page 190) or Editing a Level (page 194).

7.2. Creating Level Requests

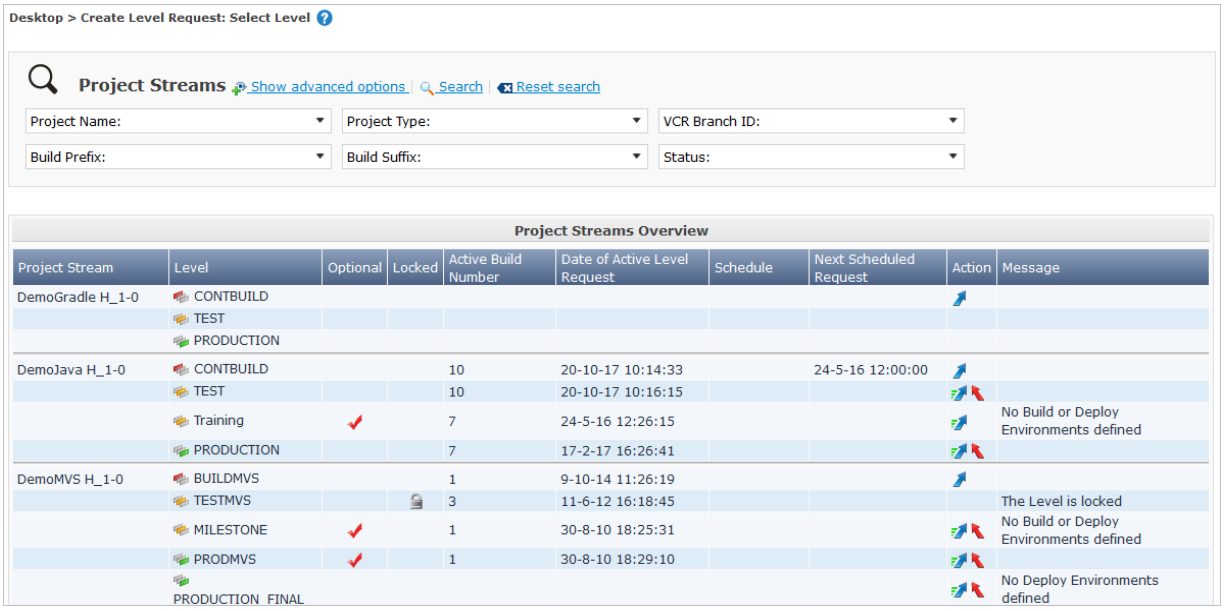
The following sections deal with the procedures involved when creating new Level Requests:

- The Create Level Request: Select Level Screen (page 45)
- Creating a Build Level Request (page 48)
- Creating a Deliver Build Level Request (page 52)
- Creating a Rollback Build Level Request (page 55)

Note: If you often need to create Level Requests for specific Levels, you can assign them to one of your Desktop Tab Pages. See Adding Elements to a Desktop Tab Page on page 140.
This way, you will be able to easily create Level Requests by simply clicking an icon.

The Create Level Request: Select Level Screen

1. Select *Level Requests > Create Level Request* on the Main Menu.
   The *Create Level Request: Select Level* screen is displayed:



2. Define search criteria on the *Search Project Stream* panel.
   Level Requests are always defined for a Project Stream.
   If you do not immediately find the required Project Stream on the Overview, define search criteria for Projects and/or Project Streams in the *Search Project Stream* panel.

3. Verify the information on the *Project Streams Overview* screen.
   The Project Streams and Levels matching the search criteria, are displayed below the *Search Project Stream* panel. If no search criteria were defined, all available Levels and Project Streams will be displayed.
   The following information is available for each displayed Level.

Field	Description
Project Stream	This field contains the identification of the Project Stream. This name is composed of: • Project Name • Project Stream Type: H (Head) or B (Branch) • Project Stream Prefix, optionally followed by the Suffix in case of a Branch Project Stream Example: `Webpad H_1-0`
Level	This field contains the name and type of the Level. There are three Level Types: • Build • Test • Production See the sections Creating a Build Level (page 188) and Creating a Test or Production Level (page 190).
Optional	This field indicates whether or not the Level is optional in the Lifecycle attached to the Project Stream. • If the Level is optional, the field is marked by a red cross. • If the Level is not optional, this field is empty. See Lifecycles Overview Screen on page 176.
Locked	This field indicates whether or not the Level is locked: • If the Level is locked, the field contains a red check mark. • If the Level is not locked, the field is empty. It is not possible to execute Level Requests on locked Levels. Levels can be unlocked by auditing the Project. See Auditing Projects on page 247.
Active Build Number	This field contains the number of the Active Build on this Level.
Date of Active Level Request	This field indicates the date and time at which the latest successful Level Request was executed on this Level.
Schedule	This field is only applicable on a Build Level. It contains the name of the Schedule associated with this Level. The Schedule defines the frequency of the Continuous Build process as a number of seconds, minutes or days. See Schedules on page 452. If no Schedule was assigned to a particular Level, the field remains empty.
Next Scheduled Request	If a Schedule was assigned to the Level, this field contains the execution date and time of the next scheduled Level Request, under the condition that there are changes in the connected VCR.

4. In the *Action* column, click the required Level Request Creation icon.

The following icons may be available:

Icon	Level Request Type	Description
	Request/Force	Click this icon to create a Build Level Request. If no schedule is attached to the Build Level, a build will be *requested*. If a schedule is attached to it, and if the Force Build Option is activated for the Project Stream, a build can be *forced*. See Creating a Build Level Request on page 48.
	Deliver	Click this icon to create a Level Request that will deliver a Build to the selected Test or Production Level. See Creating a Deliver Build Level Request on page 52.
	Rollback	Click this icon to create a Level Request that will restore a previous version of the application on the selected Test or Production Level. See Creating a Rollback Build Level Request on page 55.

The following messages can replace or complete the Level Request Creation Links.

Message	Description
A Level Request is pending for this Level of this Project Stream	This message is displayed, if a Level Request is being executed or waiting for approval. You will need to wait until the current Level Request is completed, before you can define a new Level Request for this Level.
The Level is locked	It is not possible to define Level Requests for Locked Levels. You (or the Project Manager) must audit the Project to unlock the Level (page 247), before you can define Level Requests for this Level.
No Request Rights	It is not possible to define Level Requests, if your User ID does not have the required access rights. This is because your User ID is not a member of the Requester User Group that is protecting the creation of Level Requests on the Level. You must connect with a User ID having the right to run Requests or ask the Global or Project Administrator to give this right to your User ID.
The Project is locked	It is not possible to define Level Requests for locked Projects. A User with Project Admin Access Rights can unlock the Project first by clicking the *Unlock* button on the *Projects Overview*. See Editing Project Settings on page 149.
The Project Stream is locked	It is not possible to define Level Requests for locked Project Streams. A User with Project Admin Access Rights can unlock the Project Stream first by clicking the *Unlock* button on the *Edit Project Stream screen*. See Editing Project Stream Settings on page 167.

Message	Description
The Project Stream is frozen	It is not possible to define Build Level Requests for frozen Project Streams. However, it is still possible to deliver Level Requests to Test and Production Levels. A User with Project Admin Access Rights can unfreeze the Project Stream first by selecting another status from the *Status* drop-down menu on the *Edit Project Stream screen*. See Editing Project Stream Settings on page 167.
No Build Environments defined	It is not possible to define a Build Level Request for a Build Level without a Build Environment. A User with Project Admin Access Rights can create a Build Environment for this Level
No Build or Deploy Environments defined	This field indicates that the Test Level is not associated to a Build or Deploy Environment. This is a warning message, indicating that there will be no deploy Action when creating a Deliver or Rollback Level Request for this Level. However, such Levels have the same Approval and Notification management options as Levels that are linked to Environments.
No Deploy Environments defined	This field indicates that the Production Level is not associated to a Deploy Environment. This is a warning message, indicating that there will be no deploy Action when creating a Deliver or Rollback Level Request for this Level. However, such Levels have the same Approval and Notification management options as Levels that are linked to Environments.
Forced Builds are not allowed	It is not possible to define manual Build Level Request on Project Streams in case the *Accept Forced Build* attribute is set to "No". See Creating a Build Level Request on page 48.
No Levels defined in the Lifecycle of the Project Stream.	It is not possible to create a Level Request, since there is no Level linked to the Lifecycle of the Project Stream.

RELATED TOPICS

- Users and Groups (page 275)
- Personal Settings (page 15)
- Managing the Desktop (page 135)
- Approvals (page 21)

Creating a Build Level Request

Level Requests are created using the Request/Force Build (/) icons.

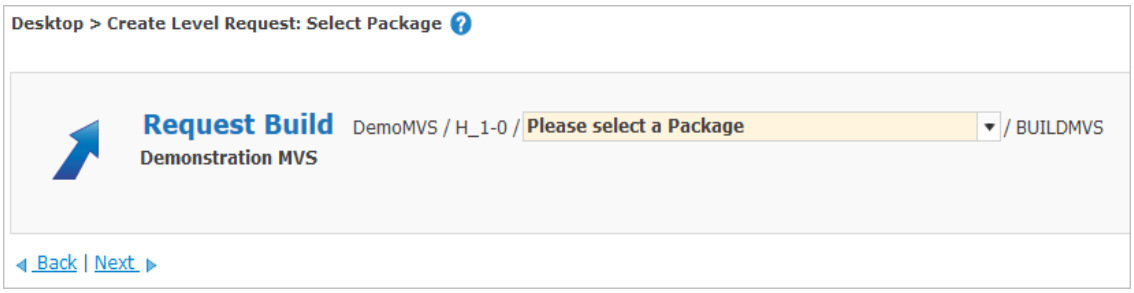
Whether it concerns a Requested or a Forced Build depends on the way the Build Level has been defined.

Build Type	Description
Requested Build ( )	If no schedule is attached to the Build Level, builds will only be generated when created manually. This is called a *Requested Build*.

Build Type	Description
Forced Build ( )	If a Continuous Build Process has been defined for the Build Level by means of a Schedule, and if the Force Build option is activated for the Project Stream concerned, a Build can still be generated manually. This is called a *Forced Build*.

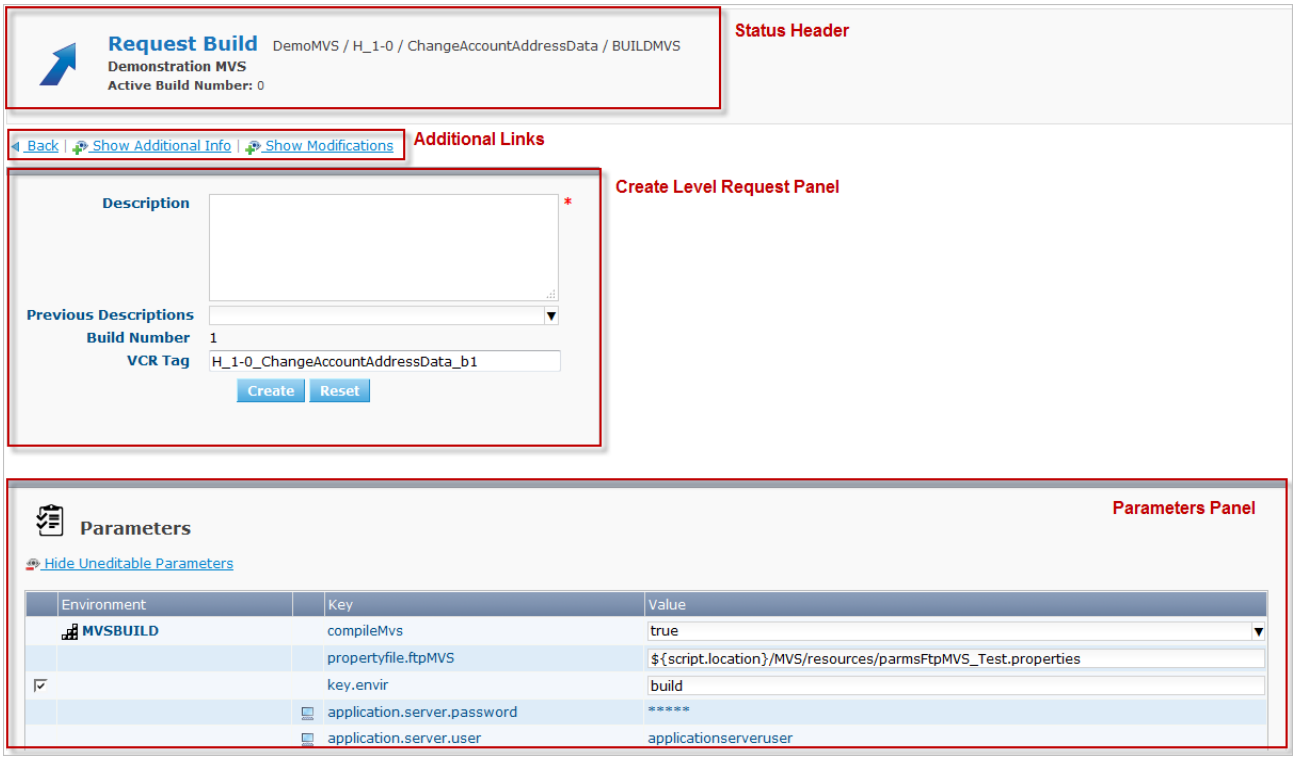
1. Select *Level Requests* > *Create Level Request* on the Main Menu.

2. If the Level belongs to a Package-based Project, you first need to select the required Package.



Note: If you add the selected package to a Desktop Tab Page, this step is avoided when creating a Level Request. See Adding Elements to a Desktop Tab Page on page 140.

3. The *Create Level Request* screen is displayed.



On this screen you will find the following sections:

- The Status Header
- Links for navigation and for showing/hiding panels with extra information
- The *Create Level Request* panel

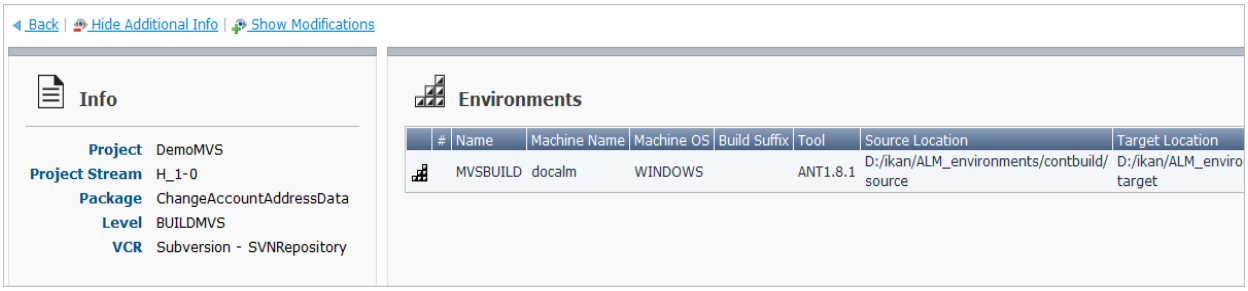
- The optional *Select Deploys to Execute* panel becomes available in case several Deploy Environment have been defined for the Level and if the option *Make Level Optional* is activated for the Level. See also Enabling or Disabling Optional Deploys (page 182).
- The Parameters panel (only available if Parameters are linked to the involved Environments or Machines)

4. Verify the information provided in the Status Header and via the additional links in the upper part of the screen.
   - The Status Header

     The header displays the type of Build Level Request (Force or Request Build) and its corresponding symbol, followed by the context of the Level Request (Project Name/Project Stream Identification[/Package Name]/Level Name), the description of the Project and the active build number.
   - The *Back* link

     Click this link to return to the *Create Level Request: Select Level* page, or the Desktop Page, depending on where you launched the Create Build Level Request.
   - The *Show/Hide Additional Info* link

     Click this link to display or hide information concerning the Project, Project Stream, [Package,] Level and Version Control Repository, as well as information about the Environments linked to the Build Level.



   - The *Show/Hide Modifications* link

     Click this link to display or hide the *Modifications since previous successful Level Request* panel, containing the added, deleted and modified files in the VCR compared with the previous successful Level Request.

5. Complete the fields in the *Create Level Request* panel.

The following fields are available:

Field	Description
Description	In this field, enter a description for the Level Request or select one of the previously entered descriptions.
Previous Descriptions	From the drop-down list, select one of the descriptions you entered previously to automatically fill in the *Description* field.
Build Number	This field contains the next available sequential Build Number for this Level. This number is only indicative, as another Level Request for this Level may be defined almost simultaneously, resulting in a higher Build Number for this Level Request.
VCR Tag	This field contains the VCR Tag that is *likely to be assigned* to the Build resulting from the Level Request, if it is executed successfully. The Tag matches the Tag Template defined for the Head or Branch Stream. In the exceptional event that another Level Request is defined almost simultaneously for this Level, the actual VCR Tag will contain a higher Build Number. The user can override or edit the suggested Tag in order to specially mark the Build. For instance, if the BUILD is a release candidate, he or she might change it to RC_1. Keep in mind that the VCR Tag must be unique in the Project Stream and that it may not contain special characters or spaces depending on the VCR type. **Note:** In the case of Tag-based Builds, this field is left empty. The Tag must be provided by the User. The tag has to match the user-defined tag in the head or branch of the VCR. For more information, refer to the section Creating a Branch Project Stream (page 161).

6. If available, verify and/or edit the settings for the available Build or Deploy Parameters in the *Parameters* panel.

The Parameters will be grouped per Environment linked to the Level.

A Build/Deploy Parameter can have the following characteristics:

- *Mandatory Parameters* will always be provided to the Build/Deploy Script, when the Level Request is executed. Mandatory Parameters lack the activation check box.
- *Non-Mandatory Parameters* can be provided to the Build/Deploy Script, when the Level Request is executed. If you want to provide the Non-Mandatory Parameter, select the check box. If you do not want to provide the Non-Mandatory Parameter, clear the check box.
- *Editable Parameters* have a default value, but you can change this value each time you create a Level Request.
- *Uneditable Parameters* have a fixed value, which cannot be changed when you create a Level Request. Use the *Show Uneditable Parameters* link to display them.
- *Dynamic Parameters* dispose of a list of allowed values. You can select one of these allowed values from the drop-down list, when you create a Level Request.
- *Secured Parameters* are non-editable parameters whose value cannot be read by any IKAN ALM User.

By default, the uneditable parameters are hidden. Use the *Show Uneditable Parameters* option to display them.

Note: A Machine Parameter can have all the same characteristics and applies for all the Environments related to the Machine.

7. Once you have defined and verified all settings, click *Create*.

   The *Level Requests Overview* screen is displayed. It contains the information about the new Level Request (as well as about the older Level Requests).

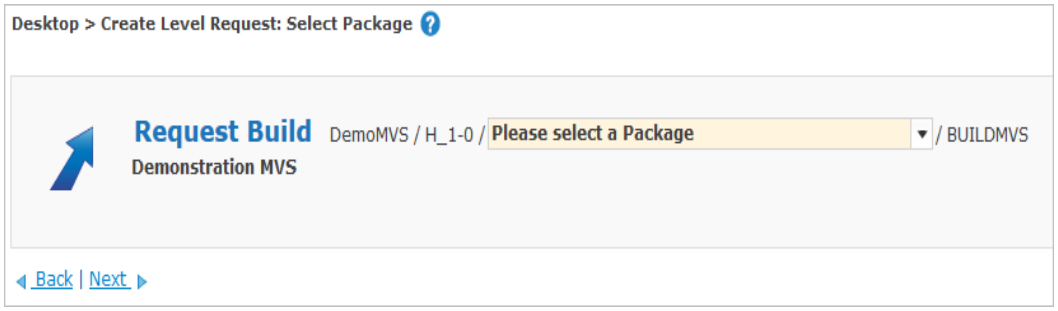
   For a detailed description of this screen refer to Level Requests Overview (page 58).

Creating a Deliver Build Level Request

1. Select *Level Requests* > *Create Level Request* on the Main Menu.

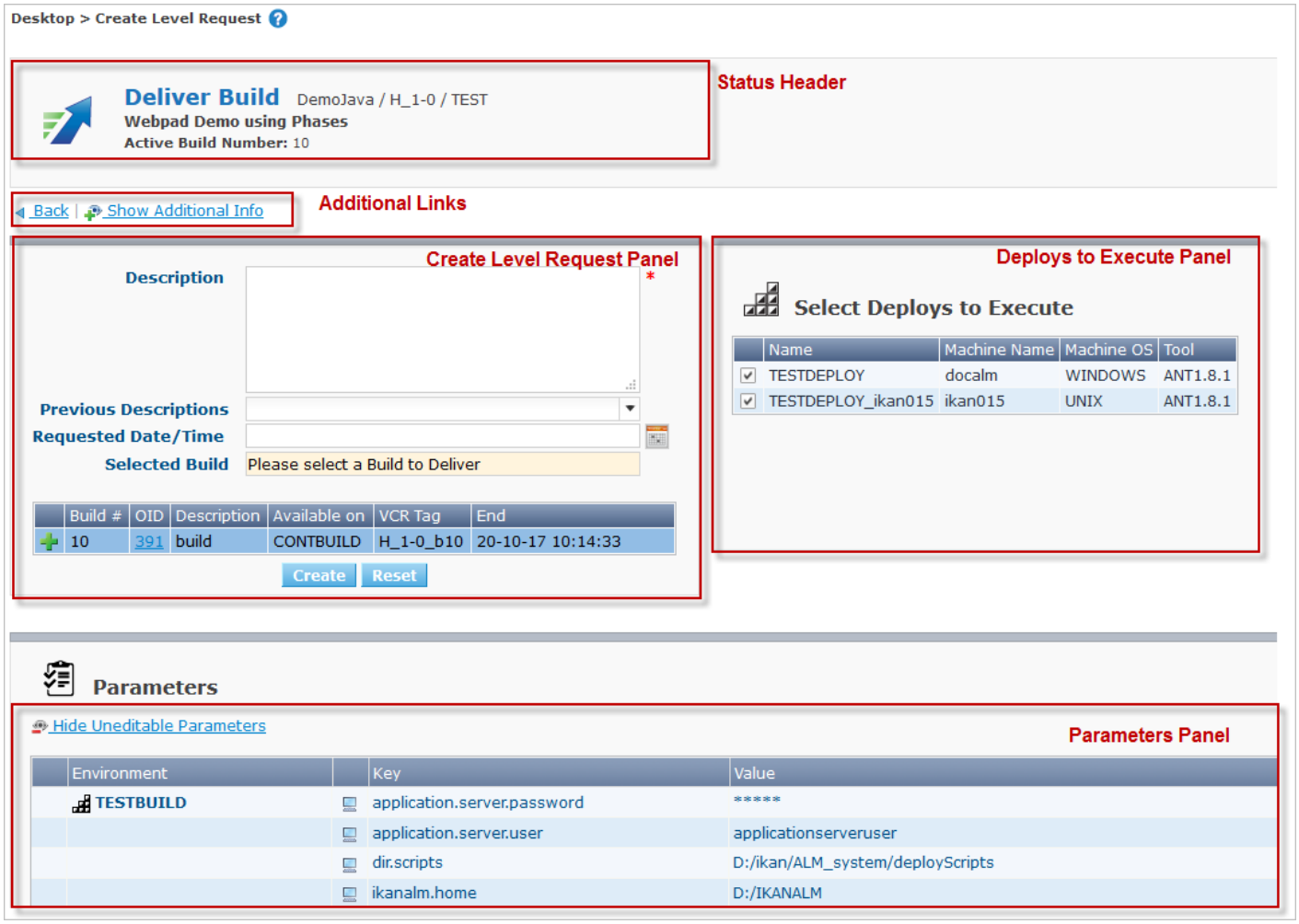
2. Click the *Deliver* icon ( ) to deliver a Build to the selected Test or Production Level.

3. If the Level belongs to a Package-based Project, you first need to select the required Package.



Note: If you add the selected package to a Desktop Tab Page, this step is avoided when creating a Level Request. See Adding Elements to a Desktop Tab Page on page 140.

4. The *Create Level Request* screen is displayed.



On this screen you will find the following sections:

- The Status Header
- Links for navigation and for showing/hiding panels with extra information
- The *Create Level Request* panel
- The optional *Select Deploys to Execute* panel becomes available in case several Deploy Environment have been defined for the Level and if the option *Make Level Optional* is activated for the Level. See also Enabling or Disabling Optional Deploys (page 182).
- The Parameters panel (only available if Parameters are linked to the involved Environments or Machines)

5. Verify the information provided in the Status Header and via the additional links in the upper part of the screen.

On this screen you will find the following sections:

- The Status Header

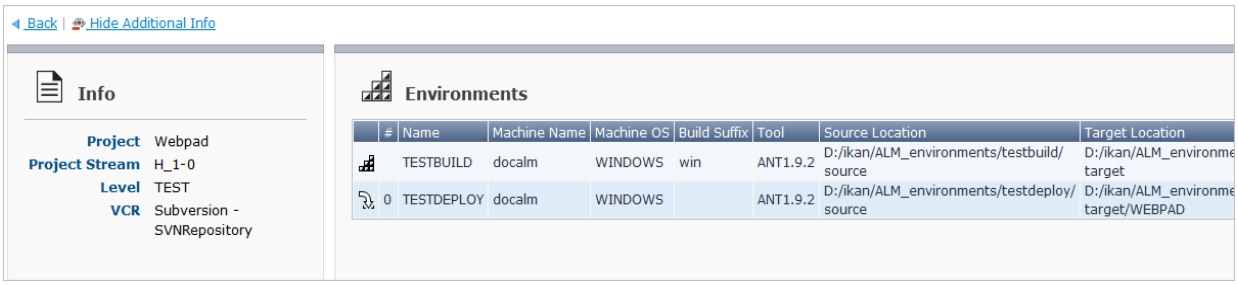
  The header displays the type of Level Request and its corresponding symbol, followed by the context of the Level Request (Project Name/Project Stream Identification[/Package Name]/Level Name), the description of the Project and the active build number.

- The *Back* link

  Click this link to return to the *Create Level Request: Select Level* page, or the Desktop Page, depending on where you launched the Create Build Level Request.

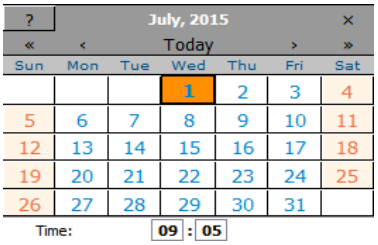
- The *Show/Hide Additional Info* link

  Click this link to display or hide information concerning the Project, Project Stream, [Package,] Level and Version Control Repository, as well as information about the Environments linked to the Build Level.



6. Complete the fields in the *Create Level Request* panel.

The following fields are available:

Field	Description
Description	In this field, enter a description for the Level Request or select one of the previously entered descriptions.
Previous Descriptions	From the drop-down list, select one of the descriptions you entered previously to automatically fill in the *Description* field.
Requested Date/Time	Leave this field blank to execute the Level Request as soon as possible. If required, enter an execution Date and Time for the Level Request in the format set in the User's local settings. You can also click the icon to select the execution date. The following screen is displayed:  Click the required date to copy it into the *Requested Date/Time* field. The execution time will be set to the current time. However you can still change the execution time manually.
Selected Build	Select the Build to be delivered to the Test or Production Level. The list contains all Builds available on the previous Level that have not yet been delivered to this Level and that have the same (Redeliver) or a higher Build Number than the current active Build. If the previous Level in the Lifecycle is marked as *Optional* (See Making a Level optional or required on page 181.) the list contains the available Builds from the previous Level AND from the Level before that one. The column *Available on* indicates on which Level the available Build resides. The current active build on a Level can be redelivered. If such a Build exists, it will be marked in blue. In that case, the *Level Request Action Type* will be "Redeliver Build".

7. If available, select the deploys to be executed in the S*elect Deploys to Execute* panel.

8. If available, verify and/or edit the settings for the available Build and Deploy Parameters in the Parameters panel.

   See Creating a Build Level Request (page 48) for more information on the available parameters.

9. Once you have defined the required settings, click *Create*.

   The *Level Requests Overview* screen is displayed. It contains the information about the new Level Request (as well as about the older Level Requests).

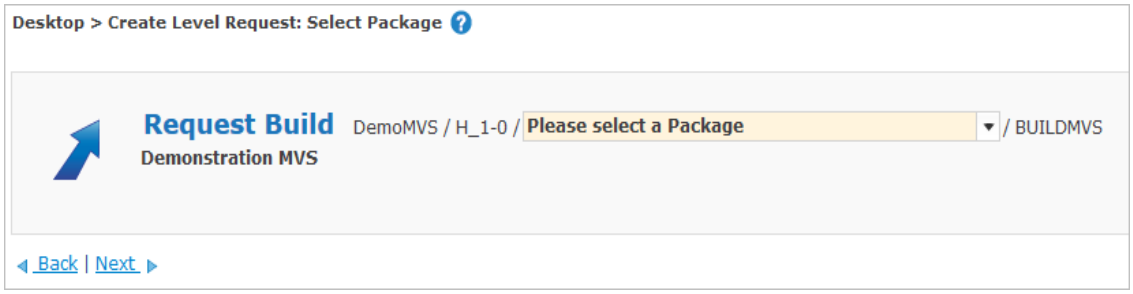
   For a detailed description of this screen refer to Level Requests Overview (page 58).

Creating a Rollback Build Level Request

1. Select *Level Requests* > *Create Level Request* on the Main Menu.

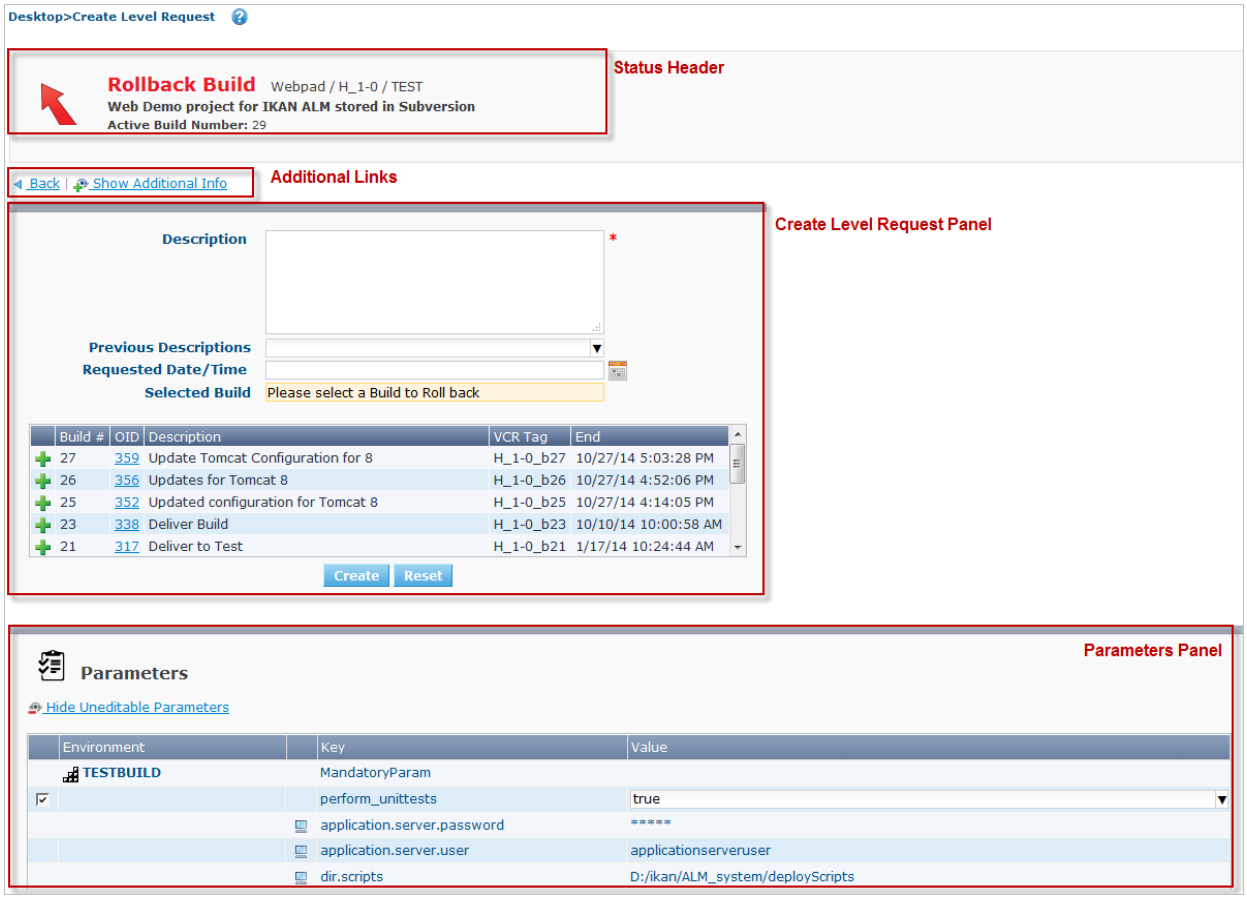
2. Click the *Rollback* icon ( ) to restore the previous Build onto the selected Test or Production Level.

3. If the Level belongs to a Package-based Project, you first need to select the required Package.



Note: If you add the selected package to a Desktop Tab Page, this step is avoided when creating a Level Request. See Adding Elements to a Desktop Tab Page on page 140.

4. The *Create Level Request* screen is displayed.



On this screen you will find the following sections:

- The Status Header
- Links for navigation and for showing/hiding panels with extra information
- The *Create Level Request* panel
- The optional *Select Deploys to Execute* panel becomes available in case several Deploy Environment have been defined for the Level and if the option *Make Level Optional* is activated for the Level. See also Enabling or Disabling Optional Deploys (page 182).
- The Parameters panel (only available if Parameters are linked to the involved Environments or Machines)

5. Verify the information provided in the Status Header and via the additional links in the upper part of the screen.

On this screen you will find the following sections:

- The Status Header

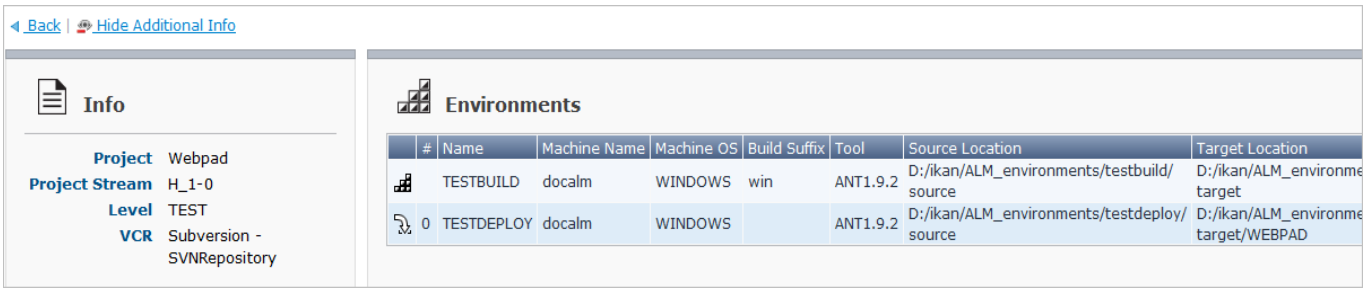
  The header displays the type of Build Level Request (Force or Request Build) and its corresponding symbol, followed by the context of the Level Request (Project Name/Project Stream Identification[/Package Name]/Level Name), the description of the Project and the active build number.

- The *Back* link

  Click this link to return to the *Create Level Request: Select Level* page, or the Desktop Page, depending on where you launched the Create Build Level Request.

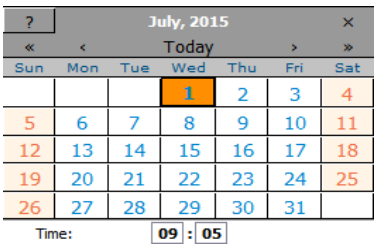
- The *Show/Hide Additional Info* link

  Click this link to display or hide information concerning the Project, Project Stream, [Package,] Level and Version Control Repository, as well as information about the Environments linked to the Build Level.



6. Complete the fields in the *Create Level Request* panel below.

   The following fields are available:

Field	Description
Description	In this field, enter a description for the Level Request or select one of the previously entered descriptions.
Previous Descriptions	From the drop-down list, select one of the descriptions you entered previously to automatically fill in the *Description* field.
Requested Date/Time	Leave this field blank to execute the Level Request as soon as possible. If required, enter an execution Date and Time for the Level Request in the format set in the User's local settings. You can also click the icon to select the execution date. The following screen is displayed:  Click the required date to copy it into the *Requested Date/Time* field. The execution time will be set to the current time. However you can still change the execution time manually.
Selected Build	Select the Build to be restored on the selected Test or Production level. The list contains all Builds that have been delivered to this Level (except the current active Build on this Level).

7. If available, select the deploys to be executed in the *Select Deploys to Execute* panel.

8. If available, verify and/or edit the settings for the available Build and Deploy parameters in the *Parameters* panel.

   Refer to the section Creating a Build Level Request (page 48) for more information on the available parameters.

9. Once you have defined the required settings, click *Create*.

   The *Level Requests Overview* screen is displayed. It contains the information about the new Level Request (as well as about the older Level Requests).

   For a detailed description of this screen refer to Level Requests Overview (page 58).

7.3. Level Requests Overview

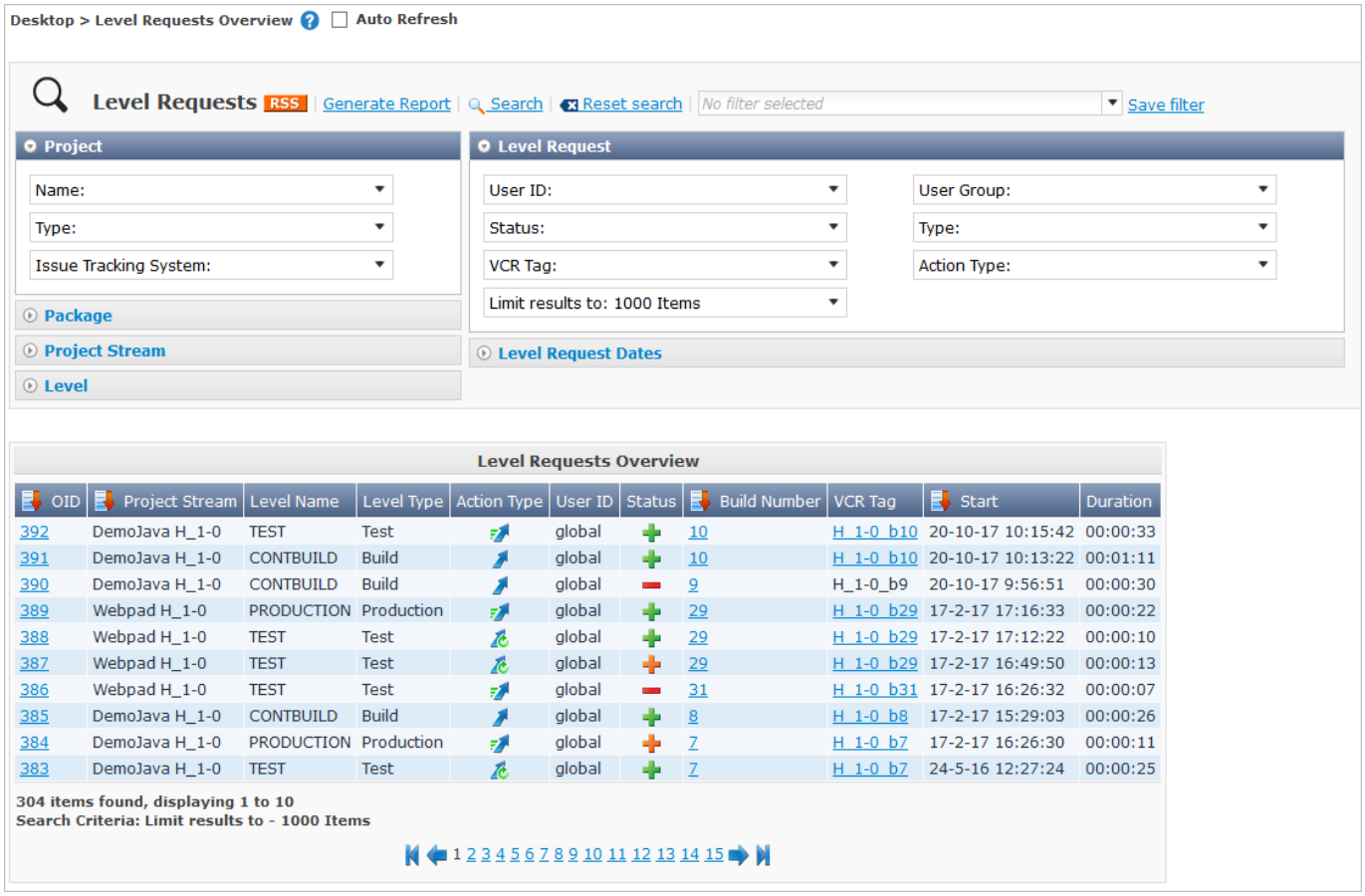
The following sections deal with the procedures involved when using the *Level Requests Overview*:

- The Level Requests Overview Screen (page 58)
- The IKAN ALM RSS Functionality (page 62)
- Generating a Report (page 63)
- Level Request Detail (page 65)

The Level Requests Overview Screen

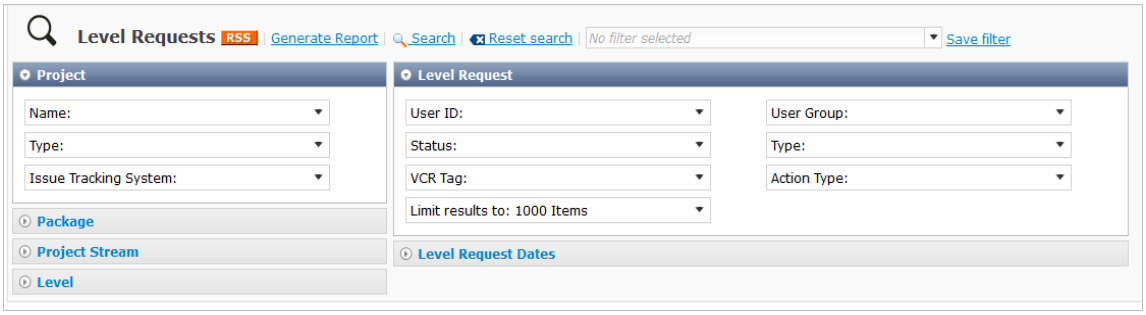
1. Select *Level Requests > Overview Level Requests* on the Main Menu.

   The *Level Requests Overview* screen is displayed:



Note: If the *Auto Refresh* option is activated, the *Level Requests Overview* screen will be refreshed each time the defined rate is expired. See Auto Refresh on page 12.

2. Use the search criteria on the *Search* panel to only display the Level Requests you are looking for.



The following options are available:

- Search: in principle it is not necessary to click the *Search* option. The results on the overview will be automatically synchronized in function of the selected criteria.
- Reset search: to clear all search criteria and display the full list of items.
- Select an existing filter from the drop-down list.
- Save filter: to save the current search criteria for future use.

For more information on the usage of search panels and filters, refer to the sections Search Panels (page 10) and Defining Search Filters (page 16).

3. Click the *Search* button once again if you want to verify the changing status of existing and new Level Requests.

Note: If the *Auto Refresh* option is activated, the *Level Requests Overview* will be refreshed following the interval specified by the Auto Refresh Rate specified in the System Settings. See Auto Refresh on page 12.

4. Use the *Generate Report* button to run the *Level Requests Overview* Report.

See Generating a Report (page 63) for more information on *Generating a Level Requests Overview Report*.

5. On the *Level Requests Overview*, verify the Level Request Information fields for the required Level Request.

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

The following information fields are available:

Field	Description
OID	This field contains the OID (Object Identifier) of the Level Request. This is a unique sequential number assigned to each Level Request when it is created. The Level Request OIDs are displayed as a link. Click this link to display the details for this Level Request. For more information, refer to the section explaining the Summary tab page of the *Level Request Detail* screen.
Project Stream	This field contains the identification of the Project Stream. This name is composed of: • Project Name • Project Stream Type: H (Head) or B (Branch) • Project Stream Prefix, optionally followed by the Suffix in case of a Branch Project Stream Example: `Webpad H_1-0`
Level Name	This field contains the name of the Level concerned by the Level Request.
Level Type	This field contains the type of the Level concerned by the Level Request (Build, Test or Production).
Action Type	This field contains the type of the Level Request Action. The following types are available: • Build initiated by Scheduler • Force Build • Request Build • Deliver Build • Redeliver Build • Rollback Build For a description of the latter four Level Request Action Types, refer to Creating Level Requests (page 45). The Build initiated by the Scheduler is similar to the Request Build Level Request Action Type, but it is triggered automatically.
User ID	This field contains the User ID of the User who created the Level Request. For Level Requests initiated by the Scheduler, this field remains empty.

Field	Description
Status	This field contains the Level Request Status. The following status indication icons are possible: • (Success): the Level Request is executed successfully. • (Warning): the Level Request has been successfully executed, but at least one non-critical Level, Build or Deploy Phase failed, e.g., for debugging reasons. • (Fail): the execution of the Level Request as a whole failed. This is due to the failure of one or more critical Level, Build or Deploy Phases. • (Run): the Level Request is being executed at this moment. • (Aborting): the Level Request is being aborted at this moment. • (Awaiting requested Date/Time): the requested execution is in the future, or is waiting for the Monitor process to pick it up • (Awaiting Pre-Approval or Awaiting Post-Approval): the Level Request is awaiting a Pre- or Post-Approval. • (Rejected): An Approval associated with the Level Request was reiected. The Level Request will never be executed. • (Canceled): the Level Request has been canceled before it was run. It will never be executed. • (Aborted): the Level Request has been aborted during execution. The results (such as Build Results) that were already available at the time of the abort have been cleaned up and cannot be used.
Build Number	This field contains the Build Number of the Level Request. Use this link to access the *Build History Detail* screen. See Build History Screen on page 104.
VCR Tag	This field contains the VCR Tag of the Level Request. This Tag matches a Build with its source code in the VCR. The format of the VCR Tag normally matches the Tag Template defined for the Stream. See The Project Streams Overview Screen on page 165. However, the user can override the default VCR Tag while creating a Level Request (and is obliged to do so for a Build Level Request in a Tag Based Project Stream), so that the Tag Format can be completely different. The Level Request VCR Tag is displayed as a link leading to the *Sources* tab page on the *Level Request Detail* screen. For more information, refer to the section Sources (page 97).
Start	This field indicates the date and time when the Level Request execution started.
Duration	This field indicates the total duration of the Level Request.

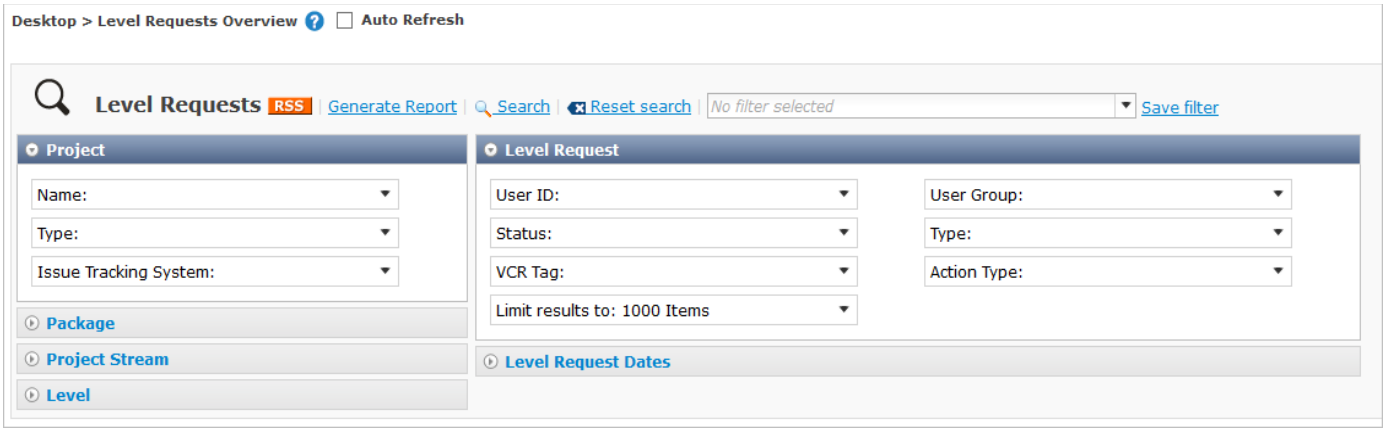
6. View the details of a specific Level Request.

   Click the Level Request's *OID* link in front of the required Level Request.

   For more information, refer to the section Level Request Detail (page 65).

The IKAN ALM RSS Functionality

If your Global IKAN ALM Manager has activated RSS Feeds at System Settings level, the orange *RSS* button is available on the *Search Level Request* panel.



RSS is a web format used to publish frequently updated digital content, such as blogs, news feeds or podcasts. Consumers of RSS content use special browsers called aggregators to watch for new content in dozens or even hundreds of web feeds. Programs known as feed readers or aggregators can check a list of feeds on behalf of a user and display any updated articles that they can find.

RSS feeds can be shown by a plug-in in the user's IDE or by other RSS Readers including the Mozilla Firefox browser.

IKAN ALM provides RSS Feeds for displaying data about the last 10 Level Requests that meet specified criteria.

1. Select *Level Requests > Overview Level Request*s on the Main Menu.

2. Specify for which Level Requests you want information to appear in the RSS feed.

   Initially the URL for the RSS Feed does not contain any criteria, except for the current user's language. To specify which Level Requests you want to appear in the RSS Feed, define the search criteria on the *Search Level Request* panel.

   The list of Level Requests matching the set criteria will appear in the *Level Requests Overview* panel.

   Most of the criteria will be added to the URL. See the RSS URL Details to see which criteria might be used.

3. Display the RSS Feed

   Click the *RSS* button. A browser window will open, displaying the RSS Feed for the Level Requests you selected.

   Note: If your browser does not have an integrated RSS Reader, you must manually add the URL for the RSS Feed. To do so, select and copy the URL from the Location Bar of your browser window, and paste it in the Properties Settings of your RSS plug-in or reader.

   The RSSOwl plug-in can be found on the Eclipse update site: http://www.rssowl.org/.

   You find a detailed explanation of the structure of the IKAN ALM URL in the section RSS URL Details (page 518).

Generating a Report

This functionality allows you to generate a report for specified Level Requests. This report can be exported to PDF, CSV, RTF or XLS format.

1. Switch to the *Level Requests Overview* screen and specify for which Level Requests you want to generate a report.

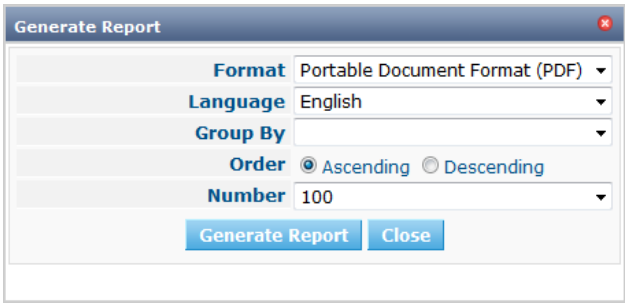
   See Level Requests Overview on page 58.

   To specify which Level Requests you want to appear in the Report, define the search criteria and click the *Search* button.

   The list of Level Requests matching the set criteria will appear in the *Level Requests Overview* panel. These criteria will be used by the Report Generation.

2. Click the *Generate Report* button.

   The following dialog is displayed:



The following selection fields are available:

Field	Description
Format	Select the required export format from the drop-down menu. The following formats are available: • Portable Document Format (PDF) • Comma Separated Values (CSV) • Rich Text Format (RTF) • MS Excel Worksheet (XLS)
Language	Select the required language for the report from the drop-down menu. The following languages are available: • English • French • German
Group By	Optional field which enables to group the reported Level Requests by • Project Name • Level Name
Order	Select whether the reported Level Requests are to be ordered ascending or descending.
Number	Select the maximum number of results that may appear in the report. The choices are: • 20 • 50 • 100 (= default) • 200 • 500

Make the required selections and click *Generate Report*.

The report is generated. The following is an example of a report saved in PDF format:

IKANALM

IKAN ALM Level Request Report

generated by IKAN ALM on October 10, 2014 9:50:22 AM CEST

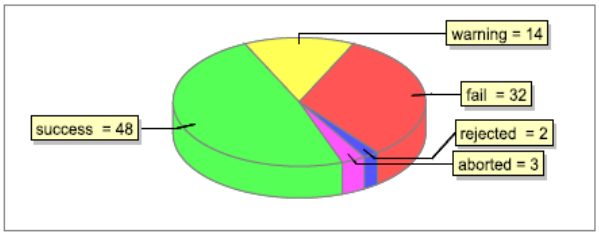
OID	Project Name	Project Stream Type	Level Name	Level Type	Type	Action Type	User ID	Status	VCR Tag	Start	End
336	Monocalendar	Head	TEST	Test	Builds based on tagged code	Deliver Build	global	success	H_1-0_b21	10/9/14 11:26 AM	10/9/14 11:27 AM
335	DemoMVS	Head	BUILDMVS	Build	Builds based on latest code	Request Build	global	success	H_1-0_Package-1_b1	10/9/14 11:26 AM	10/9/14 11:26 AM
334	Webpad	Head	PRODUCTION	Production	Deploys of archived Build	Deliver Build	global	awaiting Pre-Approval	H_1-0_b21		
333	Webpad	Head	CONTBUILD	Build	Builds based on latest code	Force Build	global	success	H_1-0_b23	10/8/14 10:13 AM	10/8/14 10:13 AM
332	Webpad	Head	CONTBUILD	Build	Builds based on latest code	Force Build	global	success	H_1-0_b22	10/8/14 9:38 AM	10/8/14 9:38 AM
331	DemoJava	Head	CONTBUILD	Build	Builds based on latest code	Force Build	global	warning	H_1-0_b5	10/6/14 3:19 PM	10/6/14 3:20 PM
330	DemoJava	Head	CONTBUILD	Build	Builds based on latest code	Build initiated by Scheduler		warning	H_1-0_b4	4/17/14 7:22 PM	4/17/14 7:22 PM
329	DemoJava	Head	PRODUCTION	Production	Deploys of archived Build	Deliver Build	global	success	H_1-0_b3	4/17/14 7:11 PM	4/17/14 7:11 PM
328	DemoJava	Head	TEST	Test	Builds and Deploys on tagged code	Deliver Build	global	warning	H_1-0_b3	4/17/14 7:10 PM	4/17/14 7:10 PM
327	DemoJava	Head	CONTBUILD	Build	Builds based on latest code	Force Build	global	warning	H_1-0_b3	4/17/14 7:09 PM	4/17/14 7:09 PM
326	DemoJava	Head	PRODUCTION	Production	Deploys of archived Build	Deliver Build	global	success	H_1-0_b2	4/17/14 6:43 PM	4/17/14 6:43 PM
325	DemoJava	Head	PRODUCTION	Production	Deploys of archived Build	Deliver Build	global	fail	H_1-0_b2	4/17/14 6:37 PM	4/17/14 6:37 PM

1/8

Level Requests Overview

Total Number of Level Requests : 100

Successful Level Requests : 48



More options are available when Generating a Report with the IKAN ALM Command Line. For more information, refer to the section Command Line Interface (Optional) (page 475).

3. Use the *Close* button to return to the *Level Requests Overview* screen.

7.4. Level Request Detail

The *Level Request Detail* screen contains the detailed information concerning the selected Level Request.

The screen is structured as follows:

1. Status Header

   The header displays the status and corresponding symbol of the selected Level Request, as well as the Level Request OID and description, the requester (User or Schedule) and the date and time at which the Level Request has been requested.

2. Tab Pages with detailed information

   Underneath the status indication, several tabs are available, each of them displaying additional information concerning the Level Request. By default the *Summary* tab page is displayed.

   Refer to one of the following sections for more information.

   - Summary (page 66)
   - Phase Logs (page 77)
   - Results (page 89)
   - Approvals (page 91)
   - Issues (page 93)
   - Sources (page 97)
   - Modifications (page 98)
   - Dependencies (page 99)

3. Back, Refresh and Build History links

   - Use the *Back* link to return to the previous screen.
   - Use the *Refresh* link to update the displayed information. This link reloads the currently selected tab page, as well as the header information.
   - Use the *Build History* link to get information about the Build's Lifecycle.

     For more detailed information, refer to the section Build History Screen (page 104).

4. Auto Refresh option

   In some cases it might be useful to activate the *Auto Refresh* option.

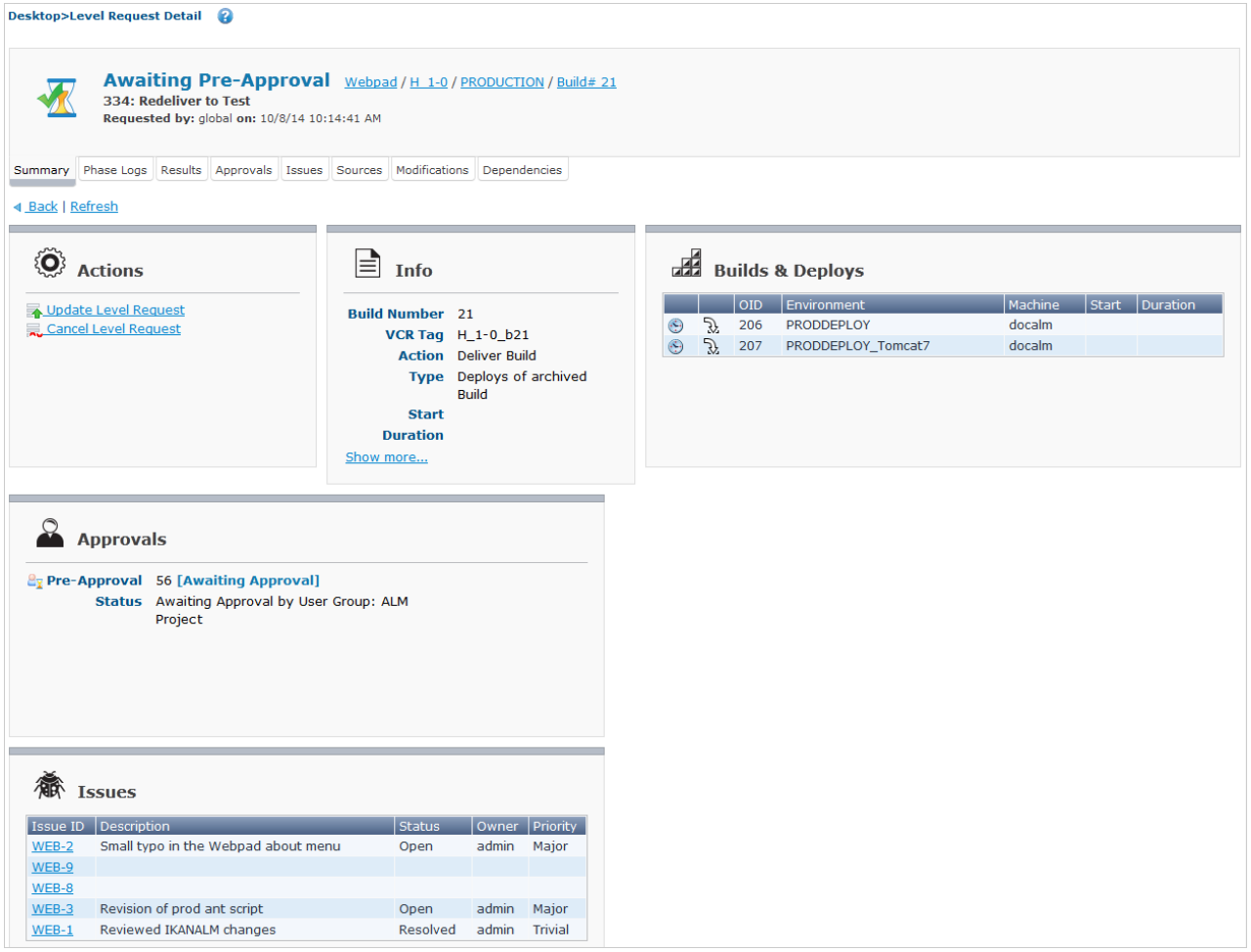
   On the *Phase Logs* tab page, for example, it allows you to follow the execution steps of a Level Request. Auto Refresh is also available on the *Summary*, *Approvals*, *Issues* and *Dependencies* tab pages.

   Once the Level Request has reached a final status (*Success*, *Rejected*, *Canceled*, *Aborted*, *Fail* or *Warning*), the *Auto Refresh* function will be stopped automatically.

   For more information on the *Auto Refresh* settings, refer to the section Auto Refresh (page 12).

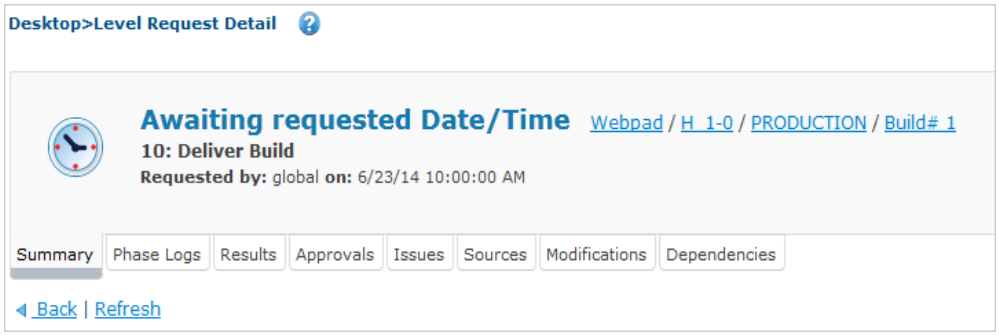
Summary

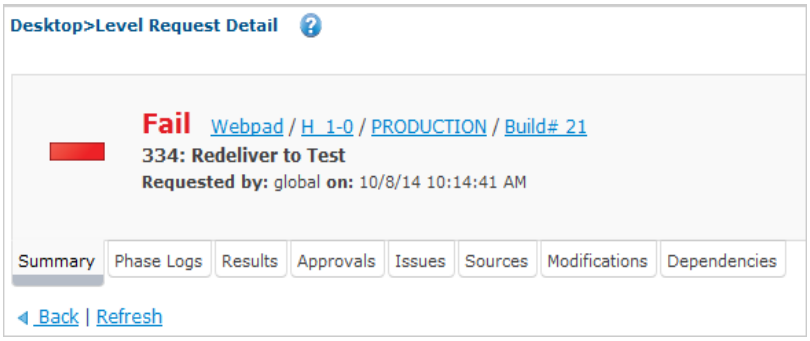
The *Summary* page displays the status of the Level Request and, underneath, several panels providing detailed information. The panels displayed depend on the status of the Level Request.



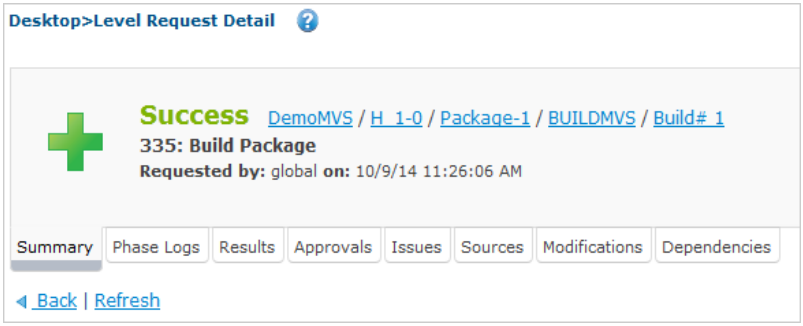
Status Header

Some examples of Level Requests for release-based Projects:





Example of a Level Request for a package-based Project:



The header of the *Level Request Detail* screen displays the status and the corresponding symbol of the selected Level Request, as well as the Level Request OID and description, the requester (User or Schedule) and the date and time at which the Level Request has been requested.

Note: The links next to the status indication lead to the *Level Request Overview* screen. Depending on the link element you select, more information will already be filled in on the *Search Level Request* panel to limit the Level Requests displayed on the overview.

Link Element	Preselected Search Details
Project	Project Name
Project Stream	Project Name and Build Prefix (and, optionally, the Build Suffix in case of a Level Request for a Branch Project Stream).
Package (only for Package-based Projects)	Project Name, Build Prefix (and, optionally, the Build Suffix) and Package Name.
Level	Project Name, Build Prefix, Package Name and Level Name.
Build Number	Project Name and VCR Tag.

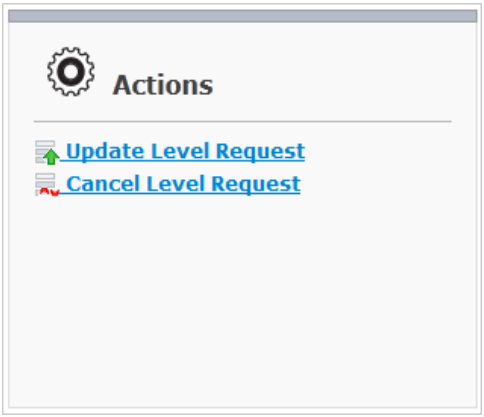
Note: When selecting another Tab Page, this header is not being refreshed.

Depending on the status of the Level Request, the *Summary* page may contain the following panels:

- Actions Panel (page 68)
- Info Panel (page 70)
- Builds and Deploys Panel (page 71)
- Approvals Panel (page 72)
- Issues Panel (page 73)
- Error/Warning Log Panel (page 75)

Actions Panel

The actions available in this panel depend on the status of the Level Request.



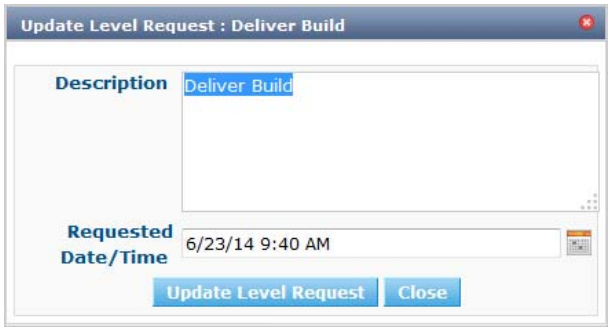
Actions are available when the Level Request execution time is set to a moment in the future, if an Approval is pending for the Level Request or if the Level Request is still being executed.

1. The Level Request execution time is set to a moment in the future or the Level Request is waiting for an Approval:

The following action links will be available:

Link	Meaning
Update Level Request	Click this link to update the Level Request Description and/or Execution Time. **Note:** This action is not available for a Build level.
Cancel Level Request	Click this link to cancel the Level Request. Once you have confirmed the cancellation, the Level Request Status will be set to *Canceled*. It is no longer possible to cancel a Level Request, once an assigned Approval has been granted.

Clicking the *Update Level Request* link shows the *Update Level Request* screen.



The following fields may be edited:

Field	Meaning
Description	This field contains the description entered by the user, when he or she created the Level Request.
Requested Date/Time	This field indicates when the execution of the Level Request should start. This date and time cannot be in the past. If left blank, the current system time will be taken as value for this field.

Click *Update Level Request* to save the changes and return to the *Level Request Detail* screen.
You can also click *Close* to cancel the update and return to the *Level Request Detail* screen.

2. The Level Request is currently being executed:

If the Level Request is currently being executed, the following button is available:

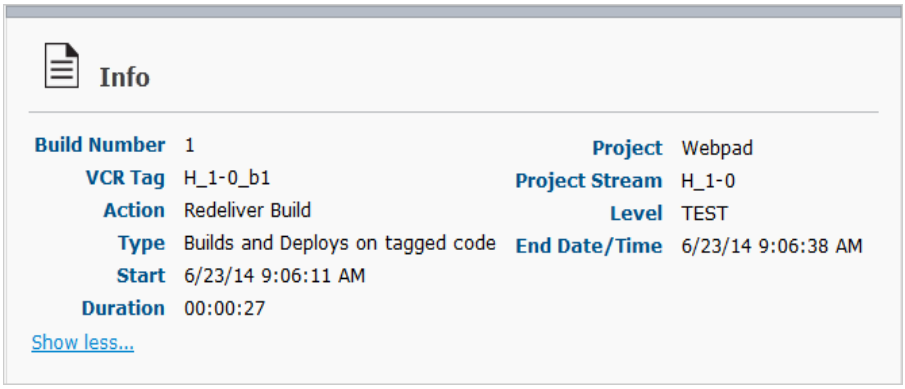
Button	Meaning
Abort Level Request	Click this button to abort the Level Request execution. Once you have confirmed the abort, the Level Request Status will be set to *Aborting*. Once the current Monitor, Build or Deploy Agent action is completed, the Level Request execution will be halted and the Level Request status will be set to *Aborted*.

Clicking the *Abort Level Request* action button displays the following screen.

Click *Abort Level Request* to confirm the action and return to the *Level Request Detail* screen.
You can also click *Close* to cancel the abort process and return to the *Level Request Detail* screen.

Info Panel

This panel contains detailed information concerning the Level Request.



Note: The *Show more...* and *Show less...* links respectively show or hide more data about the Level Request.

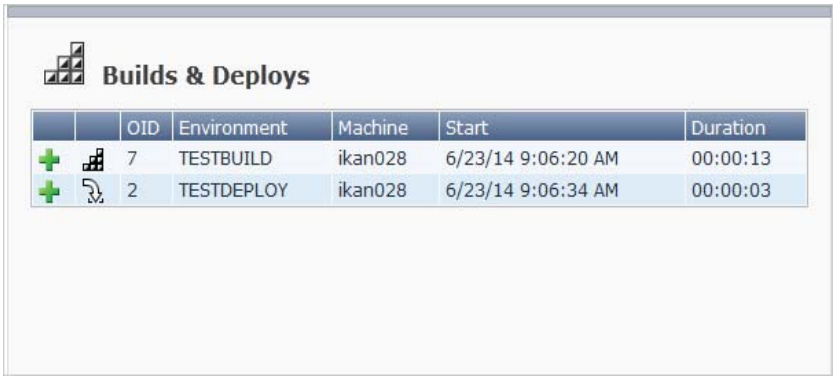
The following information is available.

Field	Meaning
Build Number	This field contains the Build number of the Level Request.
VCR Tag	This field contains the VCR Tag of the Level Request. This Tag matches a Build with its source code in the VCR. The format of the VCR Tag normally matches the Tag Template defined for the Stream. See Project Streams on page 161. However, the user can override the default VCR Tag while creating a Level Request, so that the Tag Format can be completely different.
Action	This field contains the Level Request Action Type. The following types exist: • *Build initiated by Scheduler* • *Force Build* • *Request Build* • *Deliver Build* • *Redeliver Build* • *Rollback Build* • *Dependency Build*

Field	Meaning
Type	The Level Request Type. The following types exist: • *Build based on latest code* • *Builds based on tagged code* • *Builds and Deploys on latest code* • *Builds and Deploys on tagged code* • *Deploys of archived Build* • *No Builds and Deploys*
Start	This field contains the date and time when the Level Request execution started.
Duration	This field contains the total execution time of the Level Request.
Project	This field contains the Project name.
Project Stream	This field contains the Project Stream, the Build Prefix and, optionally, the Build Suffix (in case of a Branch Project Stream).
Package	This field contains the Package name in case of Package-based Project Streams.
Level	This field contains the Level name.
End Date/Time	This field contains the date and time when the Level Request execution ended.
Partial Build Tag	In case of a Partial Build type Project Stream (see Creating a Branch Project Stream (page 161)), this field contains the Partial Build Tag. Only the sources that differ from this Tag have been retrieved and made available for the Build during the *Retrieve Code* Phase.

Builds and Deploys Panel

This panel contains the different Builds and/or Deploys that are related to the Level Request.



The following information is available:

Field	Meaning
Status icon	This field contains the Build/Deploy Status indication. This Status indication is derived from the status of the different Build/Deploy Phases. Possible status indications are: • *Wait* The Build/Deploy is waiting to be started. • *In progress* The Build/Deploy is ready to be started. • *Run* The Build/Deploy is currently being executed. • *Success* The Build/Deploy has finished successfully. • *Warning* The Build/Deploy has finished successfully, but there were some non-critical errors. • *Fail* The Build/Deploy has failed. • *Canceled* The Build/Deploy was canceled before it was executed. • *Aborted* The Build/Deploy was aborted while it was being executed. • *Rejected* The Build/Deploy was rejected.
Type icon	This field indicates the type: Build ( ) or Deploy ( ).
OID	This field contains the OID (Object Identifier) of the Build/Deploy. This is a unique sequential number assigned to each Build/Deploy Action when it is created. **Note:** The OID is not equal to the Build/Deploy Number!
Environment	This field contains the name of the Build/Deploy Environment where this Build/Deploy was executed.
Machine	This field contains the name of the Machine hosting the Build/Deploy Environment where this Build/Deploy was executed.
Start	This field indicates the date and time when the Build/Deploy execution started.
Duration	This field indicates the total execution time of the Build/Deploy.

Note: For more detailed information about the Build/Deploy of the Level Request, select the *Phase Logs* tab underneath the Status header.

Approvals Panel

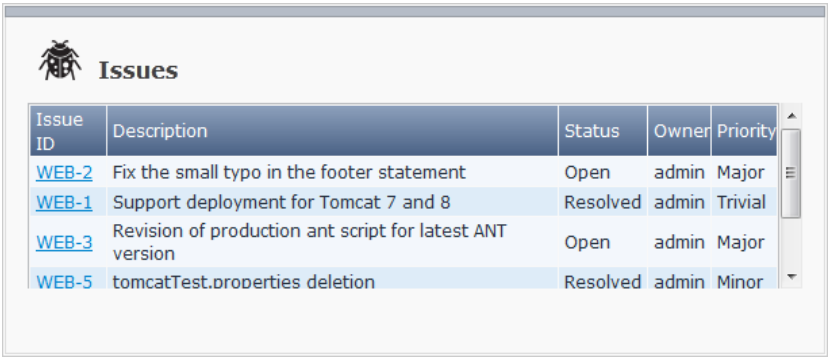
This panel is only displayed if the Level Request has been rejected or if the status of the Level Request is Awaiting a Pre- or Post-approval.

It displays the type and OID of the *Awaiting* or *Rejected* Approval for the Level Request, its status and to which User Group the User has to belong to for approving or rejecting the Approval. If the Level Request has been Rejected, the Reason will also be displayed.

Note: For a complete list of all Approvals defined for the Level Request, select the *Approvals* tab underneath the Status header. See Approvals on page 91.

Issues Panel

This panel is only shown if there are Issues related to the Level Request.



Issues can get linked to a Level Request in two ways: automatically by IKAN ALM during the execution of the Level Request or manually by a User after the Level Request has ended. Refer to the section Issue Tracking (page 381) for more information on defining an external Issue Tracking System, and to the section Phases - General Information (page 520) for more information on the Issue Tracking Phase.

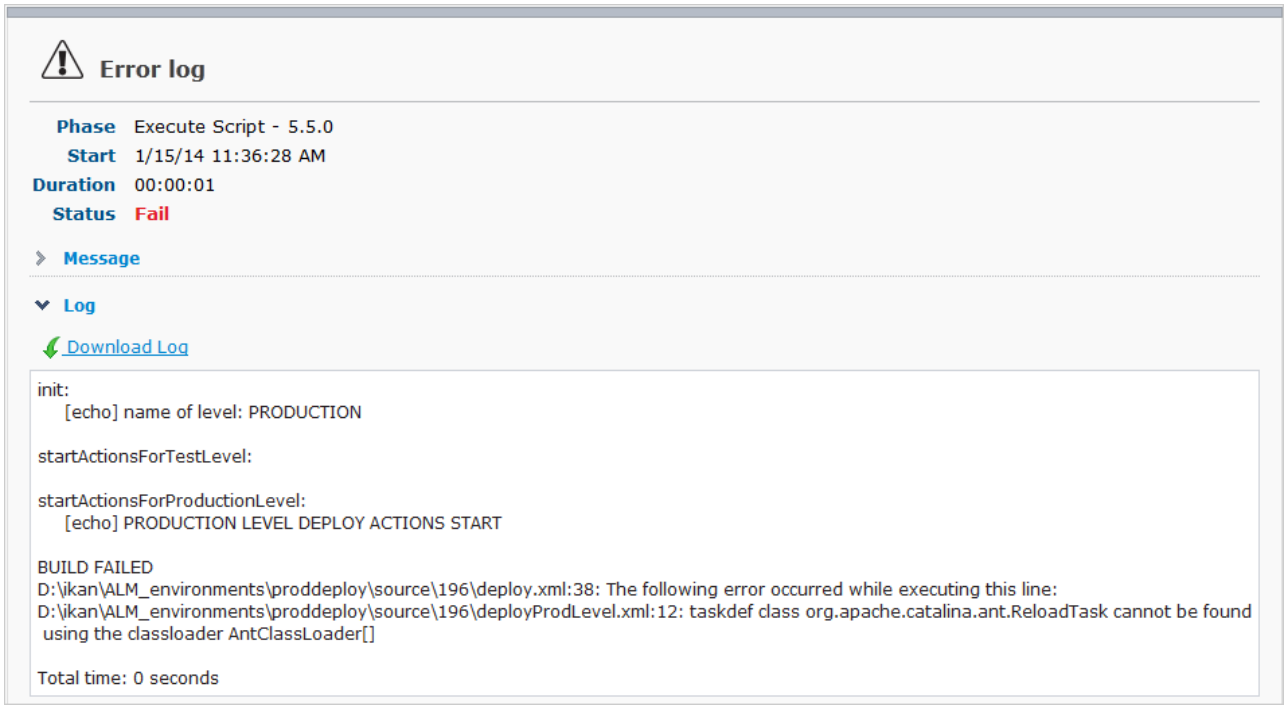
For each issue the following information is displayed:

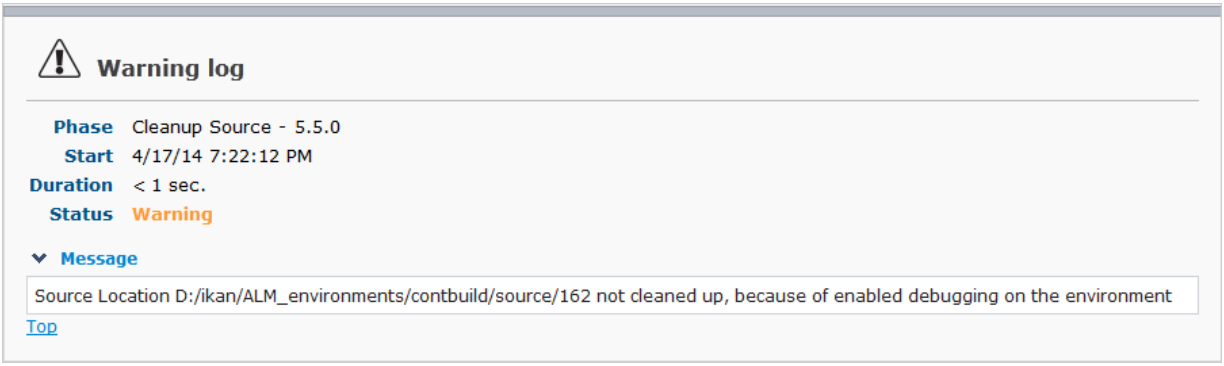
Field	Meaning
Issue ID	This field displays the ID by which the Issue is defined in the external Issue Tracking System. If the *URL* field in the definition of the Issue Tracking System is not empty, this field will be displayed as a link. Click the link to view the Issue in the external Issue Tracking System's Web interface. For more information on the *URL* field, refer to the section Creating an Issue Tracking System (page 382).
Description	This field contains the description of the Issue.
Status	This field contains the status of the Issue.
Owner	This field contains the owner of the Issue.
Priority	This field contains the Issue priority.

Note: On this panel you cannot modify any of the Issues. If you want to edit, delete or synchronize them, select the *Issues* tab underneath the Status header. See Issues on page 93.

Error/Warning Log Panel

This panel is only shown if the Level Request status is set to *Fail* or *Warning*. It only shows the log of the **first** Phase with status *Fail* or *Warning*. This is not necessarily the cause of the error status of the Level Request. For a complete overview of the status of **all** the Phases, refer to the *Phase Logs* tab page (Phase Logs (page 77)).





The following information is displayed.

Field	Meaning
Phase	This field contains the Display Name of the Phase, combined with the Phase Version.
Start	This field displays the date/time when the Phase started.
Duration	This field displays the total execution time of the Phase.
Status	This field displays the status of the Phase. Possible status indications are: • *Success*: The Phase finished successfully • *Warning*: The Phase finished with a warning • *Fail*: The Phase failed • *Running*: The Phase is currently being executed • *Not executed:* The Phase has not been executed • *Aborted*: The Phase was aborted

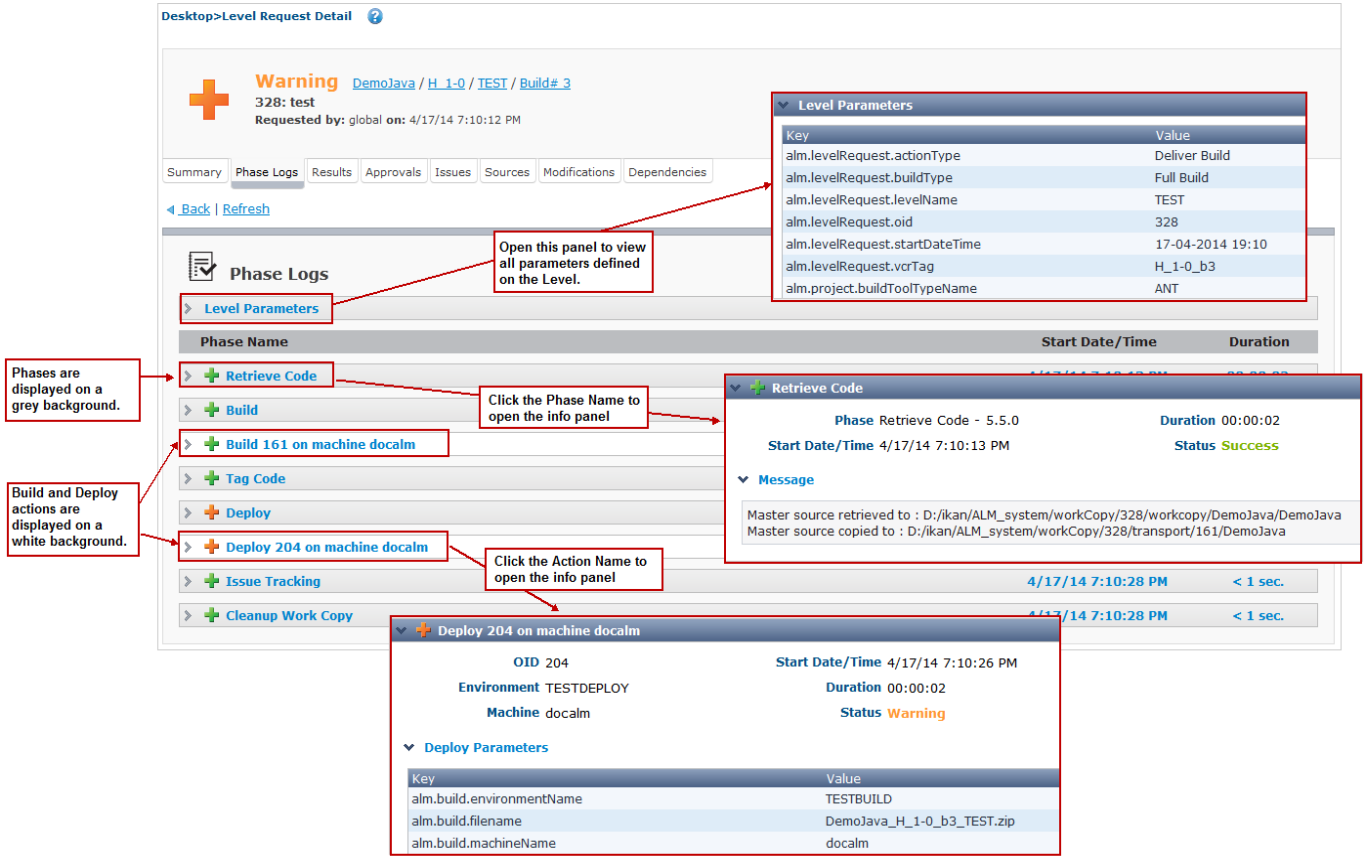
Field	Meaning
Message	Clicking this link displays the message. If no log is available, the message will be displayed immediately. **Note:** if you hover over the word *Message*, the first 256 characters of the message text are shown as tooltip.
Stack Trace	If available, this fields displays the Stack Trace. **Note:** if you hover over the word *Stack Trace*, the first 256 characters of the stack trace text are shown as tooltip.
Log	This field displays the log of the Phase. The log is only available for Phases executed by a Scripting Tool (Ant, NAnt or Maven2). If the Log Format is set to *TXT*, you can download the error/warning log on the *Phase Logs* tab page. For more information on specifying the format of the log, refer to the section Scripting Tools (page 394).

Use the *Top* link to quickly return to the top of the page.

RELATED TOPICS
- Level Request Detail (page 65)
- Phase Logs (page 77)
- Results (page 89)
- Approvals (page 91)
- Issues (page 93)
- Sources (page 97)
- Modifications (page 98)
- Dependencies (page 99)

Phase Logs

This page displays the logs of the Level Phases, the Build and Deploy actions and their Build and Deploy Phases executed during the handling of a Level Request. It also provides more detailed information regarding the used Parameters.



The following panels can be available:

- Level Parameters (page 80)
- Level Phases (page 80)

  Possible subpanels:

  - Phase Parameters
  - Message
  - Stack Trace
  - Logs
- Build Actions (page 82)

  Possible subpanels:

  - Used Build Parameters
  - Build Phases

    Just as the Level Phases panel, the Build Phases panel may contain the following subpanels:

    - Phase Parameters
    - Message
    - Stack Trace
    - Logs

- Deploy Actions (page 85)

  Possible subpanels:

  - Used Deploy Parameters
  - Deploy Phases

    Just as the Level Phases panel, the Deploy Phases panel may contain the following subpanels:

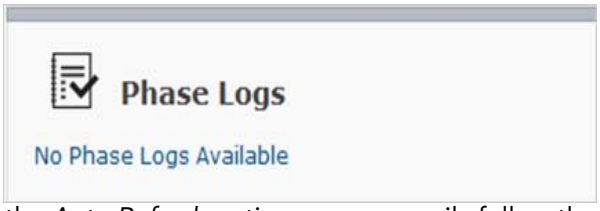
    - Phase Parameters
    - Message
    - Stack Trace
    - Logs

If a Phase is still running or if one of the Phases failed, the log of that Phase will be automatically opened. The parameters (for Custom Phases), the message, the log and/or the stack trace will be displayed.

Items on a gray background represent the different Phases, items on a white background represent the Build or Deploy actions.

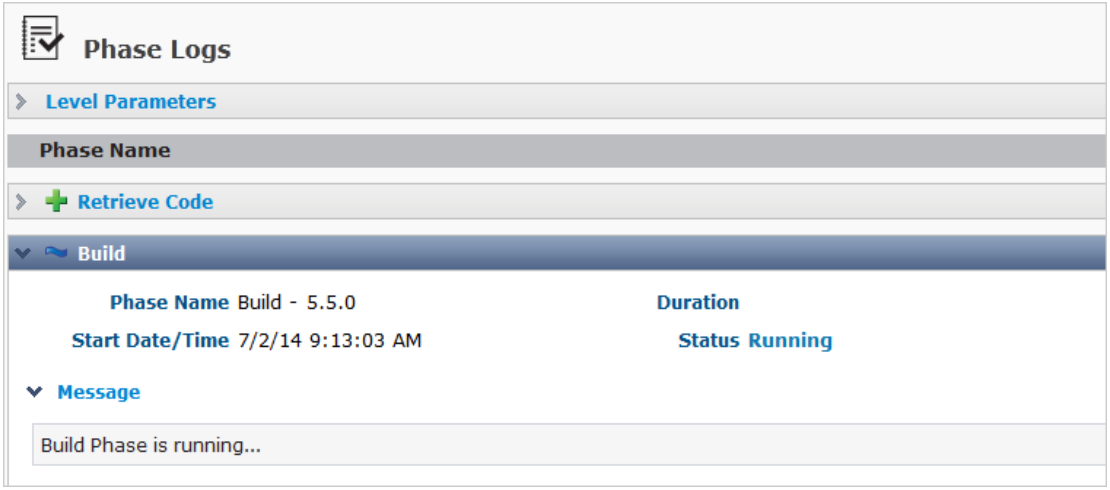
Click a Phase/Action Name to expand its information panel.

Note: If the Level Request status is set to *Awaiting requested Date/Time*, *Awaiting Pre-Approval*, *Canceled* or *Rejected* due to a Pre-approval that has been rejected, no Phase logs are available.

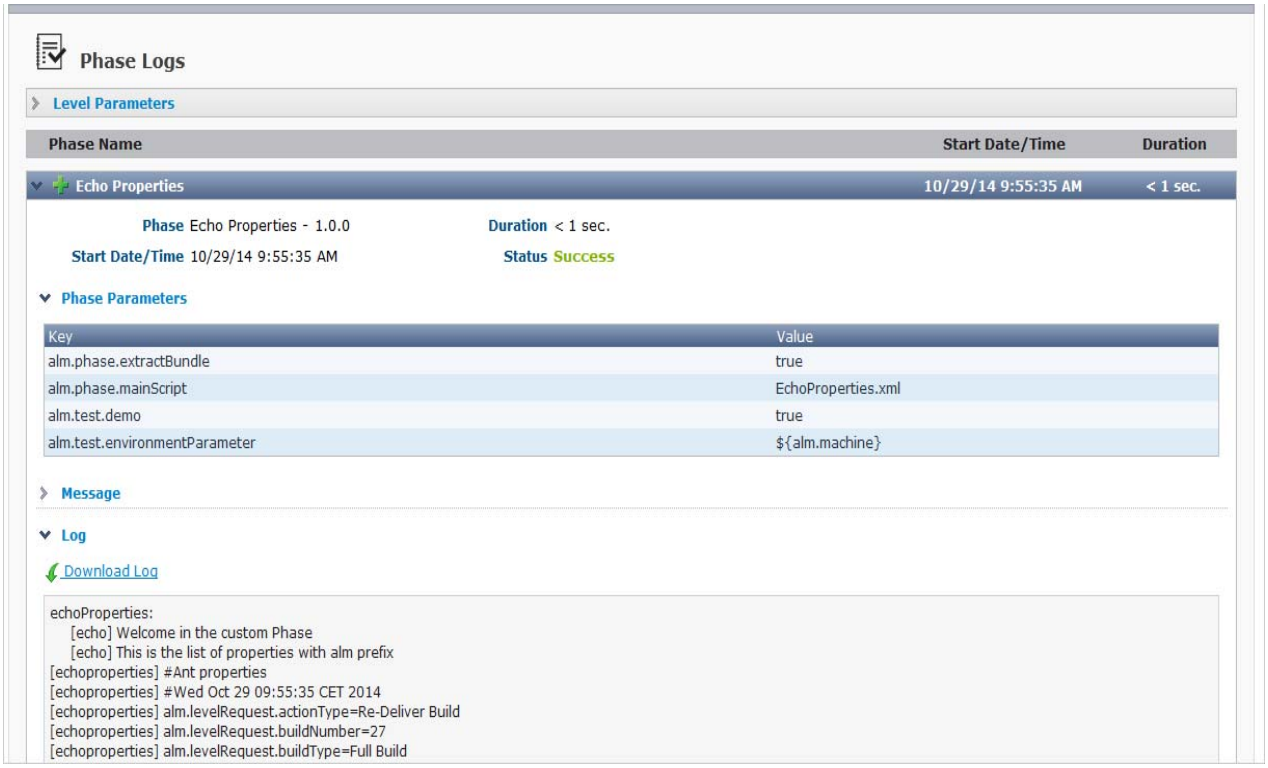


If you activate the *Auto Refresh* option, you can easily follow the execution of the different Phases.

Example of a running Phase

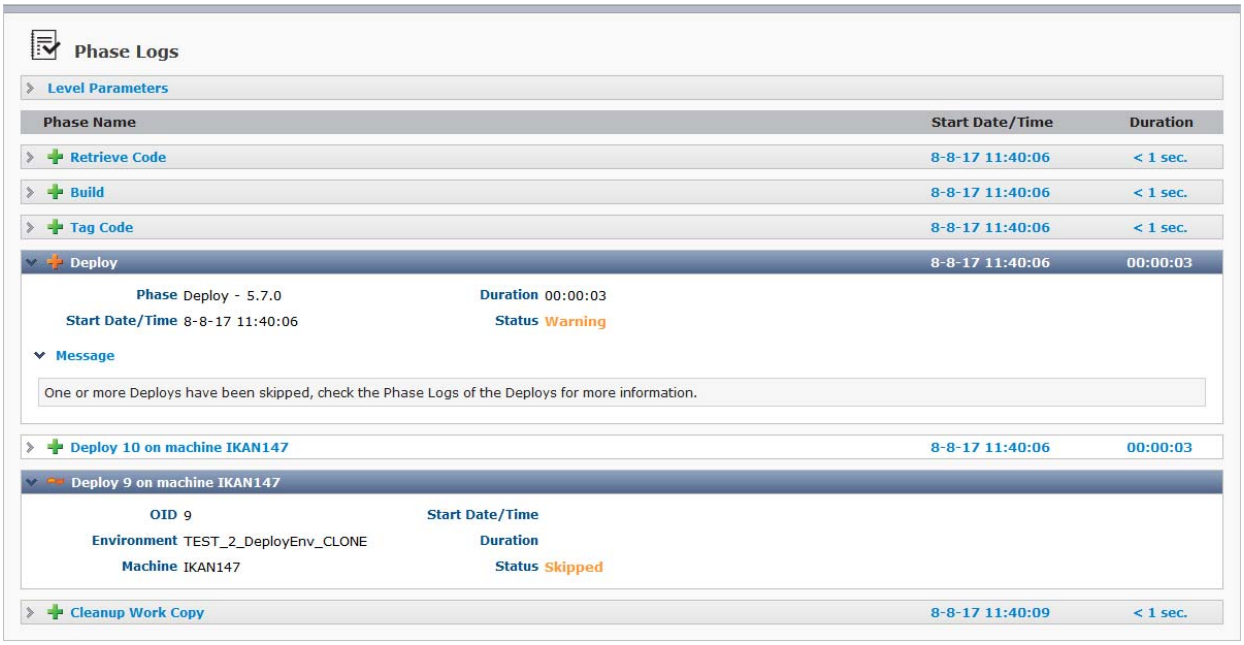


Example of a Phase with status Success



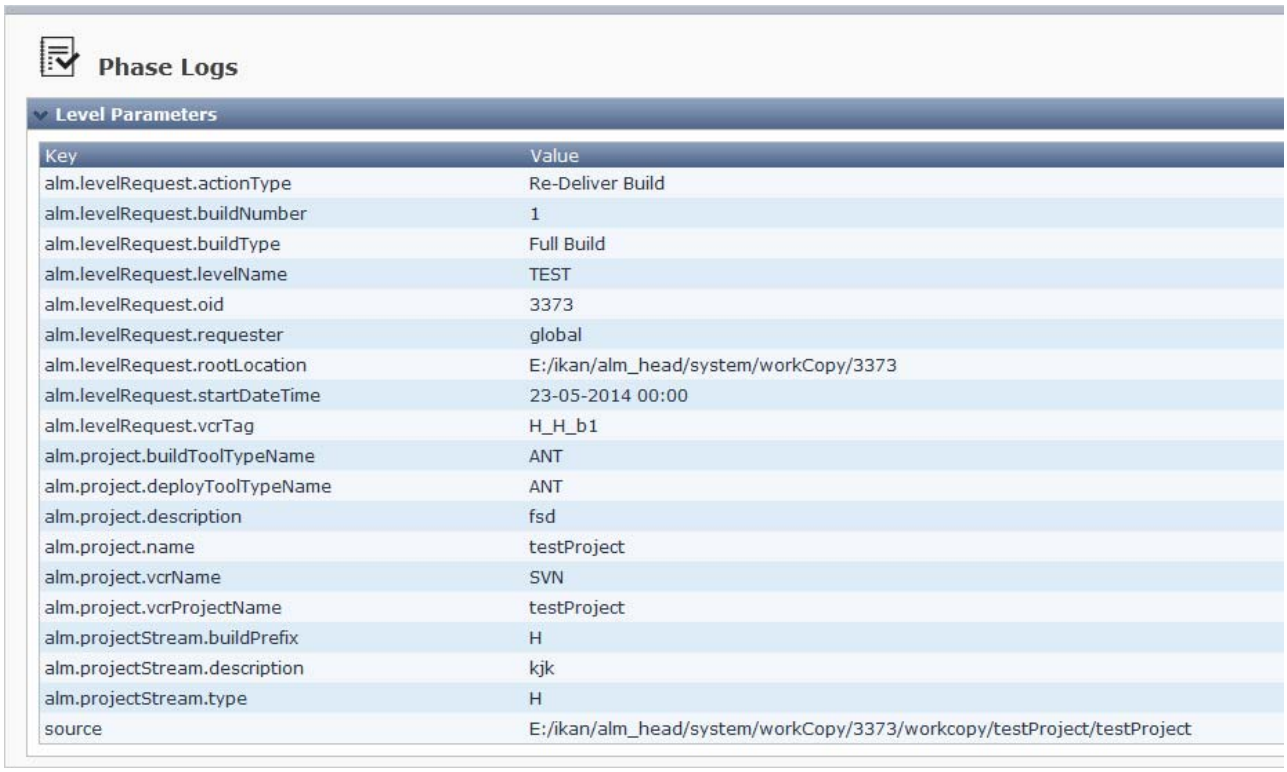
Example of a skipped Deploy Phase

In case of optional Deploys, (a) specific Deploy(s) can be skipped when creating the Level Request. The Level Request will always end with status *Warning* due to the skipped Deploy, even if the execution of the Level Request was successful.



Level Parameters

To view the Level Parameters used to execute the Level Request, click the Level Parameters heading to expand the panel.



Key	Value
alm.levelRequest.actionType	Re-Deliver Build
alm.levelRequest.buildNumber	1
alm.levelRequest.buildType	Full Build
alm.levelRequest.levelName	TEST
alm.levelRequest.oid	3373
alm.levelRequest.requester	global
alm.levelRequest.rootLocation	E:/ikan/alm_head/system/workCopy/3373
alm.levelRequest.startDateTime	23-05-2014 00:00
alm.levelRequest.vcrTag	H_H_b1
alm.project.buildToolTypeName	ANT
alm.project.deployToolTypeName	ANT
alm.project.description	fsd
alm.project.name	testProject
alm.project.vcrName	SVN
alm.project.vcrProjectName	testProject
alm.projectStream.buildPrefix	H
alm.projectStream.description	kjk
alm.projectStream.type	H
source	E:/ikan/alm_head/system/workCopy/3373/workcopy/testProject/testProject

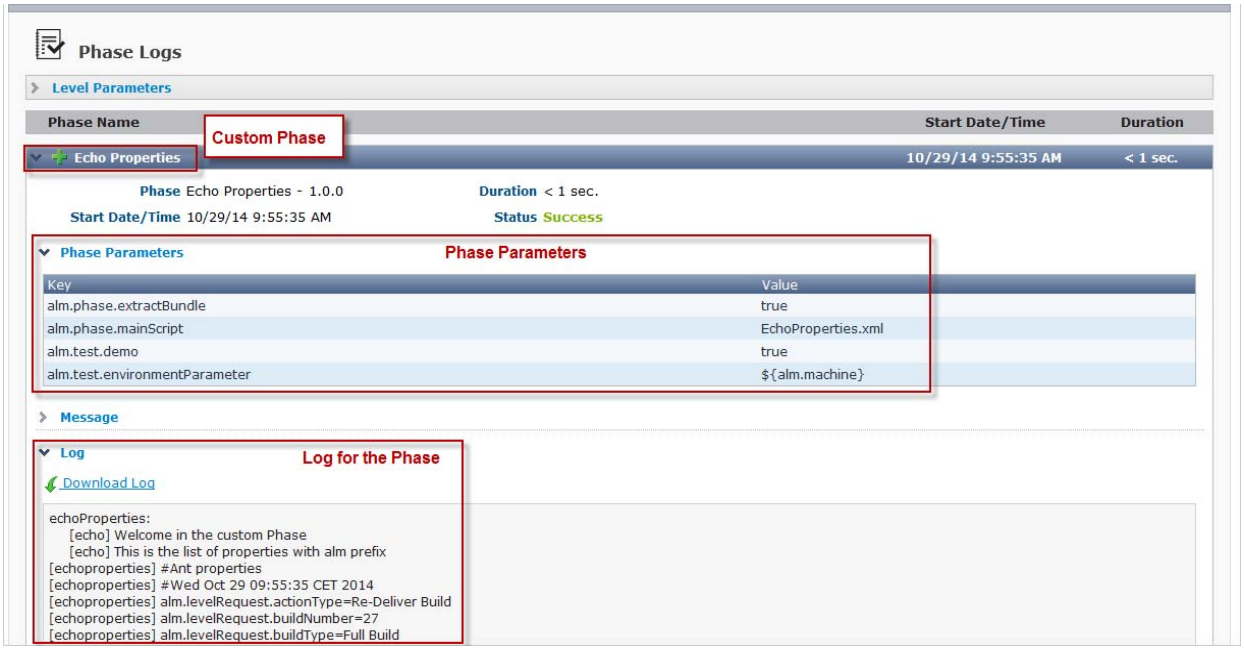
For each of the parameters, the table displays its key and value.

For more information on Level Parameters, refer to the section Predefined Level Parameters (page 503).

Level Phases

Example of a Core Phase

Example of a Custom Phase using Phase Parameters



Underneath the Level Parameters, the list of all different Level Phases and their status (*Success*, *Warning*, *Fail*, *Running*, *Not executed* or *Aborted*), Start Date/Time and Duration is displayed.

For more information, refer to Phases - General Information (page 520).

Note: An error status does not always mean that the Level Request has failed. That depends on the *Fail on Error* setting of the Level Phase which is defined on the *Edit Level Phase* screen. See Inserting a Level Phase on page 201.

Click the name of a Phase to expand its information panel. The logs of Phases in error and running status are automatically opened.

The following information is displayed for each of the Phases:

Field	Description
Phase	This field displays the name and the version of the Phase.
Start Date/Time	This field displays the date/time when the Phase execution started.
Duration	This field displays the total execution time of the Phase.
Status	This field displays the status of the Phase. The possible statuses are: *Success*, *Warning*, *Fail*, *Running*, *Not executed* and *Aborted*.

Each Phase may contain the following subpanels:

- Phase Parameters (only for Custom Phases)
- Message
- Stack Trace (only in case of a Fail)
- Log (only for Phases executed by a Scripting Tool)

  The Log can be downloaded if the Log Format type of the Scripting Tool is set to *TXT*.

Level Phases are handled by the IKAN ALM Server. By default, the Level Phases listed in the following table may be available. However, since the defined Level workflow can be altered (See The Level Phases Overview Screen on page 196.), not all Phases may be present in the Phases Log.

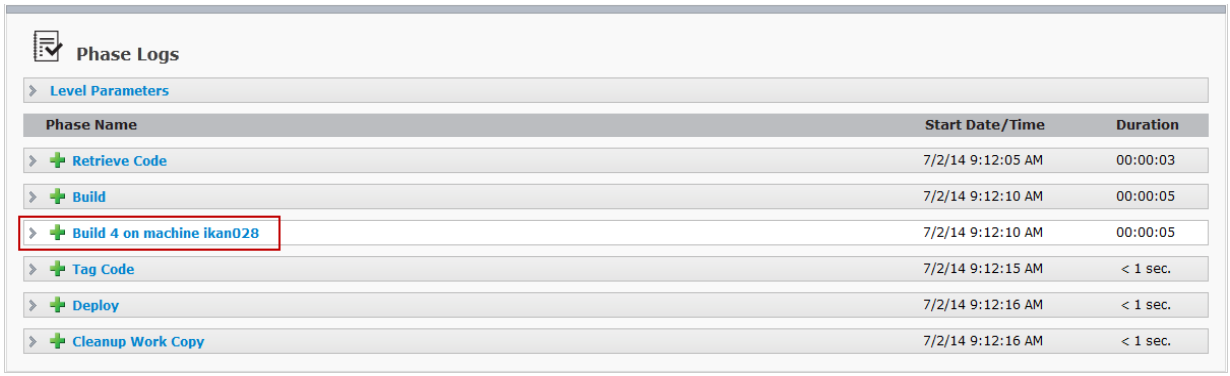
Note: The description in this table is very concise. For a more detailed description, refer to the section Level Phases (page 522).

Field	Description
Retrieve Code	The IKAN ALM Server retrieves the source code from the VCR (and sometimes, in the case of certain dependencies, also the build results from the Build Archive) and stores it in a sub folder of the Work Copy Location so that the Agent can use it to perform a build.
Build	This Phase monitors the status of the Builds linked to the Level Request. The IKAN ALM Server notifies and does the follow-up of the IKAN ALM Agents that execute the Builds linked to the Level Request.
Tag Code	The IKAN ALM Server tags the source code after a successful Build on a Build Level. The tagging will not be performed for a Build Level in a Tag-based Project Stream, nor for Deliver or Rollback Level Requests.
Deploy	This Phase monitors the status of the Deploys linked to the Level Request. The IKAN ALM Server notifies and does the follow-up of the IKAN ALM Agents that execute the Deploys linked to the Level Request.
Link File Revisions	This is a specific Phase for Levels in Package-based Projects, which is absent for Release-based Projects. Based on the contents of the Package at the moment of the Level Request execution, the IKAN ALM Server registers the checked-out versions extracted from the VCR for the Level. This information allows verifying the contents (i.e., the linked File Revisions) of the Package at the moment of Level Request execution, given the fact that this contents may vary in time.
Issue Tracking	This is a specific Phase for Projects linked with an Issue Tracking System. Depending on the settings in the Issue Tracking system (Issue Tracking (page 381)), on the VCR type and on the level type, different actions may be logged. For Build Level Requests, the log will contain Issue matching information from commit comments in the VCR. For Deliver Level Requests, it will contain the enumeration of Issues since the previous Deliver to the Level.
Cleanup Work Copy	The IKAN ALM Server cleans up the Work Copy folder where the source files (and other stored artifacts) have been checked out from the VCR in the *Retrieve Code* Phase.

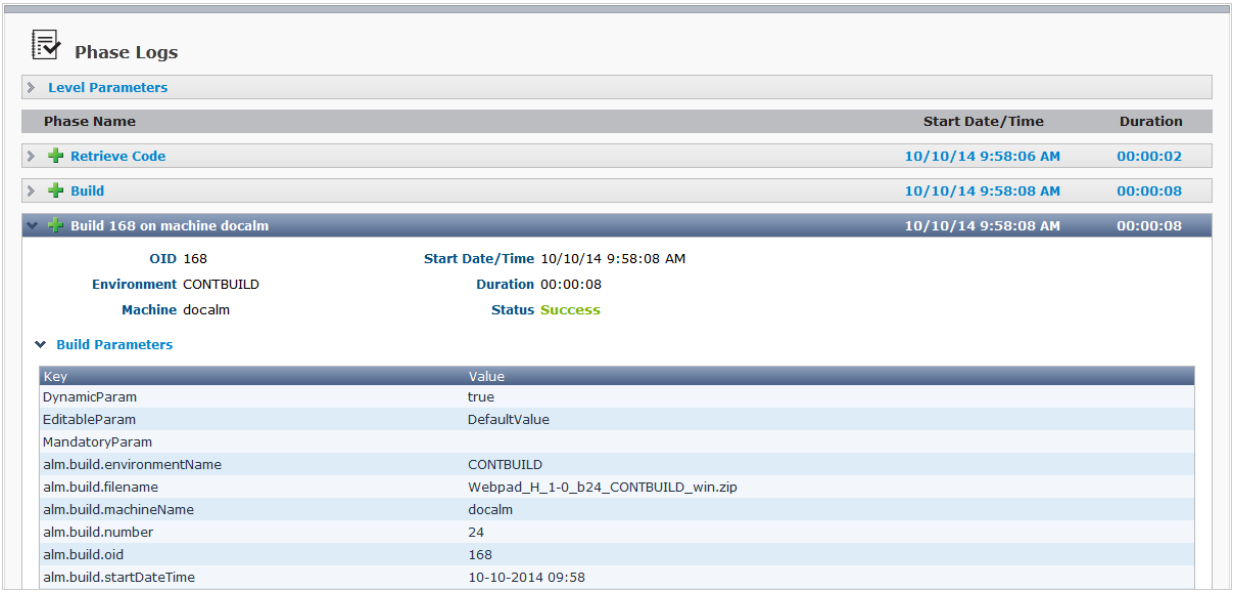
Build Actions

Build Actions are displayed on a white background in the Phase Logs list. Their description contains the Build OID and the name of the Machine on which they are executed. Their Start Date/Time and duration are also indicated.

Example of a successful Build Action



Example showing the list of Build Parameters used during the Build Action



If you open the information panel of a Build Action, the following information is displayed.

Field	Description
OID	The OID of the Build.
Environment	The name of the Build Environment linked to the Level Request.
Machine	The name of the Machine.
Start Date/Time	This field displays the date/time when the Build started.
Duration	This field displays the total execution time of the Build.

Field	Description
Status	The status of the Build. Possible status indications are: • *Wait* The Build is waiting to be started. • *In progress* The Build is ready to be started. • *Run* The Build is currently being executed. • *Success* The Build has finished successfully. • *Warning* The Build has finished successfully, but there were some non-critical errors. • *Fail* The Build has failed. • *Canceled* The Build was canceled before it was executed. • *Aborted* The Build was aborted while it was being executed. • *Rejected* The Build was rejected.
Used Build Parameters	Click this link to display all the Build parameters used during the execution of the Level Request. For each of the parameters, the table displays its key and value. For more information, refer to the section Predefined Build Parameters (page 505).

Build Phases

Build Phases are executed by the IKAN ALM Agent while handling a Build.

Each of these Phases have a status indication. An *Error* status does not always mean that the Build as a whole has failed. That depends on the *Fail on Error* setting of the Build Environment Phase, defined on the *Edit Build Environment Phase* screen. See Editing a Build Environment Phase on page 219.

The Phases listed in the following table may be available. However, since the defined Build Environment workflow can be altered (The Build Environment Phases Overview Screen (page 216)), not all Phases may be present in the Phase Log.

The description in this table is very concise. For a complete description of the Build Phases, refer to Phases - General Information (page 520).

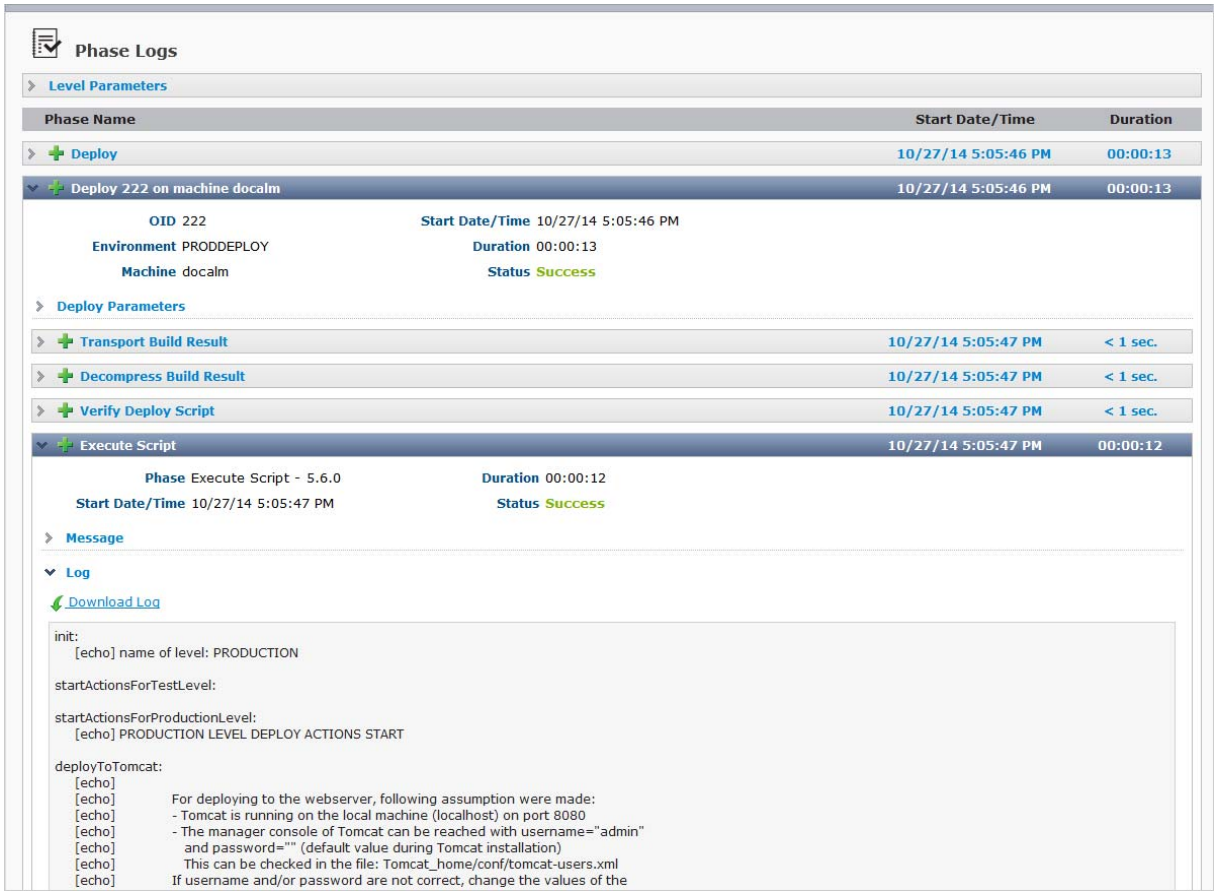
Phase	Description
Transport Source	The *Transport Source Phase* transports the required Source code from the Work Copy Location on the IKAN ALM Server to the Build Environment Source Location. It logs what was transferred and which Transporter was used in the *Message* field.

Phase	Description
Verify Build Script	The *Verify Build Script Phase* tries to locate the defined build script in the checked-out source code or in the IKAN ALM Script Location. It fails when it can't find a build script. It logs where it found the Build Script in the *Message* field.
Execute Script	The *Execute Script Phase* will execute the Build Script using the defined Scripting Tool and Build Parameters. It logs the target folder that contains the Build results in the *Message* field. It also logs the full output as generated by the Build Script in a separate *Log* panel under the *Message* field.
Transport Deploy Script	If a Deploy Script is available on the Source Location of the Build Environment, this Deploy Script is copied to the Target Location. It logs where it found the deploy script in the *Message* field.
Compress Build	The *Compress Build Phase* compresses the Build result files in the Target Location. It logs the full path of the compressed file in the *Message* field.
Archive Result	The *Archive Result Phase* copies the compressed Build file to the Build Archive Location on the IKAN ALM Server. It logs what was transferred and which Transporter was used in the *Message* field.
Cleanup Source	Cleaning up means that all files in the Build Environment Source location (Source files, Build Script and possibly Deploy Script) are deleted. It logs the result of the cleanup in the *Message* field.
Cleanup Result	Cleaning up means that all files in the Build Environment Target location (uncompressed and compressed Build files as well as the Deploy Script) are deleted. It logs the result of the cleanup in the *Message* field.

Deploy Actions

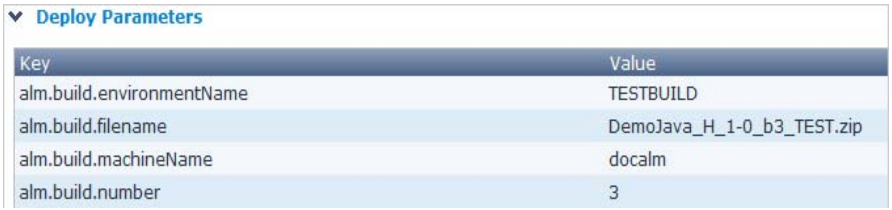
Deploy Actions are displayed on a white background in the Phase Logs list. Their description contains the Deploy OID and the name of the Machine on which they are executed. Their Start Date/Time and duration are also indicated.

Example of a Deploy with status Success



Example of a Deploy with status Warning

If you open the information panel of a Deploy Action, the following information is displayed.

Field	Description
OID	The OID of the Deploy.
Environment	The name of the Deploy Environment linked to the Level Request.
Machine	The name of the Machine.
Start Date/Time	This field displays the date/time when the Deploy started.
Duration	This field displays the total execution time of the Deploy.
Status	The status of the Deploy. Possible status indications are: • *Wait* The Deploy is waiting to be started. • *In progress* The Deploy is ready to be started. • *Run* The Deploy is currently being executed. • *Success* The Deploy has finished successfully. • *Warning* The Deploy has finished successfully, but there were some non-critical errors. • *Fail* The Deploy has failed. • *Canceled* The Deploy was canceled before it was executed. • *Aborted* The Deploy was aborted while it was being executed. • *Rejected* The Deploy was rejected.
Used Deploy Parameters	Click this link to display all the Deploy parameters used during the execution of the Level Request. For each of the parameters, the table displays its key and value. For more information, refer to the section Predefined Deploy Parameters (page 508). 

Deploy Phases

Each of these Phases have a status indication. An Error status does not always mean that the Deploy as a whole has failed, this depends on the *Fail on Error* setting of the Deploy Environment Phase, defined on the *Edit Deploy Environment Phase* screen. See Editing a Deploy Environment Phase on page 234.

The Phases listed in the following table may be available. However, since the defined Deploy Environment workflow can be altered (The Deploy Environment Phases Overview Screen (page 232)), not all Phases may be present in the Phase Log.

The description in this table is very concise. For a complete description of the Deploy Phases, refer to Phases - General Information (page 520).

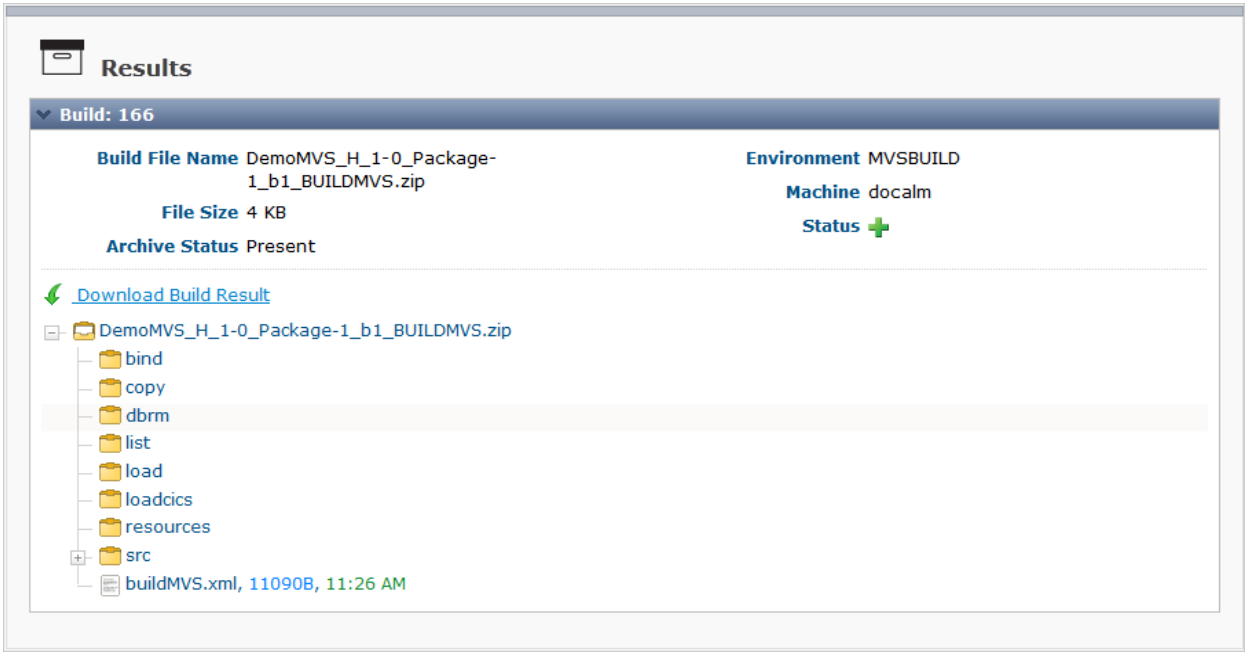
Phase	Description
Transport Build Result	The *Transport Build Phase* transports the Build from the Build Archive on the IKAN ALM Server to the Deploy Environment Source Location. It logs what was transferred and which Transporter was used in the *Message* field.
Decompress Build Result	The *Decompress Build Result Phase* decompresses the Build file into the Deploy Environment Source Location. It logs where the Build results were decompressed in the *Message* field.
Verify Deploy Script	The *Verify Deploy Script Phase* tries to locate the defined deploy script in the decompressed Build result or in the IKAN ALM Script Location. It fails when it can't find a deploy script. It logs where it found the deploy script in the *Message* field.
Execute Script	The *Execute Script Phase* will execute the Deploy Script using the defined Scripting Tool and Deploy Parameters. It logs the target folder of the Deploy in the *Message* field. It also logs the full output as generated by the Deploy Script in a separate *Log* panel under the Message field.
Cleanup Build Result	The *Cleanup Build Result Phase* cleans up the Build files on the Source Location for this Deploy Environment. It logs the result of the cleanup in the *Message* field.

RELATED TOPICS

- Level Request Detail (page 65)
- Summary (page 66)
- Results (page 89)
- Approvals (page 91)
- Issues (page 93)
- Sources (page 97)
- Modifications (page 98)
- Dependencies (page 99)

Results

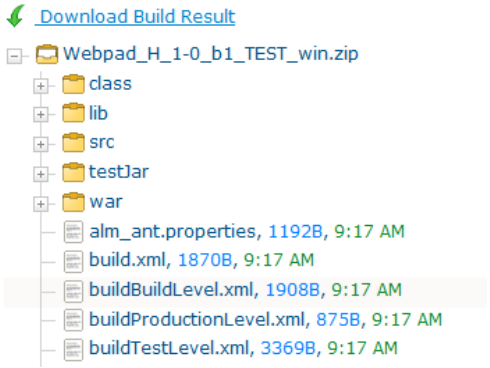
On the *Results* page you can (for each Build related to the Level Request) display the content of the compressed Build Result File, i.e., the files that were present in the Target Location at the moment the Build script ended and which are now stored in the Build Archive.



If the option *Downloadable Build* is activated for the Environment, you can also download the Build File or view its composition using a File Explorer.

The following information is displayed.

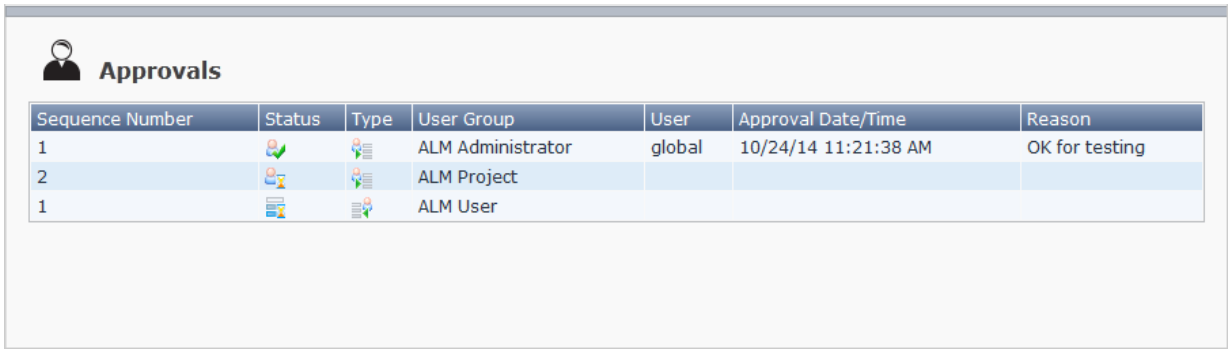
Note: If there are no Builds, the following information will be displayed:

Field	Description
Build File Name	The name of the Build File that holds the compressed Build results. This file is stored in the IKAN ALM Build Archive Location. If the option Downloadable Build (page 212) is activated for this Build Environment, this name is displayed as a hyperlink. Click the hyperlink to save a local copy of the Build file on your system. This name may have the following format: *WEBPAD_H_1-0_b9_CONTBUILD_win.zip* where: • *WEBPAD* = Project Name • *H* = Stream type indication: H = Head, B = Branch • *1-0* = Build Prefix • *b9* = Build Number • *CONTBUILD* = Build Environment name • *win* = Build Environment suffix • *zip* = filename extension. For Builds on a Windows Machine, this is .zip, for Linux/Unix Builds, the extension is .tar.gz. If the Build failed, this field is empty
File Size	The size of the Build File (in bytes, kB, MB, GB or TB).
Archive Status	The status of the Archive. To be able to download the Build file, the archive must be *Present* in the IKAN ALM database. Possible values are: • *Present* • *Non Existing*: In case the Build is not executed yet, or when the Build failed • *Deleted*: Removed from the Build Archive with the Housekeeping functionalities
Environment	The name of the Build Environment where this Build was executed.
Machine	The name of the Machine hosting the Build Environment where this Build was executed.
Status	The status of the Build that produced the result.
Download Build Result	This link allows you to download the Build Result or to open it using a File Explorer. Note that this is only possible if the option *Downloadable Build* has been activated on the Build Environment. For more information, refer to the section Editing a Build Environment (page 213). If not, you can always visualize the tree by clicking the plus sign in front of the zip file. 

RELATED TOPICS
- Level Request Detail (page 65)
- Summary (page 66)
- Phase Logs (page 77)
- Approvals (page 91)
- Issues (page 93)
- Sources (page 97)
- Modifications (page 98)
- Dependencies (page 99)

Approvals

The *Approvals* page displays the status of the Approvals that are defined for the Level Request. The Approvals are ordered according to the sequence in which they should be approved.



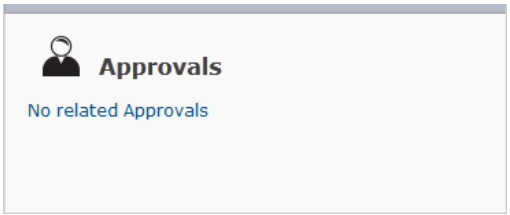
Before the Level Request can start, all Pre-Approvals need to be approved. Post-approvals need to be given at the end of the Level Request and enable to approve or reject the results or actions of the Level Request. Approvals can be approved or rejected by the Users belonging to the User Group of the Approval.

For more information on defining Approvals on Levels, refer to Level Approvals (page 205).

Note: A user can check whether he/she still has to approve/reject an Approval by using the Outstanding Approvals menu option. See Approvals on page 21.

The following information is displayed:

Note: If there are no Approvals, the following information will be displayed:



Field	Meaning
Sequence Number	This field contains the *Sequence Number* of the Approval. The Approvals are ordered per type.

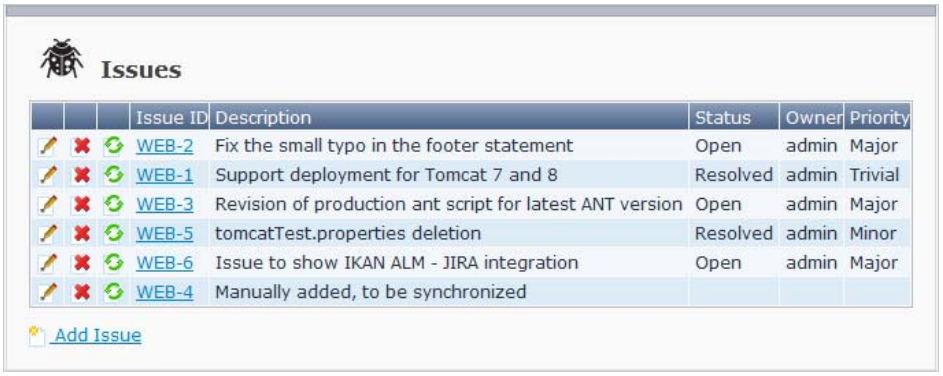
Field	Meaning
Status	This field indicates the Approval Status. Possible status indications are: • Awaiting Approval • Awaiting Level Request Finish: This status is assigned to a Post-Approval when a Level Request starts, and is changed to "Awaiting Approval" when the Level Request ends with status Success or Warning. • Awaiting Predecessor Approval: this status is assigned to the Approval when a preceding Approval on the Level Request must be granted first. • Approved • Rejected • Canceled: when a previous Level Approval was rejected, or when the Level Request was canceled or aborted.
Type	This field indicates the Approval Type. Possible Approval Types are: • Pre-Approval • Post-Approval
User Group	This field contains the name of the User Group whose members can grant or reject the Approval. This matches the settings on the *Level Approvals Overview* screen (Level Approvals (page 205)).
User	This field contains the User ID of the User who granted or rejected the Approval. This field is empty for waiting Approvals.
Approval Date/Time	This field contains the Approve or Reject Timestamp. This field is empty for waiting Approvals.
Reason	This field contains the comment the User entered in the *Reason* field when he or she granted or rejected the Approval. For some special events, like Level Request cancellations, abortions or failures, this reason is automatically provided by IKAN ALM.

RELATED TOPICS

- Level Request Detail (page 65)
- Summary (page 66)
- Phase Logs (page 77)
- Results (page 89)
- Issues (page 93)
- Sources (page 97)
- Modifications (page 98)
- Dependencies (page 99)

Issues

The *Issues* page displays the list of Issues that have been linked to a Level Request. Issues can get linked to a Level Request automatically by IKAN ALM during the execution of the Level Request, or manually by a User after the Level Request has ended.



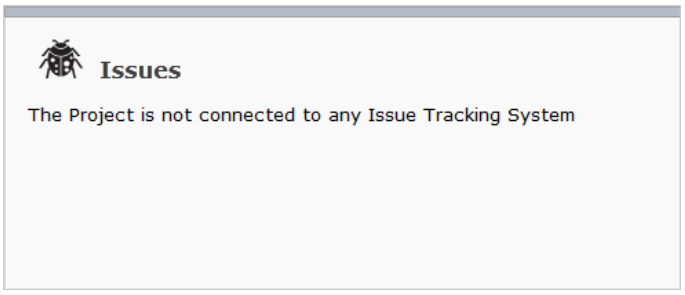
Refer to the section Issue Tracking (page 381)for more information on defining on external Issue Tracking System, and to Phases - General Information (page 520)for more information on the Issue Tracking Phase.

Note: Issue information is only visible when the Project of the Level Request is linked to an Issue Tracking System. See Editing Project Settings on page 149.
If the project is connected to an HP Quality Center, Atlassian JIRA, Collabnet TeamForge or Microsoft Team Foundation Issue Tracking System, the description, status, owner and priority information will automatically be filled in based on the Issue information in the Issue Tracking System concerned. For other Issue Tracking Systems, you can add the information manually.

The following information is available:

Field	Meaning
Issue ID	This field displays the ID by which the Issue is defined in the external Issue Tracking System. If the URL field of the Issue Tracking System definition is not empty, this field will be displayed as a link. Click the link to directly view the Issue in the external Issue Tracking System's Web interface. Refer to Creating an Issue Tracking System (page 382)for more information on the URL field.
Description	This field displays the Description of the Issue.
Status	This field displays the Status of the Issue
Owner	This field displays the owner of the Issue.
Priority	This field displays the Priority of the Issue.

Note: If there are no Issues, the following information will be displayed:



Note: Apart from the Issue ID, the information in the fields may not be automatically filled in when IKAN ALM synchronizes the Issues with the Issue Tracking System. In that case, the values need to be provided manually by editing the Issue.

Use the appropriate link for editing, deleting or adding an Issue.

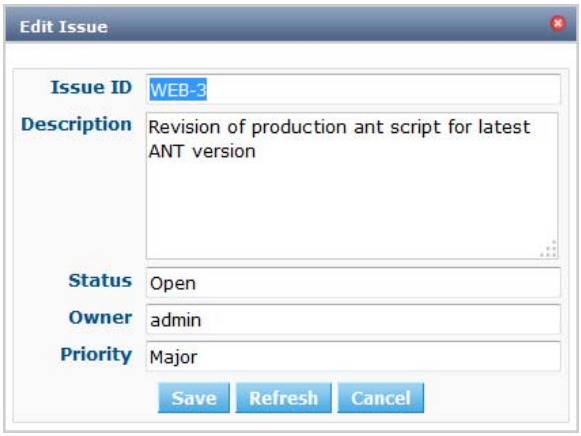
Note: These links are only available for Build Levels.

- *Edit*. See Editing an Issue on page 94.
- *Delete*. See Deleting an Issue on page 95.
- *Synchronize*. See Synchronizing an Issue on page 95.
- *Add Issue*. See Adding an Issue on page 96.

Editing an Issue

This option is provided to manually update the fields of an Issue linked to this Level Request.

1. Click the *Edit* link.
   The *Edit Issue* screen is displayed:



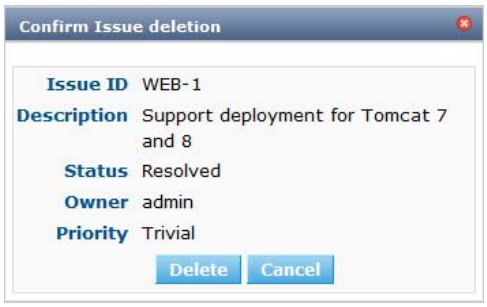
2. Change the fields as required.
   For a detailed description of the fields, refer to Adding an Issue (page 96).

3. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Cancel* to return to the previous screen without saving your changes.

Deleting an Issue

This option is provided to manually update the fields of an Issue linked to this Level Request.

1. Click the *Delete* link.
   The *Confirm Issue Deletion* screen is displayed:



2. Click *Delete* to confirm the Issue Deletion.
   You can also click *Cancel* to return to the *Issues* page without deleting the Issue.

Synchronizing an Issue

Note: This option is only available for Atlassian JIRA, HP Quality Center, Collabnet TeamForge and Microsoft Team Foundation.

HP Quality Center Defects, JIRA Issues, TeamForge and Team Foundation Artifacts may be linked to an IKAN ALM Level Request, based on the comments provided by the developers when they commit their code to the VCR. IKAN ALM retrieves information from the corresponding Issue, such as a short description, the owner and the priority. This information is synchronized during the Issue Tracking Phase each time the Level Request is delivered to the next Level in the IKAN ALM Lifecycle.

The *Synchronize* option is provided to manually synchronize the Status, Owner and Priority fields of an Issue linked to this Level Request. This may be useful when Issues are manually added to the Level Request, e.g., when they were omitted by the developers in the commit comments.

1. Click the *Synchronize* link.

2. If the synchronization succeeds, the *Status*, *Owner* and *Priority* fields will be updated and a message confirming the successful synchronization is displayed at the top of the screen.
   If the synchronization fails, an error message indicating the problem will be displayed.

Adding an Issue

This option is provided to manually link Issues to a Level Request.

This may be necessary when the automatic process did not detect some Issues, or when Issues need to be linked with Level Requests that were executed at a time when an external Issue Tracking System was not yet properly configured in IKAN ALM.

1. Click the *Add Issue* link.

   The *Add Issue* screen is displayed:

The following fields are available. Fields marked with an asterisk are mandatory.

Field	Description
Issue ID	Enter the number by which the Issue is defined in the external Issue Tracking System. This value is used in the *Related Issues* panel on the *Level Request Detail* screen to provide the direct link to the Issue in the external Issue Tracking System's Web interface. This field is mandatory
Description	Enter text describing the Issue. This field is optional.
Status	Enter the Status of the Issue (e.g.: open, closed, in progress, …) This field is optional.
Owner	Enter the Owner (or reporter) of the Issue. This field is optional
Priority	Enter the Priority of the Issue (e.g.: high, medium, low,…) This field is optional.

Note: For JIRA Issues, HP Quality Center Defects, TeamForge and Team Foundation Artifacts, it's sufficient to enter the Issue ID. After having created the Issue, synchronize the issue (Synchronizing an Issue (page 95)) to automatically retrieve the additional fields from the Issue Tracking System.

2. Fill out the fields as required and click *Create*.

   You can also click:

   - *Reset* to retrieve the settings from the database.
   - *Cancel* to return to the previous screen without saving your changes.

RELATED TOPICS
- Level Request Detail (page 65)
- Summary (page 66)
- Phase Logs (page 77)
- Results (page 89)
- Approvals (page 91)
- Sources (page 97)
- Modifications (page 98)
- Dependencies (page 99)

Sources

The *Sources* page displays the list of all the sources (and their revision numbers) in the VCR used by this Level Request.

For Builds based on latest code, these files were tagged by IKAN ALM in the Tag Code Phase of this Level Request.

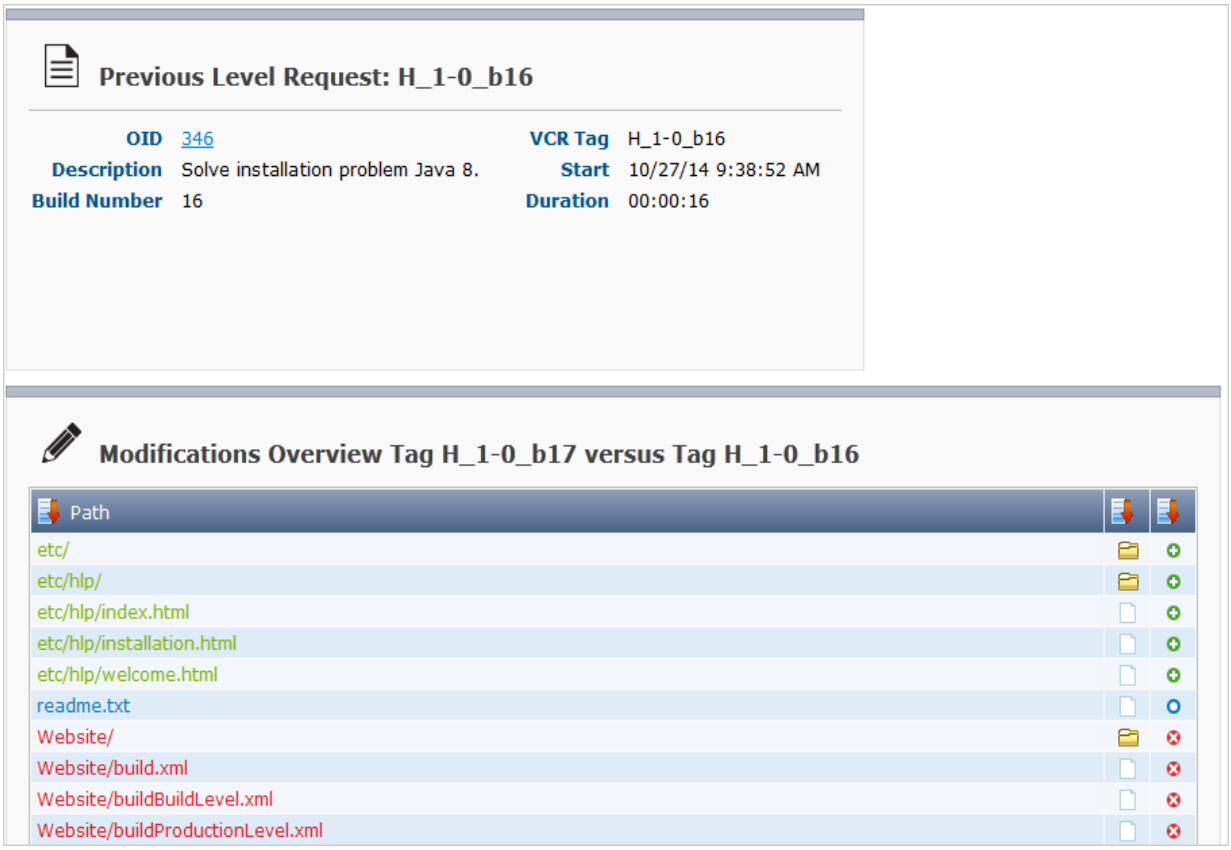
For Builds based on tagged code, these files were already tagged by either a previous Level Request on the Build Level of this Lifecycle or, in the case of a Tag-Based build, by an external system.

Note: This information is only visible if the Level Request has ended with status *Success* or *Warning*.

RELATED TOPICS
- Level Request Detail (page 65)
- Summary (page 66)
- Phase Logs (page 77)
- Results (page 89)
- Approvals (page 91)
- Issues (page 93)
- Modifications (page 98)
- Dependencies (page 99)

Modifications

The *Modifications* page allows you to verify the differences between the sources used to create the current Build and the ones used for the Build generated by the previous Level Request. Although there is no Build on Levels with only Deploy Environments, this information will still be visible for Level Requests on such Levels, since in this case there is always a Build that is deployed to that Level.



The list of modifications shows additions in green, modifications in blue and deletions in red. The Type (File or Directory) and Modification Type are displayed in separate columns.

The following information is displayed for the previous Level Request:

Field	Meaning
OID	This field displays the OID of the previous Level Request. When clicking this OID link, the detail of the previous successful build will be displayed.
Description	This field displays the description of the previous Level Request.
Build Number	This field displays the Build number of the previous Level Request.
VCR Tag	This field contains the VCR Tag of the previous Level Request. This Tag matches a Build with its source code in the VCR. The format of the VCR Tag normally matches the Tag Template defined for the Stream. See Project Streams on page 161. However, the user can override the default VCR Tag while creating a Level Request, so that the Tag Format can be completely different.
Start	This field displays the date/time when the previous Level Request started.
Duration	This field displays the total execution time of the previous Level Request.

Note: Modifications cannot be shown if there is no previous successful Level Request to compare with. On the Build Level, modifications are only shown for the statuses *Warning* or *Success*.

RELATED TOPICS
- Level Request Detail (page 65)
- Summary (page 66)
- Phase Logs (page 77)
- Results (page 89)
- Approvals (page 91)
- Issues (page 93)
- Sources (page 97)
- Dependencies (page 99)

Dependencies

The Project Dependency functionality makes it possible to reuse common libraries or sources from Child Projects.

Projects can be reused in two ways:

- as a Build result retrieved from the Build Archive, or
- as sources retrieved from the Versioning System.

In the case of **reuse of a Build result**, there will be a link between the **parent Build** object and the **child Build** object.

In the case of **reuse of sources**, there will be a link between the **parent Build** object and the **child Level Request** object.

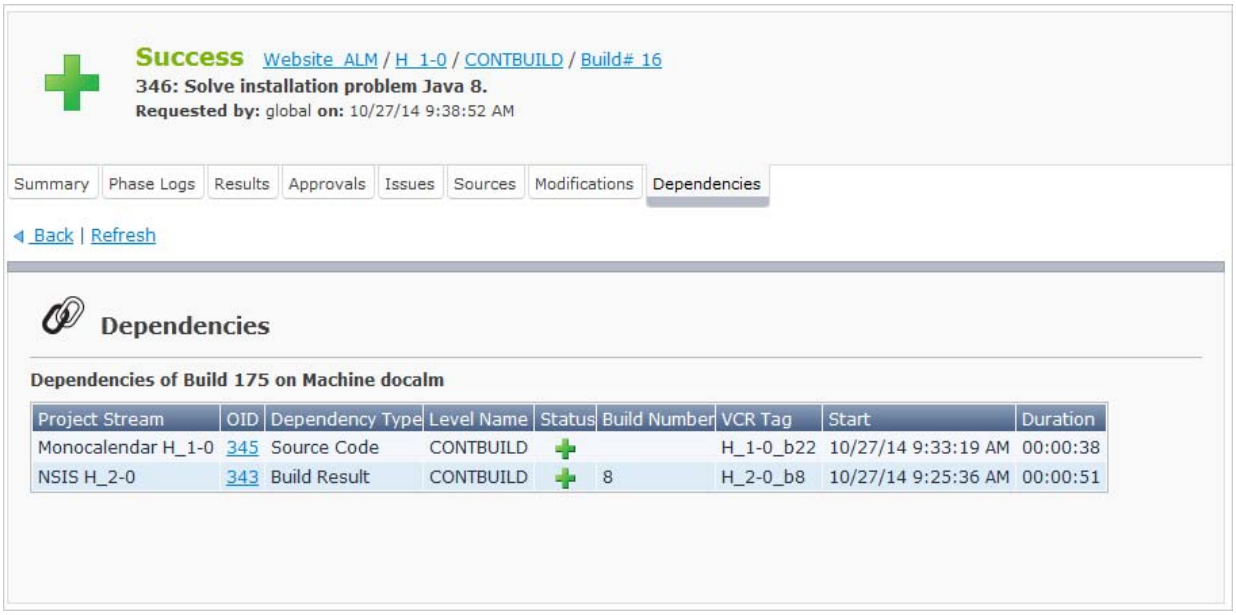
Note: Project Stream dependencies are defined on the Edit Project Stream Dependency Detail screen using the Code Retrieval field. See Editing a Project Stream Dependency on page 173.

The *Dependencies* page is the starting point to find all the information concerning:

1. Build Dependencies (page 100)
2. Build Usage (page 101)
3. Level Request Usage (page 103)

Build Dependencies

The *Dependencies* page displays the specific Builds and/or Sources of the Child Projects where this Build depends on. Note that the transitive dependencies of a Child whose sources are used will also be displayed, since they were also collected in the *Retrieve Code* Phase.



Note: If there are no Dependencies, the following information will be displayed:

Per Build, a Dependency table will be displayed.

1. Verify the information on the *Dependencies* table.

   The following information fields are available:

Field	Meaning
Project Stream	This field displays the Name followed by the Build Prefix of the Child Project Stream.
OID	This field displays the Level Request OID of the Child Build. Click the link to display the details for this Level Request.

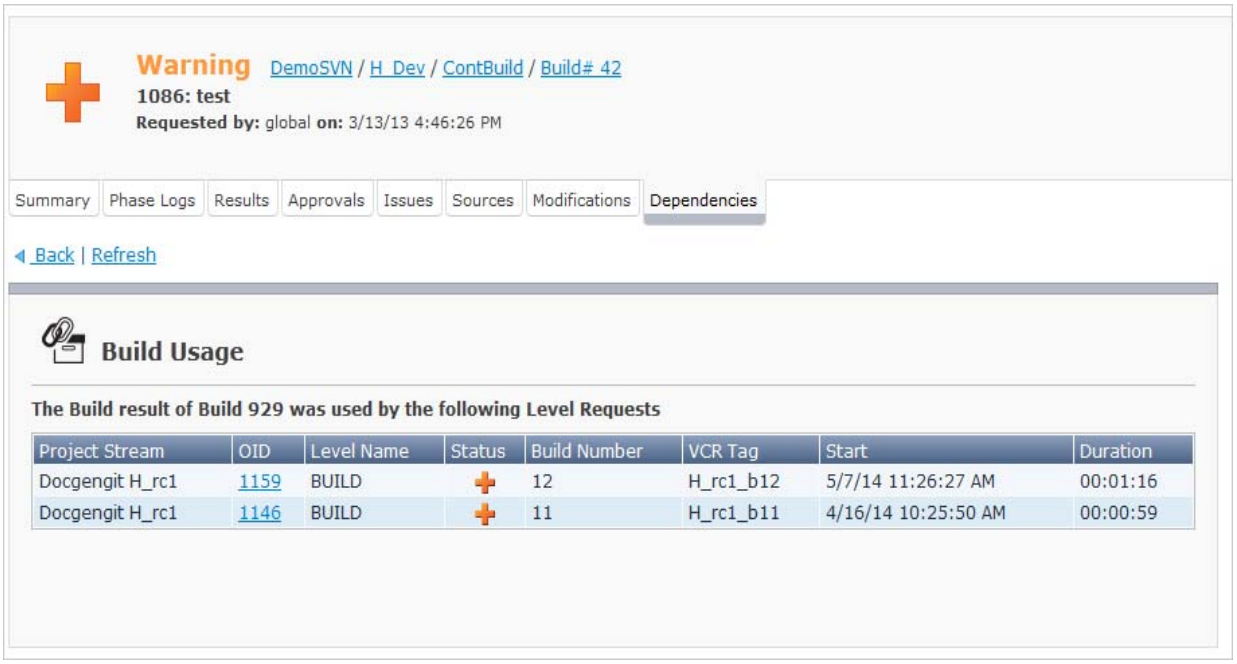
Field	Meaning
Dependency Type	This field displays the Dependency Type of the Build Dependency. Possible types are: • *Source Code*: the sources of the Child Build are used • *Build Result*: the Build result of the Child Build is used See the description of the *Code Retrieval* field in the section Editing a Project Stream Dependency (page 173).
Level Name	This field displays the name of the Level.
Status	This field displays the status of the Level Request of the Child Build.
Build Number	This field displays the Build Number of the Child Build. This field is empty if the Dependency Type is *Source Code*.
VCR Tag	This field displays the VCR Tag of the Level Request of the Child Build.
Start	This field displays the date/time when the Level Request of the Child Build started.
Duration	This field displays the total execution time of the Level Request of the Child Build.

2. You click through on one of the *OID* links to display its Level Request Details.

Build Usage

In the case of **reuse of a Build result**, there will be a link between the **parent Build** object and the **child Build** object.

In that case, the *Build Usage* panel provides information concerning the Level Requests using the Build Result of a specific Build.



This information is useful in case there is a problem with a child library, or with the sources of a Project which is used by another Project. It allows you to find out whether other Projects use this child library, or these child sources.

Note: This information is only visible if a parent (dependent) project has used this Level Request's Build result in a Build of its own.

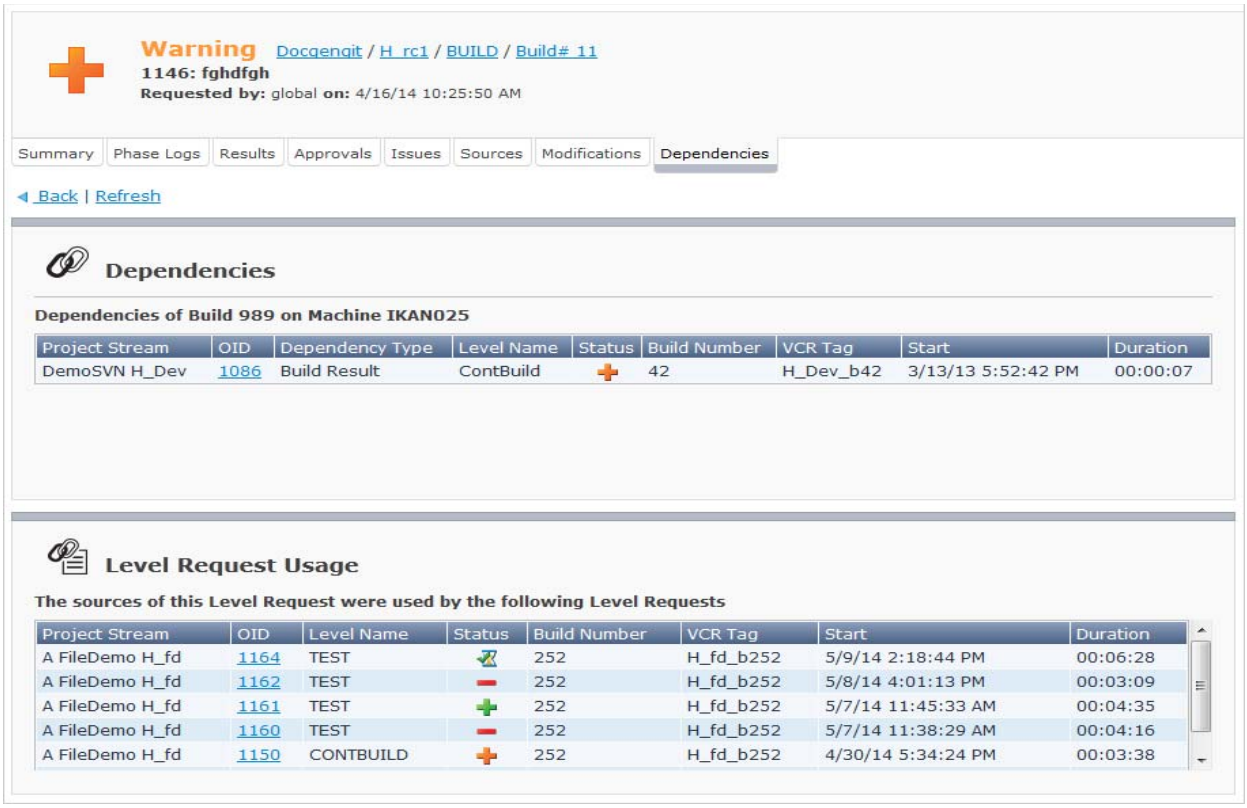
1. Verify the information on the *Dependencies* table.
   The following information fields are available:

Field	Meaning
Project Stream	This field displays the Name followed by the Build Prefix of the Parent Project Stream.
OID	This field displays the Level Request OID of the Parent Build. Click the link to display the details for this Level Request.
Level Name	This field displays the name of the Level.
Status	This field displays the status of the Level Request of the Parent Build.
Build Number	This field displays the Build Number of the Parent Build.
VCR Tag	This field displays the VCR Tag of the Level Request of the Parent Build.
Start	This field displays the date/time when the Level Request of the Parent Build started.
Duration	This field displays the total execution time of the Level Request of the Parent Build.

2. You can click through on one of the *OID* links to display its Level Request Details.

Level Request Usage

In the case of **reuse of sources**, there will be a link between the **parent Build** object and the **child Level Request** object. In that case, the *Level Request Usage* table displays the Level Requests of parent projects that use this Level Request's sources.



Note: This information is only visible if a parent (dependent) project has used this Level Request's sources in a Build of its own.

1. Verify the information on the *Level Request Usage* table.

   The following information fields are available:

Field	Meaning
Project Stream	This field displays the Name followed by the Build Prefix of the Parent Project Stream.
OID	This field displays the Level Request OID of the Parent Build. Click the link to display the details for this Level Request.
Level Name	This field displays the name of the Level.
Status	This field displays the status of the Level Request of the Parent Build.
Build Number	This field displays the Build Number of the Parent Build.
VCR Tag	This field displays the VCR Tag of the Level Request of the Parent Build.

Field	Meaning
Start	This field displays the date/time when the Level Request of the Parent Build started.
Duration	This field displays the total execution time of the Level Request of the Parent Build.

2. You can click through on one of the *OID* links to display its Level Request Details.

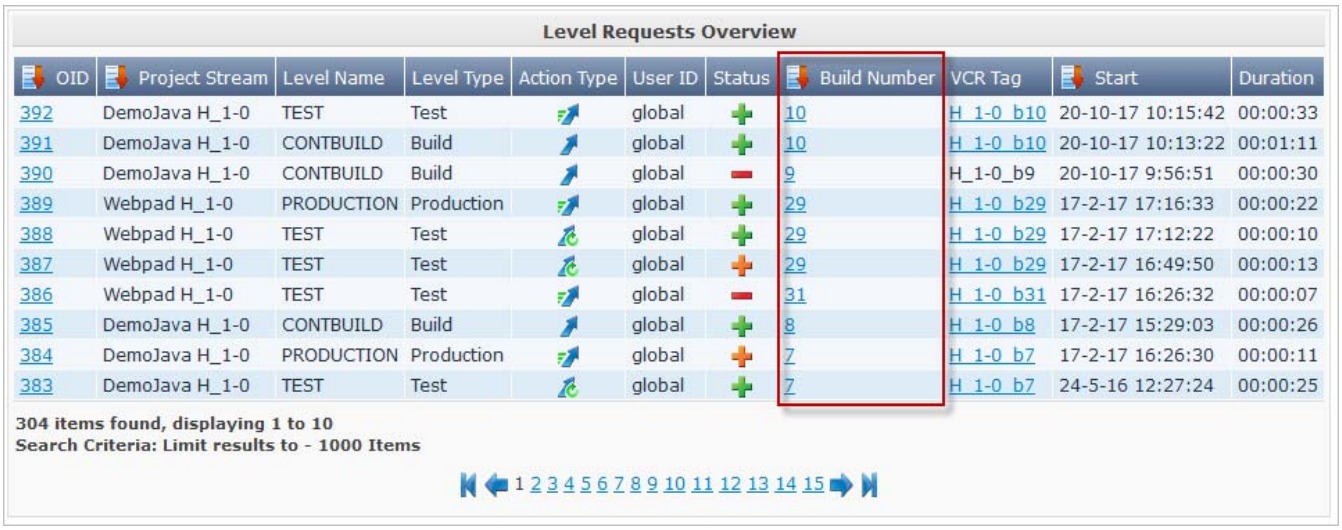
RELATED TOPICS
- Level Request Detail (page 65)
- Summary (page 66)
- Phase Logs (page 77)
- Results (page 89)
- Approvals (page 91)
- Issues (page 93)
- Sources (page 97)
- Modifications (page 98)

7.5. Build History Screen

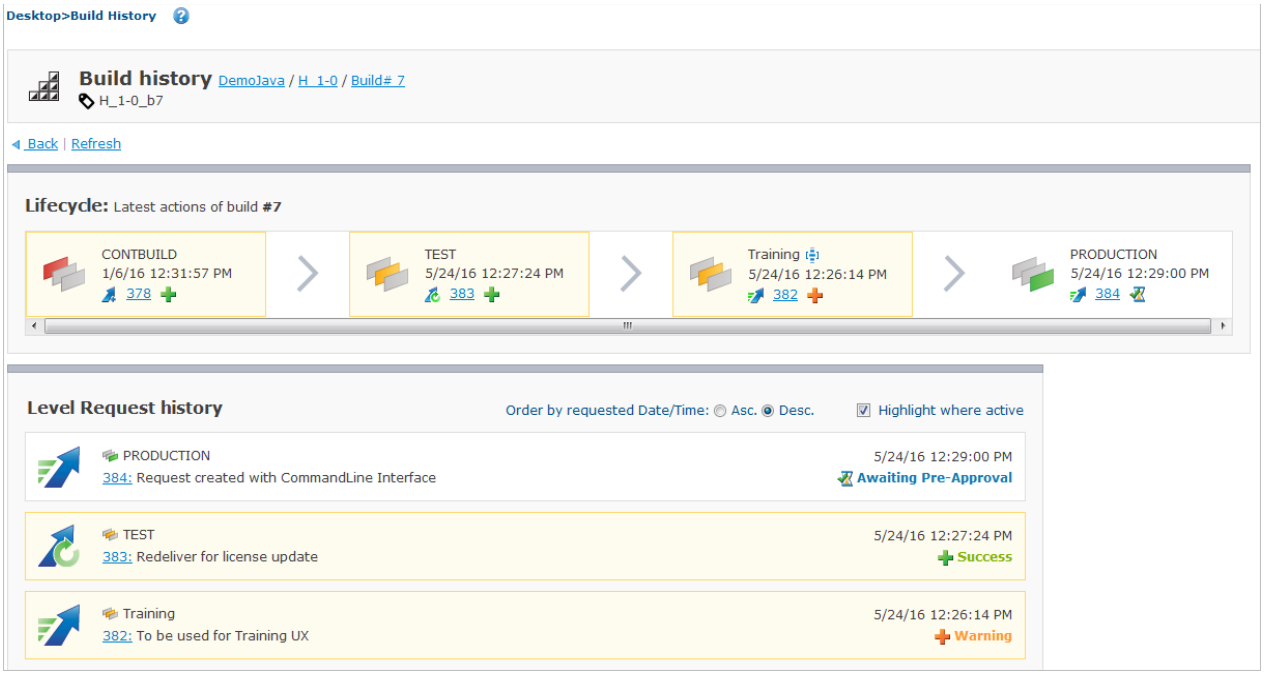
The *Build History* option allows to quickly verify the Lifecycle of a specific Build (identified by its number).

1. Select *Level Requests > Overview Level Requests* on the Main Menu.

2. Click the *Build Number* link you want to see the history for.



The *Build History* screen is displayed:



If the *Auto Refresh* option is activated, the *Build History* screen will be refreshed each time the defined rate is expired. See Auto Refresh on page 12.

Note: The *Build History* can also be accessed from the *Builds and Deploys Overview*, and from the *Level Request Detail* screen.

The screen is structured as follows:

2.1.Status Header

The header displays the Project Name, Project Stream Identification and Build number of the Level Request, and the VCR tag.

Those links lead to the *Level Request Overview* screen. Depending on the link element you select, more information will already be filled in on the *Search Level Request* panel to limit the Level Requests displayed on the overview.

2.2.Lifecycle Panel

This panel shows an ordered view of the latest Build actions for all Levels of the Lifecycle connected to the Project Stream.

For each Level, the latest Level Request of the Build is shown, with the Requested Date/Time, the Type, a link to the Level Request Detail screen and the Status. If this Level Request is also the latest for a Level, it will be highlighted. If the Build was never executed/delivered to the Level, only the Level Name will be visible.

The Level type is indicated as follows:

Icon	Description
	Build Level
	Test Level
	Production Level

If a Level is optional, it is preceded by the icon.

2.3.Level Request History

The Level Request History provides information about all Level Requests related to the selected Build. The Level Requests are ordered so that the most recent Level Request (i.e., the Level Request indicating the current situation) is displayed at the top of the list.

The following information is displayed for each of the Level Requests:

- The Action Type of the Level Request (Requested or Forced Build, Deliver or Rollback)
- The Level Type (Build, Test or Production)
- The Level Request OID

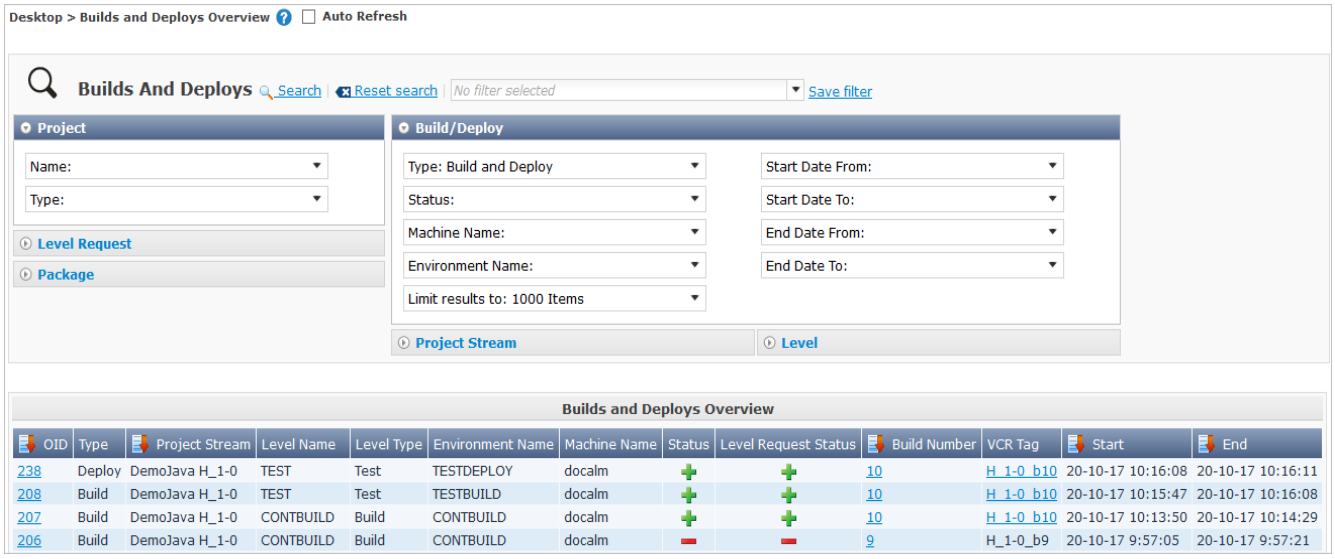
  The OID is displayed as a link. Clicking this link will display the Level Request Detail screen).
- The Requested Date/Time
- The Level Request status

The Level Requests can be sorted in ascending or descending order.

By default, the "active" Level Requests are high-lighted in a different color. A Level Request is active if it is the last successful (status is set to *Success* or *Warning*) Level Request on the Level. You can deactivate this, by deselecting the *Highlight where active* option.

7.6. Builds and Deploys Overview

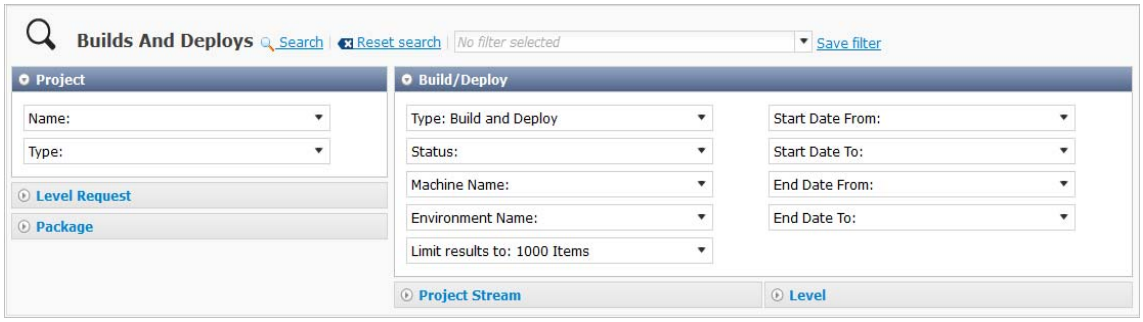
1. Select *Level Requests > Overview Builds and Deploys Level Request* on the Main Menu.
   The *Builds and Deploys Overview* screen is displayed:



Note: If the *Auto Refresh* option is activated, the *Build and Deploys Overview* screen will be refreshed each time the defined rate is expired. See Auto Refresh on page 12.

2. Use the search criteria on the *Search Builds and Deploys* panel to only display the Builds and/or Deploys you are looking for.

Note: You can select multiple statuses and/or action types on the select boxes using the *SHIFT* or *CTRL* key.



The following options are available:

- Show advanced options: to display all available search criteria.
- Search: in principle it is not necessary to click the *Search* option. The results on the overview will be automatically synchronized in function of the selected criteria.
- Reset search: to clear all search criteria and display the full list of items.
- Select an existing filter from the drop-down list.
- Save filter: to save the current search criteria for future use.

For more information on the usage of search panels and filters, refer to the sections Search Panels (page 10) and Defining Search Filters (page 16).

3. Verify the information on the *Builds and Deploys Overview* panel.

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

The following information fields are available:

Field	Description
OID	This field contains the OID (Object Identifier) of the Build or Deploy action. Click this link to display the details for this Level Request which created the Build or Deploy. For more information, refer to the section explaining the Summary tab page of the *Level Request Detail* screen.
Type	This field contains the object type: Build or Deploy.
Project Stream	This field contains the identification of the Project Stream. This name is composed of: • Project Name • Project Stream Type: H (Head) or B (Branch) • Project Stream Prefix, optionally followed by the Suffix in case of a Branch Project Stream Example: `Webpad H_1-0`
Project Name	This field contains the name of the Project concerned by the Level Request which created the Build or Deploy.
Level Name	This field contains the name of the Level concerned by the Level Request which created the Build or Deploy.
Level Type	This field contains the type of the Level concerned by the Level Request (Build, Test or Production) which created the Build or Deploy.
Environment Name	This field contains the name of the Build or Deploy Environment.
Machine Name	This field contains the name of the Machine.
Status	This field contains the status of the Build or the Deploy. The following status indications are possible: • *Success* • *Warning* • *Fail* • *Run* • *Wait* • *In progress* • *Rejected* • *Canceled* • *Aborted* For more information, refer to the section Summary (page 66).

Field	Description
Level Request Status	This field contains the Level Request status. The following status indications are possible: • *Success* • *Warning* • *Fail* • *Run* • *Aborting* • *Awaiting requested Date/Time* • *Awaiting Pre-Approval* • *Awaiting Post-Approval* • *Rejected* • *Canceled* • *Aborted* For more information, refer to the section The Level Requests Overview Screen (page 58).
Build Number	This field contains the Build Number of the Level Request. Use this link to access the *Build History Detail* screen.
VCR Tag	This field contains the VCR Tag of the Level Request. This Tag matches a Build with its source code in the VCR. The Level Request VCR Tag is displayed as a link leading to the *Sources* tab page on the *Level Request Detail* screen. For more information, refer to the section Sources (page 97).
Start	This field indicates the date and time when the Build or Deploy execution started.
End	This field indicates the date and time when the Build or Deploy execution ended.

4. View the details of a specific Build or Deploy

Click the Level Request's *OID* link which created the Build or Deploy.

For more information, refer to the section explaining the Summary tab page of the *Level Request Detail* screen.

CHAPTER 8
Packages

A Package allows moving one or more individual files selected manually from a VCR Stream (Head or Branch) through the IKAN ALM Lifecycle. One or multiple Packages may be created in each Project Stream of a Package-based Project. This is different from the original way of working in the Project Streams of Release-based projects, where a configurable automated process defines which file revisions are retrieved from the head (trunk) or branch of the versioning system and moved in the Lifecycle.

From a **logical point of view**, Packages are partial views or views at a certain point in time on a Project Stream. The (logical) content of a Package may evolve in between two Builds and may be reused several times, over a period of time, throughout the Lifecycle of the Project Stream it belongs to.

Packages are attached to one single Project Stream and can contain one up to all files of the Project Stream from the moment a revision is available in the VCR. On top of that, several Packages of a Project Stream can contain the same file, preferably with a different revision. However, there is no automated control on concurrent development on a file used by Packages belonging to the same Project Stream.

During the whole Lifecycle, each Package has its own Build number and evolves independently of the other Packages

From a **physical point of view**, it comes down to putting together a set of files to be extracted from the VCR during the execution of the Build Level Request as to present them to the Build and Tag Phases, and to finally execute the Deploy of this set of files. For Release-based Projects, the rules for extracting files from the VCR are specified in the Project Stream definition by selecting the type of Build, and IKAN ALM will automatically take care of the selection of the objects. For Package-based Projects on the other hand, the selection of files to be extracted from the VCR is done manually by the Users of the Project in the form of a Package. Refer to the section Creating a Package (page 111)for more in information.

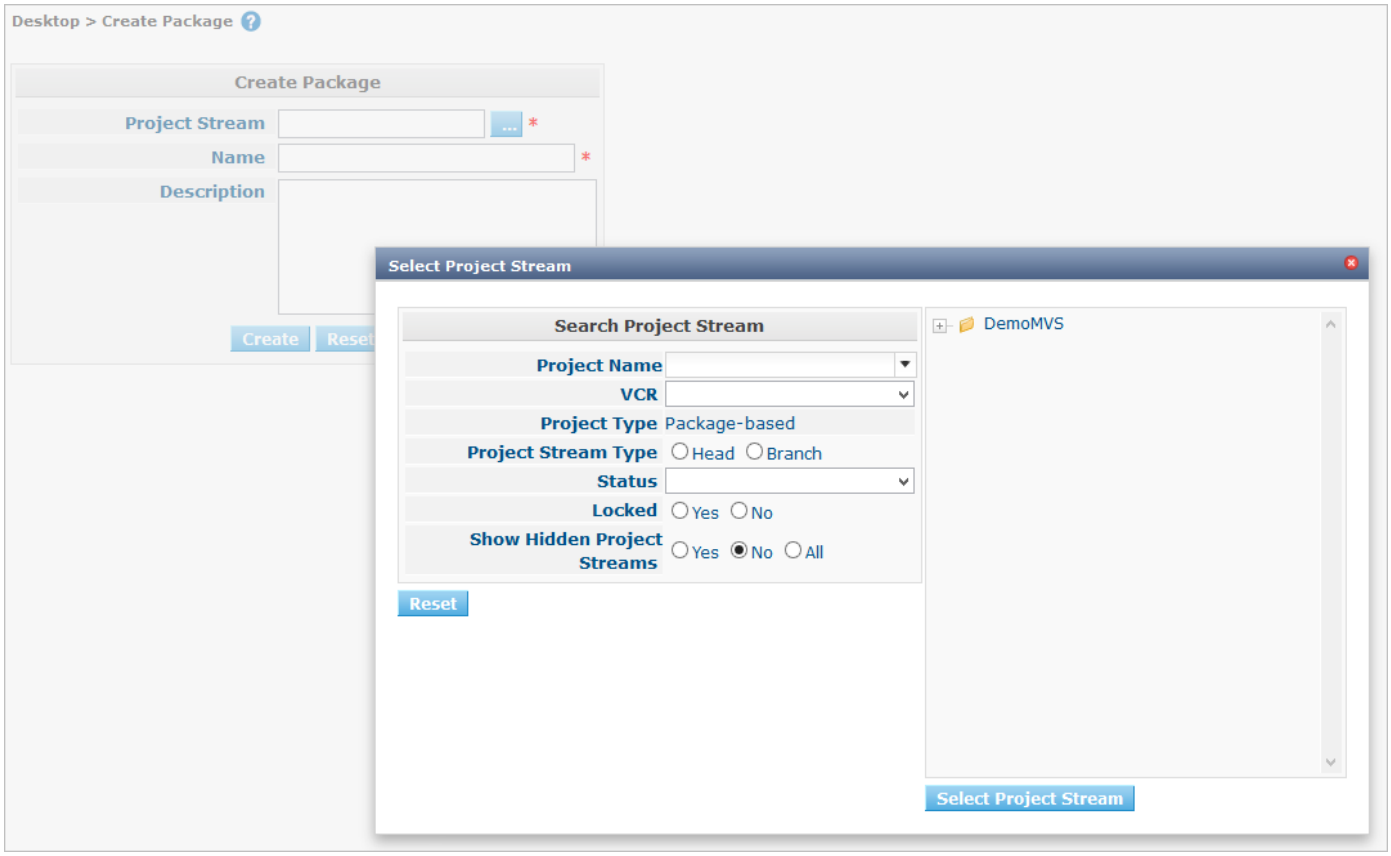
From a **technical point of view**, Projects are defined and separated to use either Packages or Releases, and can never be mixed.

Refer to the following procedures for more information:

- Creating a Package (page 111)
- The Package Overview Screen (page 113)
- Viewing Package Settings (page 115)
- Editing a Package (page 116)
- Hiding/Showing a Package (page 117)
- Viewing the Package History (page 117)

8.1. Creating a Package

1. Select *Packages* > *Create Package* on the Main Menu.
   The *Select Project Stream* window is displayed.



2. Define search criteria on the *Search Project Stream* panel.
   If you do not immediately find the Project Stream in which you want to create the Package, you can use the search panel on the left.
   The following search criteria are available:

Search Criteria	Meaning
Project Name	From the drop-down list, select the name of the Project for which you want to display the Project Streams.
VCR	Select the VCR name.
Project Type	This field is set to *Package-based*.
Project Stream Type	Select the type of Project Stream: Head or Branch.
Status	Specify the status of the Project Stream: • Under construction • Planning • Development • Testing • Stable • General available • Frozen • Closed

Search Criteria	Meaning
Locked	Indicate whether or not you want to display locked Project Streams. By default, both locked and unlocked Project Streams are displayed.
Show Hidden Project Streams	Indicate whether or not you want to display hidden Project Streams. By default, hidden Project Streams are not shown.

The list of elements displayed in the tree at the right, will be limited to the Project Streams matching these search criteria.

Click the *Reset* button to clear the Search fields.

3. Select the required Project Stream

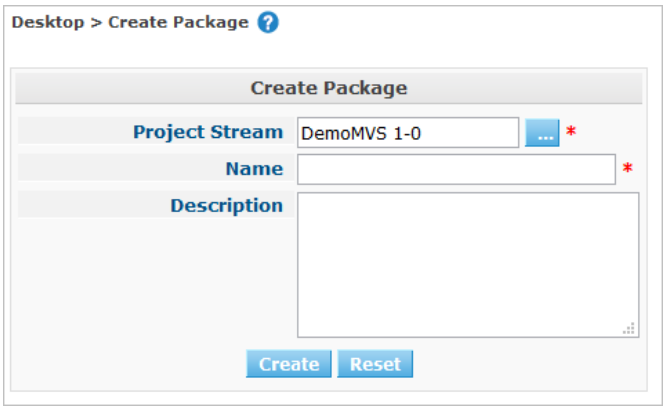
Expand the tree for the corresponding Project using the ⊞ icon to display its Project Streams.

Select the required Project Stream and click *Select Project Stream*.

The *Create Package* screen is displayed. The selected Project Stream name is automatically filled in. In case you want to select a different Project Stream, select the ... at the right to return to the *Select Project Stream* window.

4. Enter the name and, optionally, a description for the new Package, and click *Create* to confirm the creation of the Package.

The *Create Package* window is displayed.

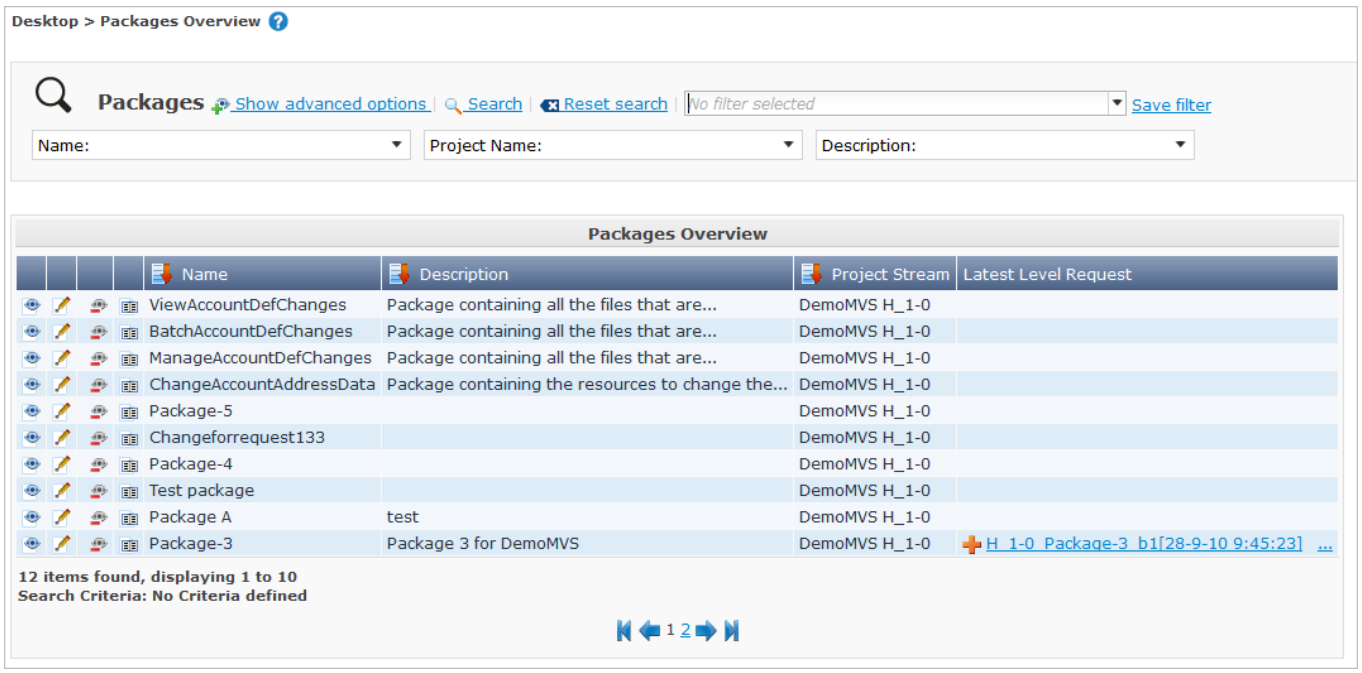


The *Edit Contents* tab page of the *Package Details* screen will be displayed.

A newly created package is empty. For more information on adding File Revisions to the newly created Package, refer to the section Package Details (page 117).

8.2. The Package Overview Screen

1. Select *Packages* > *Overview Packages* on the Main Menu.
   The *Packages Overview* window is displayed.



2. Use the search criteria on the *Search* panel to only display the Packages you are looking for.

The following options are available:

- Show advanced options: to display all available search criteria.
- Search: in principle it is not necessary to click the *Search* option. The results on the overview will be automatically synchronized in function of the selected criteria.
- Reset search: to clear all search criteria and display the full list of items.
- Select an existing filter from the drop-down list.
- Save filter: to save the current search criteria for future use.

For more information on the usage of search panels and filters, refer to the sections Search Panels (page 10) and Defining Search Filters (page 16).

3. Depending on your access rights, the following links may be available on the *Packages Overview* panel:

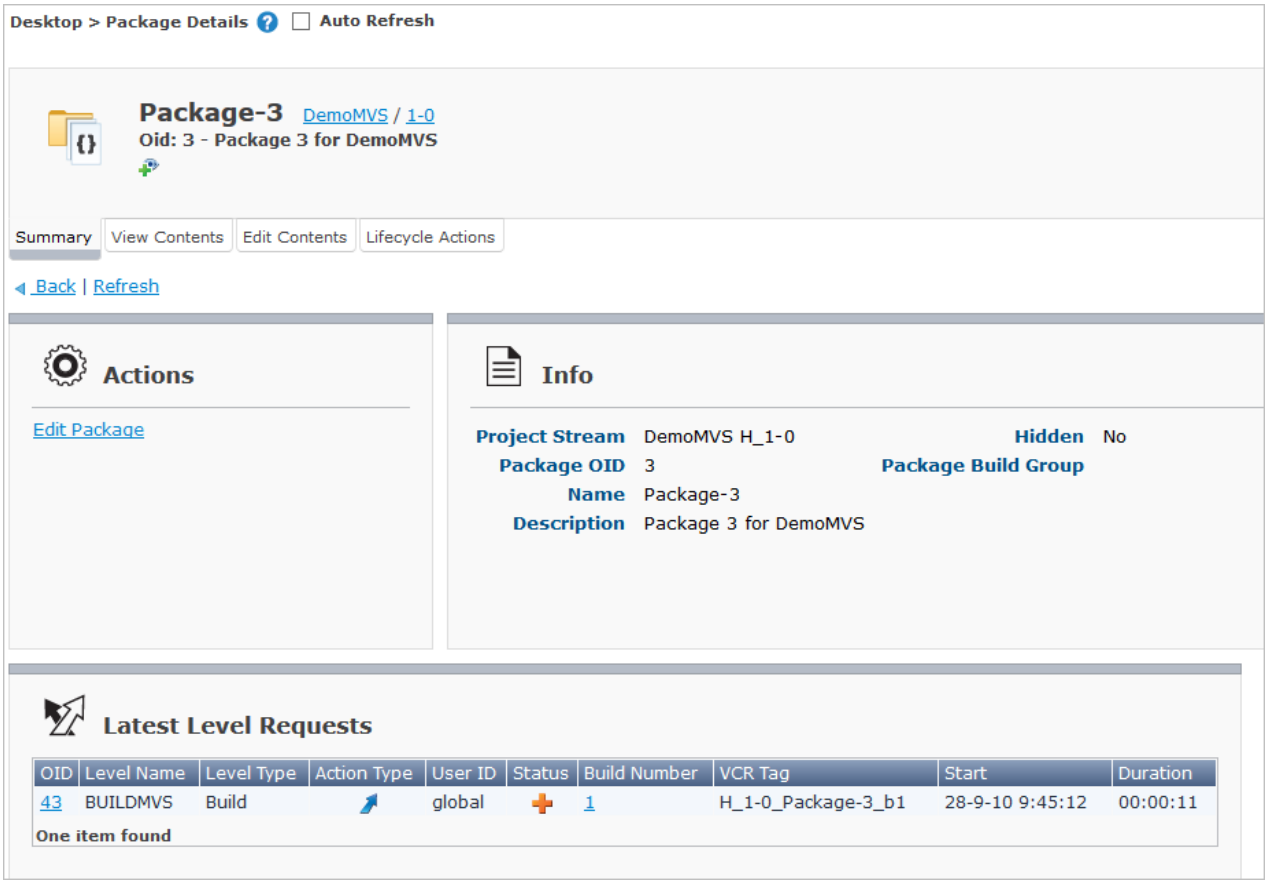
	View This option is available to all IKAN ALM Users. It allows viewing the selected Package definition and its connected file revisions. See Viewing Package Settings on page 115.
	Edit This option is available to IKAN ALM Users with Project Administrator Access Rights. It allows editing the Package definition and/or its connected file revisions. See Editing a Package on page 116.
	Hide / Show This option is available to IKAN ALM Users with Project Administrator Access Rights. It allows hiding the selected Package. See Hiding/Showing a Package on page 117.
	History This option is available to all IKAN ALM Users. It allows to display the History of all create, update and delete operations performed on a Package. See Viewing the Package History on page 117.

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

8.3. Viewing Package Settings

1. Select *Packages* > *Overview Packages* on the Main Menu.

2. Click the *View* link on the *Packages Overview* panel.
   The *Summary* tab page of the *Package Details* screen is displayed.



3. Verify the settings.
   The *Package Details* screen contains 4 tab pages:

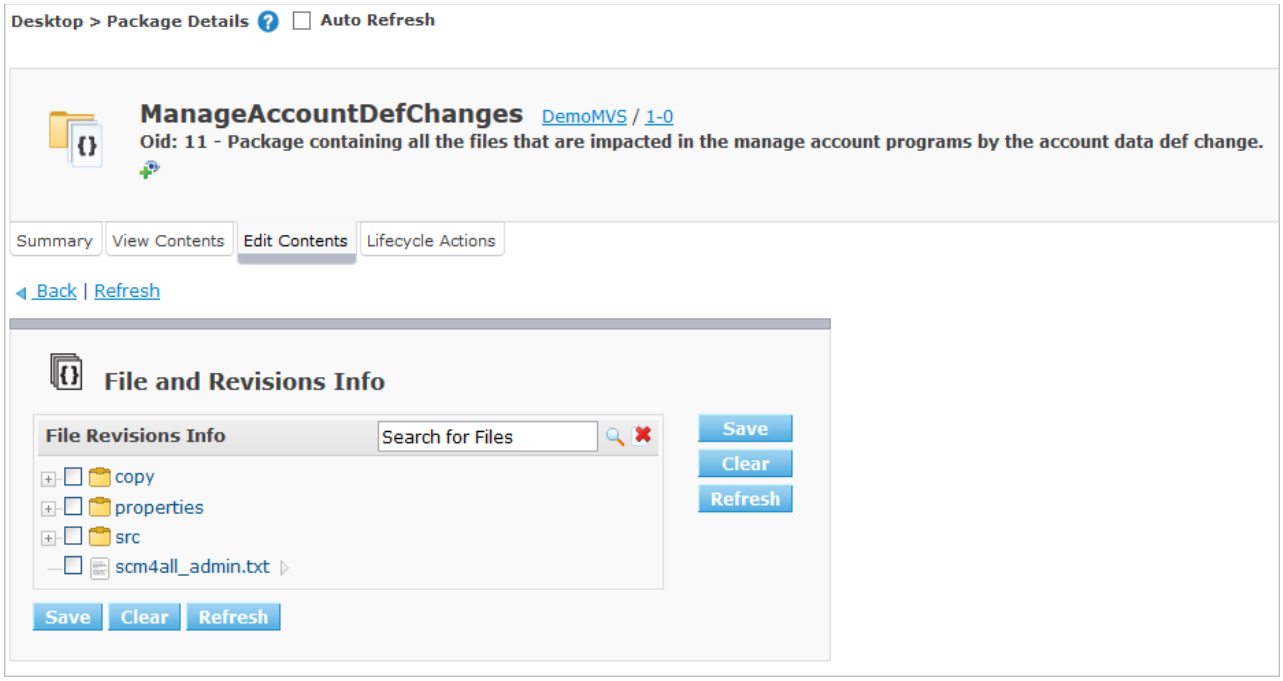
Summary	Displays the basic information concerning the Package
View Contents	Displays the contents of the Package
Edit Contents	Allows modifying the Package's contents
Lifecycle Actions	Displays the actions executed on the Lifecycle. Allows to launch Level Requests for the selected Package.

For more detailed information on the different tab pages, refer to the section Package Details (page 117).

8.4. Editing a Package

1. Select *Packages* > *Overview Packages* on the Main Menu.

2. Click the *Edit* link on the *Packages Overview* panel.
   The *Edit Contents* tab page of the *Package Details* screen is displayed.



On this tab page, you can modify the File Revisions contained in the Package.
For a more detailed explanation on how to do that, refer to the section Edit Contents (page 121)
Other available buttons:

- *Refresh*: to refresh the information on the information panels
- *Back*: to return to the *Package Overview* screen

Note: If the *Auto Refresh* option is activated, the screen will be refreshed each time the defined rate is expired. See Auto Refresh on page 12.

3. Click *Save* to confirm your changes.
   Other available buttons:

- *Clear*: to deselect all File Revisions. If in search mode, all matched files (and directories) will be unchecked.

  Note that when clicking the Clear button while holding down the CTRL key, will select all files in the current tree.
- *Refresh*: to refresh the information on the information panels

8.5. Hiding/Showing a Package

Specifying that a Package is "hidden", causes it not to be displayed by default on Overview panels. For example, its Level Requests will be filtered out by default on the *Level Requests Overview*.

This can be very useful to hide older Packages on the overviews, without losing the historical information associated with those Packages.

1. Select *Packages* > *Overview Packages* on the Main Menu.

2. Click the *Hide* link in front of the Package you want to hide.

   The icon in front of the Package changes to .

Note:	Hidden Packages will not appear on the Overview screens for Level Requests, Approvals, Build and Deploys, nor on the *Create Level Request: Select Package* screen when creating a Level Request for Package-based Projects.

3. To "unhide" a hidden Package, click the *Show* link.

   The icon will be changed appropriately.

Note:	A search criterion is available on the *Search Package* panel to specify whether or not you want to display hidden Packages.

8.6. Viewing the Package History

1. Select *Packages* > *Overview Packages* on the Main Menu.

2. Click the *History* link on the *Package Overview* panel to display the *Package History View*.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

   Click *Overview* to return to the *Packages Overview* screen.

8.7. Package Details

The *Package Details* screen contains the detailed information concerning the selected Package.

The screen is structured as follows:

1. Status Header

   The header displays the name, OID and description of the selected Package, and whether it is a hidden package or not, as well as links to the Project and Project Stream it is used on.

2. Tab Pages with detailed information

   Underneath the status indication, several tabs are available, each of them displaying additional information concerning the Level Request. By default the *Summary* tab page is displayed.

Refer to one of the following sections for more information.

- Summary (page 118)
- View Contents (page 120)
- Edit Contents (page 121)
- Lifecycle Actions (page 124)

3. Back and Refresh links

- Use the *Back* link to return to the previous screen.
- Use the *Refresh* link to update the displayed information. This link reloads the currently selected tab page, as well as the header information.

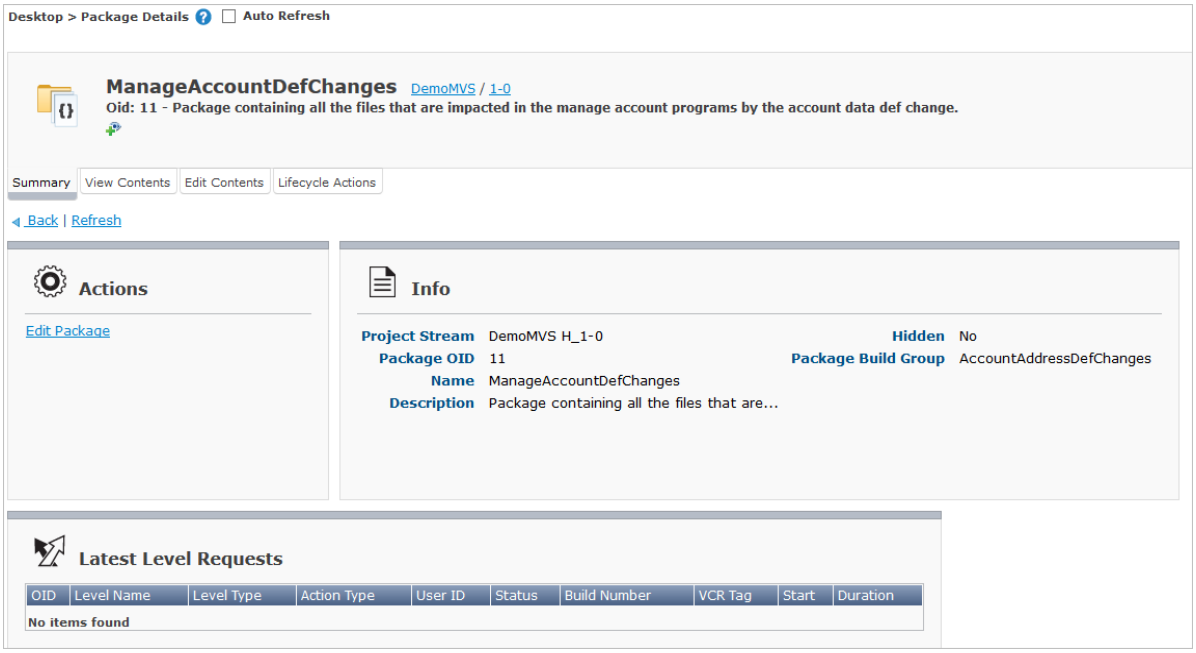
4. Auto Refresh option

The *Auto Refresh* option is only of use on the *Summary* and *Lifecycle Actions* tab pages.

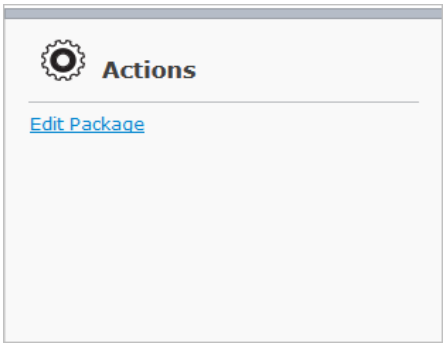
For more information on the *Auto Refresh* settings, refer to the section Auto Refresh (page 12).

Summary

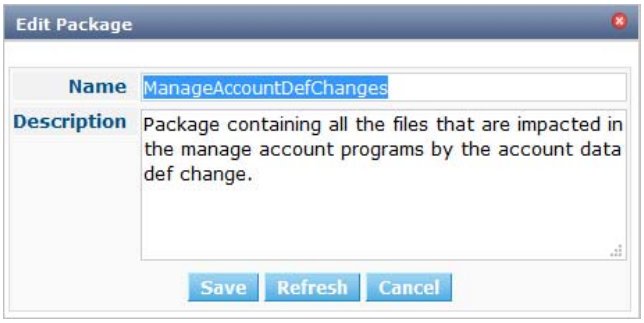
The information concerning the Package and the possible actions are spread over different panels.



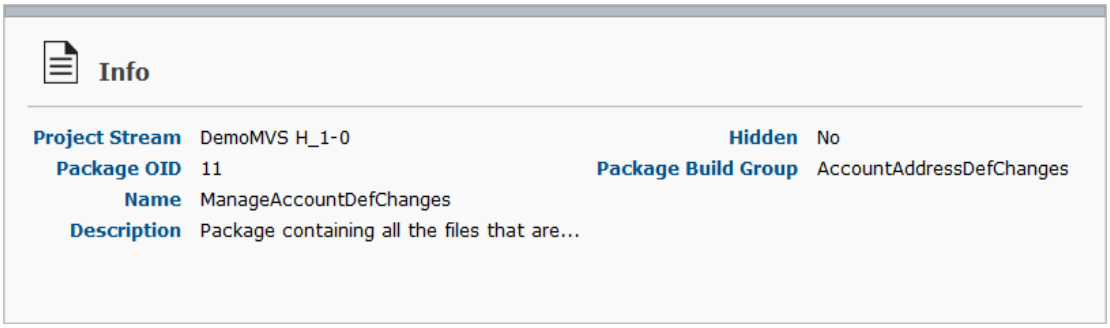
Actions Panel



The *Actions* panel contains the *Edit Package* link. This link will open a pop-up window allowing you to modify the Name and the Description of the Package.



Info Panel

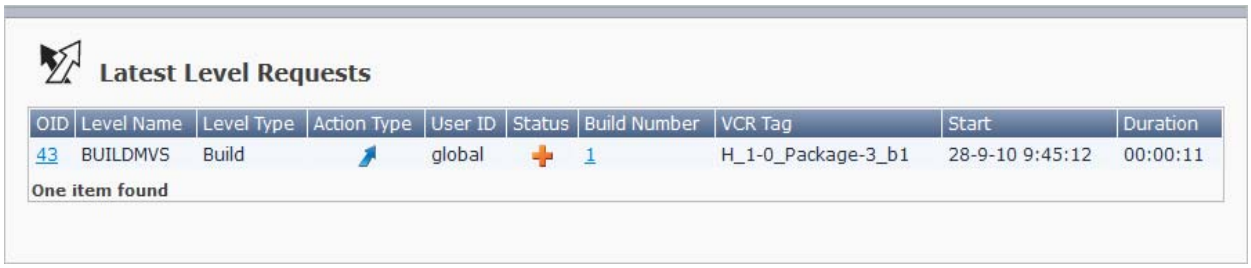


This panel displays the information found in the Header as well as some additional information.

Field	Meaning
Project Stream	This field contains the identification of the Project Stream. This name is composed of: • Project Name • Project Stream Type: H (Head) or B (Branch) • Project Stream Prefix, optionally followed by the Suffix in case of a Branch Project Stream Example: `Webpad H_1-0`
Package OID	This field contains the internal OID of the Package.
Name	This field contains the name of the Package.

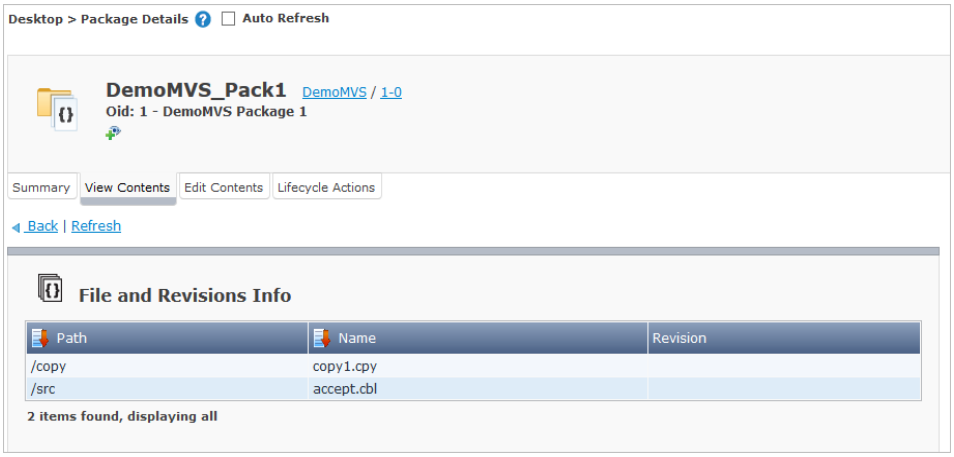
Field	Meaning
Description	This field contains the description of the Package.
Hidden	This field indicates whether or not the Package is hidden. By default newly created Packages are not hidden. For more information on hiding Packages, refer to Hiding/Showing a Package (page 117).
Package Build Group	If applicable, this field contains the Package Build Group the Package is linked to. For more information, refer to the section Package Build Groups (page 125).

Latest Level Requests Panel



This panel displays the ten latest Level Requests executed for the selected Package. On this list, you can click the *OID* link of one of the Level Requests to display its details. Clicking the Build Number will show the Build History of that Level Request.

View Contents

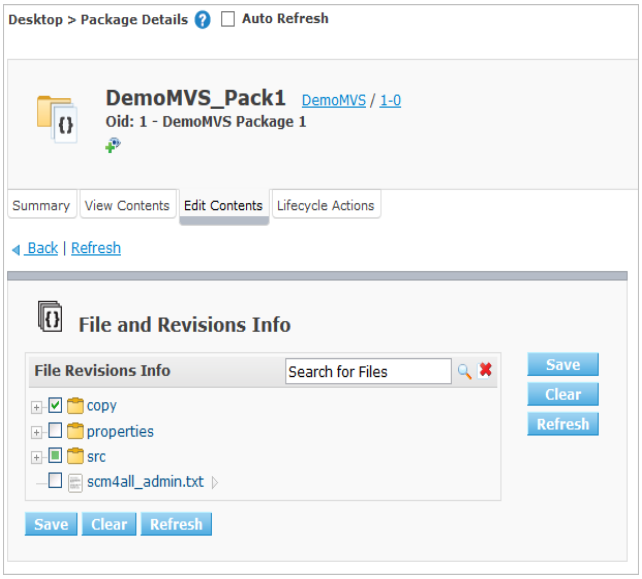


The *View Contents* panel lists all File Revisions currently connected to the Package.

The following fields are available:

Field	Meaning
Path	The path to the directory containing the file.
Name	The name of the Revision File contained in the Package.
Revision	The revision number in the VCR of the selected file. This field may be empty, indicating that the latest File Revision is retrieved at the moment the Package is built.

Edit Contents



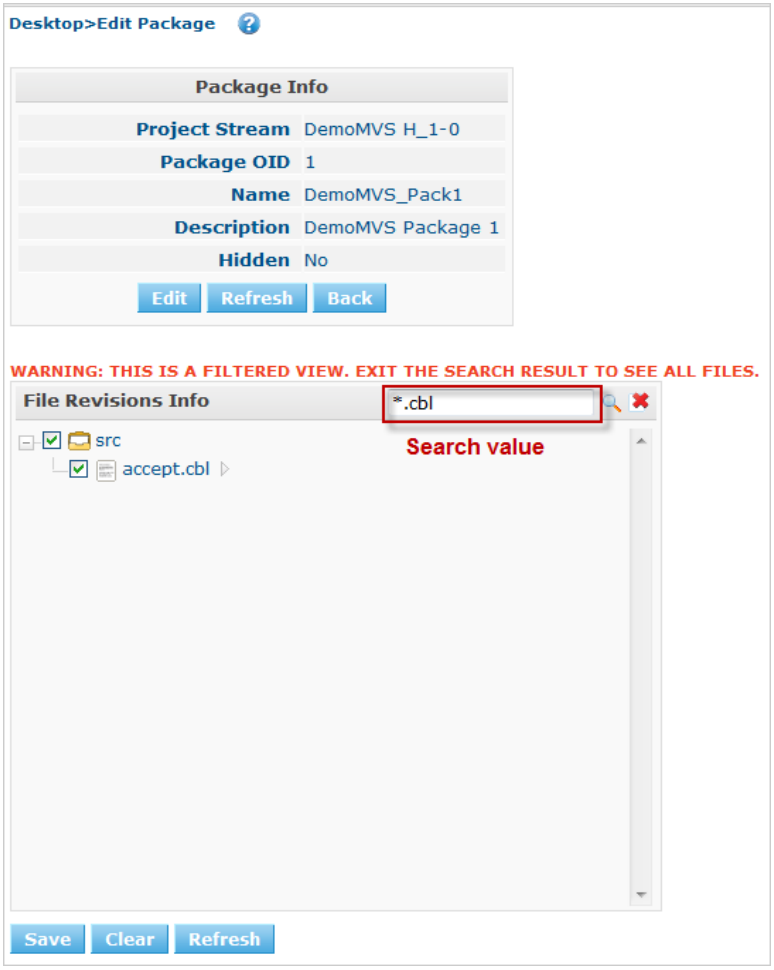
On this panel, you can modify the contents of the selected Package.

Note: The information on the *File Revisions Info* panel is directly retrieved from the VCR. Use the *Refresh* button to display the latest state.

Using the *Clear* button will deselect all items in the Package. Clicking the *Clear* button while pressing the *CTRL* key will select all items in the Package.

1. Use the *Search* field for selecting specific File Revisions.

   You can use the Search field for selecting specific File Revisions. Wildcards like '*' and '?' may be used in any position. Sample search values: `foo.*`, or `?oo.txt` or combined `f??.*`



   To exit the search mode and redisplay all files and folders in the tree, click the *Exit Search* icon.

2. Select the File Revisions you want to add to the Package and click *Save*.

   The information on the *File Revisions Info* panel is directly retrieved from the VCR. Use the *Refresh* button to display the latest state.

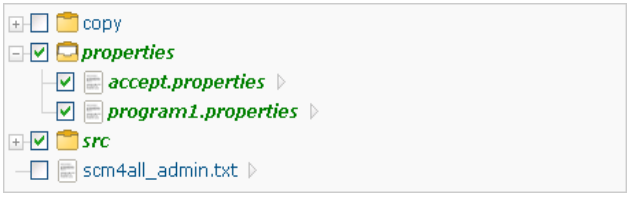
   Using the *Clear* button will deselect all items in the Package. Clicking the *Clear* button while pressing the *CTRL* key will select all items in the Package.

Note:	Files which have been deleted from the VCR are shown struck through on the tree. At folder level, a red exclamation mark is displayed after the folder name.

The following actions are possible:

- Adding a File or Folder to the Package

  To select a File or Folder, expand the tree by clicking the ⊞ icon to display all contained folders and files. Select the folder(s) or file(s) you want to add to the Package.



- Adding a specific Revision of a selected File to the Package

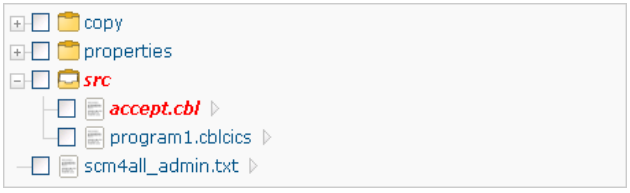
  To select a specific Revision for a File, click the ▷ at the right of a selected File name. The *Select Revision* pop-up window is displayed. Select the required revision from the drop-down list and click *Select*. The revision number will be displayed next to the selected File.

  Note: If you do not select a specific revision of a file, the latest revision will be used at the moment the Package is built.

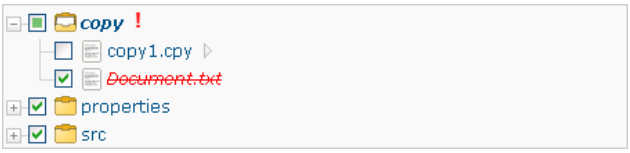
- Deleting a File of Folder from the Package

  To delete a File or Folder from the Package, deselect it in the tree. The name will be displayed in red.

  To deselect all Files in the Package, you can use the *Clear* button.



  Note: If a File or Folder has been removed from the VCR, this is indicated by an exclamation mark after the File or Folder name. The removed file is shown by strike-through. It is advised to delete the file or folder from the package as to synchronize it with the VCR.



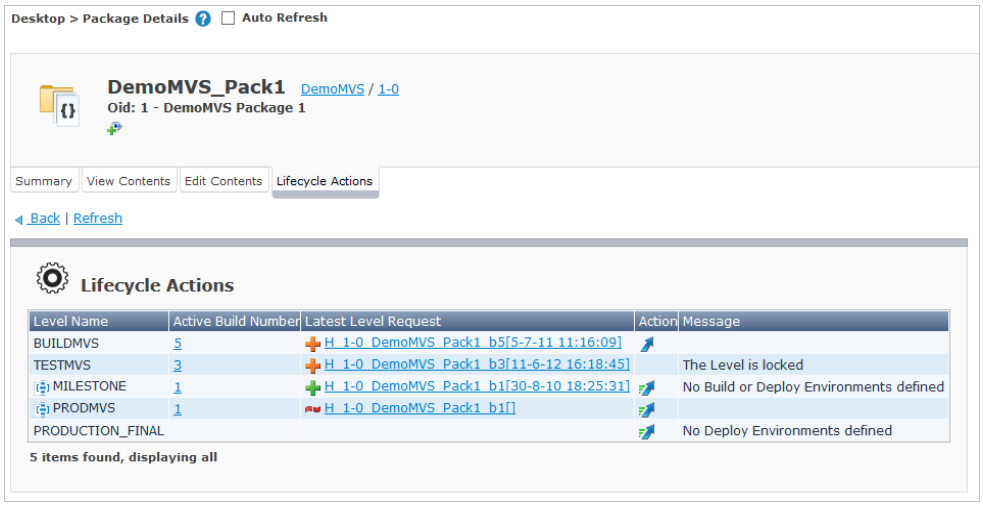
- Changing from a specific Revision to the latest Revision of a File in the Package

  To change from a specific Revision to the latest Revision of a File, click the ▷ at the right revision number. The *Select Revision* pop-up window is displayed. Deselect the revision by selecting the empty value on top of the drop-down list and click *Select*. The revision number will disappear and the file will be marked with a red ▶.

- Modifying a Revision of a File from the Package

  To modify to another specific Revision of a File, click the ▷ at the right revision number. The *Select Revision* pop-up window is displayed. Select the required revision from the drop-down list and click *Select*. The new revision number will be displayed next to the selected File.

Lifecycle Actions



This panel displays the latest Level Request for each of the Levels defined in the Lifecycle.

Using the action icons, you can also start a Level Request for one of the Levels. Clicking such an action icon, will take you to the Create Level Request screen.

For more information on creating Level Requests, refer to the section Creating Level Requests (page 45).

CHAPTER 9
Package Build Groups

A Package Build Group establishes a relationship between Packages and allows transferring the latest builds from all or specific packages in such a Group before the execution of the build process of a Package.

This is, for example, useful for a Mainframe lifecycle in which copybooks may be modified in one Package, which is used by programs in other Packages. If those Packages are linked into a Package Build Group, the correct (version of the) copybooks will be transported at compilation time.

Note: This chapter only describes how to create and manage a Package Build Group.

To transfer Packages belonging to a Package Build Group when building a Package, a new core Phase has been added, namely the *Transport Package Results* Phase. This Phase must be inserted in the workflow of the Build Environment.

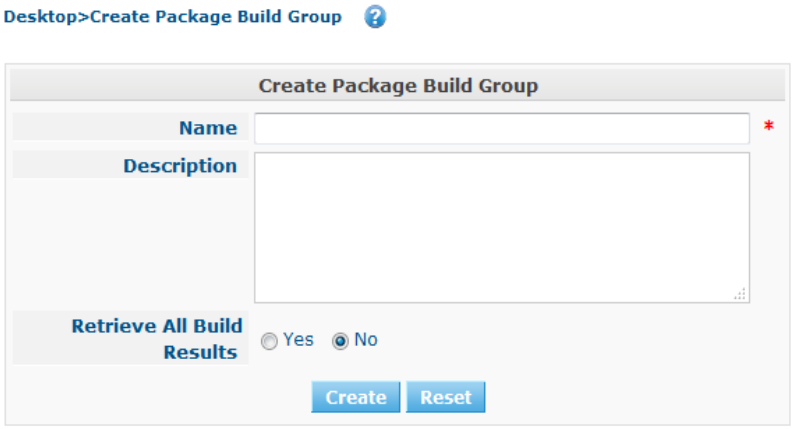
For more details, refer to the description of this Phase in the appendix Phases - General Information (page 520).

Refer to the following procedures for more information:

- Creating a Package Build Group (page 126)
- The Package Build Groups Overview Screen (page 127)
- Editing a Package Build Group (page 129)
  - Adding a Package to a Package Build Group (page 130)
  - Removing a Package from a Package Build Group (page 133)
- Hiding/Showing a Package Build Group (page 133)
- Viewing the Package Build Group History (page 134)

9.1. Creating a Package Build Group

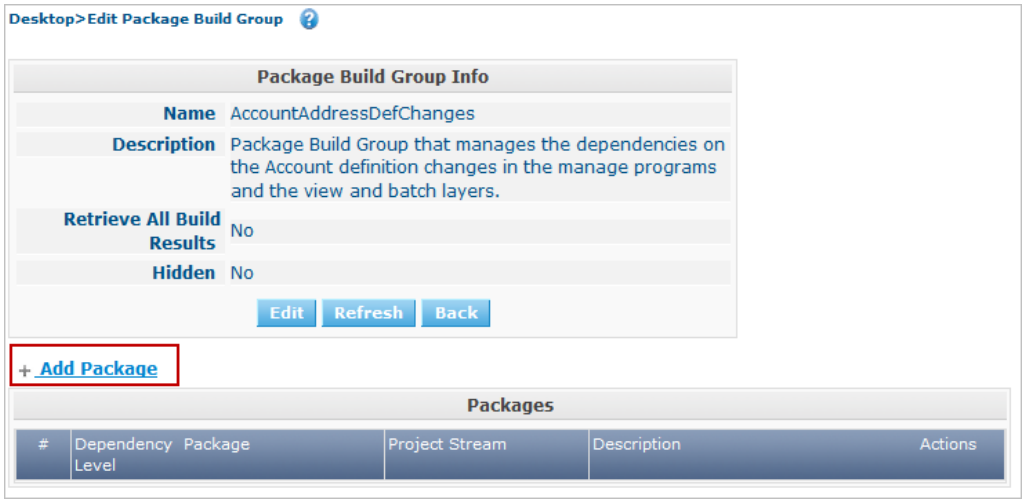
1. Select *Packages* > *Create Package Build Group* on the Main Menu.
   The *Create Package Build Group* screen is displayed.



2. Fill out the fields. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	Enter the name of the new Package Build Group in this field. This field is mandatory.
Description	Enter a description for the new Package Build Group in this field. This field is optional.
Retrieve All Build Results	Select whether the "Transport Package Results" Phase should retrieve all build results of the other packages in the Package Group. • *Yes* If you set this option to *Yes*, all build results of the connected Packages will be retrieved when a Package of the Package Build Group is built. In this case, the Dependency Levels of the Packages will be ignored. • *No* Default value. If you set this option to *No*, only the build results of the connected Packages with a lower Dependency Level will be retrieved when a Package of the Package Build Group is built. The Dependency Level is specified when adding a package to the group. For more information on adding Packages and rearranging the order of Packages, refer to the section Adding a Package to a Package Build Group (page 130).

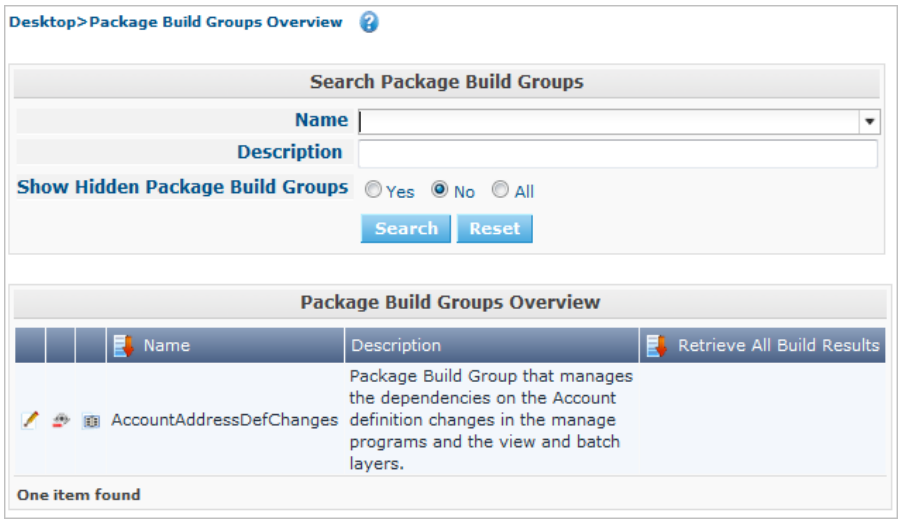
3. Once you have filled out the fields, click *Create*.
   The *Edit Package Build Group* is displayed.



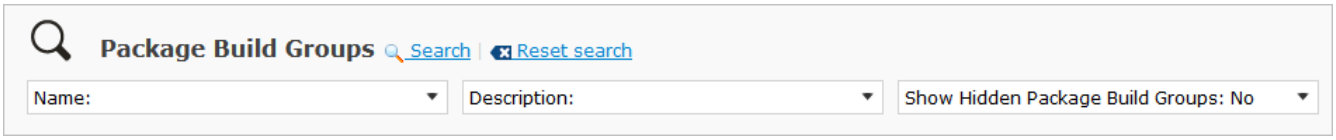
4. Next, you can add Packages using the *Add Package* link.
   For more information, refer to the section Adding a Package to a Package Build Group (page 130).

9.2. The Package Build Groups Overview Screen

1. Select *Packages > Overview Package Build Groups* on the Main Menu.
   The *Package Build Groups Overview* window is displayed.



2. Use the search criteria on the *Search Package Build Groups* panel to only display the Package Build Groups you are looking for.



The following options are available:

- Show advanced options: to display all available search criteria.
- Search: in principle it is not necessary to click the *Search* option. The results on the overview will be automatically synchronized in function of the selected criteria.
- Reset search: to clear all search criteria and display the full list of items.
- Select an existing filter from the drop-down list.
- Save filter: to save the current search criteria for future use.

For more information on the usage of search panels and filters, refer to the sections and Defining Search Filters (page 16).

3. Depending on your access rights, the following links may be available on the *Package Build Groups Overview* panel:

	Edit This option allows editing the Package Build Group and/or its connected Packages. See Editing a Package Build Group on page 129.
	Hide / Show This option allows hiding the selected Package Build Group. See Hiding/Showing a Package Build Group on page 133.
	History This option allows to display the History of all create, update and delete operations performed on a Package Build Group. See Viewing the Package Build Group History on page 134.

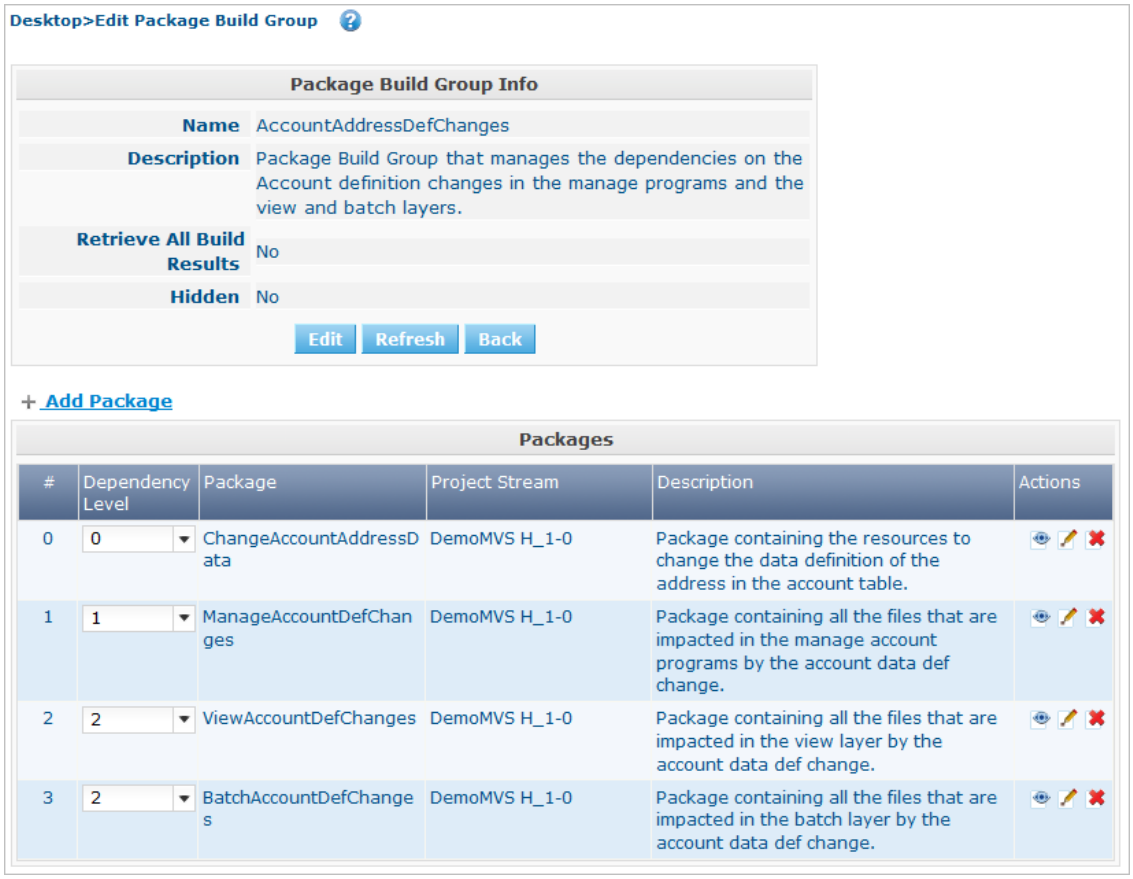
Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

9.3. Editing a Package Build Group

1. Select *Packages* > *Overview Package Build Groups* on the Main Menu.

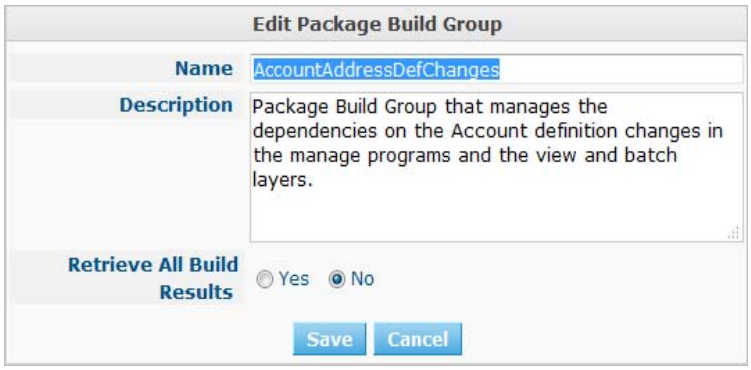
2. On the *Package Build Groups Overview* panel, click the *Edit* link next to the Package Build Group you want to edit.

   The following screen is displayed.



3. Click the *Edit* button if you need to modify the definition of the Package Build Group.

   The following pop-up window is displayed:



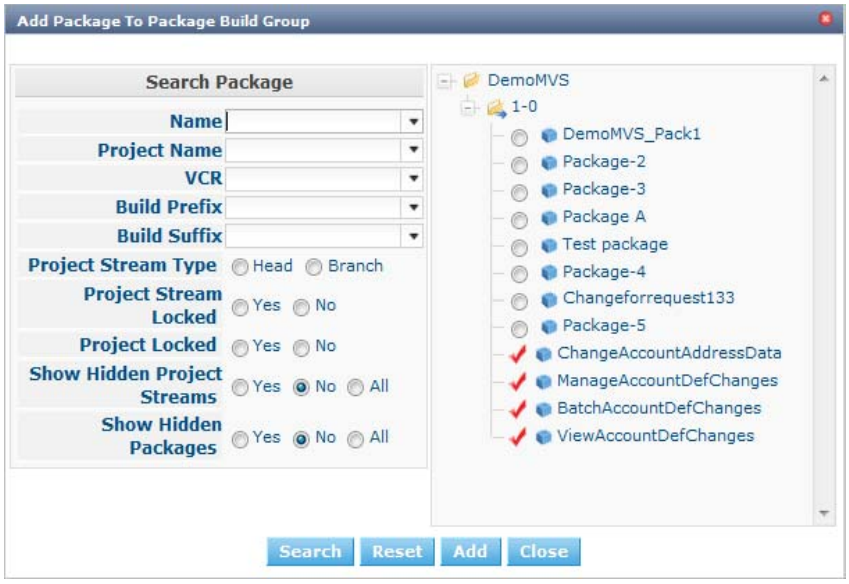
   Edit the Package Build Group Information as required and click the *Save* button.

   For more information on the different fields, refer to the section Creating a Package Build Group (page 126).

4. You can also add or remove packages and modify their order and Dependency Level.
   For more information, refer to the section Adding a Package to a Package Build Group (page 130).

Adding a Package to a Package Build Group

1. Click the *Add Package* link on the *Edit Package Build Group* screen or right-click the *Packages* overview table and select *Add Package*.
   The following window is displayed.



2. Search for the Package you want to add.
   You can search for the Package using the selection criteria on the *Search Package* panel on the left, or you can immediately use the tree view on the right to select the *Package*.
   The following search criteria are available.

Field	Meaning
Name	Enter the name of the Package you want to add.
Project Name	Enter the name of the Project.
VCR	Enter the VCR name.
Build Prefix	Enter the Build prefix.
Build Suffix	Enter the Build suffix.
Project Stream Type	Enter the type of the Project Stream: Head or Branch.
Project Stream Locked	Indicate whether or not you want to display locked Project Streams.
Project Locked	Indicate whether or not you want to display locked Projects.
Show Hidden Project Streams	Indicate whether or not you want to display hidden Project Streams.
Show Hidden Packages	Indicate whether or not you want to display hidden Packages.

3. In the tree overview, select the Package you want to add.

Note: Packages can only be part of one Package Build Group. If you try to add a Package that is part of another Package Build Group, an error is displayed. For example: Sample error message: A Package can only be part of one Package Build Group. Package 'xyz' is already part of Package Build Group 'zyx'

4. Click the *Add* button to add the Package.

The Package will be added at the end of the list. You can change the order of the packages by changing the Dependency Level, or by dragging and dropping the Package to the appropriate place, as explained in steps 4 and 5.

Packages can only be added one at a time. Repeat this action for all the packages you want to add.

Once you have selected all the packages you want to add, click the *Close* button to close the window. The list of added Packages will be displayed on the *Packages* panel.

You can also click:

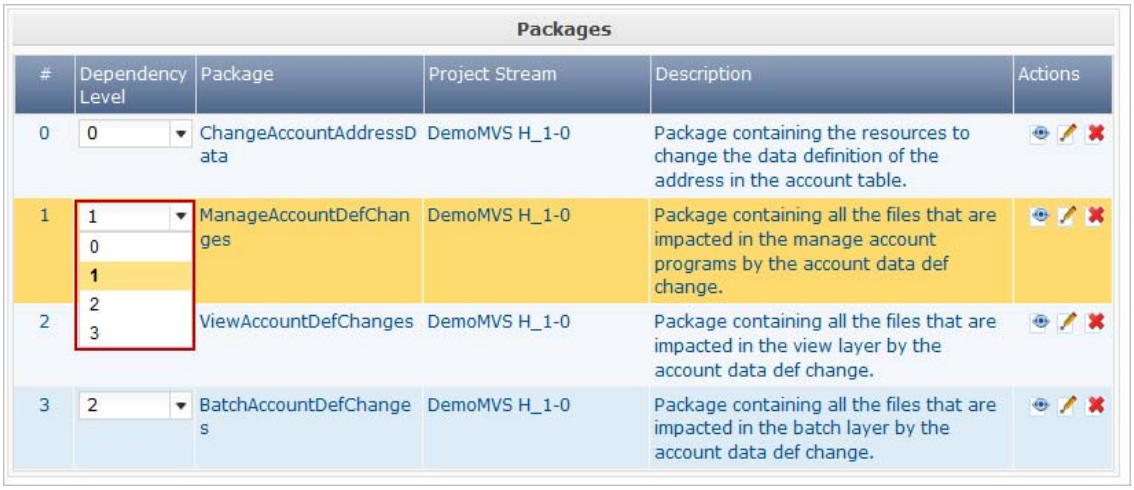
- *Search* to refresh the tree with the selected search criteria.
- *Reset* to clear the Search fields.
- *Close* to return to the *Edit Package Build Group* screen.

5. Specify the Dependency Levels.

When a Package is built, the *Transport Package Results* Phase will only retrieve the latest Build Results of the Packages with a **lower** Dependency Level in the Package Build Group.

Note: Exception: if the option *Retrieve all Build Results* is set to *Yes*, all Build Results will be retrieved regardless of their Dependency Level. See Creating a Package Build Group on page 126.

Modify the Dependency Level by selecting the appropriate level from the drop-down list.



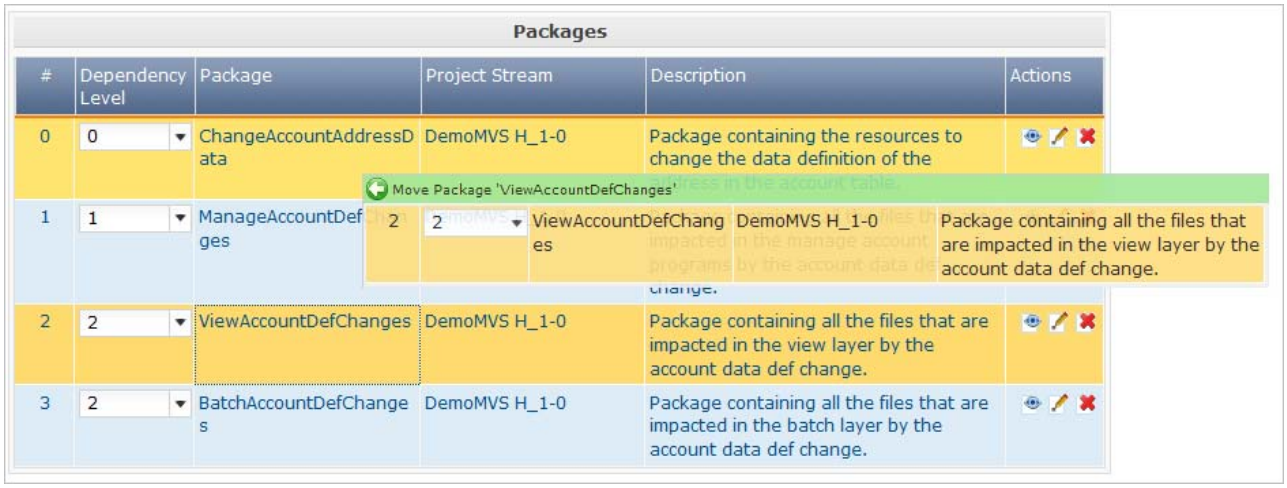
6. Modifying the order of the packages.

If required, you can modify the order of the Packages in the Package Build Group.

Select the Package you want to reposition by clicking on it and dragging it to the required position on the *Packages* Overview. You can only move one Package at a time.

Note: The orange line indicates the target position of the selected elements. Valid positions are indicated with a green bar above the selected elements, invalid positions with a red bar.

Example:



Note: If you change the order of the Packages, the Dependency Level may change as well.

7. Viewing a Package's settings.

   7.1. To display a connected Package's Settings, click the *View* link in the *Actions* column after the required Package.

   The *View Package* screen is displayed. See Viewing Package Settings on page 115.

   7.2. Click *Back* to return to the *Edit Package Build Group* screen.

8. Editing a Package's settings

   8.1. To modify a connected Package Settings, click the *Edit* link in the Actions column after the required Package.

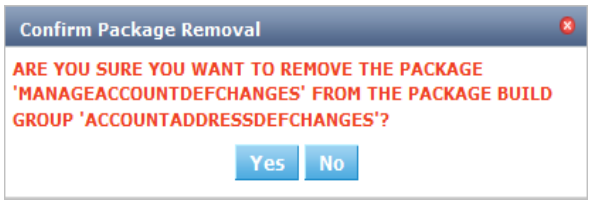
   The *Edit Package* screen is displayed. For more information on how to edit the Package, refer to Editing a Package (page 116).

   8.2. Click *Back* to return to the *Edit Package Build Group* screen.

Removing a Package from a Package Build Group

1. On the *Packages* panel of the *Edit Package Build Group* screen, click the *Remove* link after the required Package in the *Actions* column.

   A pop-up window asking you to confirm the removal is displayed.



2. Click *Yes* to confirm the removal.

   You can also click *No* to return to the previous screen without removing the Package.

Note: If you remove one of the Packages, the Dependency Level of the other Packages may also change.

9.4. Hiding/Showing a Package Build Group

Specifying that a Package Build Group is "hidden", causes it not to be displayed by default on the Overview panels.

This can be very useful to hide older Package Build Groups, without losing the historical information associated with them.

1. Select *Packages > Overview Package Build Groups* on the Main Menu.

2. Click the *Hide* link in front of the Package Build Group you want to hide.

   The icon in front of the Package Build Group changes to .

3. To "unhide" a hidden Package Build Group, click the *Show* link.

   The icon will be changed appropriately.

Note: A search criterion is available on the *Search Package Build Group* panel to specify whether or not you want to display hidden Package Build Groups.

9.5. Viewing the Package Build Group History

1. Select *Packages* > *Overview Package Build Groups* on the Main Menu.

2. Click the *History* link on the *Package Build Groups Overview* panel to display the *Package Build Group History View*.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

   Click *Back* to return to the *Package Build Groups Overview* screen.

CHAPTER 10

Managing the Desktop

Each IKAN ALM User has access to a personal Desktop made up of adjustable Tab Pages, each of them containing current information about the Project Streams, Levels and Packages he or she is interested in. You can personalize up to 10 Tab Pages on your Desktop and each Tab Page can contain up to 20 Element Types.

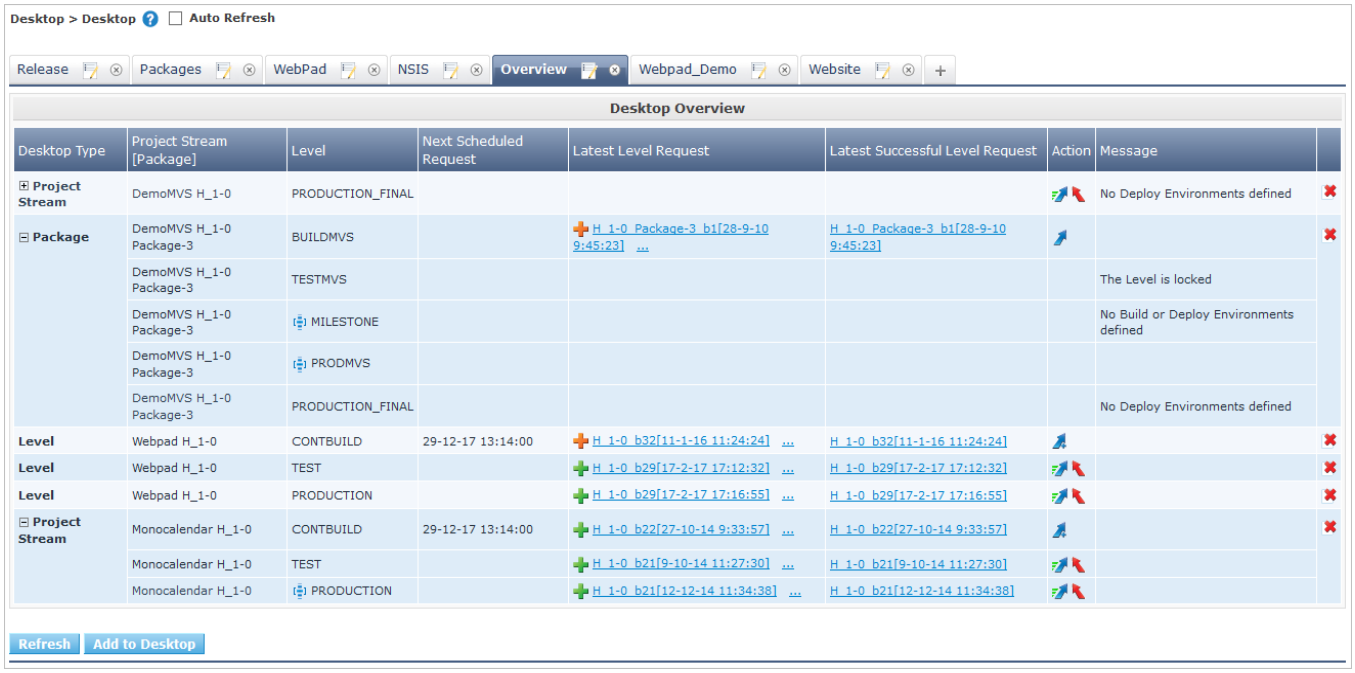
Furthermore the personal Desktop provides a shortcut to the Level Request creation and view options.

The following actions are possible:

- Managing the Desktop Tab Pages
  - The Desktop Overview Screen
  - Adding a Desktop Tab Page
  - Editing the Desktop Tab Page Label
  - Removing a Desktop Tab Page
- Managing the Desktop content
  - Adding Elements to a Desktop Tab Page
  - Rearranging Desktop Elements
  - Removing Desktop Elements

10.1. The Desktop Overview Screen

1. Select *Desktop* on the Main Menu.
   Your personal *Desktop Overview* will be displayed.



Note: If your Desktop Overview is empty, you can add content to it using the *Add to Desktop* button. See Adding Elements to a Desktop Tab Page on page 140.

2. Verify the elements on your Desktop.
   Per Tab Page, the following fields are available for each desktop element type.

Field	Description
Desktop Type	This field contains an indication of the Element type assigned to the Desktop. There are three possibilities: • Project Stream • Level • Package **Note:** Click the ⊞ icon to expand assigned Project Streams or Packages and to display the Levels that belong to it.
Project Stream [Package]	There are three possibilities: • For assigned Project Streams, this field contains the identification of that Project Stream. Example: `Webpad H_3-0` • For assigned individual Levels, this field contains the identification of the Project Stream this Level belongs to. Example: `Webpad B_3-1` • For assigned Packages, this field contains the identification of the Project Stream and the name of the Package-based Project. Example: `DemoMVS H_1-0 DemoMVS_Pack1` The Project Stream identification is composed of: • Project Name • Project Stream Type: H (Head) or B (Branch) • Project Stream Prefix, optionally followed by the Suffix in case of a Branch Project Stream

Field	Description
Level	For individual Levels, this field contains the name of that Level. For non-expanded Project Streams, this field contains the name of the highest Level belonging to the Project Stream. This is logically the Production Level. In case a Level is optional, the *Optional* icon will be displayed in front of the Level Name.
Next Scheduled Request	For Build Levels with active Schedule, this field displays the timestamp that the next Level Request is expected to run. The Level Request will not run, if: • the code was not changed, • the Scheduler is deactivated. For other Levels, this field remains empty.
Latest Level Request	This field contains the following information for the latest Request on the displayed Level: • Request status icon indicating the status of the latest Request on this Level: - : Success - : Warning - : Running - : Fail - : Awaiting Pre-Approval or Awaiting Post-Approval - : Awaiting Run Time - : Rejected - : Aborted - : Canceled • The VCR Tag (matching the defined template) and timestamp for Requests with status *Success*, *Failed* and *Warning*. These indications are displayed as a link. Click the link to access the Detailed Overview (page 65) for the selected Level Request. • Link ( ) to the Level Requests Overview (page 58) with Project name and Level name already filled in as Search Criteria.
Latest Successful Level Request	This field contains the VCR Tag (matching the defined template) and timestamp for the latest successful Request on the displayed Level. These indications are displayed as a link. Click the link to access the Detailed Overview (page 65) selected Level Request.
Action	This field contains the action icons which are available for the Level: • : For a Build Level, this icon allows creating a Level Request that will generate a new Build. For Build Levels with an active Scheduler, this matches a Forced Build. For Build Levels without active Scheduler, this matches a Requested Build. See Creating a Build Level Request on page 48. For Test and Production Levels, this icon allows creating a Level Request for delivering an existing Build to that Level (page 52). • : This icon (only available for Build Levels) allows creating a Level Request that will generate a new Build. The Create Level Request dialog will contain additional information about the modifications since the last Level Request. For Levels with an active Scheduler, this matches a Force+ Build. For Build Levels without active Scheduler, this matches a Request+ Build. See Creating a Build Level Request on page 48. • : This icon (only available for Production Levels) allows creating a Level Request that will restore a previous Build on that Level. Refer to Creating a Rollback Build Level Request (page 55).

Field	Description
Message	If it is not possible to define a Request for a specific Level, the messages in this field may indicate the reason. • *A Level Request is pending for this Level of this Project Stream* This message is displayed, if a Level Request is being executed or waiting for approval. You will need to wait until the current Level Request is completed, before you can define a new Level Request for this Level. • *The Level is locked* It is not possible to define Level Requests for Locked Levels. You (or the Project Manager) must audit the Project to unlock the Level (page 247), before you can define Level Requests for this Level. • *No Request Rights* It is not possible to define Level Requests, if the User ID you are logged in with does not have the required access rights. You must connect with a User ID having the right to run Requests or ask the Global Administrator to give this right to your User ID. • *The Project is locked* It is not possible to define Level Requests for locked Projects. Unlock the Project first by clicking the *Unlock* button on the *Projects Overview*. See Editing Project Settings on page 149. • *The Project Stream is locked* It is not possible to define Level Requests for locked Project Streams. You (or the Project Manager) can unlock the Project Stream first by clicking the *Unlock* button on the *Edit Project Stream screen*. See Editing Project Stream Settings on page 167. • *The Project Stream is frozen* It is not possible to define Build Level Requests for frozen Project Streams. You (or the Project Manager) can unfreeze the Project Stream first by selecting another status from the *Status* drop-down menu on the *Edit Project Stream screen*. See Editing Project Stream Settings on page 167.
	Click this link to remove the selected element from the Desktop Overview.

3. Click the *Refresh* button if you want to verify the changing status of the information on your Desktop.

   The screen will be refreshed each time you click the *Refresh* button.

Note: If the *Auto Refresh* option is activated, the Desktop will be refreshed each time the defined rate is expired. See Auto Refresh on page 12.

4. Click the *Add to Desktop* button to open the *Add to Desktop* pop-up window allowing you to add other elements to your desktop.

   See Managing the Desktop on page 135.

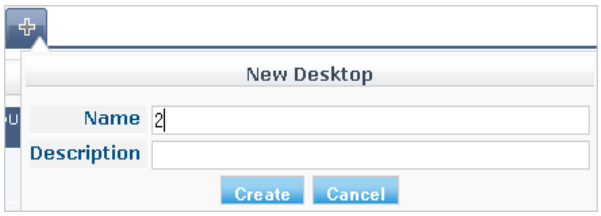
10.2. Adding a Desktop Tab Page

1. Select *Desktop* on the Main Menu.

2. Click the + *New Desktop* button.

Note: You can also right-click the Tab Page label of any existing Tab Page and select *New Desktop* from the menu.

The following screen is displayed:



3. Fill out the fields for the new Tab Page.
The following fields are available:

Field	Meaning
Name	Enter the name of the new Tab Page. By default, the number of the Tab Page is filled in.
Description	Enter a description for the new Tab Page. This description is shown when you point your mouse cursor to the name of the Tab Page.

4. Click *Create* to confirm the creation of the new Tab Page.
You can also click *Cancel* to return to *Desktop Overview* screen without making any changes.

Note: It is possible to create up to 10 tab pages.

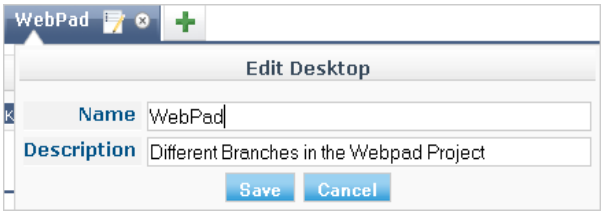
10.3. Editing the Desktop Tab Page Label

1. Select *Desktop* on the Main Menu.

2. Double-click the Tab Page label you want to edit.

Note: You can also right-click the Tab Page label and select *Edit Desktop* from the menu.

The following screen is displayed:



3. Edit the fields for the Tab Page.
   For a detailed description of the fields, refer to Adding a Desktop Tab Page (page 139).

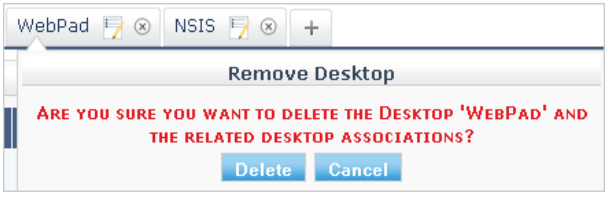
4. Click *Save*, once you have finished your changes.
   You can also click *Cancel* to return to *Desktop Overview* screen without making any changes.

10.4. Removing a Desktop Tab Page

To remove a Desktop Tab Page, click the *Remove Desktop* button.

Note: You can also right-click the Tab Page label and select *Remove Desktop* from the menu.

The following screen is displayed:



Confirm or cancel the deletion of the Tab Page.

10.5. Adding Elements to a Desktop Tab Page

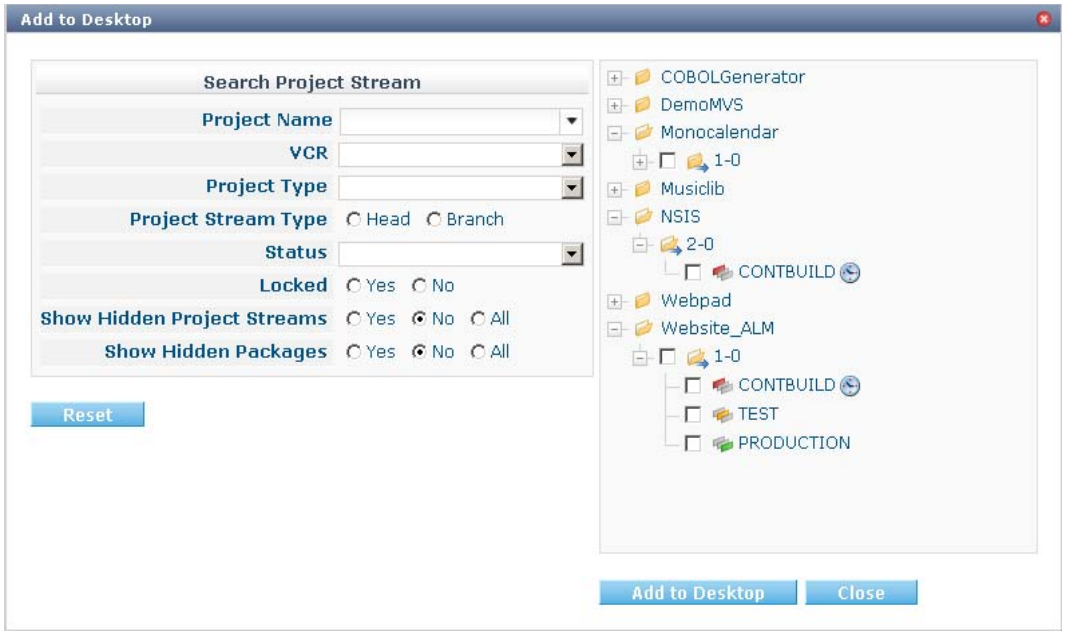
1. Select *Desktop* on the Main Menu.

2. Select the Tab Page you want to add elements to.
   The *Desktop Overview* screen for the selected Tab Page is displayed.

   Note: If you want to add elements to a new Tab Page, you should create the Tab Page first. See Adding a Desktop Tab Page on page 139.

3. Switch to the *Add to Desktop* screen.
   Click the *Add to Desktop* button underneath the *Desktop Overview* screen, or right-click the overview and select *Add to Desktop*.

The following screen is displayed.



4. Define search criteria on the Search Project Stream panel.

If you do not immediately find the element you want to add, you can use the search panel on the left. The following search criteria are available:

Search Criteria	Meaning
Project Name	Enter the name of the Project for which you want to display the Project Streams.
VCR	Enter the VCR name.
Project Type	Enter the type of the Project: Release-based of Package-based.
Project Stream Type	Enter the type of the Project Stream: Head or Branch.
Status	Enter the status of the Project Stream: • Under construction • Planning • Development • Testing • Stable • General available • Frozen • Closed
Locked	Indicate whether or not you want to display locked Project Streams.
Show Hidden Project Streams	Indicate whether or not you want to display hidden Project Streams.
Show Hidden Packages	Indicate whether or not you want to display hidden Packages.

The list of elements displayed in the tree at the right, will be limited to the Project Streams matching these search criteria.

Click the *Reset* button to clear the Search fields.

5. Select the element(s) you want to add.

In the tree at the right, select the Project Stream, Level or Package you want to add and click the *Add to Desktop* button.

5.1. Adding a complete Project Stream

Expand the tree for the corresponding Project using the ⊞ icon to display its Project Streams. Select the Project Stream you want to add and click the *Add to Desktop* button.

If you add a Project Stream to your Desktop, all Levels belonging to that Project Stream will also be available on your Desktop.

5.2. Adding an individual Level

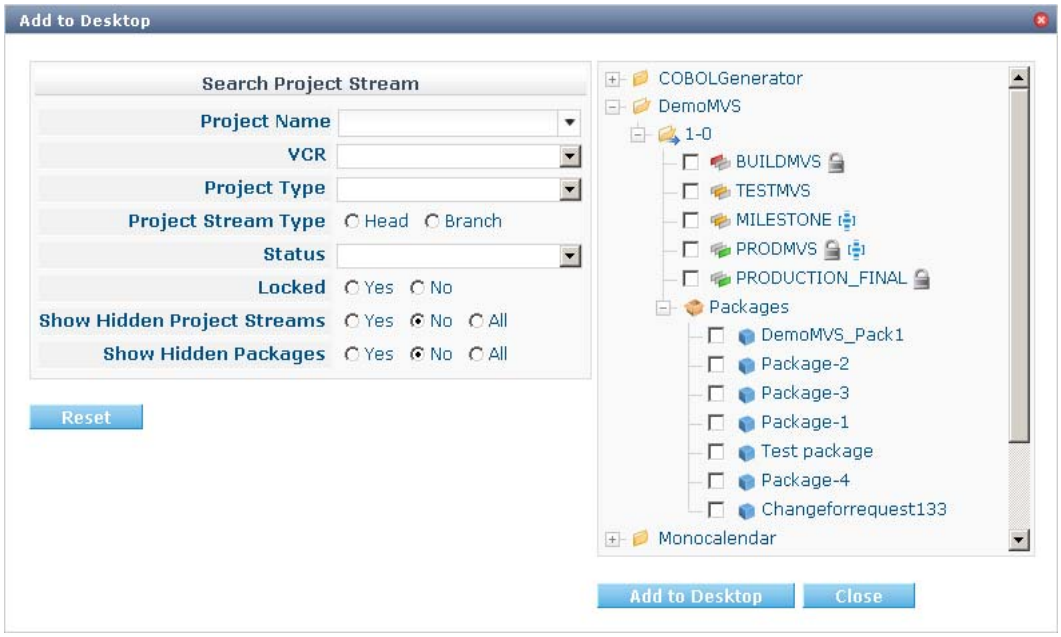
Expand the tree for the corresponding Project Stream using the ⊞ icon to display its Levels (and Packages). Select the Level(s) you want to add and click the *Add to Desktop* button.

5.3. Adding a Package

Expand the tree for the corresponding Project Stream using the ⊞ icon to display its Packages (and Levels). Select the Package(s) you want to add and click the *Add to Desktop* button.

If you add a Package to your Desktop, all Levels belonging to the Project Stream of the Package will also be available on your Desktop.

Example of the selection of a Package and a Project Stream:



When adding an element to the Desktop, the underlying *Desktop Overview* is immediately updated.

Note:	You can select several Project Streams, Levels and Packages at once. You can add maximum 20 elements to each of your Desktop Tab Pages. If you add a Project Stream or Package containing multiple Levels, this Project Stream or Package is considered to be only one element.

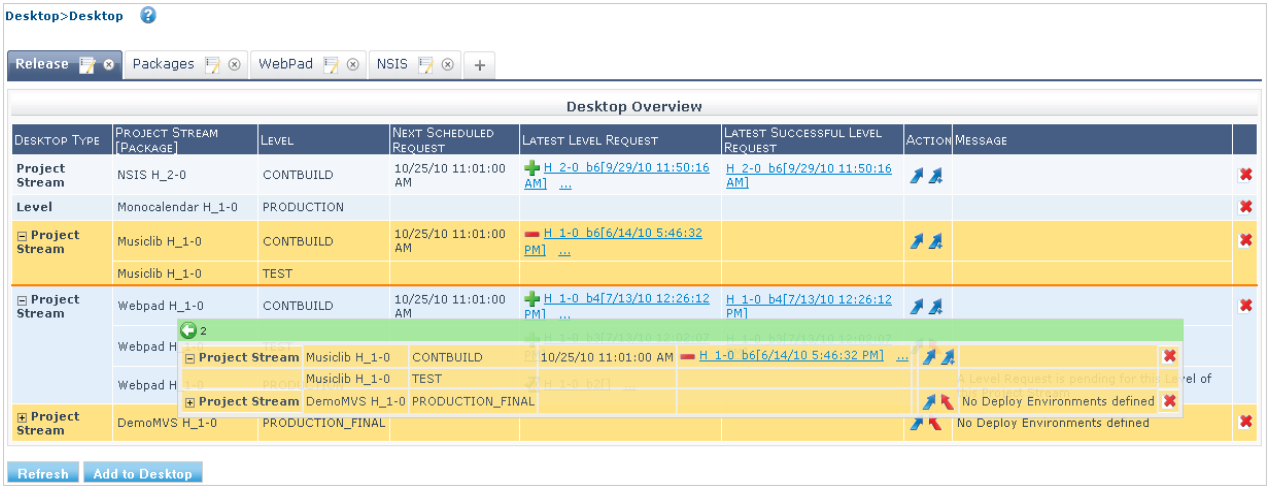
6. Click the *Close* button to return to the *Desktop Overview* screen.

10.6. Rearranging Desktop Elements

To modify the order of the elements, select the element(s) using the *SHIFT* or *CTRL* keys, and, while holding down the left-hand mouse button, drag the element(s) to the required position on the Overview.

Note: The orange line indicates the target position of the selected elements. Valid positions are indicated with a green bar above the selected elements, invalid positions with a red bar.

Example:



10.7. Removing Desktop Elements

To remove a Project Stream, Level or Package, click the *Remove* button at the right of the element row. The Desktop element is immediately removed from the *Desktop Overview* screen.

Project Administration

CHAPTER 11
Project Administration

The following IKAN ALM elements are typically used by a Project Administrator:

- Projects (page 146)
- Project Administration Options (page 155)

11.1. Accessing the Projects Overview

On the Main Menu, click the *Project Administration* icon.
The overview of all projects will be displayed.

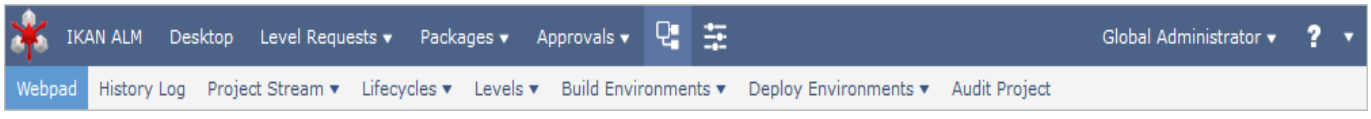
11.2. Accessing the Project Administration Options

From the *Projects Overview*, you can access the Project Administration Options.
In the remainder of this document, we will refer to this section of the interface as the "Project Administration context".

1. On the *Projects Overview*, click the *Edit* link in front of the Project you want to edit.

> **Note:** If you have User Access Rights, you can access this Overview using the *View* link (the available options will be limited).

2. Now you are in the so-called "Project Administration context".
   On the submenu, the different Project Administration options have become available.



CHAPTER 12
Projects

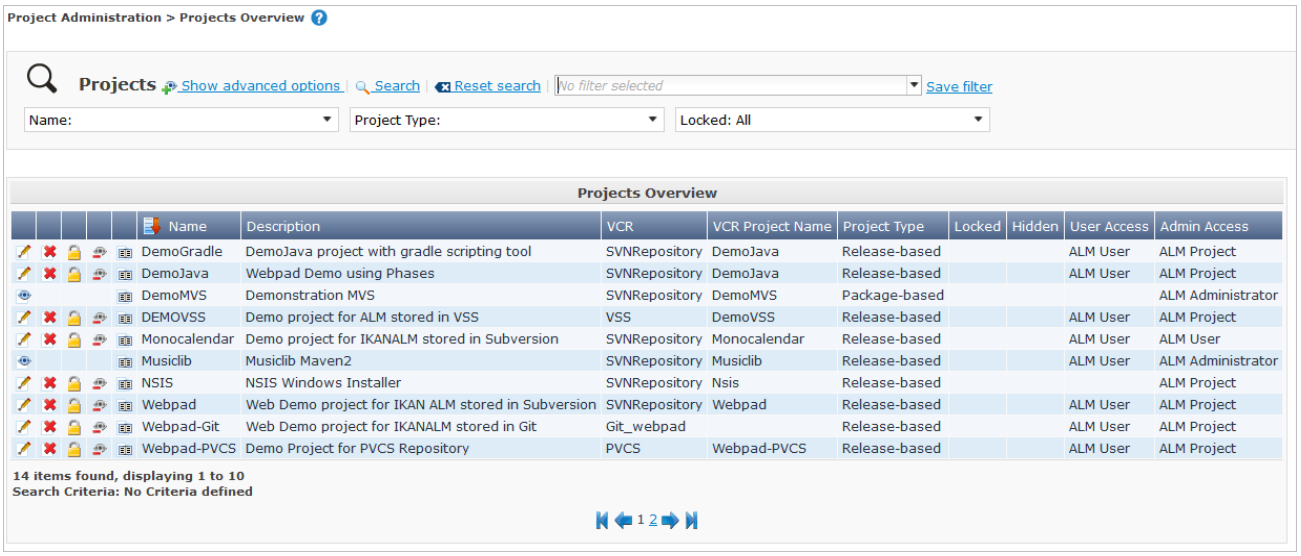
The *Projects Overview* screen provides information about the IKAN ALM Projects. All IKAN ALM Users can display the Projects Overview page. However, only users with view rights (belonging to the Project's Users group) can view the Projects' settings, and only users with Project Administration rights (belonging to the Project's Administrators group) can edit the projects.

Refer to the following procedures for more information:

- The Projects Overview Screen (page 146)
- Viewing Project Settings (page 148)
- Editing Project Settings (page 149)
- Deleting a Project (page 153)
- Locking/Unlocking a Project (page 153)
- Hiding/Showing a Project (page 154)
- Viewing the Project History (page 154)

12.1. The Projects Overview Screen

1. Click the *Project Administration* icon on the Main Menu.
   The *Projects Overview* screen is displayed.



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.

You can also:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- click the *Search* link to refresh the list based on the current search criteria,
- click the *Reset search* link to clear the search fields,
- select a filter from the drop-down menu (if filters have been defined) or use the *Save filter* link to save the current search criteria as a filter for future use.

  For more information on the usage of filters, refer to the section Defining Search Filters (page 16)

3. Verify the available information on the *Projects Overview* panel.

   For a description of the fields, refer to Creating a Project (page 461).

4. Depending on your access rights, the following links may be available on the *Projects Overview* panel:

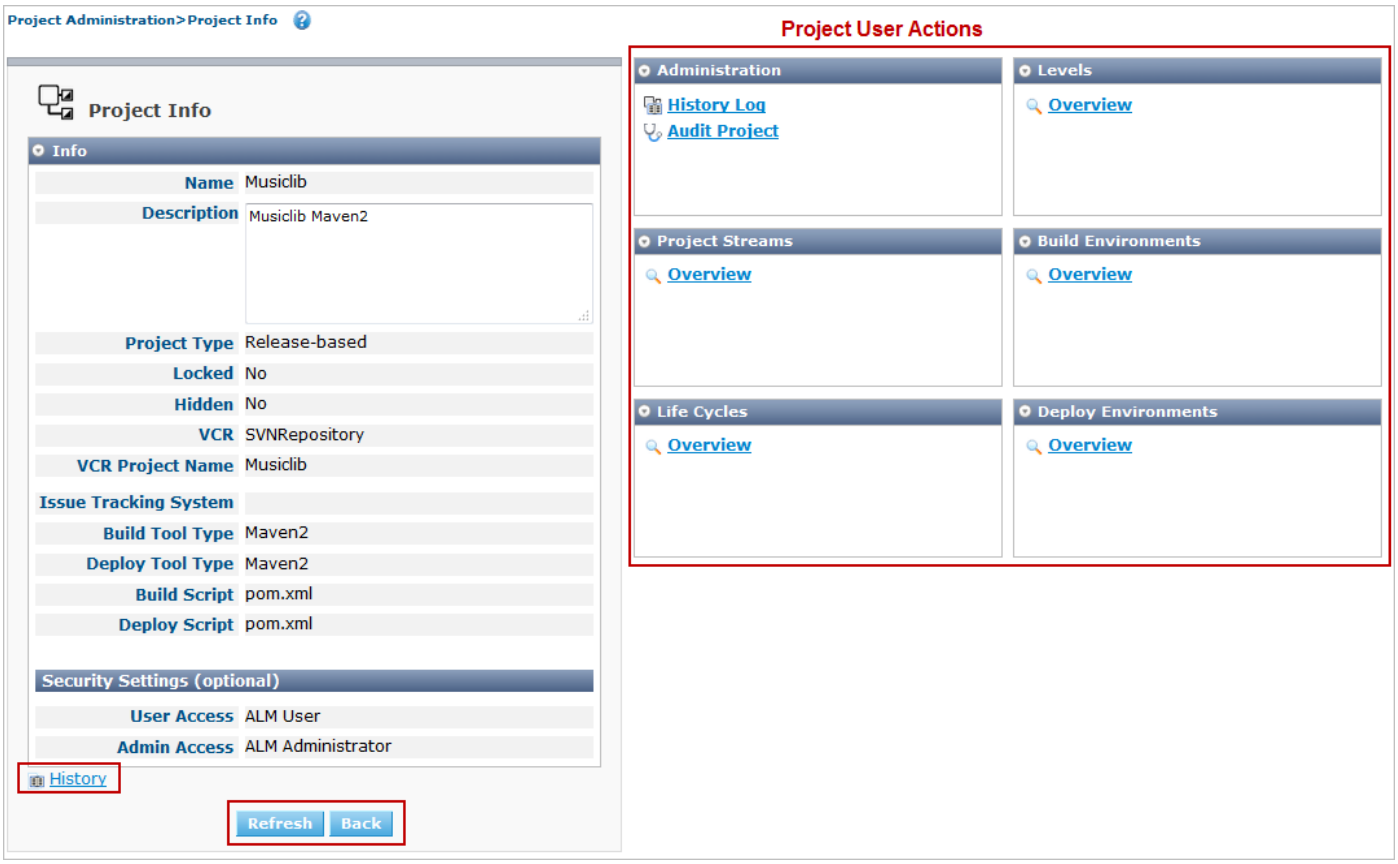
	View This option is available to IKAN ALM Users with Project User Access Rights. It allows viewing the selected Project definition. See Viewing Project Settings on page 148.
	Edit This option is available to all IKAN ALM Users with Project Admin Access Rights. It allows editing the selected Project definition and switching to the other Project Administration options. See Editing Project Settings on page 149.
	Delete This option is available to IKAN ALM Users with Project Admin Access Rights. It allows deleting a Project. See Deleting a Project on page 153.
	Lock / Unlock This option is available to IKAN ALM Users with Project Admin Access Rights. It allows locking or unlocking the selected Project. See Locking/Unlocking a Project on page 153.
	Hide / Show This option is available to IKAN ALM Users with Project Admin Access Rights. It allows hiding the selected Project. See Hiding/Showing a Project on page 154.
	History This option is available to all IKAN ALM Users with Project User Access Rights. It allows to display the History of all create, update and delete operations performed on a Project. See Viewing the Project History on page 154.

12.2. Viewing Project Settings

Note: This option is only available to IKAN ALM Users with User Access Rights.

1. Click the *Project Administration* icon on the Main Menu.

2. Click the *View* link on the *Projects Overview* panel.
   This link is available if you have User Access Rights to the Project.
   The *Project Info* screen is displayed:



3. Verify the settings.
   For a description of the fields, refer to Creating a Project (page 461).
   Underneath the *Project Info* panel, the following links and buttons are available:
   - *History*. This link will display the *Project History View* screen.
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the *Projects Overview* screen.

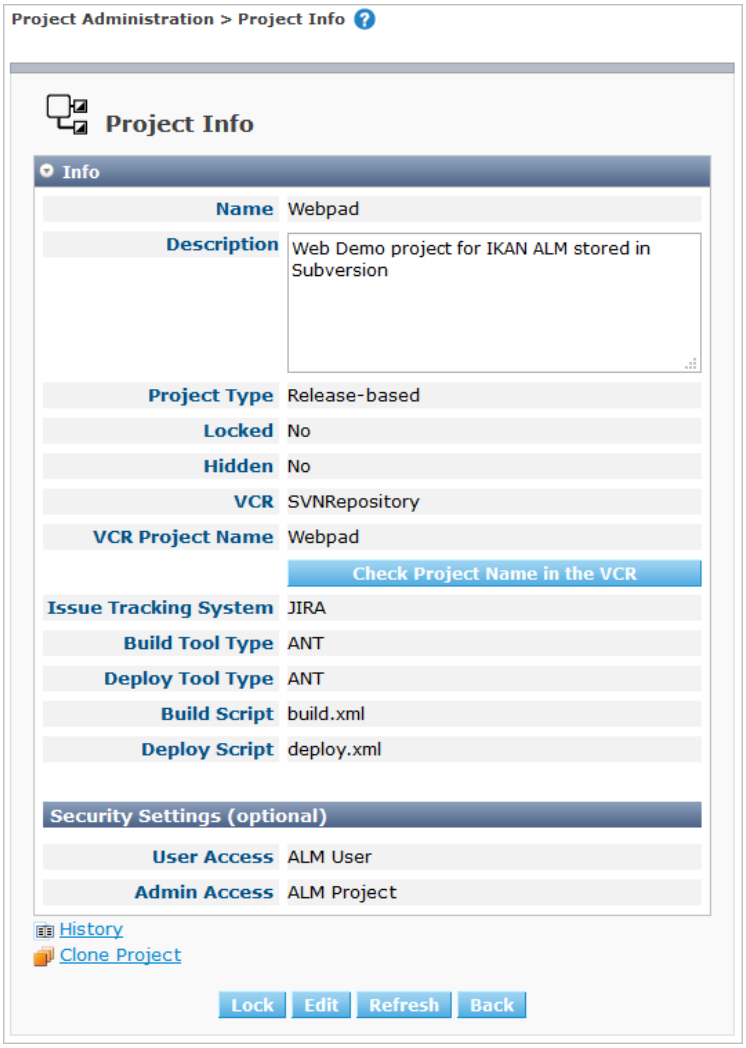
Note: To make it easier to view all project's settings, the most important Project User actions have been grouped in subpanels aside the *Project Info* panel.

12.3. Editing Project Settings

1. Click the *Project Administration* icon on the Main Menu.
2. Click the *Edit* link on the Projects Overview panel.

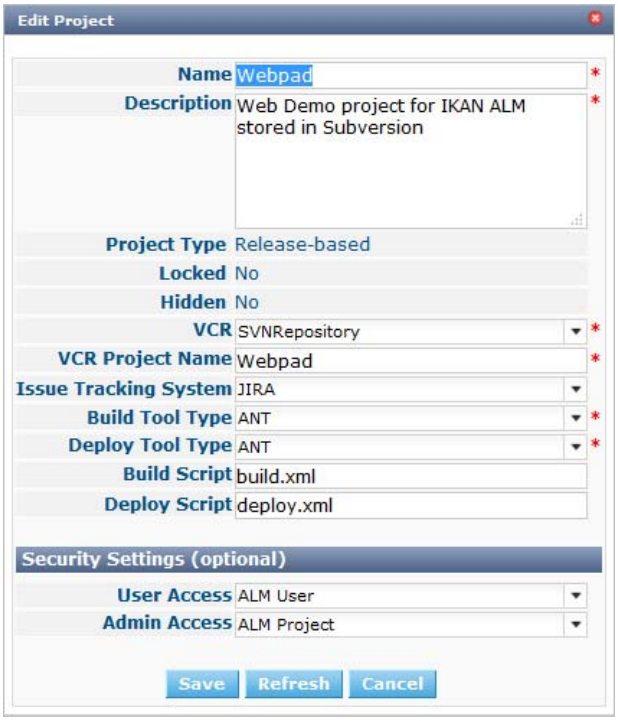
Note: This link is only available if you have Project Admin Access Rights to the Project.

The *Project Info* screen is displayed.



3. Click the *Edit* button.

The *Edit Project* window pops up.



4. Edit the fields as required and click *Save*.

> Once Builds have been created for a Project, changing the Project name has a big impact as all Builds already saved in the Build archive will become inaccessible.
> You can always move the old builds to the new Build Archive location for the different Streams of the project. This will be indicated when you audit the project.

The following fields are available for each Project:

Project Attributes

Field	Meaning
Name	This field contains the Project Name.
Description	This field contains the Project Description.
Project Type	This read-only field contains the Project Type: • Release-based • Package-based The type of the Project is defined at the moment the Project is created (or cloned) by the Global Administrator. See Creating a Project on page 461.
Locked	This read-only field indicates whether or not the Project is locked: Level Requests cannot be created for a locked Project.
Hidden	This read-only field indicates whether or not the Project is hidden. By default newly created Projects are not hidden. For more information on hiding Projects, refer to Hiding/Showing a Project (page 154).
VCR	This field contains the VCR used for the Project.

Field	Meaning
VCR Project Name	This field contains the VCR Project Name.
Issue Tracking System	This field contains the name of the external Issue Tracking System. If an Issue Tracking System is selected, IKAN ALM will from then on automatically create an Issue Tracking Level Phase when a new Level is created for the Project. See Issue Tracking Phase on page 524. This field is only visible when there are external Issue Tracking Systems defined in IKAN ALM. See also Issue Tracking (page 381).
Build Tool Type	This field contains the Build Tool Type used for the Project.
Deploy Tool Type	This field contains the Deploy Tool Type used for the Project.
Build Script	This field **may** contain the Build Script used for the Project. If the Build Script file is located in the root folder of the Project, it refers to a file name directly, e.g., *build.xml* If the Build Script file is located in another folder belonging to the Project, it refers to the relative path to that folder, e.g., */dir1/build/build.xml*. If this field is left blank, the Build Script may be specified on the Build Environment, or as a Phase Parameter of the Execute Script Phase.
Deploy Script	This field **may** contain the Deploy Script used for the Project. If the Deploy Script file is located in the root folder of the Project, it refers to a filename directly, e.g., *deploy.xml* If the Deploy Script file is located in another folder belonging to the Project, it refers to the relative path to that folder, e.g., */dir1/deploy/deploy.xml*. If this field is left blank, the Deploy Script may be specified on the Deploy Environment, or as a Phase Parameter of the Execute Script Phase.
User Access	This field contains the name of the User Group whose members have regular access rights to this Project and are notified when Level Requests are created.
Admin Access	This field contains the name of the User Group whose members have administrative access rights to this Project.

5. You can also click:
   - *Save* to save your changes.
   - *Refresh* to retrieve the settings from the database.
   - *Cancel* to close the pop-up window.

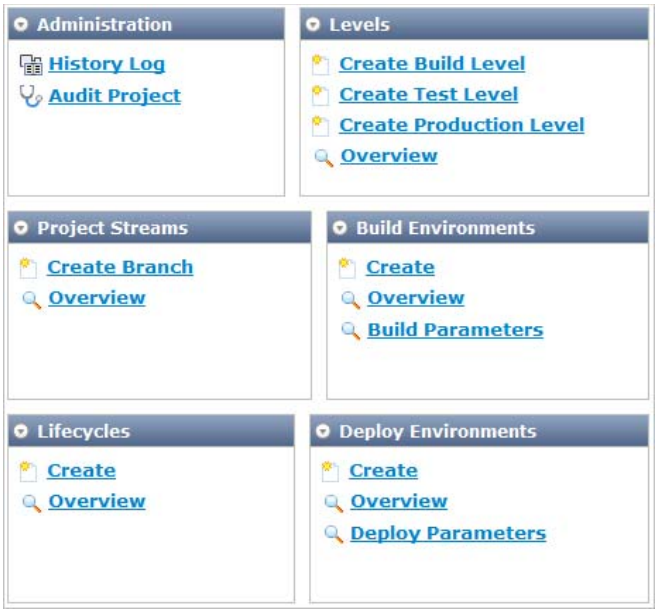
6. When clicking the *Save* or *Cancel* buttons, the *Project Info* page will be displayed.

   On this page the following buttons are available:
   - *Check Project Name in the VCR* to check if the VCR Project name is available. This function will use the settings in the Global Administration connect to the VCR linked to the Project, in order to check if the VCR Project Name exists in the VCR.
   - *History* to display the *Project History View* screen.
   - *Clone Project* to display the *Clone Project* screen. For more information, refer to the section Cloning an Existing Project (page 468).

- *Lock* to lock the Project. This means blocking all activity on that Project, such as manual or scheduled Level Requests. You may want to lock a Project, if configuration parameters are changed, or if administrative actions are required on the VCR.
- *Unlock* to lift the blocking of all activity on this Project, like manual or scheduled Level Request. You must unlock a Project after having completed the maintenance.
- *Refresh* to retrieve the settings from the database.
- *Back* to return to the *Projects Overview* screen.

7. Notice the submenus at the right containing the *Project Administration* options.

To make setting up a project easier, the most important project administration actions have also been grouped in subpanels aside the *Project Info* panel.

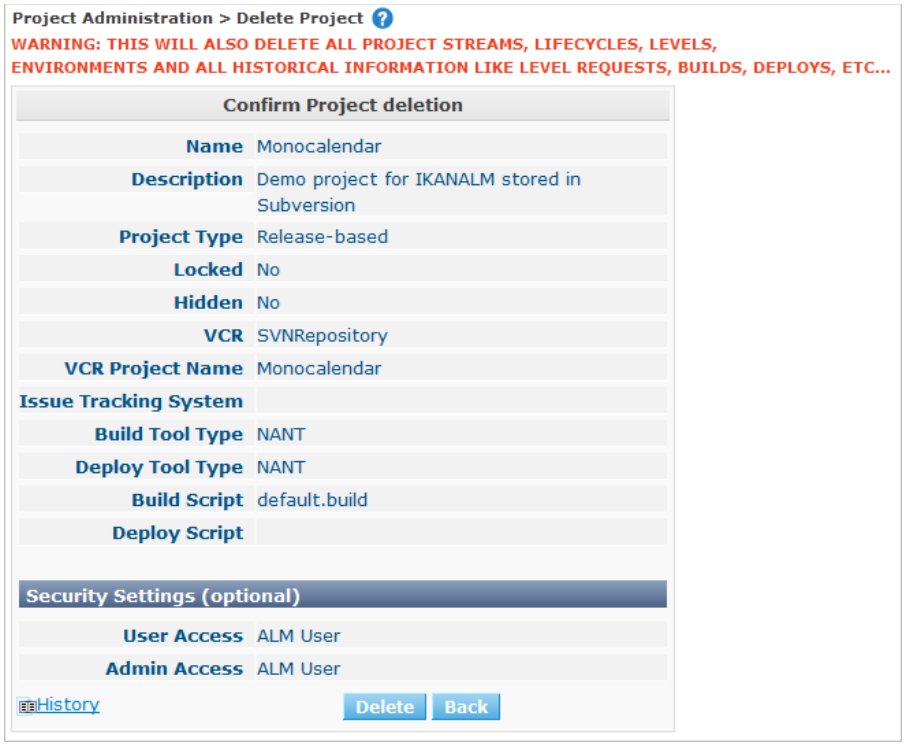


For an overview of the submenu options, refer to section Project Administration Options (page 155).

12.4. Deleting a Project

1. Click the *Project Administration* icon on the Main Menu.

2. Click the *Delete* link on the *Projects Overview* panel.
   The *Delete Project* screen is displayed:



3. Verify if you really want to irrevocably delete the Project.
   Deleting a Project will also delete all Project Streams, Lifecycles, Levels, Environments and all historical information (Level Requests, Builds, Deploys, etc.) of that Project.

4. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the *Projects* screen without deleting the Project.

12.5. Locking/Unlocking a Project

Locking a Project means blocking all activity on that Project, such as manual or scheduled Level Requests.

You may want to lock a Project, if configuration parameters are changed, or if administrative actions are required on the VCR.

Unlocking a Project means reallowing these actions on a Project.

1. Click the *Project Administration* icon on the Main Menu.

2. Click the *Lock* or *Unlock* link on the *Projects Overview* panel.

Note: This functionality is also available via the *Lock/Unlock* button on the *Project Info* panel. See Editing Project Settings on page 149.

12.6. Hiding/Showing a Project

Specifying that a Project is "hidden", causes it not to be displayed by default on Overview panels. This can be very useful to hide older Projects on the overviews, without losing the historical information (Level Requests, …) associated with those Projects.

1. Click the *Project Administration* icon on the Main Menu.

2. Click the *Hide* link in front of the Project you want to hide.
   The Project will disappear from the overview since Hidden Projects will not be shown by default. In order to show it again, you have to set the *Show Hidden Projects* search criterion to *Yes* or *All* in the Search panel.

3. To "unhide" a hidden Project, click the *Show* link.
   The icon will be changed appropriately.

Note: The criterion *Show Hidden Projects* has been added to the Search panels to specify whether or not you want to display hidden Projects. Click the *Show advanced options* link to display this criterion.

12.7. Viewing the Project History

1. Click the *Project Administration* icon on the Main Menu.

2. Click the *History* link on the *Projects Overview* panel to display the *Project History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the previous screen.

Note: This link is also available on the *Project Info* panel. See Viewing Project Settings on page 148.

CHAPTER 13

Project Administration Options

Once you have selected a Project from the Projects Overview, the Project Administration options become available. The available options depend on your access rights.

For each of the Project Administration options (except for *History Log* and *Audit Project*), the following submenu options are available: *Create* and *Overview*.

To avoid repeating the same information in several places, all procedures for creating, editing and deleting elements will be described in detail for each *Create* submenu option. References will be made to this description for the other submenu options.

Submenu option	Description
[Project Name]	The first option on the Project Administration submenu refers to the name of the selected Project or to the Project to which the selected Project Stream belongs. Select this option to return to the *Project Info* screen.
History Log	Select this option to view the History Log of the selected Project. See History Log on page 158.
Project Stream	This drop-down menu contains two options: • *Create Branch* Select this option to create a new Branch for this Project. See Creating a Branch Project Stream on page 161. • *Overview* Select this option to switch to the *Project Streams* screen. See Project Streams on page 161.
Lifecycles	This drop-down menu contains the following options: • *Create* Select this option to create a new Lifecycle. See Creating a Lifecycle on page 175. • Overview Select this option to switch to the *Lifecycles Overview* screen. See The Lifecycles Overview Screen on page 176.
Levels	This drop-down menu contains the following options: • *Create Build Level* See Creating a Build Level on page 188. • *Create Test Level* See Creating a Test or Production Level on page 190. • *Create Production Level* See Creating a Test or Production Level on page 190. • *Overview* See The Levels Overview Screen on page 193.

Submenu option	Description
Build Environments	This drop-down menu contains the following options: • *Create* Select this option to create a new Build Environment for the selected Project. See Creating a Build Environment on page 210. • *Overview* Select this option to switch to the Build Environment Overview for the selected Project. See The Build Environments Overview Screen on page 212. • *Build Parameters* Select this option to manage Build parameters for the selected Project. See Environment Parameters on page 240.
Deploy Environments	This drop-down menu contains the following options: • *Create* Select this option to create a new Deploy Environment for the selected Project. See Creating a Deploy Environment on page 225. • *Overview* Select this option to switch to the Deploy Environment Overview for the selected Project. See The Deploy Environments Overview Screen on page 228. • *Deploy Parameters* Select this option to manage Deploy parameters for the selected Project. See Environment Parameters on page 240.
Audit Project	Select this option to switch to audit the selected Project. See Auditing Projects on page 247.

CHAPTER 14
[Project Name]

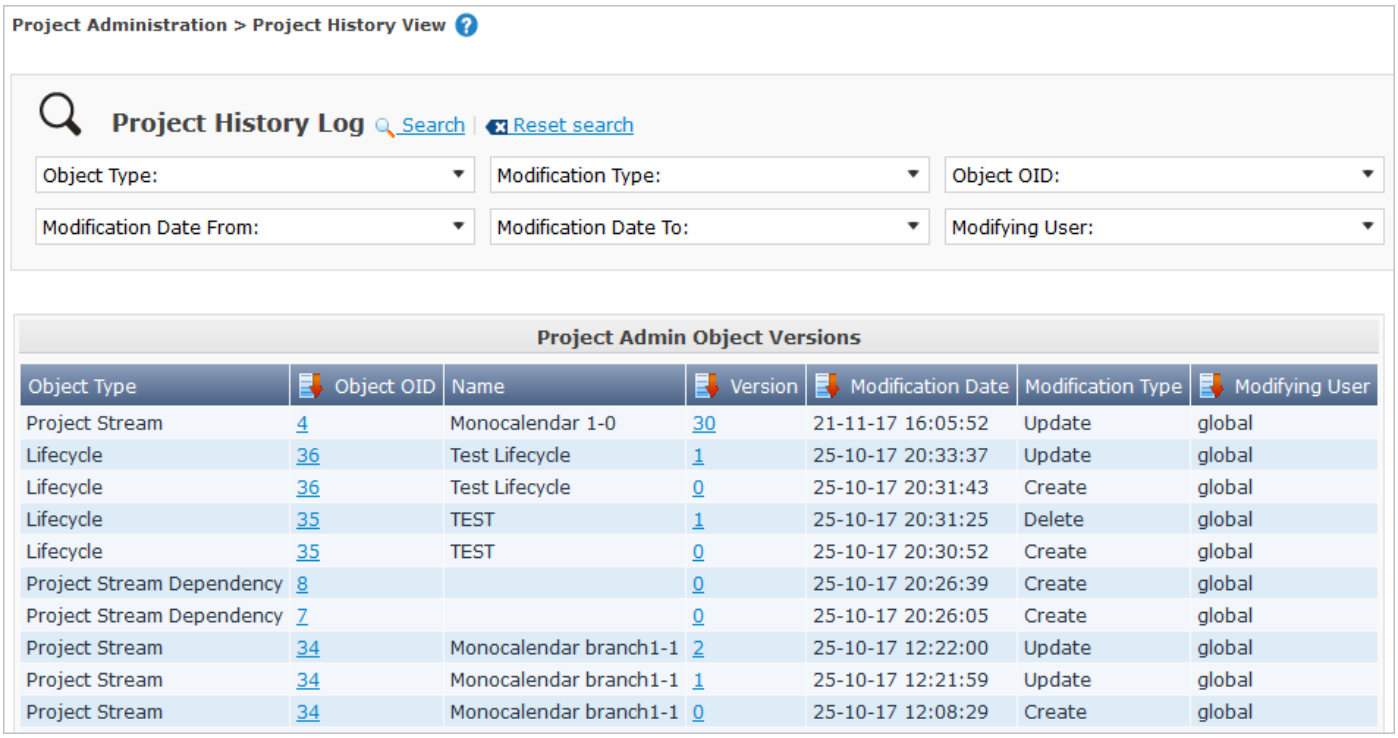
The first option on the Project Administration submenu refers to the name of the selected Project or to the Project to which the selected Project Stream belongs.

Select this option to return to the *Project Info* screen.

For more information, refer to the section Viewing Project Settings (page 148).

CHAPTER 15
History Log

1. In the *Project Administration context*, select *History Log*.
   The *Project History View* screen is displayed:



Project Admin Object Versions

Object Type	Object OID	Name	Version	Modification Date	Modification Type	Modifying User
Project Stream	4	Monocalendar 1-0	30	21-11-17 16:05:52	Update	global
Lifecycle	36	Test Lifecycle	1	25-10-17 20:33:37	Update	global
Lifecycle	36	Test Lifecycle	0	25-10-17 20:31:43	Create	global
Lifecycle	35	TEST	1	25-10-17 20:31:25	Delete	global
Lifecycle	35	TEST	0	25-10-17 20:30:52	Create	global
Project Stream Dependency	8		0	25-10-17 20:26:39	Create	global
Project Stream Dependency	7		0	25-10-17 20:26:05	Create	global
Project Stream	34	Monocalendar branch1-1	2	25-10-17 12:22:00	Update	global
Project Stream	34	Monocalendar branch1-1	1	25-10-17 12:21:59	Update	global
Project Stream	34	Monocalendar branch1-1	0	25-10-17 12:08:29	Create	global

2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. Verify the information on the *Project Admin Object Versions* panel.

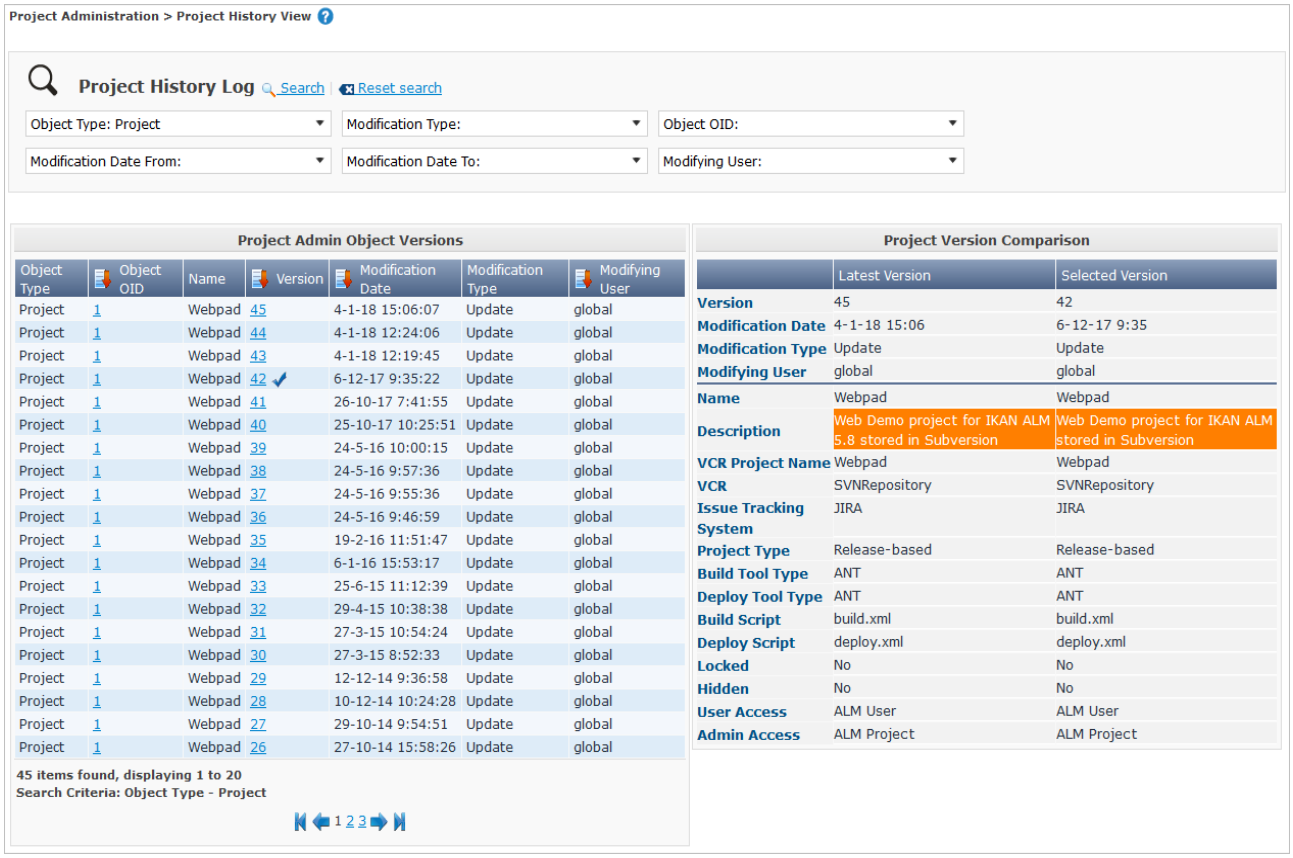
Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

The following information fields are available:

Field	Meaning
Object Type	The type of object that was modified.
Object OID	The OID number of the object. **Note:** Clicking this number will display all modifications for that object.
Name	The name of the object.
Version	The version the object had at the moment the modification was saved.
Modification Date	The date and time at which the modification took place.
Modification Type	The modification type can be one of the following: • *Create* The object was created • *Update* The object was updated • *Delete* The object was deleted
Modifying User	The User ID of the User who performed the modification.

4. Click a specific version on the *Project Admin Object Versions* panel to show the differences between that version and the current version of the object.

   A *Comparison* panel is added to the right of the *Versions* panel.



The *Version Comparison* panel displays the differences between the latest version of the object (shown in the "Latest" column) and the selected version of the object (shown in the "Selected" column). If the Object was deleted, the latest version will be equal to the status of the Object before it was deleted.

- The upper part of the *Comparison* panel displays the Version, Modification Date, Modification Type and User ID of the two versions that are being compared.
- The lower part displays the object-specific values of the two versions that are being compared, highlighting the fields where there are differences.

CHAPTER 16
Project Streams

A Project Stream is a working entity within IKAN ALM. A Level Request is created for a Level linked to a Project Stream. Main development will mostly be done in the Head Project Stream. Maintenance, patches and parallel development can be realized in one or more Branch Project Streams.

When the Global Administrator creates a Project, he or she will also have to define the settings for the Head Project Stream.

The *Project Stream* submenu option lets you start the following procedures:

- Creating a Branch Project Stream (page 161)
- Modifying the Project Stream definition via the Project Streams Overview (page 165)

The *Project Streams Overview* screen lets you verify and change the information concerning Project Streams. The following actions are possible:

- Modifying Project Stream definitions
  - Viewing Project Stream Settings (page 166)
  - Editing Project Stream Settings (page 167)
  - Deleting a Project Stream (page 168)
  - Locking/Unlocking a Project Stream (page 168)
  - Hiding/Showing a Project Stream (page 168)
  - Viewing the Project Stream History (page 169)
- Modifying the Project Stream Dependencies
  - Project Stream Dependencies (page 169)
  - Editing a Project Stream Dependency (page 173)
  - Adding a Project Stream Dependency (page 171)
  - Deleting a Project Stream Dependency (page 173)

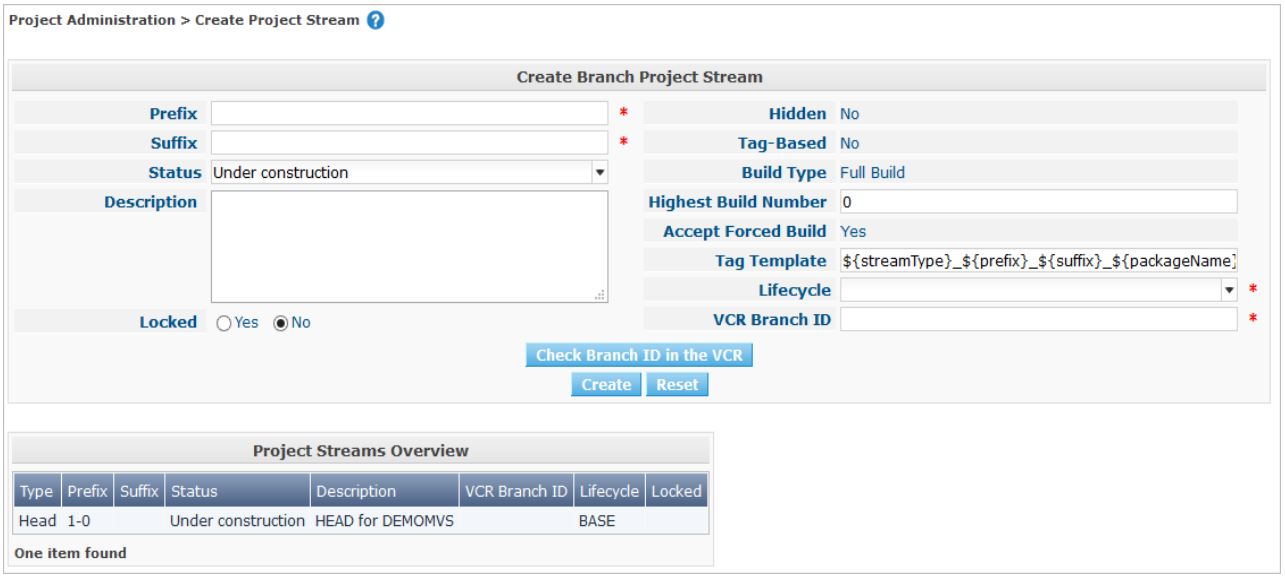
16.1. Creating a Branch Project Stream

You can create a new Branch if you want to

- start a parallel development
- create a maintenance version
- create patches

Create an additional Lifecycle before creating a Branch. See Creating a Lifecycle on page 175.

1. In the *Project Administration context*, select *Project Stream > Create Branch*.
   The following screen is displayed:



At the bottom of the screen, the *Project Streams Overview* panel displays the currently defined Project Streams for the selected Project.

2. Fill out the fields in the *Create Branch Project Stream* panel.
   The following fields are available:

Project Stream Attributes

Field	Description
Prefix	In this field, enter the Build Prefix for the Project Stream. The Build Prefix is used, among other things, for generating unique Build archive file names, and is therefore required.
Suffix	For Branch Streams: enter in this field the Build Suffix. For Head Streams: this field is not available.
Status	In this field, select the status of the new Branch Project Stream from the drop-down list. The possible values are: • under construction • planning • development • testing • stable • general available • frozen • closed
Description	In this field, enter a description for the new Branch Project Stream.
Locked	Indicate whether or not the Project Stream is locked. Locking a Project Stream means blocking all activity on that Project Stream.
Hidden	This field indicates whether or not the Project Stream is hidden. By default newly created Project Streams are not hidden. For more information on hiding Project Streams, refer to Hiding/Showing a Project Stream (page 168).

Field	Description
Tag-Based	Indicate whether or not the Project Stream is tag-based. If the Project Stream is tag-based, it is possible to retrieve sources labeled with a user-defined tag in the head or branch stream for a build action on the Build Level. **Warning:** This functionality requires strict adherence to the internal tagging conventions, as by manual tagging artifacts are no longer under the exclusive control of IKAN ALM. Tags may be defined before a build, but should remain consistent for the affected artifacts for their full Lifecycle. If tag-based builds are activated for the Project Stream, the Tag Template has no effect any more as the VCR Tag will be defined by the User during the Level Request creation process. Furthermore, when the IKAN ALM Scheduler notices that a Build Level is connected to a tag-based Project Stream, it will take no action since it has no reference to check for changes in the versioning system. Triggered builds (for example, after the creation of a tag in the VCR) can be activated via a hook script in the versioning system which will call the IKAN ALM command line to create a Level Request on the Build Level. Therefore, it is recommended to disconnect any Schedule from the Build Level of a tag-based Project Stream. If that is not possible (for example, because the Level is also used in another Project Stream), make sure that *Accept Forced Build* is set to *Yes*. If the Project Stream is Package-based, this field is unchangeably set to *No*, since Tag-Based Builds make no sense in a Package-based project: a Package created in IKAN ALM may be compared with a Tag created in the VCR: it contains one or more selected file revisions out of the Head (trunk) or Branch stream from the VCR.
Build Type	Select the Build type from the drop-down list. The following types are possible: • *Full Build* • *Partial Build*: only the sources that were modified since the last Build will be rebuilt. • *Production-based Partial Build*: only the sources that are different from the version on the Production Level will be rebuilt • *Tag-based Partial Build*: only the sources that are different from the selected *Partial Build VCR Tag* will be rebuilt. This option is absent when creating a Head or Branch Project Stream. It is only available when a successful Build Level Request exists. If the Project Stream is Package-based, this field is fixed to *Full Build*.
Partial Build VCR Tag	Select a VCR Tag from the drop-down list containing the *Successful Build Level Requests* for this Project Stream. This field is mandatory if the option *Tag-based Partial Build* has been selected asBuild Type. Only sources that are different from the selected VCR Tag will berebuilt. This option is absent when creating a Head or Branch Project Stream. It is onlyavailable when a successful Build Level Request exists.
Highest Build Number	In this field, enter the highest Build number for this Project Stream. This number is incremented automatically each time a Level Request is created. This number does not necessarily match the number of successful Builds. Refer to the Build section of the VCR Tag to view the number of successful Builds. If the Project Stream is Package-based, this field is not available.
Partial Build VCR Tag	Select a VCR Tag from the drop-down list of Successful Build Level Requests for this Project Stream. This field is mandatory if the option *Tag-based Partial Build* has been selected as Build Type. Only sources that are different from the selected VCR Tag will be rebuilt. This option is absent when creating a Head or Branch Project Stream. It is only available when a successful Build Level Request exists.

Field	Description
Accept Forced Build	Indicate whether or not the Project Stream accepts Forced Builds, i.e., Builds on the Build Level which overrule the Schedule connected to the Level. If the Project Stream is Package-based, this field is unchangeably set to *No*. A Force Build make no sense since it is not possible to set a Schedule on a Build Level in a Package-based Project.
Tag Template	In this field, enter the Template that should be used for tagging Builds generated in this Project Stream. The Tags will be used to display these Builds in the VCR. A template can be a combination of *constants* and *variables*. The following *constants* are allowed: • letters • numbers • underscores • hyphens • Some other characters (such as , ; $ @) may be illegal, depending on the VCR type. The following *variables* are available. They all have the format ${name}: • ${prefix}: This variable returns the Build Prefix defined for the Project Stream. • ${suffix}: This variable returns the Build Suffix defined for the Branch Stream. (If this variable is included in a Template for a Head Stream, it is ignored.) • ${streamType}: This variable returns the Type indication for a Stream: *H* for Head Stream or *B* for Branch Stream. • ${buildNumber}: This variable returns the number of the Build, so that the Build can be matched with its Tag. • ${projectName}: This variable returns the name of the IKAN ALM Project. • ${vcrProjectName}: This variable returns the VCR name of the Project. • ${dateTime(dateformat)}: This variable returns the timestamp of the Building action in the indicated format. *yyyy-MM-dd* is a valid format. For other valid formats, refer to http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html • ${packageName}: This variable returns the name of the package in a Package-based Project Stream.
Lifecycle	Select the Lifecycle to be associated with this Project Stream.
VCR Branch ID	For Branch Streams, enter in this field the VCR Branch ID. The same ID must be available in the VCR linked to the Project. For Head Streams belonging to a ClearCase UCM Project, enter the name of the UCM Integration Stream. For Head Streams belonging to a Team Foundation VC Project, enter the name of the Main Branch ID. For Head Streams belonging to other Project types, this field is not available.

3. Click *Create* to create the new Project Stream.

   You can also click:

   - *Check Branch ID in the VCR* to check if the Branch ID is available in the VCR linked to the Project.
   - *Reset* to clear the fields and restore the initial values.

4. Additional Information

   Underneath the *Create Branch Project Stream* panel, you will find an overview of the existing Project Streams in the Project.

   For more information, refer to The Project Streams Overview Screen (page 165).

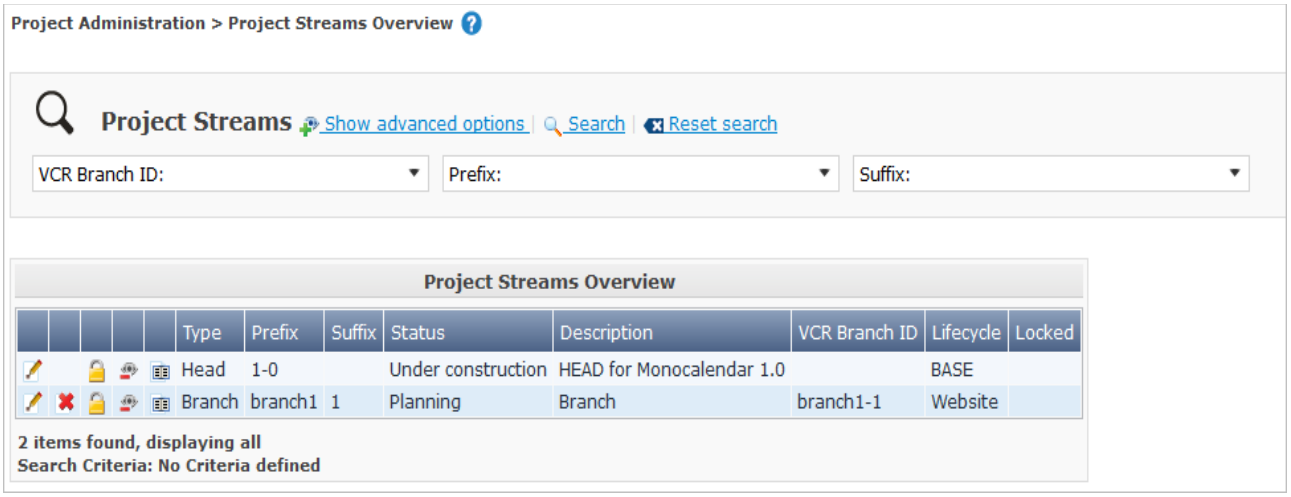
16.2. Project Streams Overview

The *Project Streams Overview* screen lets you verify and change the information concerning Project Streams. The following actions are possible:

- Modifying Project Stream definitions
  - Viewing Project Stream Settings (page 166)
  - Editing Project Stream Settings (page 167)
  - Deleting a Project Stream (page 168)
  - Locking/Unlocking a Project Stream (page 168)
  - Hiding/Showing a Project Stream (page 168)
  - Viewing the Project Stream History (page 169)
- Modifying the Project Stream Dependencies
  - Project Stream Dependencies (page 169)
  - Editing a Project Stream Dependency (page 173)
  - Adding a Project Stream Dependency (page 171)
  - Deleting a Project Stream Dependency (page 173)

The Project Streams Overview Screen

1. In the *Project Administration context*, select *Project Stream > Overview*.
   The *Project Streams Overview* screen is displayed:



   At the bottom of the screen, the *Project Streams Overview* panel displays the currently defined Project Streams for the selected Project.

   For a detailed description of the fields and the links, refer to Creating a Branch Project Stream (page 161).

2. Depending on your access rights, the following links may be available:

Edit
This option is available to IKAN ALM Users with User or Project Administrator Access Rights. It allows viewing (page 167) or editing the selected Project Stream definition (page 167).

Delete
This option is available to IKAN ALM Users with Project Administrator Access Rights. It allows deleting the selected Project Stream. See Deleting a Project Stream on page 168.

Lock / Unlock
This option is available to IKAN ALM Users with Project Administrator Access Rights. It allows locking or unlocking the selected Project Stream. See Locking/Unlocking a Project Stream on page 168.

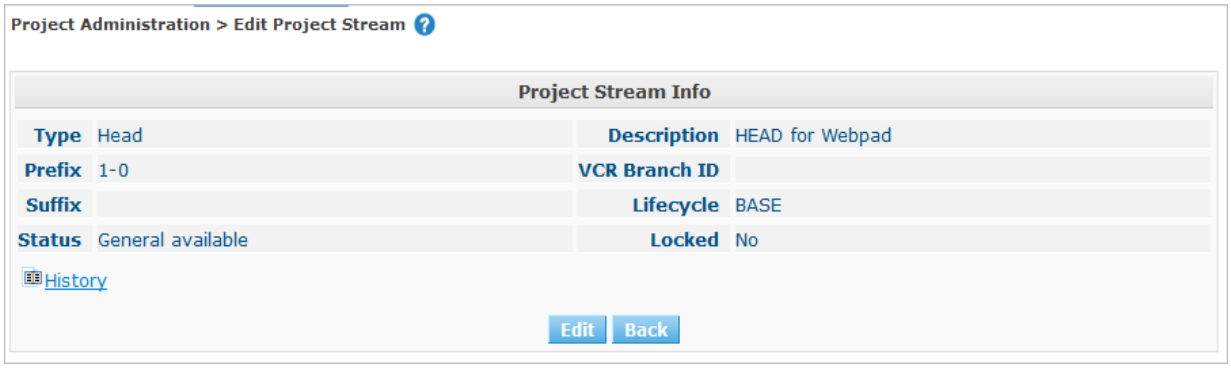
Hide / Show
This option is available to IKAN ALM Users with Project Administrator Access Rights. It allows hiding or showing the selected Project Stream.
See Hiding/Showing a Project Stream on page 168.

History
This option is available to all IKAN ALM Users with Project User Access Rights. It allows to display the History of all create, update and delete operations performed on a Project Stream.
See Viewing the Project Stream History on page 169.

Viewing Project Stream Settings

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. Click the *Edit* link in front of the required Project Stream.
   The *Edit Project Stream* screen is displayed:



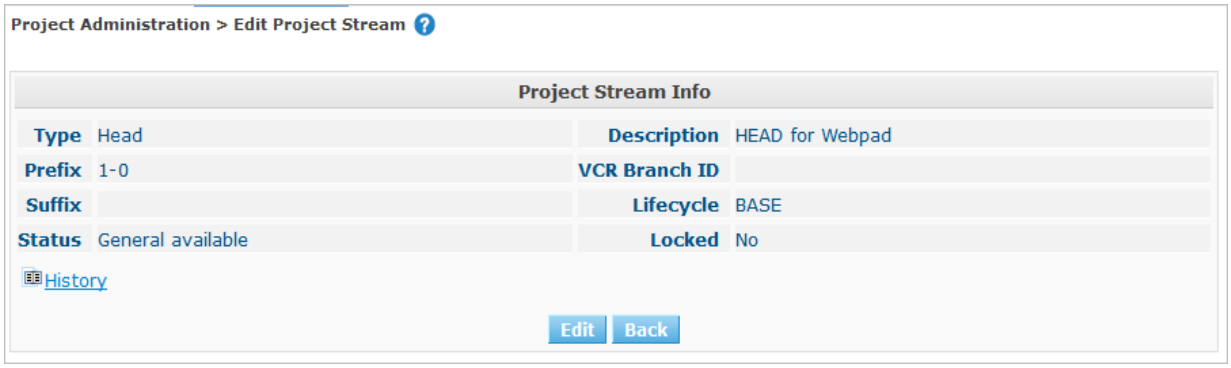
3. Verify the settings.
   For a description of the fields, refer to Creating a Branch Project Stream (page 161).

4. Click *Back* to return to the *Project Streams Overview* screen.

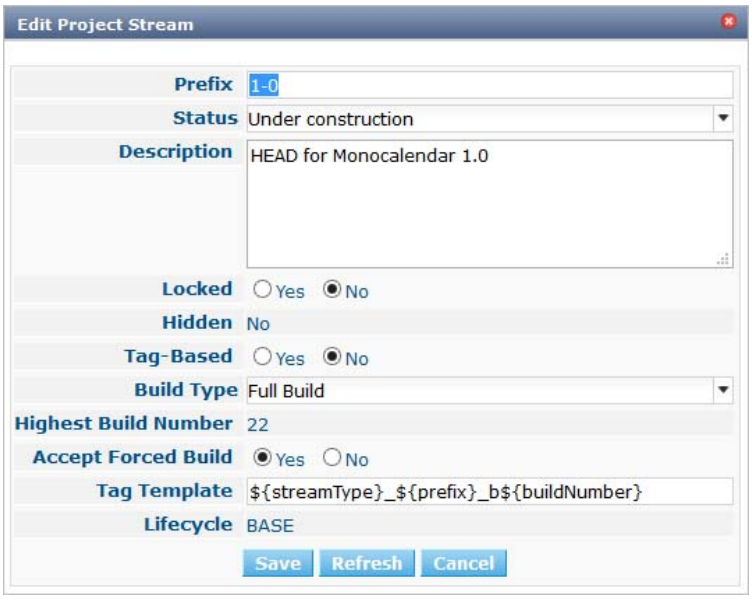
Editing Project Stream Settings

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. Click the *Edit* link in front of the required Project Stream.
   The *Edit Project Stream* screen is displayed:



3. Click the *Edit* button on the *Project Stream Info* panel to edit the fields.



   For a description of the fields, refer to Creating a Branch Project Stream (page 161).

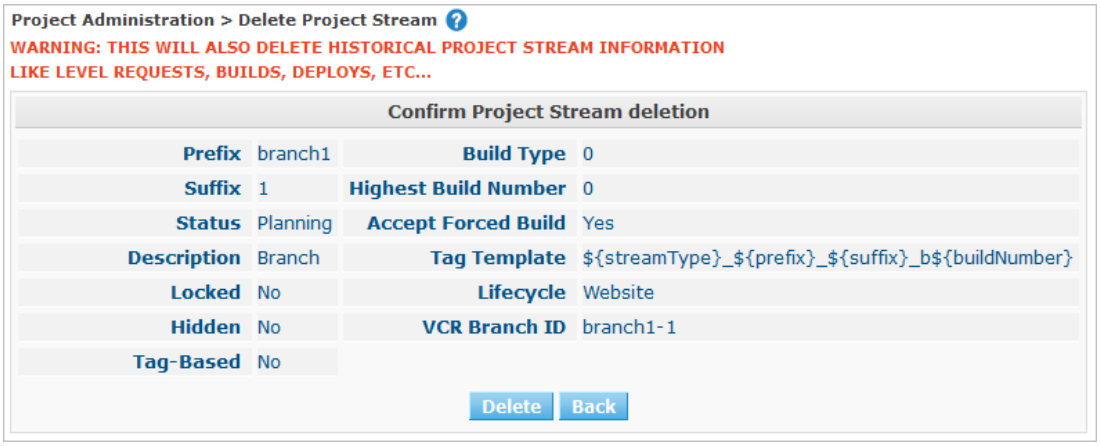
4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Cancel* to return to the previous screen without saving the changes

RELATED TOPICS
- Project Stream Dependencies (page 169)
- Adding a Project Stream Dependency (page 171)
- Editing a Project Stream Dependency (page 173)
- Deleting a Project Stream Dependency (page 173)

Deleting a Project Stream

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. Click the *Delete* link on the *Projects Streams Overview* panel.
   The following screen is displayed:



3. Verify if you really want to irrevocably delete the Project Stream.

4. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the Project.

Locking/Unlocking a Project Stream

Locking a Project Stream means blocking all activity on that Project Stream, such as manual or scheduled Level Requests.

You may want to lock a Project Stream, if configuration parameters have changed, or if administrative actions are required on the VCR. Unlocking a Project Stream means reallowing these actions on a Project Stream.

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. Click the *Lock* or *Unlock* link on the *Project Streams Overview* panel.

Hiding/Showing a Project Stream

Specifying that a Project Stream is "hidden", adds an extra search attribute to that Project Stream. This can be very useful to leave out older Project Streams on the *Overview* panels.

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. Click the *Hide* link in front of the Project Stream you want to hide.
   The Project Stream will be hidden and disappears from the *Project Streams Overview*.

3. To "unhide" a hidden Project Stream, set the *Hidden* criterion to *Yes* or *All* and click the *Show* link.

   The Project Stream will become visible and the icon will be changed appropriately.

Note: A criterion has been added to the *Search* panels to specify whether or not you want to display hidden Projects. Click the *Show advanced options* link to display this criterion.

Viewing the Project Stream History

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. Click the *History* link on the *Project Streams Overview* panel to display the *Project Stream History View*.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

   Click *Back* to return to the previous screen.

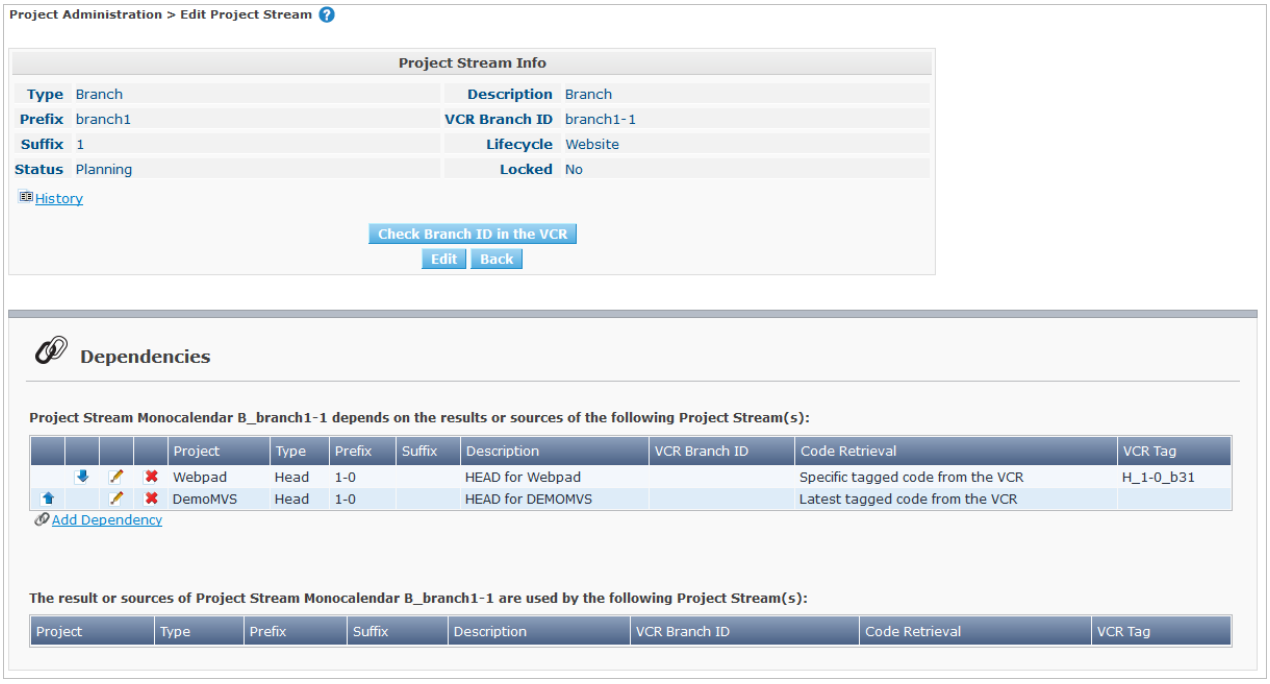
Project Stream Dependencies

Project Stream dependencies allow the reuse of common Projects (e.g., libraries) by other Projects. When the code in the Master Project Stream is built, the process will use the source code or the build result from the Child Project Stream(s), which is stored in the Source Environment in a new folder having the same name as the name of the Project in the VCR.

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. On the *Project Streams Overview*, click the *Edit* link in front of the required Project Stream.

   The *Edit Project Stream* screen is displayed, containing the *Project Stream Info* and the *Dependencies* panels.



The Dependencies panel contains:

- the overview of the Project Streams whose results or sources the current Project Stream depends on,
- the overview of the Project Streams using the results or sources of the current Project Stream, and
- the *Add Dependency* link. See Adding a Project Stream Dependency on page 171.

For each of the Project Streams on the Overviews, the following information is available:

Field	Description
Project	This field displays the name of the Project to which the Project Stream belongs.
Type	This field displays the Project Stream Type. There are two possibilities: • Head • Branch
Prefix	This field contains the Build Prefix defined for this Project Stream.
Suffix	This field contains the Build Suffix for this Branch. It is empty for the Head Stream. This suffix will be attached to the Build Prefix for easy identification of the Project Stream.
Description	This field contains the user-defined description of the Project Stream.
VCR Branch ID	This field contains the VCR Branch ID for this Project Stream. This field remains empty for Head Project Streams.
Code Retrieval	This field indicates how the Child Project Stream code will be retrieved. • *Latest tagged code from the VCR* • *Specific tagged code from the VCR*: If this option has been selected, the selected tag is displayed in the *VCR Tag* field. • *Latest tagged build from the Build Archive* • *Specific build from the Build Archive*: If this option has been selected, the selected tag is displayed in the *VCR Tag* field. • *Latest Production code from the VCR* • *Latest Production Build from the Build Archive*
VCR Tag	This field contains the Project Stream VCR Tag used to retrieve the code. Refer to the *Code Retrieval* field explanation above.

3. Use the links on the *Project Stream XXX depends on the results or sources of the following Project Stream(s)*, if required.

The following links may be available:

- The *Up* and *Down* links to modify the order of the Project Streams.
- *Edit*: This option allows editing the selected dependency. See Editing a Project Stream Dependency on page 173.
- *Remove*: This option allows removing the selected dependency. See Deleting a Project Stream Dependency on page 173.

Note: There are no links available on the *The result or sources of Project Stream XXX are used by the following Project Stream(s)* panel. Switch to the *Project Stream Dependency* screen of these Master Project Streams to edit their Dependency settings.

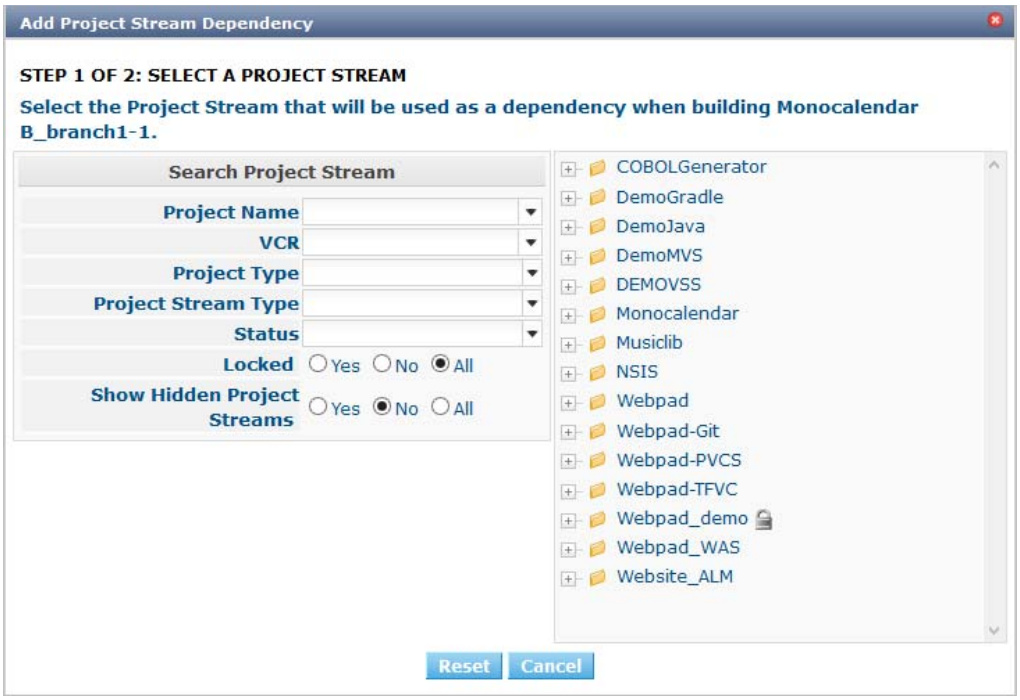
4. Click *Back* to return to the previous screen.

Adding a Project Stream Dependency

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. On the *Project Streams Overview*, click the *Edit* link in front of the required Project Stream.

3. Click the *Add Dependency* link on the *Dependencies* panel.
   The following pop-up screen is displayed:

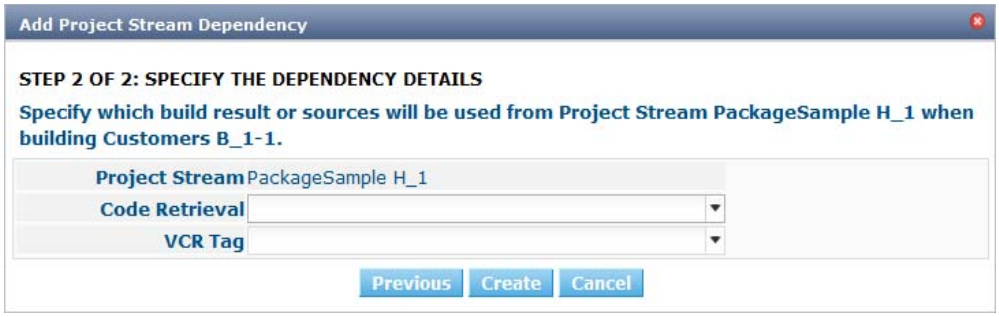


4. Define search criteria on the *Search Project Stream* panel in case you do not immediately find the Project Stream you want to add.
   The list of elements displayed in the tree at the right, will be limited to the Project Streams matching these search criteria.
   Click the *Reset* button to clear the Search fields.

5. Expand the tree for the corresponding Project using the ⊞ icon to display its Project Streams. Select the Project Stream you want to add and click *Next*.

Note: If a Project Stream can not be added, an error message will be displayed.

6. Specify the Build results or sources to be used.



6.1. Select the *Code Retrieval* mode from the drop-down list.

From the drop-down list, select the required code retrieval type. The following options are available:

- *Latest tagged code from the VCR*

  Select this option, if you want to use the **source code** of the last successful Build of the Child Branch to build the code of the Master Project.

- *Specific tagged code from the VCR*

  Select this option, if you want to use the **source code** of the Child Branch tagged with the VCR tag selected in the *VCR Tag* field below to build the code of the Master Project. The VCR Tag field is required.

- *Latest tagged build from the Build Archive*

  Select this option, if you want to use the **Build result** of the last successful Build of the Child Branch to build the code of the Master Project. The VCR Tag field must remain empty.

- *Specific build from the build archive*

  Select this option, if you want to use the **Build result** of the Build of the Child Branch with the selected VCR Tag to build the code of the Master Project. The VCR Tag field is required.

- *Latest Production code from the VCR*

  Select this option, if you want to use the **source code** that was tagged with the VCR tag of the Build that is currently Active on the last Level (usually a Production Level) of the Lifecycle of the Child Project Stream to build the code of the Master Project.

- *Latest Production Build from the Build Archive*

  Select this option, if you want to use the **Build result** of the Build that is currently Active on the LAST Level of the Lifecycle of the Child Project Stream to build the code of the Master Project.

6.2. Select the *VCR Tag* from the drop-down list.

From the drop-down list, select the required VCR Tag.

The VCR Tag is only relevant if the Code Retrieval field is set to Specific tagged code from the VCR or Specific build from the Build Archive.

Note: This option can only be used if a Build already exists as otherwise no VCR tag is available yet.

7. Click *Create* to create the Project Stream Dependency.

You can also click:

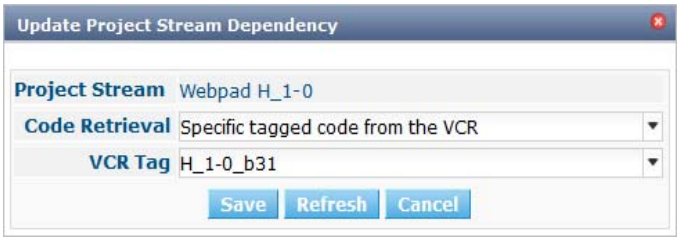
- *Cancel* to clear the fields and restore the initial values.
- *Previous* to return to the previous screen without saving the changes.

Editing a Project Stream Dependency

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. On the *Project Streams Overview*, click the *Edit* link in front of the required Project Stream.

3. Click the *Edit* link in front of the required Dependency.
   The *Update Project Stream Dependency* pop-up screen is displayed:



4. Edit the *Code Retrieval* mode and *VCR Tag*, if required.
   For a description of these fields, refer to the section Adding a Project Stream Dependency (page 171).

5. Click *Save* to save your modifications.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Cancel* to return to the previous screen without saving the changes.

Deleting a Project Stream Dependency

1. In the *Project Administration context*, select *Project Stream > Overview*.

2. On the *Project Streams Overview*, click the *Edit* link in front of the required Project Stream.

3. Click the *Remove* link in front of the required Dependency.
   A confirmation message appears to remove the Dependency. Click *Yes* if you want to remove the dependency or *No* if you want to cancel the action. When clicking Yes, the Project Stream will be immediately removed from the list.

CHAPTER 17
Lifecycles

Each Project Stream has a Lifecycle connected to it. This Lifecycle defines the logical steps of the ALM processes. Such a logical step is called a Level in IKAN ALM. The Lifecycles Submenu lets you define Lifecycles and set up the sequence of Levels for each Lifecycle.

During the creation process of a Project, the Head Project Stream is created automatically, as well as the BASE Lifecycle connected to that Head Project Stream.

Three different Level types can be connected to a Lifecycle: Build, Test or Production Levels. At least one Build Level must have been assigned to a Lifecycle before Level Requests can be created in a Project Stream connected to that Lifecycle. The number of Test and Production Levels in a Lifecycle is unlimited.

Levels may be reused in other Lifecycles. For example, an "Urgency Fix" or "Patch" Lifecycle can contain the same Build and Production Levels as the normal Lifecycle, but the Test Levels in between may be skipped.

Note: The BASE Lifecycle cannot be reused in other (Branch) Project Streams, so a new Lifecycle must be created before creating a Branch Project Stream.

The *Lifecycles* submenu option lets you start the following procedures:

- Creating a Lifecycle (page 175)
- Lifecycles Overview Screen (page 176)

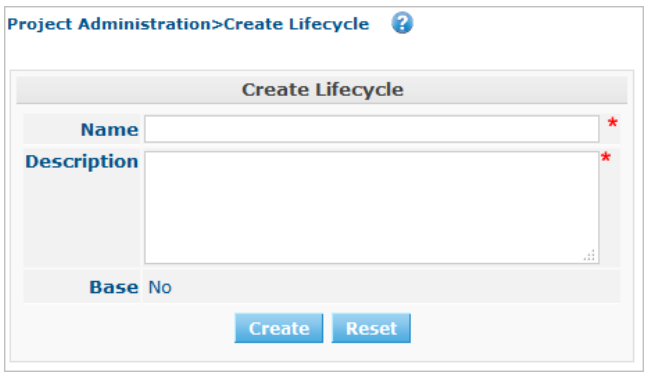
The *Lifecycles Overview* screen lets you verify and change the information concerning Lifecycles and their related Levels. The following actions are possible:

- Modifying Lifecycle definitions
  - Editing a Lifecycle (page 177)
  - Deleting a Lifecycle (page 182)
- Modifying the contents of a Lifecycle
  - Creating Additional Levels (page 180)
  - Inserting an Existing Level (page 180)
  - Changing the order of the Levels (page 181)
  - Making a Level optional or required (page 181)
  - Enabling or Disabling Optional Deploys (page 182)
- Modifying Levels connected to the Lifecycle
  - Editing a Level (page 184)
  - Removing a Level (page 186)

Note: Levels (Build, Test and Production) can also be created via the *Levels* submenu option. the advantage of creating them at this stage, via the *Edit Lifecycle* screen, is that the Level will automatically be connected to the Lifecycle.

17.1. Creating a Lifecycle

1. In the *Project Administration context*, select *Lifecycles > Create*.
   The *Create Lifecycle* panel is displayed at the top of the *Create Lifecycle* screen:



   The following fields are available. Fields marked with an asterisk are mandatory.

Field	Meaning
Name	In this field, enter the name of the new Lifecycle.
Description	In this field, enter a description for the new Lifecycle.
Base	This read-only field is always set to *No*, because the BASE Lifecycle is automatically created during project creation. It is not possible to create additional BASE Lifecycles.

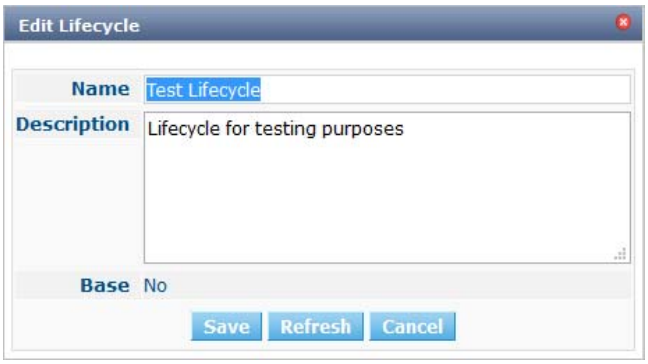
2. Fill out the fields as required and click *Create*.

Note: You can also click *Reset* to empty the fields and restore the initial values.

   The new Lifecycle is added to the list of Lifecycles for the selected Project.

3. Additional Information
   Underneath the *Create Lifecycle* panel, you will find an overview of the existing Lifecycles.
   For more information on this panel, refer to Lifecycles Overview Screen (page 176).

4. Click the Create button to create the Lifecycle.
   The *Edit Lifecycle* screen is displayed.



   For more information, refer to the section Editing a Lifecycle (page 177).

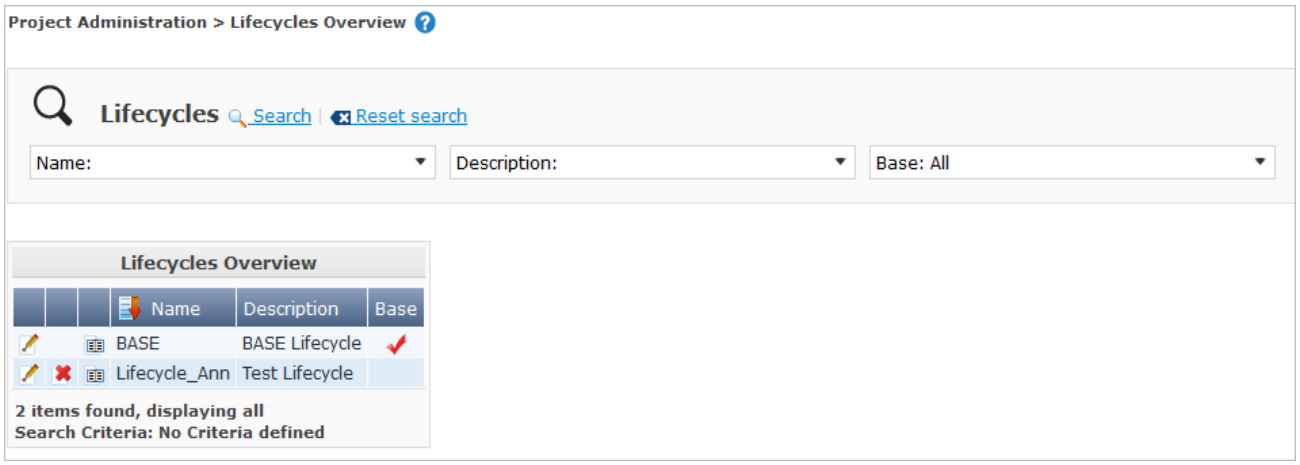
17.2. Lifecycles Overview Screen

The *Lifecycles Overview* screen lets you verify and change the information concerning Lifecycles and their related Levels. The following actions are possible:

- Modifying Lifecycle definitions
  - Editing a Lifecycle (page 177)
  - Deleting a Lifecycle (page 182)
  - Viewing the Lifecycle History (page 184)

The Lifecycles Overview Screen

1. In the *Project Administration context*, select *Lifecycles > Overview*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. On the Lifecycles Overview panel, verify the Lifecycle Information fields.
   For a description of the fields, refer to the section Creating a Lifecycle (page 175).

4. Depending on your access rights, the following links may be available on the *Lifecycles Overview* panel:

	Edit This option is available to IKAN ALM Users with Project Administrator Access Rights. It allows editing the Lifecycle definition. See Editing a Lifecycle on page 177.

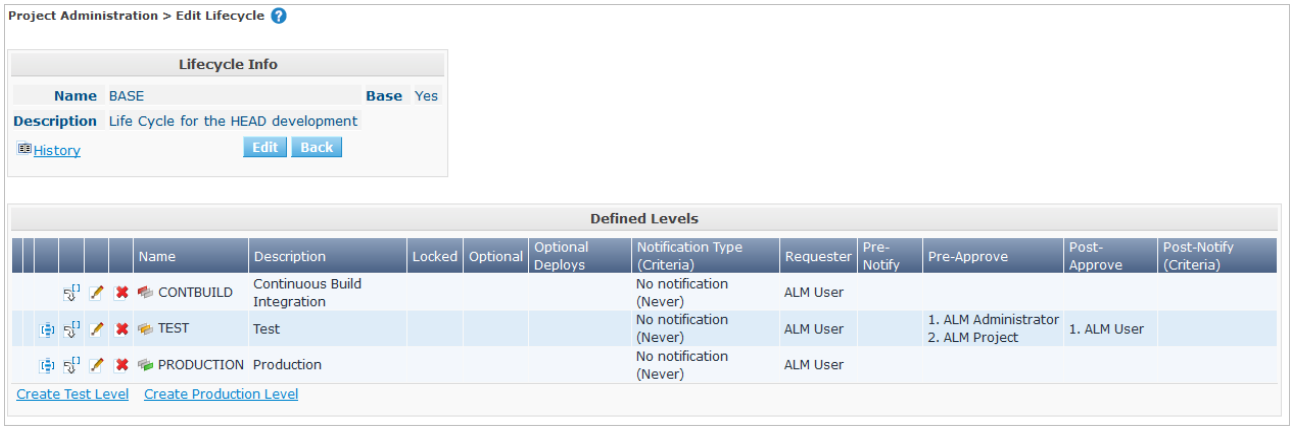
	Delete This option is available to IKAN ALM Users with Project Administrator Access Rights. It allows deleting a Lifecycle definition, except for the BASE Lifecycle that is connected to the Head Project Stream. See Deleting a Lifecycle on page 182.
	History This option is available to all IKAN ALM Users. It allows to display the History of all create, update and delete operations performed on a Lifecycle. See Viewing the Project History on page 154.

5. Verify the information on the *Defined Levels* panel.
   For more information, refer to the section Editing a Lifecycle (page 177).

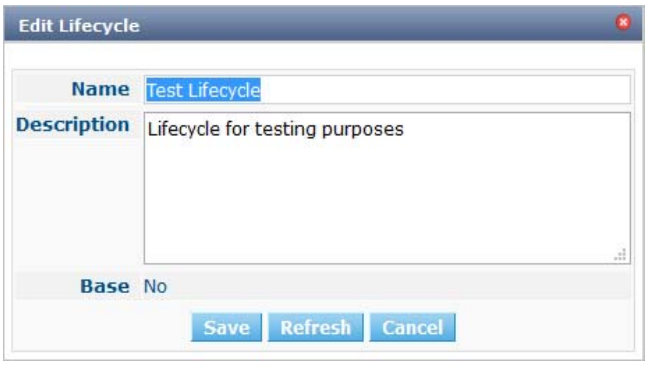
Editing a Lifecycle

1. Switch to the *Lifecycles Overview* screen.
   See The Lifecycles Overview Screen on page 176.

2. Click the *Edit* link in front of the Lifecycle you want to edit.
   The *Edit Lifecycle* screen is displayed:



3. Click the *Edit* button.



3.1. Change the fields as required.
   For a detailed description of the fields, refer to Creating a Lifecycle (page 175).

3.2. Click *Save* to save your changes.

You can also click:

- *Refresh* to retrieve the settings from the database.
- *Back* to return to the previous screen without saving the changes.

Note: On the *Edit Lifecycle* screen you can also edit the Levels, make them optional, activate the Optional Deploys option and add additional Levels using the Create Level links, and modify their order. For more information, refer to the section Levels (page 187).

4. Underneath the *Lifecycle Info* panel, an overview of the defined Levels, as well as the necessary links to create Test and Production Levels are available.

For more information, refer to the sections:

- Defined Levels Panel (page 178)
- Creating Additional Levels (page 180)
- Inserting an Existing Level (page 180)
- Changing the order of the Levels (page 181)
- Making a Level optional or required (page 181)
- Enabling or Disabling Optional Deploys (page 182)

Defined Levels Panel

On the Defined Levels panel, the following information is available for each of the defined Levels:

Field	Meaning
Name	This field contains the user-defined name of the defined Level. It typically refers to the Level's type or function.
Description	This field contains a user-defined description of the Level.
Type	This field indicates the Level Type. The following Level Types are available: • Build • Test • Production
Locked	This field indicates whether or not this Level is locked. If a Level is locked, this field contains the locked icon ( ). No Level Requests can be created for a Level when it is locked. If a Level is not locked, this field remains empty.
Optional	This field indicates whether or not this Level is optional. If a Level is optional, this field contains a blue check mark ( ).
Notification Type (Criteria)	This field indicates the Notification Type associated to the Level (*Mail*, *Netsend* or *No Notification*), followed, between brackets, by the Notification Criterion (*Always*, *Fail*, *Successful* or *Never*).
Requester	This field indicates the name of the Requester User Group. Members of this User Group have the right to create Level Requests for that Level.
Pre-Notify	This field indicates the name of the Pre-Notification User Group. Members of this User Group will be notified when a Level Request is created for this Level, this means before its execution.
Pre-Approve	This field lists the Pre-Approval User Groups that have been set for this Level, along with a sequence number. All Pre-Approvals must be granted before a Level Request for this Level is started. A Pre-Approval can be granted or rejected by any member of the indicated User Group.
Post-Approve	This field lists the Post-Approval User Groups that have been set for this Level, along with a sequence number. Post-Approvals can be granted or rejected after a Level Request for this Level has ended. If all approvals are granted, the status of the Level Request will be set to *Success* or *Warning*, otherwise the Level Request will get the status Rejected. A Post-Approval can be granted or rejected by any member of the indicated User Group.
Post-Notify	This field indicates the name of the Post-Notification User Group, followed, between brackets, by the Post-Notification Criterion. Members of this group will be notified when a Level Request has ended, depending on the Levels Post-Notification Criteria.

Note: The Pre-Notify, Pre-Approve, Post-Approve and Post-Notify fields do not apply for Build Levels.

Creating Additional Levels

Underneath the *Defined Levels* panel, the following links may be available:

- *Create Build Level*

  This option is only available if no Build Level has been defined yet. The first Level defined for a Lifecycle must be a Build Level. Once the Build Level has been created, this option is no longer available, because a Lifecycle must contain exactly one Build Level.

  For a description of the fields, refer to the section Creating a Build Level (page 188).

- *Create Test Level* and *Create Production Level*

  These options become available, once the mandatory Build Level for the Lifecycle has been defined.

  For a description of the fields, refer to the section Creating a Test or Production Level (page 190).

Inserting an Existing Level

Note: Levels can only be inserted one by one.

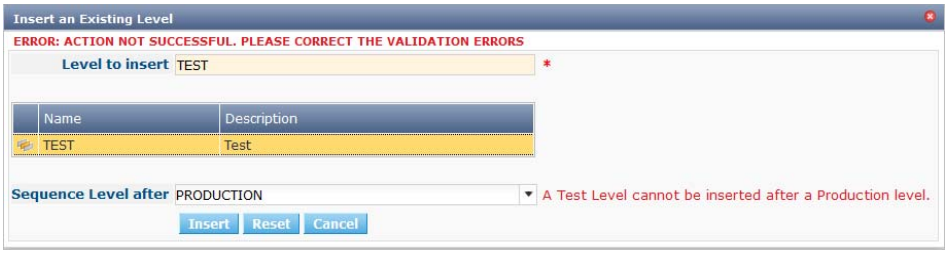
1. Click the *Insert an Existing Level* link on the *Defined Levels* panel.
   One of the following screens is displayed:

2. From the table, select the Level to be inserted.

Note: If no Level has been assigned to the Lifecycle yet, you can only insert a BUILD Level

3. Determine the position in the Level Sequence by selecting the preceding Level from the *Sequence Level after* drop-down list.
   If you do not specify the position, the Level will be added after the last Level of the same type.

If you try to insert a Level at an unauthorized position, a warning will be displayed.



Note: Be careful when inserting Test or Production Levels with Deploy Environments that have already been assigned to other Lifecycles. When Builds are delivered to the Deploy Environment, you risk to accidentally overwrite files in the Target directory of those Deploy Environments.

4. Click the *Insert* button.

   You can also click:

   - *Reset* to clear the fields.
   - *Cancel* to return to the previous screen without saving the changes.

5. The Level will be inserted and displayed on the *Defined Levels* panel.

6. Unlock the Level using the *Audit Project* function.

   For more information on auditing a Project, refer to Auditing Projects (page 247).

Changing the order of the Levels

On the Defined Levels panel, use the *Up* and *Down* links in front of the Level to change the position of the selected Level in the sequence of defined Levels.

Note: These links are not available for Levels that cannot be moved up or down. Test Levels may not be placed after Production Levels.

Making a Level optional or required

Making a Level Optional means that it can be skipped in the Lifecycle.

For example: consider a Lifecycle that defines four Levels: Build, Test, Training and Production. Making the Training Level optional, means that Builds can be delivered from the Test Level straight to Production, without ever needing to be delivered to the Training Level.

1. On the *Defined Levels* panel, click the *Make Level Optional* link in front of the Level to make it optional.

   The following message is displayed:

Note: The Level that was made *Optional*, and all following Levels in the Lifecycle will be locked.

2. Unlock the Level using the *Audit Project* function.
   For more information on auditing a Project, refer to Auditing Projects (page 247).

   Note: The icon in front of the Level has changed to . You can use this *Set Required* link to make the Level required again.

Enabling or Disabling Optional Deploys

If you enable the *Optional Deploys* functionality on a Level, you will be able to skip Deploys to specific Deploy Environments at the moment you create a Level Request (The Action Type may be (Re)Deliver Build or Rollback Build). See also Creating Level Requests (page 45).

This can be useful, for example, if you execute a Level Request with multiple Deploys and one of the (non-blocking) Deploys fails. With this option, it will be possible to execute the Level Request while skipping the failing Deploy and still continue to the next step in the Lifecycle.

Note: In case Deploys are skipped, the status of the Level Request will be set to warning (even if it executed successfully). See also Phase Logs (page 77).

1. Switch to the *Lifecycles Overview* screen.
   See The Lifecycles Overview Screen on page 176.

2. Click the *Edit* link in front of the required Lifecycle.

3. On the *Defined Levels* panel, click the *Enable Optional Deploys* link in front of the Level.

4. Unlock the Level using the *Audit Project* function.
   For more information on auditing a Project, refer to Auditing Projects (page 247).

Note: The icon in front of the Level has changed to . You can use this *Disable Optional Deploys* link to make all Deploys of the Level Request required again.

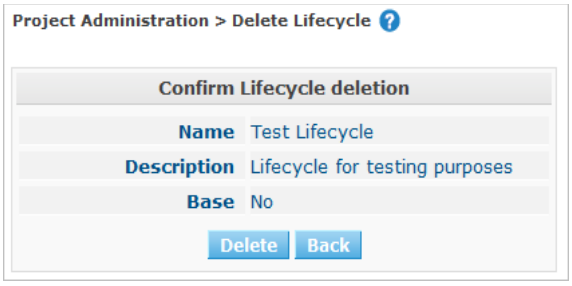
Deleting a Lifecycle

1. Switch to the *Lifecycles Overview* screen.
   See The Lifecycles Overview Screen on page 176.

2. Click the *Delete* link in front of the Lifecycle you want to delete.

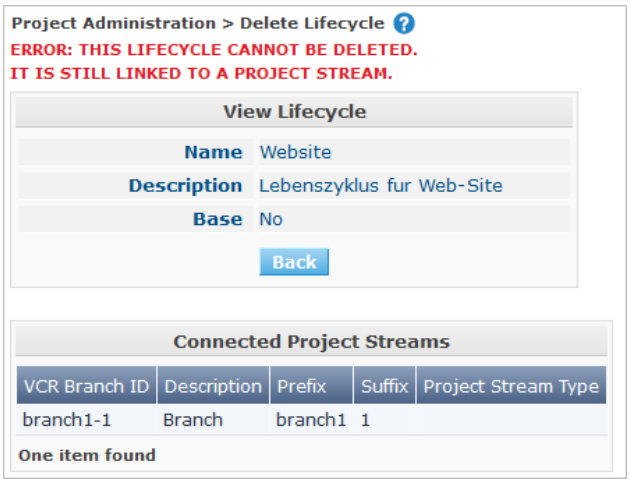
Note: This link is not available for the BASE Lifecycle, because the BASE Lifecycle cannot be deleted.

The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back*, if you want to return to the previous screen without deleting the Lifecycle.

4. If you try to delete a Lifecycle that is connected to a Project Stream, the following screen is displayed:

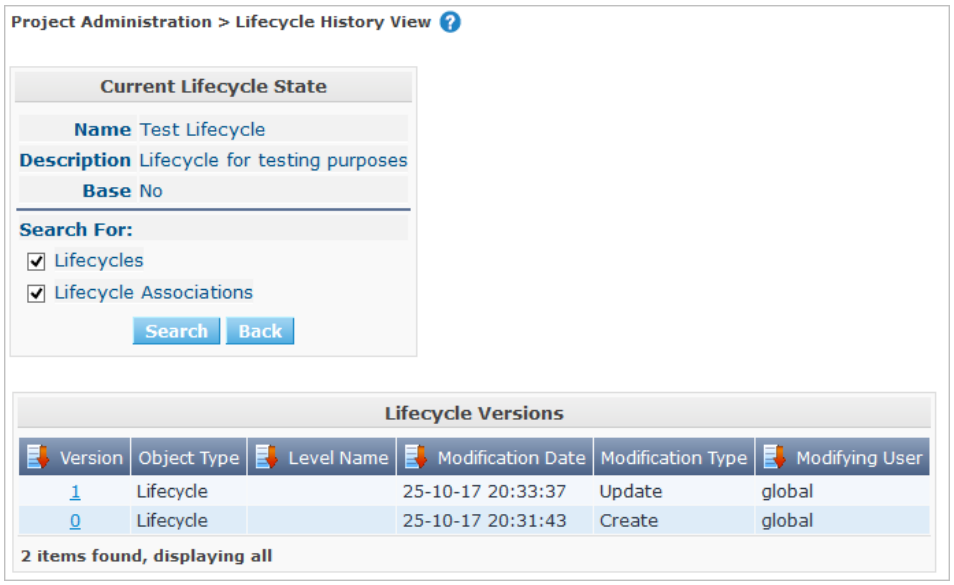


Remove the Lifecycle from the Project Stream before deleting it. See Editing Project Stream Settings on page 167.

Viewing the Lifecycle History

1. Switch to the *Lifecycles Overview* screen.
   See The Lifecycles Overview Screen on page 176.

2. Click the *History* link in front of the Lifecycle to display its *History View*.



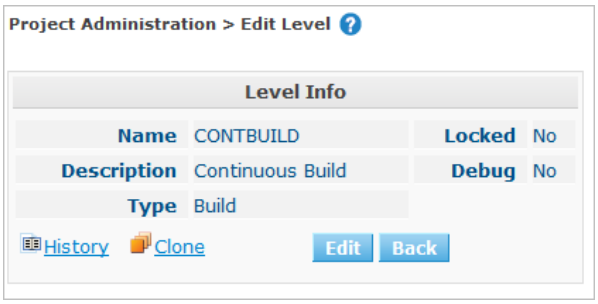
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the previous screen.

Editing a Level

1. Switch to the *Lifecycles Overview* screen.
   See The Lifecycles Overview Screen on page 176.

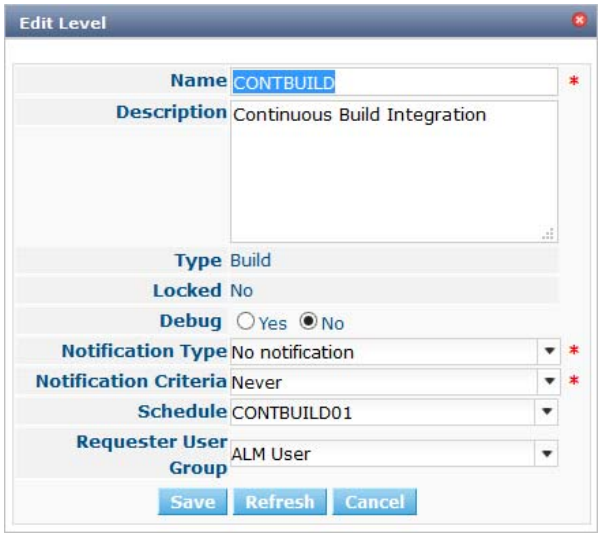
2. Click the *Edit* link in front of the required Lifecycle.

3. On the *Defined Levels* panel, click the *Edit* link for the required Level.
   The *Edit Level* screen is displayed:



4. Click the *Edit* button on the *Level Info* panel and edit the fields as required.

   The following pop-up window will be displayed:



5. Edit the fields on the *Edit Level* panel.

   For a description of the fields, refer to the sections Creating a Build Level (page 188) and Creating a Test or Production Level (page 190).

6. Click *Save* to save your changes.

   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the previous screen without saving your changes.

7. Underneath the *Level Info* panel, an overview of the defined Phases, Environments and, optionally, Pre- and Post-Approvals is displayed, as well as the necessary links to edit Phases, to create Build and Deploy Environments and, optionally, to edit the Deploy Sequence and the Approvals.

   For more information, refer to the sections:

   - Editing a Level Phase (page 197)
   - Creating a Build Environment (page 210)
   - Creating a Deploy Environment (page 225)
   - Level Approvals (page 205)
   - Editing the Deploy Sequence (page 204)

Removing a Level

Note: Removing a Level from the Lifecycle does not remove them from the Project. The Level remains available for insertion in any Lifecycle. To delete a Level, refer to Deleting a Level (page 206).

1. Switch to the *Lifecycles Overview* screen.
   See The Lifecycles Overview Screen on page 176.

2. Click the *Edit* link in front of the required Lifecycle.

3. Click the *Remove* link in front of the Level you want to remove.

4. Click *Yes* to confirm the removal.
   The Level is removed from the Lifecycle, but is not deleted. You can re-insert the Level at any time. To permanently delete a Level, refer to the section Deleting a Level (page 206).

CHAPTER 18
Levels

This option allows defining Levels and their Environments, which can be assigned to a IKAN ALM Project Lifecycle.

A Level is a logical environment, a step in the Lifecycle. It consists of one (or more) Build and/or Deploy Environment(s) which are physical environments.

There are three Level Types:

- Build Level: must have at least one Build Environment.
- Test Level: must have at least one Build or Deploy Environment. A typical Test Level may have one or more Build Environments for Rebuilds and one or more Deploy Environments for Deployment and Testing.
- Production Level: must have at least one Build or Deploy Environment. A typical Production Level has one or more Deploy Environments.

Note: Before you can define Build or Deploy Environments, the required Machines (page 289) and Scripting Tools (page 394) must have been defined. This is the task of the IKAN ALM Global Administrator. Once you have created the Levels, you need to assign them to a Lifecycle in order to use them.

The Levels submenu option lets you start the following procedures:

- Creating Levels
  - Creating a Build Level (page 188)
  - Creating a Test or Production Level (page 190)
- The Levels Overview Screen (page 193)

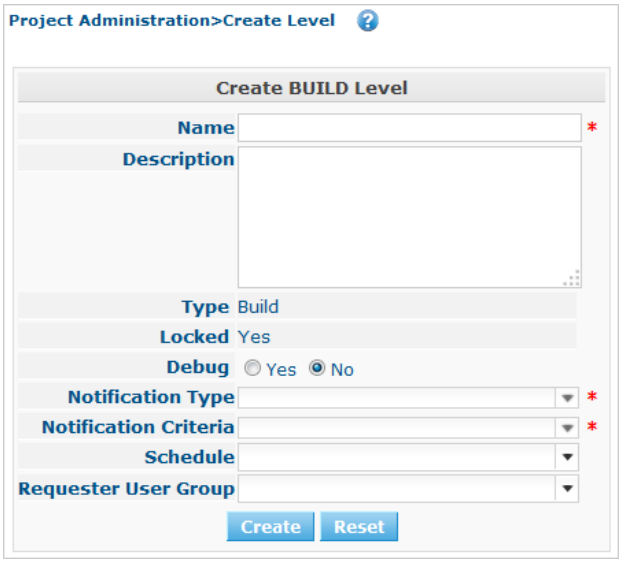
The *Levels Overview* screen lets you verify and change the information concerning Levels. The following actions are possible:

- Modifying Level definitions
  - Editing a Level (page 194)
  - Deleting a Level (page 206)
  - Cloning a Level (page 208)
- Modifying the Level Phases
  - The Level Phases Overview Screen (page 196)
  - Editing a Level Phase (page 197)
  - Inserting a Level Phase (page 201)
- Modifying the Level Approvals
  - Level Approvals (page 205)

- Displaying the Level History View
  - Viewing the Level History (page 208)

18.1. Creating a Build Level

1. In the *Project Administration context*, select *Levels > Create Build Level*.
   The *Create BUILD Level* screen is displayed:



Note: You can also switch to this screen via the *Edit Lifecycle* screen. Select *Lifecycles > Overview*, select the required Lifecycle and click the *Create Build Level* link.

2. Fill out the fields for the new Build Level.
   The following fields are available. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	In this field, enter the name of the new Build Level.
Description	In this field, you may enter a description for the new Build Level.
Type	This read-only field indicates the Level Type, in this case *Build*.
Locked	This read-only field indicates that the new Level is currently still Locked. You need to audit the Project to unlock the Build Level. See Auditing Projects on page 247.
Debug	Indicate whether or not the *Debug* option must be activated for this Level. If *Debug* is activated for a Level, the *Clean-up Work Copy* Phase will not be performed, so that the contents of this directory may be verified by an Administrative User and he can run a Custom Phase manually for testing purposes.

Field	Meaning
Notification Type	From the drop-down list, select the required Notification Type for this Build Level. The following options are available: • Mail • Netsend • No Notification If *Mail* or *Netsend* is chosen, notifications will be created at the end of a Level Request on the Level, depending on the Notification Criteria. The notification is sent to the members of the Project User, Project Admin and Level Requester User Groups.
Notification Criteria	From the drop-down list, select the required Notification Criteria for this Build Level. The following options are available: • *Always* A Notification will be sent when a Level Request for this Level has ended with status *Fail*, *Warning* or *Success*. • *Fail* A notification will be sent when a Level Request for this Level has ended with status *Fail*. • *Successful* A notification will be sent when a Level Request for this Level has ended with status *Warning* or *Success*. • *Never* No notification will be sent when a Level Request for this Level has ended.
Schedule	From the drop-down list, you may select a Schedule for this Build Level. The purpose of this Schedule is to determine the interval for generating automatic Builds. Schedules are defined by the Global Administrator. They verify at the requested time or interval whether the code in the VCR was changed. If this is the case, a Level Request is created automatically. Examples: At global administration level, define: • *Nightly Build:* define a schedule that will verify the code each night and generate a Level Request, if the code was changed. • *Continuous Build:* define a schedule that will verify the code every 10 or 20 minutes and generate a Level Request, if the code was changed. **Note:** This option is not available for Package-based projects.
Requester User Group	From the drop-down list, you may select the required Requester User Group. Members of this User Group have the right to create Level Requests for this Build Level. If a Project User Group has been defined, this User Group will be selected as the default Requester User Group.
Lifecycle	From the drop-down list, select the Lifecycle to which the Level must be assigned. **Note:** If you create a Build Level using the *Create Build Level* link on the *Edit Lifecycle* screen, the name of the Lifecycle will already be filled in.

3. Additional Information

   The *Build Levels Defined in Project* panel displays the different Build Levels defined for the Project. For more information on this panel, refer to Levels Overview (page 192).

4. Click *Create* to confirm the creation of the new Build Level.

   You can also click *Reset* to clear the fields and restore the initial values.

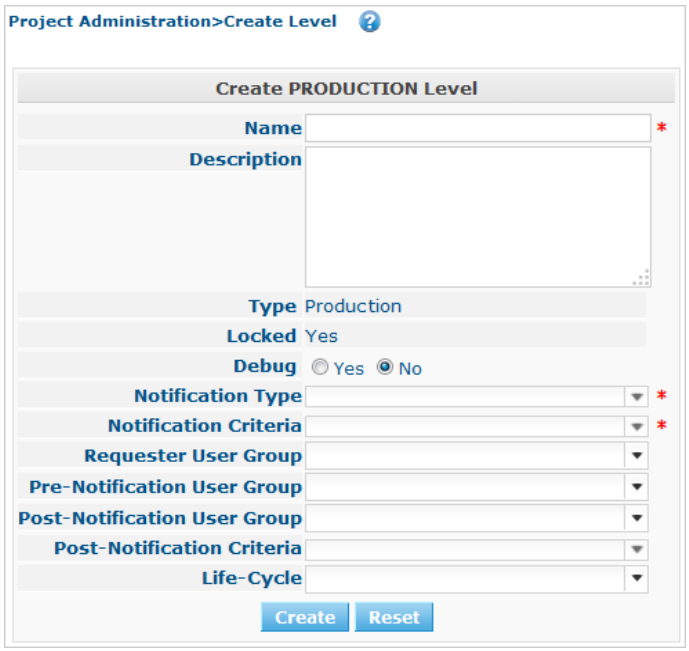
   • *Back* to return to the previous screen without saving the changes.

18.2. Creating a Test or Production Level

Note: The procedure for creating Test and Production Levels is identical.

1. In the *Project Administration context*, select *Levels > Create Test Level* or *Create Production Level*.

   The *Create TEST Level* or *Create PRODUCTION Level* screen is displayed:



Note: You can also switch to this screen via the *Edit Lifecycle* screen. Select *Lifecycles > Overview*, select the required Lifecycle and click the *Create Test Level* or *Create Production Level* link.

2. Fill out the fields for the new Test or Production Level.

   The following fields are available. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	In this field, enter the name of the new Test or Production Level.
Description	In this field, you may enter a description for the new Test or Production Level.
Type	This read-only field indicates the Level Type, in this case *Test* or *Production*.
Locked	This read-only field indicates that the new Level is currently still *Locked*. You need to audit the Project to unlock the Test or Production Level. See Auditing Projects on page 247.
Debug	Indicate whether or not the *Debug* option must be activated for this Level. If *Debug* is activated for a Level, the *Clean-up Work Copy* Phase will not be performed, so that the contents of this directory may be verified by an Administrative User and he can run a Custom Phase manually for testing purposes.

Field	Meaning
Notification Type	From the drop-down list, select the required Notification Type for this Test or Production Level. The following options are available: • netsend • mail • No notification If *Mail* or *Netsend* is chosen, notifications will be created at the end of a Level Request on the Level, depending on the Notification Criteria. The notification is sent to the members of the Project User, Project Admin and Level Requester User Groups.
Notification Criteria	From the drop-down list, select the required Notification Criteria for this Test or Production Level. The following options are available: • *Always* A Notification will be sent when a Level Request for this Level has ended with status *Fail*, *Warning* or *Success*. • *Fail* A notification will be sent when a Level Request for this Level has ended with status *Fail*. • *Successful* A notification will be sent when a Level Request for this Level has ended with status *Warning* or *Success*. • *Never* No notification will be sent when a Level Request for this Level has ended.
Requester User Group	From the drop-down menu, you may select the required Requester User Group. Members of this User Group have the right to create Level Requests for this Test or Production Level. If a Project User Group has been defined, this User Group will be selected as the default Requester User Group.
Pre-Notification User Group	From the drop-down menu, you may select the required Pre-Notification User Group. Members of this group will be notified when a Level Request is created for this Level.
Post-Notification User Group	From the drop-down menu, you may select the required Post-Notification User Group. Members of this group will be notified when a Level Request for this Level has ended, depending on the Post-Notification Criteria.
Post-Notification Criteria	From the drop-down list, select the required Post-Notification Criteria for this Test or Production Level. The following options are available: • Always A Post-notification will be sent if the Level Request ends with status *Success*, *Warning*, *Fail* or in case of a rejected **Post-**Approval. • Fail/Reject A Post-notification will be sent if the Level Request fails or in case of a rejected **Post-**Approval. • Success A Post-notification will be sent if the Level Request ends with status *Success* or *Warning*.

Field	Meaning
Sequence Level After	This field is only available if you create a Test or Production Level using the *Create Test Level* or *Create Production Level* link on the *Edit Lifecycle* screen. From the drop-down menu, select the level that must precede the Test or Production Level you are defining. If you used the Levels > Create Test (or Production) Level option on the Project Administration submenu, the new Test or Production Level will be created after the existing Test or Production Level, if any. You can always modify the Level Sequence using the *Up* or *Down* options on the *Defined Levels* panel on the *Edit Lifecycle* screen. See Changing the order of the Levels on page 181.
Lifecycle	From the drop-down list, select the Lifecycle to which the Level must be assigned. **Note:** If you create a Test or Production Level using the *Create Test Level* or *Create Production Level* link on the *Edit Lifecycle* screen, the name of the Lifecycle will already be filled in.

3. Verify the settings of the other Test or Production Levels defined in the Project.

   The *Test Levels Defined in Project* or *Production Levels Defined in Project* panel displays the different Test or Production Levels defined for the Project.

   For more information on this panel, refer to Levels Overview (page 192).

4. Click *Create* to confirm the creation of the new Test or Production Level.

   You can also click

   - *Reset* to clear the fields and restore the initial values,
   - *Back* to return to the previous screen without saving the changes.

18.3. Levels Overview

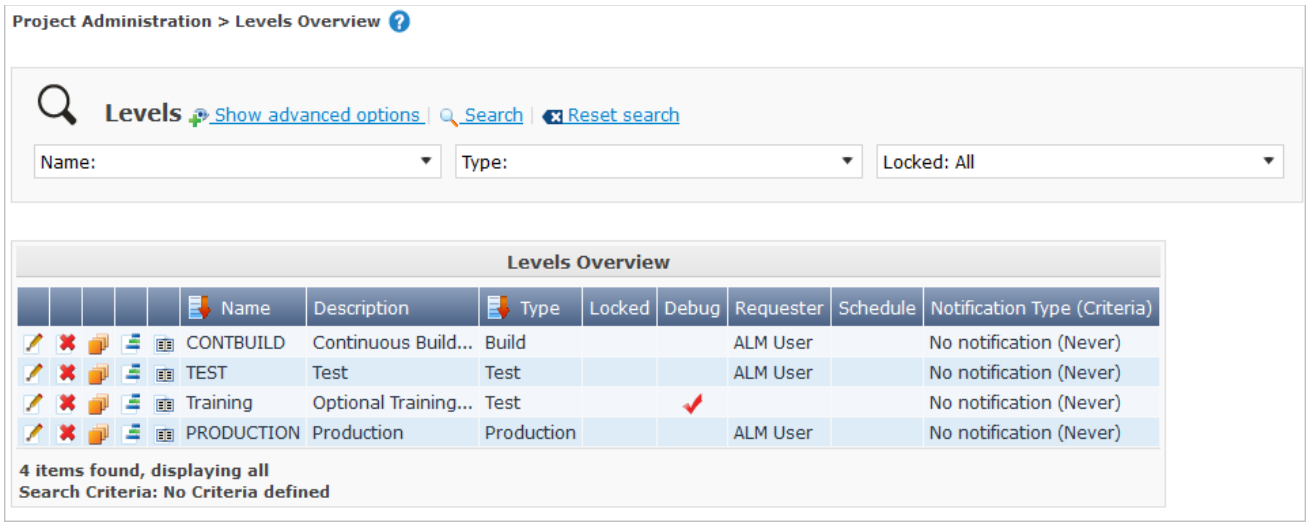
The *Levels Overview* screen displays the information concerning the Build, Test and Production Levels defined for a Project.

The following actions are possible:

- Modifying Level definitions
  - Editing a Level (page 194)
  - Deleting a Level (page 206)
  - Cloning a Level (page 208)
- Modifying the Level Phases
  - The Level Phases Overview Screen (page 196)
  - Editing a Level Phase (page 197)
  - Inserting a Level Phase (page 201)
- Displaying the Level History View
  - Viewing the Level History (page 208)

The Levels Overview Screen

1. In the *Project Administration context*, select *Levels > Overview*.
   The *Levels Overview* screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields,

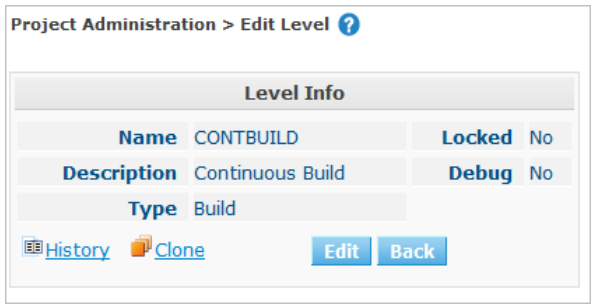
3. On the Levels Overview, verify the Level information fields.
   For a description of the fields, refer to the sections Creating a Build Level (page 188) and Creating a Test or Production Level (page 190).

4. The following links are available:
   - Editing a Level (page 194)
   - Deleting a Level (page 206)
   - Cloning a Level (page 208)
   - Level Phases (page 195)
   - Viewing the Level History (page 208)

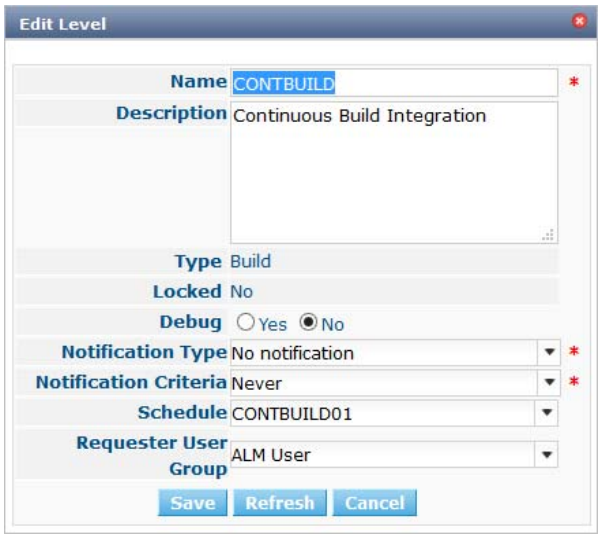
Editing a Level

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. On the *Levels Overview* panel, click the *Edit* link for the required Level.
   The *Edit Level* screen is displayed.



3. Click the *Edit* button on the *Level Info* panel.
   The following pop-up window will be displayed:



   3.1. Edit the fields on the *Edit Level* pop-up window.
   For a description of the fields, refer to the sections Creating a Build Level (page 188) and Creating a Test or Production Level (page 190).

   3.2. Click *Save* to save your changes.
   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the previous screen without saving your changes.

4. Underneath the *Level Info* panel, an overview of the defined Phases, Environments and, optionally, Pre- and Post-Approvals is displayed, as well as the necessary links to edit Phases, to create Build and Deploy Environments and, optionally, to edit the Deploy Sequence and the Approvals.

For more information, refer to the sections:

- Level Phases (page 195)
- Creating a Build Environment (page 203)
- Creating a Deploy Environment (page 204)
- Editing the Deploy Sequence (page 204)
- Level Approvals (page 205)

Level Phases

When creating a Level (Build, Test or Production), IKAN ALM will automatically link the default flow of core Level Phases to it.

The core Phases are:

- Retrieve Code
- Build
- Tag Code
- Deploy
- Cleanup Work Copy

Levels for Projects that are linked to an Issue Tracking System will have an extra Issue Tracking Phase. On a Build Level, the search for handled Issue Numbers in the VCR commit comments will happen during this Phase.

The Build and Deploy Phases will communicate with the Agent running the Build(s) or Deploy(s) in order to start, follow up and set the end status of Build and Deploy actions.

Note: The behavior of the Phase is dependent of the Level Type to which it is connected. For example; both Test and Production Levels will have a *Tag Code* Phase connected by default, but nothing will happen during this Phase as tagging will only be done on a Build Level type. This behavior may change in future releases. At present you can manually remove the *Tag Code* Phase from Test and Production Levels.

When executing a Level Request for this Level, a log will be created for each of the Phases. See Level Request Detail on page 65.

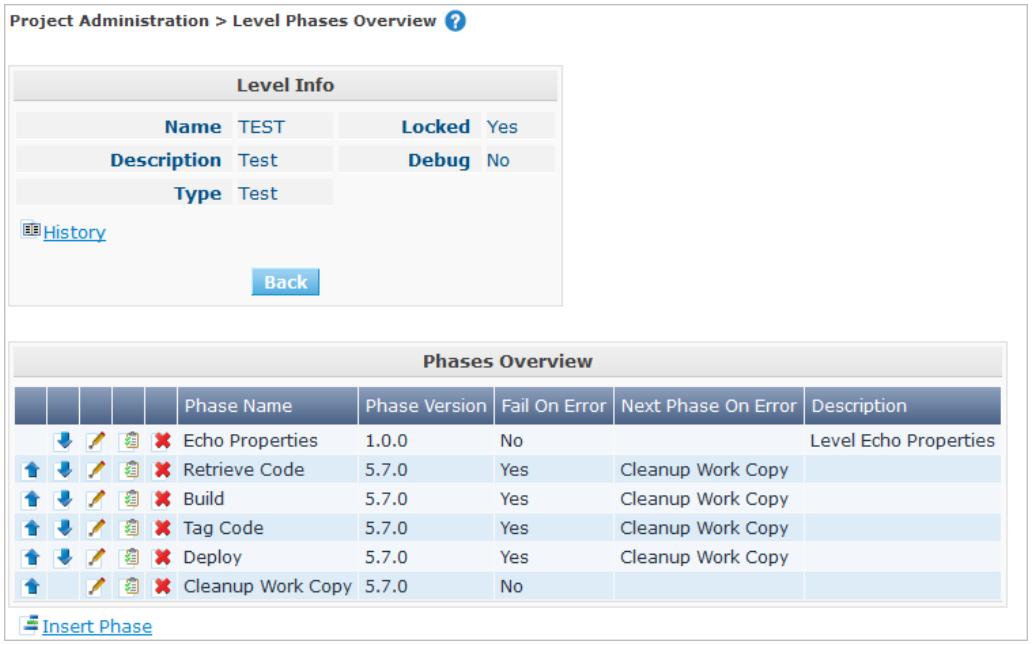
Refer to the following procedures for more information:

- The Level Phases Overview Screen (page 196)
- Changing the Order of the Level Phases (page 197)
- Editing a Level Phase (page 197)
- Viewing the Level Phase Parameters (page 198)
- Deleting a Level Phase (page 201)
- Inserting a Level Phase (page 201)

The Level Phases Overview Screen

1. In the *Project Administration context*, select *Levels* > *Overview*.

2. Click the *Edit Phases* link on the *Levels Overview* panel.
   The *Level Phases Overview* screen is displayed:



3. Use the links on the *Phases Overview* panel to edit a Phase.
   The following links are available:

   - The *Up* and *Down* links to change the order of the Phases. See Changing the Order of the Level Phases on page 197.
   - The *Edit* link to edit the Phase's settings. See Editing a Level Phase on page 197.
   - The *View Parameters* link to manage the mandatory and optional Phase Parameters. See Viewing the Level Phase Parameters on page 198.
   - The *Delete* link to delete a Phase. See Deleting a Level Phase on page 201.

Note: Changing the Phases can have undesirable consequences on the Lifecycle. For more information, refer to the document *HOW TO Using and Developing a Phase in IKAN ALM*.

4. Insert a Phase, if required.
   Select the *Insert Phase* link underneath the *Phases Overview* panel. See Inserting a Level Phase on page 201.

5. When done, click *Back* to return to the *Levels Overview* screen.

Changing the Order of the Level Phases

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. Click the *Edit Phases* link on the *Levels Overview* panel.

3. Use the *Up* and *Down* links in front of a Level Phase to change its position in the sequence.

4. Click *Back* to return to the *Levels Overview* screen.

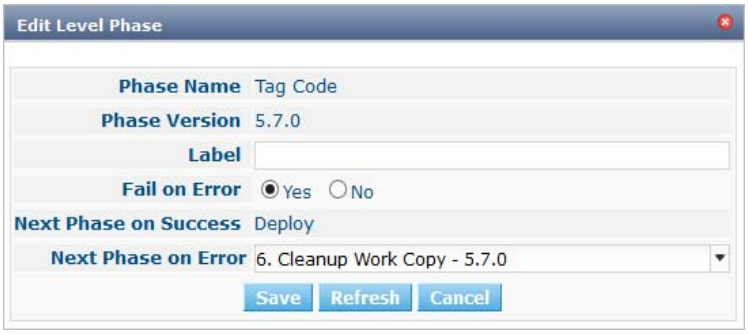
> Avoid changing a Phase's position in such a way that its *Next Phase on Error* is in an earlier position in the workflow: this could result in an infinite loop.

Editing a Level Phase

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. Click the *Edit Phases* link on the *Levels Overview* panel.

3. Click the *Edit* link in front of the Phase you want to edit.
   The *Edit Level Phase* screen is displayed.



4. Edit the fields on the *Edit Level Phase* window.
   For a description of the fields, refer to the section Inserting a Level Phase (page 201).

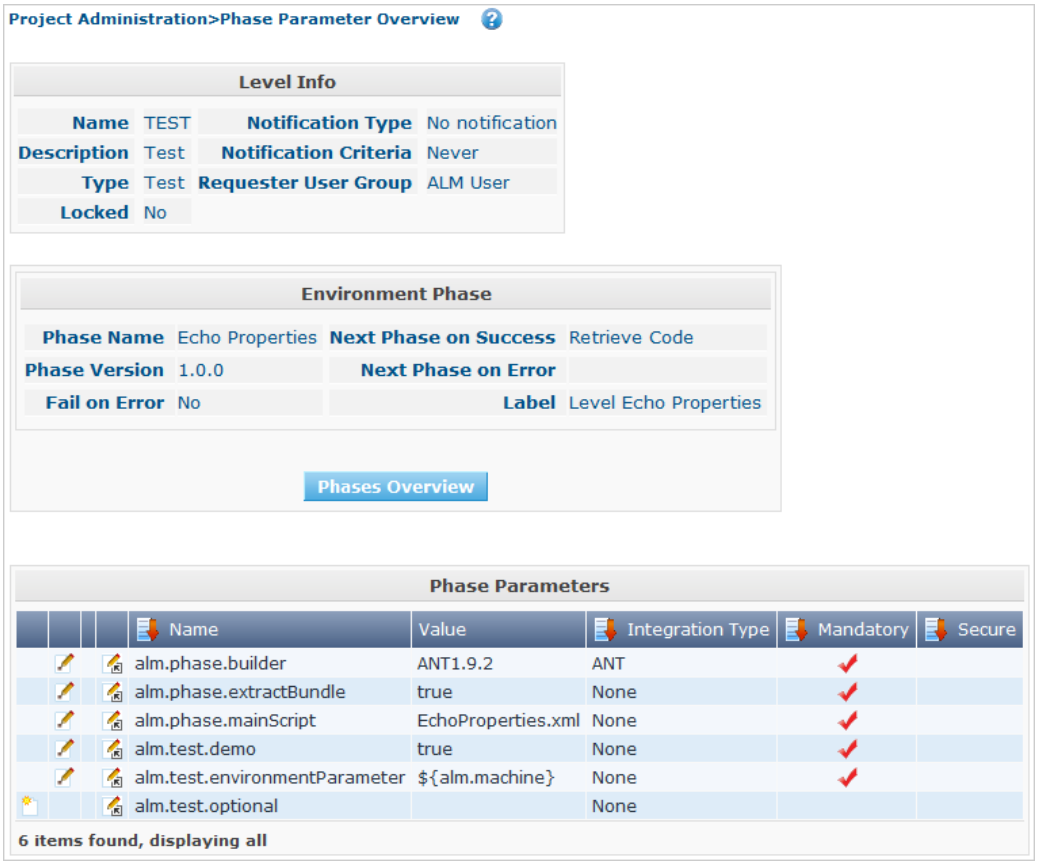
5. Click *Save* to save your changes.
   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the previous screen without saving your changes.

Viewing the Level Phase Parameters

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. Click the *Edit Phases* link on the *Levels Overview* panel.

3. Click the *View Parameters* links in front of a Level Phase you want to manage the Phase Parameters for.
   The *Phase Parameter Overview* screen is displayed.



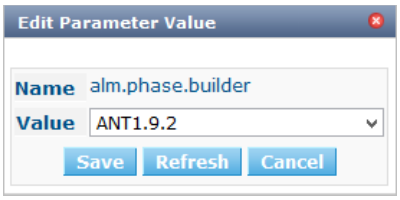
4. Verify the Level Phase Parameters.
   The *Phase Parameters* panel displays all the defined Parameters of the Level Phase and allows you to create non-mandatory Phase Parameters.
   The following fields are available:

Field	Meaning
Name	The name of the Parameter. This field may not be changed since it is defined in Global Administration.
Value	The Value of the Parameter. Initially, when the Phase is inserted, the value will be copied from the Default Value in Global Administration (if provided). This field may be changed by Editing the Phase Parameter.

Field	Meaning
Integration Type	This field indicates whether the value of the Parameter is a simple text value, or whether it represents a link (an integration) to an IKAN ALM Global Administration object type. The possible values are: • None: the value is simple text • Transporter: link to a Transporter • VCR: link to a Version Control Repository • ITS: link to an Issue Tracking System • Scripting Tool: link to a Scripting Tool • ANT: link to an Ant Scripting Tool • GRADLE: link to a Gradle Scripting Tool • NANT: link to a NAnt Scripting Tool • MAVEN2: link to a Maven2 Scripting Tool This field may not be changed since it is defined in Global Administration.
Mandatory	This field indicates whether the Parameter has been created automatically when the Phase is inserted in the Level. This is the case for Mandatory Parameters. Non-mandatory Parameters must be created after the Phase has been inserted in the Level, using the *Create Parameter* link. This field may not be changed since it is defined in Global Administration.
Secure	This field indicates whether the Parameter is secured or not. This field may not be changed since it is defined in Global Administration.

5. Click the *Edit Parameter* link next to a Phase Parameter.

The following pop-up window will be displayed.



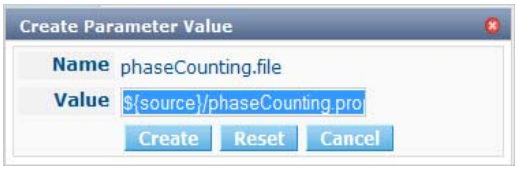
Set the value of the Level Phase Parameter and click *Save* to save the value.

You can also click:

- *Refresh* to retrieve the settings from the database.
- *Cancel* to return to the *Phase Parameter Overview* screen without saving a value.

6. If you want to create a non-mandatory Phase Parameter, click the *Create Parameter* link next to a Phase Parameter.

The following pop-up window will be displayed.



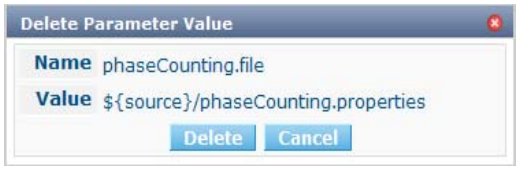
If a default Parameter value has been set in Global Administration, that value will be suggested.

Set the value of the Level Phase Parameter and click *Create*.

- *Reset* to retrieve the settings from the database.
- *Cancel* to return to the *Phase Parameter Overview* screen without saving a value.

7. If you want to delete a non-mandatory Phase Parameter, click the *Delete Parameter* link next to a Phase Parameter.

   The following pop-up window will be displayed.

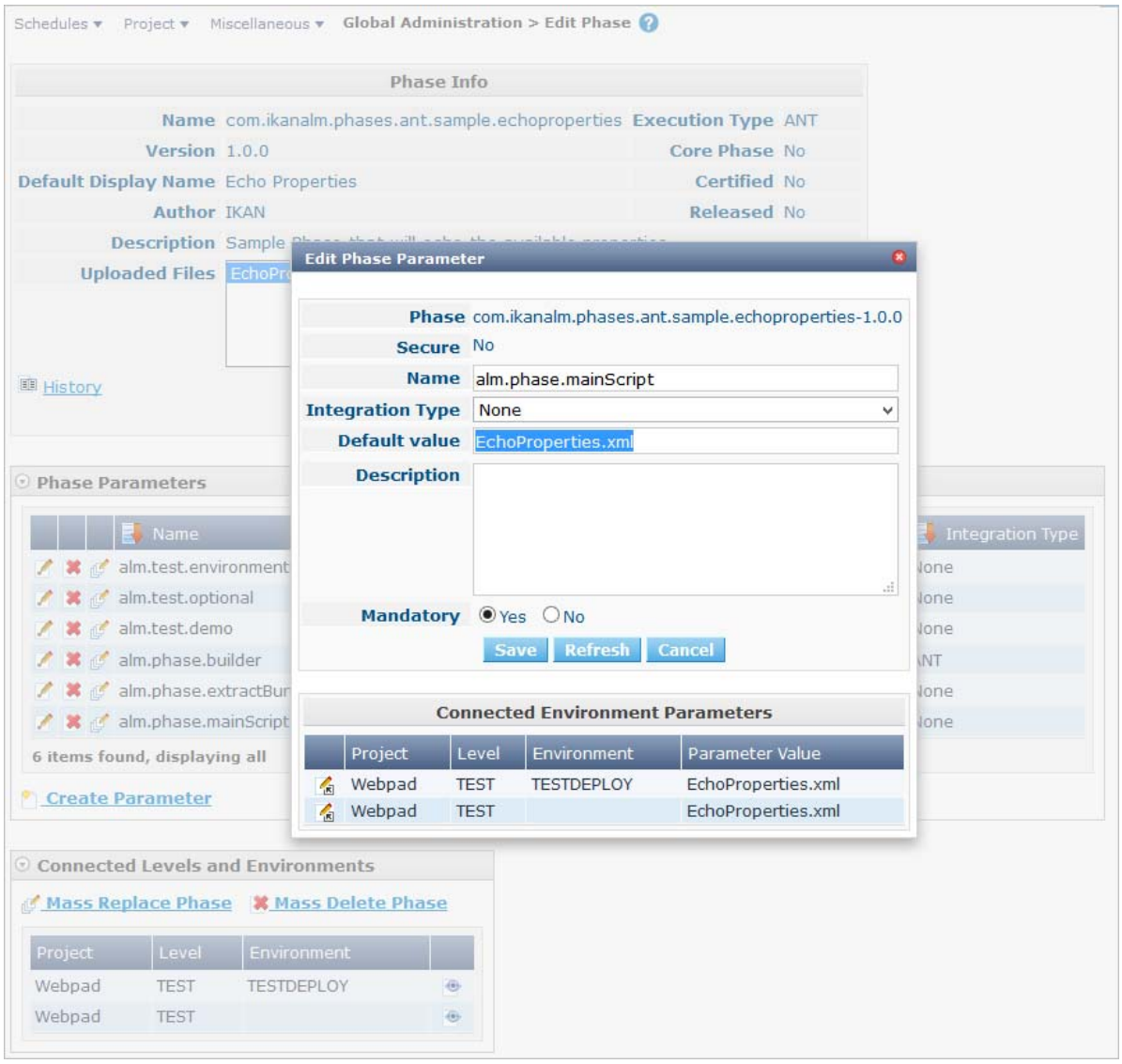


   Click *Delete* to confirm the deletion of the mandatory Level Phase Parameter.

   You can also click *Cancel* to close the pop-up window without deleting the Parameter.

8. Click the *Edit Global Phase Parameter* link next to a Phase Parameter.

   The User will be redirected to the *Edit Phase* screen (in the Global Administration context) and the *Edit Phase Parameter* pop-up window is opened.



   You can edit the Global Phase Parameter as described in the section Editing Phase Parameters (page 437).

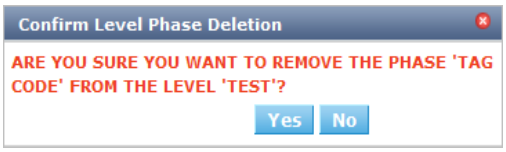
   To go back to the Phase Parameter in the Project Administration context, click the appropriate *Edit Environment Phase Parameter* link in the *Connected Environment Parameters* panel.

Deleting a Level Phase

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. Click the *Edit Phases* link on the *Levels Overview* panel.

3. Click the *Delete* link in front of the Level Phase you want to delete.
   The *Delete Level Phase* screen is displayed.



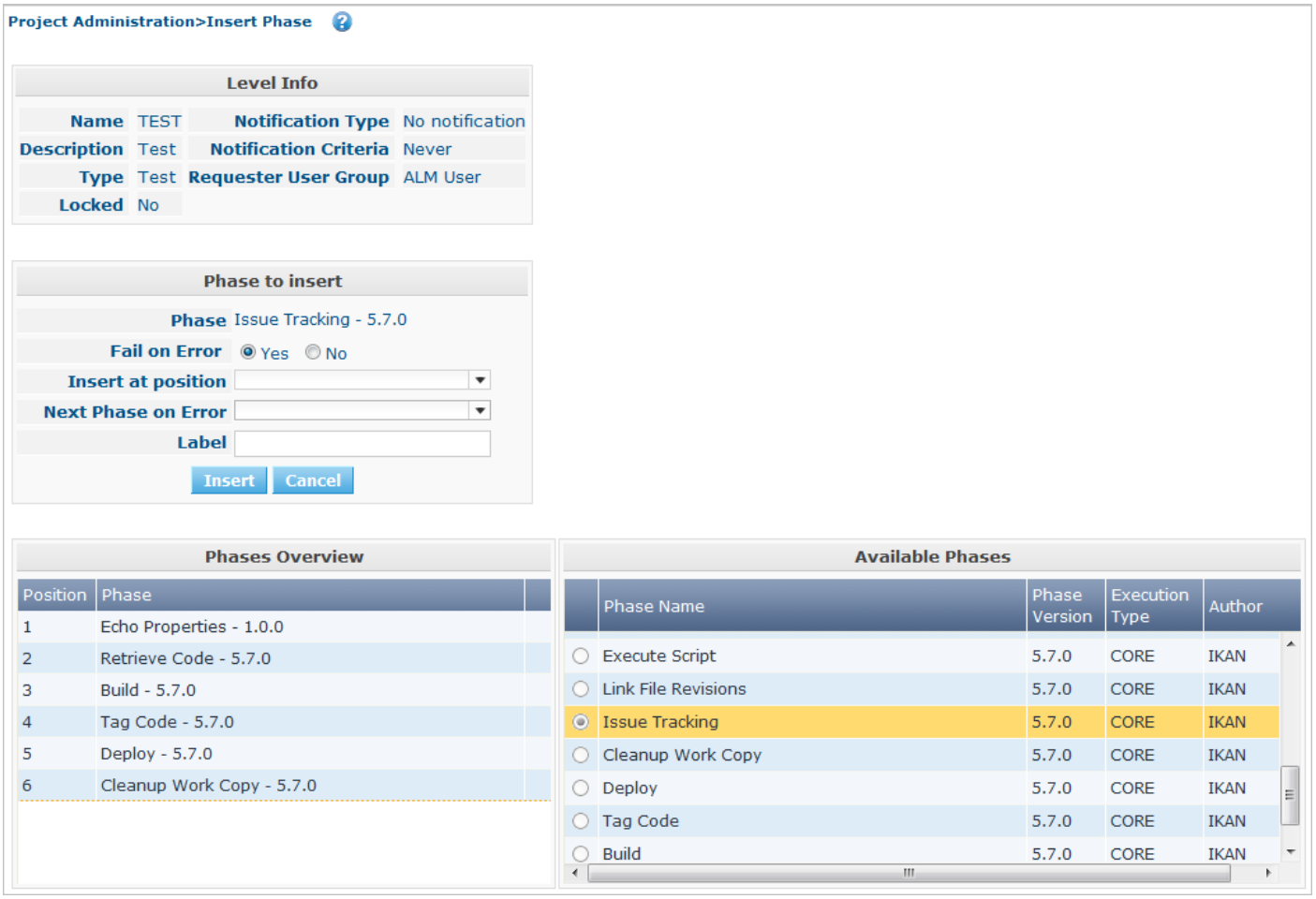
4. Confirm the deletion by clicking the *Yes* button.
   You can also click *No* to return to the previous screen without deleting the Level Phase.

Inserting a Level Phase

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. Click the *Edit Phases* link on the *Levels Overview* panel.

3. On the *Phases Overview*, click the *Insert Phase* link.
   The *Insert Phase* screen is displayed.



4. Select a Phase to insert from the *Available Phases* panel.

5. Fill out the fields for the new Phase.
   The following fields are available:

Field	Meaning
Phase	From the *Available Phases* panel, select the Level Phase to add.
Fail on Error	In this field, indicate whether the Level Request is considered failed when this Phase goes in Error.
Insert at Position	This field indicates at which position the Phase will be inserted into the Level workflow. The Phase Position is also indicated on the *Phases Overview* panel.
Next Phase On Error	This field indicates the next Phase to execute in case this Phase goes in Error.
Label	In this optional field you can add a Label for the Phase to be inserted. In case you use the same Phase several times, adding a label is useful to provide additional information concerning the usage of the Phase.

6. Click *Insert* to confirm the creation of the new Phase.
   You can also click *Cancel* to return to the previous screen without saving the changes.

Build and Deploy Environment Settings

On the *Edit Level* screen, you will also find the information concerning the Build and Deploy Environments defined for the different Levels.

Links for creating Build and Deploy Environments as well as for Changing the Deploy Sequence are available on the Environments Panel.

Note: You can also use the Build Environments and Deploy Environments submenu options for creating Environments. The Deploy sequence, however, can only be modified on the *Levels Overview* screen.

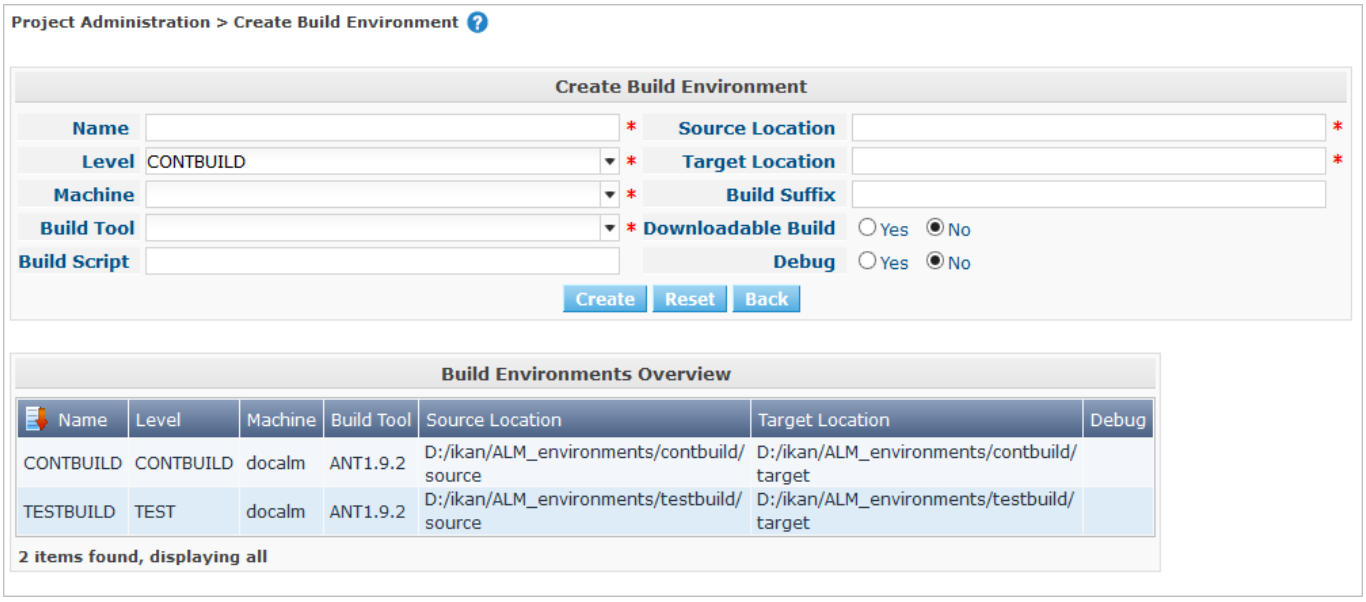
Refer to the following procedures for more information:

- Creating a Build Environment (page 203)
- Creating a Deploy Environment (page 204)
- Editing the Deploy Sequence (page 204)

Creating a Build Environment

1. Switch to the *Edit Level* screen.
   See Editing a Level on page 194.

2. Underneath the *Environments* panel, click *Create Build Environment*.
   The following screen is displayed:

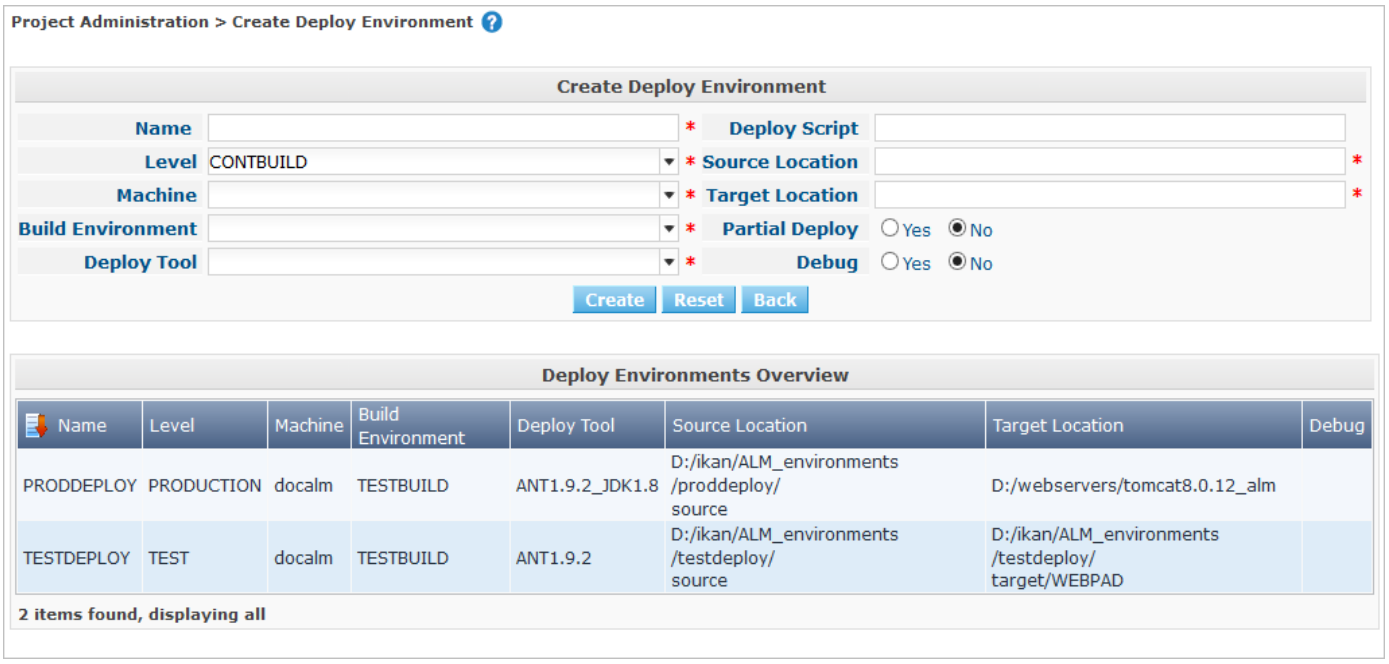


For a complete description of the *Create Build Environment* procedure, refer to Creating a Build Environment (page 210).

Creating a Deploy Environment

1. Switch to the *Edit Level* screen.
   See Editing a Level on page 194.

2. On the *Environments* panel, click *Create Deploy Environment*.
   The following screen is displayed:



For a complete description of the *Create Deploy Environment* procedure, refer to Creating a Deploy Environment (page 225).

Editing the Deploy Sequence

When a Deploy Environment is created, it will automatically get Sequence Number 0. When there are multiple Deploy Environments linked to a Level, they will be treated in parallel, since they all have 0 as Sequence Number. In order to handle them sequentially, you must change the Sequence Number. When executing a Level Request with multiple Deploys, first the Deploys with sequence number 0 will be executed in parallel. When all of them have executed successfully, the Deploys with sequence number 1 will be executed in parallel, and so on.

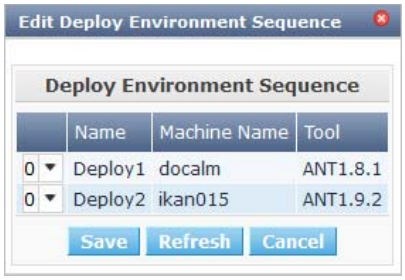
When creating a Level Request, this Sequence Number will also be displayed in front of the Deploy Environment name.

This functionality is useful in case a Deploy Environment depends on another and you do not want to create multiple Levels. For example, one Deploy Environment first updates an application database, and a second Environment will deploy the application code (if the database deploy is successful).

1. Underneath the Environments panel on the *Edit Level* screen, click *Edit Deploy Environment Sequence*.
   See Editing a Level on page 194.

Note: This link is only available if there is more than one Deploy Environment linked to the Level.

The following screen is displayed:



2. Change the *Sequence Number*.
   Use the drop-down menu at the left of the Deploy Environment to change the Sequence Number.

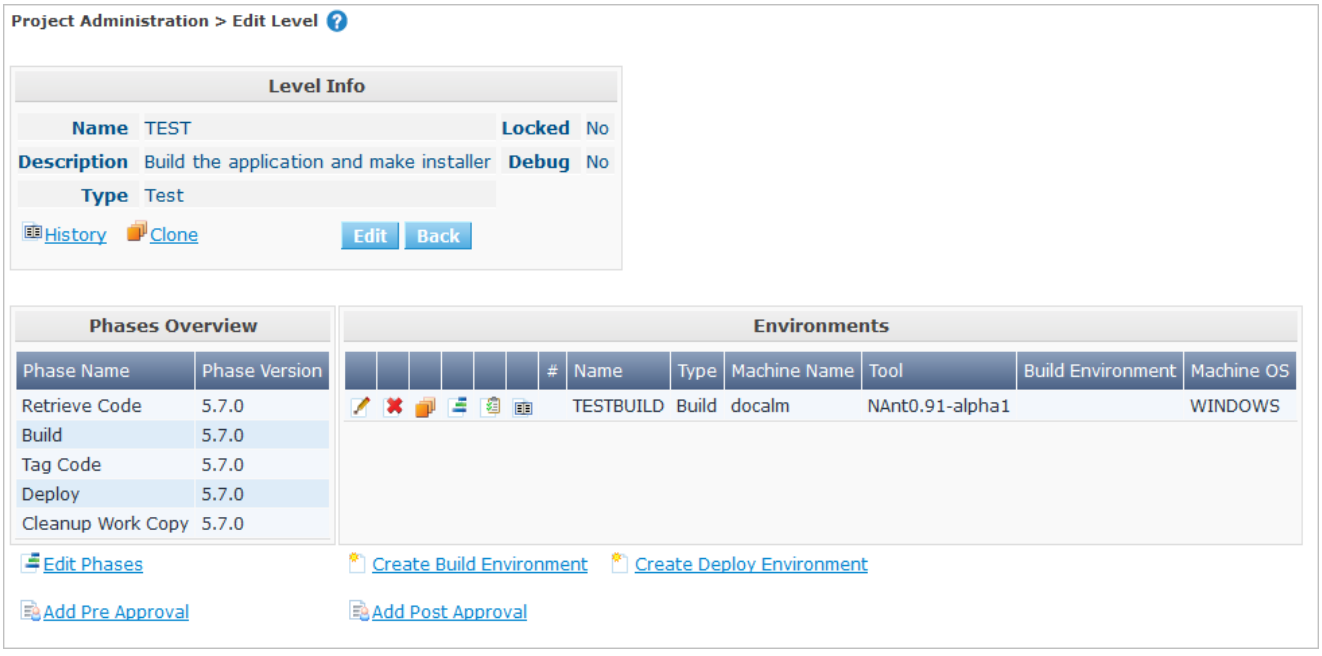
3. Click Save to save the new sequence order.
   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the previous screen without saving your changes.

Level Approvals

For Test and Production Levels, you can specify the sequence of Pre- and Post-Approvals. For more information on Approvals, refer to Approvals (page 21).

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. On the *Levels Overview* panel, click the *Edit* link for the required Level.
   The *Edit Level* screen is displayed.



Note: Approvals can only be defined for Test and Production Levels.

3. Click the *Add Pre-Approval* or *Add Post-Approval* link at the bottom of the *Edit Level* screen.

   The *Add Approval* window is displayed:

4. Select the required *User Group* from the drop-down list.

   Pre-Approvals need to be given before the Level Request is started, as they allow to avoid the execution of a Level Request. Post-Approvals need to be given after the Level Request is executed, as they allow to avoid the delivering of a Build to the next Level in the Lifecycle.

5. Define where the new approving User Group should be inserted.

   Select the User Group after which the new User Group should be inserted from the *Insert After* drop-down list.

> **Note:** If you do not select an existing Approval User Group from the *Insert after* drop-down list, the new Approval User Group will be inserted at the top of the sequence.

6. Click the *Insert* button, to confirm your changes.

   The Approval is added to the Sequence on the *Edit Level* screen.

7. If necessary, you can use the *Up* and *Down* buttons to modify the order of the different User Groups, or use the *Remove* button to remove a User Group from the Sequence list.

8. Repeat this procedure to define all required Pre- and Post-Approvals.

Deleting a Level

Deleting a Level will also delete the connected Build and Deploy Environments, as well as the historical information concerning Level Requests, Builds and Deploys.

> **Note:** If the Level is still linked to a Lifecycle you cannot delete the Level.

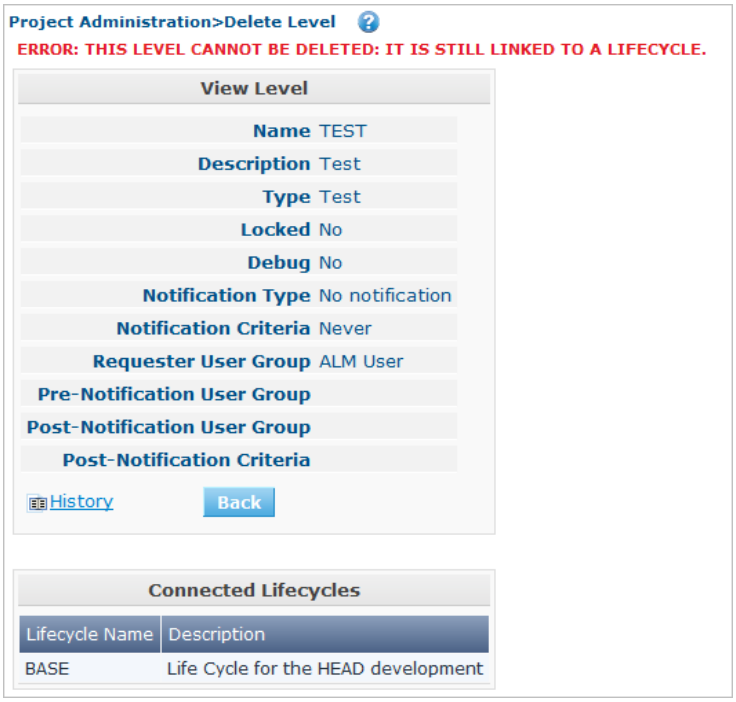
1. Switch to the *Levels Overview* screen.

   See The Levels Overview Screen on page 193.

2. Click the *Delete* link on the *Levels Overview* panel.
   The following screen is displayed:

3. Click *Delete* to confirm the deletion or *Back* to return to the previous screen without deleting the Level.

4. If you try deleting a Level that is assigned to a Lifecycle, the following screen is displayed:

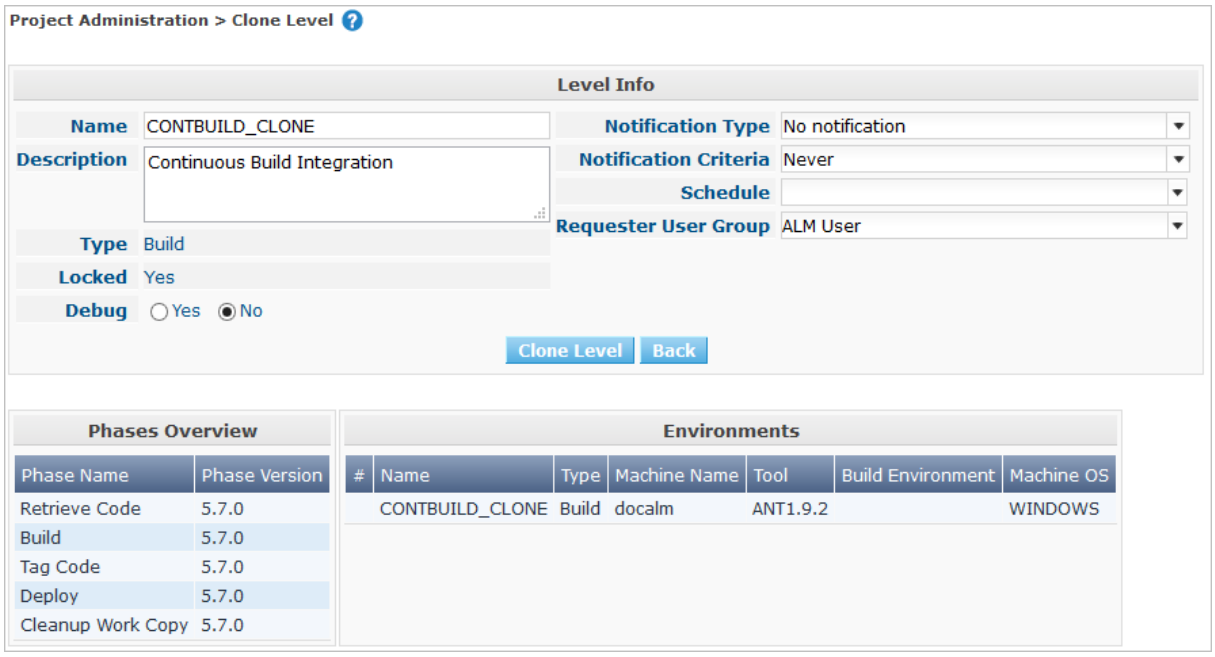


Remove the Level from the Lifecycle before deleting it. See Removing a Level on page 186.

Cloning a Level

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. Click the *Clone* link in front of the Level you want to clone.
   The *Clone Level* screen is displayed.



3. Edit the fields on the *Level Info* panel.
   For a description of the fields, refer to the sections Creating a Build Level (page 188) or Creating a Test or Production Level (page 190).

4. Click *Clone Level* to confirm the action.
   You can also click *Back* to return to the *Levels Overview* screen.
   The Environments linked to the Level will be automatically cloned together with the Level. The environment names will be set automatically by adding a "CLONE" suffix. You might want to change this after the Level has been cloned. You also might want to change the Source and/or Target Locations, especially the Target Locations of cloned Deploy Environments.

Viewing the Level History

1. Switch to the *Levels Overview* screen.
   See The Levels Overview Screen on page 193.

2. Click the *History* link on the *Levels Overview* panel.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the previous screen.

CHAPTER 19
Build Environments

Build Environments are locations where code retrieved from the VCR is built. The environment can be specific for an operating system and it is possible to use general and specific Build Parameters with each Build Environment.

Build Environments must be assigned to a Level in order to be active.

Typical uses of Build Environments:

- When assigned to a Build Level, the Build Environment allows the compilation of source code in the VCR.
- When assigned to a Test Level, the Build Environment allows rebuilds in order to ensure that the source code can be compiled on other platforms, with other Build Parameters, etc.

The Build Environments submenu option lets you start the following procedures:

- Creating a Build Environment (page 210)
- The Build Environments Overview Screen (page 212)

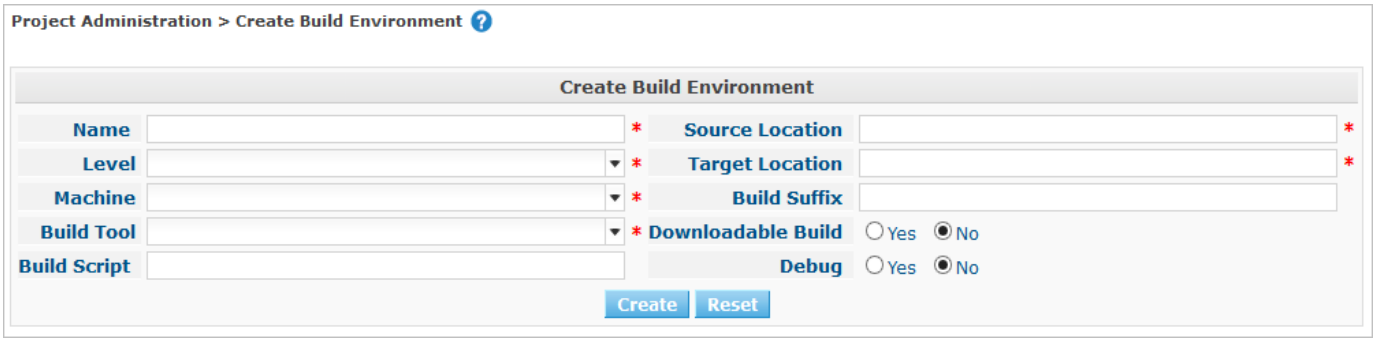
  The *Build Environments Overview* screen lets you verify and change the information concerning Build Environments and related Phases. The following actions are possible:

  - Editing a Build Environment (page 213)
  - Deleting a Build Environment (page 214)
  - Cloning a Build Environment (page 215)
  - Editing a Build Environment Phase (page 219)
  - Viewing the Build Environment Parameters (page 223)
  - Viewing the Build Environment History (page 224)
- Build Parameters

  The *Parameters Overview* screen lets you create, edit, delete and copy Build Parameters and allows to check their history. For more information, refer to the chapter Environment Parameters (page 240).

19.1. Creating a Build Environment

1. In the *Project Administration context*, select *Build Environments > Create*.
   The following screen is displayed:



2. Fill out the fields for the new Build Environment.
   The following fields are available. The fields marked with a red asterisk are mandatory:

Field	Meaning
Name	In this field, enter the name for the Build Environment.
Level	From the drop-down list, select the Level to which you want to assign this Build Environment.
Machine	From the drop-down list, select the Machine this Build Environment must be related to. **Note:** The Source and Target Locations must exist or be creatable on the selected Machine.
Build Tool	From the drop-down list, select the Build Tool for this Build Environment. Paths defined in the selected Build Tool must exist on the linked Machine.
Build Script	In this field, you may enter the relative path to the Build Script you want to select specifically for this Build Environment. If you leave this field empty, the default Build Script, as defined on the Project Settings screen will be used.
Source Location	This field contains the path to the physical location where the source code, the dependent source code or the dependent build result will be placed IKAN ALM adds the Build OID and the Project Name: *SourceLocation/BuildOID/ProjectName_VCR* This location is passed to the Build Script as a parameter named *source*.

Field	Meaning
Target Location	For a Build Environment, this field contains the path to the physical location where IKAN ALM expects the result of the executed Build Script to be found. IKAN ALM adds Build OID: *TargetLocation/BuildOID* This location is passed to the Build Script as a parameter named *target*. The Deploy Script is added to this result, delivering version control of the Deploy Script and its related Build. The result is compressed and stored into the Build Archive. This file has a name in the following format: *webpad_H_1-0_b12_CONTBUILD_win.zip* where: • *webpad*: Project name • *H*: Stream Type indication: *H* for Head, *B* for Branch • *1-0*: Build Prefix • *b12*: Build Number • *CONTBUILD*: Build Environment name • *win*: Build suffix • *.zip*: extension referring to the compression method, *zip* for Windows platforms, *tar.gz* for Unix and Linux platforms.
Build Suffix	In this field, enter the Build Suffix for the Build Environment, if required.
Downloadable Build	Indicate whether you want the compressed Build Result created on this Build Level to be downloadable. If the option is activated, the compressed Build Results appear as a hyperlink on the *Results* tab page of the *Level Request Detail* screen. See Results on page 89.
Debug	Indicate whether or not the Debug option must be activated for this Build Environment. If Debug is activated for a Build Environment, the Source and Result Clean-up actions (actions 15 and 16 of the Build and Deploy Level Requests Overview (page 35)) will not be performed, so that the user may use the available source to run the build script manually for testing purposes.

3. Verify the information on the *Build Environments Overview* panel.

Note: This Overview lists **all** Build Environments defined for **all** Levels belonging to the selected Project.

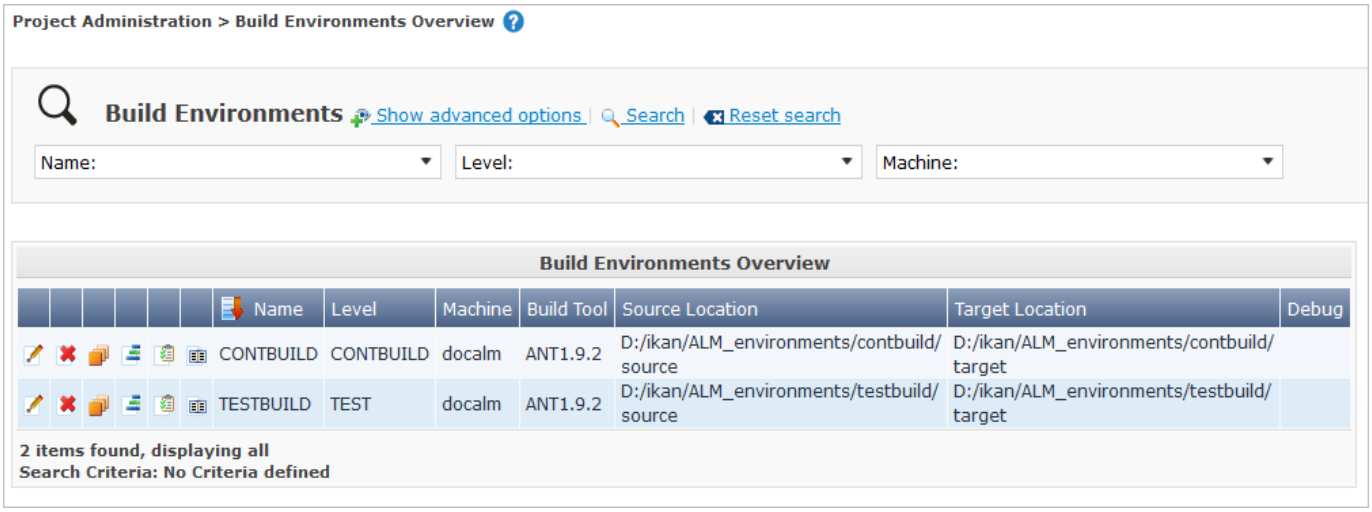
For more information on the available links, refer to The Build Environments Overview Screen (page 212).

4. Click *Create* to confirm the creation of the new Build Environment.

You can also click the *Reset* button to clear the fields and restore the initial values.

19.2. The Build Environments Overview Screen

1. In the *Project Administration context*, select *Build Environments > Overview*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields,

3. Verify the Information available on the *Build Environments Overview*.
   For a detailed description of the fields, see Creating a Build Environment (page 210).

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

4. Depending on your access rights, the following links may be available on the *Build Environments Overview* panel:

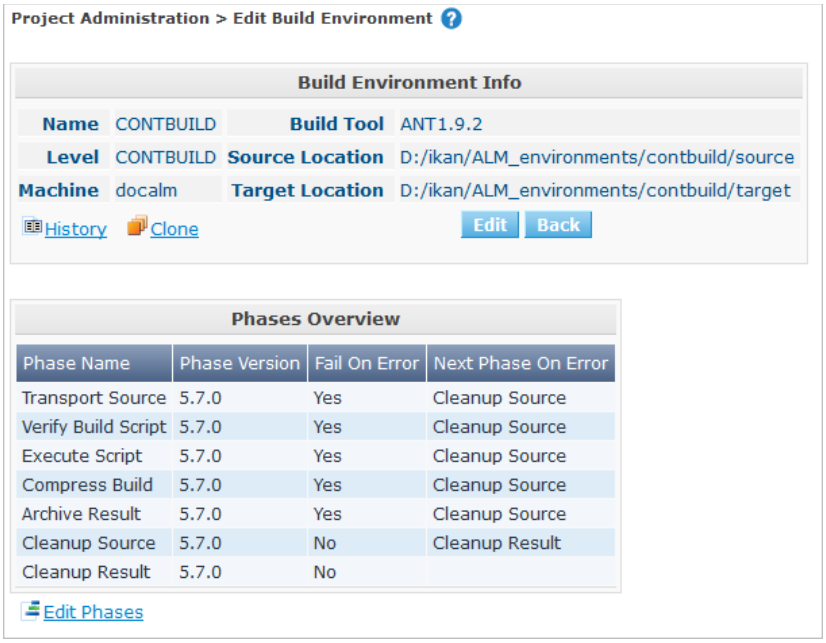
	Edit This option allows editing a the Build Environment settings. See Editing a Build Environment on page 213.
	Delete This option allows deleting a Build Environment. See Deleting a Build Environment on page 214.
	Clone This option allows cloning a Build Environment. See Cloning a Build Environment on page 215.
	Edit Phases This option allows editing the Build Environment Phases. See Editing a Build Environment Phase on page 219.

	View Parameters This option allows viewing the Build Environment Parameters. See Viewing the Build Environment Parameters on page 223.
	History This option allows viewing the Build Environment History. See Viewing the Build Environment History on page 224.

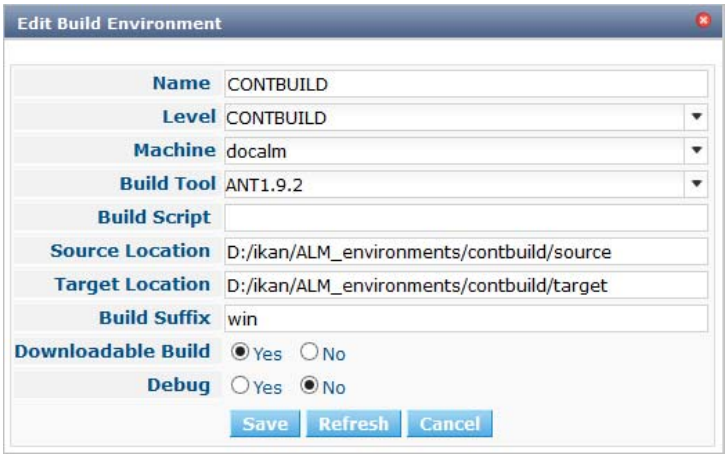
Editing a Build Environment

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. Click *Edit* to change the selected Build Environment definition.
   The following screen is displayed:



3. Click the *Edit* button on the *Build Environment Info* panel.
   The *Edit Build Environment* window is displayed:



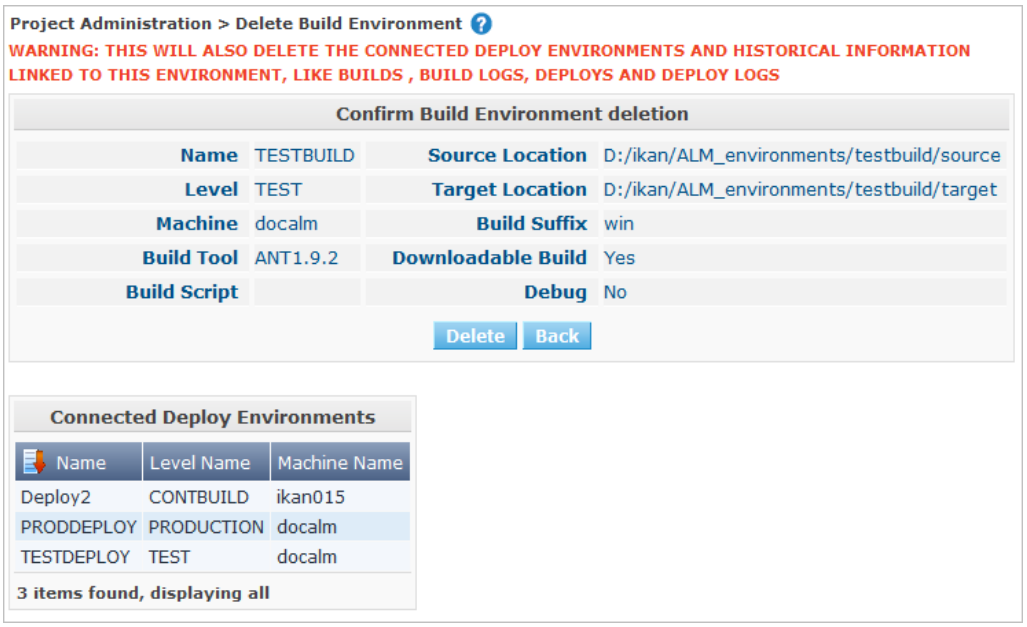
4. Edit the fields as required.
   For a detailed description of the fields, refer to Creating a Build Environment (page 210).

5. Click Save, once you have finished your changes.
   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the *Edit Build Environment* screen without saving your changes.

Deleting a Build Environment

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. Click *Delete* to delete the selected Build Environment definition.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the Environment.

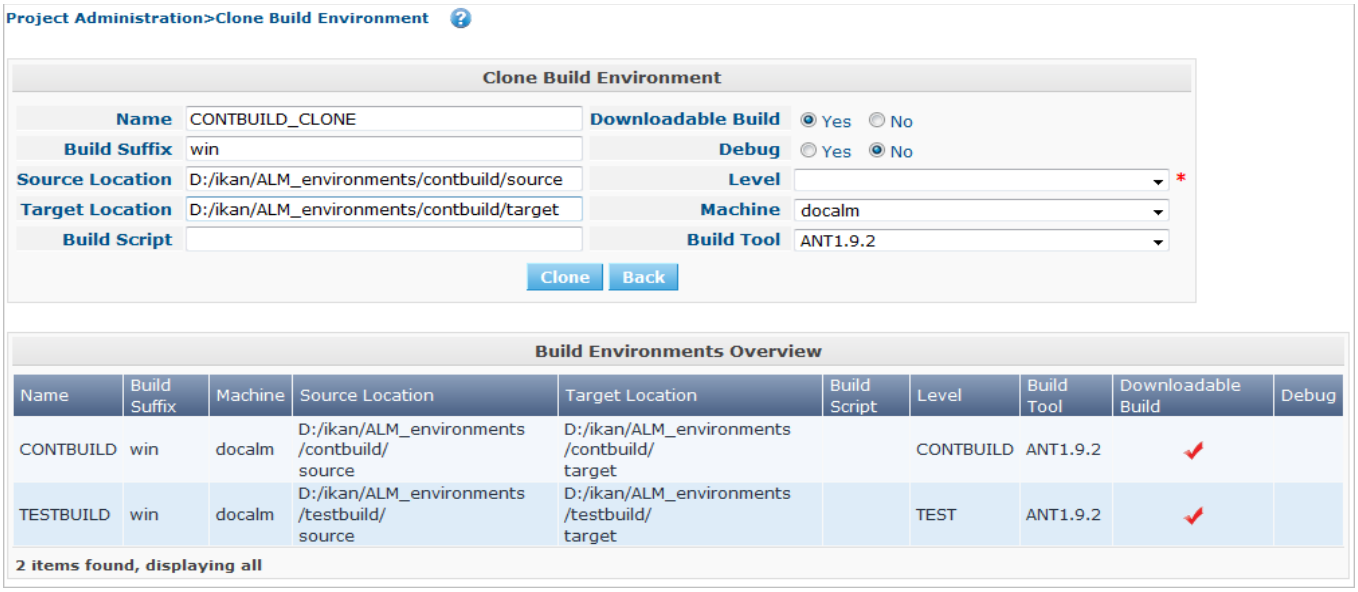
Note: Deleting a Build Environment may also delete connected Deploy Environments and Historical information linked to the Environment, like Builds, Build Logs, Deploys and Deploy Logs.

Cloning a Build Environment

When cloning an Environment, all settings, including the Phases and Parameters, will be cloned.

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. Click *Clone* to clone the selected Build Environment definition.
   The following screen is displayed:



3. On the *Clone Build Environment* panel, specify the *Name* and *Target Location*, and specify a Level for the new Environment.
   If required, you can also edit the other fields. For a detailed description of the fields, refer to Creating a Build Environment (page 210).

4. Click *Clone* to confirm the action.
   You can also click *Back* to return to the previous screen without cloning the Environment.

Build Environment Phases

When creating a Build Environment IKAN ALM will automatically link the default flow of Build Environment Phases to it.

The default Phases are:

- Transport Source
- Verify Build Script
- Execute Build Script
- Transport Deploy Script
- Compress Build
- Archive Result
- Cleanup Source
- Cleanup Result

When executing a Build for this Environment, a log will be created for each of the Phases. See Level Request Detail on page 65.

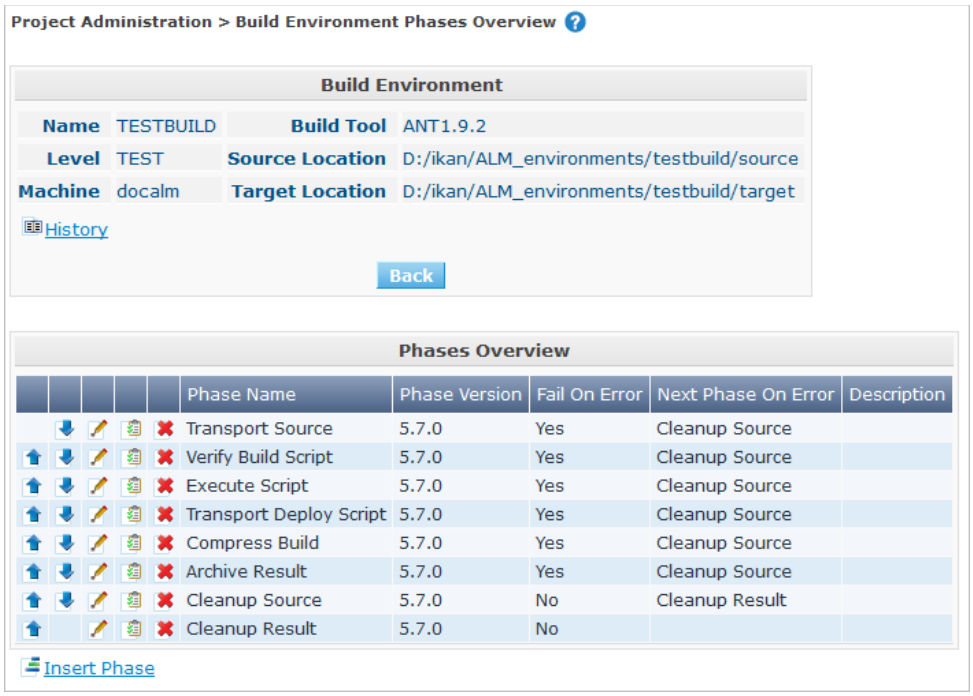
Refer to the following procedures for more information:

- The Build Environment Phases Overview Screen (page 216)
- Inserting a Build Environment Phase (page 217)
- Editing a Build Environment Phase (page 219)
- Changing the Order of the Build Environment Phases (page 219)
- Viewing the Build Environment Phase Parameters (page 220)
- Deleting a Build Environment Phase (page 223)

The Build Environment Phases Overview Screen

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. On the *Build Environments Overview* panel, click the *Edit Phases* link.
   The *Build Environment Phases Overview* screen is displayed:



Note: A link to this screen is also available on the *Edit Build Environment* screen.

3. Use the links on the *Phases Overview* panel, if required.
   The following links are available:
   - The *Up* and *Down* links to change the order of the Phases.
   - The *Edit* link to edit the Phase's settings. See Editing a Build Environment Phase on page 219.
   - The *View Parameters* link to manage the mandatory and optional Phase Parameters. See Viewing the Build Environment Phase Parameters on page 220.
   - The *Delete* link to delete a Phase. See Deleting a Build Environment Phase on page 223.

   Note: Changing the Phases can have undesirable consequences on the Lifecycle. For more information, refer to the document *HOW TO Using and Developing a Phase in IKAN ALM*.

4. Insert a Phase, if required.
   Select the *Insert Phase* link underneath the *Phases Overview* panel.
   See Inserting a Build Environment Phase on page 217.

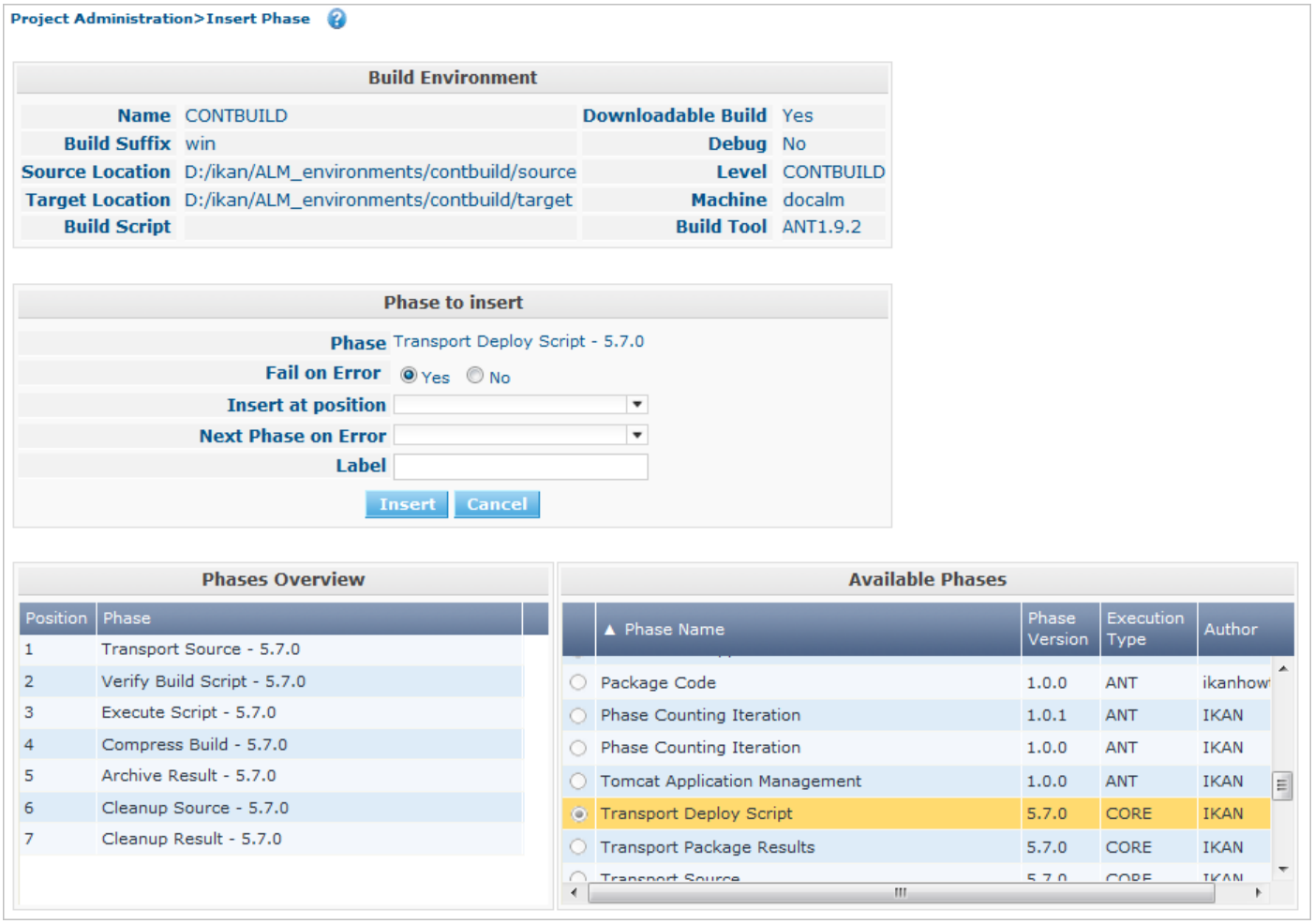
5. When done, click *Back* to return to the *Build Environments Overview* screen.

Inserting a Build Environment Phase

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. On the *Build Environments Overview* panel, click the *Edit Phases* link.

3. On the *Phases Overview* panel, click the *Insert Phase* link.
   The *Insert Phase* screen is displayed.



4. Select a Phase to insert from the *Available Phases* panel.

5. Fill out the fields for the new Phase.
   The following fields are available:

Field	Meaning
Phase	from the *Available Phases* panel, select the Phase to add.
Fail on Error	In this field, indicate whether the Build is considered failed when this Phase goes into Error.
Insert at Position	This field indicates at which position the Phase will be inserted into the Build Environment workflow. The Phase Position is also indicated on the *Phases Overview* panel.
Next Phase On Error	In this field, indicate the next Phase to execute in case this Phase goes into Error.
Label	In this optional field you can add a Label for the Phase to be inserted. In case you use the same Phase several times, adding a label is useful to provide additional information concerning the usage of the Phase.

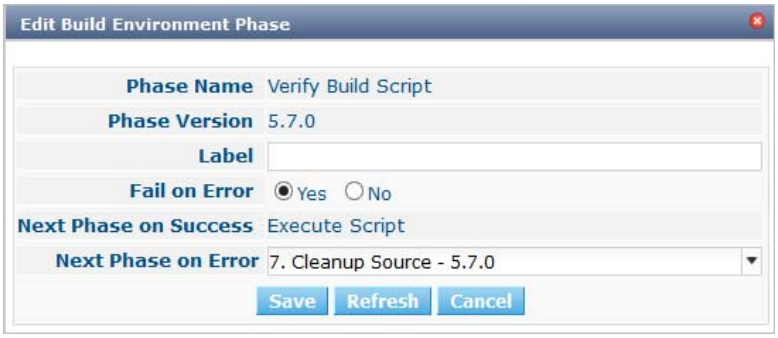
6. Click *Insert* to confirm the creation of the new Phase.
   You can also click *Cancel* to return to the previous screen without saving the changes.

Editing a Build Environment Phase

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. On the *Build Environment Overview* panel, click the *Edit Phases* link.

3. Click the *Edit* link in front of the Phase you want to edit.
   The *Edit Build Environment Phase* window is displayed.



4. Edit the fields on the *Edit Build Environment Phase* panel.
   For a description of the fields, refer to the section Inserting a Build Environment Phase (page 217).

5. Click *Save* to save your changes.
   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the previous screen without saving your changes.

Changing the Order of the Build Environment Phases

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. On the *Build Environments Overview* panel, click the *Edit Phases* link.

3. Use the *Up* and *Down* links in front of a Build Environment Phase to change its position in the sequence.

4. Click *Back* to return to the *Build Environments Overview* screen.

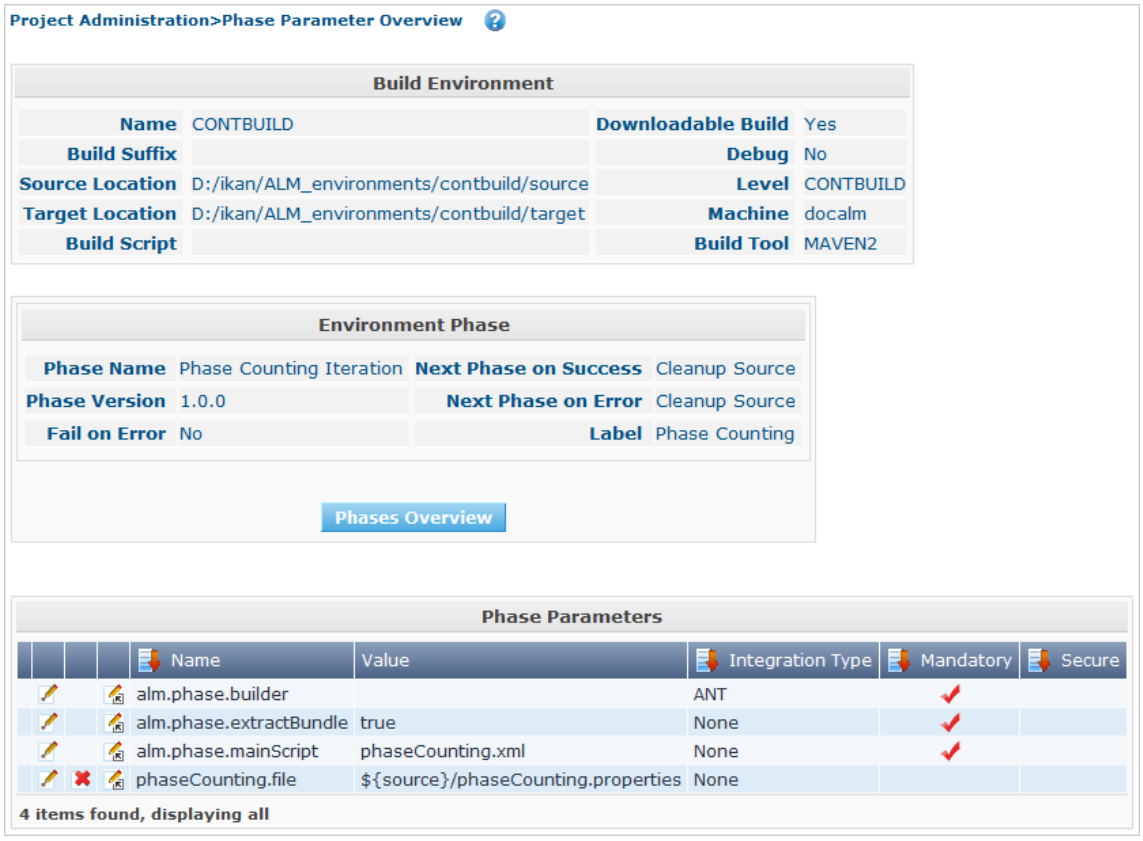
Avoid changing a Phase's position in such a way that its *Next Phase on Error* is in an earlier position in the workflow: this could result in an infinite loop.

Viewing the Build Environment Phase Parameters

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. On the *Build Environments Overview* panel, click the *Edit Phases* link.

3. Click the *View Parameters* links in front of a Build Environment Phase you want to manage the Phase Parameters for.
   The *Phase Parameter Overview* screen is displayed.



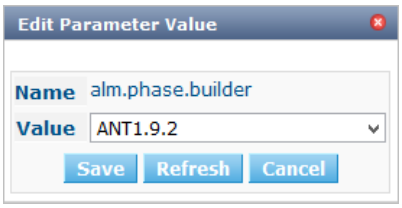
4. Verify the Build Environment Phase Parameters.
   The *Phase Parameters* panel displays all the defined Parameters of the Build Environment Phase and allows you to create non-mandatory Phase Parameters.
   The following fields are available:

Field	Meaning
Name	The name of the Parameter. This field may not be changed since it is defined in Global Administration.
Value	The Value of the Parameter. Initially, when the Phase is inserted, the value will be copied from the Default Value in Global Administration (if provided). This field may be changed by Editing the Phase Parameter.

Field	Meaning
Integration Type	This field indicates whether the value of the Parameter is a simple text value, or whether it represents a link (an integration) to an IKAN ALM Global Administration object type. The possible values are: • None: the value is simple text • Transporter: link to a Transporter • VCR: link to a Version Control Repository • ITS: link to an Issue Tracking System • Scripting Tool: link to a Scripting Tool • ANT: link to an Ant Scripting Tool • GRADLE: link to a Gradle Scripting Tool • NANT: link to a NAnt Scripting Tool • MAVEN2: link to a Maven2 Scripting Tool This field may not be changed since it is defined in Global Administration.
Mandatory	This field indicates whether the Parameter has been created automatically when the Phase is inserted in the Level. This is the case for Mandatory Parameters. Non-mandatory Parameters must be created after the Phase has been inserted in the Level, using the *Create Parameter* link. This field may not be changed since it is defined in Global Administration.
Secure	This field indicates whether the Parameter is secured or not. This field may not be changed since it is defined in Global Administration.

5. Click the *Edit Parameter* link next to a Phase Parameter.

The following pop-up window will be displayed.



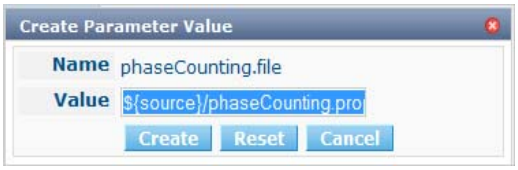
Set the value of the Build Environment Phase Parameter and click *Save* to save the value.

You can also click:

- *Reset* to retrieve the settings from the database.
- *Cancel* to return to the *Phase Parameter Overview* screen without saving a value.

6. If you want to create a non-mandatory Phase Parameter, click the *Create Parameter* link next to a Phase Parameter.

The following pop-up window will be displayed.



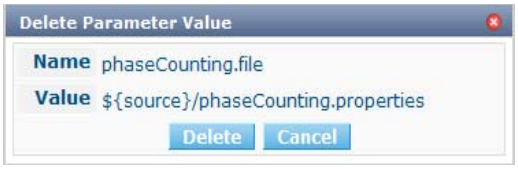
If a default Parameter value has been set in Global Administration, that value will be suggested.

Set the value of the Build Environment Phase Parameter and click *Create*.

- *Reset* to retrieve the settings from the database.
- *Cancel* to return to the *Phase Parameter Overview* screen without saving a value.

7. If you want to delete a non-mandatory Phase Parameter, click the *Delete Parameter* link next to a Phase Parameter.

   The following pop-up window will be displayed.

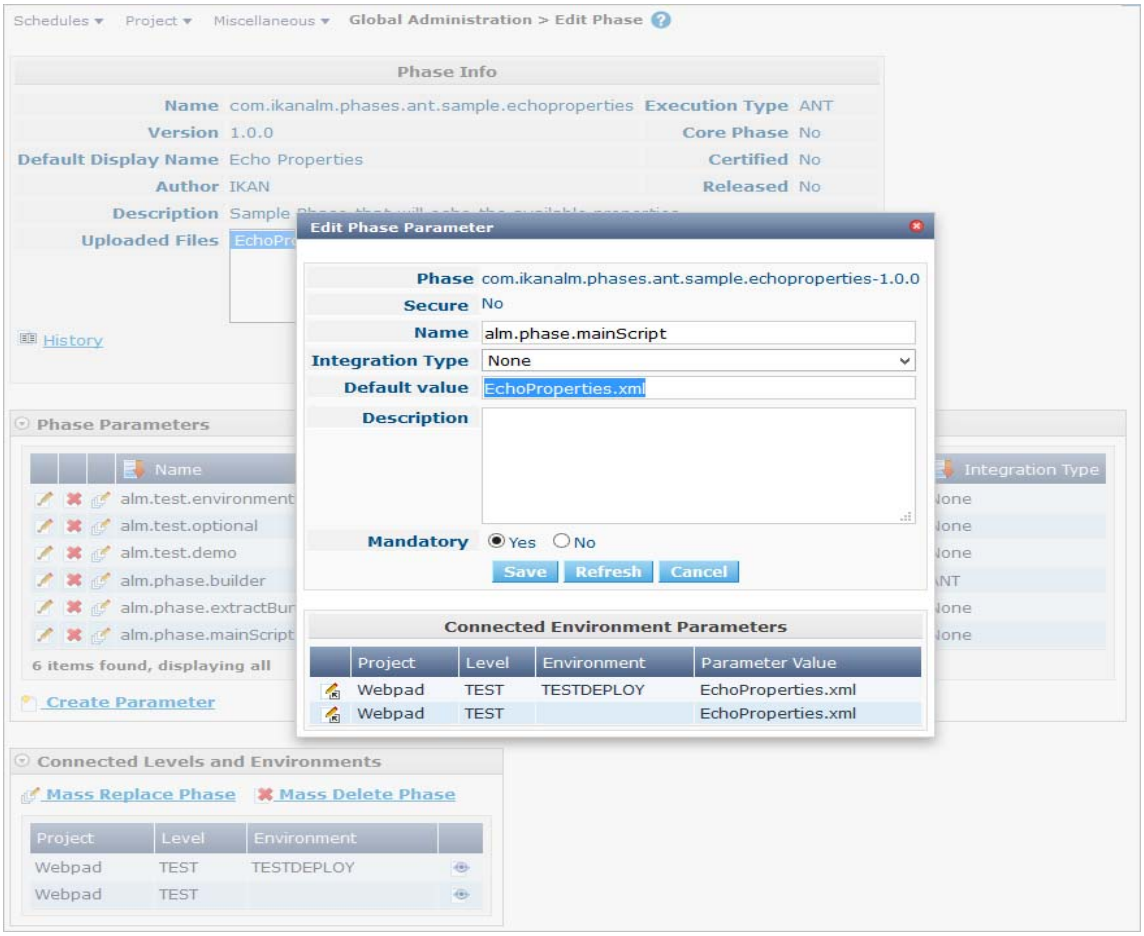


   Click *Delete* to confirm the deletion of the mandatory Build Environment Phase Parameter.

   You can also click *Cancel* to close the pop-up window without deleting the Parameter.

8. Click the *Edit Global Phase Parameter* link next to a Phase Parameter.

   The User will be redirected to the *Edit Phase* screen (in the Global Administration context) and the *Edit Phase Parameter Value* pop-up window is opened.



   You can Edit the Global Phase Parameter as described in the section Editing Phase Parameters (page 437).

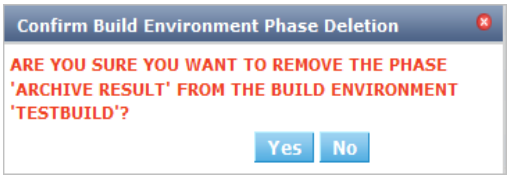
   To go back to the Phase Parameter in the *Project Administration* context, click the appropriate *Edit Environment Phase Parameter* link in the *Connected Environment Parameters* panel.

Deleting a Build Environment Phase

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. On the *Build Environments Overview* panel, click the *Edit Phases* link.

3. On the *Phases Overview* panel, click the Delete link.
   The *Delete Build Environment Phase* screen is displayed.



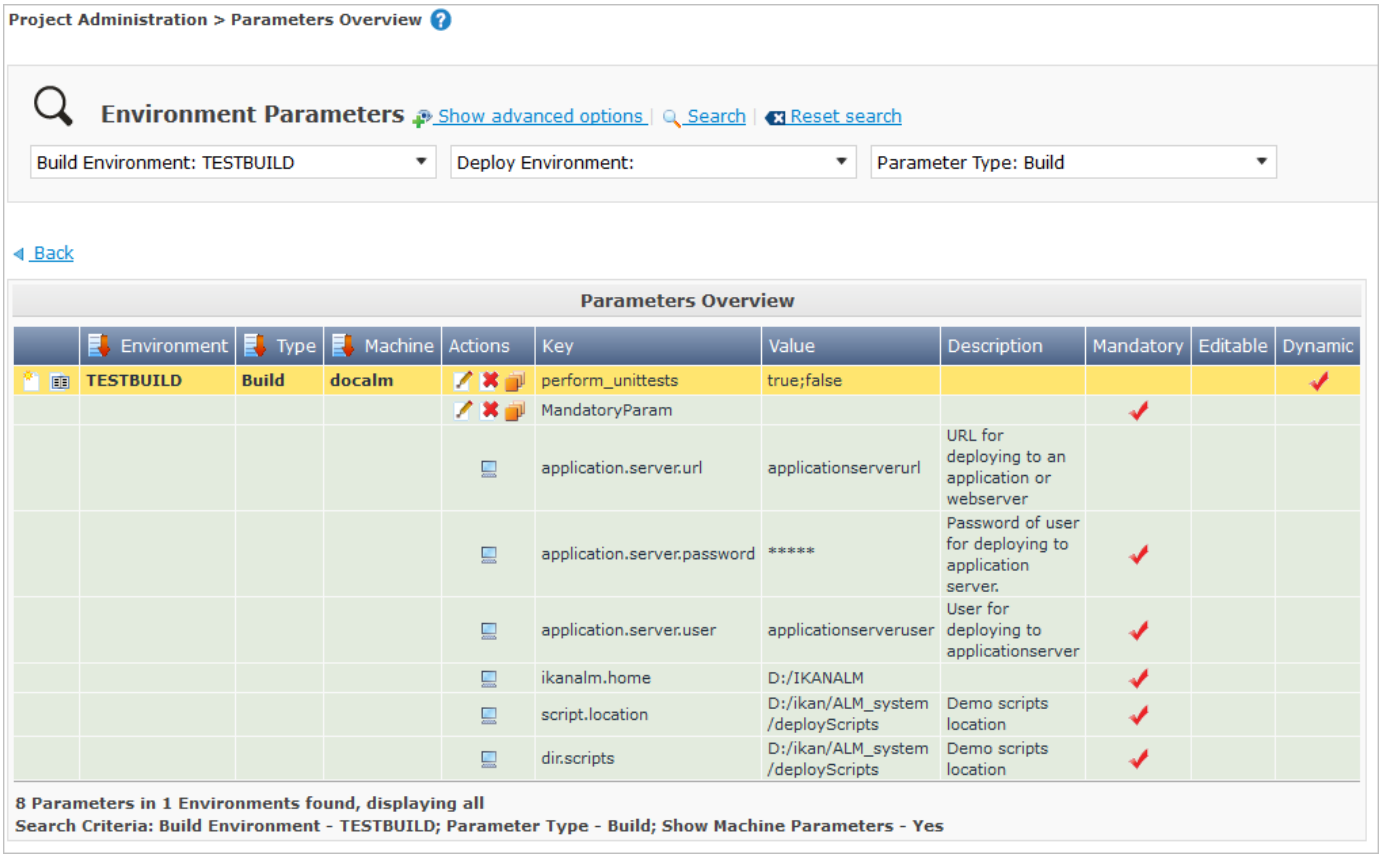
4. Click *Yes* to confirm the deletion of the Phase.
   You can also click *No* to return to the previous screen without deleting the Build Environment Phase.

Viewing the Build Environment Parameters

1. Switch to the *Build Environments Overview* screen.
   See The Build Environments Overview Screen on page 212.

2. Click *View Parameters* to view all parameters defined for the selected Build Environment.

The following screen is displayed:



For a more detailed description of this screen, refer to the section Creating Environment Parameters (page 242).

Viewing the Build Environment History

1. Switch to the Build Environments Overview.
   See The Build Environments Overview Screen on page 212.

2. Click the *History* link to display the *Build Environment History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the previous screen.

CHAPTER 20
Deploy Environments

Deploy Environments are locations where a Deploy Tool uses a Deploy script to unpack and deploy a Build Result. The Deploy Environment can be OS-specific and the Deploy process can be steered with Deploy Parameters.

Deploy Environments must be assigned to a Level in a Lifecycle in order to be active.

The Deploy Environments submenu option lets you start the following procedures:

- Creating a Deploy Environment (page 225)
- The Deploy Environments Overview Screen (page 228)

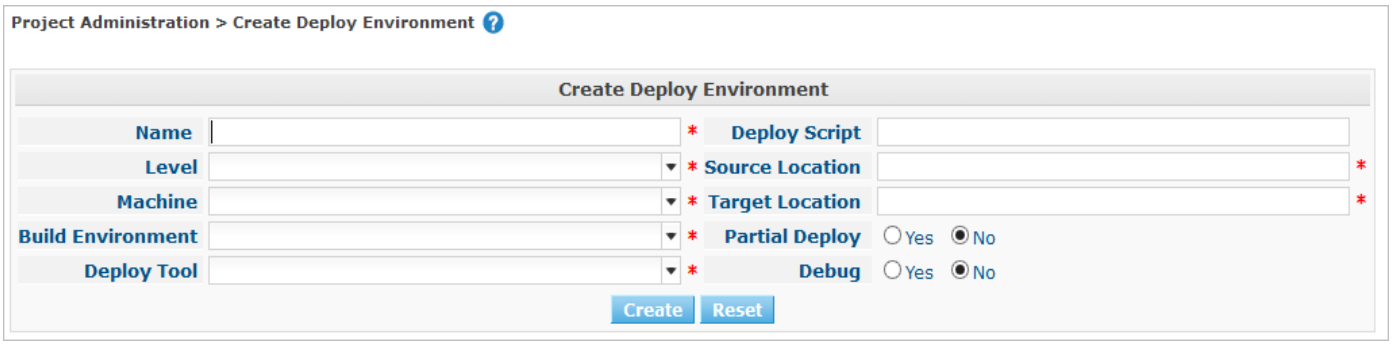
  The *Deploy Environments Overview* screen lets you verify and change the information concerning Deploy Environments and related Phases. The following actions are possible:

  - Editing a Deploy Environment (page 229)
  - Deleting a Deploy Environment (page 230)
  - Cloning a Deploy Environment (page 230)
  - Editing a Deploy Environment Phase (page 234)
  - Viewing the Deploy Environment Phase Parameters (page 235)
  - Viewing the Deploy Environment History (page 239)
- Deploy Parameters

  The *Parameters Overview* screen lets you create, edit, delete and copy Deploy Parameters and allows to check their history. For more information, refer to the chapter Environment Parameters (page 240).

20.1. Creating a Deploy Environment

1. In the *Project Administration context*, select *Deploy Environments > Create*.
   The following screen is displayed:



2. Fill out the fields for the new Deploy Environment.

The following fields are available. Fields marked with a red asterisk are mandatory

Field	Meaning
Name	In this field, enter the name for the new Deploy Environment.
Level	From the drop-down list, select the Level to which you want to assign this Deploy Environment.
Machine	From the drop-down list, select the Machine on which this Deploy Environment is physically available.
Build Environment	From the drop-down list, select the Build Environment this Deploy Environment is based on. The Build Result of this Build Environment will be deployed.
Deploy Tool	From the drop-down list, select the Deploy Tool for this Deploy Environment. Paths defined in the selected Deploy Tool must exist on the linked Machine.
Deploy Script	This field may contain the relative path to the Deploy Script that is selected specifically for this Deploy Environment. If the field is empty, the default Deploy Script whose relative path is defined on the Project Settings screen will be used for this Deploy Environment.
Source Location	This field contains the path to the physical location where the unpacked build result is placed. IKAN ALM adds the Deploy OID: *SourceLocation/DeployOID/ProjectName_VCR* This location is passed to the Deploy Script as a parameter named *source*.
Target Location	For a Deploy Environment, this field contains the path to the physical location where the Deploy Script will deploy the Build result. This location is passed to the Deploy Script as a parameter named *target*.

Field	Meaning
Partial Deploy	Indicate whether the Partial Deploy option must be activated for this Deploy Environment. If the option is activated, IKAN ALM will analyze the differences between the two build results on the IKAN ALM Server and only new and modified files will be transported to the Agent. This is achieved as follows: Both the newly compiled Build result and the base-line Build (the one the new Build results are compared with) are decompressed (unzipped). The decompressed files are compared and the following actions are taken: • Identical files are deleted and not deployed. • Modified and new files (available in the new but not in the base-line Build results) are transported. • For deleted files (available in the base-line Build result, but not in the new one), dummy files with names in the format *File-Name.to_be_deleted* are created. The Deploy Script must be written in such a way, that it deletes the matching files from the Deploy Environment's Source Location.
Debug	Select whether or not the Debug option must be activated for this Deploy Environment. If Debug is activated for a Deploy Environment, the Build Clean-up action (_ action 28 of the Build and Deploy Level Requests Overview (page 35)) will not be performed, so that the user may use the available build results to run the deploy script manually for testing purposes.

3. Verify the information on the *Deploy Environments Overview* panel.

Note: This Overview lists **all** Deploy Environments defined for **all** Levels belonging to the selected Project.

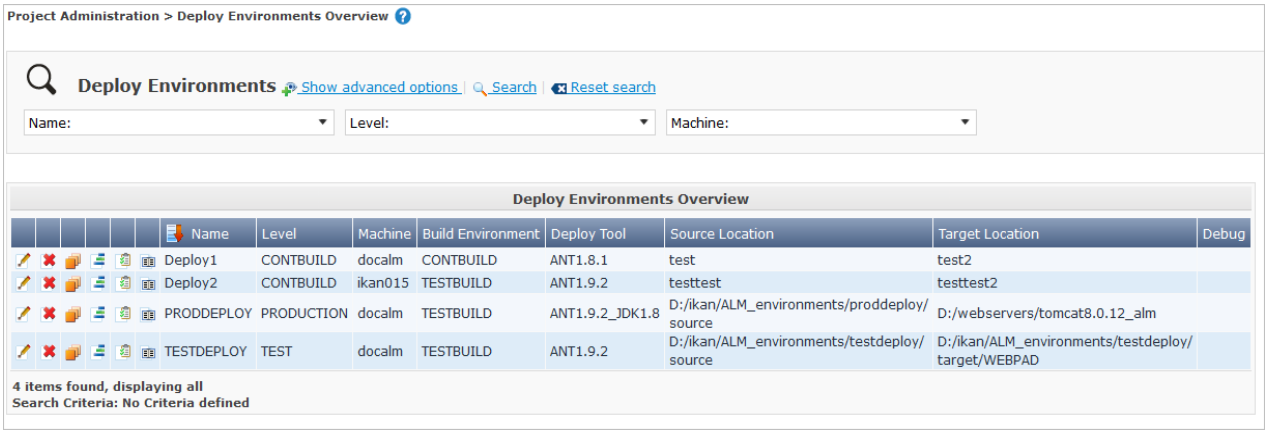
For more information on the available links, refer to The Deploy Environments Overview Screen (page 228).

4. Click *Create* to confirm the creation of the new Deploy Environment.

You can also click the *Reset* button to clear the fields and restore the initial values.

20.2. The Deploy Environments Overview Screen

1. In the *Project Administration context*, select *Deploy Environments > Overview*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields,

3. Verify the Information available on the *Deploy Environments Overview*.
   For a detailed description of the fields, see Creating a Deploy Environment (page 225).

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

4. Depending on your access rights, the following links may be available on the *Deploy Environments Overview* panel:

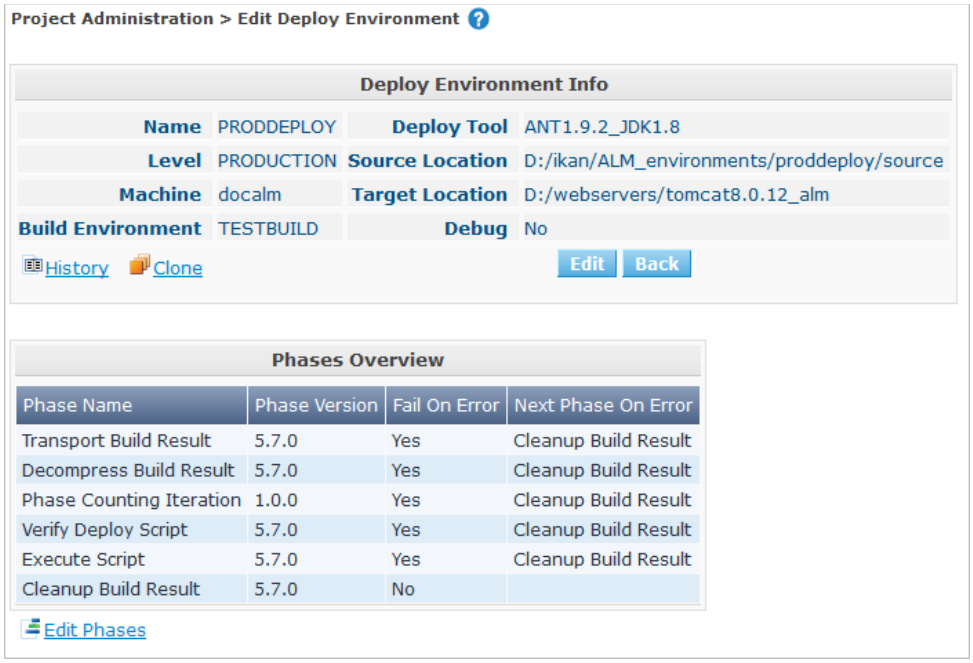
	Edit This option allows editing a the Deploy Environment settings. See Editing a Deploy Environment on page 229.
	Delete This option allows deleting a Deploy Environment. See Deleting a Deploy Environment on page 230.
	clone This option allows cloning a Deploy Environment. See Cloning a Deploy Environment on page 230.
	Edit Phases This option allows editing the Deploy Environment Phases. See Editing a Deploy Environment Phase on page 234.

	View Parameters This option allows viewing the Deploy Environment Parameters. See Viewing the Deploy Environment Parameters on page 239.
	History This option allows viewing the Deploy Environment History. See Viewing the Deploy Environment History on page 239.

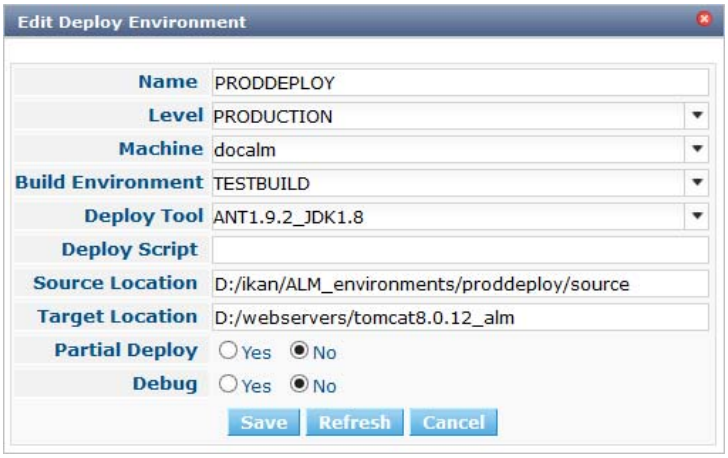
Editing a Deploy Environment

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. Click *Edit* to change the selected Deploy Environment definition.
   The following screen is displayed:



3. Click the *Edit* button on the *Build Environment Info* panel.
   The *Edit Build Environment* window is displayed:



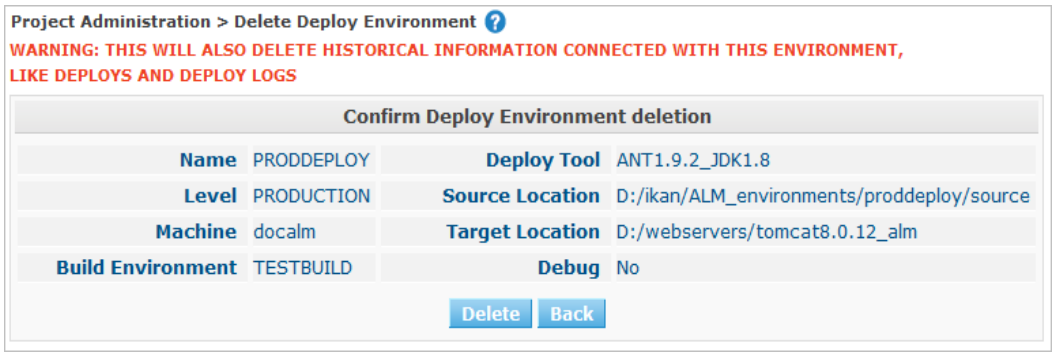
4. Edit the fields as required.
   For a detailed description of the fields, refer to Creating a Deploy Environment (page 225).

5. Click Save, once you have finished your changes.
   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the *Edit Build Environment* screen without saving your changes.

Deleting a Deploy Environment

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. Click *Delete* to delete the selected Deploy Environment definition.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the Environment.

Note: Deleting a Deploy Environments may also delete Historical information linked to the Environment, like Deploys and Deploy Logs.

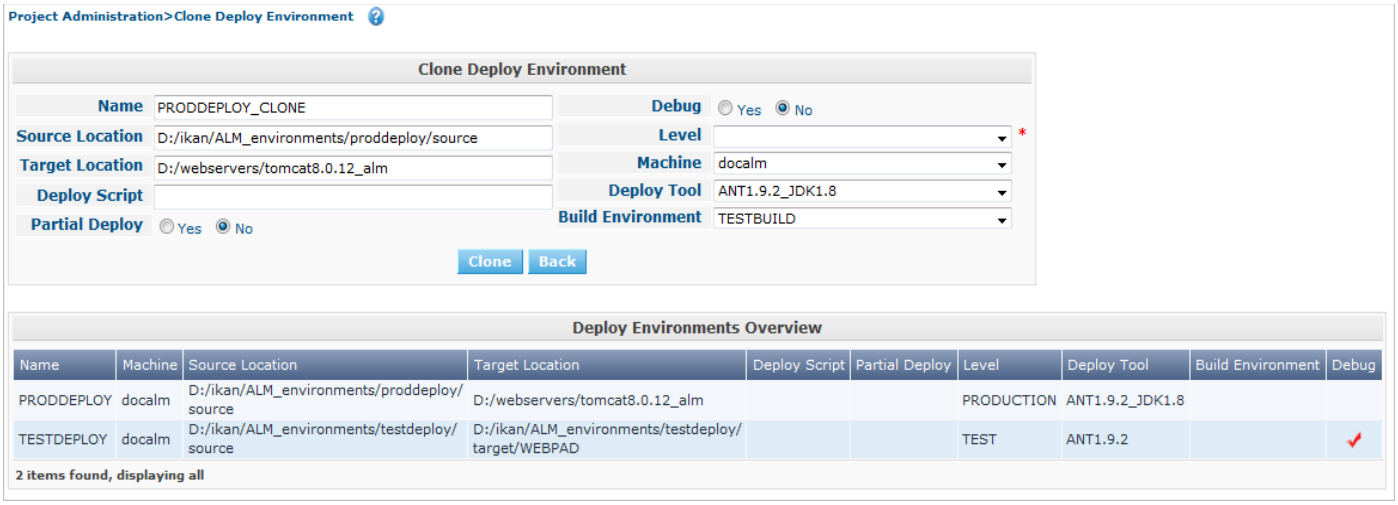
Cloning a Deploy Environment

When cloning an Environment, all settings, including the Phases and Parameters, will be cloned.

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. Click *Clone* to clone the selected Deploy Environment definition.

The following screen is displayed:



3. On the *clone Deploy Environment* panel, specify the *Name* and *Target Location*, and specify a Level for the new Environment.

   If required, you can also edit the other fields. For a detailed description of the fields, refer to Creating a Deploy Environment (page 225).

4. Click *Clone* to confirm the action.

   You can also click *Back* to return to the previous screen without cloning the Environment.

Deploy Environment Phases

When creating a Deploy Environment, IKAN ALM will automatically link the default flow of Deploy Environment Phases to it.

The default Phases are:

- Transport Build Result
- Decompress Build Result
- Verify Deploy Script
- Execute Deploy Script
- Cleanup Build Result

When executing a Deploy for this Environment, a log will be created for each of the Phases. See Level Request Detail on page 65.

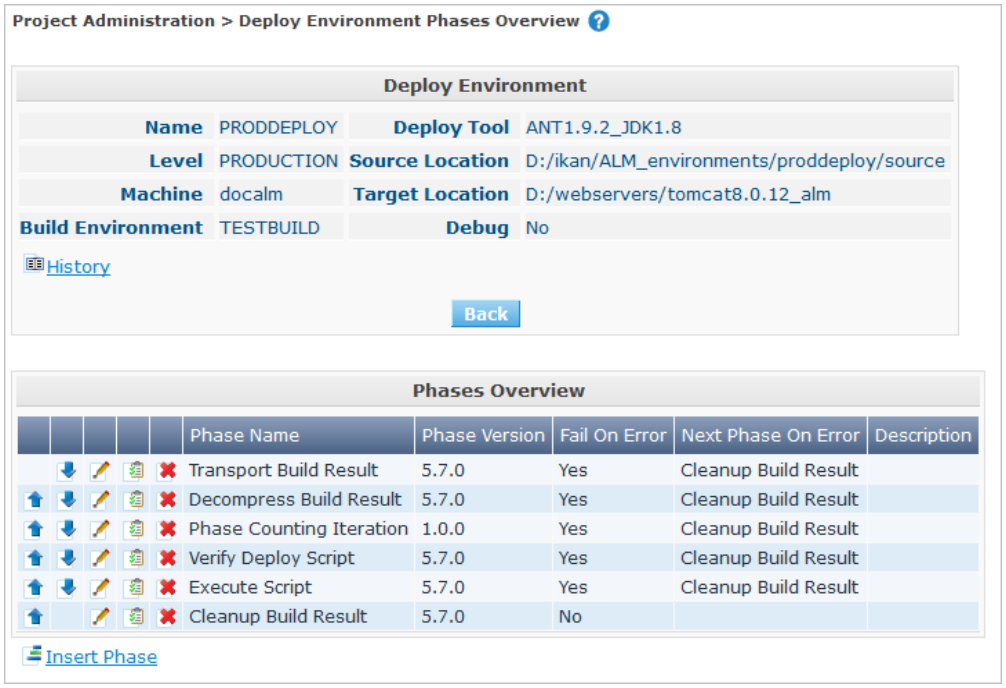
Refer to the following procedures for more information:

- The Deploy Environment Phases Overview Screen (page 232)
- Inserting a Deploy Environment Phase (page 233)
- Editing a Deploy Environment Phase (page 234)
- Changing the Order of the Deploy Environment Phases (page 234)
- Viewing the Deploy Environment Phase Parameters (page 235)
- Deleting a Deploy Environment Phase (page 238)

The Deploy Environment Phases Overview Screen

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. On the *Deploy Environments Overview* panel, click the *Edit Phases* link.
   The *Deploy Environment Phases Overview* screen is displayed:



Note: A link to this screen is also available on the *Edit Deploy Environment* screen.

3. Use the links on the *Phases Overview* panel, if required.
   The following links are available:
   - The *Up* and *Down* links to change the order of the Phases.
   - The *Edit* link to edit the Phase's settings. See Editing a Deploy Environment Phase on page 234.
   - The *View Parameters* link to manage the mandatory and optional Phase Parameters. See Viewing the Deploy Environment Phase Parameters on page 235.
   - The *Delete* link to delete a Phase. See Deleting a Deploy Environment Phase on page 238.

4. Insert a Phase, if required.
   Select the *Insert Phase* link underneath the *Phases Overview* panel.
   See Inserting a Deploy Environment Phase on page 233.

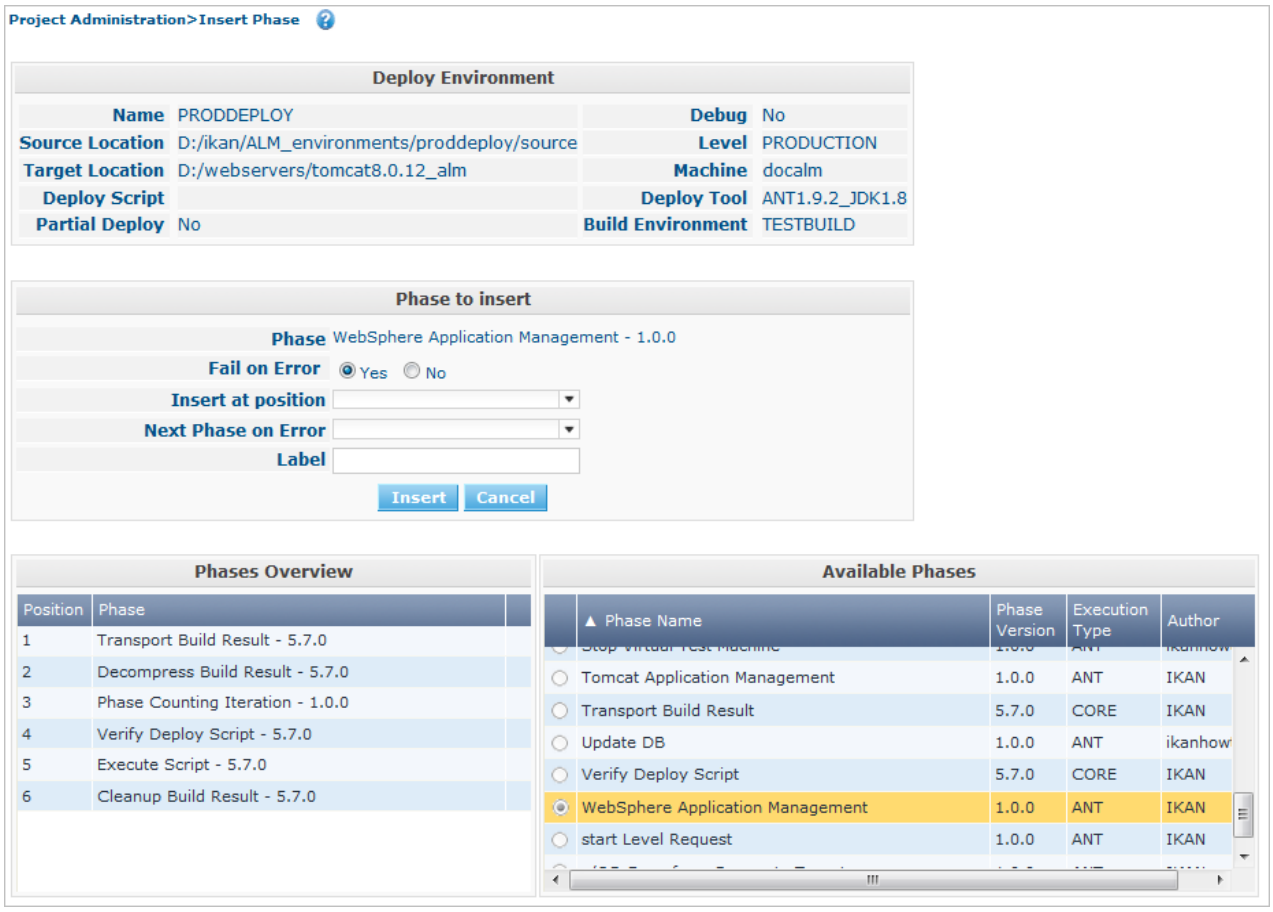
5. When done, click *Back* to return to the *Deploy Environments Overview* screen.

Inserting a Deploy Environment Phase

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. On the *Deploy Environments Overview* panel, click the *Edit Phases* link.

3. On the *Phases Overview* panel, click the *Insert Phase* link.
   The *Insert Phase* is displayed.



4. Select a Phase to insert from the *Available Phases* panel.

5. Fill out the fields for the new Phase.
   The following fields are available:

Field	Meaning
Phase	from the *Available Phases* panel, select the Phase to add.
Fail on Error	In this field, indicate whether the Deploy is considered failed when this Phase goes into Error.

Field	Meaning
Insert at Position	This field indicates at which position the Phase will be inserted into the Deploy Environment workflow. The Phase Position is also indicated on the *Phases Overview* panel.
Next Phase On Error	In this field, indicate the next Phase to execute in case this Phase goes into Error.
Label	In this optional field you can add a Label for the Phase to be inserted. In case you use the same Phase several times, adding a label is useful to provide additional information concerning the usage of the Phase.

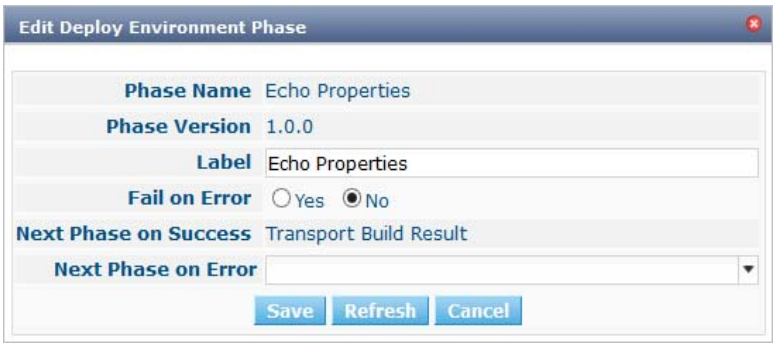
6. Click *Insert* to confirm the creation of the new Phase.
   You can also click *Cancel* to return to the previous screen without saving the changes.

Editing a Deploy Environment Phase

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. On the *Deploy Environments Overview* panel, click the *Edit Phases* link.

3. Click the Edit link in front of the Phase you want to edit.
   The *Edit Deploy Environment Phase* window is displayed.



4. Edit the fields on the *Edit Deploy Environment Phase* panel.
   For a description of the fields, refer to the section Inserting a Deploy Environment Phase (page 233).

5. Click *Save* to save your changes.
   You can also click *Refresh* to retrieve the settings from the database or *Cancel* to return to the previous screen without saving your changes.

Changing the Order of the Deploy Environment Phases

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. On the *Deploy Environments Overview* panel, click the *Edit Phases* link.

3. Use the *Up* and *Down* links in front of a Deploy Environment Phase to change its position in the sequence.

4. Click *Back* to return to the *Deploy Environments Overview* screen.

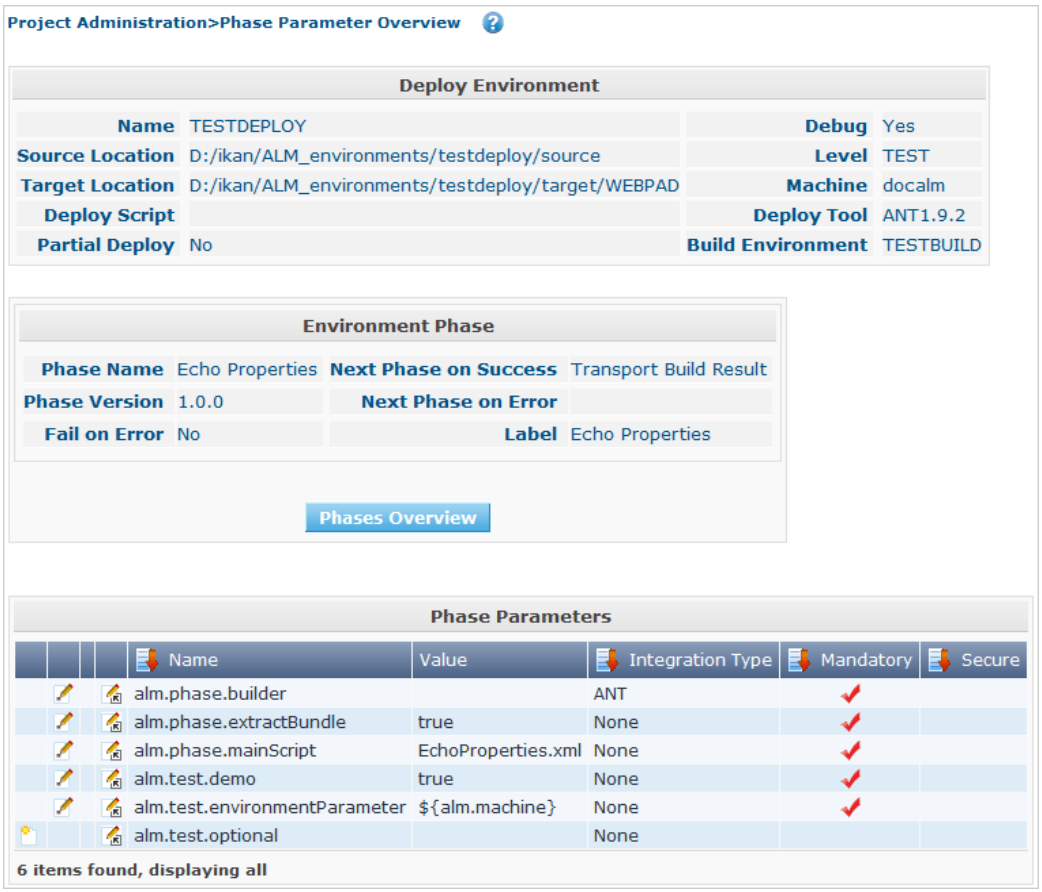
> Avoid changing a Phase's position in such a way that its *Next Phase on Error* is in an earlier position in the workflow: this could result in an infinite loop.

Viewing the Deploy Environment Phase Parameters

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. On the *Deploy Environments Overview* panel, click the *Edit Phases* link.

3. Click the *View Parameters* links in front of a Deploy Environment Phase you want to manage the Phase Parameters for.
   The *Phase Parameter Overview* screen is displayed.



4. Verify the Deploy Environment Phase Parameters.

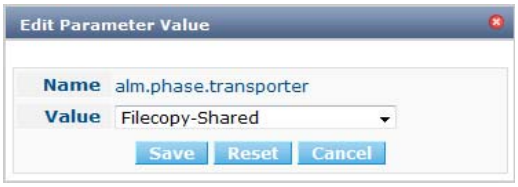
The *Phase Parameters* panel displays all the defined Parameters of the Deploy Environment Phase and allows you to create non-mandatory Phase Parameters.

The following fields are available:

Field	Meaning
Name	The name of the Parameter. This field may not be changed since it is defined in Global Administration.
Value	The Value of the Parameter. Initially, when the Phase is inserted, the value will be copied from the Default Value in Global Administration (if provided). This field may be changed by Editing the Phase Parameter.
Integration Type	This field indicates whether the value of the Parameter is a simple text value, or whether it represents a link (an integration) to an IKAN ALM Global Administration object type. The possible values are: • None: the value is simple text • Transporter: link to a Transporter • VCR: link to a Version Control Repository • ITS: link to an Issue Tracking System • Scripting Tool: link to a Scripting Tool • ANT: link to an Ant Scripting Tool • GRADLE: link to a Gradle Scripting Tool • NANT: link to a NAnt Scripting Tool • MAVEN2: link to a Maven2 Scripting Tool This field may not be changed since it is defined in Global Administration.
Mandatory	This field indicates whether the Parameter has been created automatically when the Phase is inserted in the Level. This is the case for Mandatory Parameters. Non-mandatory Parameters must be created after the Phase has been inserted in the Level, using the *Create Parameter* link. This field may not be changed since it is defined in Global Administration.
Secure	This field indicates whether the Parameter is secured or not. This field may not be changed since it is defined in Global Administration.

5. Click the *Edit Parameter* link next to a Phase Parameter.

The following pop-up window will be displayed.



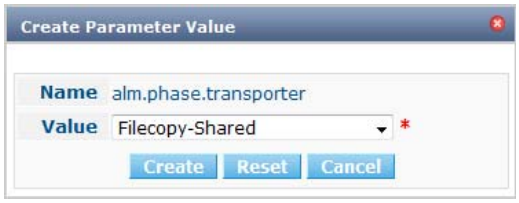
Set the value of the Deploy Environment Phase Parameter and click *Save* to save the value.

You can also click:

- *Reset* to retrieve the settings from the database.
- *Cancel* to return to the *Phase Parameter Overview* screen without saving a value.

6. If you want to create a non-mandatory Phase Parameter, click the *Create Parameter* link next to a Phase Parameter.

   The following pop-up window will be displayed.



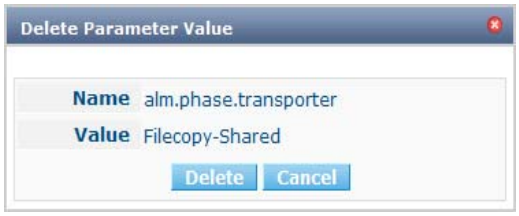
   If a default Parameter value has been set in Global Administration, that value will be suggested.

   Set the value of the Deploy Environment Phase Parameter and click *Create*.

   - *Reset* to retrieve the settings from the database.
   - *Cancel* to return to the *Phase Parameter Overview* screen without saving a value.

7. If you want to delete a non-mandatory Phase Parameter, click the *Delete Parameter* link next to a Phase Parameter.

   The following pop-up window will be displayed.

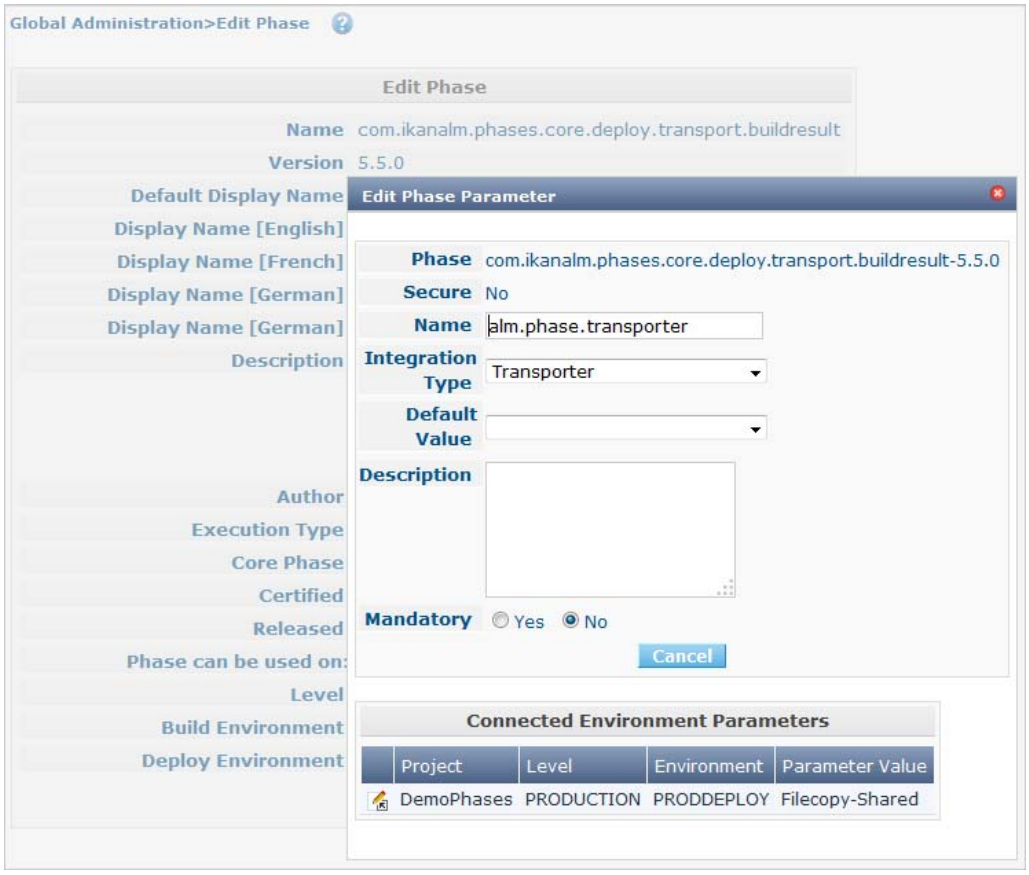


   Click *Delete* to confirm the deletion of the mandatory Deploy Environment Phase Parameter.

   You can also click *Cancel* to close the pop-up window without deleting the Parameter.

8. Click the *Edit Global Phase Parameter* link next to a Phase Parameter.

The User will be redirected to the *Edit Phase* screen (in the Global Administration context) and the *Edit Phase Parameter Value* pop-up window is opened.



You can Edit the Global Phase Parameter as described in the section Editing Phase Parameters (page 437).

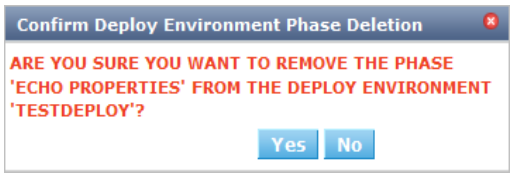
To go back to the Phase Parameter in the Project Administration context, click the appropriate *Edit Environment Phase Parameter* link in the *Connected Environment Parameters* panel.

Deleting a Deploy Environment Phase

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. On the *Deploy Environments Overview* panel, click the *Edit Phases* link.

3. On the *Phases Overview* panel, click the Delete link.
   The *Delete Deploy Environment Phase* screen is displayed.

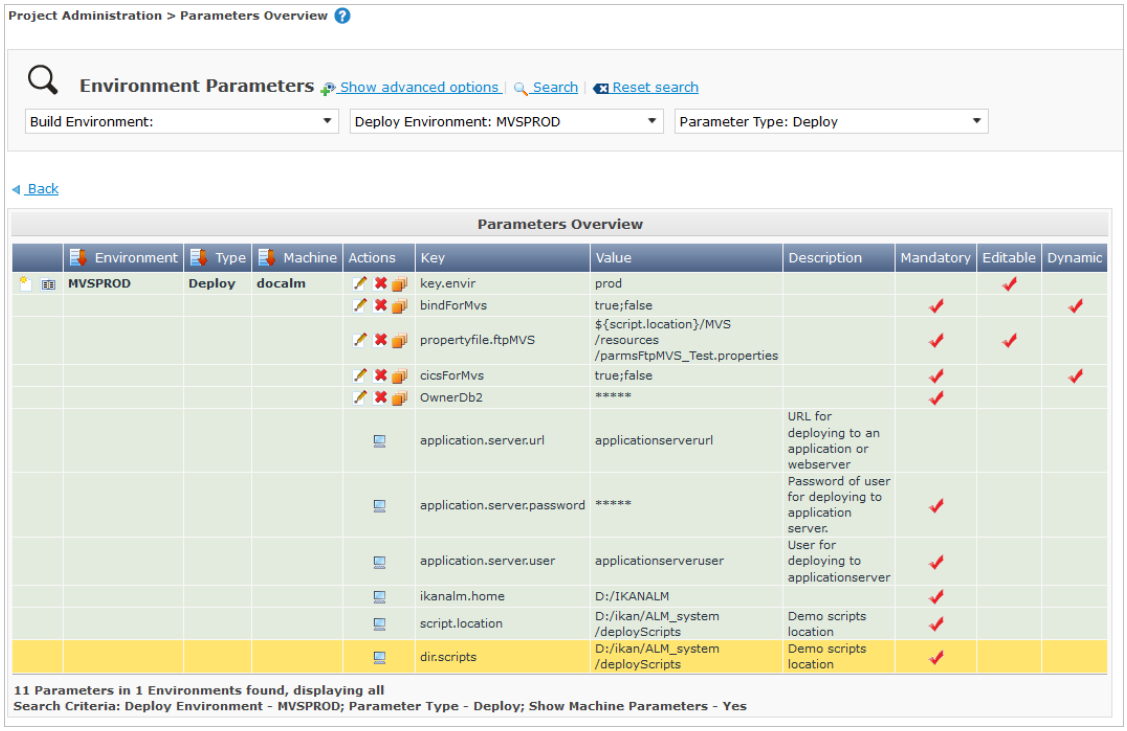


4. Click *Yes* to confirm the deletion of the Phase.
   You can also click *No* to return to the previous screen without deleting the Deploy Environment Phase.

Viewing the Deploy Environment Parameters

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. Click *View Parameters* to view all parameters defined for the selected Deploy Environment.
   The following screen is displayed:



   For a more detailed description of this screen, refer to the section Creating Environment Parameters (page 242).

Viewing the Deploy Environment History

1. Switch to the *Deploy Environments Overview* screen.
   See The Deploy Environments Overview Screen on page 228.

2. Click the *History* link to display the *Deploy Environment History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the previous screen.

CHAPTER 21
Environment Parameters

Environment Parameters are parameters that can be used during the *Execute Script* Phase which runs a Build/Deploy Script. They may also be used during the execution of a Custom Phase.

Depending on the Build/Deploy Tool (Execute Script Phase) or the Execution Type (Custom Phase), the defined parameters will be:

- added to the command which executes the Script (in the case of NAnt and Maven2)
- written to a file named *alm_ant.properties* (in the case of Ant) which is automatically loaded with the option `-propertyfile`. This property file is created on the fly in the source location of the Environment in the directory containing the Script (this may be a subdirectory of the source location of the Environment in case the location of the Script was defined using a relative path).

  In case of a Custom Phase, this file is located in a subdirectory of the source location, together with the extracted script and other resources of the Custom Phase.

  Once the Build/Deploy process has terminated, this file is automatically deleted, unless the *Debug* option for the Environment linked to the Level has been activated.

Environment Parameters can be divided into two categories:

- *Predefined Parameters*: these are Environment Parameters that are always available. A list of these predefined parameters can be found in:
  - Predefined Build Parameters (page 505)
  - Predefined Deploy Parameters (page 508)

  To override the predefined parameter value for a specific Environment, define a Parameter with the same name on that Environment.
- *User-defined Parameters*: The option *Build Environments* > *Build Parameters* or *Deploy Environments* > *Deploy Parameters* allows you to define your own Environment Parameters.

Note: On the *Miscellaneous* tab of the *System Settings* screen, you can specify the prefix for the predefined Parameters and properties file. See Miscellaneous on page 261.

There also exist Predefined Level Parameters. They are available during the execution of a Custom Phase on a Level, They may not be overridden, since it is not (yet) possible to define Level Parameters. For more information, refer to the appendix Predefined Level Parameters (page 503).

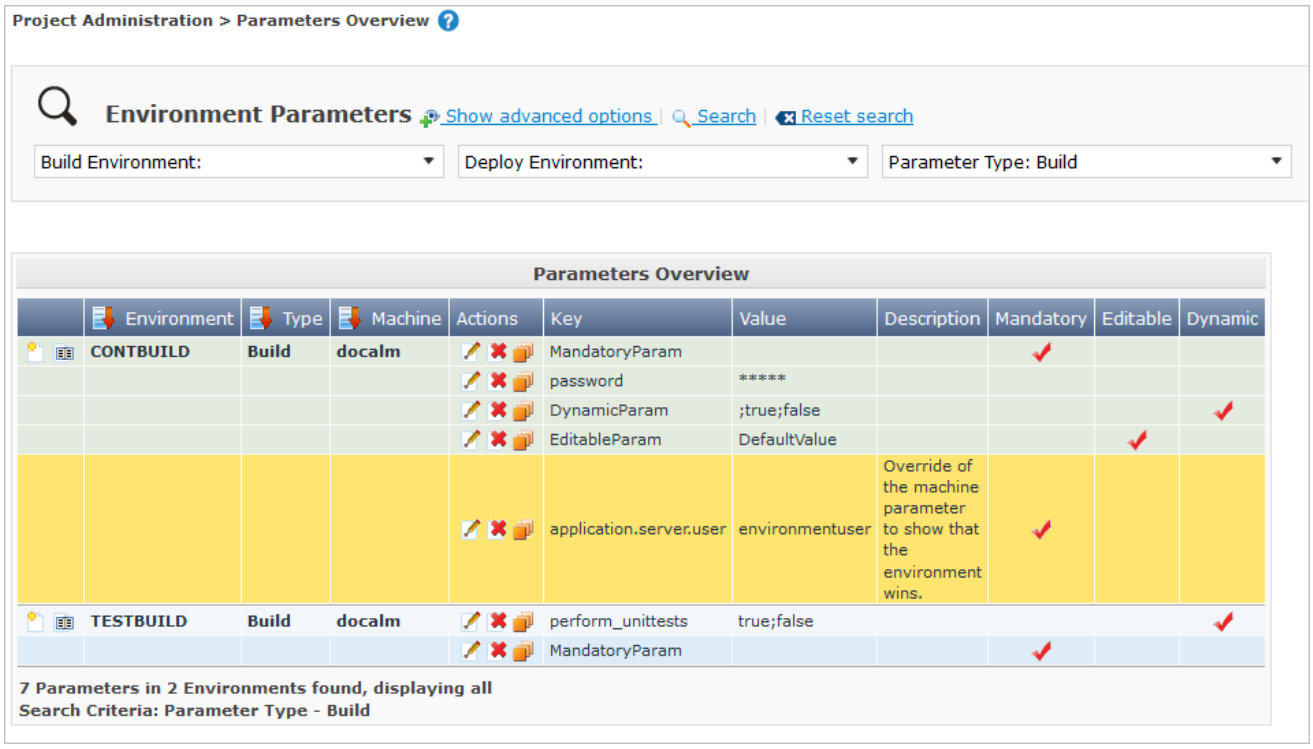
The *Parameters Overview* screen lets you create, edit, delete and copy Environment Parameters and allows checking their history. The following actions are possible:

- Creating Environment Parameters (page 242)
- Viewing the Environment Parameters History (page 244)
- Editing Environment Parameters (page 244)
- Deleting Environment Parameters (page 245)
- Copying Environment Parameters (page 245)

21.1. The Parameters Overview Screen

Note: This option is only available if you have Project Administrator Access Rights.

1. In the *Project Administration context*, select *Build Environments > Build Parameters* or *Deploy Environments > Deploy Parameters*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields,

3. Verify the information on the *Parameters Overview* panel.
   The *Parameters Overview* panel displays the defined Environment Parameters for each of the Environments.
   For a description of the fields, see Creating Environment Parameters (page 242).

Note: The icon indicates that the parameter is a Machine Parameter. See Machine Parameters on page 300. In order to show these Parameters, you should set the *Show Machine Parameters* criterion to *Yes*.
Columns marked with the icon can be sorted alphabetically (ascending or descending).

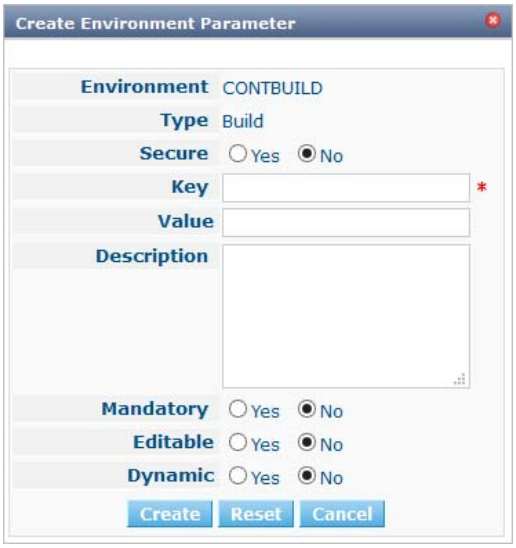
4. Depending on your access rights, the following links may be available on the *Parameters Overview* panel:

Link	Description
	Create This option is available to all Users with Project Administrator Access Rights. It allows creating an Environment Parameter. See Creating Environment Parameters on page 242.
	History This option is available to all Users with Project Administrator Access Rights. It allows viewing the Parameter history of the selected Environment. See Viewing the Environment Parameters History on page 244.
	Edit This option is available to all Users with Project Administrator Access Rights. It allows editing the selected Environment Parameter definition. See Editing Environment Parameters on page 244.
	Delete This option is available to all Users with Project Administrator Access Rights. It allows deleting the selected Environment Parameter definition and (optionally) deleting Environment Parameters with the same key linked to other Build or Deploy Environments. See Deleting Environment Parameters on page 245.
	Copy Parameter This option is available to all Users with Project Administrator Access Rights. It allows copying the selected Environment Parameter definition to one or more Environments. See Copying Environment Parameters on page 245.

21.2. Creating Environment Parameters

1. Switch to the *Parameters Overview* screen for the required Project.
   See The Parameters Overview Screen on page 241.

2. Click the *Create Parameter* link to display the Parameter Action window.



3. Fill out the fields for the new Environment Parameter.

The following fields are available. The *Key* field is mandatory:

Field	Meaning
Environment	This field displays the name of the current Environment.
Type	This field displays the type of Parameter being created: *Build* or *Deploy*.
Secure	This field indicates whether the Parameter is secured or not.
Key	In this field, enter the Key (Name) for the Environment Parameter. **Note:** If an Environment Parameter and a Machine Parameter have the same Key, the Environment Parameter takes precedence.
Value	In this field, enter the value(s) for the new Environment Parameter. The following possibilities apply: • Enter the fixed value, if you are creating a non-editable Environment Parameter. • Enter the default value, if you are creating an editable Environment Parameter. • Enter the list of possible preset values, separated by a semicolon (;), if you are creating a dynamic Environment Parameter.
Repeat Value	Required field for secured Environment Parameters: repeat the secured value.
Description	In this field, enter a description for the Parameter.
Mandatory	Select the *Yes* option button, if the new Environment Parameter must be defined as mandatory. When you create a Level Request for this Environment, the mandatory Environment Parameters will always be provided to the Script. Select the *No* option button, if the new Environment Parameter should not be defined as mandatory. When you create a Level Request for this Environment, you can decide whether you want to provide the non-mandatory Environment Parameter to the Script.
Editable	Select the *Yes* option button, if the new Environment Parameter must be defined as editable. When you create a Level Request for this Environment, you can accept the default value (the one you enter in the Value field during creation) or define a value yourself for this Environment Parameter. Select the *No* option button, if the new Environment Parameter should not be defined as editable. When you create a Level Request for this Environment, only the preset value (the one you enter in the Value field during creation) for this Environment Parameter can be offered to the Script. This field is not provided for secured Environment Parameters.
Dynamic	Select the *Yes* option button, if the new Environment Parameter must be defined as dynamic. When you create a Level Request for this Environment, you can select one of the predefined values from the drop-down list. These are the values you enter in the Value field during creation and which you separate by a semicolon (;). The selected value will be offered to the Script. Select the *No* option button, if the new Environment Parameter should not be defined as dynamic. This field is not provided for secured Environment Parameters.

4. Click *Create* to confirm the creation of the Environment Parameter.
   You can also click:
   - *Reset* to clear the fields and restore the initial values.
   - *Cancel* to return to the previous screen without saving the changes.

21.3. Viewing the Environment Parameters History

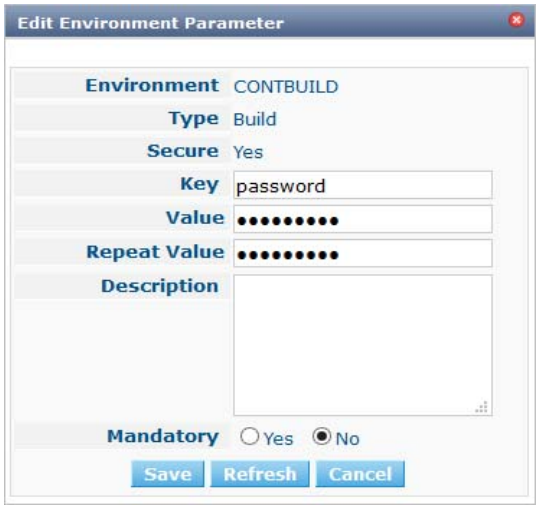
1. Switch to the *Parameters Overview* screen for the required Project.
   See The Parameters Overview Screen on page 241.

2. Click the *History* link on the *Parameters Overview* panel to display the *Environment History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the previous screen.

21.4. Editing Environment Parameters

1. Switch to the *Parameters Overview* screen for the required Project.
   See The Parameters Overview Screen on page 241.

2. In the *Actions* column, click the *Edit* link in front of the Environment Parameter to be edited.
   The following window is displayed:



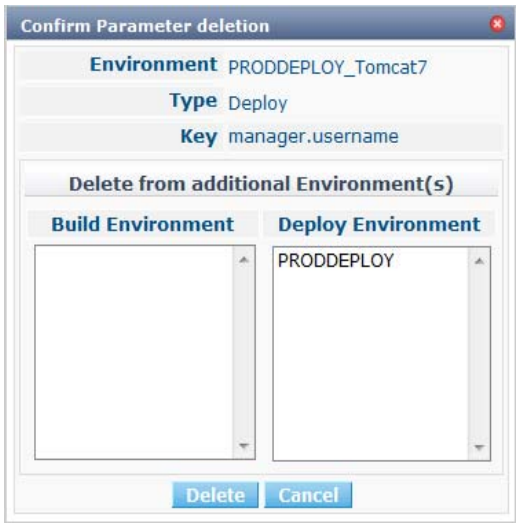
   For a description of the fields, refer to Creating Environment Parameters (page 242).

3. Edit the fields as required, and click *Save* to save your changes.
   You can also click:
   - *Refresh* to clear the fields and restore the initial values.
   - *Cancel* to return to the previous screen without saving your changes.

21.5. Deleting Environment Parameters

1. Switch to the *Parameters Overview* screen for the required Project.
   See The Parameters Overview Screen on page 241.

2. In the *Actions* column, click the *Delete* link in front of the Environment Parameter to be deleted.
   The following confirmation window is displayed:



3. Optionally, select additional Environments. This allows for deleting parameters with the same key name on the selected Build and Deploy Environments.

4. Click *Delete* to confirm the deletion of the Environment Parameter.
   You can also click *Cancel* to return to the previous screen without saving your changes.

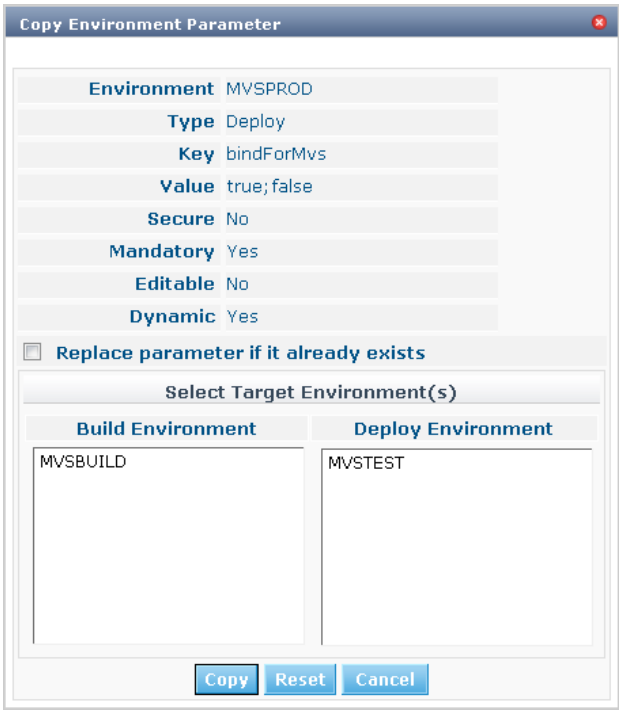
21.6. Copying Environment Parameters

To avoid having to redefine Environment Parameters which are identical for multiple Environments, you can copy the required Environment Parameter to other Build and/or Deploy Environments.

1. Switch to the *Parameters Overview* screen for the required Project.
   See The Parameters Overview Screen on page 241.

2. In the *Actions* column, click the *Copy* link in front of the Environment Parameter to be copied.

The following window displays the values of the parameter you are about to copy.



3. Indicate whether you want to replace the parameter in case it already exists in the target Build or Deploy Environment(s).

4. Select the Target Environment(s)

5. Click *Copy* to confirm copying the Environment Parameter.
   You can also click:
   - *Reset* to clear the fields and restore the initial values.
   - *Cancel* to return to the previous screen without saving the changes.

CHAPTER 22
Auditing Projects

When the user requests the audit of a Project, IKAN ALM performs the following checks:

- The physical Build Archive locations must exist for all Project Streams.
  - For the Head Stream, this location matches the following path: *SystemSettingsBuildArchive/ProjectName/StreamBuildPrefix*.
  - For the Branch Streams, this location matches the following path: *SystemSettingsBuildArchive/ProjectName/StreamBuildPrefix.StreamBuildSuffix*.

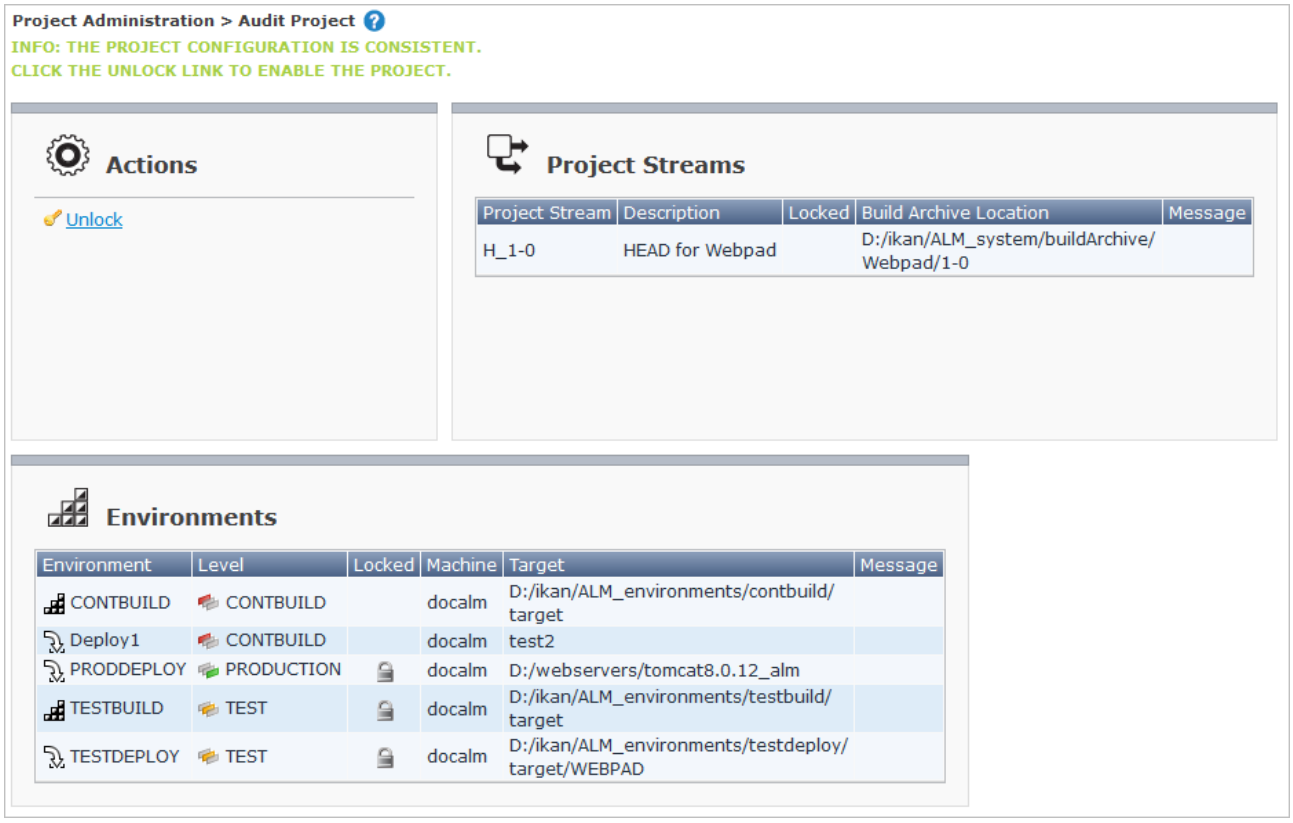
  IKAN ALM blocks the Project Streams for which this test fails.
- There is exactly one Build Level defined for each Lifecycle. IKAN ALM blocks the Project Streams for which this test fails.
- There is exactly one Build Environment defined for the Build Level. IKAN ALM blocks the Levels for which this test fails.
- If no Build Script is defined in the Project, every Build Environment must have a Build Script defined. IKAN ALM blocks the Levels for which this test fails.
- If no Deploy Script is defined in the Project, every Deploy Environment must have a Deploy Script defined. IKAN ALM blocks the Levels for which this test fails.
- If a Custom Phase is inserted in a Level, IKAN ALM verifies if a value is specified for the mandatory parameter 'alm.phase.builder' of the Custom Phase. IKAN ALM blocks the Levels for which this test fails.

The following tests will return a warning:

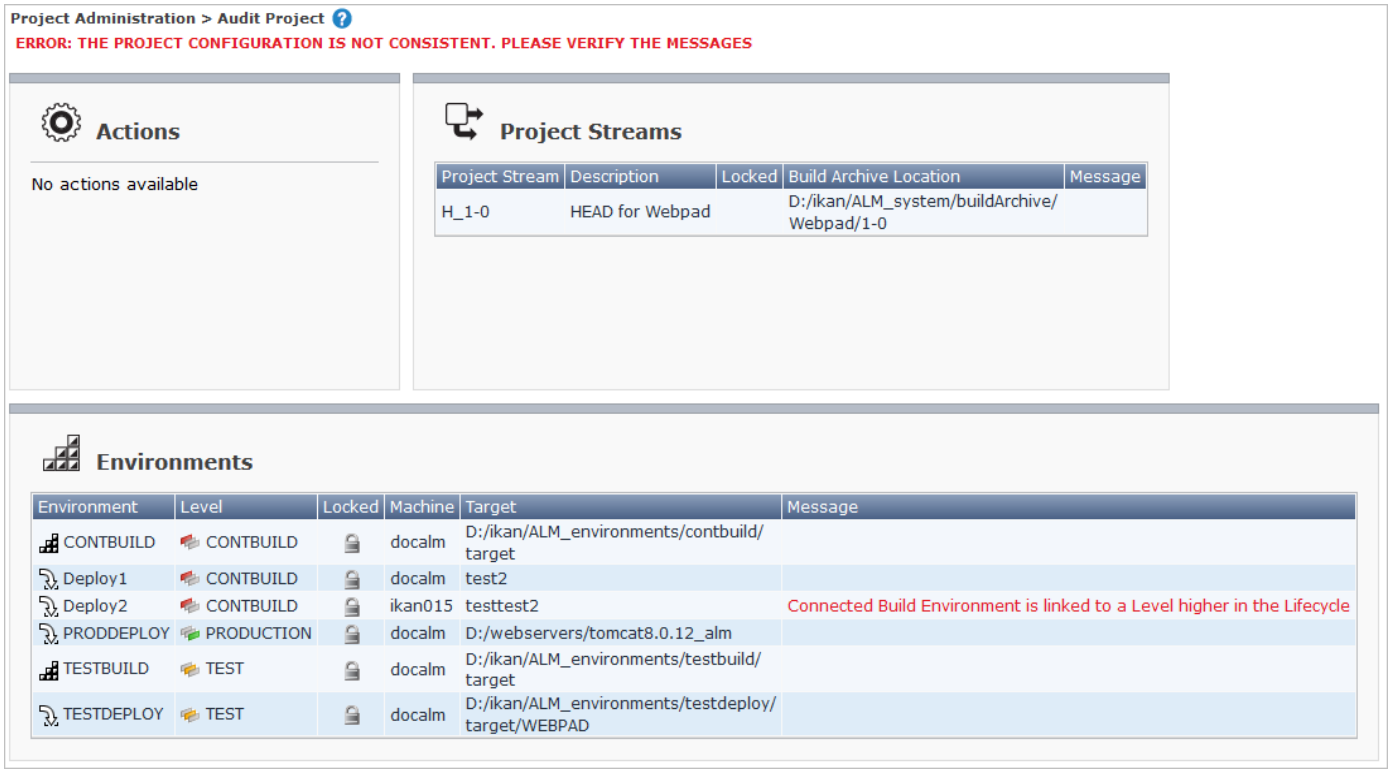
- There is at least one Build or Deploy Environment defined for each Test Level. Since version 5.0, IKAN ALM allows Test levels without Build or Deploy Environment, which can be seen as a sort of “Milestone” Test Levels.
- There is at least one Deploy Environment defined for each Production Level. Since version 5.0, IKAN ALM allows Production levels without Build or Deploy Environment, which can be seen as a sort of “Milestone” Production Levels.

22.1. Auditing a Project

1. In the *Project Administration context*, select *Audit Project*.
   If the Project configuration is consistent, the following screen is displayed:



If there are errors in the Project configuration, the following screen is displayed:



Locked Project Streams and Levels have a *Locked* icon in the *Locked* field. In the sample screen above, this is the case for the *TestPartialBuild* Level. Error or warning messages are displayed in red in the *Message* field.

For the list of possible error and warning messages, refer to the section Error and Warning Messages (page 249).

2. Verify if all Project Streams and Levels are properly defined and accessible for IKAN ALM, and correct the errors if required.

3. If the Project Configuration is consistent, click the *Unlock* button in the *Actions* panel to unlock all locked Project Streams and Levels of the Project.

22.2. Error and Warning Messages

The following sections list the possible error and warning messages when auditing a Project.

- Error Messages (page 249)
- Warning Messages (page 250)

Error Messages

Error Message	Solution
`No Lifecycle linked`	Error message on a Project Stream: a Project Stream must be linked to a Lifecycle.
`No Build Level linked to the Lifecycle of the Project Stream`	Error message on a Project Stream: its Lifecycle must have one Build Level.
`Build Archive Location not found`	Error message on a Project Stream: there is no Build Archive location under the indicated path and IKAN ALM cannot create it. During the audit, IKAN ALM verifies if there is a Build Archive location for the Project Stream under the System Settings Build Archive Location. If it is absent, it tries to create a Build Archive location for the Project Stream, but this process did not succeed, e.g., caused by security problems. Contact the IKAN ALM administrator, who can verify the log messages to find the cause of this problem.
`No Build Environment found`	Error message on Build Level: a Build Level must be linked to exactly one Build Environment.
`More than one Build Environment found`	Error message on a Build Level: a Build Level must be linked to exactly one Build Environment.
`Please Specify a unique Build Suffix for each Build Environment`	Error message on a Test or Production Level with more than one Build Environment. In such a case, each Build Environment must have a unique Build Suffix in order to not overwrite Build result files in the Build Archive.

Error Message	Solution
`No Build Script specified`	Error on a Build Environment: there is no Build script specified for the Build Environment. A Build script may be specified in the Project definition, or be overwritten in the Build Environment definition.
`Build tool is inconsistent with project build tool type`	Error message on a Build Environment. The Build Tool connected to the Build Environment is not of the Build Tool type that was specified in the Project Definition. For example: a Project definition with Ant as Build Tool type has a Build Environment that is linked to a Maven2 Scripting Tool definition.
`Connected Build Environment is not linked to a Level in the Lifecycle`	Error message on a Deploy Environment: the Build Environment that is linked to the Deploy Environment is not in the Lifecycle of the Level containing the Deploy Environment. The reason is that the Level that contains the Build Environment is not connected to this Lifecycle.
`Connected Build Environment is linked to a Level higher in the Lifecycle`	Error message on a Deploy Environment: its connected Build Environment is part of a Level that, in the Lifecycle, comes after the Level containing the Deploy Environment. This makes Deploys impossible, since the Build Result must have been created earlier or at the same point in the Lifecycle.
`Connected Build Environment is linked to an optional Level lower in the Lifecycle`	Error message on a Deploy Environment: the linked Build Environment may not be part of an optional Level. Otherwise, it would be possible to skip the optional Level and deploy a Build Result that was not created.
`Deploy tool is inconsistent with project deploy tool type`	Error message on a Deploy Environment. The Deploy Tool connected to the Deploy Environment is not of the Deploy Tool type that was specified on the Project Definition. For example: a Project definition with NAnt as Deploy Tool Type has a Deploy Environment that is linked to an Ant Scripting Tool definition.
`Empty mandatory parameter 'alm.phase.builder' in Phase 'Phase Display Name'`	Error message on a Level. A Custom Phase with the given Display Name is inserted in the Level, but the value of the (automatically) created alm.phase.builder Parameter of this Phase is empty. Navigate to the *Level Phases Overview* of the Level, and click the *View Parameters* link next to the indicated Phase. Here you may set a value for the alm.phase.builder parameter by clicking the *Edit* link (See The Phase Parameters Overview Screen on page 434.).

Warning Messages

Warning Message	Solution
`No Build or Deploy Environment found`	Warning message on a Test Level. As from the release of IKAN ALM 5.0, it is allowed for Test Levels to have no Build nor Deploy Environment connected. Such a "no operation" Test Level can be considered as a sort of a milestone acquired in the Lifecycle.

Warning Message	Solution
`No Deploy Environment found`	Warning message on a Production Level. Since IKAN ALM 5.0 it is allowed for Production Levels to have no Deploy Environment connected. Such a "no operation" Level can be considered as a sort of a milestone acquired in the Lifecycle.

Global Administration

CHAPTER 23

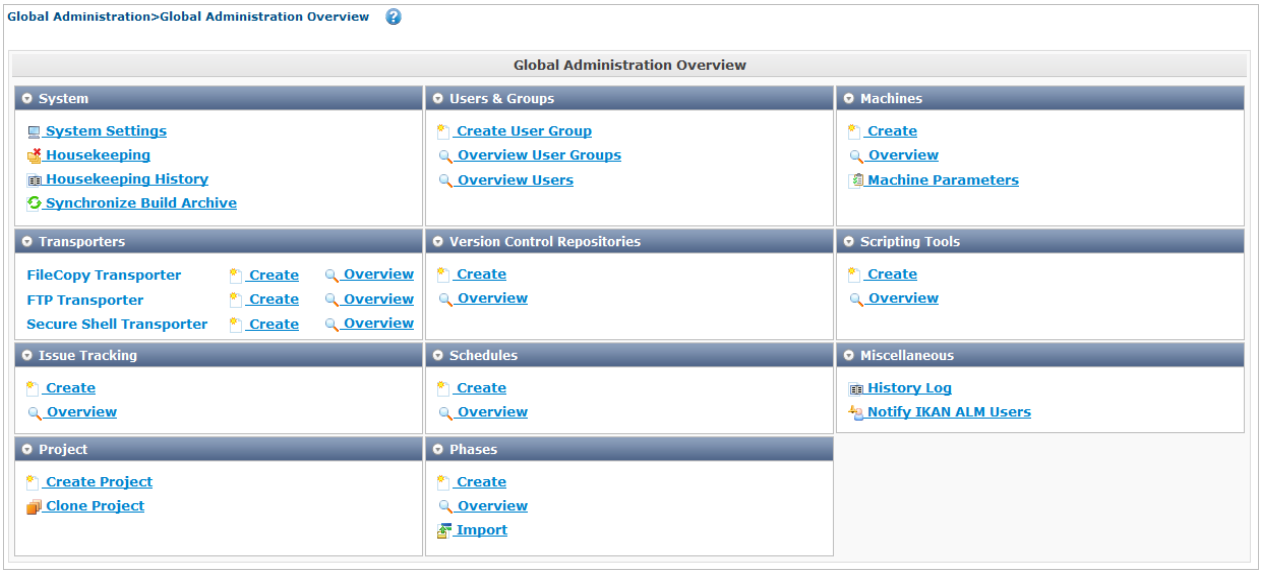
Global Administration

On the Global Administration landing page, you will find an overview of all tasks which can also be called via the Global Administration menu.

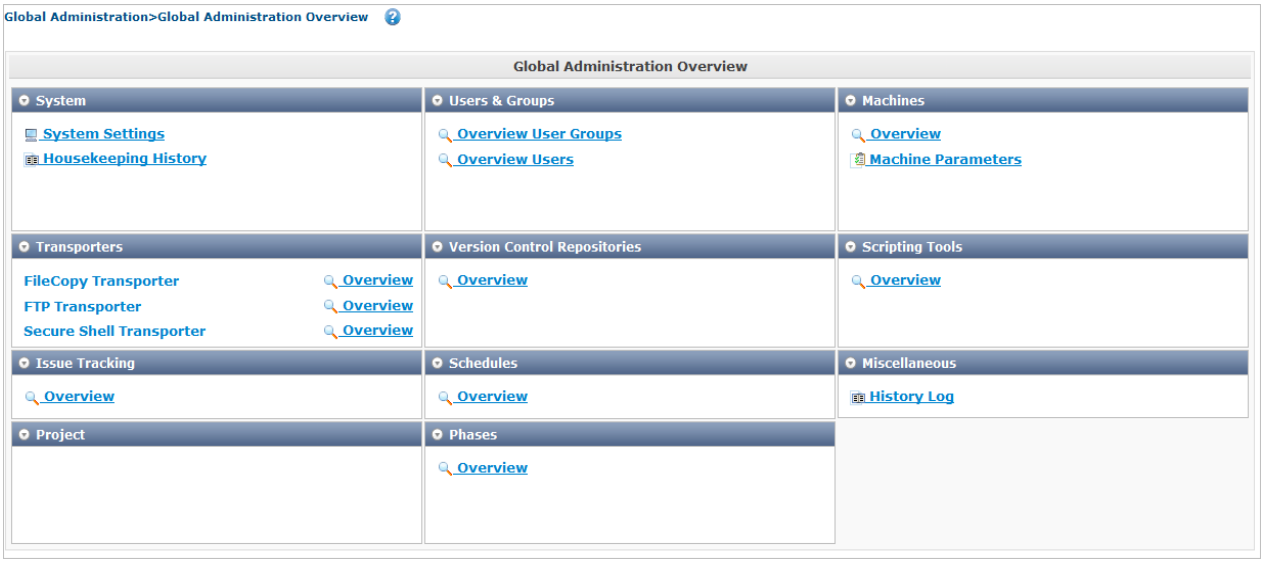
1. On the Main Menu, click the *Global Administration* icon.

   In the remainder of this document, we will refer to this section of the interface as the "Global Administration context".

2. Users with Global Admin Access Rights can perform the following tasks:



3. For Users without Global Admin Access Rights, the number of available tasks on the landing pages will be limited.



Note: Each of the sections on the landing page corresponds with a Submenu item under Global Administration, except for the *History Log* which has been added to the *Miscellaneous* section on the landing page.

Refer to the following sections for detailed information:

- System (page 255)
- History Log (page 271)
- Users (page 275)
- User Groups (page 280)
- Machines (page 289)
- Transporters (page 307)
- Version Control Repositories (page 330)
- Scripting Tools (page 394)
- Issue Tracking (page 381)
- Schedules (page 452)
- Miscellaneous (page 472)
- Phases (page 424)
- Projects (page 461)

CHAPTER 24
System

The *System* menu provides an entry to view or edit the IKAN ALM Server configuration settings and to perform Housekeeping Actions in the IKAN ALM Build Archive.

Users which have Global Admin Access rights will have access to the following Submenus:

- System Settings (page 255)
- Housekeeping (page 262)
- Housekeeping History (page 265)
- Synchronize Build Archive (page 268)

Users who do not have Global Admin Access rights will only have access to the following Submenus:

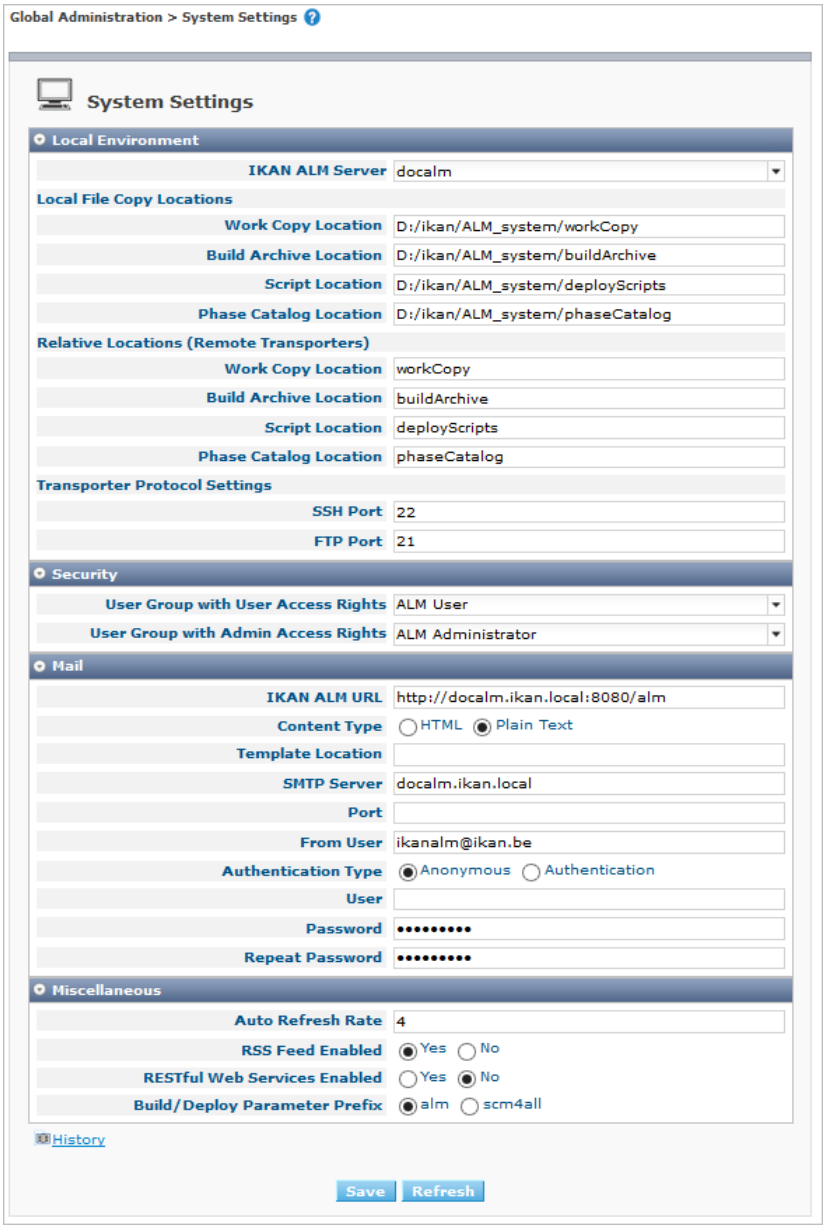
- System Settings (page 255)
- Housekeeping History (page 265)

24.1. System Settings

On the *System Settings* screen, the global IKAN ALM Server settings are defined, such as the different locations (build archive, script, work copy and phase catalog) and the mail and security settings. Some of these fields are filled in automatically during the installation process.

Note: Changing some of these settings requires preparatory actions on the IKAN ALM server.

1. In the Global Administration context, select *System > System Settings*.
   The following screen is displayed.



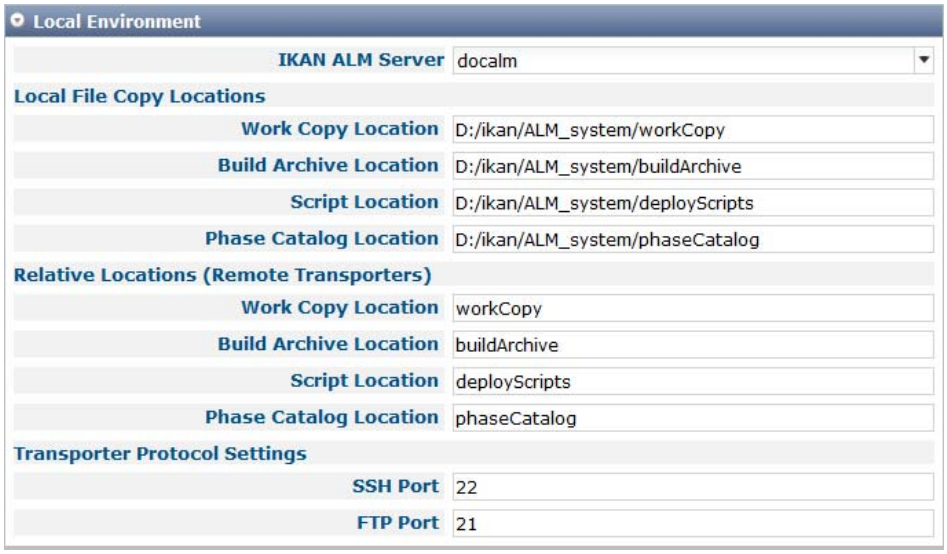
Note: If you do not have Global Administrator access rights in IKAN ALM, you will not be able to edit the fields.

The following subpanels are available:

- Local Environment (page 257)
- Security (page 259)
- Mail (page 259)
- Miscellaneous (page 261)

Underneath the *Edit System Settings* panel, you will find the *History* link. Click this link to display the *System Settings History View* screen. For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

Local Environment

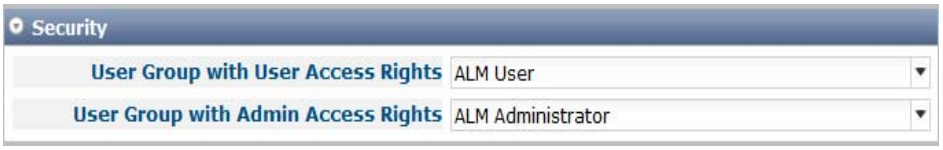


The following fields are available on the *Local Environment* subpanel:

Field	Description
IKAN ALM Server	This field contains the name of the Machine currently hosting the IKAN ALM Server. Select another Machine Name from the drop-down list, if required. Changing this field requires preliminary actions, like installing the IKAN ALM Server on the new target Machine, migrating Local File Copy Locations and setting up the necessary VCR clients.
Local File Copy Locations	These fields are used for Build and Deploy actions. These locations are automatically synchronized with the built-in Local FileCopy transporter. See FileCopy Transporters on page 310.
Work Copy Location	This field contains the path to the Work Copy Location. In preparation of the Build process, the IKAN ALM Monitor places the sources (checked out sources from a VCR and sometimes also dependent build results) in a subdirectory of this location. The tagging of the VCR after a successful Build is done from this location as well. You can edit the location if required.
Build Archive Location	This field contains the path to the location of the Build Archive on the IKAN ALM Server. Build Results will be stored in or retrieved from this location. They will be placed in a subdirectory per Project Stream and per Project. If required, you can edit the location.
Script Location	This location can be used to store build and deploy scripts. When the build or deploy script is not available in the sources or build result, it will be retrieved from this location. You can edit the location if required. This field allows that the Scripts are stored outside the Version Control Repository and managed on a (secured) central location.
Phase Catalog Location	This field contains the path to the location of the Phase Catalog on the IKAN ALM Server. Newly created Phases and Phases that have been imported will be stored in this location.

Field	Description
Relative Locations (Remote Transporters)	These locations are used for all remote transport actions (SSH, FTP and FileCopy). These locations are relative. The complete path will be assembled as follows: the prefix path (as specified in the SSH, FTP or FileCopy Transporter definition) concatenated with the relative location defined on this *Local Environment* tab. Remote Transporters are activated by defining them as the Transporter Protocol when specifying the settings for a Machine. For more information, refer to the sections Transporters (page 307)and Machines (page 289).
Work Copy Location	This field contains the location of the Work Copy displayed as a remote location. This is the location where the VCR interface places the checked out sources or dependent build results on the IKAN ALM Server, so that they are accessible for a remote IKAN ALM Agent handling a build process. You can edit the location if required.
Build Archive Location	This field contains the location of the Build Archive displayed as remote location. This is the location where the Builds are stored in or retrieved from by a remote IKAN ALM Agent handling a build or deploy process. You can edit the location if required.
Script Location	This field contains the location of the Build or Deploy Script displayed as a remote location. Scripts will be retrieved from this location if they are not stored in the Version Control Repository. You can edit the location if required. This field allows that the Scripts are stored outside the Version Control Repository and managed on a (secured) central location.
Phase Catalog Location	This field contains the location of the Phase Catalog displayed as remote location. Newly created Phases and Phases that have been imported will be stored in this location. A remote IKAN ALM Agent that needs to install a Phase will use this location to retrieve it.
Transporter Protocol Settings	The Transporter Protocol defines how Sources and Build Results will be transported.
SSH Port	Specify the SSH Port Number. This is only necessary if the SSH Server on the IKAN ALM Server uses an SSH Port other than the default port number *22*. An SSH Server daemon must be installed on the IKAN ALM server in order to use SSH as a Transporter Protocol. **Note:** This port number may be overwritten by the port number defined in the *Transporter* definition. See Creating a Secure Shell Transporter on page 323.
FTP Port	Specify the FTP Port Number. This is only necessary if the FTP Server on the IKAN ALM Server uses a TCP Port other than the default port number *21*. An FTP server must be installed on the IKAN ALM Server in order to use FTP as a Transporter protocol. **Note:** This port number may be overwritten by the port number defined in the *Transporter* definition. See Creating an FTP Transporter on page 316.

Security



The following fields are available on the *Security* subpanel.

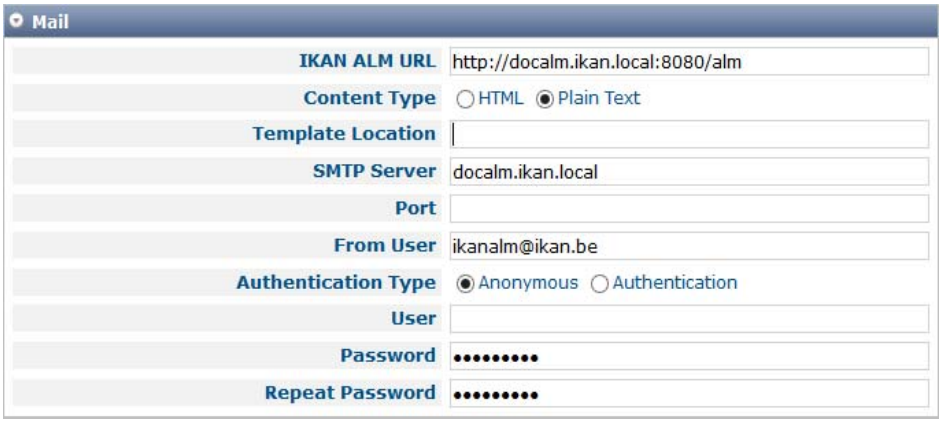
Field	Description
User Group with User Access Rights	This field contains the name of the User Group to which all Users requiring access to IKAN ALM must belong. You may select another User Group from the drop-down list, if required. You may also leave this field empty. In this case, all authenticated users will have User access rights to IKAN ALM.
User Group with Admin Access Rights	This field contains the name of the User Group to which all Users requiring Administrator Rights must belong. Note that this must be an **external** User Group. Only users belonging to this User Group may create, edit and delete Global Administration definitions in IKAN ALM. You may select another **external** User Group from the drop-down list, if required. You may also leave this field empty. In this case, all authenticated users will have Administrator access rights to IKAN ALM.

Mail

IKAN ALM can send mails automatically or on demand to notify on failed or successful Builds and Deploys, to request an approval before delivering to a Test or Production Level, or just to notify connected users on changed settings.

Settings for connecting to an external mail system can also be defined on the Mail tab.

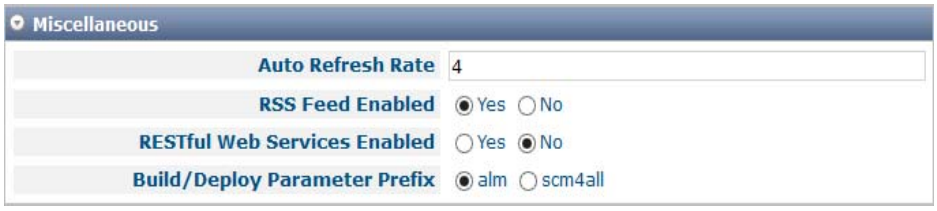
Note: Theses Mail settings can be tested via the *Notify IKAN ALM Users* submenu option. See Notifying IKAN ALM Users on page 472.



The following fields are available on the *Mail* subpanel.

Option	Meaning
IKAN ALM URL	In this field, enter the base IKAN ALM URL. This URL is used for creating hyperlinks in mails sent by IKAN ALM (Approvals, Pre-Notifications, Post-Notifications, Level Request Success/Fail Notifications) and for constructing the RSS feed link.
Content Type	Indicates which templates will be used to compose notification and approval mail messages. There are two options: • *HTML:* the templates containing HTML tags will be used. • *Plain Text*: the templates without HTML tags will be used.
Template Location	By default, the mail templates are located in the following directory: *TOMCAT_HOME/webapps/alm/WEB-INF/classes/templates* If you want to customize these templates, we advise you to change this location to a local directory on the IKAN ALM Server. The locale directories match the language settings of the user: • *en* for English • *fr* for French • *de* for German If no match is found, the templates in the *default* directory will be used. For the HTML formatted templates, a *_html* suffix is added before the *.vm* extension. For plain text templates, there is no such suffix.
SMTP Server	Select the Machine name of the SMTP Mail Server.
Port	This field contains the SMTP Port number used by IKAN ALM to connect to the Mail Server. This field is optional, by default port 25 is used.
From User	This field contains the *From:* E-Mail Address used in IKAN ALM notification e-mails. This field is mandatory if SMTP was chosen as mail Protocol.
Authentication Type	Select which Authentication Type is required: • *Anonymous*: select this option, if the SMTP Server can be used without authentication. • *Authentication*: select this option, if the SMTP Server requires authentication.
User	This field contains the User name used to authenticate on the SMTP Server. This field is mandatory if *Authentication* was chosen as Authentication Type.
Password	This field contains the masked Password used to authenticate on the SMTP Server. This field is mandatory if *Authentication* was chosen as Authentication Type. The characters you enter are displayed as asterisks.
Repeat Password	In this field, re-enter the password used to authenticate on the SMTP Server.

Miscellaneous



The following fields are available on the *Miscellanous* subpanel:

Field	Meaning
Auto Refresh Rate	In this field, you can enter the number of seconds for the automatic refresh function on the Desktop Overview (page 136), Level Request Overview (page 58) and Build History Screen (page 104). As a result, these overview screens will be refreshed each time the interval expires (if the user has activated the Auto Refresh function on his or her Desktop submenu (page 12)). If you do not want to allow the use of the Auto Refresh option, enter *0* (zero) in this field.
RSS Feed Enabled	In this field you can enable the RSS functionality. As a result an RSS button will become available on the user's *Level Requests Overview* panel. By default the RSS Feeds are enabled. For more information, refer to the section The IKAN ALM RSS Functionality (page 62).
RESTful Web Services Enabled	This feature is currently only functional for the integration with the SAP lifecycle. These services allow for querying (HEAD and GET requests) Users, User Groups, Subversion repositories and Issue Tracking Systems in Global Administration, and Projects, Project Streams and Lifecycles in Project Administration. By default, this feature is disabled for security reasons.
Build/Deploy Parameter Prefix	This option determines the value of the IKAN ALM Build/Deploy parameters prefix (by default alm) and the name of the ANT property file that is generated for Execute Build and Execute Deploy phases (by default alm_ant.properties). In case you are upgrading from a version prior to IKAN ALM 5.0, this feature allows you to keep on working with the old scm4all parameter prefix and the old scm4all_ant.properties file, and not having to modify your Build and/or Deploy scripts. **Note:** This is a compatibility solution. At a certain point in time, you will have to modify your scripts and use the new *alm* prefix.

1. Make the necessary changes on the different panels.

2. Click *Save* at the bottom of the screen.
   You may also click *Refresh* to retrieve the settings from the database.

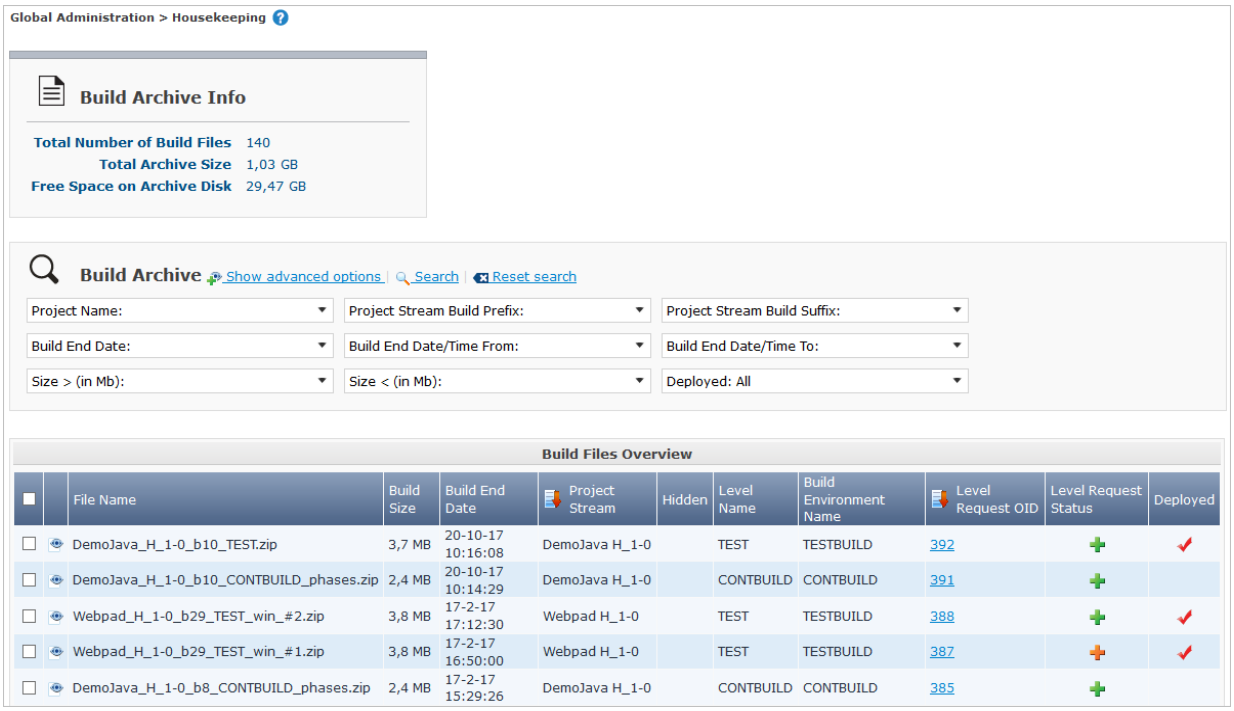
24.2. Housekeeping

The Housekeeping functionality allows you to manage the Build Archive. A search can be performed locating Build Files that may no longer be needed. These Build Files can be selected and marked for removal. The system will then physically remove them from the Build Archive and set the Archive Status of the related Builds to "deleted", hereby avoiding these Builds to be used in further Deploys.

The *Housekeeping* screen also shows information about the Build Archive, like the total number of build files in the Archive, the size of the Archive and the remaining free space on the hard drive where the Archive is located.

1. In the Global Administration context, select *System > Housekeeping*.

   The *Housekeeping* screen is displayed:



On top of the screen, the *Build Archive Info* panel displays the following information.

Note: The *Build Archive Info* is based on hardware data obtained from the Build Archive location on the IKAN ALM Server. This Build Archive location is defined in the *Local Environment* tab of the System Settings. See Local Environment on page 257.

Field	Description
Total Number of Build Files	This field contains the total number of Build Files in the Build Archive.
Total Archive Size	This field indicates the total size (in a multiple of bytes) of the Build Archive.
Free Space on Archive Disk	This field indicates the remaining free space (in a multiple of bytes) on the hard drive where the Archive is located.

2. Define the required search criteria on the search panel.

   The list of items on the overview will be automatically updated based on the selected criteria.

You can also:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- click the *Search* link to refresh the list based on the current search criteria,
- click the *Reset search* link to clear the search fields.

You can enter or select several search criteria to narrow your search.

Criterion	Description
Project Name	Enter or select the name of the Project for which you want to display the Build Files.
Project Stream Build Prefix	Enter or select the Project Stream Build Prefix.
Project Stream Build Suffix	Enter or select the Project Stream Build Suffix.
Build End Date	Specify the minimum age of the Build Files. The possible values are: • *None Specified* No age specified, and the Build End Date/Time From and To fields will be ignored • *Older than 3 years* Only Build files older than 3 years will be displayed • *Older than 1 year* Only Build files older than 1 year will be displayed • *Older than 3 months* Only Build files older than 3 months will be displayed
Build End Date/Time From	If you want to specify a range of dates, enter in this field the first *Build End Date/Time* of the range. You can also click the icon to select the Build End Date/Time. Click the required date to copy it into the field. The time will be set to the current time. However you can still change the time manually. This field will be ignored when a *Build End Date* field has been selected.
Build End Date/Time To	If you want to specify a range of dates, enter in this field the last *Build End Date/Time* of the range. You can also click the icon to select the Build End Date/Time. Click the required date to copy it into the field. The time will be set to the current time. However you can still change the time manually. This field will be ignored when a *Build End Date* field has been selected.
Size > (in Mb)	Use this field to specify the minimum size of the Build Files to be displayed on the *Build Files Overview*.
Size < (in Mb)	Use this field to specify the maximum size of the Build Files to be displayed on the *Build Files Overview*.
Deployed	Select *Yes* to display only the Build Files that have been deployed. Select *No* to display only the Build Files that have not been deployed. If the option *All* is selected, both the deployed and undeployed Build Files will be displayed.

Criterion	Description
Level Request Status	Specify the status of the Level Request that created the Build Files. Only Build Files that were created by a Level Request that currently has this status are displayed. The possible values are: • Success • Warning • Fail • Run • Awaiting Post-approval • Rejected • Aborted • Aborting
Level Name	Enter or select the name of the Level for which you want to display the Build Files.
Level Type	Enter the Level Type (Build, Test or Production).
Build Environment Name	Enter or select the name of the Build Environment for which you want to display the Build Files.
Show Hidden Project Streams	Indicate whether you want to display the hidden Project Streams or not. • *Yes* Hidden Project Streams will be displayed. • *No* Hidden Project Streams will not be displayed. • *All* (= default) All Project Streams, hidden and unhidden, will be displayed.

3. Verify the search result on the *Build Files Overview*.

The *Build Files Overview* panel lists the following information for each displayed Build File.

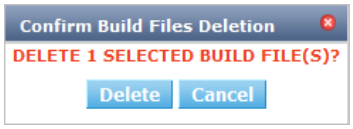
Field	Description
View Content	Click the *View* link to view the content of the Build File. See Results on page 89.
File Name	The name of the Build File.
Build Size	The size of the Build File.
Build End Date	The date and time at which the Build execution ended.
Project Stream	The name of the Project Stream.
Hidden	Indication whether the Project Stream is hidden or not.
Level Name	The name of the Level.
Build Environment Name	The name of the Build Environment.
Level Request OID	This field displays the Level Request OID of the Build. Click the link to display the details for this Level Request. See Level Request Detail on page 65.

Field	Description
Level Request Status	This field contains the Level Request Status. See The Level Requests Overview Screen on page 58.
Deployed	This field indicates whether or not the Build has been used by a Deploy.

4. Select the Build Files to be deleted and click the *Delete* button underneath the overview.

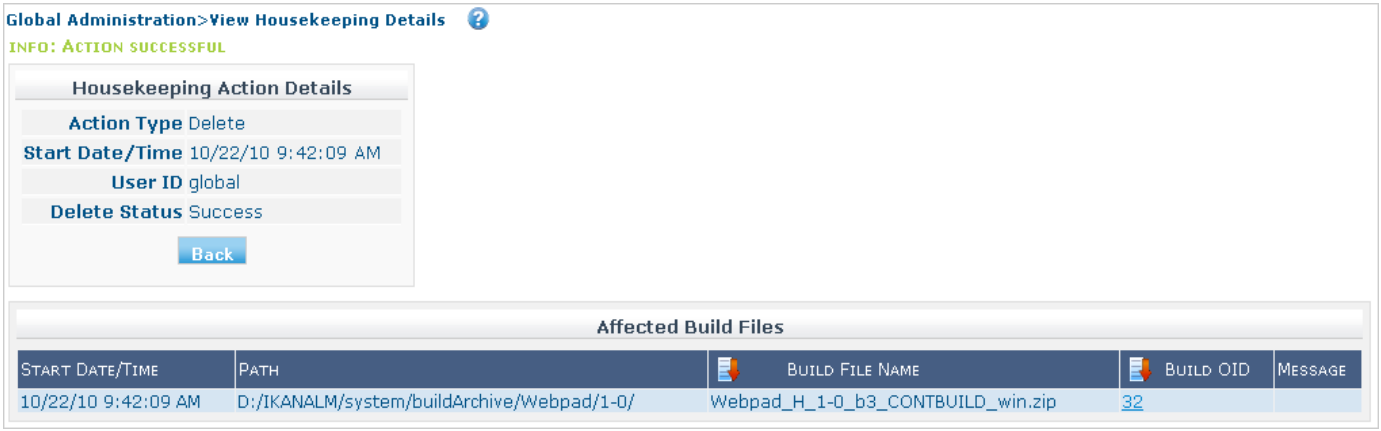
   Select the check box in the table heading in order to select all the displayed Build Files.

5. The *Confirm Build Files Deletion* popup window is displayed.



6. Click *Delete* to confirm the deletion or *Back* to return to the previous screen without deleting the Build Files.

   When the delete is confirmed, the removal is executed. The action is logged in the *Housekeeping History* and the User is redirected to the *View Housekeeping Details* screen where the result of the action is displayed.



24.3. Housekeeping History

This functionality allows you to search for previously performed delete actions.

The *Housekeeping History* screen displays the history of Housekeeping Actions that have been performed.

It provides a *Search* panel where the User can enter search criteria to restrict the list of displayed History Actions.

1. In the Global Administration context, select *System > Housekeeping History*.

   The *Housekeeping History* screen is displayed:

2. Define the required search criteria on the search panel.

   The list of items on the overview will be automatically updated based on the selected criteria.

   You can also:

   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

   You can enter or select several search criteria to narrow your search.

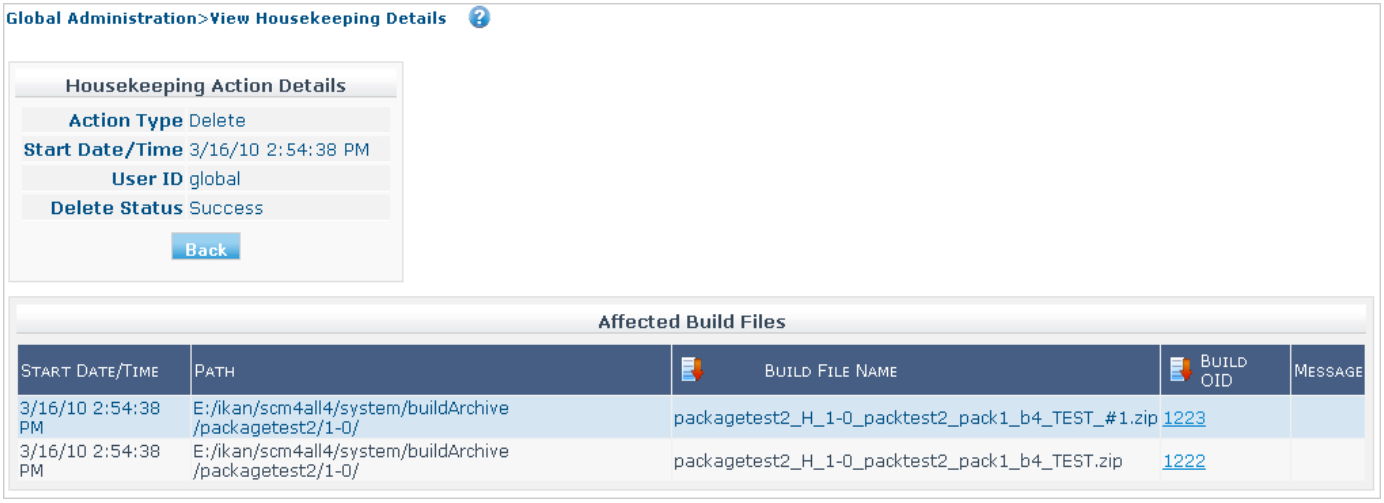
Criterion	Description
User ID	Enter or select the ID of the User whose Housekeeping Actions you want to display.
Delete Status	This field indicates the Delete status
Start From	If you want to specify a range of dates, enter in this field the start *Date/Time* of the range. You can also click the icon to select the Date/Time. Click the required date to copy it into the field. The time will be set to *0:0* (midnight). However you can still change the time manually.
Start To	If you want to specify a range of dates, enter in this field the end *Date/Time* of the range. You can also click the icon to select the Date/Time. Click the required date to copy it into the field. The time will be set to *0:0* (midnight). However you can still change the execution time manually.

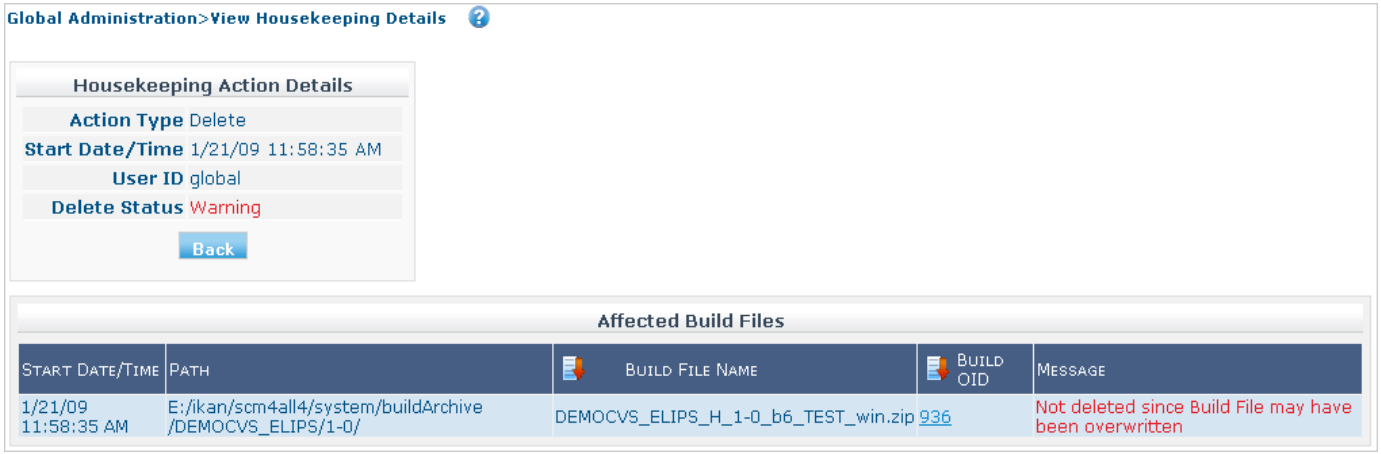
3. Verify the search result on the *Housekeeping Actions Overview*.

   The *Housekeeping Actions Overview* panel lists the following information for each displayed Housekeeping Action.

Field	Description
Action Type	This field contains the type of the Housekeeping Action.
Start Date/Time	This field indicates the date and time when the Housekeeping Action started.
User ID	This field displays the ID of the User who performed the Housekeeping Action.
Delete Status	This field indicates the Delete status. Possible statuses are: • Success • Warning • Error

4. You can click the *View* link to display the *View Housekeeping Details* screen.



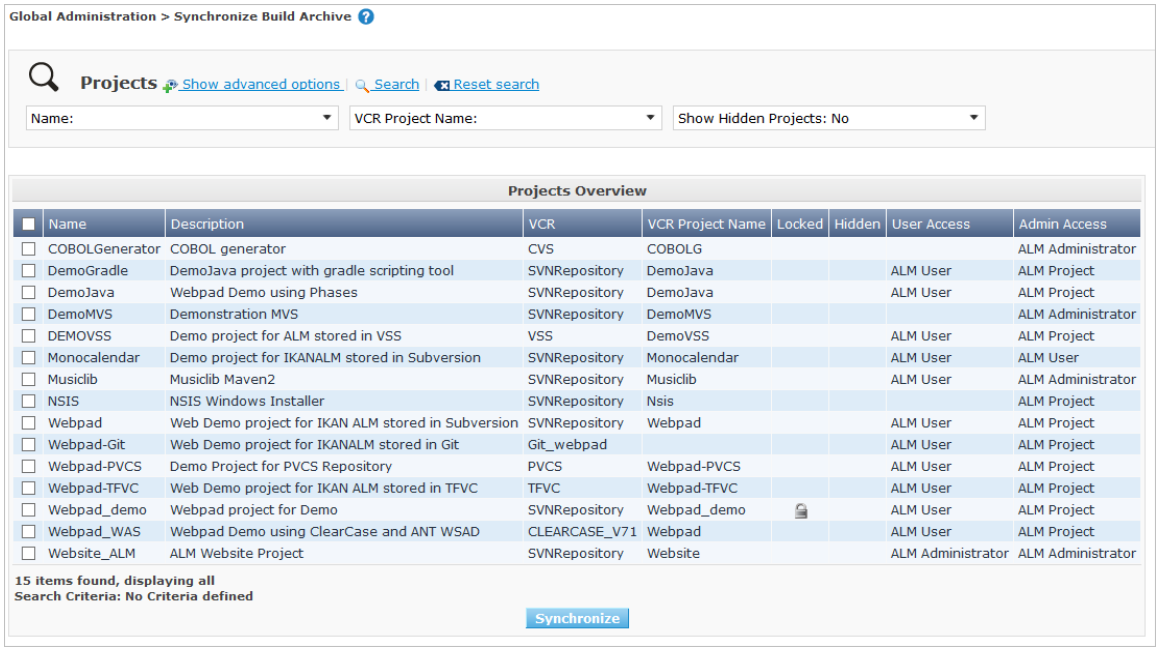


24.4. Synchronize Build Archive

The *Synchronize Build Archive* screen allows an IKAN ALM Administrator to synchronize the physical representation of the Build Archive on the file system with the logical representation of the Build Archive in the IKAN ALM database.

For example, a Build File that was manually deleted on the file system (using the OS delete command) will still have its Archive Status set to "Present" in the IKAN ALM database. During synchronization the status will be set to "Non existing".

1. In the Global Administration context, select *System > Synchronize Build Archive*.
   The *Synchronize Build Archive* screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

   You can enter or select several search criteria to narrow your search.

Criterion	Description
Name	Specify the name of the Project.
VCR Project Name	Specify the VCR Project Name.
Show Hidden Projects	Indicate whether you want to display the hidden Projects or not. • *Yes* Hidden Projects will be displayed. • *No* (= default) Hidden Projects will not be displayed. • *All* All Projects, hidden and unhidden, will be displayed.

Criterion	Description
Description	Specify the description of the Project.
VCR	Select the VCR from the drop-down list.
Locked	Specify whether the Project is locked or not.

3. Verify the search result on the *Projects Overview*.

The *Projects Overview* panel lists the following information for each displayed Build File.

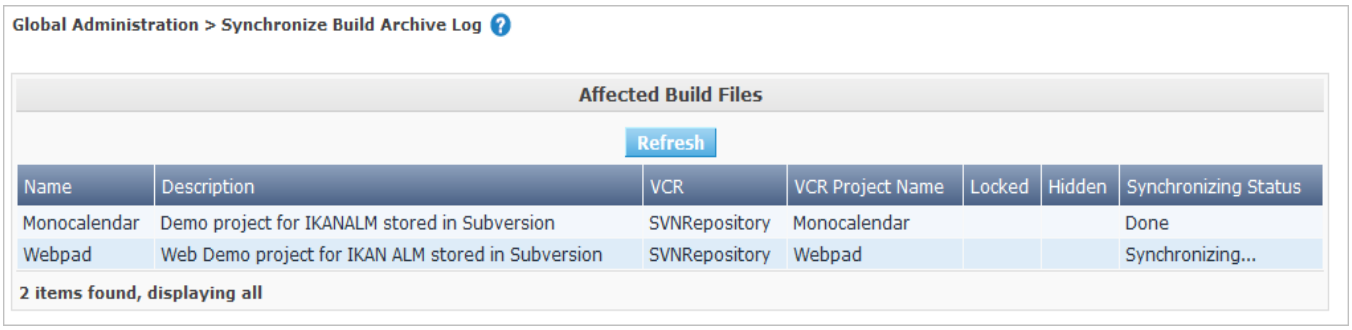
Field	Description
Name	This field contains the Project Name.
Description	This field contains the Project Description.
VCR	This field contains the VCR used for the Project.
VCR Project Name	This field contains the VCR Project Name.
Locked	This field indicates whether or not the Project is locked: • If it is locked, this field contains the locked icon ( ). • If it is not locked, this field remains empty.
Hidden	This field indicates whether the Project is hidden or not.
User Access	This field contains the name of the User Group whose members have regular access rights to this Project.
Admin Access	This field contains the name of the User Group whose members have administrative access rights to this Project.

4. Select the Projects to be synchronized and click the *Synchronize* button underneath the overview.

Select the check box in the column header to select all Projects for synchronization.

Note: In order to not overburden the IKAN ALM Server, it is best to limit the number of selected Projects and split up the Build Archive Synchronization in several steps.

5. The *Synchronize Build Archive Log* screen is displayed.



The extra field *Synchronizing Status* is displayed for each Project on the *Project Synchronization* panel. It can have the following values:

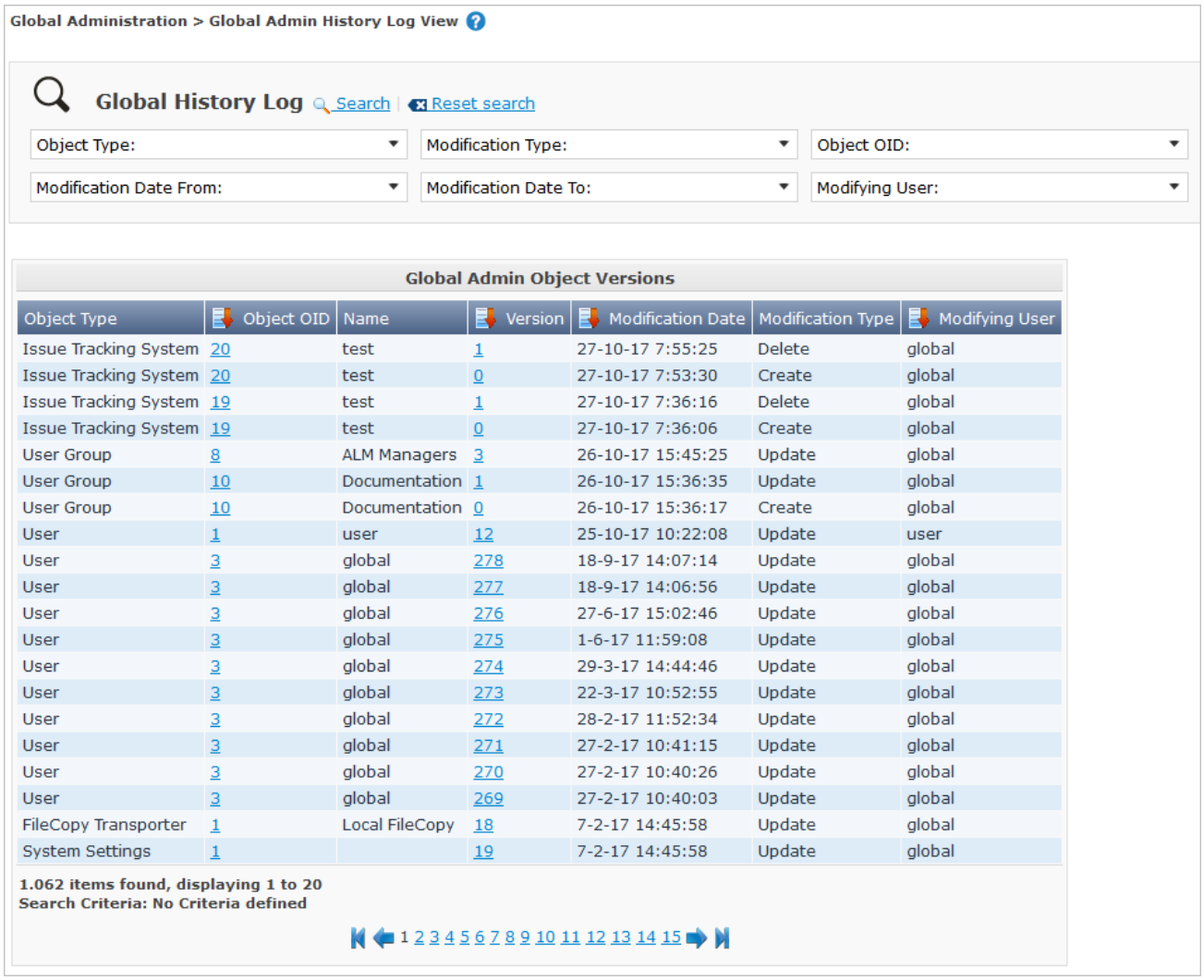
- *Done*: The Project has been synchronized successfully.
- *Synchronizing…* : The Project is currently being synchronized.

- *Waiting*: The synchronization of this Project has not yet started.
- *Error*: There was a problem synchronizing the Project.

6. Click *Refresh* to update the Synchronization statuses.

CHAPTER 25
History Log

1. In the Global Administration context, select *History Log*.
   The *Global Admin History Log View* screen is displayed.



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

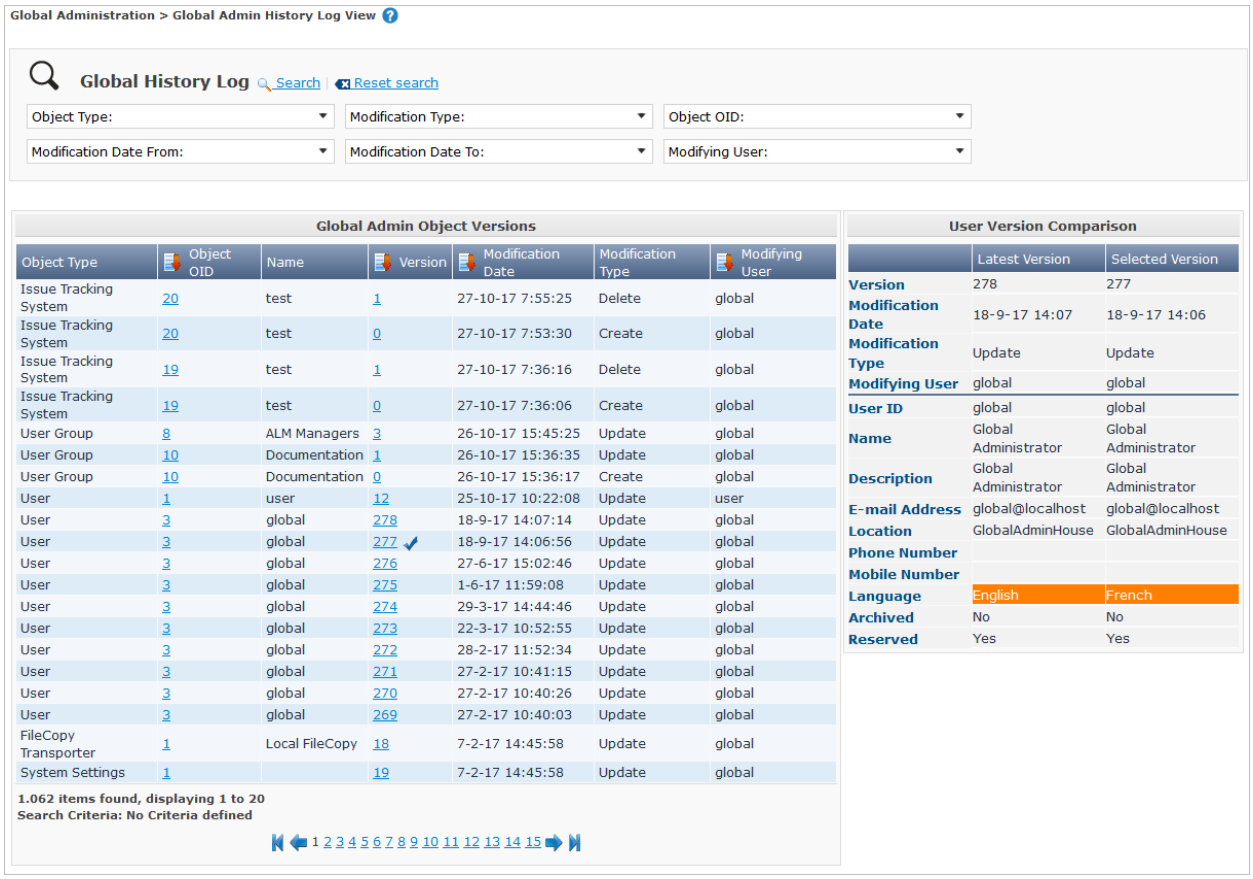
3. Verify the information on the *Global Admin Object Versions* panel.

The following information fields are available:

Field	Meaning
Object Type	The type of object that was modified.
Object OID	The OID number of the object. **Note:** Clicking this number will display all modifications for that object.
Name	The name of the object.
Version	The version the object had at the moment the modification was saved.
Modification Date	The date and time at which the modification took place.
Modification Type	The modification type can be one of the following: • *Create* The object was created • *Update* The object was updated • *Delete* The object was deleted • *Add* (for an association type) The association has been added • *Remove* (for an association type) The association has been removed
Modifying User	The User ID of the User who performed the modification.

4. Click a specific version on the *Global Admin Object Versions* panel to show the differences between that version and the current version of the object.

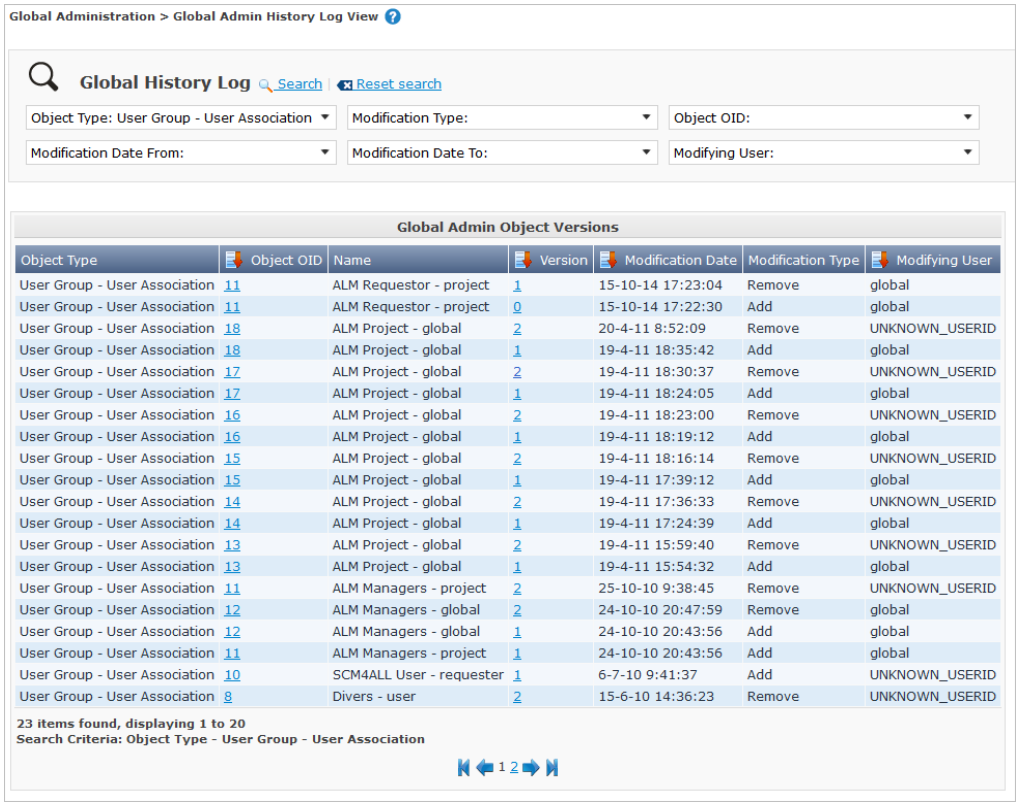
   A *Comparison* panel is added to the right of the *Versions* panel.



The *Version Comparison* panel displays the differences between the latest version of the object (shown in the "Latest" column) and the selected version of the object (shown in the "Selected" column). If the Object was deleted, the latest version will be equal to the status of the Object before it was deleted.

- The upper part of the *Comparison* panel displays the Version, Modification Date, Modification Type and User ID of the two versions that are being compared
- The lower part displays the object-specific values of the two versions that are being compared, highlighting the fields where there are differences.

Note: The lower part will be different if the object type is an association for an Add or Remove Modification Type, like in the added User Group - User Association in the example shown below.



Global Administration > Global Admin History Log View

Global History Log Search | Reset search

Object Type: User Group - User Association | Modification Type: | Object OID:

Modification Date From: | Modification Date To: | Modifying User:

Global Admin Object Versions

Object Type	Object OID	Name	Version	Modification Date	Modification Type	Modifying User
User Group - User Association	11	ALM Requestor - project	1	15-10-14 17:23:04	Remove	global
User Group - User Association	11	ALM Requestor - project	0	15-10-14 17:22:30	Add	global
User Group - User Association	18	ALM Project - global	2	20-4-11 8:52:09	Remove	UNKNOWN_USERID
User Group - User Association	18	ALM Project - global	1	19-4-11 18:35:42	Add	global
User Group - User Association	17	ALM Project - global	2	19-4-11 18:30:37	Remove	UNKNOWN_USERID
User Group - User Association	17	ALM Project - global	1	19-4-11 18:24:05	Add	global
User Group - User Association	16	ALM Project - global	2	19-4-11 18:23:00	Remove	UNKNOWN_USERID
User Group - User Association	16	ALM Project - global	1	19-4-11 18:19:12	Add	global
User Group - User Association	15	ALM Project - global	2	19-4-11 18:16:14	Remove	UNKNOWN_USERID
User Group - User Association	15	ALM Project - global	1	19-4-11 17:39:12	Add	global
User Group - User Association	14	ALM Project - global	2	19-4-11 17:36:33	Remove	UNKNOWN_USERID
User Group - User Association	14	ALM Project - global	1	19-4-11 17:24:39	Add	global
User Group - User Association	13	ALM Project - global	2	19-4-11 15:59:40	Remove	UNKNOWN_USERID
User Group - User Association	13	ALM Project - global	1	19-4-11 15:54:32	Add	global
User Group - User Association	11	ALM Managers - project	2	25-10-10 9:38:45	Remove	UNKNOWN_USERID
User Group - User Association	12	ALM Managers - global	2	24-10-10 20:47:59	Remove	global
User Group - User Association	12	ALM Managers - global	1	24-10-10 20:43:56	Add	global
User Group - User Association	11	ALM Managers - project	1	24-10-10 20:43:56	Add	global
User Group - User Association	10	SCM4ALL User - requester	1	6-7-10 9:41:37	Add	UNKNOWN_USERID
User Group - User Association	8	Divers - user	2	15-6-10 14:36:23	Remove	UNKNOWN_USERID

23 items found, displaying 1 to 20
Search Criteria: Object Type - User Group - User Association

1 2

CHAPTER 26
Users and Groups

Refer to the following sections for more information on:

- Users (page 275)
- User Groups (page 280)

26.1. Users

IKAN ALM uses the JAAS interface to base its User ID/Password authentication on existing Security Systems in your organization. This means that User IDs and external User Groups are maintained outside IKAN ALM.

As a result, Users may not be created in IKAN ALM via a menu entry: a User must first be created in the external security system. If a User has the appropriate rights, he or she may log on to IKAN ALM. Based on this first successful logon, the User will be automatically created in IKAN ALM.

A User ID can only be used to log on to IKAN ALM (page 3) and to define Personal Settings (page 15), if it exists in this external Security System.

If your IKAN ALM license allows no floating users and is fully based on reserved users it will not be possible to log on with another User ID once the maximum number of reserved active users is reached. If additional users need to log on to IKAN ALM, there are two possibilities:

- expand the IKAN ALM license (by adding reserved or floating users)
- archive one or more active users that are no longer required

Archived users can no longer log on to IKAN ALM. They are maintained in the system solely for historical purposes, in other words to display what actions they have performed *in the past*. Re-activating an archived user is only possible if this action does not exceed the user limit imposed by your IKAN ALM license.

If your IKAN ALM license allows floating users, new users can always log in, on condition that

- not all floating licenses are occupied by other User IDs
- the User is created in the external security system and has access rights to log on to IKAN ALM

Access Rights in IKAN ALM are based on the adherence of User IDs to User Groups. There are two types of User Groups: external and internal. External User Groups must exist in the external Security System, before they can be defined and used in IKAN ALM. There is no automatic mechanism for importing external User Groups from the external Security System into IKAN ALM, as a lot of these User Groups may not be required in IKAN ALM.

It is possible to add a User ID to an external User Group or to delete a User ID from such a User Group *within* IKAN ALM, but these actions have only a *temporary* effect. As soon as the IKAN ALM session is terminated and a new session is opened, IKAN ALM reads the settings from the external Security System and synchronizes the external IKAN ALM User Groups with the definitions available in this external security system. In other words, if the adherence of a User ID to an external User Group changes permanently, this change must be configured in the external Security System, not in IKAN ALM.

On the other hand, a lot of User Groups can be defined that are only used for notifications and approvals. Managing these groups in an external security system may be impractical, since the User Groups are used by IKAN ALM only. For that reason, there exist internal User Groups which can be managed in IKAN ALM only and (as a consequence) are not synchronized with an external authorization system.

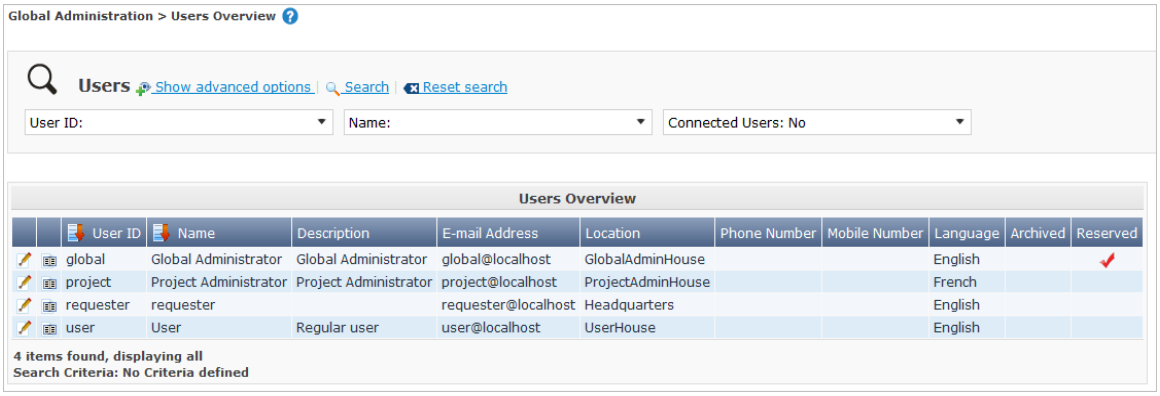
When adding/editing the User Groups, one should keep in mind the fact that, unlike the external User Groups, those internal groups are not automatically synchronized when the User logs in. To avoid situations where no User with System Admin rights can login anymore, IKAN ALM does not allow to set an internal User Group as the User Group with the Admin Access Rights.

The Users Overview option allows:

- defining additional information fields for User IDs that have been used to log in to IKAN ALM
- archiving active User IDs that are no longer required, or re-activating archived User IDs
- setting active User IDs as reserved, provided that this action does not exceed the maximum number of active reserved Users.
- displaying a list of Users connected to IKAN ALM, with the possibility to disconnect them, if needed, before their session has timed out.

The Users Overview Screen

1. In the Global Administration context, select *Users & Groups > Overview Users.*
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. Verify the information on the *Users Overview* panel.

The following information is available for each displayed User ID:

Field	Meaning
User ID	This field contains the User ID.
Name	This field contains the Name associated with the User ID.
Description	This field contains the defined description associated with the User ID.
Host Name	This field contains the name of the Machine the User used to connect to IKAN ALM. If the IP address of the Machines is displayed, this means that the host name could not be resolved. This field is only visible when the option *Connected Users* is selected in the *Search User* panel.
IP Address	This field contains the IP address of the Machine the User used to connect to IKAN ALM. This field is only visible when the option *Connected Users* is selected in the *Search User* panel.
Login Time	This field contains the date and time the User connect to IKAN ALM. This field is only visible when the option *Connected Users* is selected in the *Search User* panel.
E-mail Address	This field contains the E-mail Address associated with the User ID.
Location	This field contains the Location associated with the User ID. This field is not displayed on the Connected Users Overview.
Phone Number	This field contains the Phone Number associated with the User ID. This field is not displayed on the Connected Users Overview.
Mobile Number	This field contains the Mobile Number associated with the User ID. This field is not displayed on the Connected Users Overview.
Language	This field contains the Language setting associated with the User ID. There are three possibilities: • English • French • German This field is not displayed on the Connected Users Overview.
Archived	This field contains a check mark if the User ID has been archived. This field is empty for active names.
Reserved	This field contains a check mark if the User ID is a Reserved User ID. A Reserved non archived User ID may always log on to IKAN ALM, without occupying a Floating license. It is typically assigned to IKAN ALM Users with Global Admin Access Rights.

4. Depending on your access rights, the following links may be available:

Edit
This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a User definition. See Editing User Settings on page 278.

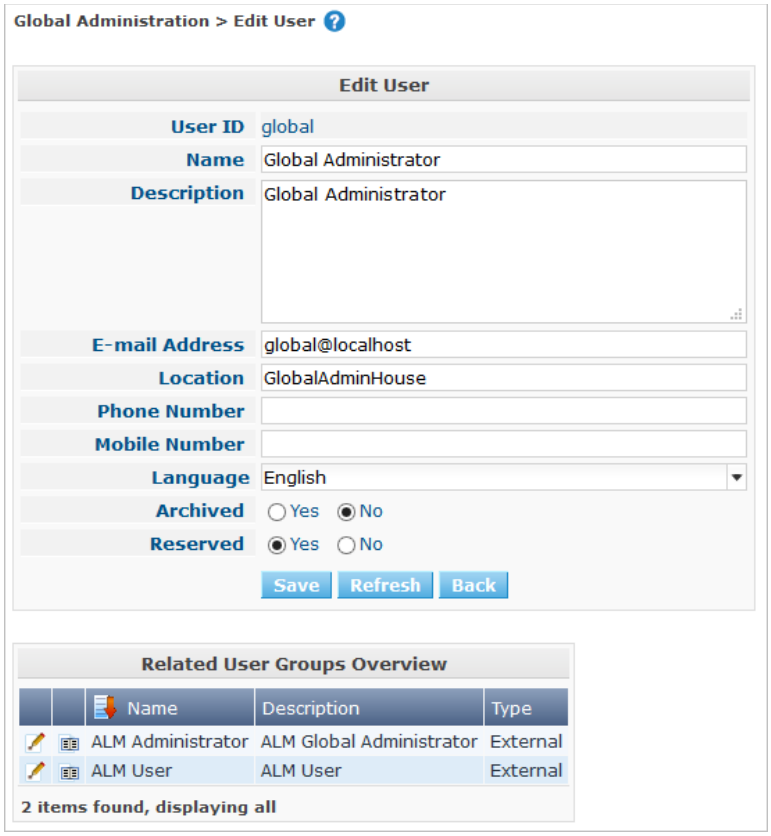
Disconnect
This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows disconnecting Users before their session has timed out. See Disconnecting Users on page 279.

History
This option is available to all IKAN ALM Users. It allows viewing the User History. See Viewing the User History on page 280.

Editing User Settings

1. In the Global Administration context, select *Users & Groups > Overview Users*.

2. Click the *Edit* link to change the selected User ID definition.
The following screen is displayed:



Description:

- At the top of the screen, the *Edit User* panel is displayed. Edit the fields as required.

  You may also reactivate an archived User by clicking the *No* option button next to the archived field.

  You may also mark a non reserved User as reserved, by clicking the Yes option button next to the reserved field. A reserved non archived user may always log in to IKAN ALM, without occupying a floating license. If the number is exceeded, the following message is displayed:

Error: Maximum number of reserved active Users (X) reached, where X is the maximum allowed number of reserved active users. You will have to expand your IKAN ALM license or archive other (obsolete) reserved User IDs, or mark other active reserved User IDs as non reserved, before you can reactive this User ID.

- At the bottom of the screen, the *Related User Groups Overview* panel is displayed. The overview contains the definitions of the User Groups to which the selected User ID belongs. See User Groups on page 280.

Note: Users can edit their personal settings. See Personal Settings on page 15.

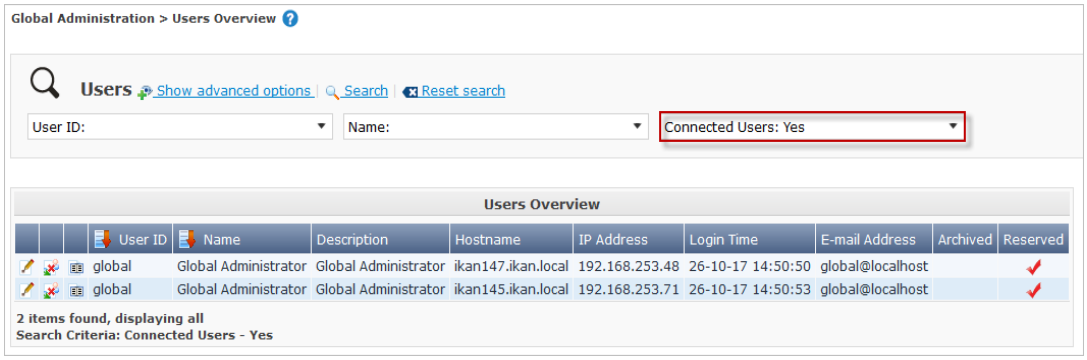
3. Edit the fields as required.
   For a description of the fields, refer to the section The Users Overview Screen (page 276).

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

Disconnecting Users

1. In the Global Administration context, select *Users & Groups > Overview Users*.

2. Set the option *Connected Users* to *Yes* to display the list of Users currently connected.



3. Click the *Disconnect* link on the *Users Overview* panel to immediately disconnect the User.
   This will end the User's session and disconnect him from IKAN ALM. As a result, the User has to log on again if he or she wants to continue his or her activity in IKAN ALM.

Viewing the User History

1. In the Global Administration context, select *Users & Groups* > *Overview Users*.

2. Click the *History* link on the *User Overview* panel to display the *User History View*.

Note: On the *Edit User* screen you also have access to the *User Group History View* screen by clicking the *History* link on the *User Groups Overview* panel.

For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

Click *Back* to return to the *Users Overview* screen.

RELATED TOPICS
- User Groups (page 280)
- Personal Settings (page 15)
- Managing the Desktop (page 135)
- Approvals (page 21)

26.2. User Groups

IKAN ALM uses the JAAS interface to base its User ID/Password authentication on existing Security Systems in your organization. This means that User IDs and their adherence to User Groups are maintained outside IKAN ALM. A User ID can only be used to log on to IKAN ALM (page 3) and to define Personal Settings (page 15), if it exists in this external Security System.

Access Rights in IKAN ALM are based on the adherence of User IDs to User Groups. There are two types of User Groups: external and internal. The external User Groups must exist in the external Security System, before they can be defined and used in IKAN ALM. There is no automatic mechanism for importing external User Groups from the external Security System into IKAN ALM, as a lot of these User Groups may not be required in IKAN ALM.

Deleting an external User Group in IKAN ALM means that it is not known to IKAN ALM anymore. It is however not deleted in the external Security System.

It is possible to add a User ID to a User Group or to delete a User ID from an external User Group *within* IKAN ALM, but these actions have only a *temporary* effect. As soon as the IKAN ALM session is terminated and a new session is opened, IKAN ALM reads the settings from the external Security System and synchronizes the external IKAN ALM User Groups with the definitions available in this external security system. In other words, if the adherence of a User ID to an external User Group changes permanently, this change must be configured in the external Security System, not in IKAN ALM.

On the other hand, a lot of User Groups can be defined that are only used for notifications and approvals. Managing these groups in an external security system is impractical, since the User Groups are used by IKAN ALM only. For that reason, there exist internal User Groups which can be managed in IKAN ALM only and which are not synchronized with an external authorization system.

When adding/editing the User Groups, one should keep in mind the fact that, unlike the external User Groups, those internal are not automatically synchronized when the user logs in. To avoid situations where no User with System Admin rights may login anymore, IKAN ALM does not allow to set an internal User Group as the User Group with the Admin Access Rights.

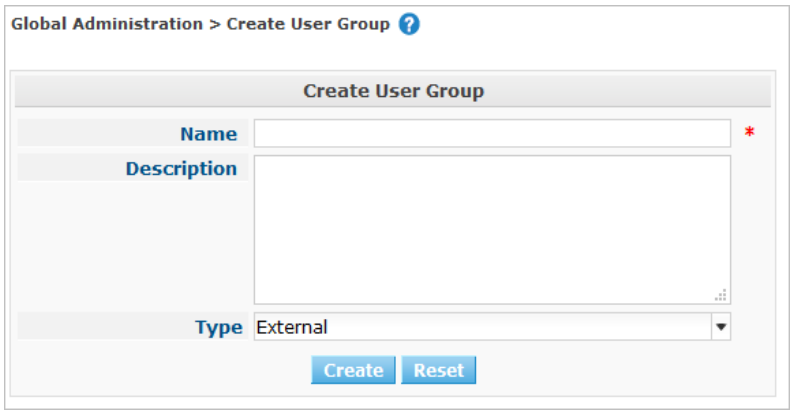
Refer to the following sections for detailed information:

- Creating a User Group (page 281)
- The User Groups Overview Screen (page 282)

Creating a User Group

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Users & Groups > Create User Group*.
   The following screen is displayed:



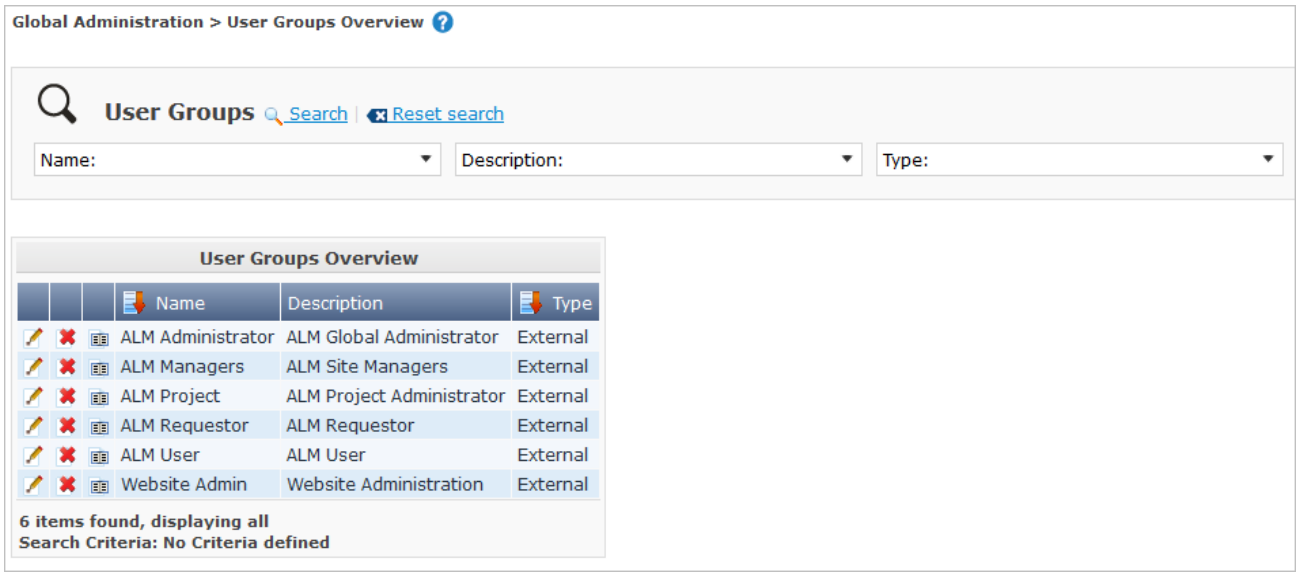
2. Fill out the fields in the *Create User Group* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	Enter the name of the new User Group in this field.
Description	Enter a description for the new User Group in this field.
Type	In this field, select the User Group type from the drop-down list. The possible values are: • External: the User Group membership is synchronized with the external security system. This is done for each User individually at login. • Internal: the User Group membership is managed in IKAN ALM For more information, refer to the section Editing a User Group (page 283).

3. Once you have filled out the fields, click *Create*.
   The newly created User Group is added to the *User Groups Overview* panel at the bottom of the screen.

The User Groups Overview Screen

1. In the Global Administration context, select *Users & Groups > Overview User Groups*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. Verify the information on the *User Groups Overview* panel.
   The following information is available for each displayed User Group:

Field	Meaning
Name	This field contains the defined User Group name.
Description	This field contains the defined description associated with the User Group.
Type	This field contains the User Group type which may be external or internal.

4. Depending on your access rights, the following links may be available:

Edit
This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a User Group definition. See Editing a User Group on page 283.

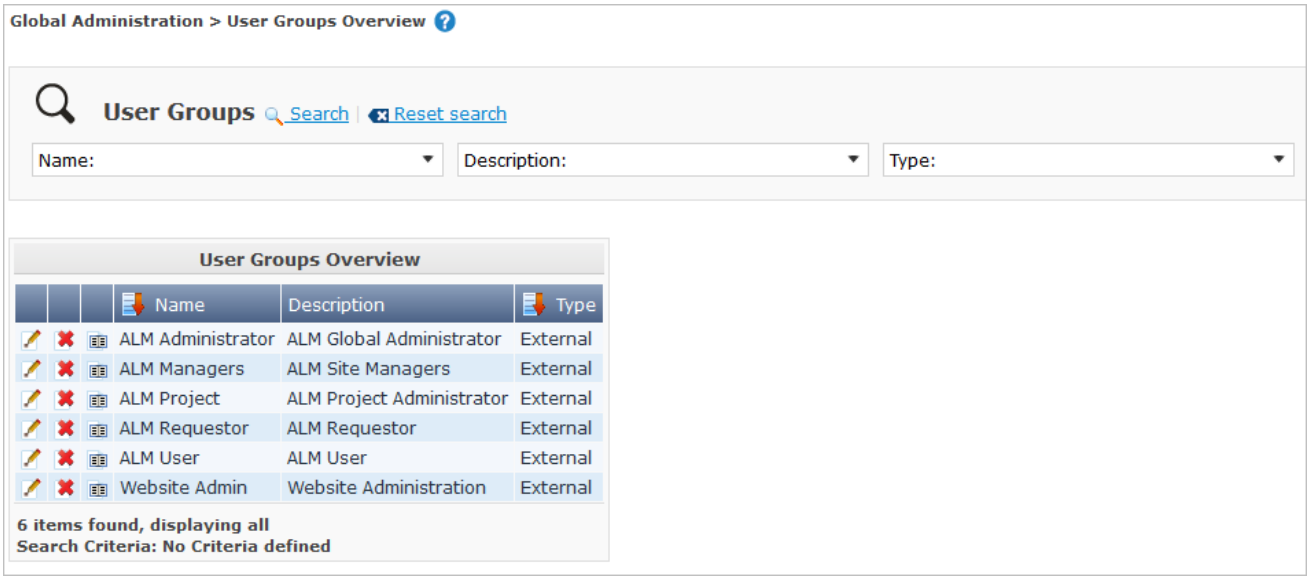
Delete
This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a User Group definition. See Deleting a User Group on page 287.

History
This option is available to all IKAN ALM Users. It allows viewing the User Group History. See Viewing the User Group History on page 288.

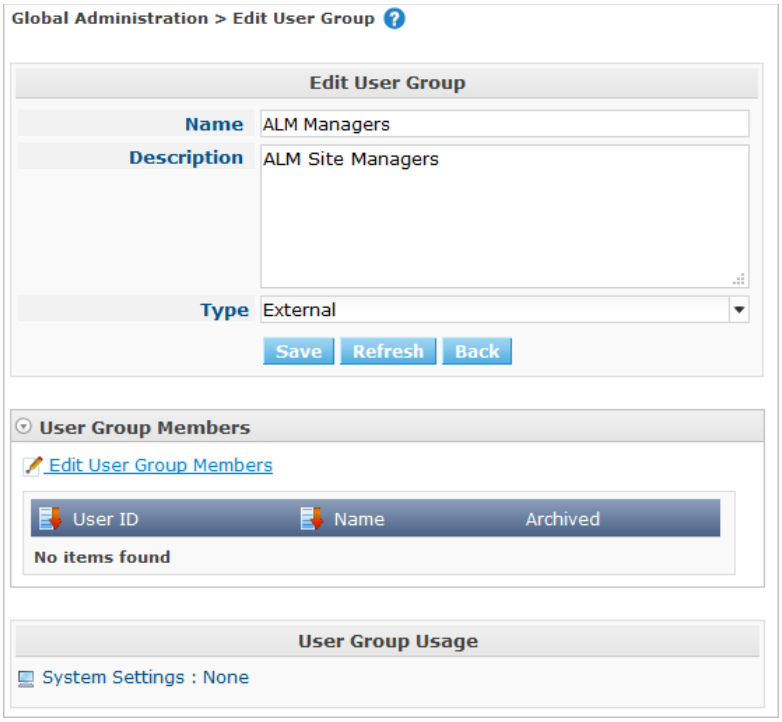
Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

Editing a User Group

1. In the Global Administration context, select *Users & Groups > Overview User Groups*.
The following screen is displayed:



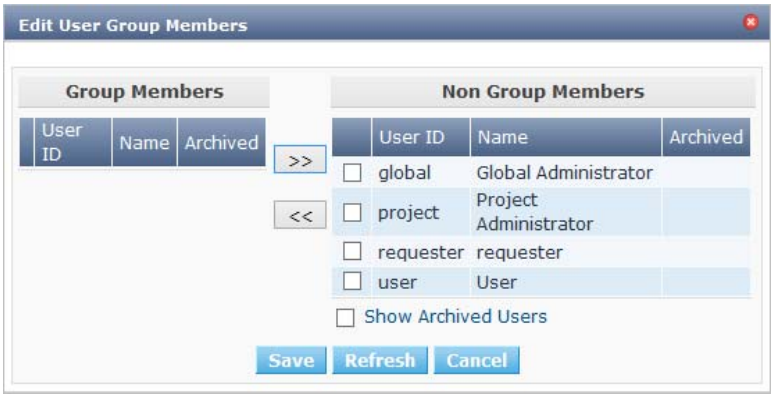
2. Click the *Edit* link in front of the User Group for which you want to modify the members.



3. In the *Edit User Group* panel, edit the fields as required.

4. Underneath the *Edit User Group* panel, click the *Edit User Group Members* link to change the selected User Group definition.

   The following screen is displayed:



   4.1. Edit the fields in the *Edit User Group Members* panel as required.

   Determine the allocation of User IDs to this User Group:

   - To add a User ID to this User Group, select the User ID in the *Non Group Members* list and click the << button.
   - To remove a User ID from this User Group, select the User ID in the *Group Members* list and click the >> button.

     By default, archived Users are filtered out of the *Non Group Members* list. To display all the Users, including the archived ones, select the *Show Archived Users* check box.

4.2. Click *Save* to save your changes.

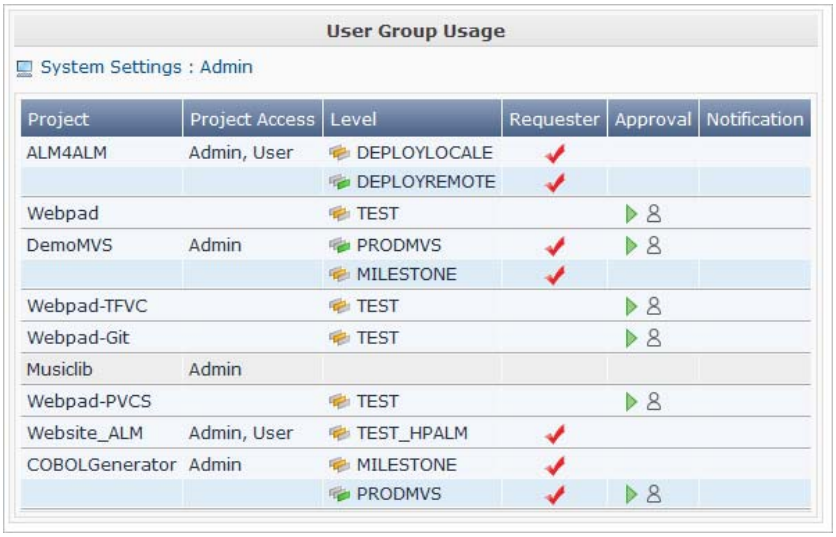
Note: For external User Groups, changing the Group Members has only a temporary effect. As soon as a User's IKAN ALM session is terminated and a new session is opened, IKAN ALM reads the settings from the external Security System and synchronizes the external IKAN ALM User Groups with the definitions available in this external security system. As a result a User added here will be deleted from the list, or vice versa.

However, the membership of internal User Groups may only be managed through this interface.

You can also click:

- *Refresh* to retrieve the settings from the database.
- *Cancel* to return to the previous screen without saving the changes.

4.3. The *User Group Usage* panel displays the System Settings Access Rights and Levels and Projects the User Groups are connected to.



5. Click *Save* to confirm your changes.

This will bring you back to the *User Groups Overview* screen.

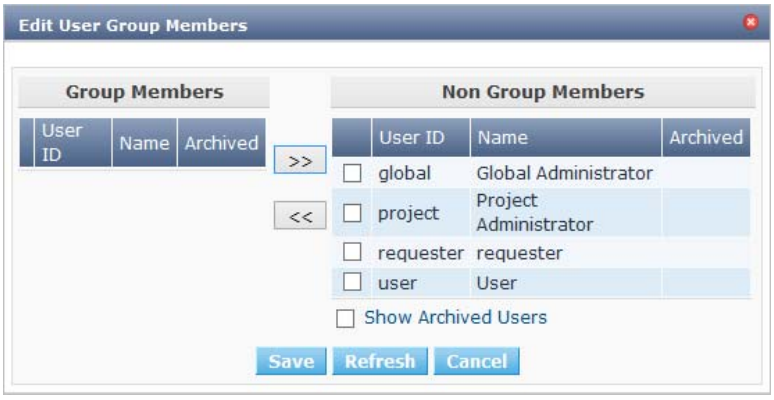
You can also click:

- *Refresh* to retrieve the settings from the database.
- *Back* to return to the previous screen without saving the changes

Editing the User Group Members

1. On the *User Group Members* panel underneath the *Edit User Group* panel (See Editing a User Group (page 283)), click the *Edit User Group Members* link to change the selected User Group definition.

   The following screen is displayed:



   Edit the fields in the *Edit User Group Members* panel as required.

   Determine the allocation of User IDs to this User Group:

   - To add a User ID to this User Group, select the User ID in the *Non Group Members* list and click the << button.
   - To remove a User ID from this User Group, select the User ID in the *Group Members* list and click the >> button.

     By default, archived Users are filtered out of the *Non Group Members* list. To display all the Users, including the archived ones, select the *Show Archived Users* check box.

2. Click *Save* to save your changes.

Note: For external User Groups, changing the Group Members has only a temporary effect. As soon as a User's IKAN ALM session is terminated and a new session is opened, IKAN ALM reads the settings from the external Security System and synchronizes the external IKAN ALM User Groups with the definitions available in this external security system. As a result a User added here will be deleted from the list, or vice versa.

However, the membership of internal User Groups may only be managed through this interface.

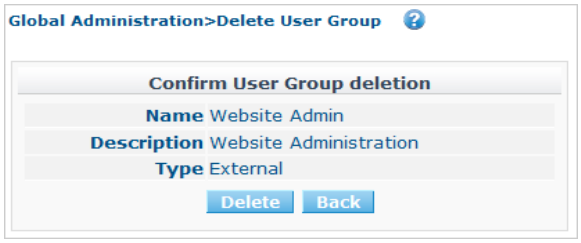
You can also click:

- *Refresh* to retrieve the settings from the database.
- *Cancel* to return to the previous screen without saving the changes.

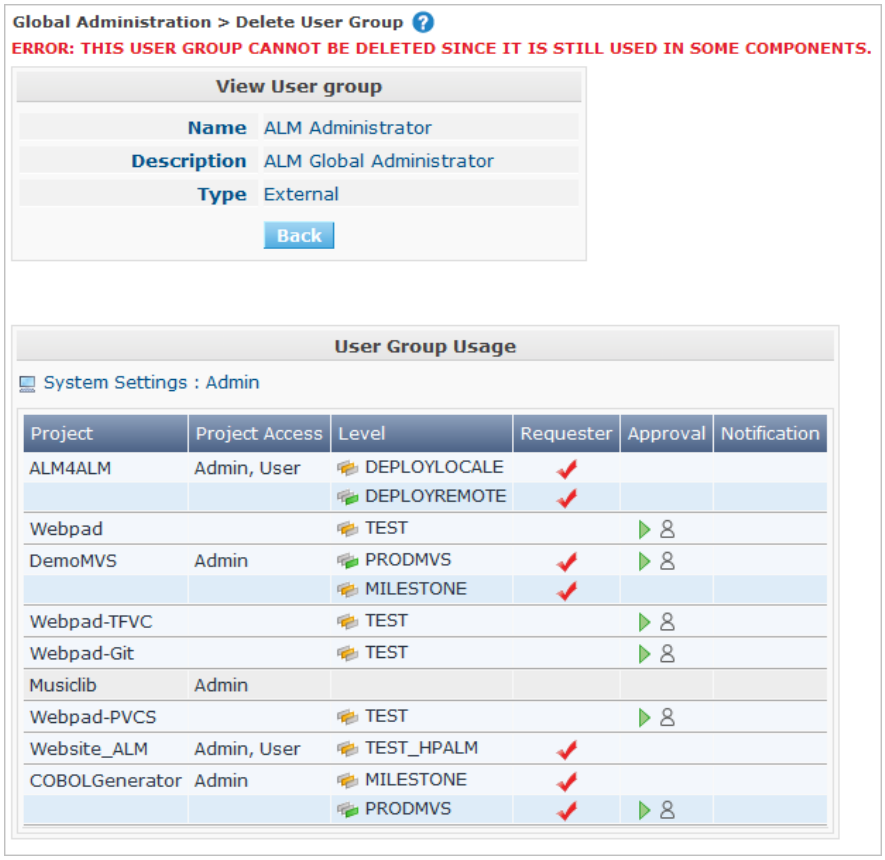
Deleting a User Group

1. In the Global Administration context, select *Users & Groups > Overview User Groups*.

2. Click the *Delete* link to delete the selected User Group definition.
   - If the User Group is not connected to any other IKAN ALM Object, the following screen is displayed:



   Click *Delete* to confirm the deletion or *Back* to return to the previous screen without deleting the entry.
   - If the User Group is still connected to IKAN ALM Objects on the System Project or Level Settings, the following dialog is displayed:



Note: You will need to cancel the displayed connections, before you can delete the User Group.

Viewing the User Group History

1. In the Global Administration context, select *Users & Groups* > *Overview User Groups*.

2. Click the *History* link to display the *User Group History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the *User Groups Overview* screen.

RELATED TOPICS
- Users (page 275)
- System Settings (page 255)
- Notifying IKAN ALM Users (page 472)
- Projects (page 461)
- Editing Project Settings (page 149)
- Creating a Build Level (page 188)
- Creating a Test or Production Level (page 190)
- Editing a Level (page 184)

CHAPTER 27
Machines

All Machines hosting Build and/or Deploy Environments that are used by IKAN ALM Projects, must be defined in IKAN ALM. This is the task of the IKAN ALM Global Administrator. Other IKAN ALM Users can verify, if all Machines used for a new Project are available. If not, they can request the IKAN ALM Global Administrator to add the Machine definitions to IKAN ALM. Build and Deploy Actions can only be handled on a Machine on which an IKAN ALM Agent is correctly installed and running.

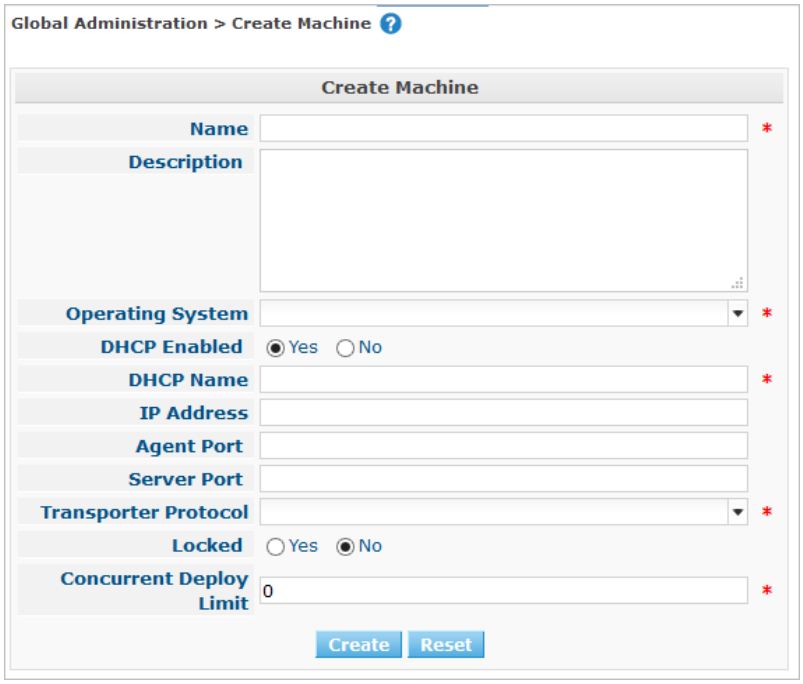
Refer to the following sections for detailed information:

- Creating a Machine Definition (page 289). This menu option is only visible for Users with Global Admin Access rights.
- The Machines Overview Screen (page 292)
- Editing a Machine Definition (page 294)
- Deleting a Machine Definition (page 296)
- Viewing the Machine Status (page 297)
- The Installed Phases Overview Screen (page 298)
- Viewing the Machine History (page 300)
- Machine Parameters (page 300)

27.1. Creating a Machine Definition

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Machines* > *Create*.
   The following screen is displayed:



2. Fill out the fields in the *Create Machine* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

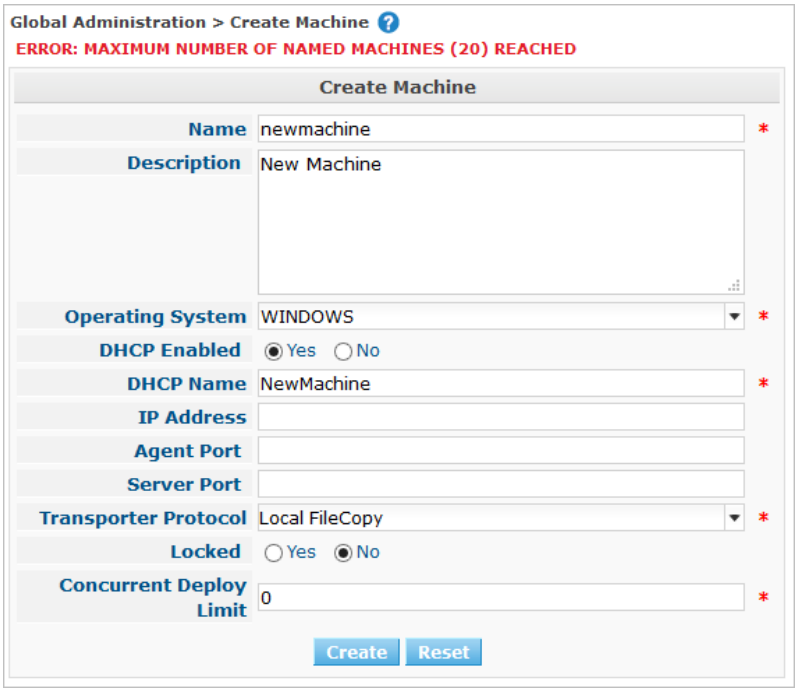
Field	Meaning
Name	Enter the name of the new Machine in this field. This field is mandatory.
Description	Enter a description for the new Machine in this field. This field is optional.
Operating System	Select the required Operating System from the drop-down list. The Select ion of an Operating System is mandatory.
DHCP Enabled	Indicate whether or not the Machine is DHCP-enabled.
DHCP Name	Enter the Machine's DHCP Name. Required field when the DHCP Enabled status is set to *Yes*. The IKAN ALM Server will try to set up a connection with this DHCP Name (using XML-RPC).
IP Address	Enter the Machine's IP Address. Required field when the DHCP Enabled status is set to *No*. The IKAN ALM Server will try to set up a connection with this IP Address (using XML-RPC).
Agent Port	In this field, you can enter the Port number (between 0 and 65536) on which the IKAN ALM Agent on this Machine will listen for communication with the IKAN ALM Server (using XML-RPC). If you leave this field empty, the default value of 20020 will be used. If you define a value different from the default value, be sure to define the same value when running the Agent install wizard on this Machine.
Server Port	In this field, enter the Port number (between 0 and 65536) on which the IKAN ALM Server will listen for communication with the IKAN ALM Agents (using XML-RPC). This port number was defined during the initial installation of the IKAN ALM Server. If the IKAN ALM Server uses the default value of 20021, you can also leave this field empty.

Field	Meaning
Transporter Protocol	Select the required Transporter Protocol from the drop-down list. The list contains the names of all Transporters defined in IKAN ALM (FTP, FileCopy and SSH). This Protocol defines how sources and build results will be transported between the IKAN ALM Server locations (Build Archive, Work Copy, Deploy Script location) and this Machine. The Select ion of a Transporter Protocol is mandatory. For more information on configuring Transporters, refer to the section Transporters (page 307).
Locked	Select whether or not the Machine must be locked for future use.
Concurrent Deploy Limit	Enter the maximum number of Deploys that may be run at the same time. • If no number is set to 0 (the default), there is no limit for running Deploys concurrently. • If the number is set to `1`, all deploys will run sequentially. • If a specific number is specified, only that number of Deploys can be run concurrently on the agent connected with the machine. If a next one is requested, it will go in the waiting queue and it will only be started if one of the running Deploys is finished (following the FIFO-principle based on the Deploy OIDs).

3. Once you have filled out the fields, click *Create*.

The newly created Machine definition is added to the *Machines Overview* at the bottom of the screen.

Your IKAN ALM User License may contain a limit on the number of Machines you can add. If this limit is exceeded, the new Machine definition is not added, and the following error message is displayed:

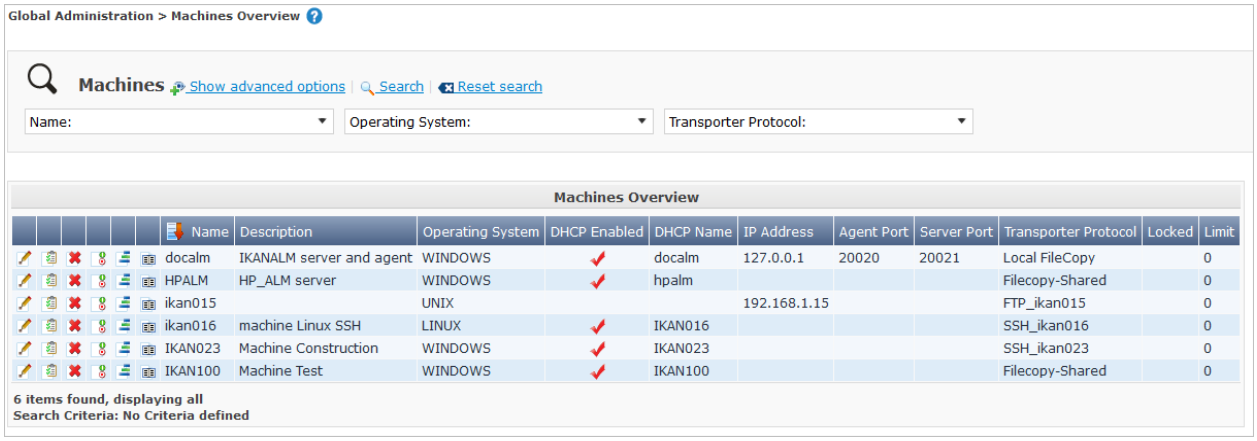


Contact your IKAN ALM Vendor if you need to purchase a license that allows for more Machine Definitions.

RELATED TOPICS
- Machines (page 289)
- Transporters (page 307)
- Build Environments (page 209)
- Deploy Environments (page 225)

27.2. The Machines Overview Screen

1. In the Global Administration context, select *Machines > Overview*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. Verify the information on the *Machines Overview* panel.
   For a detailed description of the fields, refer to Creating a Machine Definition (page 289).

4. Depending on your access rights, the following links may be available on the *Machines Overview* panel:

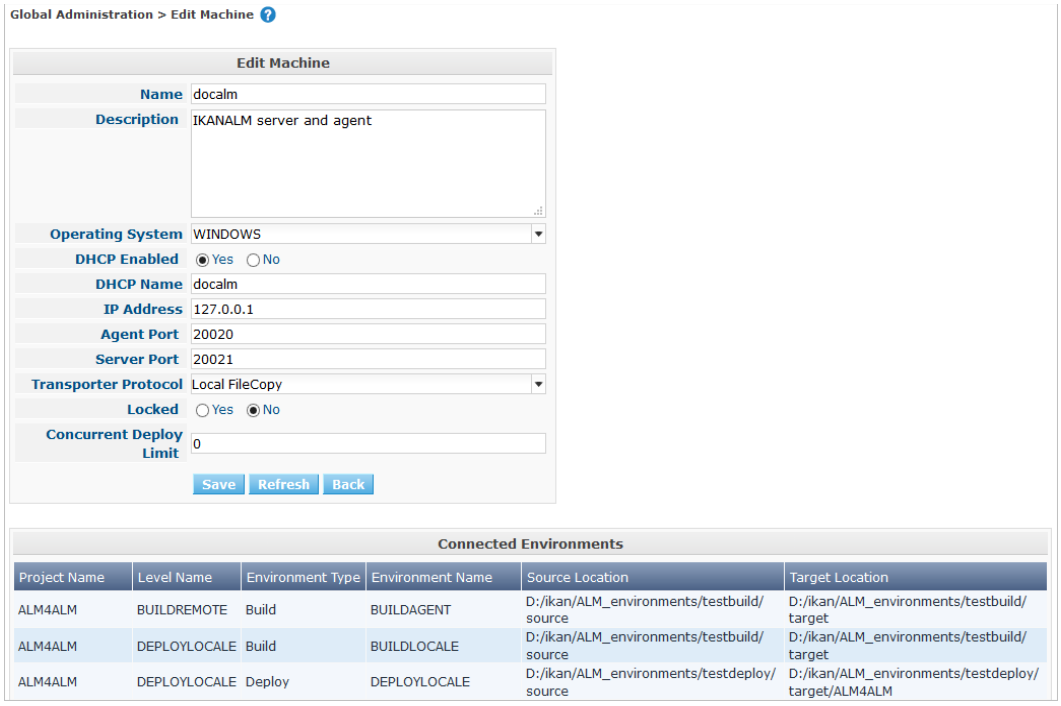
	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Machine definition. See Editing a Machine Definition on page 294.
	View Parameters This option is available to all IKAN ALM Users. It allows viewing and editing the Parameters of a Machine. See The Machine Parameters Overview Screen on page 301.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Machine definition. See Deleting a Machine Definition on page 296.
	Status This option is available to all IKAN ALM Users. It allows checking the status of a Machine. See Viewing the Machine Status on page 297.
	Installed Phases This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows viewing and uninstalling the phases that are currently installed on the Machine. See The Installed Phases Overview Screen on page 298.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Machine. See Viewing the Machine History on page 300.

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

27.3. Editing a Machine Definition

1. In the Global Administration context, select *Machines* > *Overview*.

2. Click the *Edit* link on the *Machines Overview* panel.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a Machine Definition (page 289).

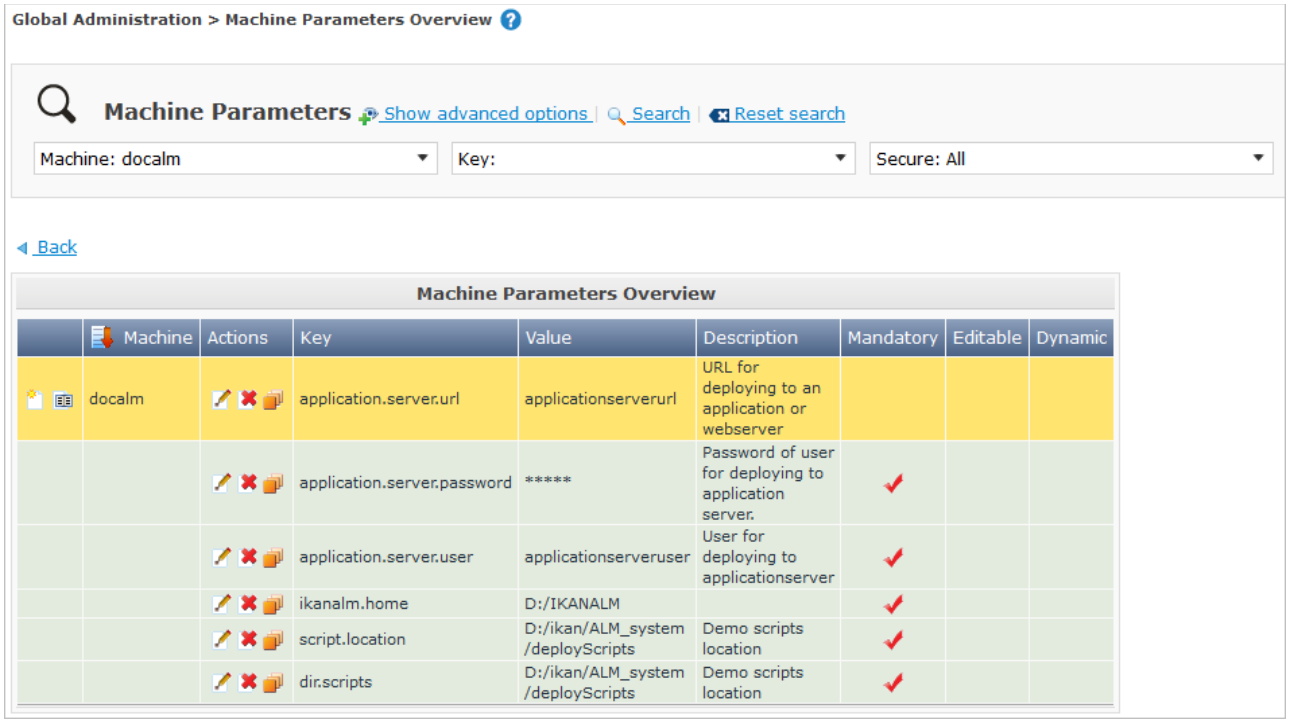
Note: The *Connected Environments* panel displays the Environments the Machine is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

27.4. Viewing the Machine Parameters

1. In the Global Administration context, select *Machines* > *Overview*.

2. Click the *View Parameters* link on the *Machines Overview* panel.
   The following screen is displayed:



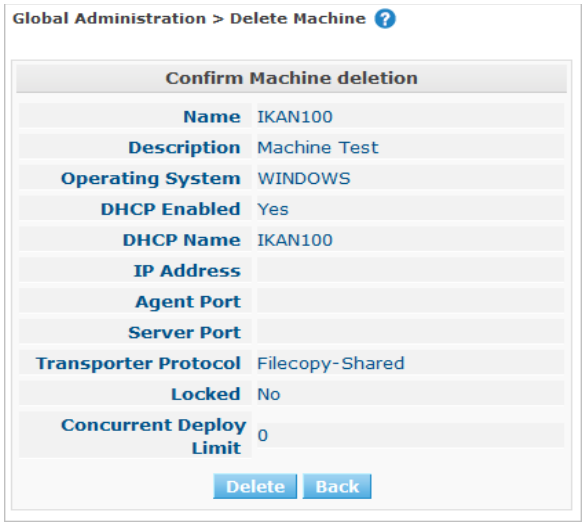
Note: You can also access the Machine Parameters Overview via the Main Menu by selecting *(Global Administration) > Machines > Machine Parameters*.

3. The *Machine Parameters Overview* screen lets you create, edit, delete and copy Machine Parameters and allows checking their history.
   For detailed information, refer to the following sections:
   - Creating Machine Parameters (page 303)
   - Editing Machine Parameters (page 304)
   - Deleting Machine Parameters (page 305)
   - Copying Machine Parameters (page 306)
   - Viewing the Machine Parameter History (page 306)

27.5. Deleting a Machine Definition

1. In the Global Administration context, select *Machines* > *Overview*.

2. Click the *Delete link* on the *Machines Overview* panel.
   The following screen is displayed:



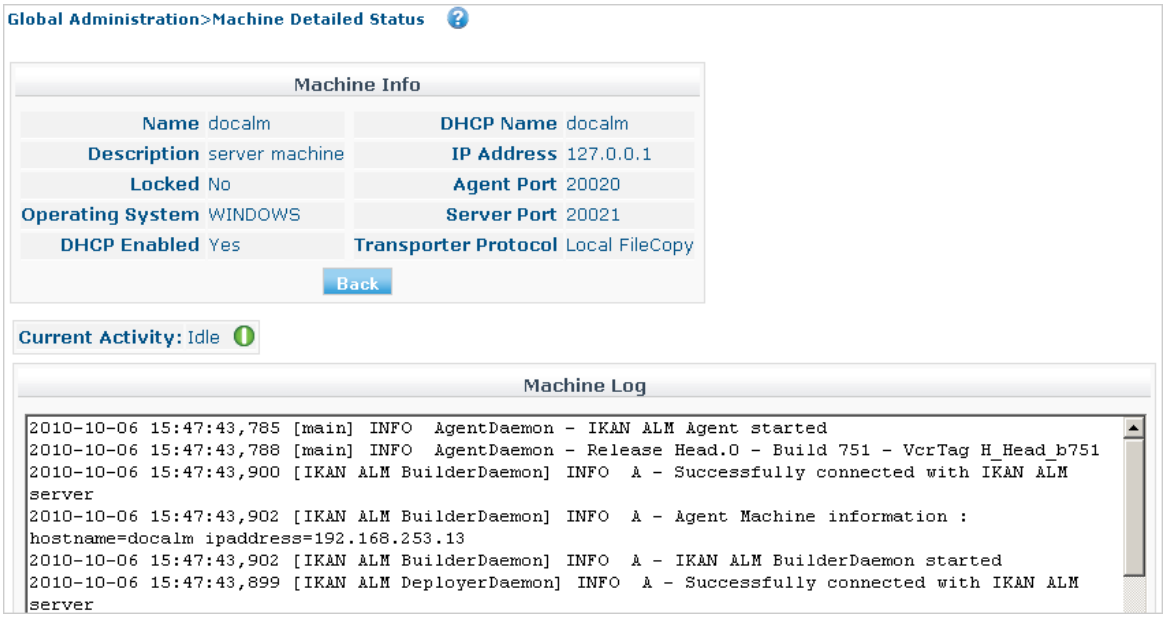
3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If you try to delete a Machine connected to a Build or Deploy Environment, the following message is displayed:

   You must link the Environments to a different Machine, or delete them from IKAN ALM, before you can delete the Machine definition.

27.6. Viewing the Machine Status

1. In the Global Administration context, select *Machines* > *Overview*.

2. Click the *Status* link on the *Machines Overview* panel.
   The following screen is displayed:



   The *Machine Detailed Status* screen displays the status of the Agent Daemon running on the Machine.
   At the top of the screen, the *Machine Info* panel is displayed. For a detailed description of the fields, refer to Creating a Machine Definition (page 289).

3. Verify the Status of the Machine.
   The possible statuses are:

Status	Description
Idle	Could successfully connect to the Agent/Server Daemon. The Agent/Server is currently not executing any Level Requests, Builds or Deploys.
Running Builds	Could successfully connect to the Agent Daemon. The Agent is currently executing Builds.
Running Deploys	Could successfully connect to the Agent Daemon. The Agent is currently executing Deploys.
Running Builds and Deploys	Could successfully connect to the Agent Daemon. The Agent is currently executing Builds and Deploys.
Running Level Requests	Could successfully connect to the Server Daemon. The Server is currently executing Level Requests.
Shutting Down	Could successfully connect to the Agent/Server Daemon. The Agent/Server is shutting down.

Status	Description
Could not connect to Agent	The connection to the Agent Daemon failed, either because the Agent Daemon is currently not running on the Machine, or due to networking issues that prevent connecting to the Agent Daemon. Contact your IKAN ALM Administrator.
Could not connect to Server	The connection to the Server Daemon failed, either because the Server Daemon is currently not running on the Machine, or due to networking issues that prevent connecting to the Server Daemon. Contact your IKAN ALM Administrator.

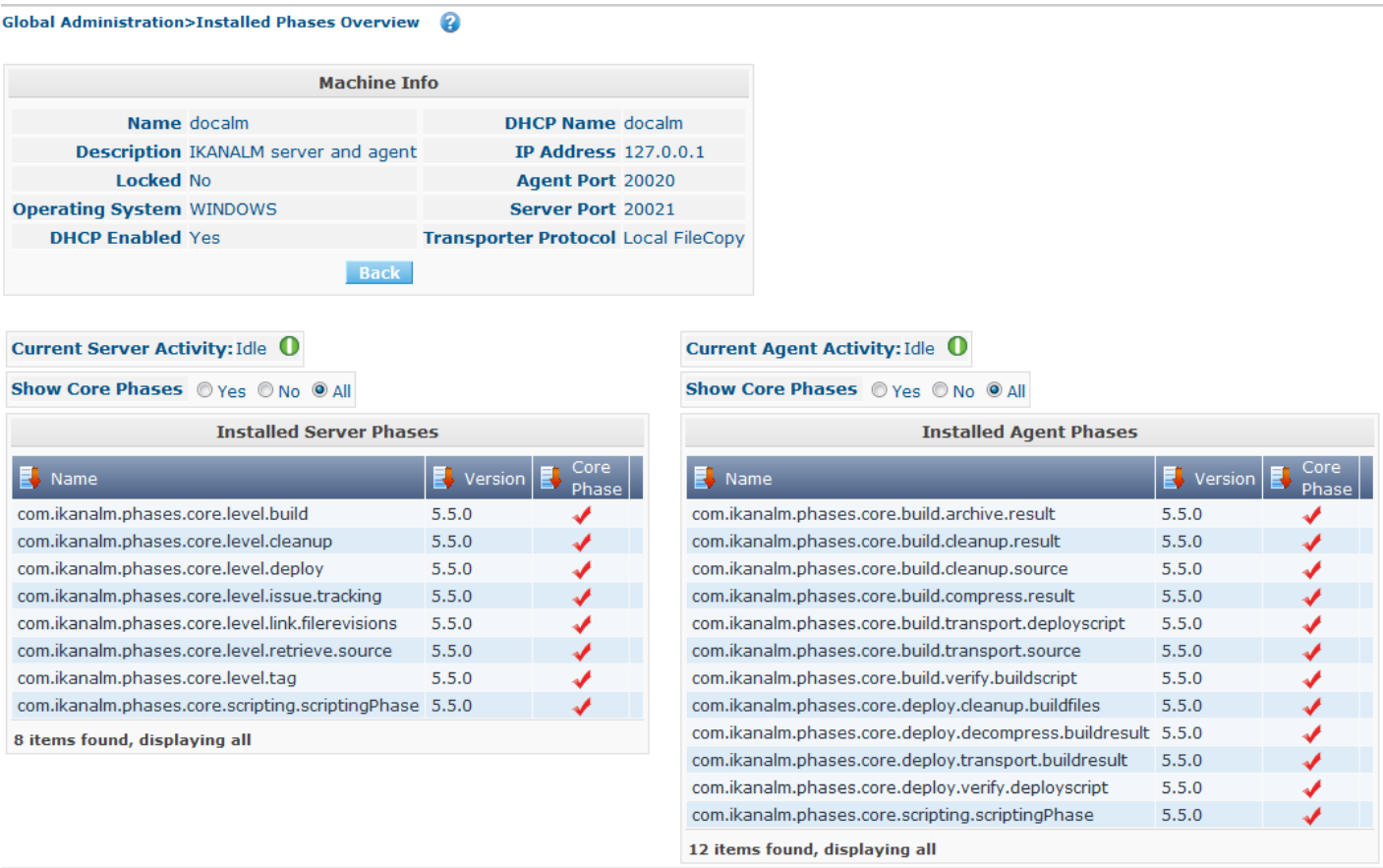
4. Verify the *Machine Log*.
   The *Machine Log* panel displays the last 150 lines of output of the Agent Daemon process running on this Machine.

5. Click *Back* to return to the *Machines Overview* screen.

27.7. The Installed Phases Overview Screen

1. In the Global Administration context, select *Machines* > *Overview*.

2. Click the *Installed Phases* link on the *Machines Overview* panel.
   The following screen is displayed.



The *Installed Phases Overview* screen displays the status of the Server and/or Agent daemons running on the Machine. It also shows the phases that are installed on the Server and Agent, and provides controls to search, sort and uninstall those phases.

At the top of the screen, the *Machine Info* panel is displayed. For a detailed description of the fields, refer to Creating a Machine Definition (page 289).

Note: The Activity and the installed phases on the Server daemon are only displayed if the Machine has been set as the "IKAN ALM Server" machine in the System Settings. See System Settings on page 255.

3. Verify the *Current Server and/or Agent Activity* on the Machine.

   For more information on the possible statuses, refer to Viewing the Machine Status (page 297).

4. Select whether to show the Core Phases or not.

   The possible options are:

   - *Yes*: show only the Core Phases
   - *No*: show only non-Core Phases
   - *All*: show Core and non-Core Phases

5. Verify the information on the *Installed Server Phases* and *Install Agent Phases* panels.

   - The *Installed Server Phases* panel shows all phases that are installed on the Server daemon of the Machine. This panel is only shown if the Machine has been set as the "IKAN ALM Server" machine in the System Settings. See System Settings on page 255.

- The *Installed Agent Phases* panel shows all phases that are installed on the Agent daemon of the Machine.

For each of the installed phases, the following information is available:

Information	Description
Name	The name of the Phase.
Version	The version of the Phase.
Core Phase	Core Phase or not?

6. Uninstalling Phases from the Server or Agent Daemons.

   To uninstall a Phase, click the *Delete* icon at the right of the Phase or Select the *Uninstall All* link to uninstall ALL non-Core phases of the Server or Agent daemon.

   Note: When a Phase is uninstalled, it is removed from the Server or Agent daemon. This does not mean that the Phase is removed from the Phase Catalog or from any connected Environments in Projects. When a Phase of a Level Request is executed on a certain Agent or Server and that Phase is not installed on that Agent or Server Daemon, IKAN ALM will automatically try to install the Phase on the Agent or Server daemon before executing it.

7. Click *Back* to return to the *Machines Overview* screen.

27.8. Viewing the Machine History

1. In the Global Administration context, select *Machines > Overview*.

2. Click the *History* link on the *Machines Overview* panel.

   The *Machine History View* is displayed.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Machines Overview* screen.

27.9. Machine Parameters

Unlike Build and Deploy Parameters, Machine Parameters are (obviously) defined for a Machine and not for a specific Environment. Parameters defined for a specific Machine, will automatically be available for all Environments using that Machine. This avoids having to (re)define Build and/or Deploy Parameters for each Environment linked to the Machine.

Note: If an Environment Parameter and a Machine Parameter have the same name, the Environment Parameter takes precedence.

Depending on the Scripting Tool linked to the environment, the defined parameters will be:

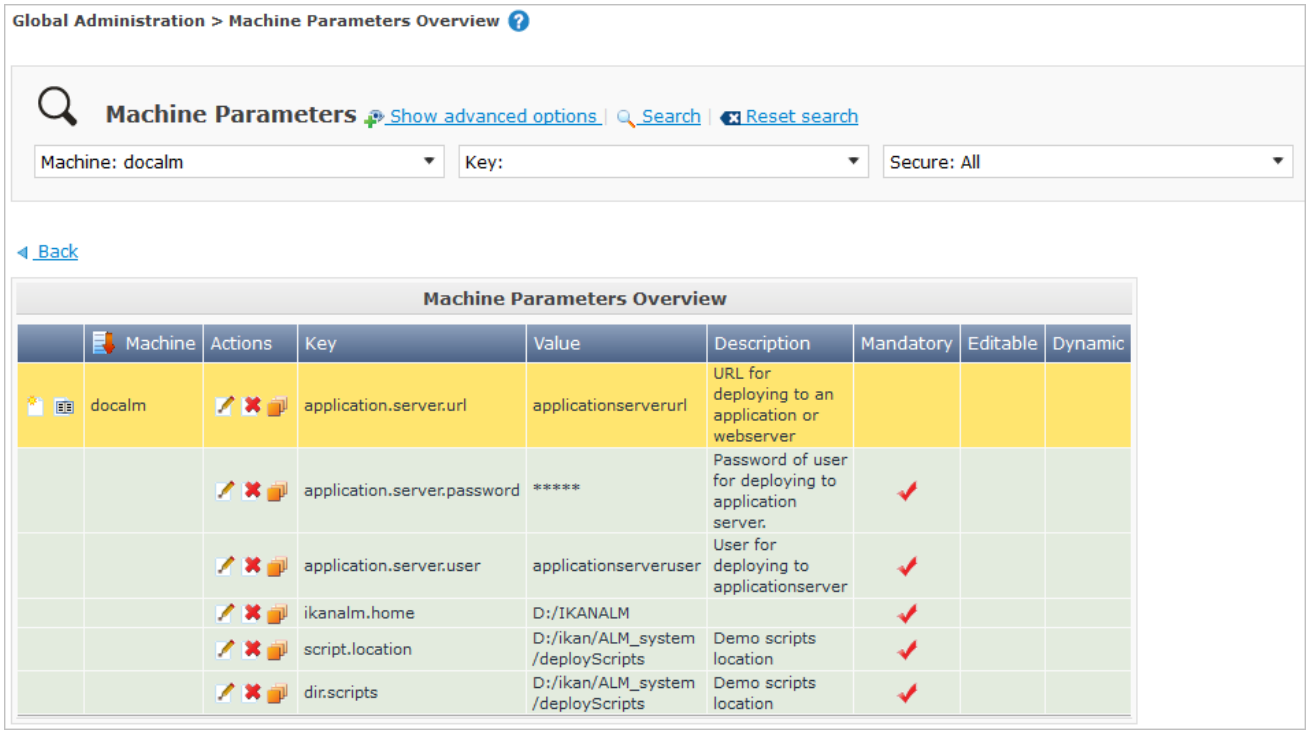
- added to the command which executes the Script (in the case of NAnt and Maven2)
- written to a specific file named *alm_ant.properties* (in the case of Ant) or *gradle.properties* (in the case of Gradel) which is automatic loaded with the `-propertyfile ANT` option. This property file is created on the fly in the source location of the Environment in the directory containing the Script (this may be a subdirectory of the source location of the Environment in case the location of the Script was defined using a relative path). Once the Build/Deploy process has terminated, this file is automatically deleted, unless the Debug option for the Environment linked to the Level has been activated.

The *Machine Parameters Overview* screen lets you create, edit, delete and copy Machine Parameters and allows checking their history. The following actions are possible:

- Creating Machine Parameters (page 303)
- Editing Machine Parameters (page 304)
- Deleting Machine Parameters (page 305)
- Copying Machine Parameters (page 306)
- Viewing the Machine Parameter History (page 306)

The Machine Parameters Overview Screen

1. In the Global Administration context, select *Machines > Machine Parameters*.
   The following screen is displayed:



Note: You can also access the Machine Parameters Overview via the Machines Overview by selecting *(Global Administration) > Machines > Overview* and, next, clicking the *View Parameters* link for the required Machine.

2. Define the required search criteria on the search panel.

   The list of items on the overview will be automatically updated based on the selected criteria.

   You can also:

   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields,

3. Verify the information on the *Machine Parameters Overview* panel.

   The *Machine Parameters Overview* panel displays the defined Machine Parameters for each Machine.

   For a description of the fields, see Creating Machine Parameters (page 303).

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

4. Depending on your access rights, the following links may be available on the *Machine Parameters Overview* panel:

Link	Description
	Create This option is available to all Users with Global Administrator Access Rights. It allows creating a Machine Parameter. See Creating Machine Parameters on page 303.
	History This option is available to all Users with Global Administrator Access Rights. It allows displaying the History of the selected Machine Parameter definition. See Viewing the Machine Parameter History on page 306.
	Edit This option is available to all Users with Global Administrator Access Rights. It allows editing the selected Machine Parameter definition. See Editing Machine Parameters on page 304.
	Delete This option is available to all Users with Global Administrator Access Rights. It allows deleting the selected Machine Parameter definition and (optionally) deleting Machine Parameters with the same key linked to other Machines. See Deleting Machine Parameters on page 305.
	Copy Parameter This option is available to all Users with Global Administrator Access Rights. It allows copying the selected Machine Parameter definition. See Copying Machine Parameters on page 306.

Creating Machine Parameters

1. In the Global Administration context, select *Machines > Parameters Overview*.

2. Click the *Create Parameter* link next to the Machine to display the *Create Machine Parameter* window.

3. Fill out the fields for the new Machine Parameter.
   The following fields are available. The *Key* field is mandatory:

Field	Meaning
Machine	This field displays the current Machine.
Secure	This field indicates whether the Parameter is secured or not.
Key	In this field, enter the Key (Name) for the Machine Parameter.
Value	In this field, enter the value(s) for the new Machine Parameter. The following possibilities apply: • Enter the fixed value, if you are creating a non-editable Machine Parameter. • Enter the default value, if you are creating an editable Machine Parameter. • Enter the list of possible preset values, separated by a semicolon (;), if you are creating a dynamic Machine Parameter (for example: `yes;no`). Subsequently, these values can be selected from a drop-down list when creating a Level Request
Repeat Value	Required field for secured Machine Parameters: repeat the secured value.
Description	In this field, enter a description for the Parameter.
Mandatory	Select the *Yes* option button, if the new Machine Parameter must be defined as mandatory. When you create a Level Request, mandatory Parameters will always be provided to the Build/Deploy Script. Select the *No* option button, if the new Machine Parameter should not be defined as mandatory. When you create a Level Request, you can decide whether you want to provide the non-mandatory Parameter to the Build/Deploy Script.

Field	Meaning
Editable	Select the *Yes* option button, if the new Machine Parameter must be defined as editable. When you create a Level Request, you can accept the default value (the one you enter in the Value field during creation) or specify the value of your choice for the Parameter. Select the *No* option button, if the new Machine Parameter should not be defined as editable. When you create a Level Request, only the preset value (the one you entered in the Value field during creation) for this Parameter can be offered to the Build/Deploy Script. This field is not provided for secured Machine Parameters.
Dynamic	Select the *Yes* option button, if the new Machine Parameter must be defined as dynamic. When you create a Level Request, you can Select one of the predefined values from the drop-down list. These are the values you enter in the Value field during creation and which you separate by a semicolon (;). The selected value will be offered to the Build/Deploy Script. Select the *No* option button, if the new Machine Parameter should not be defined as dynamic. This field is not provided for secured Machine Parameters.

4. Click *Create* to confirm the creation of the Machine Parameter.
   You can also click:
   - *Reset* to clear the fields and restore the initial values.
   - *Cancel* to return to the previous screen without saving the changes.

Editing Machine Parameters

1. In the Global Administration context, select *Machines > Parameters Overview*.

2. In the *Actions* column, click the *Edit* link in front of the Machine Parameter to be edited.
   The following window is displayed:

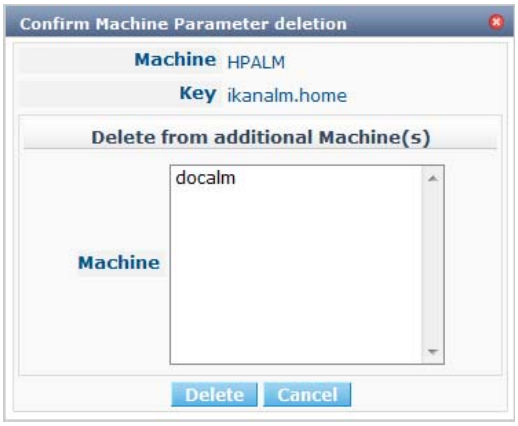
3. Edit the fields as required.
   For a description of the fields, refer to Creating Machine Parameters (page 303).

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Cancel* to return to the previous screen without saving the changes

Deleting Machine Parameters

1. In the Global Administration context, select *Machines > Parameters Overview*.

2. In the *Actions* column, click the *Delete* link in front of the Machine Parameter to be deleted.
   The following confirmation window is displayed:



3. Optionally, Select additional Machines. This allows for deleting parameters with the same key name on the selected Machines.

4. Click *Delete* to delete the Machine Parameter.
   You can also click *Cancel* to return to the previous screen without deleting the Parameter.

Copying Machine Parameters

This functionality allows copying a complete Machine Parameter definition from one source Machine to one or more target Machines.

1. In the Global Administration context, select *Machines > Parameters Overview*.

2. In the *Actions* column, click the *Copy* link in front of the Machine Parameter to be copied.
   The following window displays the values of the parameter you are about to copy.

3. Indicate whether you want to replace the parameter in case it already exists in the target Machine definition(s).

4. Select the Target Machine(s)

5. Click *Copy* to confirm copying the Machine Parameter.
   You can also click:
   - *Reset* to clear the fields and restore the initial values.
   - *Cancel* to return to the previous screen without saving the changes.

Viewing the Machine Parameter History

1. In the Global Administration context, select *Machines > Parameters Overview*.

2. Click the *History* link on the *Parameters Overview* panel to display the *Machine History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).
   Click *Back* to return to the previous screen.

CHAPTER 28
Transporters

Every Build or Deploy action handled by an IKAN ALM Agent involves sources, a script or a build result that must be transported between the IKAN ALM Agent and the IKAN ALM server.

The IKAN ALM Agent may reside locally or on a remote Machine (i.e., a server different from the IKAN ALM server.)

Custom Phases which have been created or imported into the Phase Catalog of the IKAN ALM Server need to be transported and installed in the execution environment of the IKAN ALM Server (Custom Level Phases) or Agent (Custom Build and Deploy Phases).

The Transporter is involved in the following Build and Deploy processes:

BUILD PROCESSES:

- transporting the checked out sources from the IKAN ALM server to the source location on the Build Environment,
- transporting the Build script from the script location on the IKAN ALM server to the source location on the Build Environment (only executed if the Build script is not included in the sources),
- transporting the compressed Build result from the target location on the Build Environment to the Build Archive on the IKAN ALM server.

DEPLOY PROCESSES:

- transporting the Build result from the Build Archive on the IKAN ALM server to the source location on the Deploy Environment,
- transporting the Deploy script from the script location on the IKAN ALM server to the source location on the Deploy Environment (only if the Deploy script is not included in the sources).

Transport actions are always initiated by the IKAN ALM Agent. Therefore, the definition of a Transporter will always be a view of the IKAN ALM Agent on the locations of the IKAN ALM Server. These locations may be defined locally (Agent running on the same Machine as the IKAN ALM server) or remotely (Agent running on a different Machine as the IKAN ALM server) for the IKAN ALM Agent.

As a consequence, a Transporter must be defined for a specific Machine that is linked to the Build or Deploy Environment. The IKAN ALM Agent running on that Machine will execute the transport actions.

The Transporter is also involved in the distribution process of Custom Phases from the Phase Catalog to the execution environment of the IKAN ALM Server (Level Phases) or Agent (Build and Deploy Phases):

- Level Phases:

  Before the execution of the Level Phases for a Level Request, all non-released Custom Phases will be transported from the Phase Catalog to the execution environment on the IKAN ALM Server. This will be done using the Transporter that is linked to the IKAN ALM Server. Released Level Phases will only be transported if they are not yet installed on the IKAN ALM Server. You can check the installed Phases using the *Installed Phases* link next to the IKAN ALM Server Machine on the *Machines Overview*. See The Machines Overview Screen on page 292.

- Build and Deploy Phases:

  Before the execution of the Build/Deploy Phases for Build/Deploy, all non-released Custom Phases will be transported from the Phase Catalog on the IKAN ALM Server to the execution environment on the IKAN ALM Agent. This will be done using the Transporter that is linked to the IKAN ALM Agent. Released Build/Deploy Phases will only be transported if they are not yet installed on the IKAN ALM Agent. You can check the installed Phases using the *Installed Phases* link next to the IKAN ALM Agent Machine on the *Machines Overview*. See The Machines Overview Screen on page 292.

The relative locations for all remote transporters must be defined in the *System Settings* section. See System Settings on page 255.

Note: A Transporter will be activated by linking the Transporter to a Machine representing the IKAN ALM Agent (The Machines Overview Screen (page 292)), and by linking that Machine to a Build or Deploy environment (Creating a Build Environment (page 210) or Creating a Deploy Environment (page 225)).

This chapter describes three different Transporters, each of them enabling different protocols for transferring sources, scripts or build results.

Refer to the following sections for detailed information:

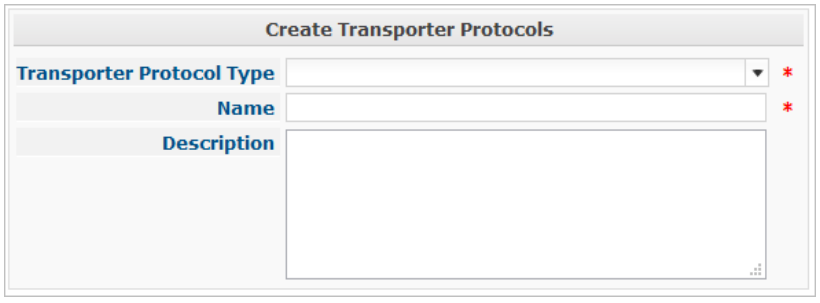
- FileCopy Transporters (page 310)
- FTP Transporters (page 316)
- Secure Shell Transporters (page 323)

28.1. Creating a Transporter Protocol

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Transporters > Create*.

   The following screen is displayed:



2. Select the *Transporter Protocol Type* you want to create.

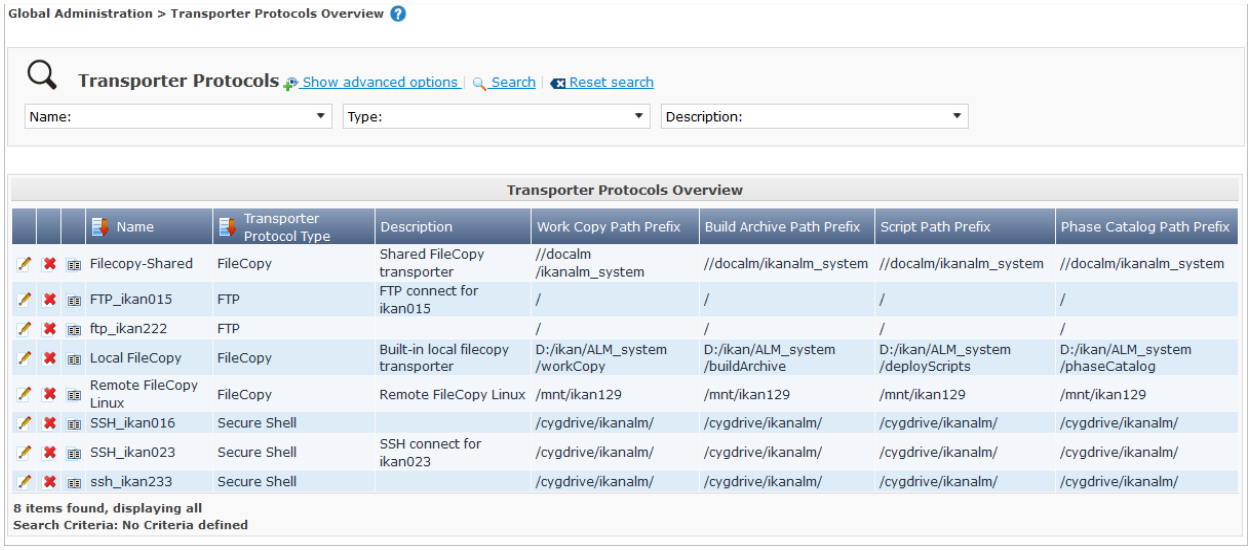
   The *Details* panel for the type of Transporter Protocol you selected is displayed underneath.

3. Enter the *Name* of the Transporter Protocol and add an optional *Description*.

4. Fill out the Transporter Protocol-specific Details.
   For more information on those details, refer to the appropriate sections:
   - Creating a FileCopy Transporter (page 311)
   - Creating an FTP Transporter (page 316)
   - Creating a Secure Shell Transporter (page 323)

28.2. The Transporter Protocols Overview Screen

1. In the Global Administration context, select *Transporters > Overview*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. Verify the information on the *Transporter Protocols Overview* panel.

4. Depending on your access rights, the following links may be available on the *Transporter Protocols Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Transporter Protocol definition.

	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Transporter Protocol definition.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Transporter Protocol.

Refer to the following sections for detailed information:

- FileCopy Transporters (page 310)
- FTP Transporters (page 316)
- Secure Shell Transporters (page 323)

28.3. FileCopy Transporters

A FileCopy Transporter makes it possible to use “file copy” for transporting Custom Phases, files and directories between the IKAN ALM server and a local or remote Agent handling the Build or Deploy processes (see the introduction on Transporters (page 307)).

Since most transport actions are handled by the Agent (transport of Custom Level Phases is handled by the IKAN ALM Server), and the FileCopy Transporter enables the copying of resources between the Agent and the IKAN ALM server, different FileCopy Transporters must be defined for a local Agent (Machine) running on the IKAN ALM Server (Machine) and for a remote Agent (Machine) running on a server other than the IKAN ALM Server.

The "Local FileCopy" Transporter is created automatically during the IKAN ALM installation and will be automatically connected to the IKAN ALM Server Machine. This "Local FileCopy" Transporter is different in this aspect that it cannot be edited, updated nor deleted on the *FileCopy Transporters Overview*. Its definition is retrieved from the System Settings' *Local FileCopy Locations* and can only be updated in the System Settings section. See System Settings on page 255.

If you want to use FileCopy for transferring resources to a Machine (Agent) other than the IKAN ALM server, a (Remote) FileCopy Transporter must be created. Keep in mind that the specified prefix locations are a view of the IKAN ALM Agent Machine on the IKAN ALM Server Machine. For Windows machines this means that the "Work Copy", "Build Archive", "Phase Catalog" and "Script" locations must be shared on the IKAN ALM Server. The prefixes in the FileCopy definition use the UNC notation containing the IKAN ALM Server Name and the share on the IKAN ALM Server. For Linux/Unix machines, this means that the "Work Copy", "Build Archive" and "Script" locations must be mounted on the IKAN ALM Agent Machine, and the mount name will be used in the different prefixes of the FileCopy Transporter definition.

Note: Be aware of access rights problems during FileCopy transport actions to the locations on the remote IKAN ALM server when running a remote IKAN ALM Agent as a Windows service or a Linux/Unix daemon process. These may be tackled by running the service/daemon as a User having access rights to the share/mount to the IKAN ALM server, or by giving the IKAN ALM Agent machine rights on the locations on the IKAN ALM server.

Refer to the following sections for detailed information:

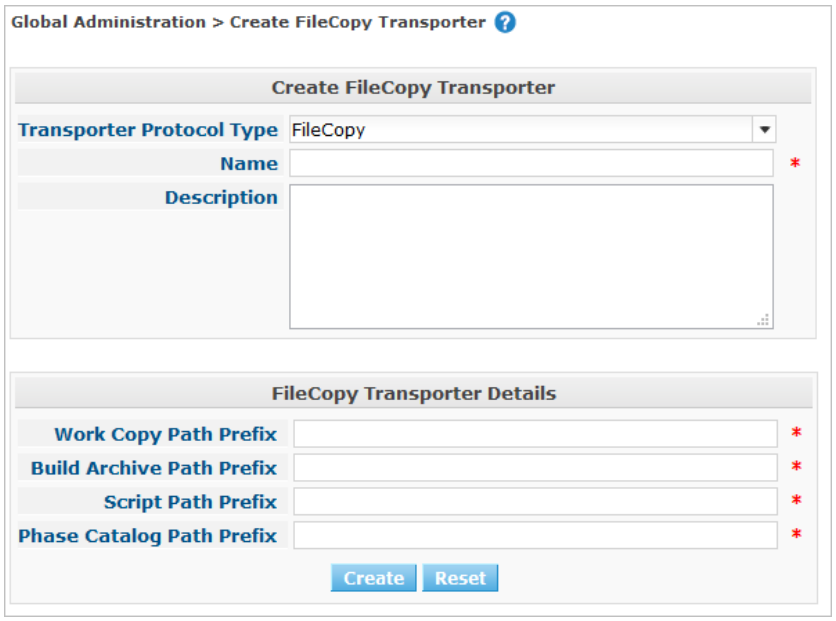
- Creating a FileCopy Transporter (page 311)
- The FileCopy Transporters Overview Screen (page 313)
- Editing a File Copy Transporter Definition (page 314)
- Deleting a File Copy Transporter Definition (page 314)
- Viewing the File Copy Transporter History (page 315)

Creating a FileCopy Transporter

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Transporters* > *Create*.

2. Select *FileCopy* from the drop-down list in the *Type* field on the search panel.
   The following screen is displayed:



3. Fill out the fields in the *Create FileCopy Transporter* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	Enter the name of the new FileCopy Transporter in this field. This field is mandatory.
Description	Enter a description for the new FileCopy Transporter in this field. This field is optional.

4. Fill out the fields in the *FileCopy Transporter Details* panel.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Work Copy Path Prefix	Enter the prefix for the Work Copy Location (the location where the VCR interface places the checked out sources on the IKAN ALM Server so that it is accessible from a remote IKAN ALM Agent handling a build process). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. **Example:** • For a Windows Transporter: `//IKAN110/alm` In this sample "IKAN110" is the name of the IKAN ALM server, "alm" is the share name on that IKAN ALM server. If in the System Settings the Relative Work Copy location is defined as "workCopy", the IKAN ALM Agent will try to retrieve the sources from a directory under `\\IKAN110/alm/workCopy`. • For a Linux/Unix Transporter: `/mnt/ikan110/alm` This location is a mount of a directory on the IKAN ALM server (on the ikan110 machine) containing the Work Copy location. If in the System Settings the Relative Work Copy location is defined as "workCopy", the IKAN ALM Agent will try to retrieve the sources from a directory under `/mnt/ikan110/alm/workCopy`.
Build Archive Path Prefix	Enter the prefix for the Build Archive Location (the location where the Builds are stored or retrieved from by an IKAN ALM Agent handling a Build or Deploy process). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. For an example, refer to *Work Copy Path Prefix*.
Script Path Prefix	Enter the prefix for the Script Location (the location where the IKAN ALM Agent will retrieve the build or deploy script from, if it cannot be found in the sources). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. For an example, refer to *Work Copy Path Prefix*.
Phase Catalog Path Prefix	Enter the prefix for the Phase Catalog Location (the location on the IKAN ALM Server where imported and newly created Phases are stored). When an IKAN ALM Agent/Server needs to install a missing Phase, it uses this prefix and the Relative Phase Catalog Location defined in the System Settings to construct a path to copy the missing Phase from.

5. Once you have filled out the fields, click *Create*.

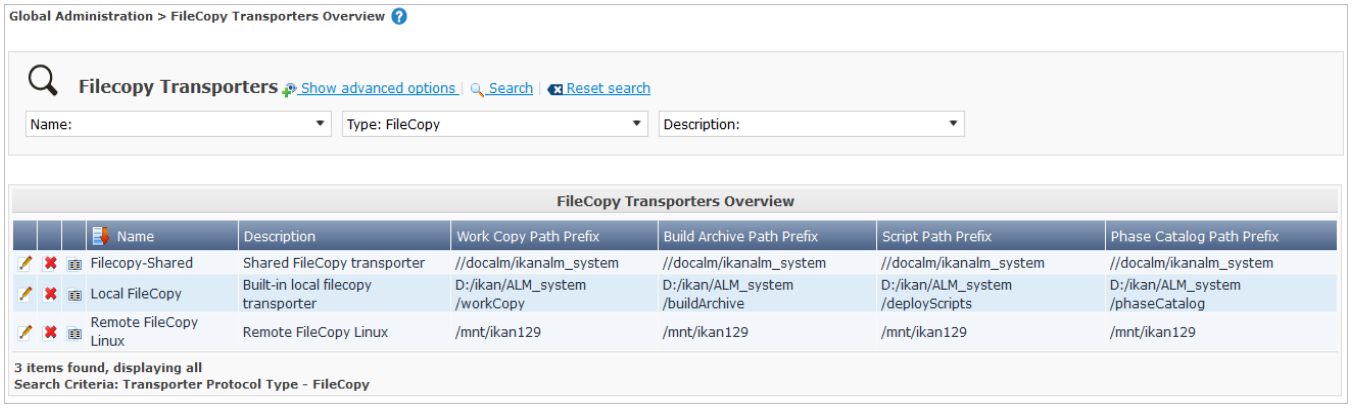
The newly created FileCopy Transporter Definition is added to the *FileCopy Transporters Overview* at the bottom of the screen.

RELATED TOPICS
- Transporters (page 307)
- System Settings (page 255)
- Machines (page 289)
- Build Environments (page 209)
- Deploy Environments (page 225)

The FileCopy Transporters Overview Screen

1. In the Global Administration context, select *Transporters> Overview*.
   The overview of all defined Transporter Protocols is displayed.

2. Specify *FileCopy* in the *Type* field on the search panel.
   The following screen is displayed:



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *FileCopy Transporters Overview* panel.
   For a detailed description of the fields, refer to Creating a FileCopy Transporter (page 311).

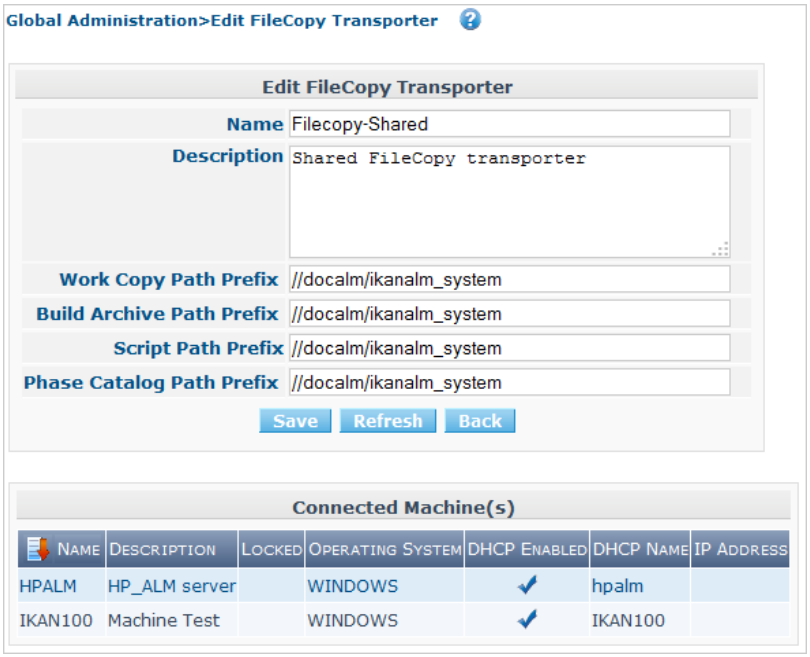
4. Depending on your access rights, the following links may be available on the *Machines Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a FileCopy Transporter definition. See Editing a File Copy Transporter Definition on page 314.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a FileCopy Transporter definition. See Deleting a File Copy Transporter Definition on page 314.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a FileCopy Transporter. See Viewing the File Copy Transporter History on page 315.

Editing a File Copy Transporter Definition

1. In the Global Administration context, select *Transporters > Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the FileCopy Transporter Protocols your are looking for.

2. Click the *Edit* link to change the selected FileCopy Transporter definition.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a FileCopy Transporter (page 311).

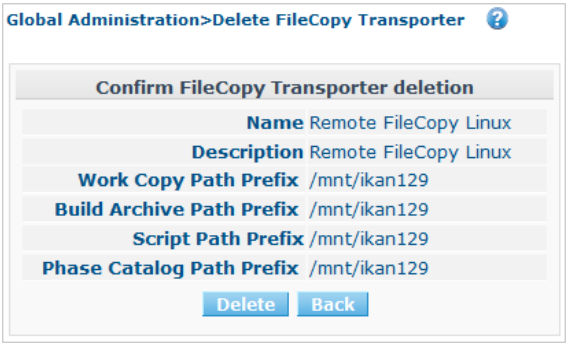
Note: The *Connected Machine(s)* panel displays the Machines the Transporter is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

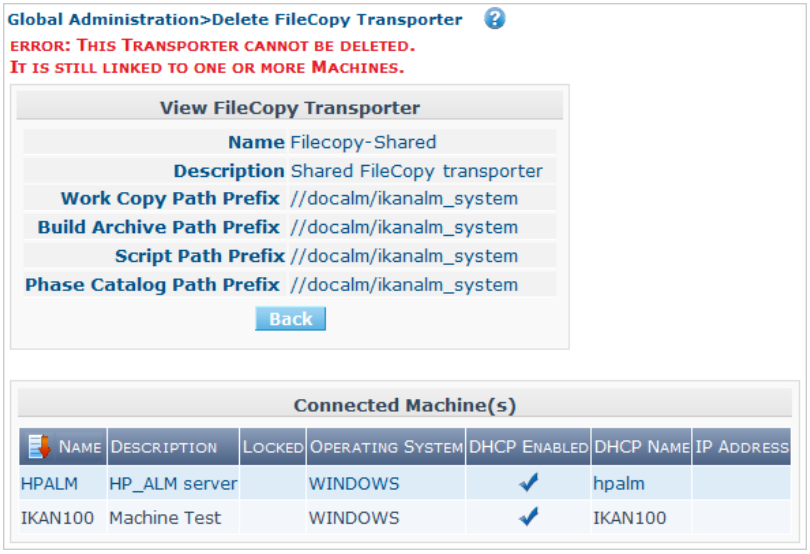
Deleting a File Copy Transporter Definition

1. In the Global Administration context, select *Transporters > Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the FileCopy Transporter Protocols your are looking for.

2. Click the *Delete* link to delete the selected FileCopy Transporter definition.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the FileCopy Transporter is linked to one or more Machines, the following screen is displayed:



   You must change the definition of the listed Machines, before you can delete the FileCopy Transporter.

Viewing the File Copy Transporter History

1. In the Global Administration context, select *Transporters > Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the FileCopy Transporter Protocols your are looking for.

2. Click the *History* link to display the *FileCopy Transporter History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *FileCopy Transporters Overview* screen.

RELATED TOPICS
- Transporters (page 307)
- System Settings (page 255)
- Machines (page 289)
- Build Environments (page 209)
- Deploy Environments (page 225)

28.4. FTP Transporters

The built-in FTP Transporter Client makes it possible to use FTP as a protocol for transporting Custom Phase, build results, scripts or sources between the IKAN ALM server and one or more Agents.

Note: An FTP Server must be installed on the IKAN ALM Server machine and must be correctly configured so that it controls the different System Settings locations (Work Copy, Build Archive, Phase Catalog and Script Locations).

In order to use the FTP Client for build and deploy actions, you must link the Transporter to a Machine representing an IKAN ALM Agent and link this Machine to the Build and Deploy Environments you want to "serve" via FTP.

Refer to the following sections for detailed information:

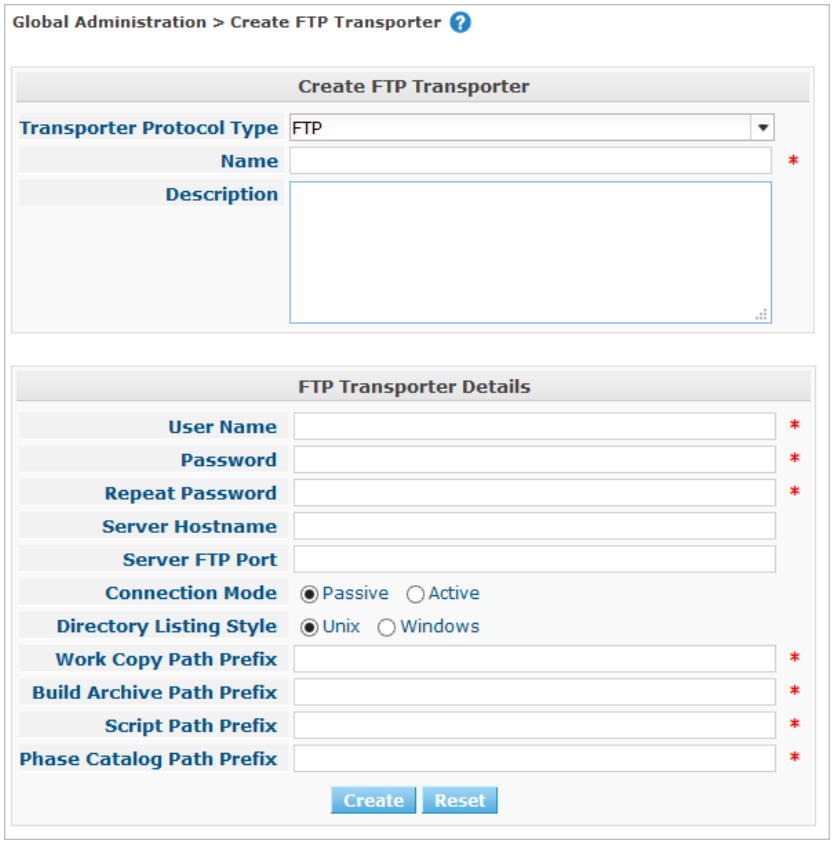
- Creating an FTP Transporter (page 316)
- The FTP Transporters Overview Screen (page 319)
- Editing an FTP Transporter Definition (page 320)
- Deleting an FTP Transporter Definition (page 321)
- Viewing the FTP Transporter History (page 322)

Creating an FTP Transporter

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Transporters* > *Create*.

2. Select *FTP* from the drop-down list in the *Type* field on the search panel.
   The following screen is displayed:



3. Fill out the fields in the *Create FTP Transporter* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	Enter the name of the new FTP Transporter in this field. This field is mandatory.
Description	Enter a description for the new FTP Transporter in this field. This field is optional.

4. Fill out the fields in the *FTP Transporter Details* panel.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
User Name	Enter the User Name in this field.
Password	Enter the Password in this field. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password in this field.

Field	Meaning
Server Hostname	This field is optional. It allows you to define a connection name that is different from the IKAN ALM Server Machine DHCP name or IP address. **Example:** FTP Server mapped to a hostname: `ftp.ikan.be`.
Server FTP Port	Enter the Port Number of the FTP Server running on the IKAN ALM Server. **Note:** The value entered in this field will override the value specified in the System Settings. See System Settings on page 255.If not specified in the System Settings nor on this screen, the default FTP Port Number `21` will be used.
Connection Mode	Select the connection mode: *Passive* or *Active*.
Directory Listing Style	Choose the Directory Listing Style: *Unix* or *Windows*.
Work Copy Path Prefix	Enter the prefix for the Work Copy Location (the location where the VCR interface places the checked out sources on the IKAN ALM Server so that it is accessible from a remote IKAN ALM Agent handling a build process). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. **Example:** If the FTP Server is configured to control the `E:/Ikan/alm/system` (Windows) or `/ikan/alm/system` (Linux/Unix) directory, a `/` will suffice as prefix. In that case, the sources will be copied to the Agent by a GET command from a subdirectory under `/workCopy` from the FTP Server.
Build Archive Path Prefix	Enter the prefix for the Build Archive Location (the location where the Builds are stored or retrieved from). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. **Example:** If the FTP Server is configured to control the `E:/Ikan/alm/system` (Windows) or `/ikan/alm/system` (Linux/Unix) directory, a `/` will suffice as prefix. In that case, the build result will be copied from the Agent to the Build Archive by a PUT command into the subdirectory `/buildArchive` on the FTP Server.
Script Path Prefix	Enter the prefix for the Script Location (the location where the IKAN ALM Agent will retrieve the build or deploy script from if it cannot be found in the sources). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. **Example:** If the FTP Server is configured to control the `E:/Ikan/alm/system` (Windows) or `/ikan/alm/system` (Linux/Unix) directory, a `/` will suffice as prefix.
Phase Catalog Path Prefix	Enter the prefix for the Phase Catalog Location (the location on the IKAN ALM Server where imported and newly created Phases are stored). When an IKAN ALM Agent/Server needs to install a missing Phase, it uses this prefix and the Relative Phase Catalog Location defined in the System Settings to construct a path to copy the missing Phase from.

5. Once you have filled out the fields, click *Create*.

   The newly created FTP Transporter Definition is added to the *FTP Transporters Overview* at the bottom of the screen.

RELATED TOPICS
- Transporters (page 307)
- System Settings (page 255)
- Machines (page 289)
- Build Environments (page 209)
- Deploy Environments (page 225)

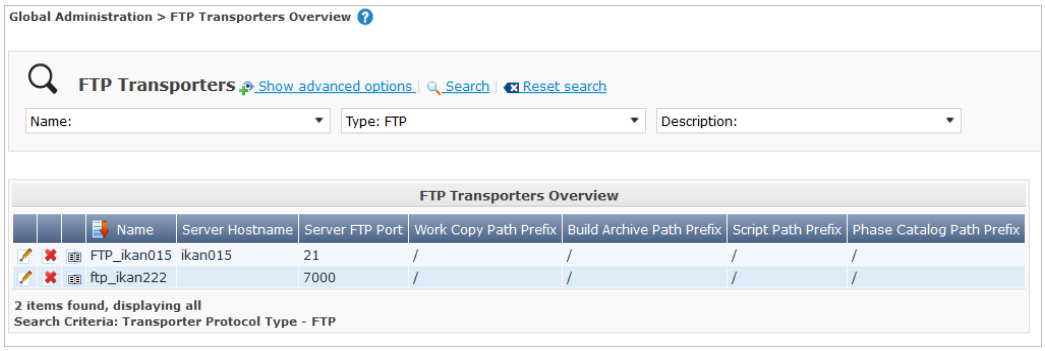
The FTP Transporters Overview Screen

1. In the Global Administration context, select *Transporters> Overview*.

   The overview of all defined Transporter Protocols is displayed.

2. Specify *FTP* in the *Type* field on the search panel.

   The following screen is displayed:



   If required, use the other search criteria to refine the items displayed on the overview.

   The following options are available:

   - the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *FTP Transporters Overview* panel.

   For a detailed description of the fields, refer to Creating an FTP Transporter (page 316).

4. Depending on your access rights, the following links may be available on the *Machines Overview* panel:

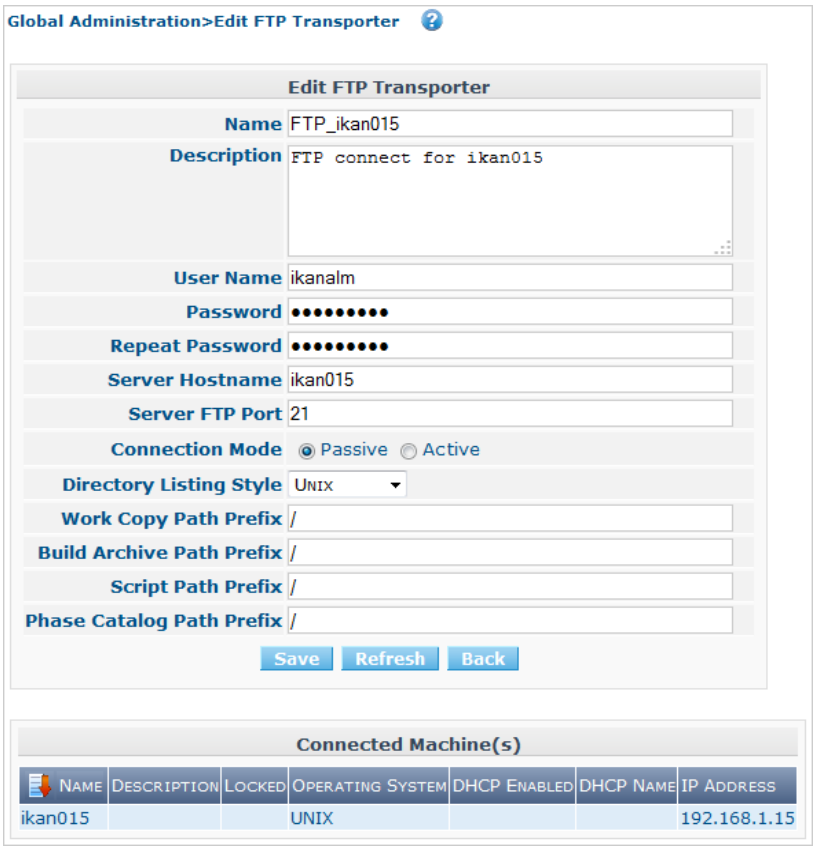
	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a FTP Transporter definition. See Editing an FTP Transporter Definition on page 320.

	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a FTP Transporter definition. See Deleting an FTP Transporter Definition on page 321.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a FTP Transporter. See Viewing the FTP Transporter History on page 322.

Editing an FTP Transporter Definition

1. In the Global Administration context, select *Transporters > Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the FTP Transporter Protocols your are looking for.

2. Click the *Edit* link to change the selected FTP Transporter definition.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating an FTP Transporter (page 316).

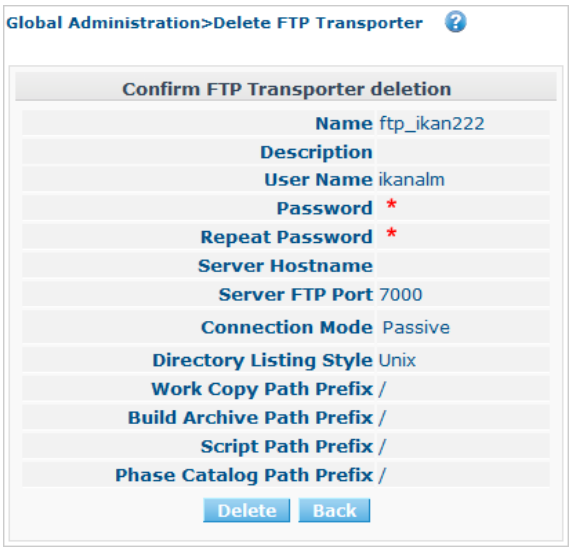
Note: The *Connected Machine(s)* panel displays the Machines the Transporter is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

Deleting an FTP Transporter Definition

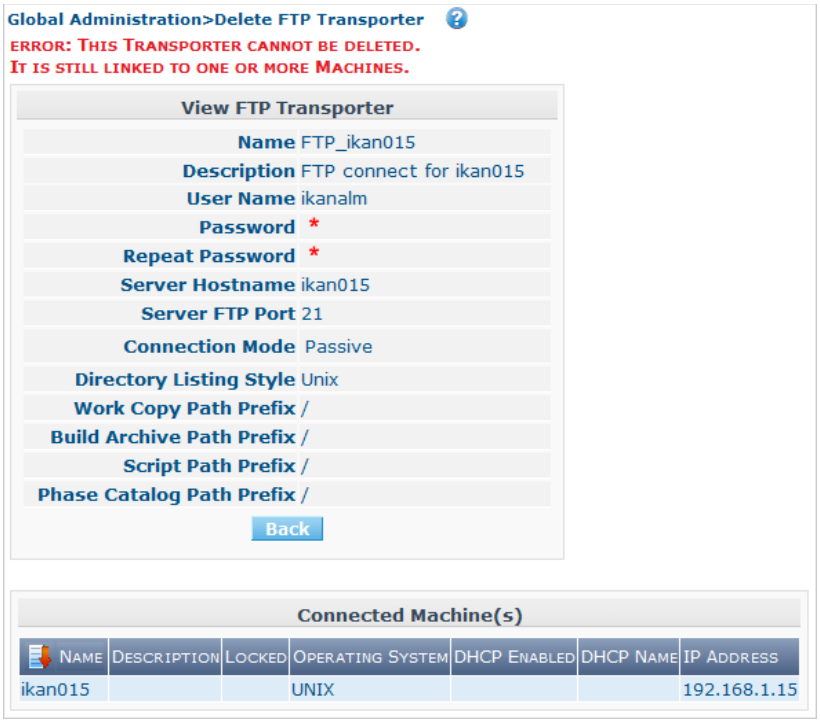
1. In the Global Administration context, select *Transporters > Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the FTP Transporter Protocols your are looking for.

2. Click the *Delete* link to delete the selected FTP Transporter definition.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.

Note: If the FTP Transporter is linked to one or more Machines, the following screen is displayed:



You must change the definition of the listed Machines, before you can delete the FTP Transporter.

Viewing the FTP Transporter History

1. In the Global Administration context, select *Transporters > Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the FTP Transporter Protocols your are looking for.

2. Click the *History* link to display the *FTP Transporter History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *FTP Transporters Overview* screen.

RELATED TOPICS
- Transporters (page 307)
- System Settings (page 255)
- Machines (page 289)
- Build Environments (page 209)
- Deploy Environments (page 225)

28.5. Secure Shell Transporters

Secure Shell Transporters allow you to securely transport Custom Phases, sources, scripts and build results between the IKAN ALM Server and an IKAN ALM Agent. The Secure Shell Transporter is part of the IKAN ALM Agent installation. Its configuration is explained in this section.

Note: An SSH Server must be installed on the IKAN ALM Server machine and must be correctly configured so that it controls the different System Settings locations (Work Copy, Build Archive, Phase Catalog and Script Locations).

In order to ensure a secure transport for build and deploy actions, you must link the Transporter to a Machine representing an IKAN ALM Agent and link that Machine to the Build and Deploy Environments you want to "serve" via a Secure Shell.

Refer to the following sections for detailed information:

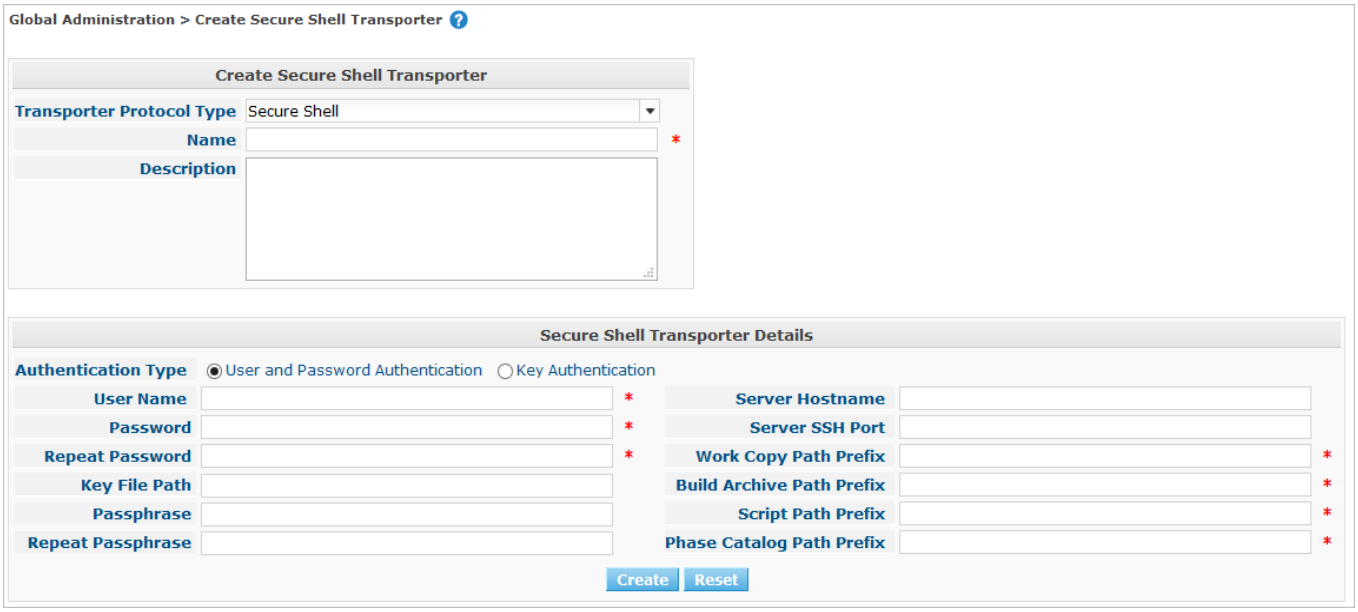
- Creating a Secure Shell Transporter (page 323)
- The Secure Shell Transporters Overview Screen (page 326)
- Editing a Secure Shell Transporter Definition (page 327)
- Deleting a Secure Shell Transporter Definition (page 328)
- Viewing the Secure Shell Transporter History (page 329)

Creating a Secure Shell Transporter

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Transporters* > *Create*.

2. Select *Secure Shell* from the drop-down list in the *Type* field on the search panel.
   The following screen is displayed:



3. Fill out the fields in the *Create Secure Shell Transporter* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	Enter the name of the new Secure Shell Transporter in this field. This field is mandatory.
Description	Enter a description for the new Secure Shell Transporter in this field. This field is optional.

4. Fill out the fields in the *Secure Shell Transporter Details* panel.

Fields marked with a red asterisk are mandatory.

Field	Meaning
Authentication type	Select the required Authentication type from the drop-down list. The following types are available: • User and Password Authentication • Key Authentication
User Name	Enter the User Name in this field, if *User and Password Authentication* is selected.
Password	Enter the Password in this field, if *User and Password Authentication* is selected. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password in this field.
Key File Path	Enter the path to the Key File in this field, if *Key Authentication* is selected.
Passphrase	Enter the Passphrase in this field if you use a Passphrase to protect your Key Authentication. The characters you enter are displayed as asterisks.
Repeat Passphrase	Re-enter the Passphrase in this field.
Server Hostname	This field is optional. It allows you to define the hostname of the IKAN ALM Server Machine in case it is different from the IKAN ALM Server Machine DHCP name or IP address.
Server SSH Port	Enter the Port Number of the SSH Server running on the IKAN ALM Server. **Note:** The value entered in this field will overrule the value specified in the System Settings. See System Settings on page 255.If not specified in the System Settings nor on this screen, the default SSH Port Number 22 will be used.

Field	Meaning
Work Copy Path Prefix	Enter the prefix for the Work Copy Location (the location where the VCR interface places the checked out sources on the IKAN ALM Server so that it is accessible from a remote IKAN ALM Agent handling a build process). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. **Example 1:** An IKAN ALM Server on a Windows machine with an SSH Server supporting the "cygdrive" notation. System Settings: the Relative Work Copy Location being defined as "workCopy" and the Local FileCopy Work Copy Location as `e:/ikan/alm/workCopy` results in `/cygdrive/e/ikan/alm` as the Work Copy Path Prefix. The IKAN ALM Agent will try to Secure Copy the sources from a location under `/cygdrive/e/ikan/alm/workCopy`. **Example 2:** An IKAN ALM Server on a Linux machine. System Settings: the Local FileCopy Work Copy location being defined as `/opt/ikan/alm/workCopy` and the Relative Work Copy Location as "workCopy" results in`/opt/ikan/alm` as Work Copy Path Prefix. The IKAN ALM Agent will try to Secure Copy the sources from a location under `/opt/ikan/alm/workCopy`.
Build Archive Path Prefix	Enter the prefix for the Build Archive Location (the location where the Builds are stored or retrieved from). The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. For an example, refer to *Work Copy Path Prefix*.
Script Path Prefix	Enter the prefix for the Script Location. The Server Location will be formed by concatenating this prefix together with the Relative Location defined in the System Settings. See System Settings on page 255. For an example, refer to *Work Copy Path Prefix*.
Phase Catalog Path Prefix	Enter the prefix for the Phase Catalog Location (the location on the IKAN ALM Server where imported and newly created Phases are stored). When an IKAN ALM Agent/Server needs to install a missing Phase, it uses this prefix and the Relative Phase Catalog Location defined in the System Settings to construct a path to copy the missing Phase from.

5. Once you filled out the fields as required, click *Create*.

The newly created Secure Shell Transporter is added to the *Secure Shell Transporter Overview* at the bottom of the screen.

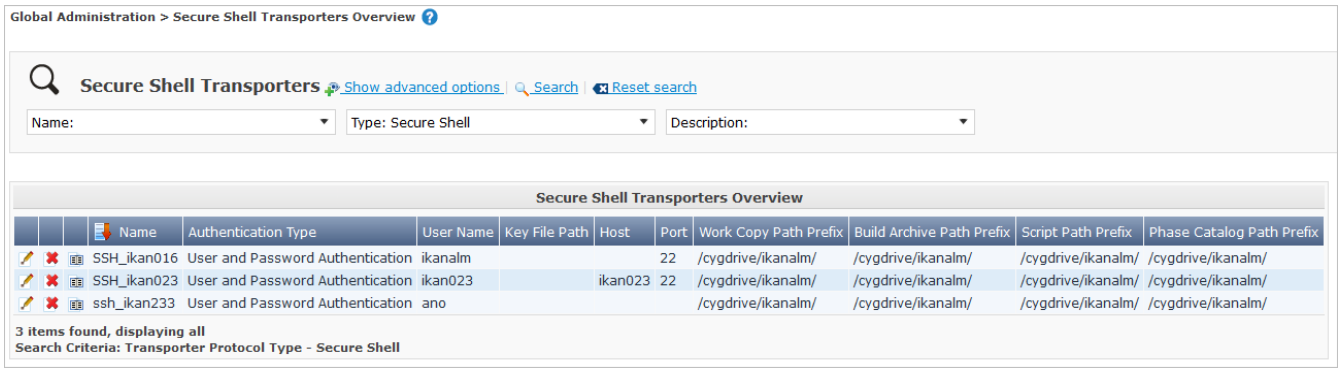
RELATED TOPICS

- Transporters (page 307)
- System Settings (page 255)
- Machines (page 289)
- Build Environments (page 209)
- Deploy Environments (page 225)

The Secure Shell Transporters Overview Screen

1. In the Global Administration context, select *Transporters> Overview*.
   The overview of all defined Transporter Protocols is displayed.

2. Specify *Secure Shell* in the *Type* field on the search panel.
   The following screen is displayed:



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *Secure Shell Transporters Overview* panel
   For a detailed description of the fields, refer to Creating a Secure Shell Transporter (page 323).

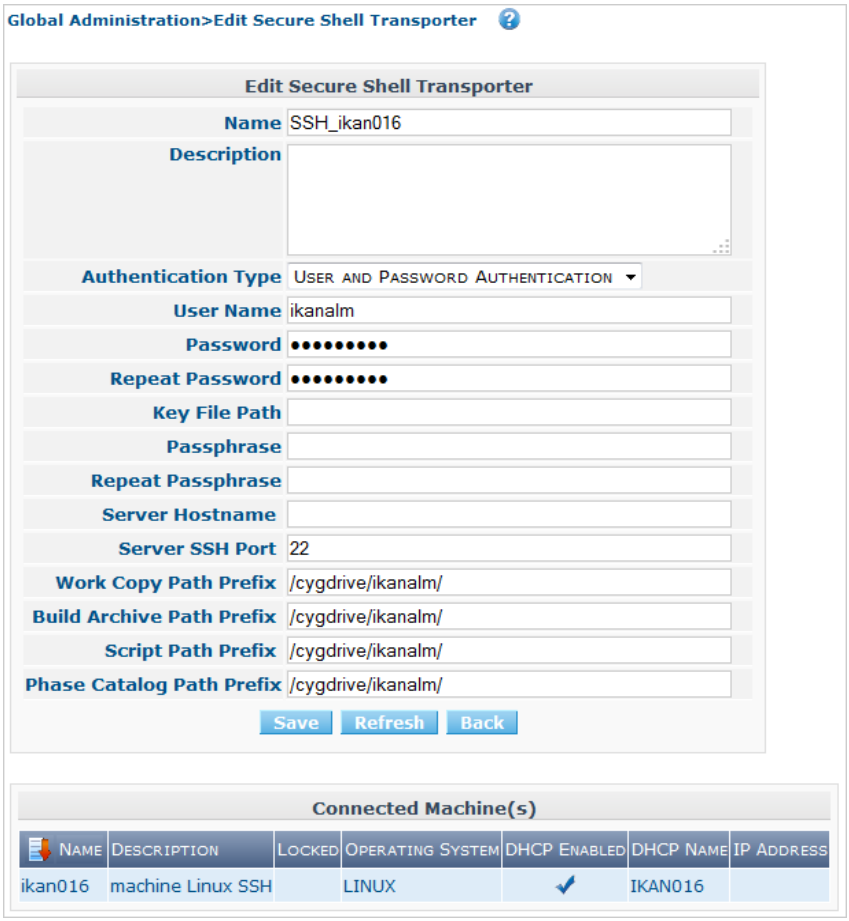
4. Depending on your access rights, the following links may be available on the *Machines Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Secure Shell Transporter definition. See Editing a Secure Shell Transporter Definition on page 327.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Secure Shell Transporter definition. See Deleting a Secure Shell Transporter Definition on page 328.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Secure Shell Transporter. See Viewing the Secure Shell Transporter History on page 329.

Editing a Secure Shell Transporter Definition

1. In the Global Administration context, select *Transporters > Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the Secure Shell Transporter Protocols your are looking for.

2. Click the *Edit* link to change the selected Secure Shell Transporter.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a Secure Shell Transporter (page 323).

> **Note:** The *Connected Machine(s)* panel displays the Machines the Transporter is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

Deleting a Secure Shell Transporter Definition

1. In the Global Administration context, select *Transporters* > *Overview*.
   The overview of all defined Transporter Protocols is displayed.
   Use the search criteria on the search panel to display the Secure Shell Transporter Protocols your are looking for.

2. Click the *Delete* link to delete the selected Secure Shell Transporter.
   If the Secure Shell Transporter is not linked to any Machine, the following screen is displayed:

3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.

Note: If the Secure Shell Transporter is linked to one or more Machines, the following screen is displayed:

You must change the definition of the listed Machines, before you can delete the Secure Shell Transporter.

Viewing the Secure Shell Transporter History

1. In the Global Administration context, select *Transporters > Overview*.

   The overview of all defined Transporter Protocols is displayed.

   Use the search criteria on the search panel to display the Secure Shell Transporter Protocols your are looking for.

2. Click the *History* link to display the *Secure Shell Transporter History View*.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Secure Shell Transporters Overview* screen.

RELATED TOPICS

- Transporters (page 307)
- System Settings (page 255)
- Machines (page 289)
- Build Environments (page 209)
- Deploy Environments (page 225)

CHAPTER 29

Version Control Repositories

IKAN ALM currently supports different types of Version Control Repositories (VCR):

- Subversion Repositories (page 332)
- Git Repositories (page 340)
- CVS Repositories (page 346)
- Visual SourceSafe (VSS) Repositories (page 353)
- ClearCase Repositories (page 360)
- PVCS Repositories (page 367)
- TFVC Version Control Repositories (page 373)

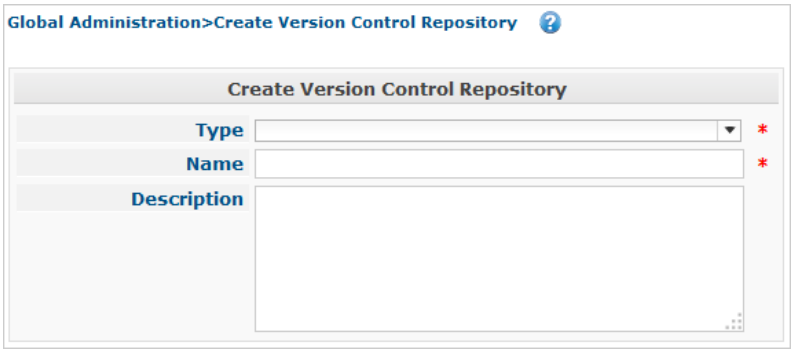
The following sections explain how to define a VCR Repository so that the IKAN ALM Server can connect to it.

Note: You must always install a VCR client on the IKAN ALM Server in order to interact with the VCR Repository.

29.1. Creating a Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Version Control Repositories > Create*.
   The following screen is displayed:



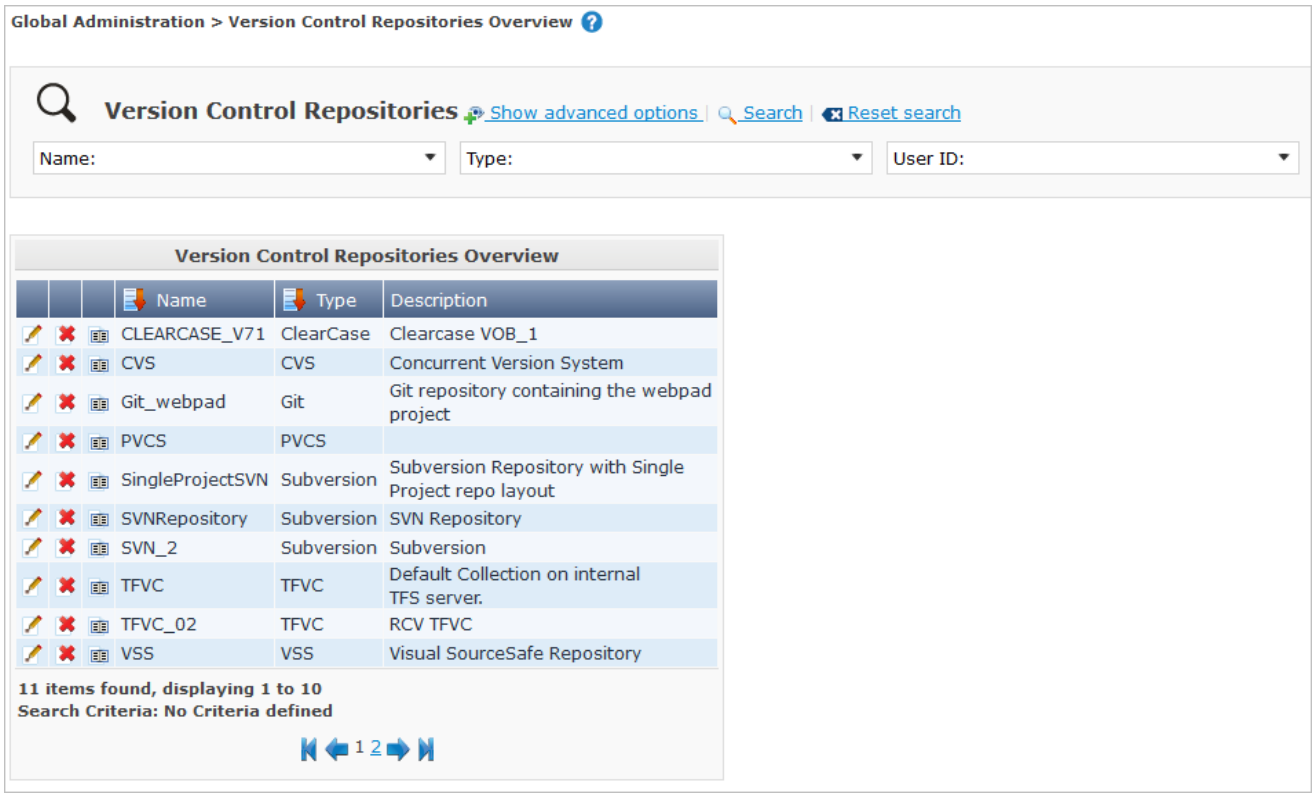
2. Select the *Type* of Repository you want to create.
   The *Connection Details* panel for the type of Repository you selected is displayed underneath.

3. Enter the *Name* of the Repository and add a *Description*.

4. Fill out the repository-specific Connection Details.
   For more information on those details, refer to the appropriate sections:
   - Creating a Subversion Repository (page 332)
   - Creating a Git Repository (page 340)
   - Creating a CVS Repository (page 346)
   - Creating a VSS Version Control Repository (page 353)
   - Creating a ClearCase Repository (page 360)
   - Creating a PVCS Repository (page 367)
   - Creating a PVCS Repository (page 367)

29.2. The Repository Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields,

3. Verify the information on the *Repositories Overview* panel.

4. Depending on your access rights, the following links may be available on the *Version Control Repositories Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Repository definition.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Repository definition.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Repository.

For more information, refer to the appropriate sections:

- Subversion Repositories (page 332)
- Git Repositories (page 340)
- CVS Repositories (page 346)
- Visual SourceSafe (VSS) Repositories (page 353)
- ClearCase Repositories (page 360)
- PVCS Repositories (page 367)
- TFVC Version Control Repositories (page 373)

29.3. Subversion Repositories

Refer to the following sections for detailed information:

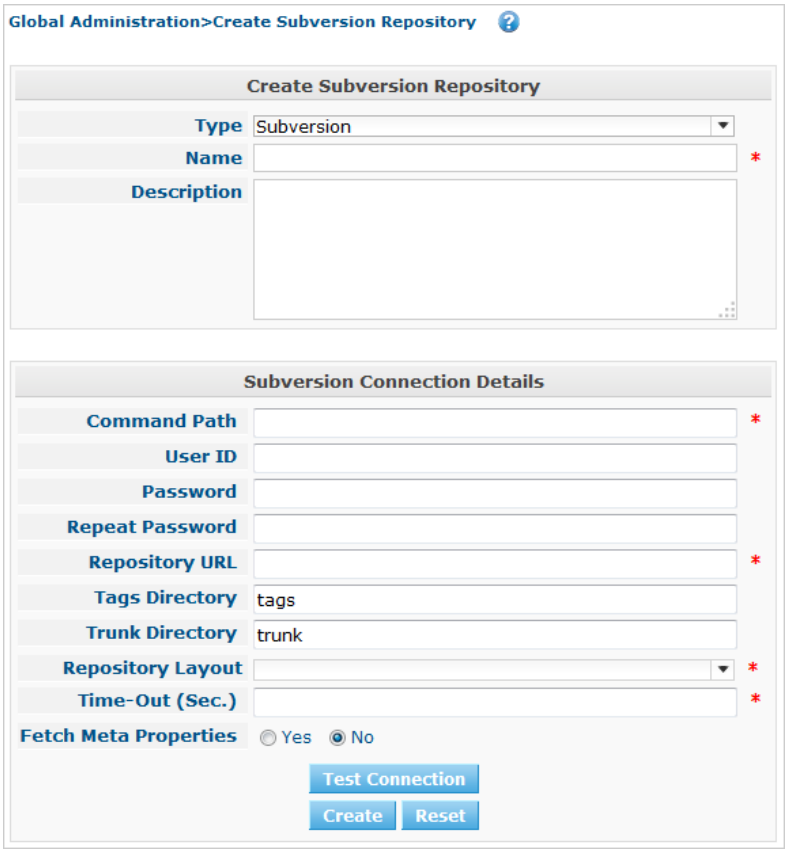
- Creating a Subversion Repository (page 332)
- The Subversion Repositories Overview Screen (page 336)
- Editing a Subversion Repository (page 337)
- Deleting a Subversion Repository (page 338)
- Viewing the Subversion Repository History (page 339)

Creating a Subversion Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Version Control Repositories* > *Create*.

2. Select *Subversion* from the drop-down list in the *Type* field.
   The following screen is displayed:



3. Fill out the fields in the *Create Subversion Repository* screen.
   Fields marked with a red asterisk are mandatory:

Field	Meaning
Type	Select the type of Version Control Repository you want to define. This field is mandatory. After you have selected the VCR type, the appropriate *Connection Details* panel will be displayed underneath.
Name	Enter the name of the new Subversion Repository definition in this field. This field is mandatory.
Description	Enter a description for the new Subversion Repository in this field. This field is optional.

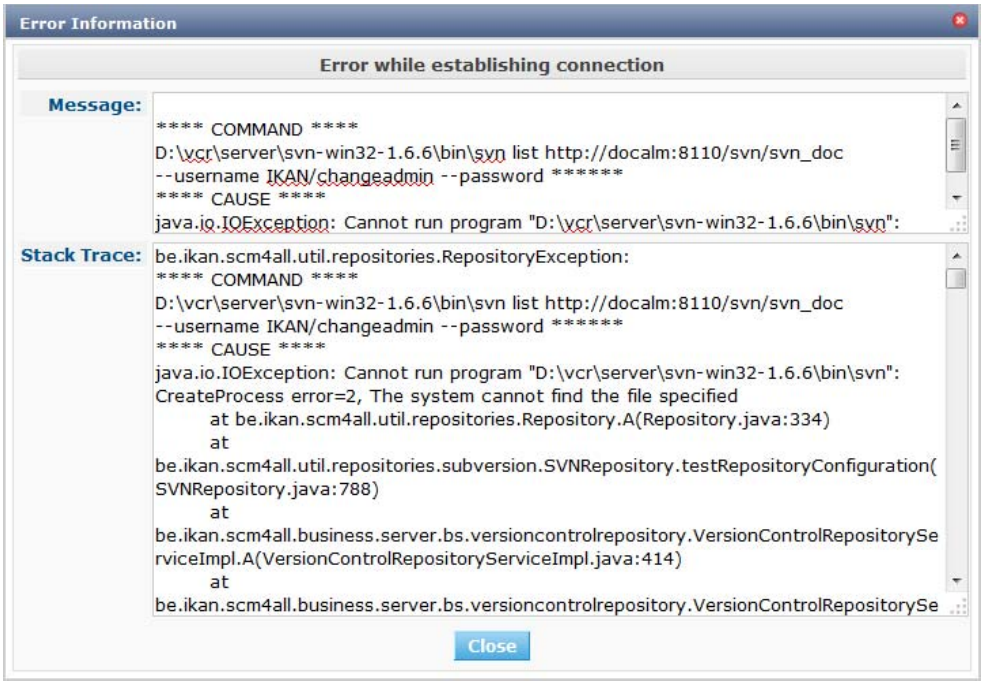
4. Fill out the fields in the *Subversion Connection Details* panel.
   Fields marked with a red asterisk are mandatory:

Field	Meaning
Command Path	Enter the path to the Location where the Subversion Client command (`svn.exe` or `svn`) resides. This field is mandatory.
User ID	Enter the User ID that IKAN ALM will use to access the Subversion Repository. This field is optional.
Password	Enter the Password that IKAN ALM will use to access the Subversion Repository. This field is optional. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password that IKAN ALM will use to access the Subversion Repository.
Repository URL	Enter the URL of the machine hosting the Subversion Repository. This field is mandatory. For more information regarding a correct Subversion Repository URL, refer to the section Subversion - General Information (page 511).
Tags Directory	Enter the Tags Directory for the Subversion Repository or accept the default setting.
Trunk Directory	Enter the Trunk Directory for the Subversion Repository or accept the default setting.
Repository Layout	Select the required Repository Layout option from the drop-down list. The following options are available: • Project-oriented • Repository-oriented • Single Project-oriented The selection of a Repository Layout is mandatory. For more information regarding the different Repository Layouts, refer to the section Subversion - General Information (page 511).
Time-Out (Sec.)	Enter the Time-Out value in seconds. If IKAN ALM is not able to establish a connection with the Subversion Repository within the defined period, it will consider the Repository to be inaccessible. The definition of a Time-Out value is mandatory.
Fetch Meta Properties	Select the *Yes* option button to automatically retrieve the Meta Properties set on source files in the Subversion VCR during the *Retrieve code* Phase. These Meta Properties can be used by the Build and Deploy Scripting Tool. For more information regarding fetching Meta Properties, refer to the section Subversion - General Information (page 511).

5. Click *Test Connection* to verify if IKAN ALM can establish a connection to the Subversion Repository. If the test is successful, the following message is displayed:

   Info: Could successfully establish a connection with the Repository.

If the test is not successful, the following screen is displayed:



Correct the errors reported in the Stack Trace field and perform the test again.

6. Once the test is successful, click *Create*.

The newly created Subversion Repository is added to the *Subversion Repositories Overview* at the bottom of the screen.

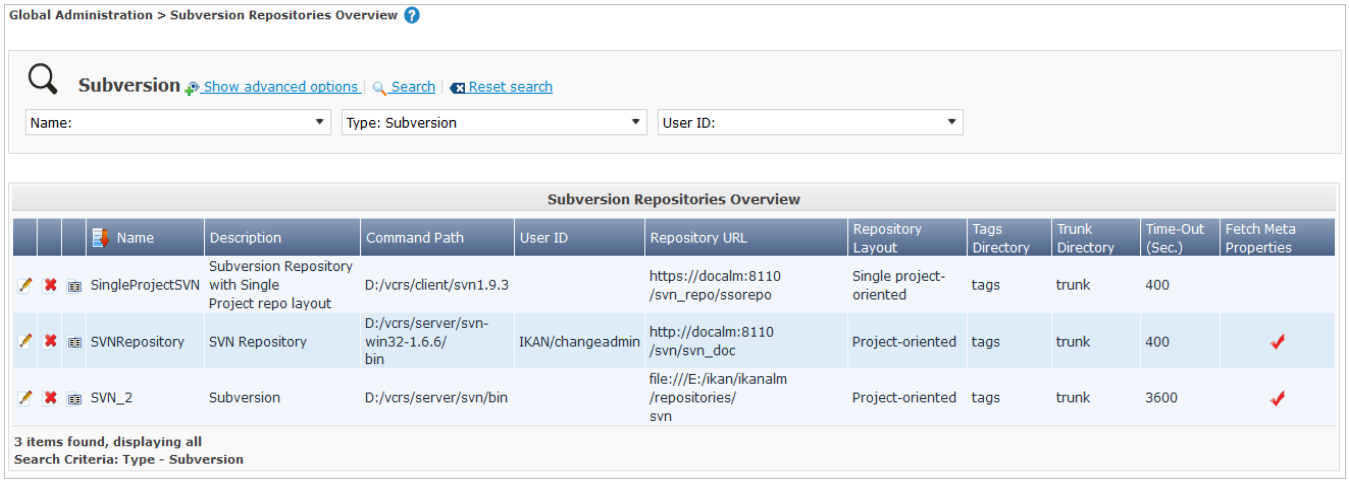
RELATED TOPICS

- Version Control Repositories (page 330)
- Editing Project Settings (page 149)
- Creating a Project (page 461)

The Subversion Repositories Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.

2. Specify *Subversion* in the *Type* field on the *Search Version Control Repositories* panel.
   Use the other search criteria to only display the Subversion Repositories you are looking for.



If required, use the other search criteria to refine the items displayed on the overview.
The following options are available:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- the *Search* link to refresh the list based on the current search criteria,
- the *Reset search* link to clear the search fields.

3. Verify the information on the *Subversion Repositories Overview* panel.
   For a detailed description of the fields, refer to Creating a Subversion Repository (page 332).

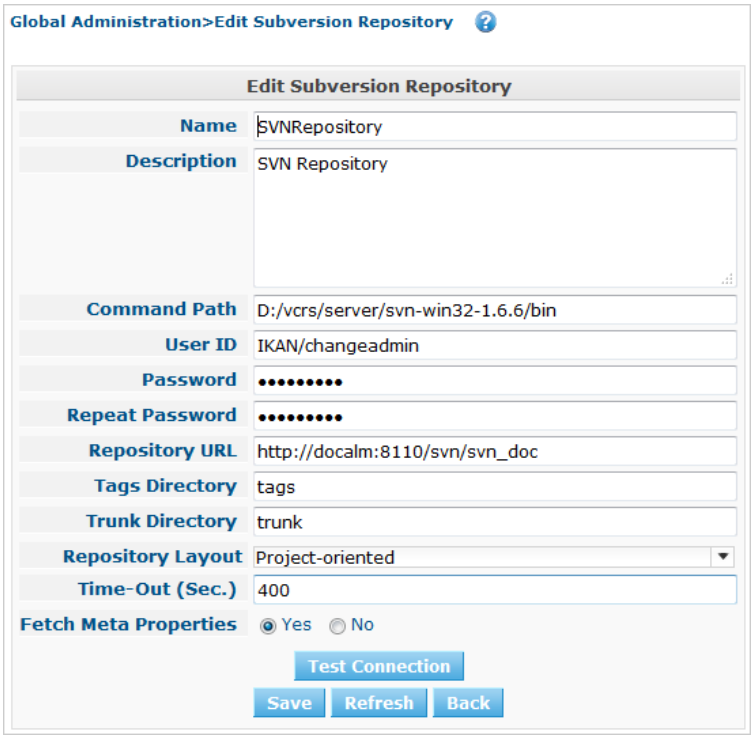
4. Depending on your access rights, the following links may be available on the *Subversion Repositories Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Subversion Repository definition. See Editing a Subversion Repository on page 337.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Subversion Repository definition. See Deleting a Subversion Repository on page 338.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Subversion Repository. See Viewing the Subversion Repository History on page 339.

Editing a Subversion Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the Subversion Repositories you are looking for.

2. Click the *Edit* link to change the selected Subversion Repository.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a Subversion Repository (page 332).

Note: Click *Test Connection* to verify if IKAN ALM can establish a connection to the Subversion Repository.
The *Connected Projects* panel displays the Projects the Repository is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

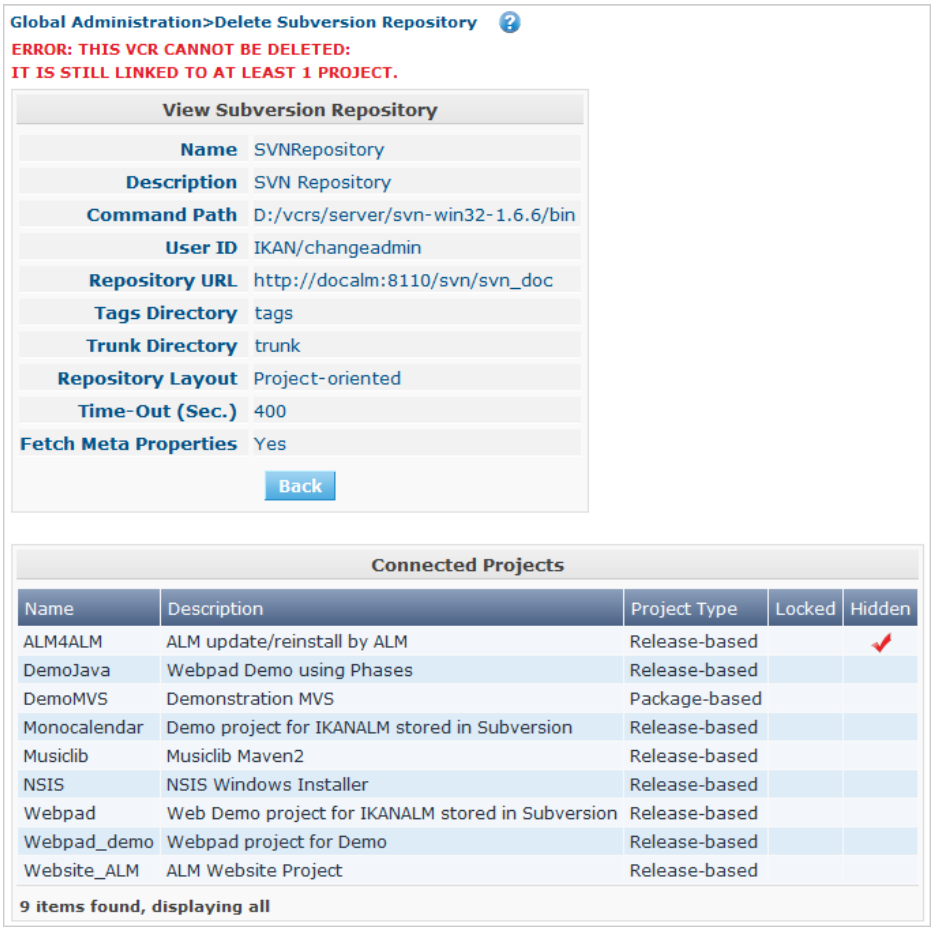
Deleting a Subversion Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the Subversion Repositories you are looking for.

2. Click the *Delete* link to delete the selected Subversion Repository.
   If the Subversion Repository is not connected to any Project, the following screen is displayed:

3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.

Note: If the Subversion Repository is connected to one or more Projects, the following screen is displayed:



Before deleting the VCR, you must connect the reported Projects to another VCR.

Viewing the Subversion Repository History

1. In the Global Administration context, select *Version Control Repositories > Overview*.

   The overview of all defined Version Control Repositories is displayed.

   Use the search criteria on the *Search Version Control Repository* panel to only display the Subversion Repositories you are looking for.

2. Click the *History* link to display the *Subversion Repository History View*.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Subversion Repositories Overview* screen.

29.4. Git Repositories

Refer to the following sections for detailed information:

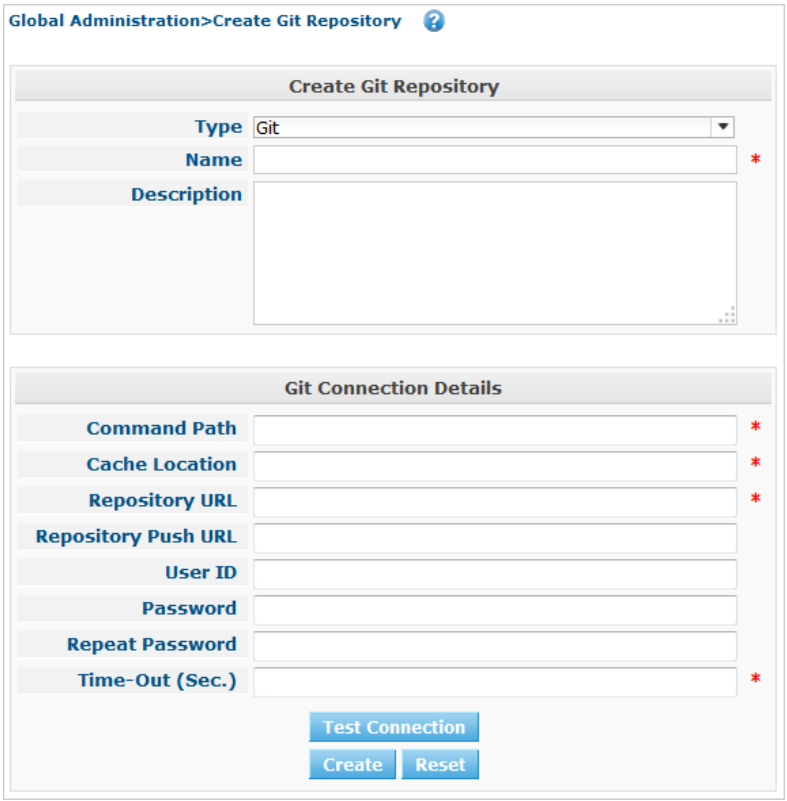
- Creating a Git Repository (page 340)
- The Git Repositories Overview Screen (page 343)
- Editing a Git Repository (page 344)
- Deleting a Git Repository (page 345)
- Viewing the Git Repository History (page 346)

Creating a Git Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM. Before you can create the Git Version Control Repository in IKAN ALM, you must install a Git Client on the IKAN ALM Server.

1. In the Global Administration context, select *Version Control Repositories > Create*.

2. Select *Git* from the drop-down list in the *Type* field.
   The following screen is displayed:



3. Fill out the fields in the *Create Git Repository* screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Type	Select the type of Version Control Repository you want to define. This field is mandatory. After you have selected the VCR type, the appropriate *Connection Details* panel will be displayed underneath.
Name	Enter the name of the new Git Repository definition in this field. This field is mandatory.
Description	Enter a description for the new Git Repository in this field. This field is optional.

4. Fill out the fields in the *Git Connection details* panel. Fields marked with a red asterisk are mandatory:

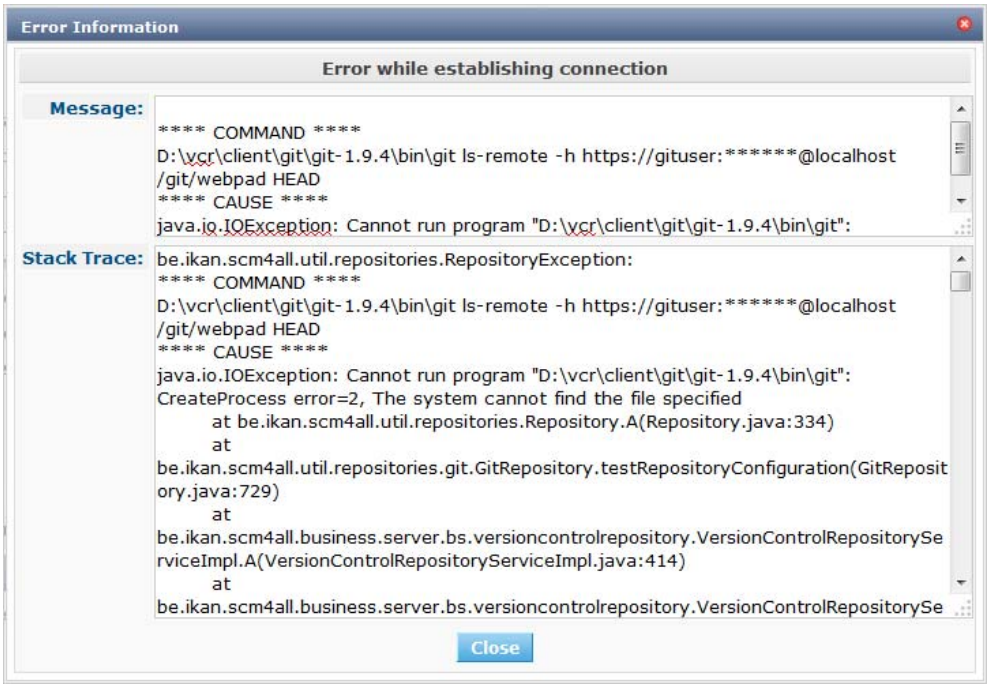
Field	Meaning
Command Path	Enter the path to the Location where the Git Client command (git or git.exe) resides. This field is mandatory.
Cache Location	Enter the path to the Cache location for this Git Repository. This directory on the IKAN ALM Server is used to clone and cache the Git repository for the IKAN ALM Server and Web Application in order to speed up the Repository processes. Make sure that the access rights on this location are correctly configured for the Git process. A sample location might be ALM_HOME/system/gitcache, e.g., `c:/ALM/system/gitcache`. It's possible to share the Cache Location among different Git Repositories.
Repository URL	Enter the URL of the Git Repository. This field is mandatory. Valid URLs are of the following format: • /path/to/repo.git • file:///path/to/repo.git • ssh://[user @]host.xz[:port]/path/to/repo.git • [user@]host.xz:path/to/repo.git • git://host.xz[:port]/path/to/repo.git • http[s]://host.xz[:port]/path/to/repo.git **Warning:** If you provide a user and, optionally, also a password in the dedicated input fields below, do not add them to the Repository (Push) URL, since IKAN ALM will insert those values in the final (Push) URL before executing any Repository command.
Repository Push URL	In case you want to use different protocols for Git read and push actions, you can specify a different URL (usually a protocol that demands authentication, ssh://, https:// or scp style URL) in this field for the push actions. Refer to the description of the *Repository URL* for valid URL formats. This field is optional.
User ID	Enter the User ID that IKAN ALM will use to access the Git Repository. This field is optional. IKAN ALM will insert the value of the User ID in the final (Push) URL before executing any Repository command.

Field	Meaning
Password	Enter the Password that IKAN ALM will use to access the Git Repository. This field is optional. The characters you enter are displayed as asterisks. IKAN ALM will insert the value of the Password in the final (Push) URL before executing any Repository commands. This is only the case for http(s) URLs. SSH and scp URLs need to function without providing a password.
Repeat Password	Re-enter the Password that IKAN ALM will use to access the Git Repository.
Time-Out (Sec.)	Enter the Time-Out value in seconds. If IKAN ALM is not able to establish a connection with the Git Repository within the defined period, it will consider the Repository to be inaccessible. The definition of a Time-Out value is mandatory.

5. Click *Test Connection* to verify if IKAN ALM can establish a connection to the Git Repository. If the test is successful, the following message is displayed:

   Info: Could successfully establish a connection with the Repository.

   If the test is not successful, the following screen is displayed:



   Correct the errors reported in the Stack Trace field and perform the test again.

6. Once the test is successful, click *Create*.

   The newly created Git Repository is added to the *Git Repositories Overview* at the bottom of the screen.

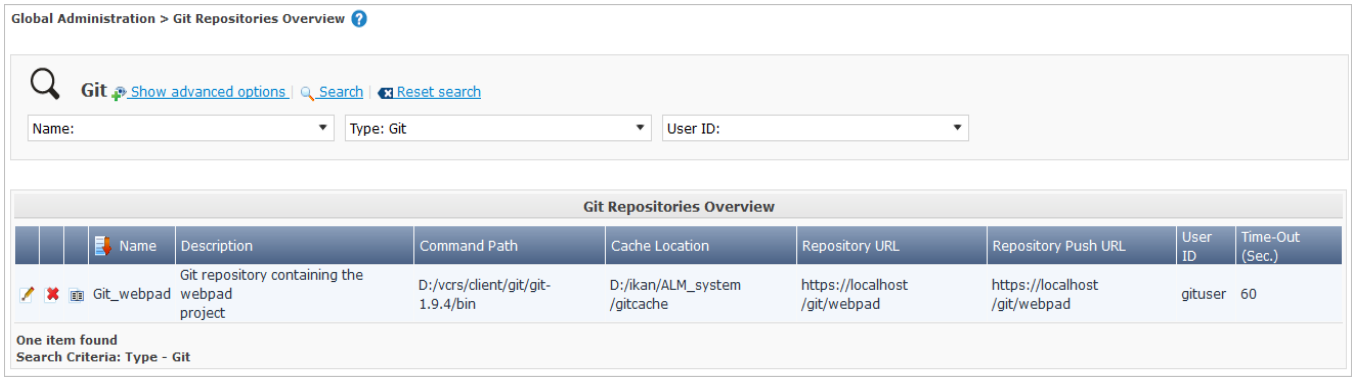
RELATED TOPICS

- Version Control Repositories (page 330)
- Editing Project Settings (page 149)
- Creating a Project (page 461)

The Git Repositories Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.

2. Specify *Git* in the *Type* field on the *Search Version Control Repositories* panel.
   Use the other search criteria to only display the Git Repositories you are looking for.



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *Git Repositories Overview* panel.
   For a detailed description of the fields, refer to Creating a Subversion Repository (page 332).

4. Depending on your access rights, the following links may be available on the *Git Repositories Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Git Repository definition. See Editing a Subversion Repository on page 337.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Git Repository definition. See Deleting a Subversion Repository on page 338.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Git Repository. See Viewing the Subversion Repository History on page 339.

Editing a Git Repository

1. In the Global Administration context, select *Version Control Repositories* > *Overview* on the Submenu.

   The overview of all defined Version Control Repositories is displayed.

   Use the search criteria on the *Search Version Control Repository* panel to only display the Git Repositories you are looking for.

2. Click the *Edit* link to change the selected Git Repository.

   The following screen is displayed:

3. Edit the fields as required.

   For a description of the fields, refer to Creating a Git Repository (page 340).

Note: Click *Test Connection* to verify if IKAN ALM can establish a connection to the Git Repository. The *Connected Projects* panel displays the Projects the Repository is linked to.

4. Click *Save* to save your changes.

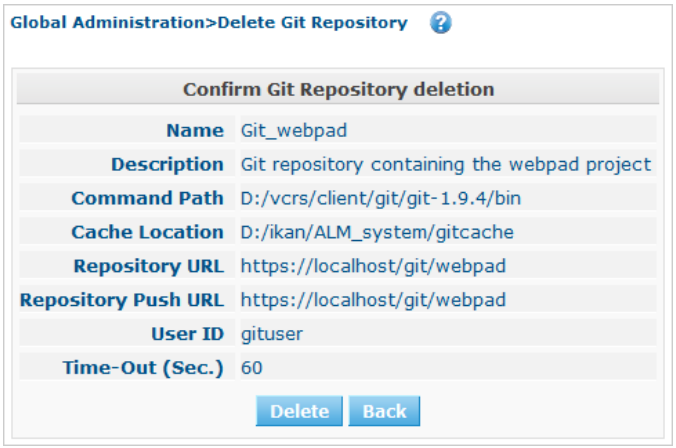
   You can also click:

   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

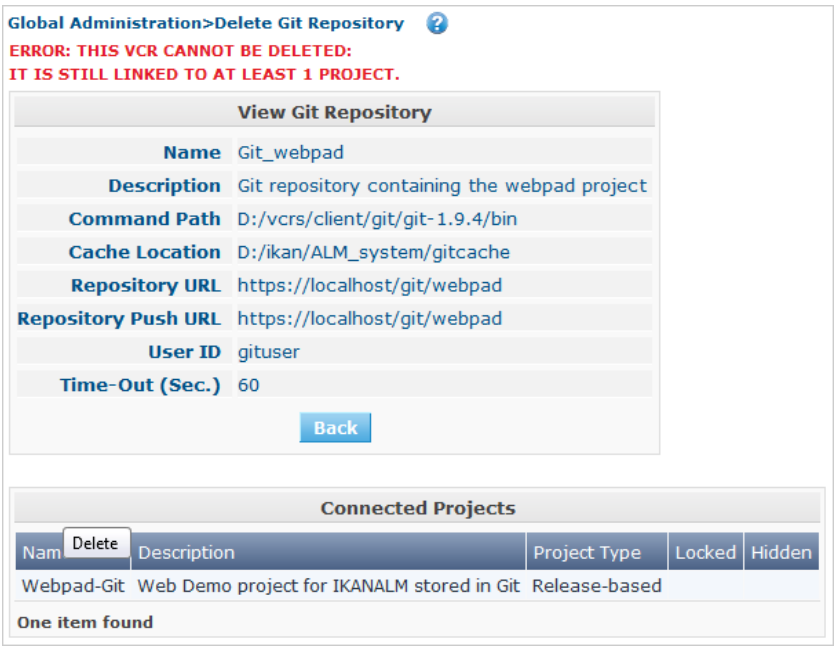
Deleting a Git Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the Git Repositories you are looking for.

2. Click the *Delete* link to delete the selected Git Repository.
   If the Git Repository is not connected to any Project, the following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the Git Repository is connected to one or more Projects, the following screen is displayed:



   Before deleting the VCR, you must connect the reported Projects to another VCR.

Viewing the Git Repository History

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the Git Repositories you are looking for.

2. Click the *History* link to display the *Git Repository History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Git Repositories Overview* screen.

29.5. CVS Repositories

Refer to the following sections for detailed information:

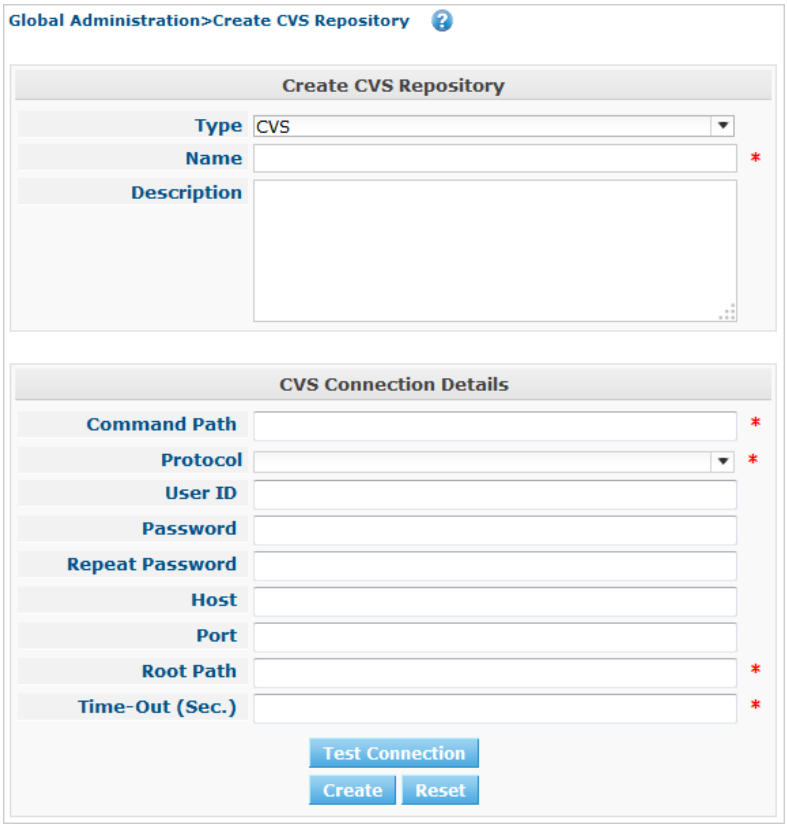
- Creating a CVS Repository (page 346)
- The CVS Repositories Overview Screen (page 349)
- Editing a CVS Repository (page 350)
- Deleting a CVS Repository (page 351)
- Viewing the CVS Repository History (page 353)

Creating a CVS Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.
Before you can create the CVS Version Control Repository in IKAN ALM, you must install a CVS Client on the IKAN ALM Server.

1. In the Global Administration context, select *Version Control Repositories > Create*.

2. Select *CVS* from the drop-down list in the *Type* field.
   The following screen is displayed:



3. Fill out the fields in the *Create CVS Repository* screen.
   Fields marked with a red asterisk are mandatory:

Field	Meaning
Type	Select the type of Version Control Repository you want to define. This field is mandatory. After you have selected the VCR type, the appropriate *Connection Details* panel will be displayed underneath.
Name	Enter the name of the new CVS Repository definition in this field. This field is mandatory.
Description	Enter a description for the new CVS Repository in this field. This field is optional.

4. Fill out the fields in the *CVS Connection details* panel.

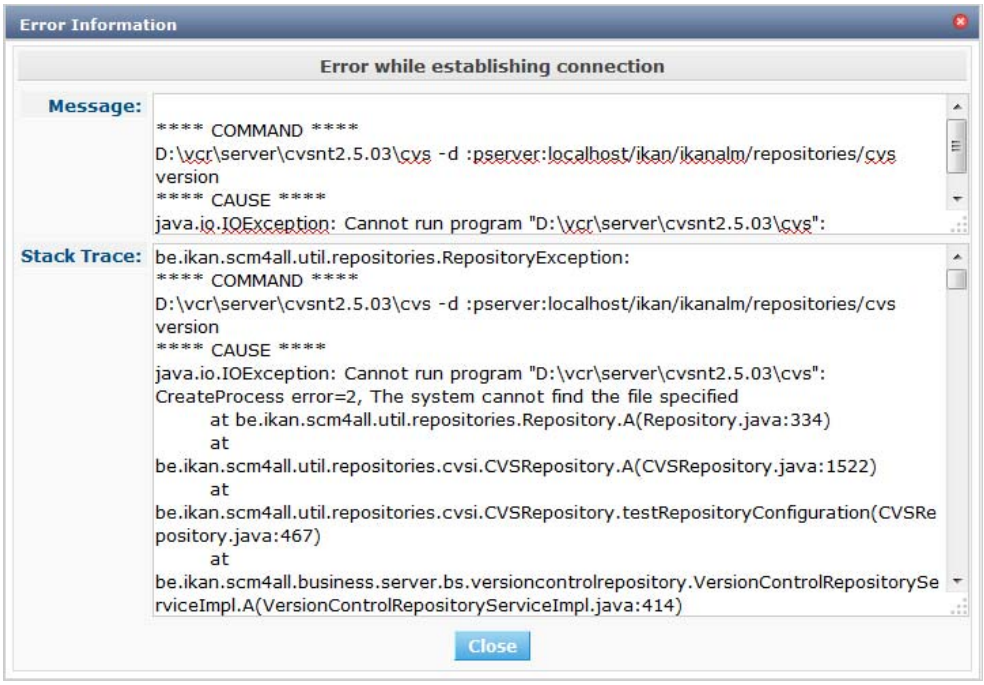
Fields marked with a red asterisk are mandatory:

Field	Meaning
Command Path	Enter the path to the Location where the CVS Client command (`cvs.exe` or `cvs`) resides on the IKAN ALM Server.
Protocol	Select the Protocol from the drop-down list. This is the Protocol that will be used to connect to the CVS Repository. The following Protocols are available: • local • pserver • rhosts • ntserver • gserver • sspi • server • ssh • ext
User ID	Enter the User ID that IKAN ALM will use to access the CVS Repository.
Password	Enter the Password that IKAN ALM will use to access the CVS Repository. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password that IKAN ALM will use to access the CVS Repository.
Host	Enter the Machine name hosting the CVS Repository. This field is mandatory, except if the *local* protocol is used.
Port	Enter the Port number used to access the CVS Repository. This field may remain empty if the *local* Protocol is used, or if the default port number 2401 is used.
Root Path	Enter the repository CVS ROOT used to log in to CVS. This is the location containing the CVSROOT directory. For instance, if CVSROOT is located under *E:/cvs/repository/CVSROOT*, the Root Path is *E:/cvs/repository*. This field is mandatory.
Time-Out (Sec.)	Enter the Time-Out value in seconds. If IKAN ALM is not able to establish a connection with the CVS Repository within the defined period, it will consider the Repository to be inaccessible. The definition of a Time-Out value is mandatory.

5. Click *Test Connection* to verify if IKAN ALM can establish a connection to the CVS Repository. If the test is successful, the following message is displayed:

   Info: Could successfully establish a connection with the Repository.

If the test is not successful, the following screen is displayed:



Correct the errors reported in the Stack Trace field and perform the test again.

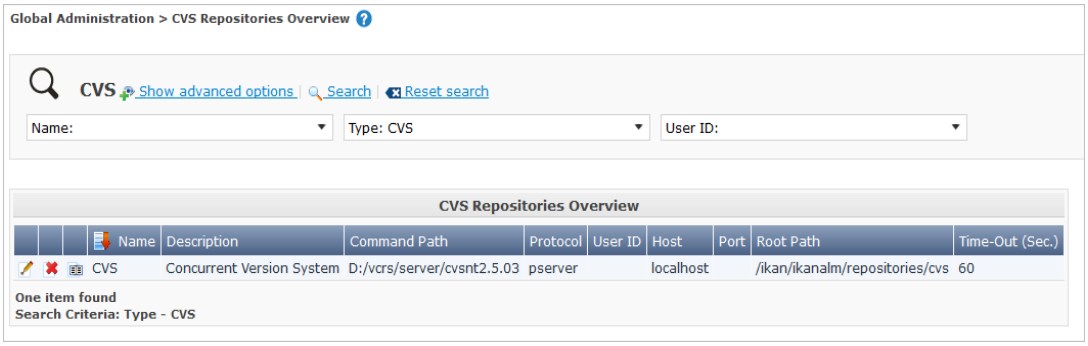
6. Once the test is successful, click *Create*.
The newly created CVS Repository is added to the *CVS Repositories Overview* at the bottom of the screen.

RELATED TOPICS
- Version Control Repositories (page 330)
- Editing Project Settings (page 149)
- Creating a Project (page 461)

The CVS Repositories Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
The overview of all defined Version Control Repositories is displayed.

2. Specify *CVS* in the *Type* field on the *Search Version Control Repositories* panel.
Use the other search criteria to only display the CVS Repositories you are looking for.



If required, use the other search criteria to refine the items displayed on the overview.

The following options are available:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- the *Search* link to refresh the list based on the current search criteria,
- the *Reset search* link to clear the search fields.

3. Verify the information on the *CVS Repositories Overview* panel.
   For a detailed description of the fields, refer to Creating a CVS Repository (page 346).

4. Depending on your access rights, the following links may be available on the *CVS Repositories Overview* panel:

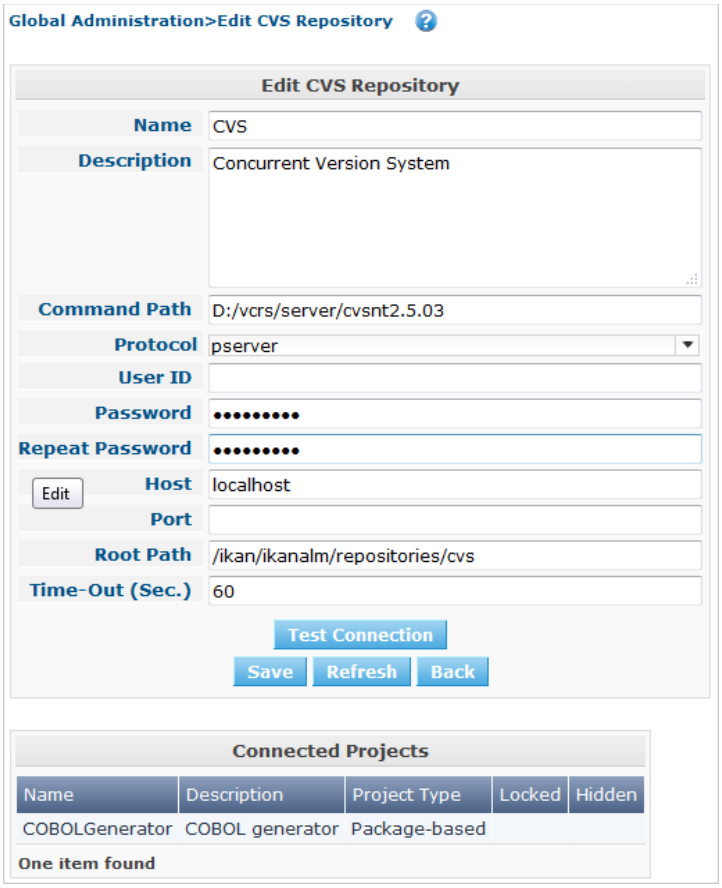
	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a CVS Repository definition. See Editing a CVS Repository on page 350.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a CVS Repository definition. See Deleting a CVS Repository on page 351.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a CVS Repository. See Viewing the CVS Repository History on page 353.

Editing a CVS Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the CVS Repositories you are looking for.

2. Click the *Edit* link to change the selected CVS Repository.

The following screen is displayed:



3. Edit the fields as required.

   For a description of the fields, refer to Creating a CVS Repository (page 346).

Note: Click *Test Connection* to verify if IKAN ALM can establish a connection to the CVS Repository. The *Connected Projects* panel displays the Projects the Repository is linked to.

4. Click *Save* to save your changes.

   You can also click:

   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

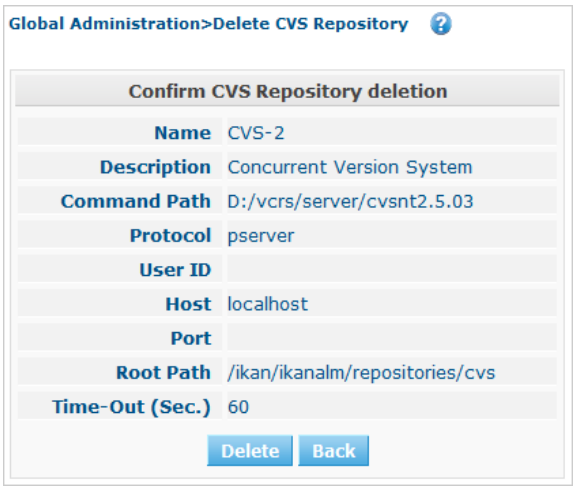
Deleting a CVS Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.

   The overview of all defined Version Control Repositories is displayed.

   Use the search criteria on the *Search Version Control Repository* panel to only display the CVS Repositories you are looking for.

2. Click the *Delete* link to delete the selected CVS Repository.
   If the CVS Repository is not connected to any Project, the following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the CVS Repository is connected to one or more Projects, the following screen is displayed:

   Before deleting the VCR, you must connect the reported Projects to another VCR.

Viewing the CVS Repository History

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the CVS Repositories you are looking for.

2. Click the *History* link to display the *CVS Repository History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *CVS Repositories Overview* screen.

29.6. Visual SourceSafe (VSS) Repositories

Refer to the following sections for detailed information:

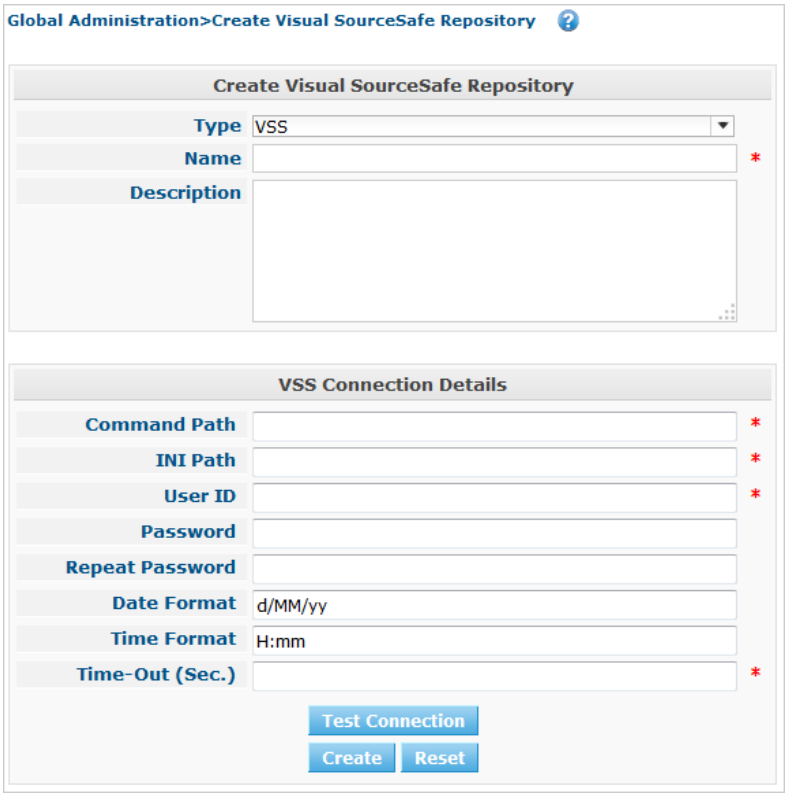
- Creating a VSS Version Control Repository (page 353)
- The VSS Repositories Overview Screen (page 356)
- Editing a VSS Repository (page 357)
- Deleting a VSS Repository (page 358)
- Viewing the VSS Repository History (page 359)

Creating a VSS Version Control Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.
Before you can create the VSS Version Control Repository in IKAN ALM, you must install a VSS Client on the IKAN ALM Server.

1. In the Global Administration context, select *Version Control Repositories > Create*.

2. Select *VSS* from the drop-down list in the *Type* field.
The following screen is displayed:



3. Fill out the fields in the *Create Visual SourceSafe Repository* screen.
Fields marked with a red asterisk are mandatory:

Field	Meaning
Type	Select the type of Version Control Repository you want to define. This field is mandatory. After you have selected the VCR type, the appropriate *Connection Details* panel will be displayed underneath.
Name	Enter the name of the new VSS Repository definition in this field. This field is mandatory.
Description	Enter a description for the new VSS Repository in this field. This field is optional.

4. Fill out the fields in the *VSS Connection details* panel.
Fields marked with a red asterisk are mandatory:

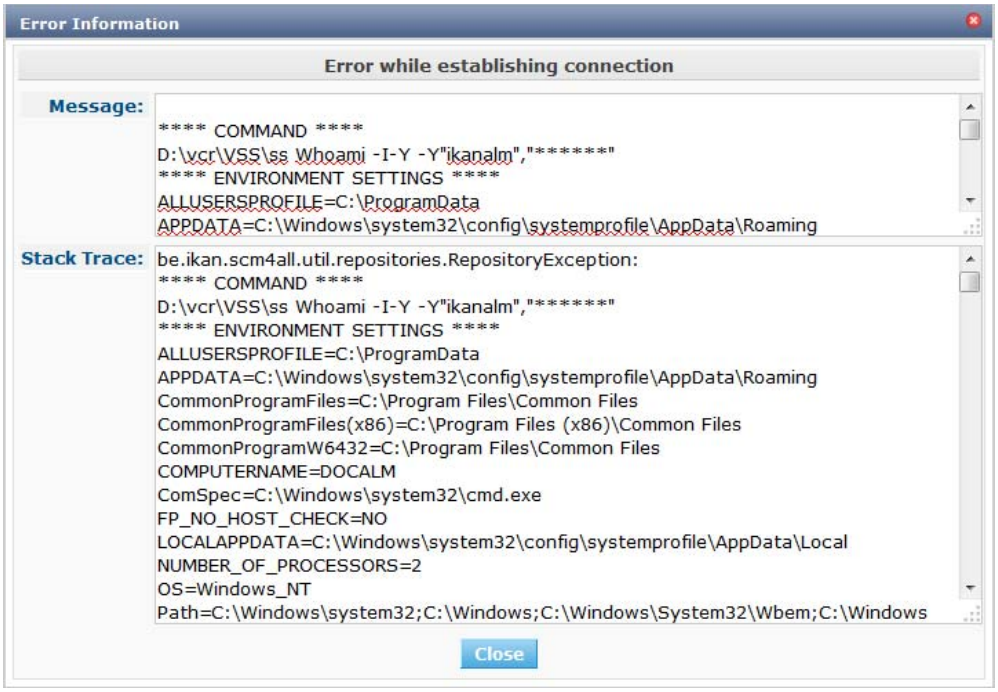
Field	Meaning
Command Path	Enter the required Command Path of the VSS Client (absolute path to the *ss.exe* file).
INI Path	Enter the VSS ROOT used to log in to VSS, or the *ssdir* environment variable (absolute path to the *srcsafe.ini* file).
User ID	Enter the User ID that IKAN ALM will use to access the VSS Repository.

Field	Meaning
Password	Enter the Password that IKAN ALM will use to access the VSS Repository. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password that IKAN ALM will use to access the VSS Repository.
Date Format	Enter the Date Format that is in use on the system where the VSS Client is running. Examples: • *MM/dd/yyyy* • *dd/MM/yy*
Time Format	Enter the Time Format that is used on the system where the VSS Client is running. Example: *HH:mm*
Time-Out (Sec.)	Enter the Time-Out value in seconds. If IKAN ALM is not able to establish a connection with the VSS Repository within the defined period, it will consider the Repository to be inaccessible. The definition of a Time-Out value is mandatory.

5. Click *Test Connection* to verify if IKAN ALM can establish a connection to the VSS Repository. If the test is successful, the following message is displayed:

   Info: Could successfully establish a connection with the Repository.

   If the test is not successful, the following screen is displayed:



   Correct the errors reported in the Stack Trace field and perform the test again.

6. Once the test is successful, click *Create*.

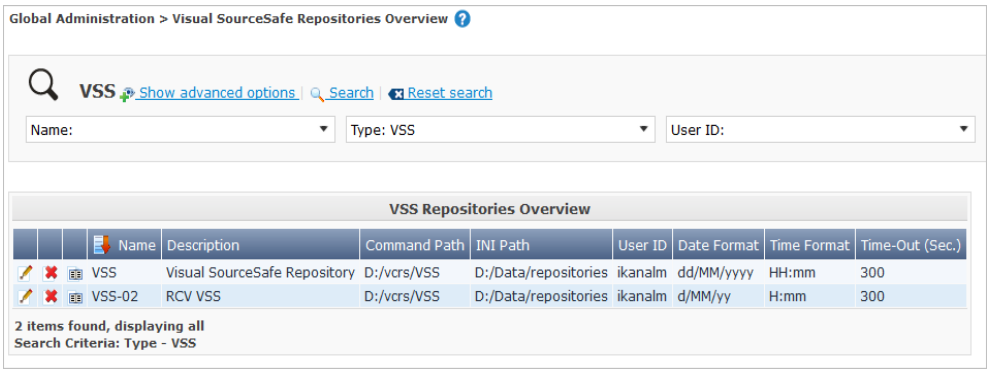
   The newly created VSS Repository is added to the *VSS Repositories Overview* at the bottom of the screen.

RELATED TOPICS
- Version Control Repositories (page 330)
- Editing Project Settings (page 149)
- Creating a Project (page 461)

The VSS Repositories Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.

2. Specify *VSS* in the *Type* field on the *Search Version Control Repositories* panel.
   Use the other search criteria to only display the VSS Repositories you are looking for.



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *VSS Repositories Overview* panel.
   For a detailed description of the fields, refer to Creating a VSS Version Control Repository (page 353).

4. Depending on your access rights, the following links may be available on the *VSS Repositories Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a VSS Repository definition. See Editing a VSS Repository on page 357.

	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a VSS Repository definition. See Deleting a VSS Repository on page 358.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a VSS Repository. See Viewing the VSS Repository History on page 359.

Editing a VSS Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the VSS Repositories you are looking for.

2. Click the *Edit* link to change the selected VSS Repository.
   The following screen is displayed:

3. Edit the fields as required.
   For a description of the fields, refer to Creating a VSS Version Control Repository (page 353).

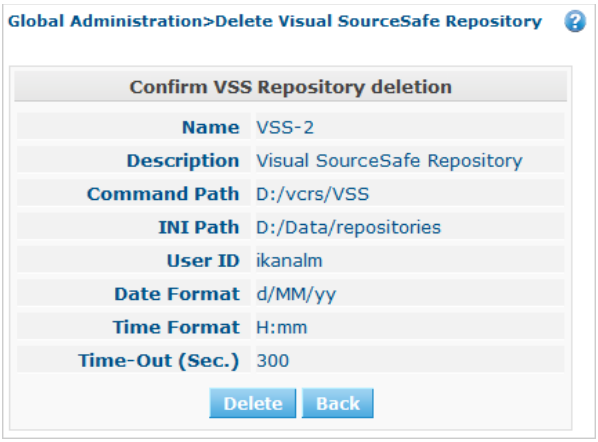
Note: Click *Test Connection* to verify if IKAN ALM can establish a connection to the VSS Repository. The *Connected Projects* panel displays the Projects the Repository is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

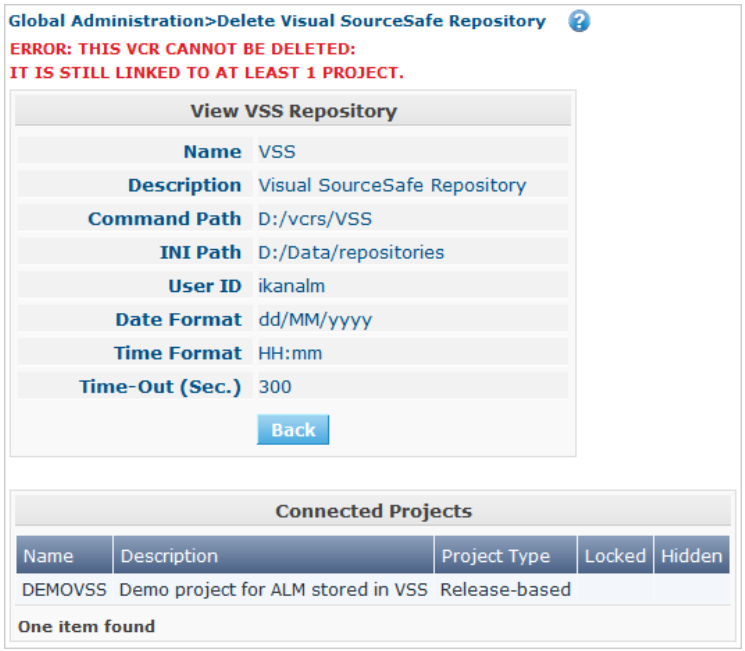
Deleting a VSS Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the VSS Repositories you are looking for.

2. Click the *Delete* link to delete the selected VSS Repository.
   If the VSS Repository is not connected to any Project, the following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the VSS Repository is connected to one or more Projects, the following screen is displayed:



Before deleting the VCR, you must connect the reported Projects to another VCR.

Viewing the VSS Repository History

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the VSS Repositories you are looking for.

2. Click the *History* link to display the *VSS Repository History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *VSS Repositories Overview* screen.

29.7. ClearCase Repositories

Refer to the following sections for detailed information:

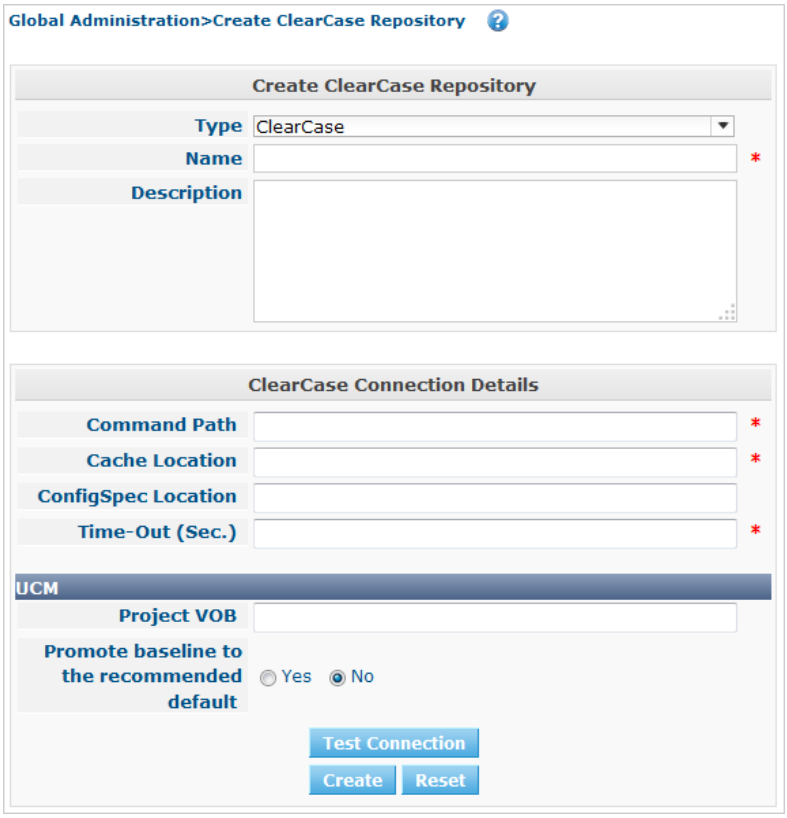
- Creating a ClearCase Repository (page 360)
- The ClearCase Repositories Overview Screen (page 363)
- Editing a ClearCase Repository (page 364)
- Deleting a ClearCase Repository (page 365)
- Viewing the ClearCase Repository History (page 366)

Creating a ClearCase Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM. Before you can create the ClearCase Version Control Repository in IKAN ALM, you must install a ClearCase Client on the IKAN ALM Server.

1. In the Global Administration context, select *Version Control Repositories > Create*.

2. Select from the drop-down list in the *Type* field.
   The following screen is displayed:



3. Fill out the fields in the *Create ClearCase Repository* screen.

Fields marked with a red asterisk are mandatory:

Field	Meaning
Type	Select the type of Version Control Repository you want to define. This field is mandatory. After you have selected the VCR type, the appropriate *Connection Details* panel will be displayed underneath.
Name	Enter the name of the new ClearCase Repository definition in this field. This field is mandatory.
Description	Enter a description for the new ClearCase Repository in this field. This field is optional.

4. Fill out the fields in the *ClearCase Connection details* panel.

Fields marked with a red asterisk are mandatory:

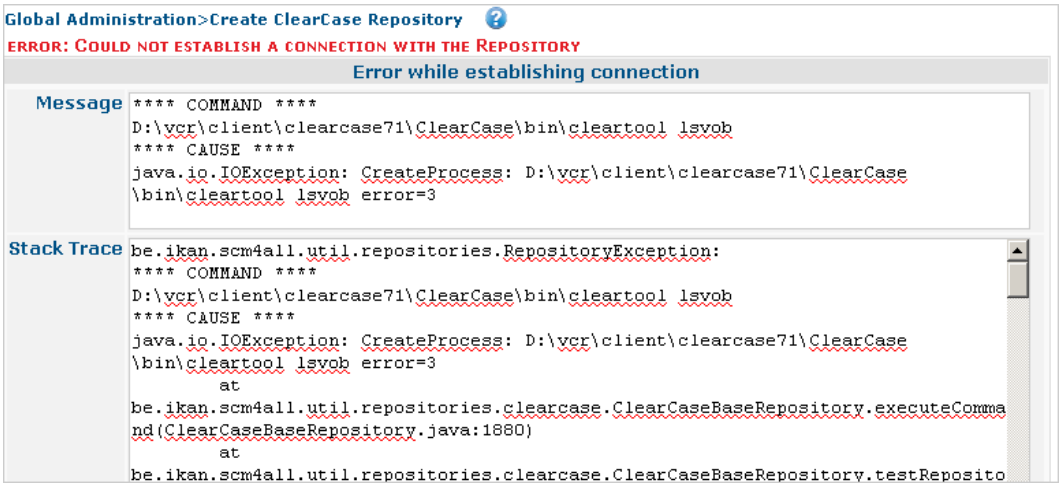
Field	Meaning
Command Path	Enter the required Command Path for the new ClearCase Repository. This is the directory containing the ClearCase client command (cleartool) to connect with the ClearCase Repository.
Cache Location	Enter the path to the Cache location for this ClearCase Repository. This directory is used by the Scheduler to check whether there are modifications in a ClearCase VOB (Versioned Objects Base). A ClearCase View, containing the VOB, will be created in this directory. On Windows, this path MUST be an UNC style path (\\server\share). If not, certain IKAN ALM operations will not function correctly.
ConfigSpec Location	In this field, enter the absolute path to your customized ClearCase Configuration specification files. IKAN ALM will search this location for ConfigSpec files with names in the following format: • ConfigSpec_ProjectName_BranchID.vm • ConfigSpec_ProjectName.vm • ConfigSpec.vm IKAN ALM looks first for the most specific match (including the Branch ID in the name). If no match is found, IKAN ALM will look for the second best match (including the ProjectName). If again no match is found, IKAN ALM will look for the general ConfigSpec.vm file. If there is no ConfigSpec.vm available in the indicated location, the default ConfigSpec.vm available in the IKAN ALM classpath will be used.
Time-Out (Sec.)	Enter the Time-Out value in seconds. If IKAN ALM is not able to establish a connection with the ClearCase Repository within the defined period, it will consider the Repository to be inaccessible. The definition of a Time-Out value is mandatory.

Field	Meaning
UCM - Project VOB	This field only applies, if the repository you want to connect to is a ClearCase UCM Repository. Enter the name of the Project VOB in which the UCM has been created.
UCM - Promote baseline to default recommended	This field only applies, if the repository you want to connect to is a ClearCase UCM Repository. **Select YES,** if you want to promote each tagged Build (*Baseline*) created by IKAN ALM to the *default recommended Baseline*. Note: If you set the stream policy *POLICY_CHSTREAM_UNRESTRICTED*, the permission checking is skipped. You can only use this option, if the account running the webserver, under which IKAN ALM is running, has the status of Project VOB Owner, Stream Owner, Root (Unix), Member of the ClearCase Administrators Group (ClearCase on Windows only), Local Administrator of the ClearCase LT Server Host (ClearCase LT on Windows only). For more information, see the mkstream reference page in the ClearCase documentation. **Select NO**, if you do not want to promote each tagged Build (*Baseline*) created by IKAN ALM to the *default recommended Baseline*. In this case, this will be a manual process performed outside IKAN ALM.

5. Click *Test Connection* to verify if IKAN ALM can establish a connection to the ClearCase Repository. If the test is successful, the following message is displayed:

   Info: Could successfully establish a connection with the Repository.

   If the test is not successful, the following screen is displayed:



   Correct the errors reported in the Stack Trace field and perform the test again.

6. Once the test is successful, click *Create*.

   The newly created ClearCase Repository is added to the *ClearCase Repositories Overview* at the bottom of the screen.

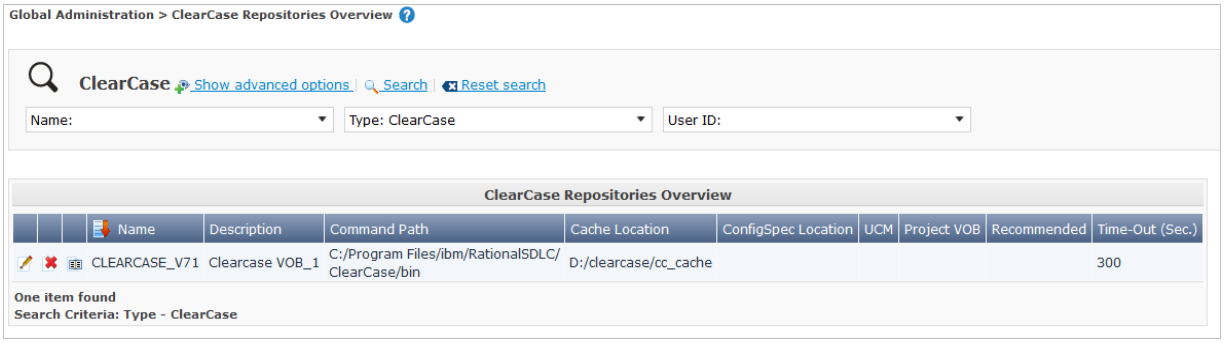
RELATED TOPICS

- Version Control Repositories (page 330)
- Editing Project Settings (page 149)
- Creating a Project (page 461)

The ClearCase Repositories Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.

2. Specify *ClearCase* in the *Type* field on the *Search Version Control Repositories* panel.
   Use the other search criteria to only display the ClearCase Repositories you are looking for.



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *ClearCase Repositories Overview* panel.
   For a detailed description of the fields, refer to Creating a ClearCase Repository (page 360).

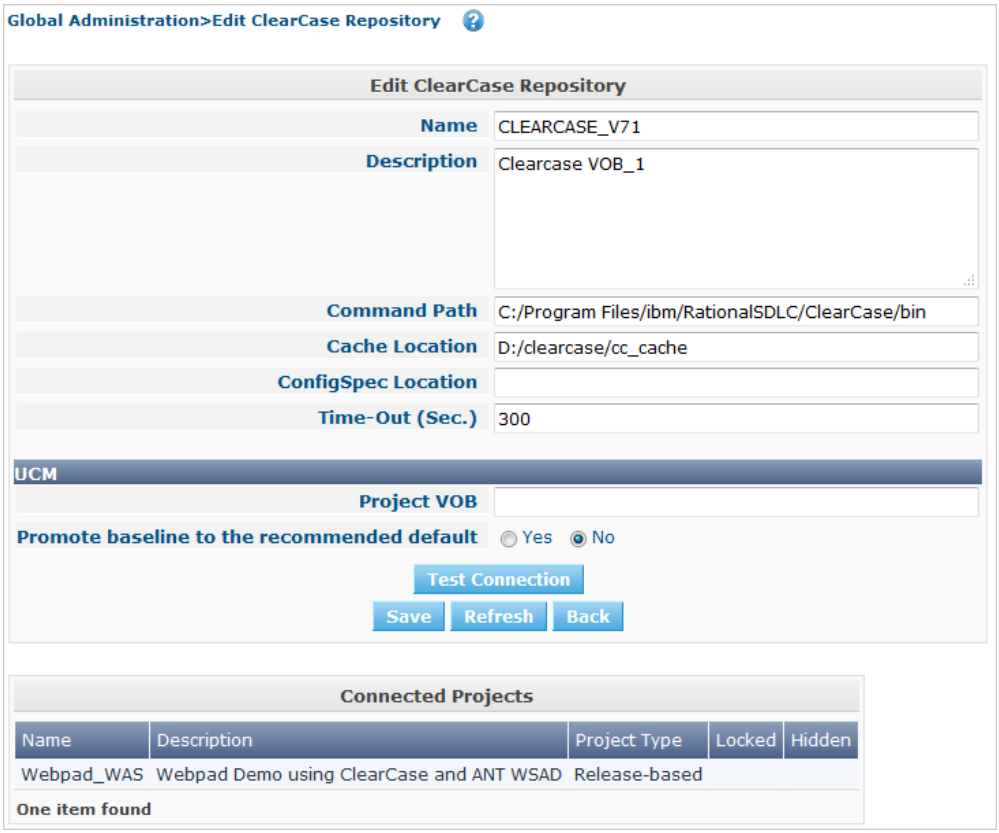
4. Depending on your access rights, the following links may be available on the *ClearCase Repositories Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a ClearCase Repository definition. See Editing a ClearCase Repository on page 364.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a ClearCase Repository definition. See Deleting a ClearCase Repository on page 365.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a ClearCase Repository. See Viewing the ClearCase Repository History on page 366.

Editing a ClearCase Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the ClearCase Repositories you are looking for.

2. Click the *Edit* link to change the selected ClearCase Repository.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a ClearCase Repository (page 360).

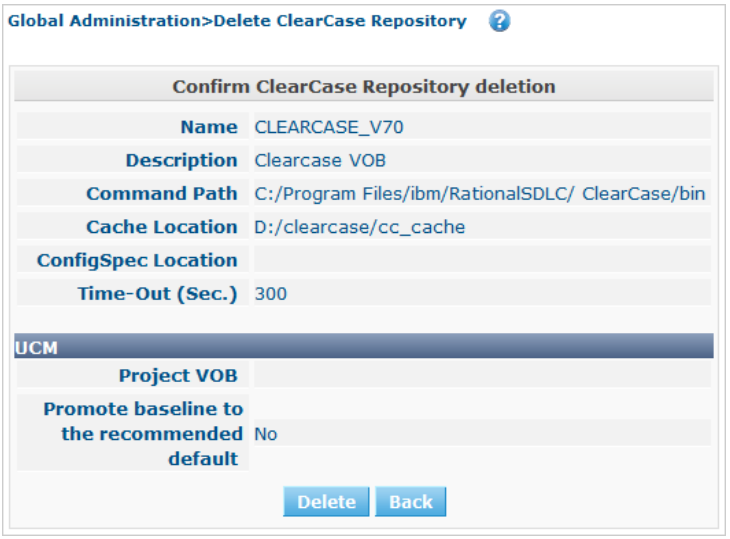
Note: Click *Test Connection* to verify if IKAN ALM can establish a connection to the ClearCase Repository.
The *Connected Projects* panel displays the Projects the Repository is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

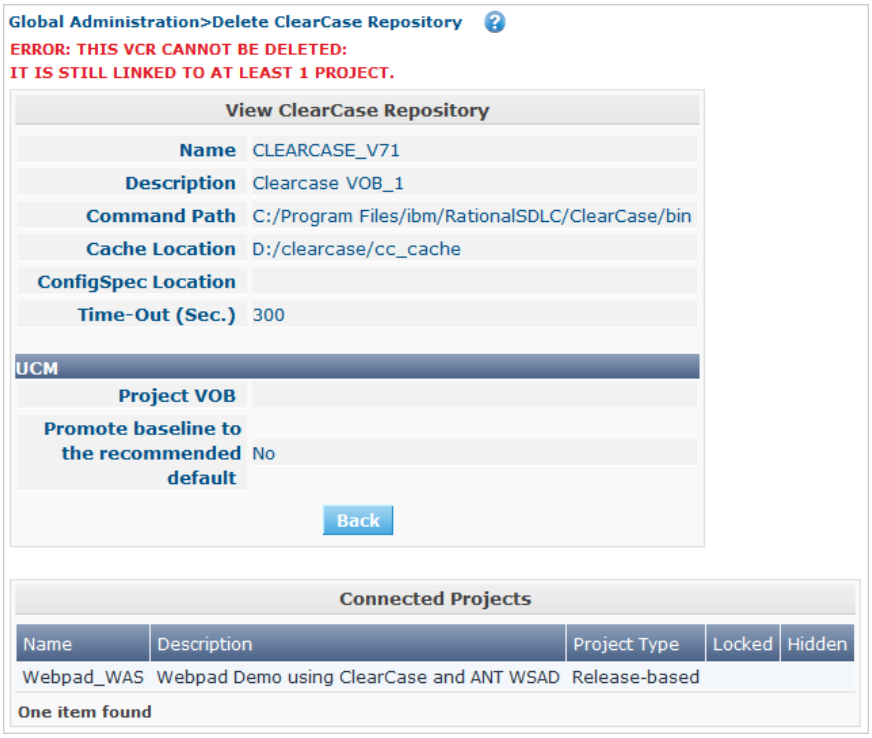
Deleting a ClearCase Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the ClearCase Repositories you are looking for.

2. Click the *Delete* link to delete the selected ClearCase Repository.
   If no Projects are connected to the Repository, the following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If one or more Projects are connected to the Repository, the following screen is displayed:



   Before deleting the VCR, you must connect the reported Projects to another VCR.

Viewing the ClearCase Repository History

1. In the Global Administration context, select *Version Control Repositories > Overview*.

   The overview of all defined Version Control Repositories is displayed.

   Use the search criteria on the *Search Version Control Repository* panel to only display the ClearCase Repositories you are looking for.

2. Click the *History* link to display the *ClearCase Repository History View*.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *ClearCase Repositories Overview* screen.

29.8. PVCS Repositories

Refer to the following sections for detailed information:

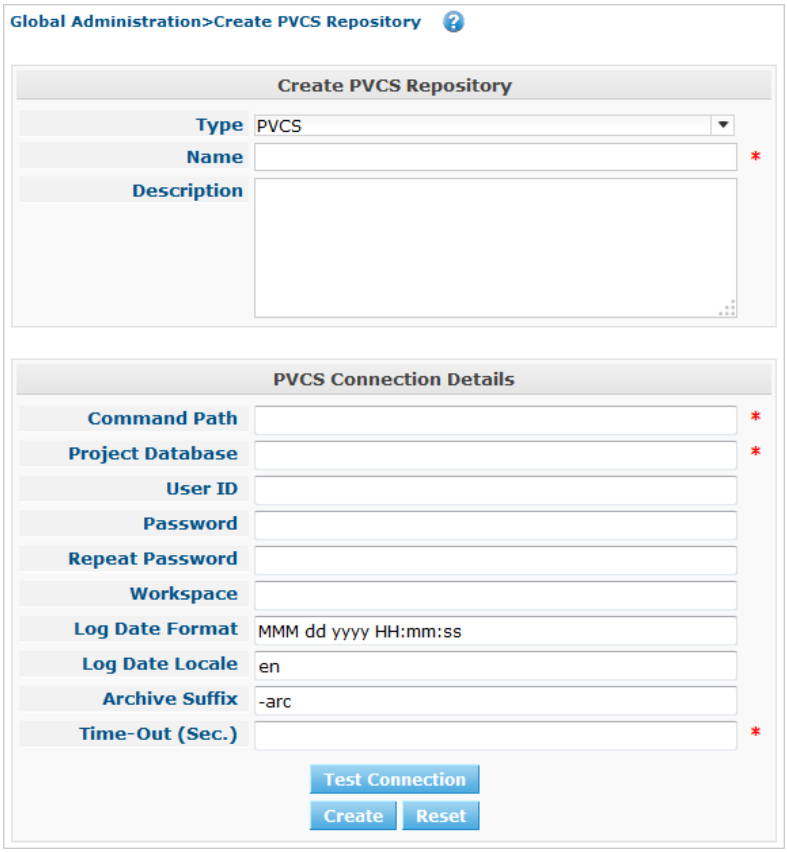
- Creating a PVCS Repository (page 367)
- The PVCS Repositories Overview Screen (page 369)
- Editing a PVCS Repository (page 371)
- Deleting a PVCS Repository (page 372)
- Viewing the PVCS Repository History (page 373)

Creating a PVCS Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM. Before you can create the PVCS Version Control Repository in IKAN ALM, you must install a PVCS Client on the IKAN ALM Server.

1. In the Global Administration context, select *Version Control Repositories > Create*.

2. Select from the drop-down list in the *Type* field.
   The following screen is displayed:



3. Fill out the fields in the *Create PVCS Repository* screen.

Fields marked with a red asterisk are mandatory:

Field	Meaning
Type	Select the type of Version Control Repository you want to define. This field is mandatory. After you have selected the VCR type, the appropriate *Connection Details* panel will be displayed underneath.
Name	Enter the name of the new PVCS Repository definition in this field. This field is mandatory.
Description	Enter a description for the new PVCS Repository in this field. This field is optional.

4. Fill out the fields in the *PVCS Connection details* panel.

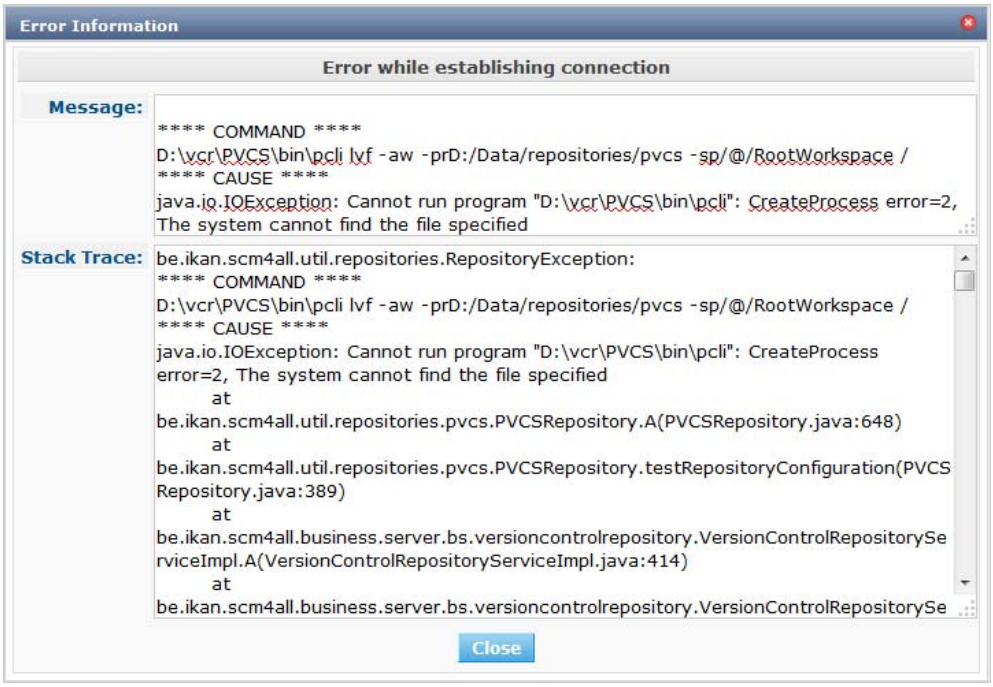
Fields marked with a red asterisk are mandatory:

Field	Meaning
Command Path	Enter the path to the PCLI executable. This field is mandatory.
Project Database	Enter the location of the Project Database. This field is mandatory.
User ID	Enter the User ID that IKAN ALM will use to access the PVCS Repository. This field is optional.
Password	Enter the Password that IKAN ALM will use to access the PVCS Repository. This field is optional. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password that IKAN ALM will use to access the PVCS Repository.
Workspace	Enter the Workspace path to use. This field is optional. If none is specified, IKAN ALM will use the Root Workspace ("/RootWorkspace").
Log Date Format	Enter the Date format of modification dates output by the "pcli vlog" command, e.g., for the default format (MMM dd yyyy HH:mm:ss): `Oct 11 2014 11:53:04`
Log Date Locale	Enter the locale of the date format of modification dates output by the "pcli vlog" command, e.g., for the default format (MMM dd yyyy HH:mm:ss) : Locale = "en" : `Oct 11 2014 11:53:04` Locale = "nl" : `Okt 11 2014 11:53:04` This field is mandatory. The default value is “en”.
Archive Suffix	Enter the suffix for PVCS archive files. This field is mandatory. The default value is "-arc"
Time-Out (Sec.)	Enter the Time-Out value in seconds. If IKAN ALM is not able to establish a connection with the PVCS Repository within the defined period, it will consider the Repository to be inaccessible. The definition of a Time-Out value is mandatory.

5. Click *Test Connection* to verify if IKAN ALM can establish a connection to the PVCS Repository. If the test is successful, the following message is displayed:

Info: Could successfully establish a connection with the Repository.

If the test is not successful, the following screen is displayed:



Correct the errors reported in the Stack Trace field and perform the test again.

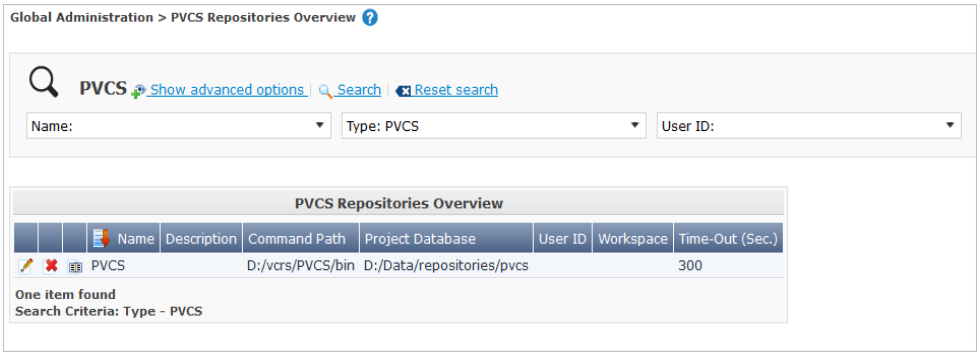
6. Once the test is successful, click *Create*.
   The newly created PVCS Repository is added to the *PVCS Repositories Overview* at the bottom of the screen.

RELATED TOPICS
- Version Control Repositories (page 330)
- Editing Project Settings (page 149)
- Creating a Project (page 461)

The PVCS Repositories Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.

2. Specify *PVCS* in the *Type* field on the *Search Version Control Repositories* panel.
   Use the other search criteria to only display the PVCS Repositories you are looking for.



If required, use the other search criteria to refine the items displayed on the overview.

The following options are available:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- the *Search* link to refresh the list based on the current search criteria,
- the *Reset search* link to clear the search fields.

3. Verify the information on the *PVCS Repositories Overview* panel.
   For a detailed description of the fields, refer to Creating a PVCS Repository (page 367).

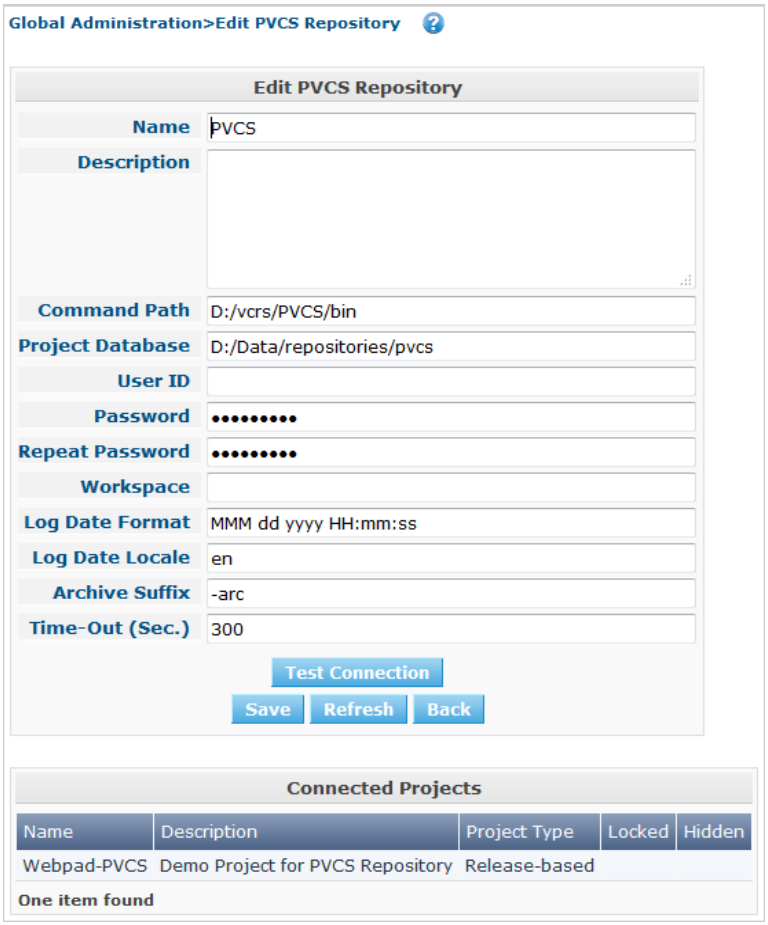
4. Depending on your access rights, the following links may be available on the *PVCS Repositories Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a PVCS Repository definition. See Editing a PVCS Repository on page 371.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a PVCS Repository definition. See Deleting a PVCS Repository on page 372.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a PVCS Repository. See Viewing the PVCS Repository History on page 373.

Editing a PVCS Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the PVCS Repositories you are looking for.

2. Click the *Edit* link to change the selected PVCS Repository.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a PVCS Repository (page 367).

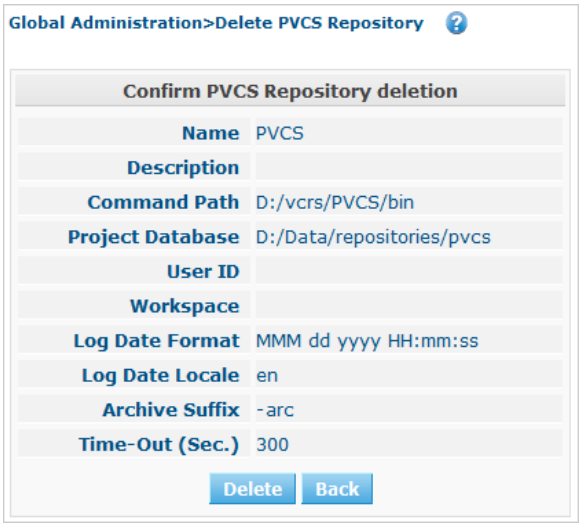
Note: Click *Test Connection* to verify if IKAN ALM can establish a connection to the PVCS Repository. The *Connected Projects* panel displays the Projects the Repository is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

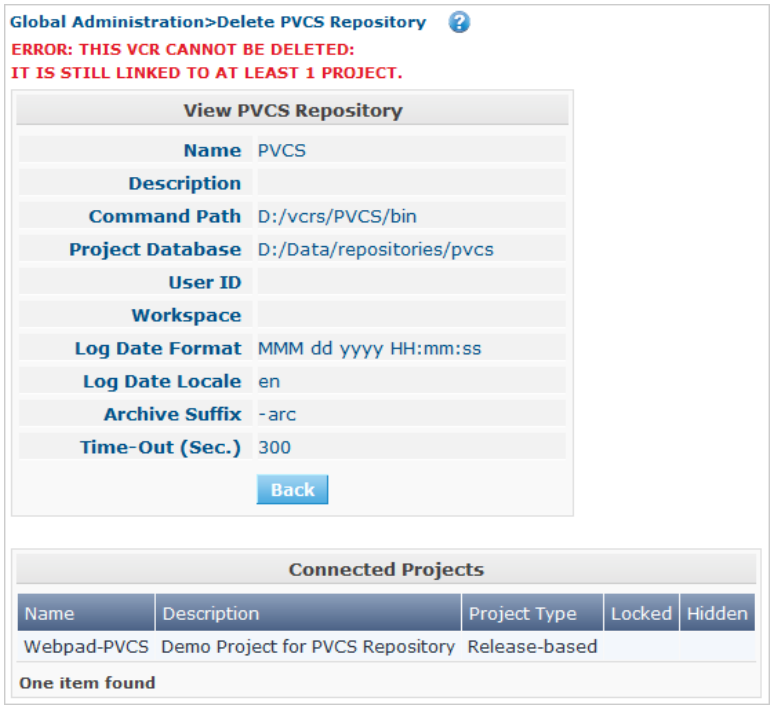
Deleting a PVCS Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the PVCS Repositories you are looking for.

2. Click the *Delete* link to delete the selected PVCS Repository.
   If the PVCS Repository is not connected to any Project, the following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the PVCS Repository is connected to one or more Projects, the following screen is displayed:



   Before deleting the VCR, you must connect the reported Projects to a different VCR.

Viewing the PVCS Repository History

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the PVCS Repositories you are looking for.

2. Click the *History* link to display the *PVCS Repository History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *PVCS Repositories Overview* screen.

29.9. TFVC Version Control Repositories

Refer to the following sections for detailed information:

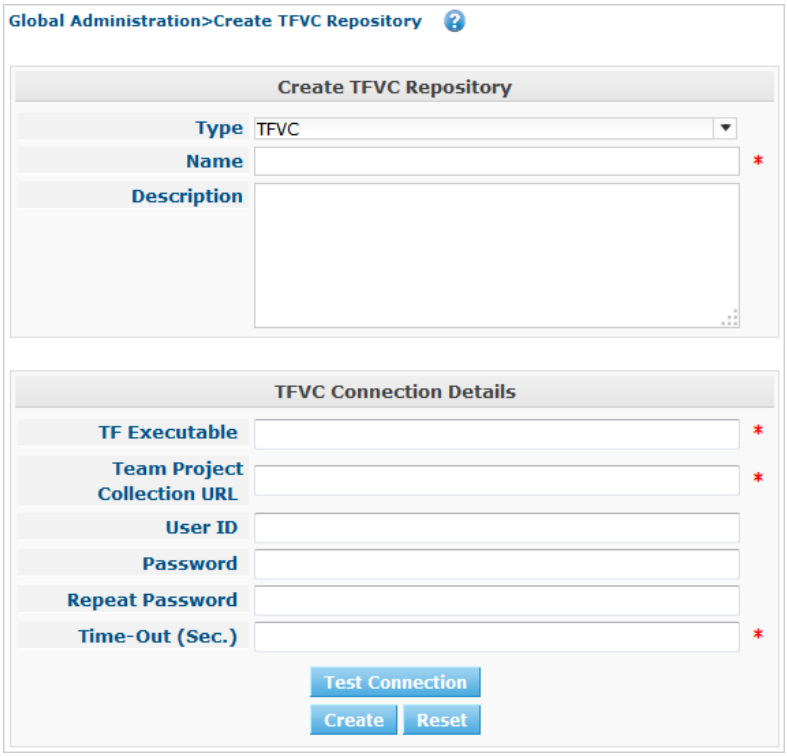
- Creating a TFVC Repository (page 373)
- The TFVC Repositories Overview Screen (page 376)
- Editing a TFVC Repository (page 378)
- Deleting a TFVC Repository (page 379)
- Viewing the TFVC Repository History (page 380)

Creating a TFVC Repository

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM. Before you can create the TFVC Version Control Repository in IKAN ALM, you must install a TFVC Client (Team Explorer for MS Visual Studio or Team Explorer Everywhere for Team Foundation Server) on the IKAN ALM Server.

1. In the Global Administration context, select *Version Control Repositories* > *Create*.

2. Select *TFVC* from the drop-down list in the *Type* field.
   The following screen is displayed:



3. Fill out the fields in the *Create TFVC Repository* screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Type	Select the type of Version Control Repository you want to define. This field is mandatory. After you have selected the VCR type, the appropriate *Connection Details* panel will be displayed underneath.
Name	Enter the name of the new TFVC Repository definition in this field. This field is mandatory.
Description	Enter a description for the new TFVC Repository in this field. This field is optional.

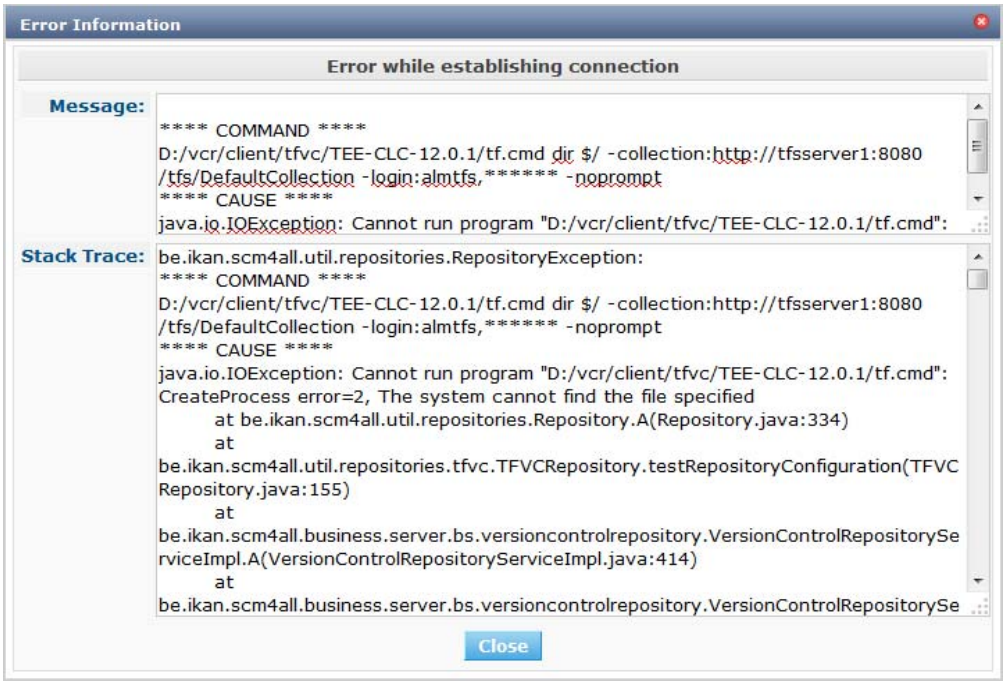
4. Fill out the fields in the *TFVC Connection details* panel. Fields marked with a red asterisk are mandatory:

Field	Meaning
Executable	Enter the location of the Team Foundation executable. This may be the *tf.cmd* from the TEE client, or the *tf.exe* from the TFVC command line client. This field is mandatory.
Team Project Collection URL	Provide the URL to your Team Project Collection, in the following format: `http[s]://ServerName[:port]/path/to/collection`. Some sample URLs for connecting to an on premises installation and a Visual Studio Online: • http://ikan_tfs:8080/tfs/DefaultCollection • https://ikan.visualstudio.com/DefaultCollection This field is mandatory.
User ID	Enter the User ID that IKAN ALM will use to access the TFVC Repository. Repository. This field is optional.
Password	Enter the Password that IKAN ALM will use to access the TFVC Repository. This field is optional. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password that IKAN ALM will use to access the TFVC Repository.
Time-Out (Sec.)	Enter the Time-Out value in seconds. If IKAN ALM is not able to establish a connection with the TFVC Repository within the defined period, it will consider the Repository to be inaccessible. The definition of a Time-Out value is mandatory.

5. Click *Test Connection* to verify if IKAN ALM can establish a connection to the TFVC Repository. If the test is successful, the following message is displayed:

   Info: Could successfully establish a connection with the Repository.

If the test is not successful, the following screen is displayed:



Correct the errors reported in the Stack Trace field and perform the test again.

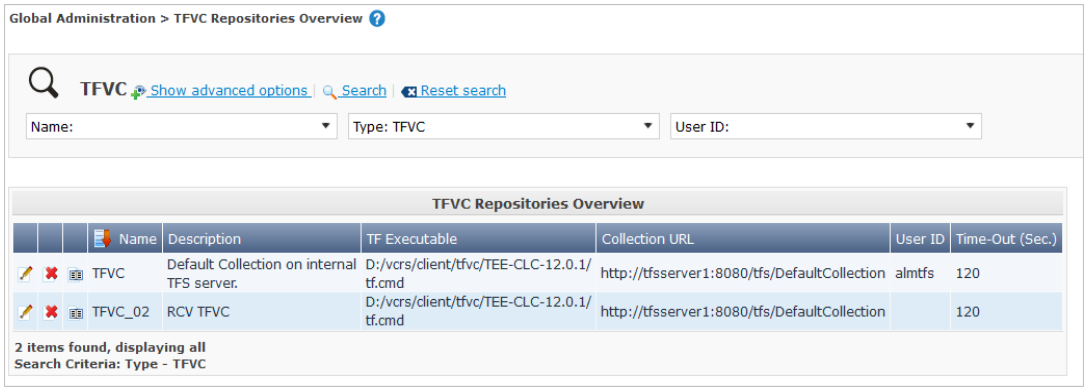
6. Once the test is successful, click *Create*.
   The newly created TFVC Repository is added to the *TFVC Repositories Overview* at the bottom of the screen.

RELATED TOPICS
- Version Control Repositories (page 330)
- Editing Project Settings (page 149)
- Creating a Project (page 461)

The TFVC Repositories Overview Screen

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.

2. Specify *TFVC* in the *Type* field on the *Search Version Control Repositories* panel.
   Use the other search criteria to only display the TFVC Repositories you are looking for.



If required, use the other search criteria to refine the items displayed on the overview.
The following options are available:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- the *Search* link to refresh the list based on the current search criteria,
- the *Reset search* link to clear the search fields.

3. Verify the information on the *TFVC Repositories Overview* panel.
   For a detailed description of the fields, refer to Creating a TFVC Repository (page 373).

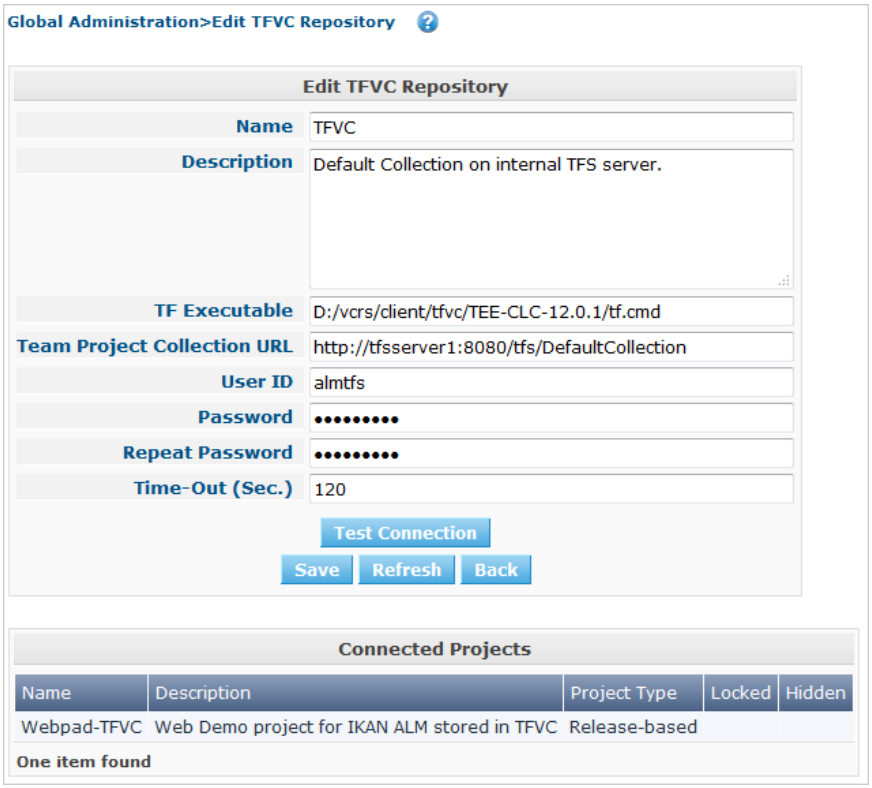
4. Depending on your access rights, the following links may be available on the *TFVC Repositories Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a TFVC Repository definition. See Editing a Subversion Repository on page 337.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a TFVC Repository definition. See Deleting a Subversion Repository on page 338.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a TFVC Repository. See Viewing the Subversion Repository History on page 339.

Editing a TFVC Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the TFVC Repositories you are looking for.

2. Click the *Edit* link to change the selected TFVC Repository.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a TFVC Repository (page 373).

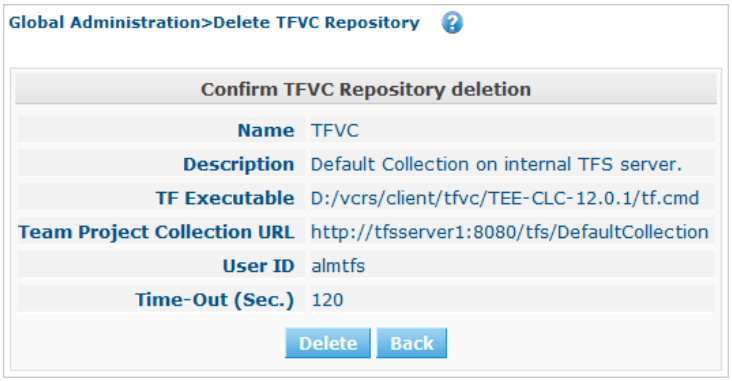
Note: Click *Test Connection* to verify if IKAN ALM can establish a connection to the TFVC Repository. The *Connected Projects* panel displays the Projects the Repository is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

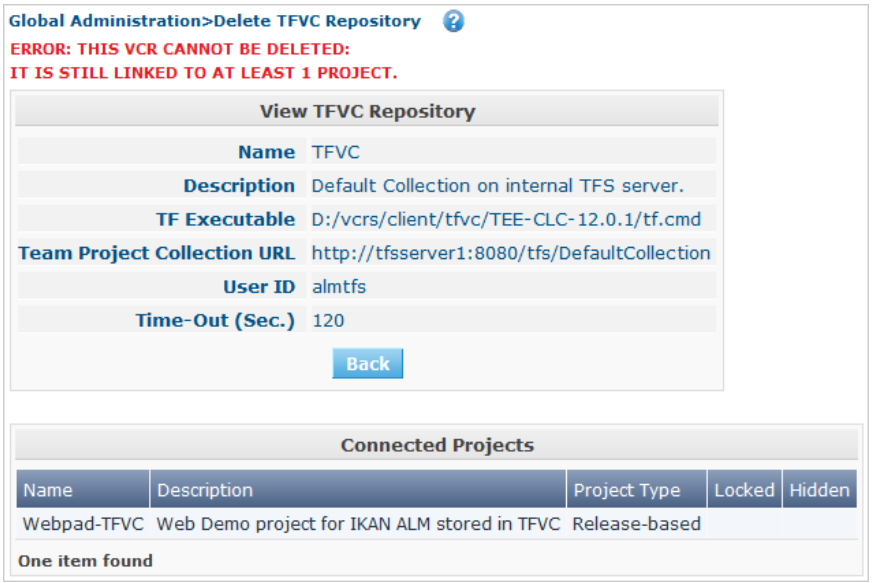
Deleting a TFVC Repository

1. In the Global Administration context, select *Version Control Repositories > Overview*.
   The overview of all defined Version Control Repositories is displayed.
   Use the search criteria on the *Search Version Control Repository* panel to only display the TFVC Repositories you are looking for.

2. Click the *Delete* link to delete the selected TFVC Repository.
   If the TFVC Repository is not connected to any Project, the following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the TFVC Repository is connected to one or more Projects, the following screen is displayed:



   Before deleting the VCR, you must connect the reported Projects to another VCR.

Viewing the TFVC Repository History

1. In the Global Administration context, select *Version Control Repositories > Overview*.

   The overview of all defined Version Control Repositories is displayed.

   Use the search criteria on the *Search Version Control Repository* panel to only display the TFVC Repositories you are looking for.

2. Click the *History* link to display the *TFVC Repository History View*.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *TFVC Repositories Overview* screen.

CHAPTER 30
Issue Tracking

The IKAN ALM Issue Tracking functionality provides the possibility to link issues from an external issue tracking system to an IKAN ALM Level Request.

Issues can get linked to a Level Request in the following ways:

- via an automatic search for handled issues in the comments provided by the developers when committing or checking in sources into the VCR. This automatic search is done in a separate Phase during the handling of a Build Level Request, and is based on a regular expression describing the issue tracking number format. Currently this is only supported for a Subversion, CVS or Microsoft Team Foundation Version Control Repository.
- via the enumeration of solved issues when delivering to Test/Production levels enabling easy generation of Release Notes
- via a manual addition of an issue to a successful Level Request, using the GUI widgets on the *Level Request Detail* screen.

Note: You can only click through from a linked issue in IKAN ALM to the issue in the third-party Issue Tracking tool if the tool offers a web interface.

On top of this basic Issue Tracking functionality, IKAN ALM offers an extra tight integration for Atlassian JIRA, HP Quality Center, Microsoft Team Foundation and CollabNet TeamForge with following additional functionality:

- possibility to test the connection with the Issue Tracking system at the moment you create or edit an Issue Tracking System definition
- automatic synchronization of handled issues with the Issue Tracking system: IKAN ALM retrieves additional information, such as a short description, the owner, the priority, ... from the corresponding JIRA Issue, HP Quality Center Defect, Team Foundation Work Item or TeamForge Artifact. This information is synchronized each time the build result evolves in the IKAN ALM Lifecycle.
- manual synchronization of issues linked to a Level Request on the *Level Request Detail* screen
- option to automatically add a comment to the issue/defect/artifact in JIRA/HP Quality Center/ TeamForge/Team Foundation after a successful Level Request. This comment will contain a link to the IKAN ALM Level Request.

Refer to the following sections for detailed information.

Issue Tracking System

- Creating an Issue Tracking System (page 382)
- The Issue Tracking Systems Overview Screen (page 385)
  - Editing an Issue Tracking System Definition (page 387)
  - Deleting an Issue Tracking System Definition (page 389)
  - Viewing the Issue Tracking System History (page 390)

Issue Tracking System Properties

- Creating an Issue Tracking System Property (page 392)
- The Issue Tracking System Properties Overview Panel (page 390)
  - Editing an Issue Tracking System Property (page 393)
  - Deleting an Issue Tracking System Property (page 393)

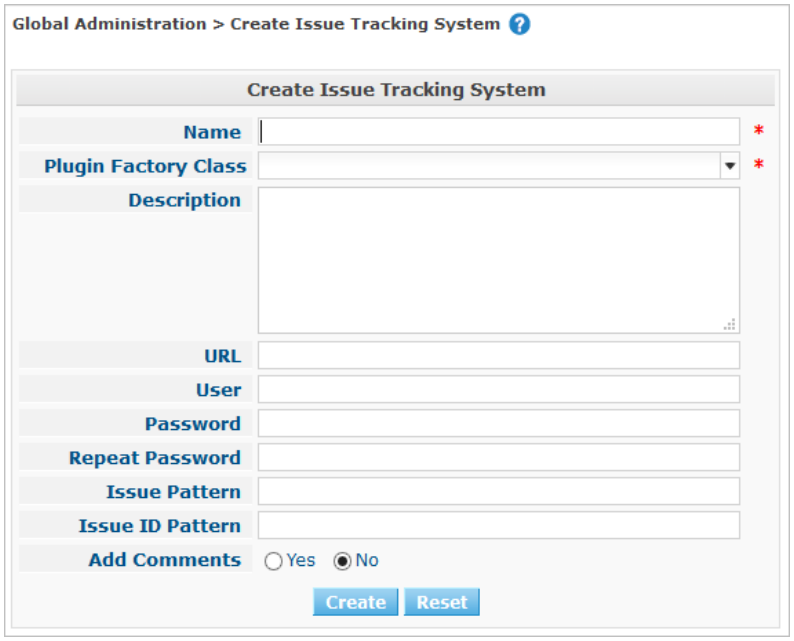
Once the Issue Tracking System has been defined in the *Global Administration* section, you must activate it on the Project. See Editing Project Settings on page 149.

If you want Issues to be automatically linked to Level Requests for certain Levels, the Issue Tracking Phase must be linked to those Levels. This will be done automatically when creating Levels for Projects which have an Issue Tracking System activated. If you activate the Issue Tracking System for a Project with existing Levels, you have to link the Issue Tracking Phase manually to each of those Levels. For more information, refer to the sections Issue Tracking Phase (page 524)and Level Phases (page 195).

30.1. Creating an Issue Tracking System

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Issue Tracking > Create*.
   The following screen is displayed:



2. Fill out the fields in the *Create Issue Tracking System* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	Enter the Name of the external Issue Tracking System. This field is the only mandatory field. However, if only the Name is provided, the integration capability is confined to being very basic, meaning that issues will have to be created and linked manually to the IKAN ALM Level Requests.
Description	Enter a description for the Issue Tracking System.
Plugin Factory Class	Select the name of the class to be used for the Issue Tracking System. The classes are available in IKAN ALM as system functions: • "generic" system (Bugzilla, Track, Mantis, ...): be.ikan.scm4all.plugin.issuetracking.generic.GenericITSPluginFactory IKAN ALM has a basic integration with these Issue Tracking System. • JIRA:be.ikan.scm4all.plugin.issuetracking.jira.JiraITSPluginFactory • HP Quality Center: be.ikan.scm4all.plugin.issuetracking.hpqc.HPQcITSPluginFactory • CollabNet TeamForge: be.ikan.scm4all.plugin.issuetracking.teamforge.TeamForgeITSPluginFactory • Microsoft Team Foundation: be.ikan.scm4all.plugin.issuetracking.tfs.TFSITSPluginFactory **Note:** The integration with JIRA, HP ALM, Team Foundation and TeamForge also retrieves information about individual issues, such as status, description and owner. Issues are synchronized with the external Issue Tracking System at each Level Request deliver in the Lifecycle.
URL	Enter the template URL used to click through from an issue in IKAN ALM to the issue in the external Issue Tracking System. Obviously, the Issue Tracking System must have a web interface that will guide you (perhaps after having provided the necessary login parameters) to the detailed description of an Issue. In the URL, the issue number variable must be provided as "${issueId}". Some examples: For JIRA: `http(s)://host:port/browse/PROJECTKEY-${issueId}` For HP Quality Center: `testdirector:host:port/qcbin,domainname,projectname,[AnyUser];2:${issueId}` For TeamForge: `http(s)://host:port/sf/go/${issueId}` For Trac: `http(s)://host/ticket/${issueId}` For Bugzilla: `http(s)/host/bugs/show_bug.cgi?id=${issueId}` For Team Foundation: `http(s)://host[:port/tfs]/DefaultCollection/PROJECT_NAME/_workitems#_a=edit&id=${issueId}` **Note:** IKAN ALM Users who will use the "testdirector" command to establish the link with HP Quality Center must install the necessary HP "Add-ins". Refer to the HP documentation for more detailed information.
User	Enter the User ID having the necessary rights to connect to the system.
Password	Enter the Password for the User ID. The characters you enter are displayed as asterisks.
Repeat Password	Re-enter the Password for the User ID.

Field	Meaning
Issue Pattern	Enter the Issue Pattern. This Issue Pattern must be a valid regular expression. For more information on how to provide valid regular expressions that may be interpreted by IKAN ALM, refer to http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html. If provided, this pattern will be used to detect issue numbers in the commit comments in the VCR (currently supported for Subversion, Git, Team Foundation and CVS). This pattern matching is done in a separate Phase at the end of a successful Build Level Request. The pattern matching is case insensitive, this is reflected in the examples below. Note that the field may be neglected (together with the Issue ID Pattern) for a Team Foundation ITS when it is connected to a Project with a Team Foundation versioning system: in that case the Work Items are directly connected to a Commit, so it's not necessary that IKAN ALM parses the issue comments to detect the connected Work Items **Examples** (the bold text is the text that will be matched): Example 1: webpad(\s)*[0-9]+((\s)*,(\s)*[0-9]+)* - Solving issue **webpad 333** by adapting.... - Added file x, and changed file y in order to solve **Webpad 45, 46**. Example 2: \[#([A-Z0-9]+)-([0-9]+)\] - Small fix in the about menu **[#WEBPAD-7]**.- Also fixed a general IKAN ALM problem **[#ALM-3788]**.- Finally also tackled issue **[#gen-344]**.
Issue ID Pattern	Enter the Issue ID Pattern. This pattern is needed to retrieve the exact Issue Number out of a matched Issue in the commit comment. It is part of the Issue Pattern and must also be a valid regular expression. This will allow to select the numbers in bold in the comments of the VCR. The field may be neglected (together with the Issue Pattern) for a Team Foundation ITS when it is connected to a Project with a Team Foundation versioning system. Example 1: `[0-9]+` - Webpad **45** , **46** - [#gen-344] Example 2: `[A-Z0-9]+`-[0-9]+ - **WEBPAD-7** - **ALM-3788**
Add Comments	This option has no effect if you use the GenericITSPlugin. Select the *Yes* option to automatically add a comment to the Issue in the Issue Tracking System. This comment will contain a link to the IKAN ALM Level Request. The addition of this comment will also be listed in the Level Request Issue Tracking Phase Log.

Note: For more detailed information on the specific Issue Tracking system settings, refer to the appropriate Integration Guide (HowToALM_Integrating HPALM TestRunner, HowToALM_Integrating HPALM QualityCenter, HowToALM_Integrating JIRA, HowToALM_Integrating TeamForge or HowToALM_Integrating Team Foundation).

3. Once you have filled out the fields, click *Create*.

   A warning may appear indicating that some required properties must be set. Refer to the section The Issue Tracking System Properties Overview Panel (page 390) for editing the Issue Tracking System properties.

   WARNING: NO VALUE SET FOR REQUIRED PROPERTIES: JIRARESTURL

   Once the Issue Tracking System has been defined, you must activate it on the Project. See Editing Project Settings on page 149.

4. In case you defined a JIRA, HP ALM, TeamForge or Team Foundation Issue Tracking System, you can test if IKAN ALM can establish the connection.

   See Editing an Issue Tracking System Definition on page 387.

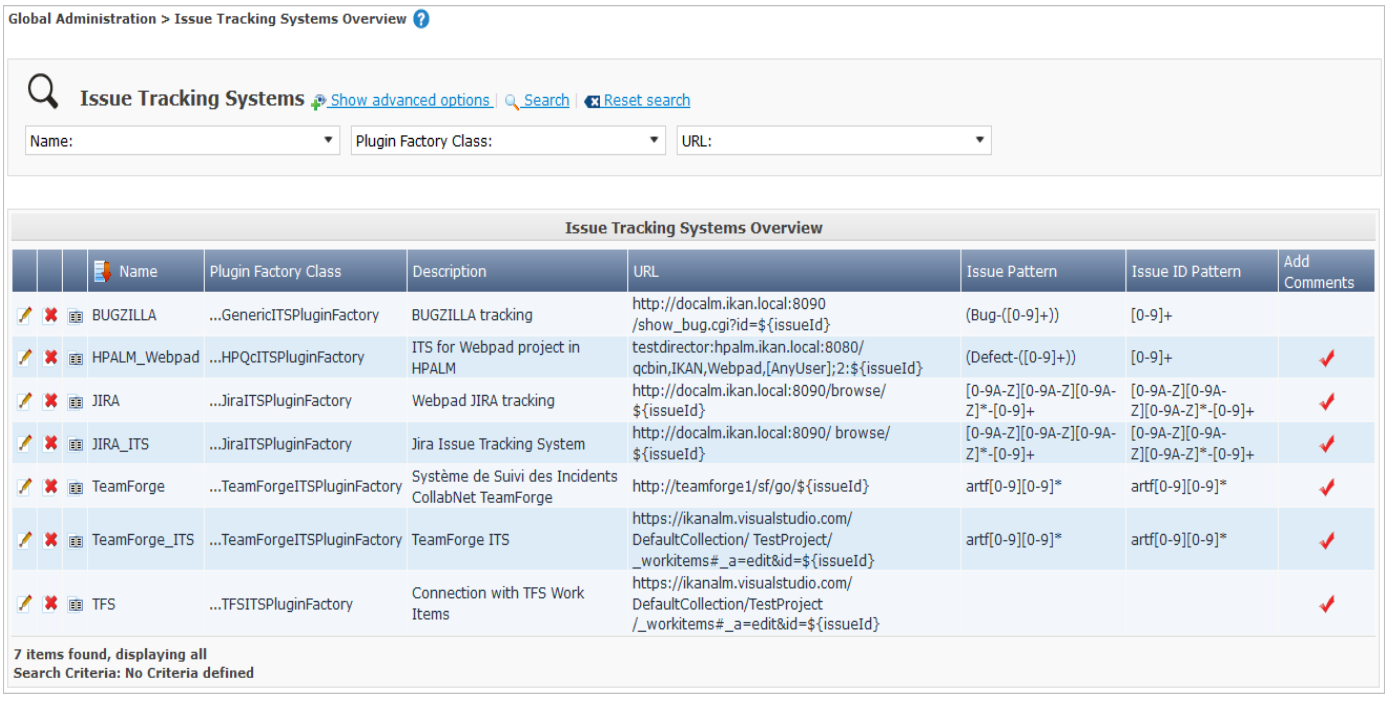
RELATED TOPICS

- Editing Project Settings (page 149)
- Issues (page 93)
- Issue Tracking (page 381)
- Issue Tracking Phase (page 524)

30.2. The Issue Tracking Systems Overview Screen

1. In the Global Administration context, select *Issue Tracking > Overview*.

   The following screen is displayed:



2. Define the required search criteria on the search panel.

   The list of items on the overview will be automatically updated based on the selected criteria.

You can also:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- click the *Search* link to refresh the list based on the current search criteria,
- click the *Reset search* link to clear the search fields,

3. Verify the information on the *Issue Tracking Systems Overview* panel.
   For a detailed description of the fields, refer to Creating an Issue Tracking System (page 382).

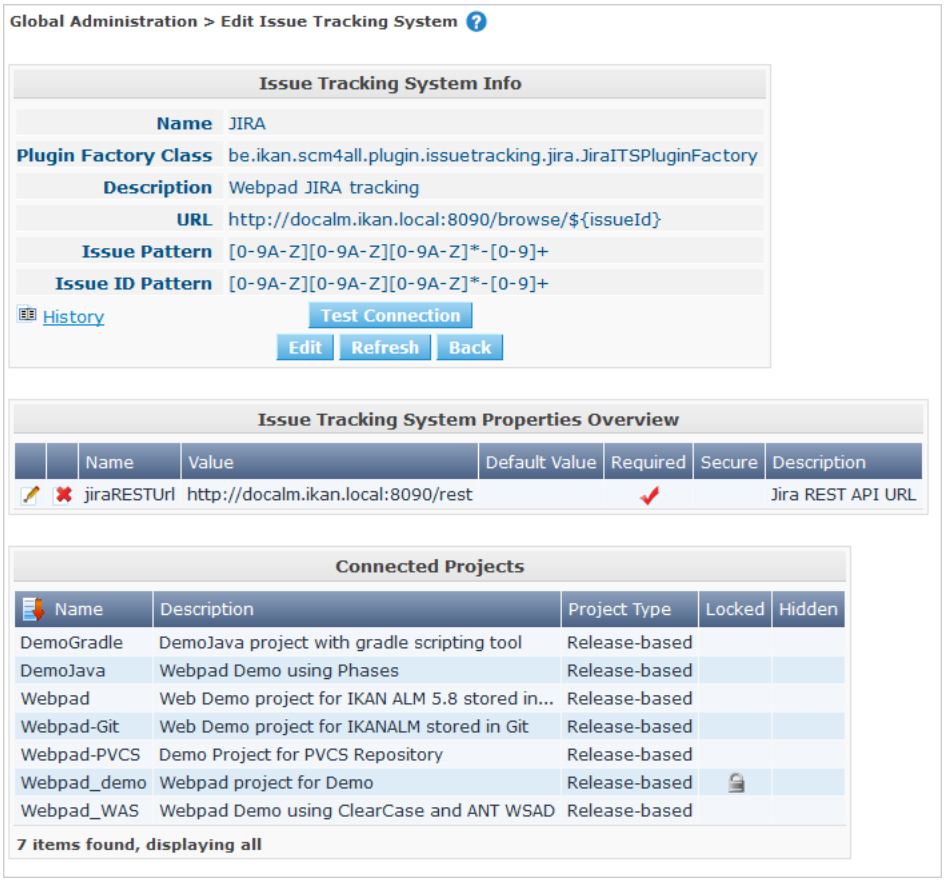
4. Depending on your access rights, the following links may be available on the *Issue Tracking Systems Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing an Issue Tracking System definition. See Editing an Issue Tracking System Definition on page 387.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting an Issue Tracking System definition. See Deleting an Issue Tracking System Definition on page 389.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on an Issue Tracking System and its properties. See Viewing the Issue Tracking System History on page 390.

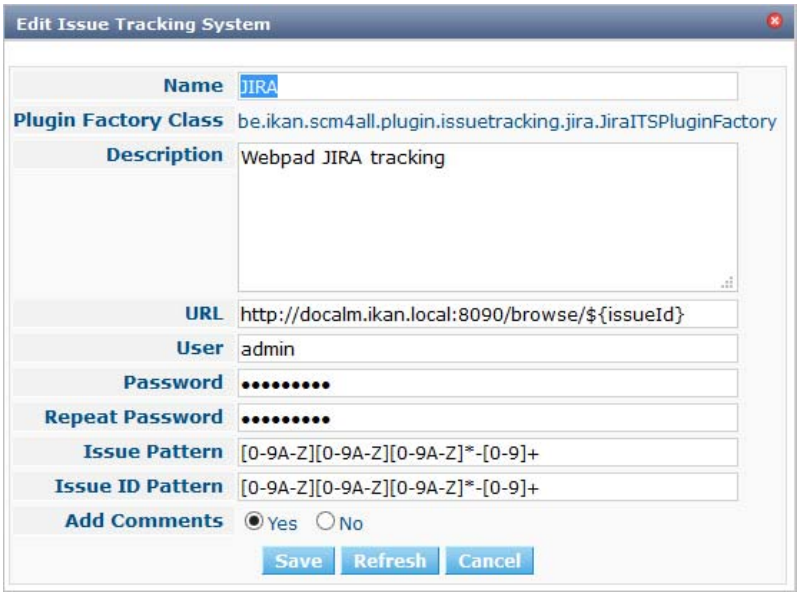
Editing an Issue Tracking System Definition

1. In the Global Administration context, select *Issue Tracking > Overview*.

2. Click the *Edit* link in front of the Issue Tracking System you want to modify. The following screen is displayed:



3. Click the *Edit* button on the *Issue Tracking System Info* panel.
   The following screen is displayed:



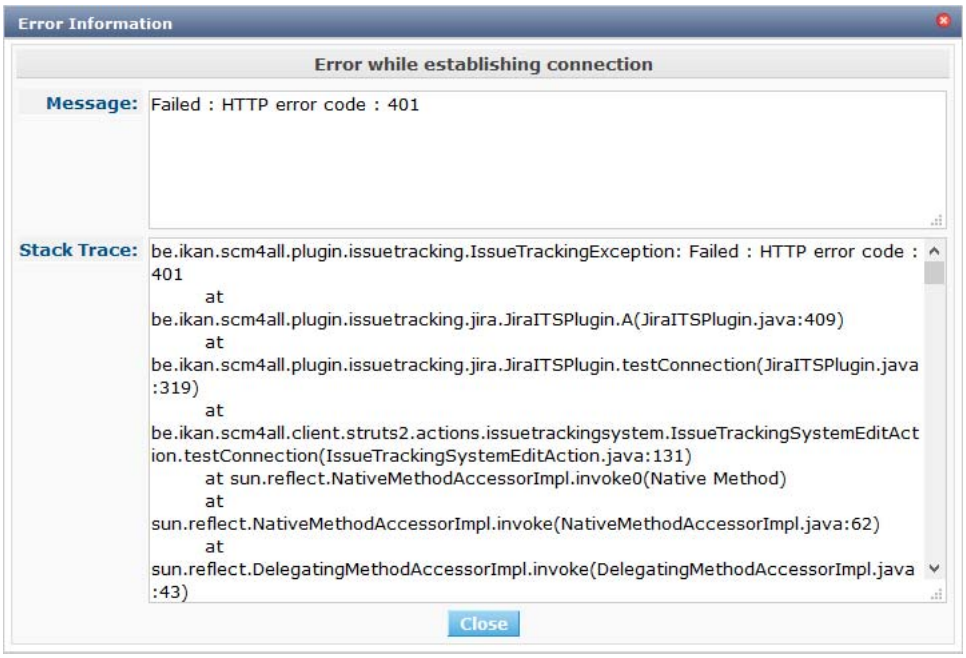
4. Edit the fields as required.
   For a description of the fields, refer to Creating an Issue Tracking System (page 382).

Note: The *Connected Projects* panel displays the Projects the Issue Tracking System is linked to.

5. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Cancel* to return to the previous screen without saving the changes

6. In case you defined a JIRA, HP ALM, TeamForge or Team Foundation Issue Tracking System with its required properties, you can test if IKAN ALM can establish the connection.
   Click the *Test Connection* button.
   Info: Could successfully establish a connection with the Issue Tracking System.

If the test is not successful, the following screen is displayed:



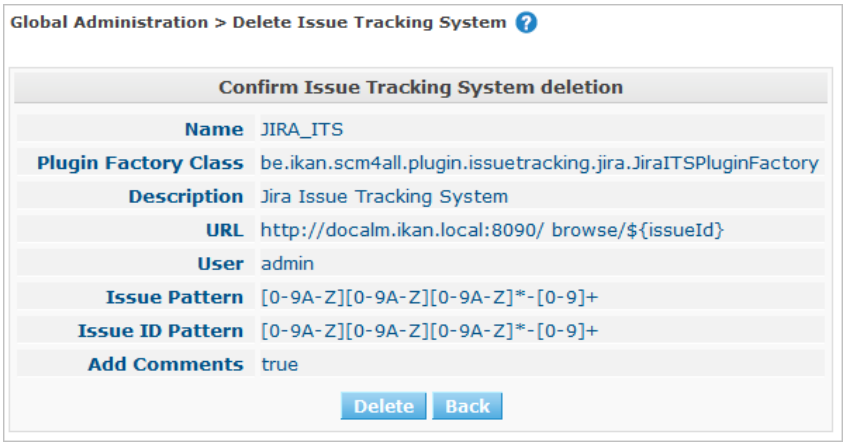
Correct the errors reported in the Stack Trace field and perform the test again.

7. On the *Issue Tracking System Properties Overview* panel, you can create and edit the Isue Tracking System Properties.
   For more information, refer to the section The Issue Tracking System Properties Overview Panel (page 390)

Deleting an Issue Tracking System Definition

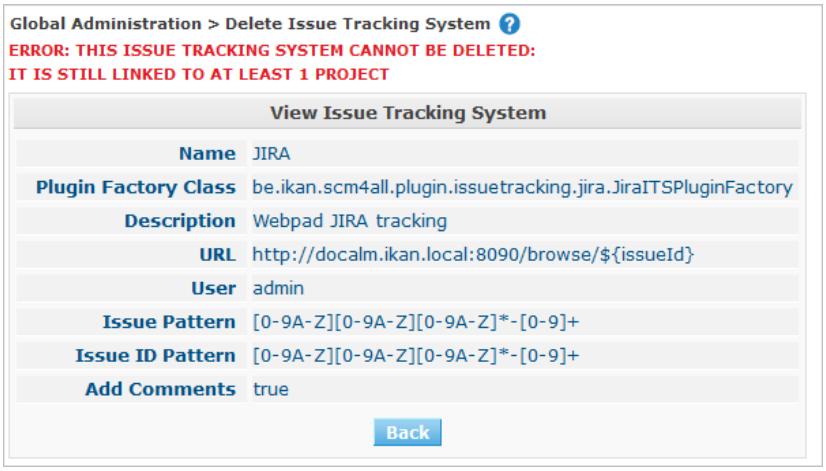
1. In the Global Administration context, select *Issue Tracking > Overview*.

2. Click the *Delete* link to delete the selected Issue Tracking System definition.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.

Note: If the Issue Tracking System is still linked to one (or more) Project(s), the following screen is displayed:



You must change the definition of the listed Projects, before you can delete the Issue Tracking System.

Viewing the Issue Tracking System History

1. In the Global Administration context, select *Issue Tracking > Overview*.

2. Click the *History* link to display the *Issue Tracking System History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Issue Tracking Systems Overview* screen.

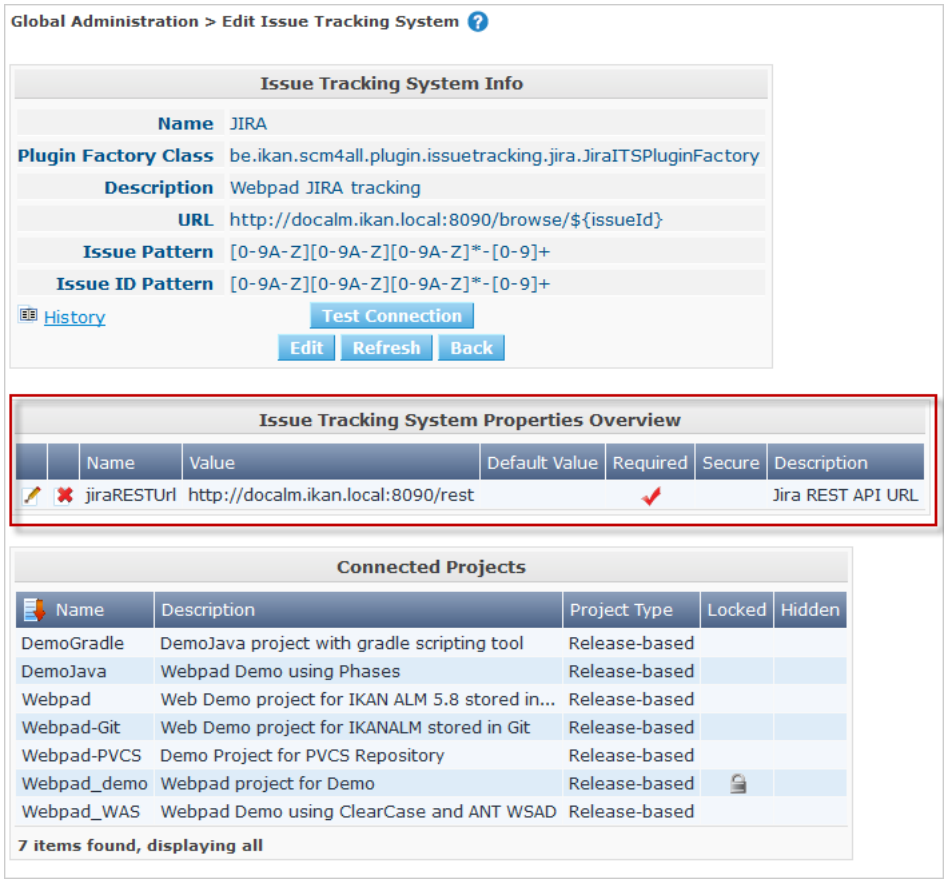
RELATED TOPICS
- Editing Project Settings (page 149)
- Issues (page 93)
- Issue Tracking (page 381)
- Issue Tracking Phase (page 524)

30.3. The Issue Tracking System Properties Overview Panel

The Issue Tracking System Properties Overview panel is available on the *Edit Issue Tracking System* screen.

1. Access the *Edit Issue Tracking System* screen.
   In the Global Administration context, select *Issue Tracking > Overview* and click the *Edit* link in front of the Issue Tracking System for which you want to display the Properties.

2. This screen contains the *Issue Tracking System Properties Overview* panel.



This screen lets you create, edit or delete ITS Properties.

Depending on the Properties available in the Issue Tracking System and on your access rights, the following links may be available on the *Issue Tracking Systems Properties Overview* panel:

Note: You can only define Properties which have been implemented by the *Plugin Factory Class*.

	Create This link is only available if a Property has been defined by the Plugin Factory Class, but its value has not been specified yet. Otherwise, only the links *Edit*, *Delete* and *History* are available. This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows creating an Issue Tracking System Property definition. See Creating an Issue Tracking System Property on page 392.
	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing an Issue Tracking System Property definition. See Editing an Issue Tracking System Property on page 393.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting an Issue Tracking System Property definition. See Deleting an Issue Tracking System Property on page 393.

Creating an Issue Tracking System Property

1. On the *Issue Tracking System Properties Overview* panel, click the *Create* link for the required Property.

   The following screen is displayed:

   The following fields are displayed:

Field	Meaning
Name	The name is predefined by IKAN ALM in function of the type of Issue Tracking System.
Value	Depending on the Issue Tracking system and the property, this field is mandatory or optional. Enter the value needed for correct usage of the Issue Tracking System. Example for the JIRA jiraRESTUrl property: `http(s)://machine:8090/rest` Example for the TeamForge teamForgeWSUrl property: `http(s)://teamforge1.my.domain` Example for the Team Foundation collectionURL property: `http(s)://ServerName[:8080/tfs]/DefaultCollection`
Default Value	This field contains the default value.
Required	Option managed by IKAN ALM. Required properties must be provided in order to have a full functional Issue Tracking integration. Possible values: *Yes* or *No*.
Secure	Option managed by IKAN ALM. The value of secured properties will be hidden (replaced by *) from the user. Possible values: *Yes* or *No*.
Description	Option managed by IKAN ALM. The description for the Issue tracking System Property.

2. Fill out the value in the *Value* field and click *Create* to confirm the creation of the new Property.

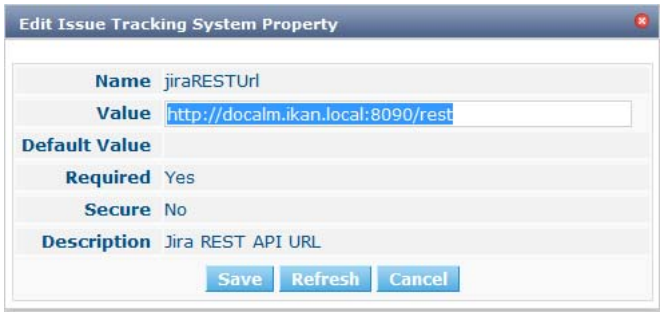
   You can also click:

   - *Reset* to clear the fields and restore the initial values.
   - *Cancel* to return to the previous screen without saving your changes.

Editing an Issue Tracking System Property

1. On the *Issue Tracking System Properties Overview* panel, click the *Edit* link for the required Property.

   The following screen is displayed:



   For a description of the fields, refer to the section Creating an Issue Tracking System Property (page 392).

2. If required, modify the value in the *Value* field and click *Save*.

   You can also click:

   - *Refresh*: to retrieve the settings from the database.
   - *Cancel*: to return to the previous screen without saving the changes to the fields.

Deleting an Issue Tracking System Property

1. On the *Issue Tracking System Properties Overview* panel, click the *Delete link* for the required Property.

   The following screen is displayed:

2. Click *Delete* to confirm the Deletion of the Property.

   You can also click Cancel to return to the Issue Tracking Systems Overview without deleting the Property.

CHAPTER 31
Scripting Tools

IKAN ALM supports the following Scripting Tools:

- ANT (page 396)
- Gradle (page 403)
- NANT (page 410)
- Maven2 (page 416)

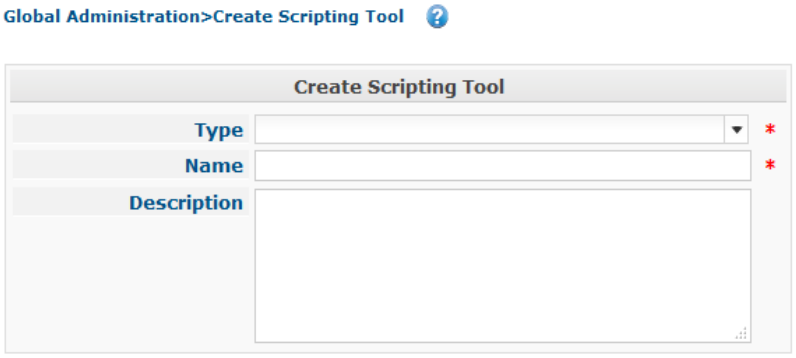
The following sections explain how to define a Scripting Tool so that the IKAN ALM Agent may use it for Build and Deploy actions.

Note: You must always install the defined Scripting Tool on the IKAN ALM Agent in order to execute Build and Deploy scripts.

31.1. Creating a Scripting Tool

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Scripting Tools > Create*.
   The following screen is displayed:



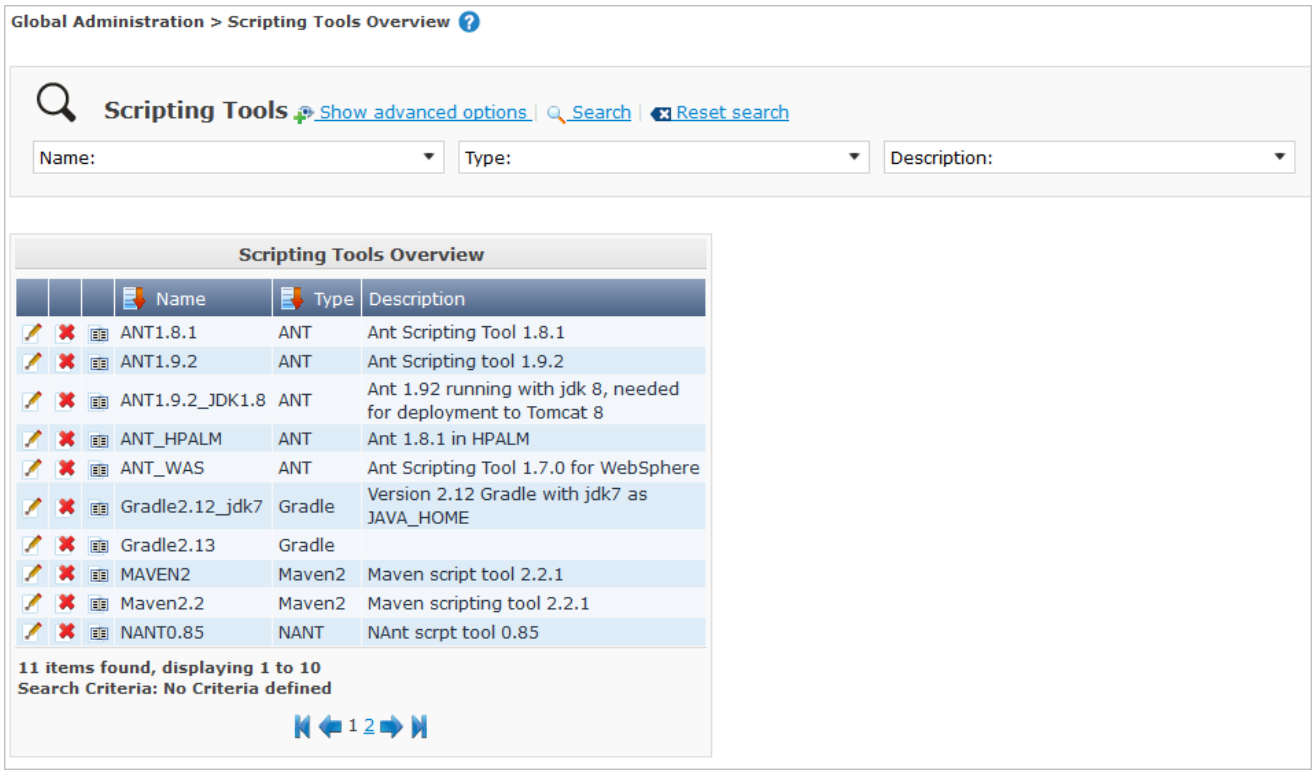
2. Select the *Type* of Scripting Tool you want to create.
   The *Details* panel for the type of Scripting Tool you selected is displayed underneath.

3. Enter the *Name* of the Scripting Tool and add an optional *Description*.

4. Fill out the Scripting Tool-specific Details.
   For more information on those details, refer to the appropriate sections:
   - Creating an ANT Scripting Tool (page 397)
   - Creating a Gradle Scripting Tool (page 404)
   - Creating a NANT Scripting Tool (page 411)
   - Creating a Maven2 Scripting Tool (page 417)

31.2. The Scripting Tools Overview Screen

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. Verify the information on the *Scripting Tools Overview* panel.

4. Depending on your access rights, the following links may be available on the *Scripting Tools Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Scripting Tool definition.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Scripting Tool definition.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Scripting Tool.

Refer to the following sections for detailed information:

- ANT Scripting Tools (page 396)
- Gradle Scripting Tools (page 403)
- NANT Scripting Tools (page 410)
- Maven2 Scripting Tools (page 416)

31.3. ANT Scripting Tools

Apache Ant is a Java-based build tool. In theory, it resembles to *make*, without *make*'s wrinkles. If ANT is installed on a machine connected to your Build/Deploy Environments, IKAN ALM will be able to interact with it. Therefore, you must define the ANT Scripting Tool in the IKAN ALM application and, when creating the Build or Deploy Environment, specify it as being the Build or Deploy Tool to be used.

Refer to the following sections for detailed information:

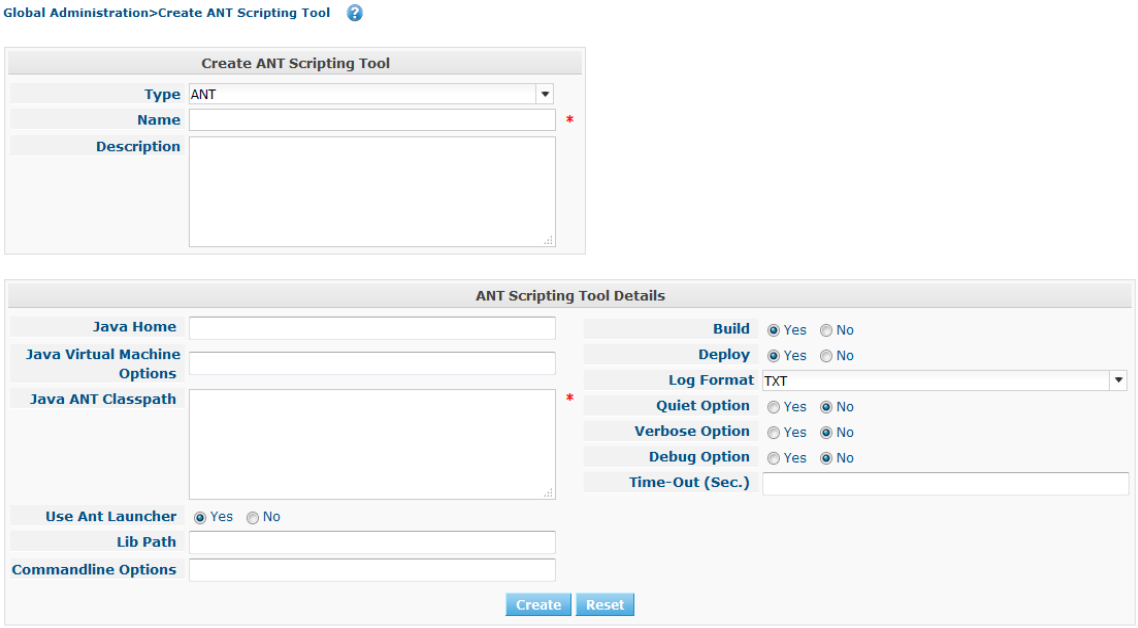
- Creating an ANT Scripting Tool (page 397)
- The ANT Scripting Tools Overview Screen (page 400)
- Editing an ANT Scripting Tool Definition (page 401)
- Deleting an ANT Scripting Tool Definition (page 402)
- Viewing the ANT Scripting Tool History (page 403)

Creating an ANT Scripting Tool

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Scripting Tools* > *Create*.

2. Select *ANT* from the drop-down list in the *Type* field on the search panel.
   The following screen is displayed:



3. Fill out the fields in the *Create ANT Scripting Tool* panel at the top of the screen.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Type	Select the type of Scripting Tool you want to define. This field is mandatory. After you have selected the type, the appropriate *Details* panel will be displayed underneath.
Name	Enter the name of the new ANT Scripting Tool in this field. This field is mandatory.
Description	Enter a description for the new ANT Scripting Tool in this field. This field is optional.

4. Fill out the fields in the *ANT Scripting Tool Details* panel.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Java Home	Enter the *JAVA_HOME* path to launch ANT in this field. If you do not enter a value in this field, the system default *JAVA_HOME* path will be used. In this case, the *JAVA_HOME* path must be defined as an environment variable on the Machine.
Java Virtual Machine Options	In this field, enter the Java Virtual Machine Options required for starting up ANT. This field is optional. Example: *-Xmx128M*: specifies the maximum size of the memory allocation pool. JVM Options of Java 8 for example are explained here: https://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html
Java ANT Classpath	Enter the JAVA ANT Classpath in this field. This field is mandatory. Since IKAN ALM launches ANT via Java, you must ensure that all required jar files are available. Some values can be set in this field, but also the Lib Patch can be used (see below). Depending on the ANT version you use, the CLASSPATH should at least include: **For ANT 1.5.x:** • ant.jar • jars/classes for your XML parser • jars/zip files for the JDK **For ANT 1.6.x or higher:** • ant.jar • ant-launcher.jar • jars/classes for your XML parser • jars/zip files for the JDK If you launch ANT via the Ant Launcher, ensure that the CLASSPATH includes: • ant-launcher.jar • required external dependencies (such as tools.jar) *Note:* If you use Ant Launcher, do not include *ant.jar* in the Java Classpath, because the dependencies will not be found and the script will end in error. Also refer to the section *Running ANT via Java* in the Apache Ant Manual.
Use Ant Launcher	Select *Yes*, if you want to use the Ant Launcher (default). Select *No*, if you do not use the Ant Launcher, but launch Ant via Java.
Lib Path	When using the Ant Launcher, you may enter the path to one or more directories containing additional required jar files (-lib option).
Commandline Options	Enter commandline options in this fields. This field is optional. The commandline options provided by default by IKAN ALM, like -buildfile, -logfiles, -verbose, -debug ,… will not be accepted. Sample options are -keep-going, or -noinput.
Build	Indicate whether or not this Scripting Tool can be linked to a Build Environment and thus be used to build code.
Deploy	Indicate whether or not this Scripting Tool can be linked to a Deploy Environment and thus be used to deploy Builds.

Field	Meaning
Log Format	Select the format of the log file generated by this ANT Scripting Tool. The following options are available: • TXT: the log file will be a simple text file • XML: the log file will be an XML file TXT logs will be smaller and their content can be displayed while the Tool is executing. The display of XML logs can be customized by providing custom XSL templates. However, XML logs are larger and can only be displayed after the Tool has finished executing.
Quiet Option	Select whether or not the Quiet Option must be activated for the new ANT Scripting Tool. ANT will print less information than normal during the Build and Deploy actions, if this option is activated.
Verbose Option	Select whether or not the Verbose Option must be activated for the new ANT Scripting Tool. ANT will print verbose debug information during the Build and Deploy actions, if this option is activated.
Debug Option	Select whether or not the Debug Option must be activated for the new ANT Scripting Tool. ANT will print additional debug information during the Build and Deploy actions, if this option is activated.
Time-Out (Sec.)	In this field, enter a Time-Out value if required. If a value is provided, a running ANT Build or Deploy process will be interrupted after this number of seconds. In this way, "hanging" Build or Deploy processes are interrupted. If no value is provided, a running ANT Build or Deploy process will never be interrupted.

5. Once you filled out the fields as required, click *Create*.

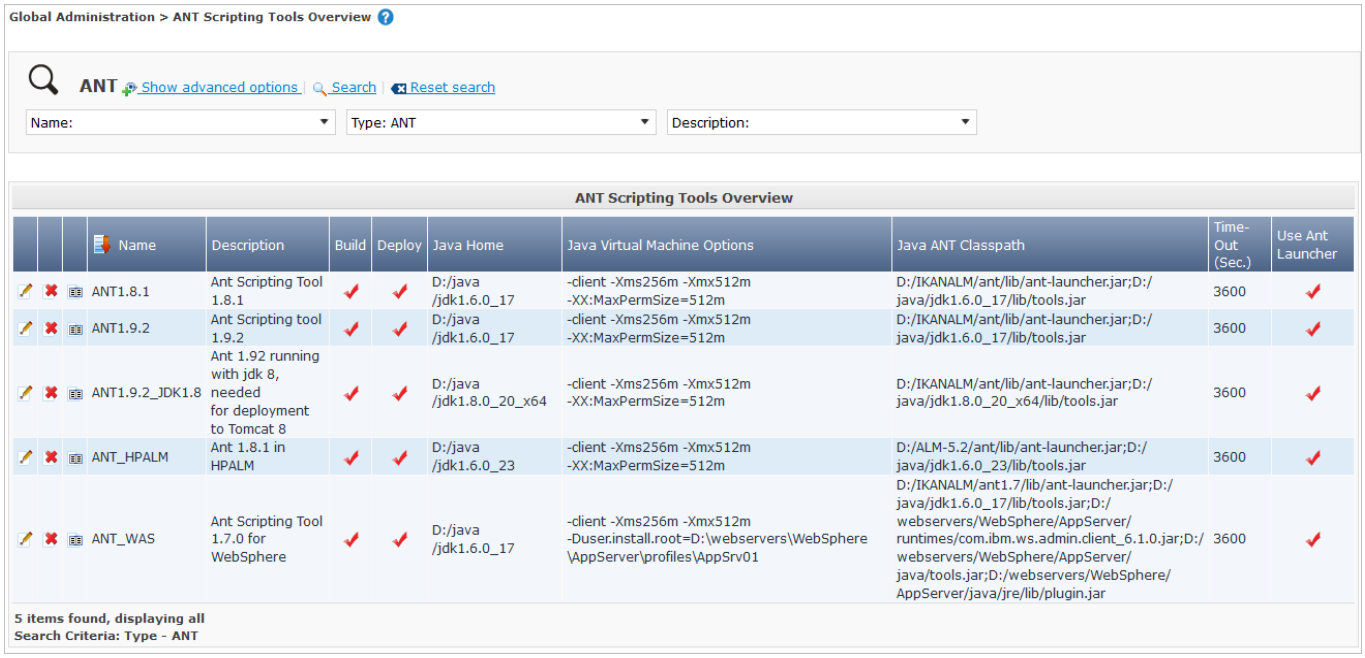
   The newly created ANT Scripting Tool is added to the *ANT Scripting Tools Overview* at the bottom of the screen.

RELATED TOPICS
- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

The ANT Scripting Tools Overview Screen

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.

2. Specify *ANT* in the *Type* field on the search panel.



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:

   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *Scripting Tools Overview* panel.
   For a detailed description of the fields, refer to Creating an ANT Scripting Tool (page 397).

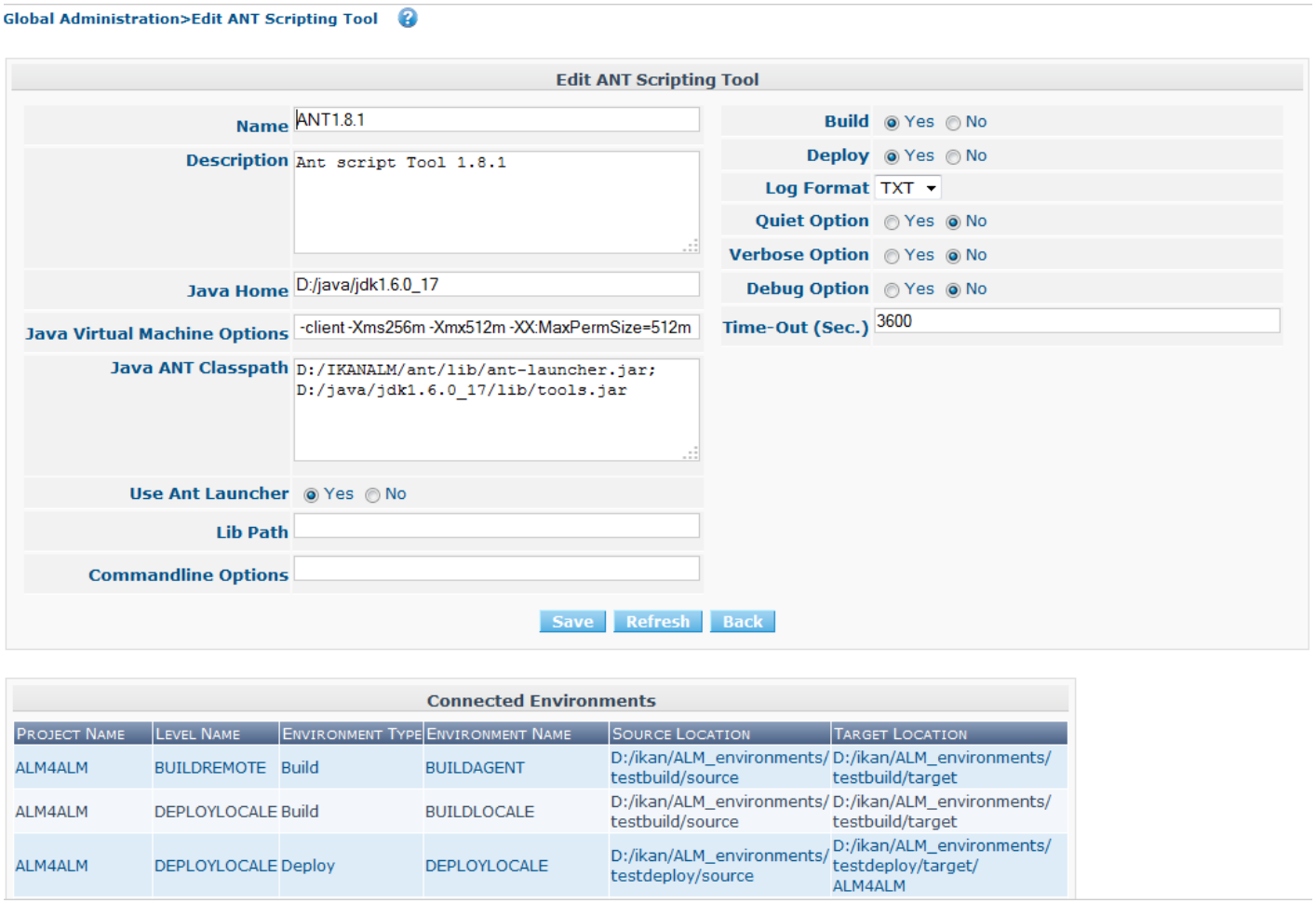
4. Depending on your access rights, the following links may be available on the *ANT Scripting Tools Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Scripting Tool definition.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Scripting Tool definition.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Scripting Tool

Editing an ANT Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the ANT Scripting Tools your are looking for.

2. Click the *Edit* link to change the selected ANT Scripting Tool.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating an ANT Scripting Tool (page 397).

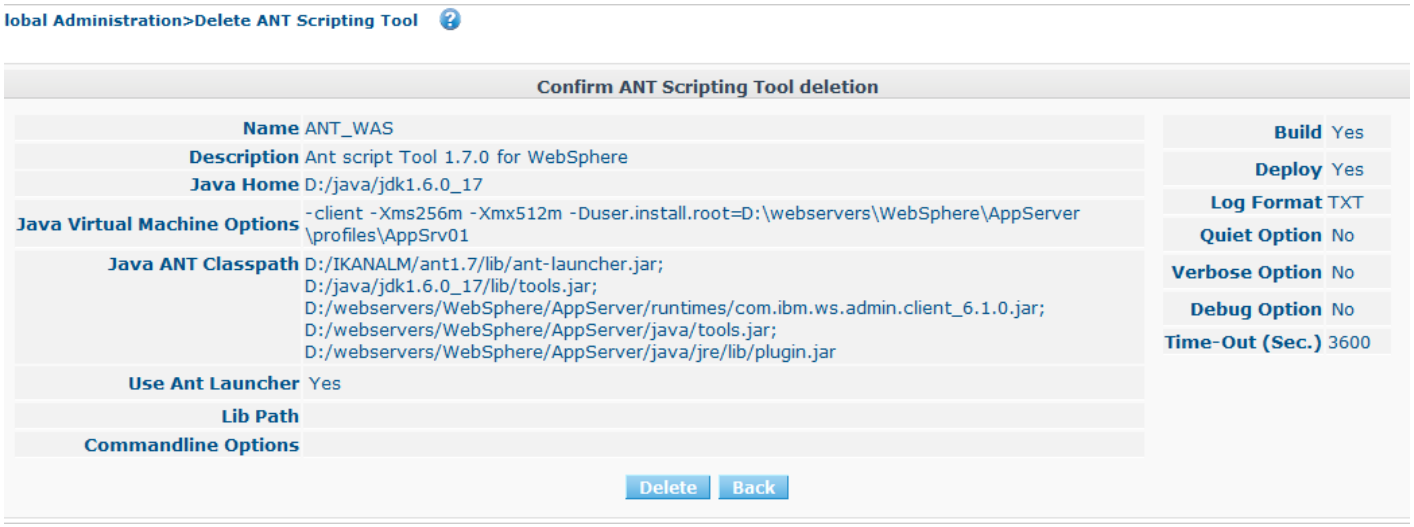
   Note: The *Connected Environments* panel displays the Environments the Scripting Tool is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

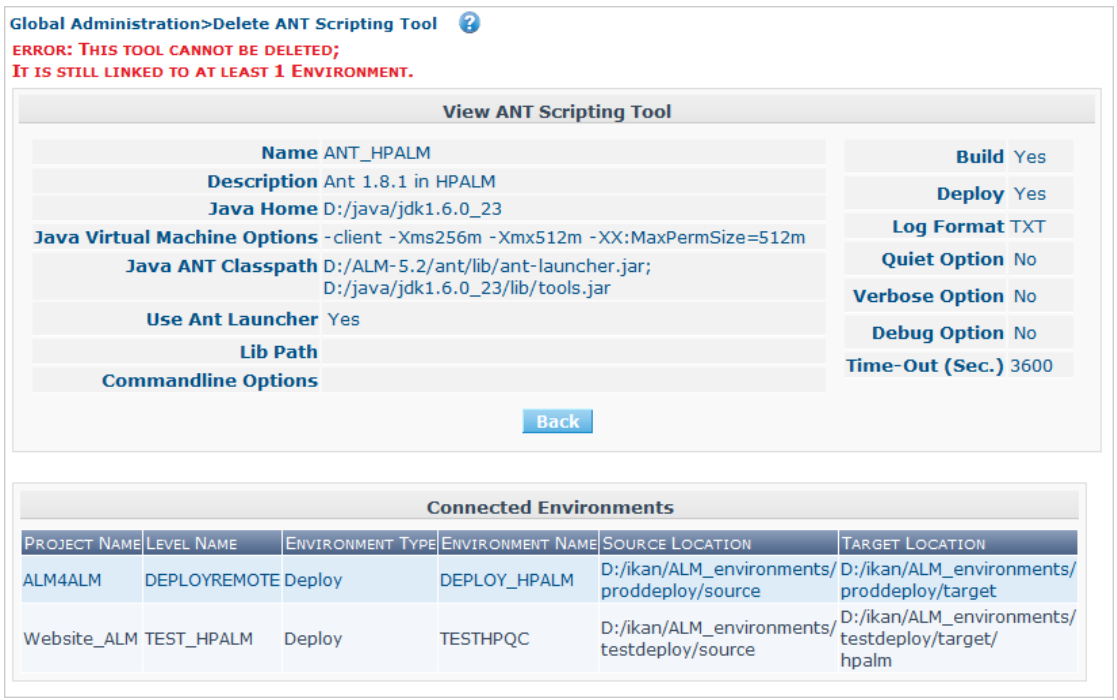
Deleting an ANT Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the ANT Scripting Tools your are looking for.

2. Click the *Delete* link to delete the selected ANT Scripting Tool.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the ANT Scripting Tool is linked to one or more Build or Deploy Environments, the following screen is displayed:



You need to assign another Scripting Tool to these Environments, before deleting this ANT Scripting Tool.

Viewing the ANT Scripting Tool History

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the ANT Scripting Tools your are looking for.

2. Click the *History* link to display the *ANT Scripting Tool History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Scripting Tools Overview* screen.

RELATED TOPICS
- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

31.4. Gradle Scripting Tools

Gradle is an open source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used by Apache Maven for declaring the project configuration. If Gradle is installed on a machine connected to your Build/Deploy Environments, IKAN ALM will be able to interact with it. Therefore, you must define the Gradle Scripting Tool in the IKAN ALM application and, when creating the Build or Deploy Environment, specify it as being the Build or Deploy Tool to be used.

Refer to the following sections for detailed information:

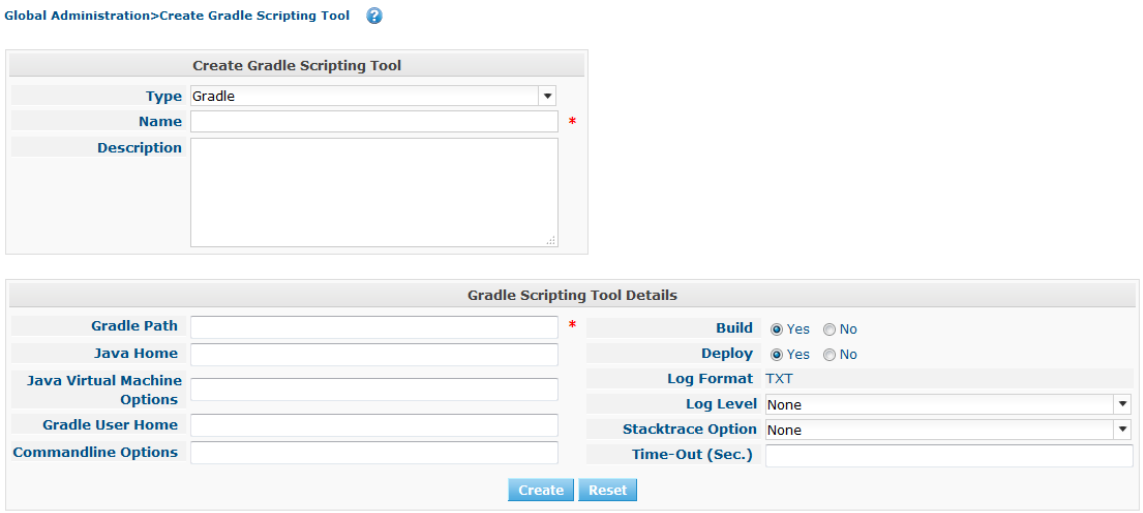
- Creating a Gradle Scripting Tool (page 404)
- The Gradle Scripting Tools Overview Screen (page 407)
- Editing a Gradle Scripting Tool Definition (page 408)
- Deleting a Gradle Scripting Tool Definition (page 409)
- Viewing the Gradle Scripting Tool History (page 410)

Creating a Gradle Scripting Tool

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Scripting Tools* > *Create*.

2. Select *Gradle* from the drop-down list in the *Type* field on the search panel.
   The following screen is displayed:



3. Fill out the fields in the *Create Gradle Scripting Tool* panel at the top of the screen.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Type	Select the type of Scripting Tool you want to define. This field is mandatory. After you have selected the type, the appropriate *Details* panel will be displayed underneath.
Name	Enter the name of the new Gradle Scripting Tool in this field. This field is mandatory.
Description	Enter a description for the new Gradle Scripting Tool in this field. This field is optional.

4. Fill out the fields in the *Gradle Scripting Tool Details* panel.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Gradle Path	This field is mandatory. Enter the path to the Gradle bat (gradle.bat-Windows) or shell file (e.g., gradle - linux). Example: `d:/javatools/gradle2.10/bin` or `/opt/javatools/gradle2.10/bin`
Java Home	Enter the *JAVA_HOME* path to launch Gradle in this field. If you do not enter a value in this field, the system default *JAVA_HOME* path will be used. In this case, the *JAVA_HOME* path must be defined as an environment variable on the Machine.
Java Virtual Machine Options	In this field, enter the Java Virtual Machine Options required for starting up Gradle. This field is optional. Example: *-Xmx128M*: specifies the maximum size of the memory allocation pool. JVM Options of Java 8 for example are explained here: https://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html
Gradle User Home	This field is optional. Enter the path to the location of the Gradle User Home, this is the location where (among other things) the Gradle dependency cache will be stored. If you do not enter a value in this field, the default *User Home* will be used. This is particularly useful on Linux, where the user running the Agent may lack a User Home. Example: `/opt/gradle_user_home` **Note:** This value will be set as a commandline option (--gradle-user-home).
Commandline Options	Enter commandline options in this fields. This field is optional. The commandline options which might be provided by default by IKAN ALM will not be accepted: .-g, --gradle-user-home, -q, --quiet, -i, --info,-d, --debug, -s, --stacktrace, -S, --full-stacktrace, -b, --build-file Sample (accepted) options are -keep-going, or -noinput.
Build	Indicate whether or not this Scripting Tool can be linked to a Build Environment and thus be used to build code.
Deploy	Indicate whether or not this Scripting Tool can be linked to a Deploy Environment and thus be used to deploy Builds.
Log Format	By default, the generated log file will be a text file. This format cannot be modified.

Field	Meaning
Log Level	Refer to the section https://docs.gradle.org/current/userguide/logging.html in the Gradle User Guide for more info regarding the Log Level and the Stacktrace From the drop-down list, select which logging options should be activated. The following options are available: • None No specific command line options will be set for the Log Level, resulting in the default (normal) LIFECYCLE Log Level: Gradle will print progress information messages. • Quiet Gradle will only print the Important Information messages, so less information than normal (LIFECYCLE level) during the Build and Deploy actions. -q will be added as a commandline option. • Info Gradle will print information messages during the Build and Deploy actions, so more information than normal (LIFECYCLE level). -i will be added as a commandline option. • Debug Gradle will print additional debug information during the Build and Deploy actions. -d will be added as a commandline option.
Stack Trace Option	From the drop-down list, select the required Stack Trace option. The following options are available: • None No stack traces are printed to the console in case of a build error (e.g., a compile error). Stack traces will only be printed in case of internal exceptions. If the option DEBUG log level is chosen, truncated stack traces are always printed. • Truncated Truncated stack traces are printed. We recommend this over full stack traces. Groovy full stack traces are extremely verbose (due to the underlying dynamic invocation mechanisms). They, however, usually do not contain relevant information to find out what has gone wrong in your code. -s will be added as a commandline option. • Full Full stack traces are printed out. -S will be added as a commandline option.
Time-Out (Sec.)	In this field, enter a Time-Out value if required. If a value is provided, a running Gradle Build or Deploy process will be interrupted after this number of seconds. In this way, "hanging" Build or Deploy processes are interrupted. If no value is provided, a running Gradle Build or Deploy process will never be interrupted.

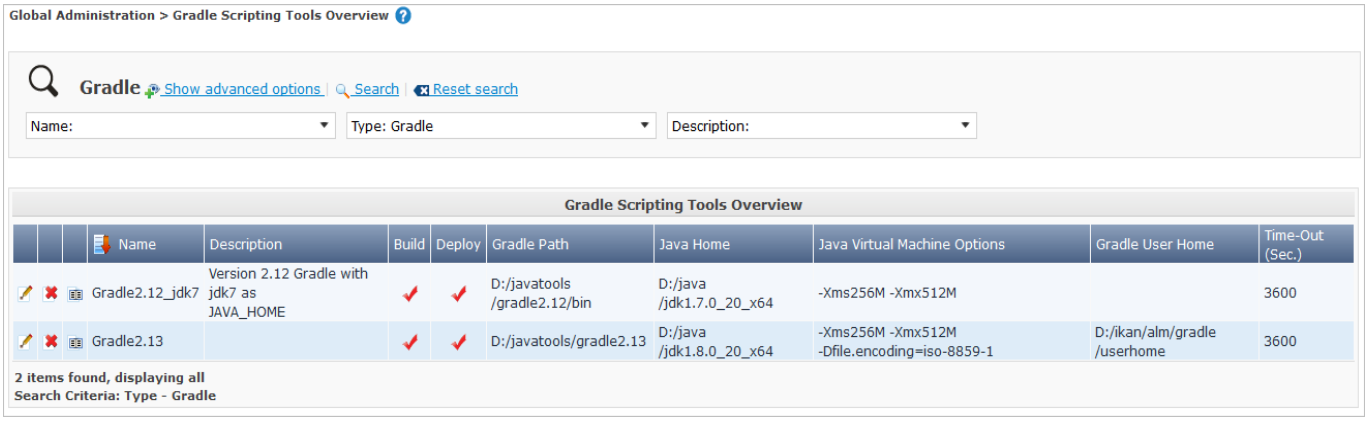
5. Once you filled out the fields as required, click *Create*.

   The newly created Gradle Scripting Tool is added to the *Gradle Scripting Tools Overview* at the bottom of the screen.

The Gradle Scripting Tools Overview Screen

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.

2. Specify *Gradle* in the *Type* field on the search panel.



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:

   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *Scripting Tools Overview* panel.
   For a detailed description of the fields, refer to The Gradle Scripting Tools Overview Screen (page 407).

4. Depending on your access rights, the following links may be available on the *Gradle Scripting Tools Overview* panel:

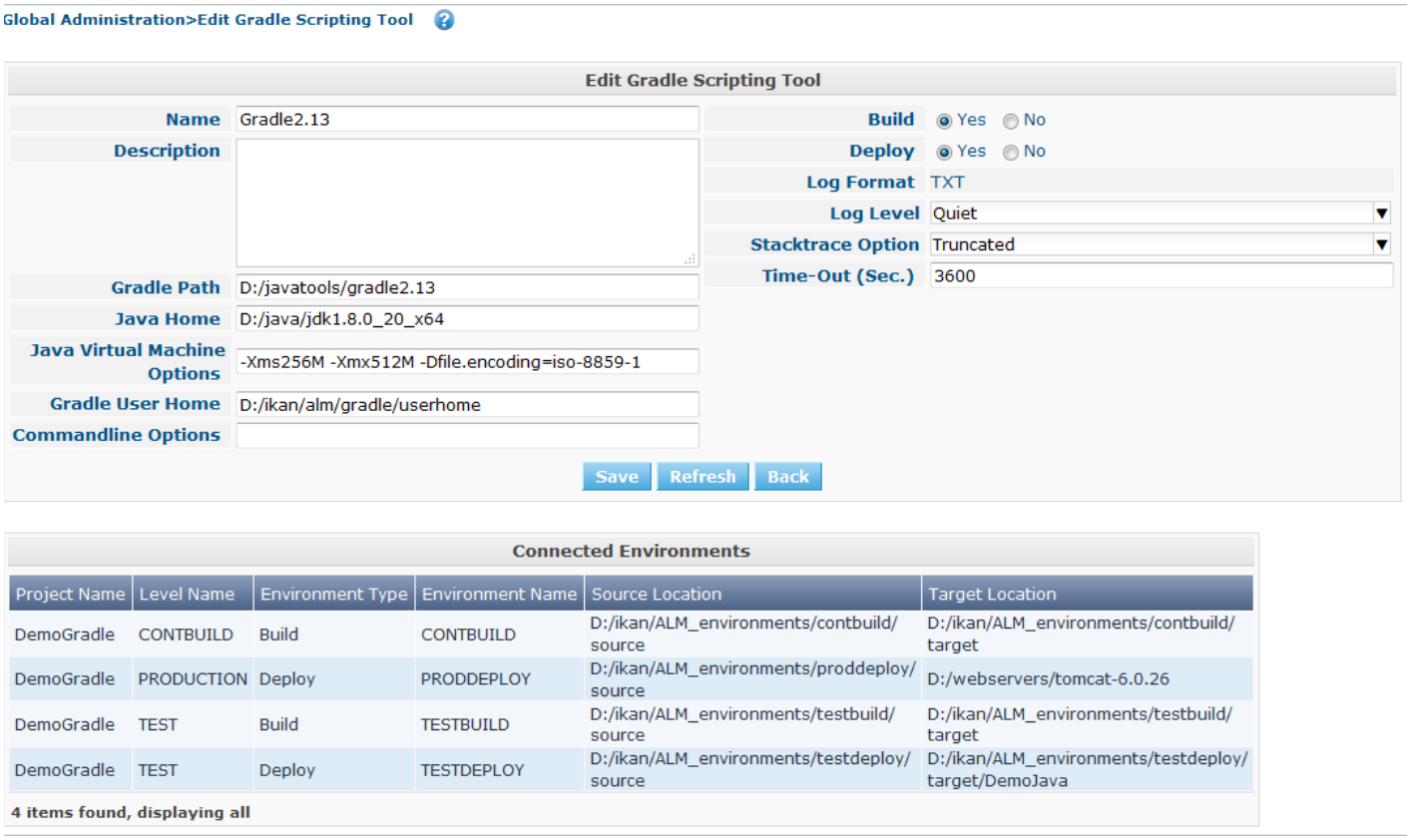
	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Scripting Tool definition.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Scripting Tool definition.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Scripting Tool

RELATED TOPICS
- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

Editing a Gradle Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the Gradle Scripting Tools your are looking for.

2. Click the *Edit* link to change the selected Gradle Scripting Tool.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a Gradle Scripting Tool (page 404).

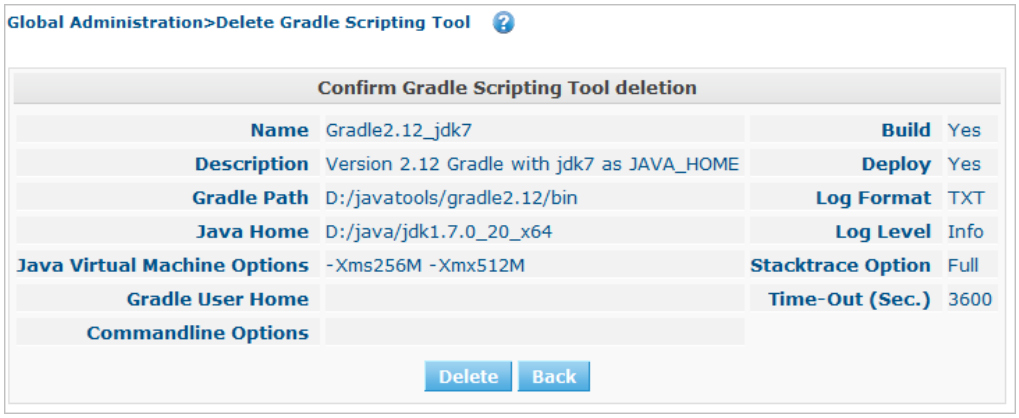
   Note: The *Connected Environments* panel displays the Environments the Scripting Tool is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

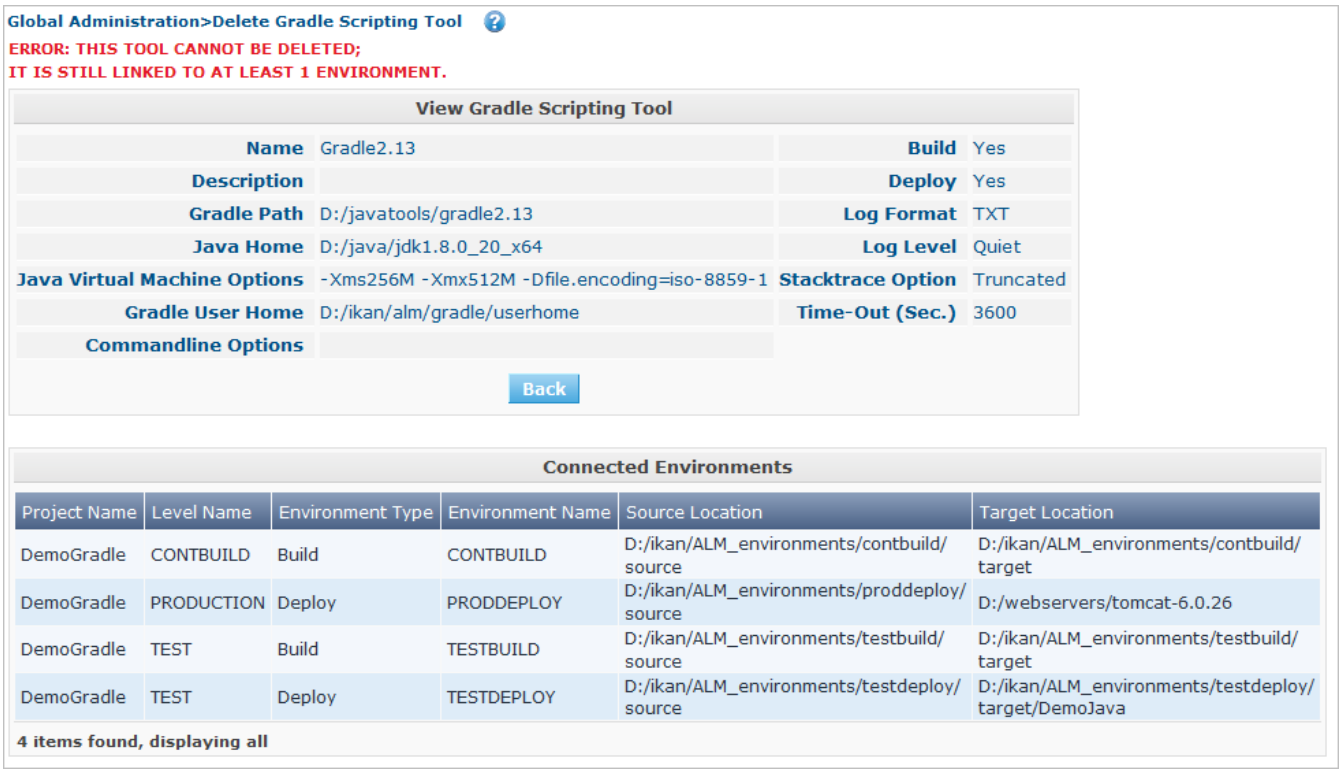
Deleting a Gradle Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the Gradle Scripting Tools your are looking for.

2. Click the *Delete* link to delete the selected Gradle Scripting Tool.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the Gradle Scripting Tool is linked to one or more Build or Deploy Environments, the following screen is displayed:



   You need to assign another Scripting Tool to these Environments, before deleting this Gradle Scripting Tool.

Viewing the Gradle Scripting Tool History

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the Gradle Scripting Tools your are looking for.

2. Click the *History* link to display the *Gradle Scripting Tool History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Scripting Tools Overview* screen.

RELATED TOPICS
- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

31.5. NANT Scripting Tools

NANT is a free .NET build tool. In theory it resembles to *make* without *make*'s wrinkles. In practice it is a lot like Ant.

If NANT is installed on a machine connected to your Build/Deploy Environments, IKAN ALM will be able to interact with it.

Therefore, you must define the NANT Scripting Tool in the IKAN ALM application and, when creating the Build or Deploy Environment, specify it as being the Build or Deploy Tool to be used.

Refer to the following sections for detailed information:

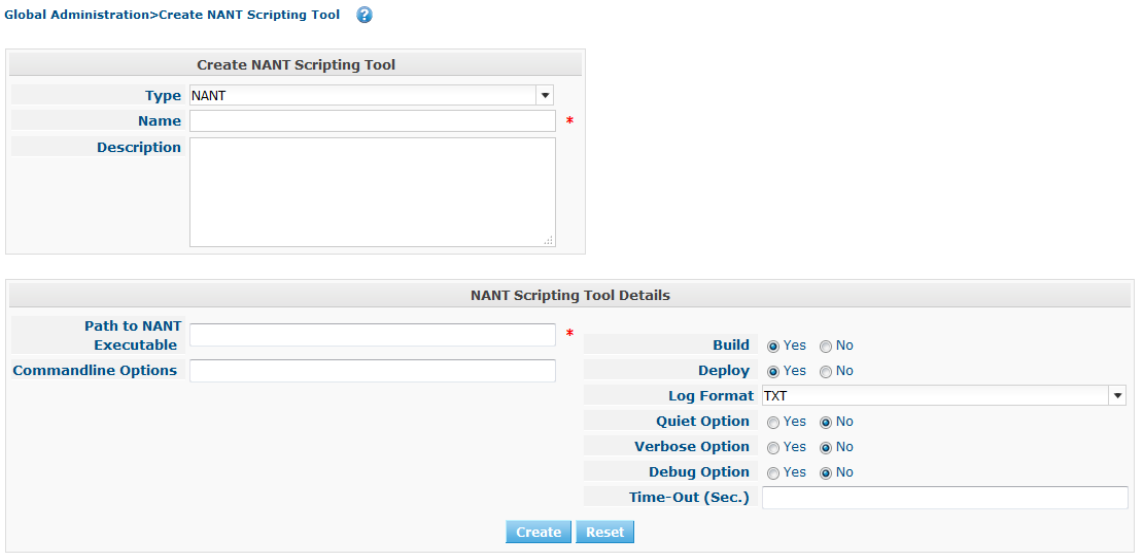
- Creating a NANT Scripting Tool (page 411)
- The NANT Scripting Tools Overview Screen (page 413)
- Editing a NANT Scripting Tool Definition (page 414)
- Deleting a NANT Scripting Tool Definition (page 415)
- Viewing the NANT Scripting Tool History (page 416)

Creating a NANT Scripting Tool

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Scripting Tools* > *Create*.

2. Select *NANT* from the drop-down list in the *Type* field on the search panel.
   The following screen is displayed:



3. Fill out the fields in the *Create NANT Scripting Tool* panel at the top of the screen.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Type	Select the type of Scripting Tool you want to define. This field is mandatory. After you have selected the type, the appropriate *Details* panel will be displayed underneath.
Name	Enter the name of the new NANT Scripting Tool in this field. This field is mandatory.
Description	Enter a description for the new NANT Scripting Tool in this field. This field is optional.

4. Fill out the fields in the *NANT Scripting Tool Details* panel.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Path to NANT Executable	Enter the path to the NANT executable (NAnt.exe) in this field. This field is mandatory.
Commandline Options	Enter commandline options in this fields. This field is optional. The commandline options provided by default by IKAN ALM, like -buildfile, -logfiles, -verbose, -debug ,… will not be accepted. Sample options are -keep-going, or -noinput.
Build	Indicate whether or not this Scripting Tool can be linked to a Build Environment and thus be used to build code.
Deploy	Indicate whether or not this Scripting Tool can be linked to a Deploy Environment and thus be used to deploy Builds.
Log Format	Select the format of the log file generated by this NANT Scripting Tool. The following options are available: • TXT: the log file will be a simple text file • XML: the log file will be an XML file TXT logs will be smaller and their content can be displayed while the Tool is executing. The display of XML logs can be customized by providing custom XSL templates. However, XML logs are larger and can only be displayed after the Tool has finished executing.
Quiet Option	Select whether or not the Quiet Option must be activated for the new NANT Scripting Tool. NANT will print less information than normal during the Build and Deploy actions, if this option is activated.
Verbose Option	Select whether or not the Verbose Option must be activated for the new NANT Scripting Tool. NANT will print verbose debug information during the Build and Deploy actions, if this option is activated.
Debug Option	Select whether or not the Debug Option must be activated for the new NANT Scripting Tool. NANT will print additional debug information during the Build and Deploy actions, if this option is activated.
Time-Out (Sec.)	In this field, enter a Time-Out value if required. If a value is provided, a running NANT Build or Deploy process will be interrupted after this number of seconds. In this way, "hanging" Build or Deploy processes are interrupted. If no value is provided, a running NANT Build or Deploy process will never be interrupted.

5. Once you filled out the fields as required, click *Create*.

The newly created NANT Scripting Tool is added to the *NANT Scripting Tools Overview* at the bottom of the screen.

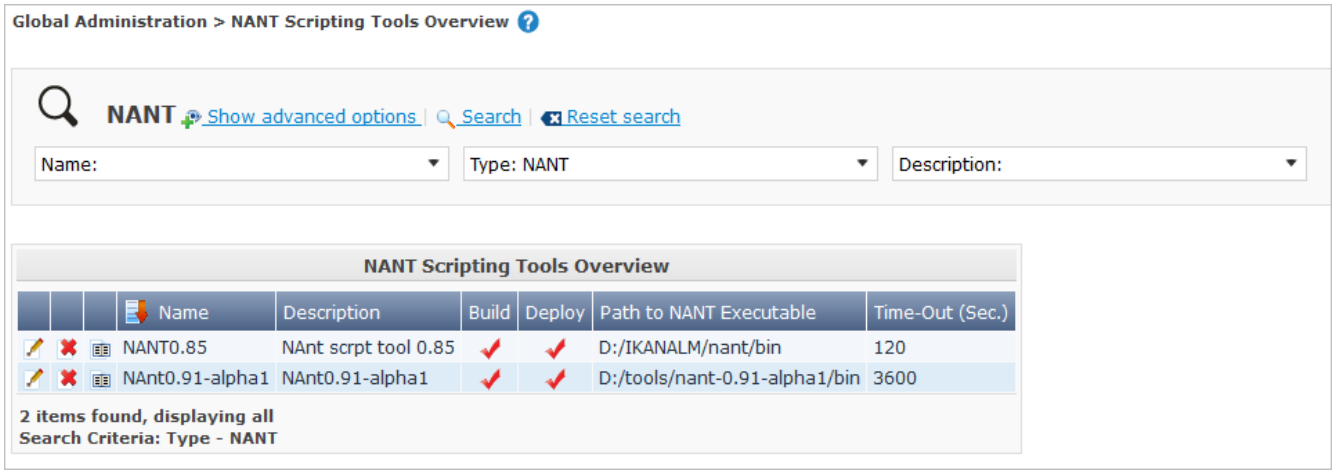
RELATED TOPICS

- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

The NANT Scripting Tools Overview Screen

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.

2. Specify *NANT* in the *Type* field on the search panel.



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *Scripting Tools Overview* panel.
   For a detailed description of the fields, refer to The NANT Scripting Tools Overview Screen (page 413).

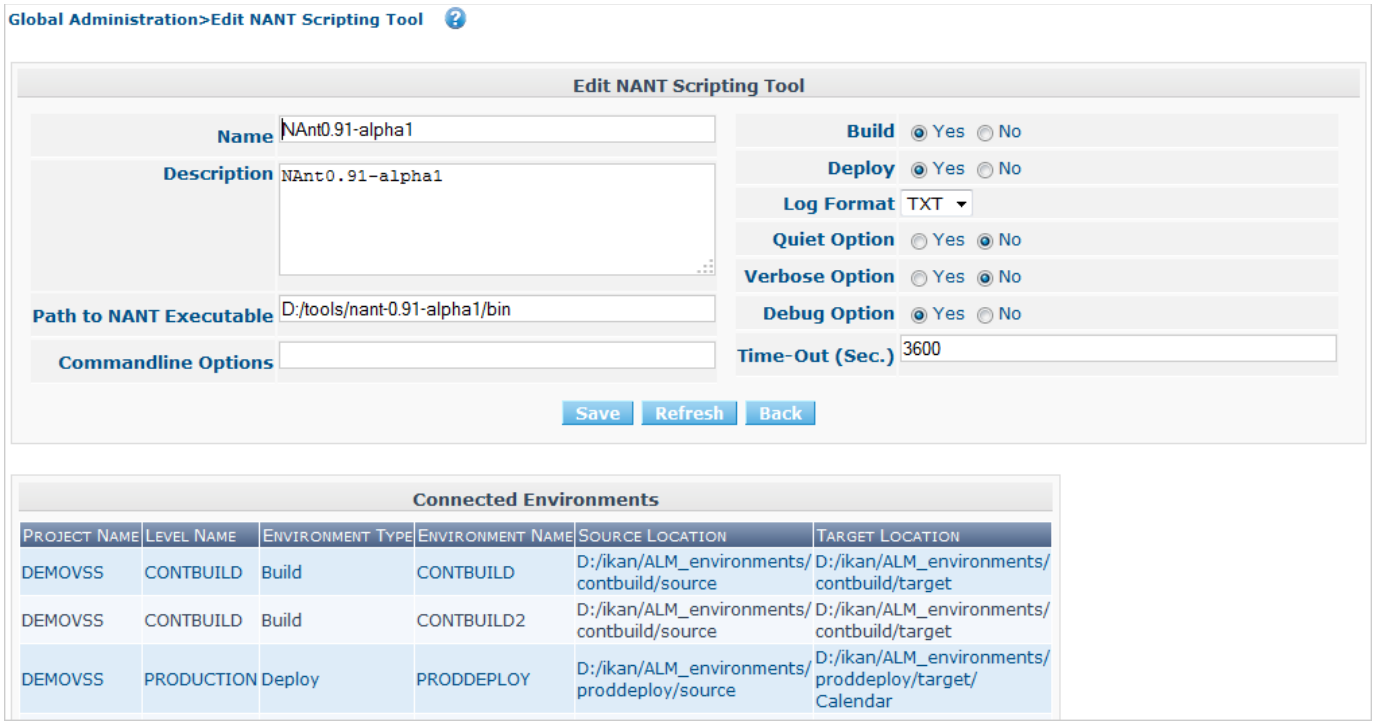
4. Depending on your access rights, the following links may be available on the *NANT Scripting Tools Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Scripting Tool definition.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Scripting Tool definition.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Scripting Tool

Editing a NANT Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the NANT Scripting Tools your are looking for.

2. Click the *Edit* link to change the selected NANT Scripting Tool.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a NANT Scripting Tool (page 411).

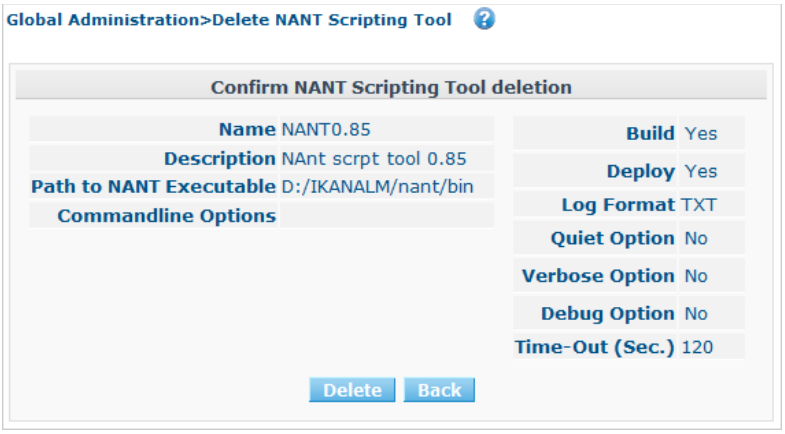
   Note: The *Connected Environments* panel displays the Environments the Scripting Tool is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

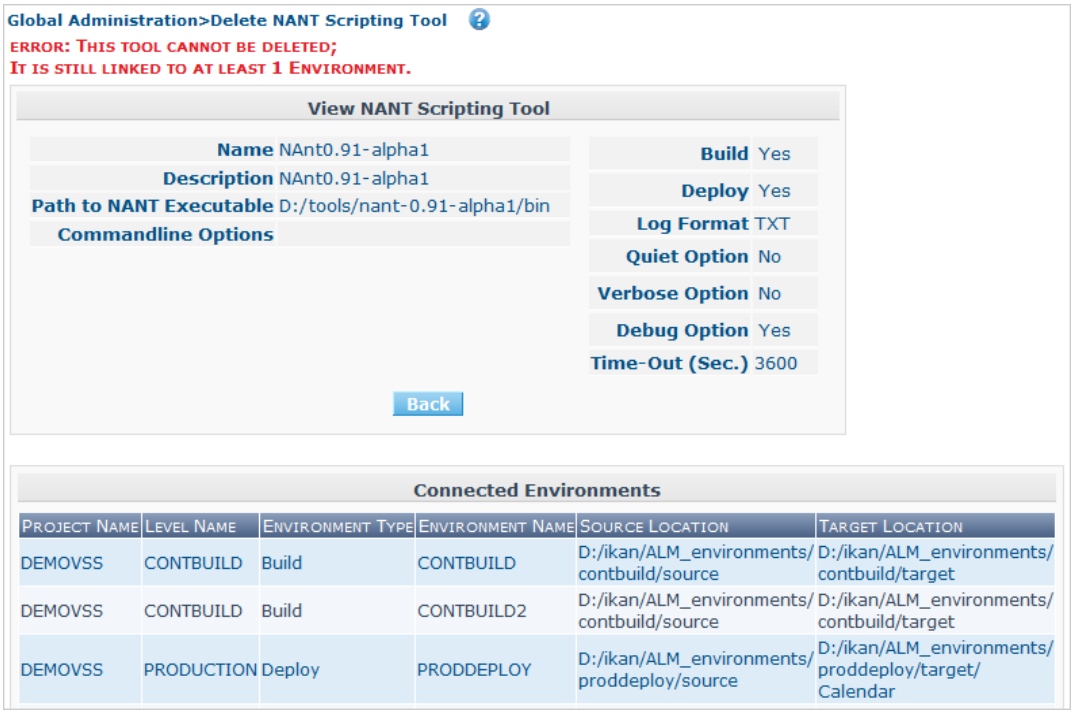
Deleting a NANT Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the NANT Scripting Tools your are looking for.

2. Click the *Delete* link to delete the selected NANT Scripting Tool.
   The following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the NANT Scripting Tool is linked to one or more Build or Deploy Environments, the following screen is displayed:



You need to assign another Scripting Tool to these Environments, before deleting this NANT Scripting Tool.

Viewing the NANT Scripting Tool History

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the NANT Scripting Tools your are looking for.

2. Click the *History* link to display the *NANT Scripting Tool History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Scripting Tools Overview* screen.

RELATED TOPICS
- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

31.6. Maven2 Scripting Tools

Maven2 is a scripting tool that can be used for building and managing any Java-based project. Its primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. If Maven2 is installed on a machine connected to your Build/Deploy Environments, IKAN ALM will be able to interact with Maven2.

Therefore, you must define the Maven2 Scripting Tool in the IKAN ALM application and, when creating the Build or Deploy Environment, specify it as being the Build or Deploy Tool to be used.

Refer to the following sections for detailed information:

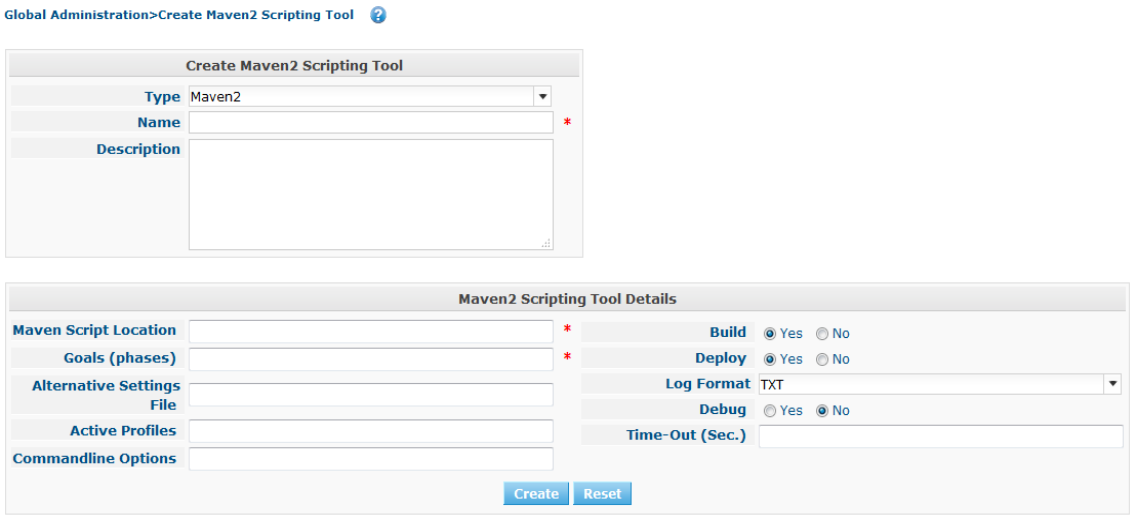
- Creating a Maven2 Scripting Tool (page 417)
- The Maven2 Scripting Tools Overview Screen (page 420)
- Editing a Maven2 Scripting Tool Definition (page 421)
- Deleting a Maven2 Scripting Tool Definition (page 422)
- Viewing the Maven2 Scripting Tool History (page 423)

Creating a Maven2 Scripting Tool

Note: This option is only available if you have Global Admin Access Rights in IKAN ALM.

1. In the Global Administration context, select *Scripting Tools* > *Create*.

2. Select *Maven2* from the drop-down list in the *Type* field in the search panel.
   The following screen is displayed:



3. Fill out the fields in the *Create Maven2 Scripting Tool* panel at the top of the screen.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Type	Select the type of Scripting Tool you want to define. This field is mandatory. After you have selected the type, the appropriate *Details* panel will be displayed underneath.
Name	Enter the name of the new Maven2 Scripting Tool in this field. This field is mandatory.
Description	Enter a description for the new Maven2 Scripting Tool in this field. This field is optional.

4. Fill out the fields in the *Maven2 Scripting Tool Details* panel.
   Fields marked with a red asterisk are mandatory.

Field	Meaning
Maven Script Location	Enter the path to the Maven2 start-up script on the target Machine linked to the Build/Deploy Environment (mvn shell script or mvn2.bat file). This field is mandatory.
Goals (phases)	Enter the Maven2 goals and/or phases to be executed by default. This field is mandatory. The goals and/or phases should be separated by a space and use the following format: `[<goal(s)] [<Phase(s)]`. For example, `clean dependency:copydependencies test` The goals and/or phases defined here will be used by default. If necessary, they can be overridden for the different build or deploy environments. In order to do so, add a build or deploy parameter "alm.mvn2.goals" with the desired value. Please refer to Creating Environment Parameters (page 242).
Alternative Settings File	Enter the alternate path for the Maven2 User Settings File. It corresponds with the -s or --settings commandline option. This field is optional. If not provided, the default Settings File under home-directory/.m2/settings.xml will be taken into account. If necessary, this file can be overridden for the different build or deploy environments. In order to do so, add a build or deploy parameter "alm.mvn2.setting" with the desired value. Please refer to Creating Environment Parameters (page 242).
Active Profiles	Enter the Activate Profiles. This is a comma-delimited list of profiles to activate. It corresponds with the -P or --activate-profiles Maven2 commandline option. This field is optional. The Activate Profiles defined here will be used by default. If necessary, they can be overridden for the different build or deploy environments. In order to do so, add a build or deploy parameter "alm.mvn2.activate-profiles" with the desired value. Please refer to Creating Environment Parameters (page 242).
Commandline Options	Enter the Commandline Options, separated by a space. For example: the option "-e" will give you more information about error messages. This field is optional. Please note that you can NOT use the following options as they are already used by IKAN ALM: "-X" or "--debug" (Debug), "-s" or "--settings" (Settings File), "-P" or "--activate-profiles" (Activate Profiles) and "-B" or "--batch-mode". The Commandline Options defined here will be used by default. If necessary, they can be overridden for the different build or deploy environments. In order to do so, add a build or deploy parameter "alm.mvn2.options" with the desired value. Please refer to Creating Environment Parameters (page 242).
Build	Indicate whether or not this Scripting Tool can be linked to a Build Environment and thus be used to build code.
Deploy	Indicate whether or not this Scripting Tool can be linked to a Deploy Environment and thus be used to deploy Builds.

Field	Meaning
Log Format	Select the format of the log file generated by this Maven2 Scripting Tool. The following options are available: • TXT: the log file will be a simple text file • XML: the log file will be an XML file TXT logs will be smaller and their content can be displayed while the Tool is executing. The display of XML logs can be customized by providing custom XSL templates. However, XML logs are larger and can only be displayed after the Tool has finished executing.
Debug	Select whether or not the Debug Option must be activated for the new Maven2 Scripting Tool. This corresponds with the -X or --debug Maven2 commandline option. If this option is activated, Maven2 will print additional debug information during the Build and Deploy actions.
Time-Out (Sec.)	In this field, enter a Time-Out value if required. If a value is provided, a running Maven2 Build or Deploy process will be interrupted after the defined number of seconds. In this way, "hanging" Build or Deploy processes are interrupted. If no value is provided, a running Maven2 Build or Deploy process will never be interrupted.

5. Once you filled out the fields as required, click *Create*.

   The newly created Maven2 Scripting Tool is added to the *Maven2 Scripting Tools Overview* at the bottom of the screen.

   You can also click *Reset* to empty the fields and restore the initial values.

6. Overriding the default parameters set in the Maven2 Scripting Tool

   You can override the default parameters set in the Maven2 Scripting Tool.

   If an "alm.mvn2.goals" parameter for a chosen environment (either build or deploy) is specified, it will be used instead of the default goals that were initially set in the Maven2 Scripting Tool.

   The same mechanism applies to the following parameters: Settings File (alm.mvn2.setting), Activate Profiles (alm.mvn2.activate-profiles) and Commandline Options (alm.mvn2.options).

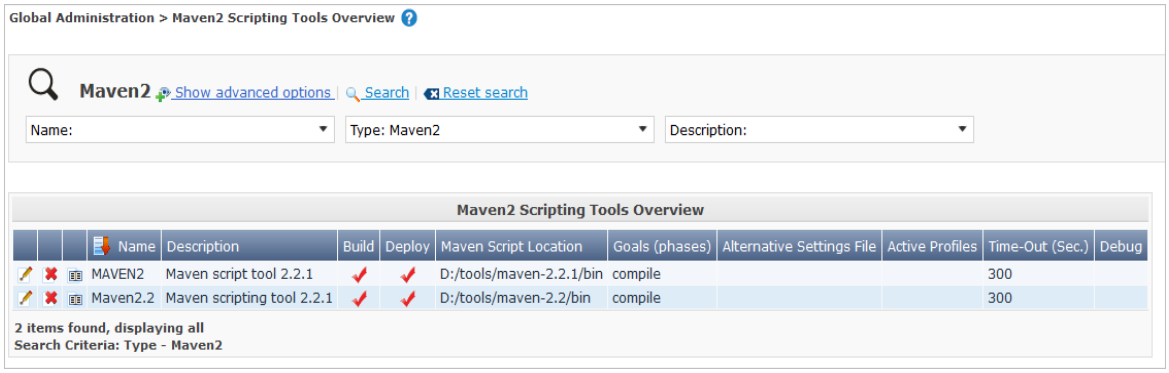
   When creating a build or deploy parameter, you can also set the option "*Editable*" to *Yes*. In that case you will be able to modify the value of the parameters at the moment of creating the level request. See Editing Environment Parameters on page 244.

RELATED TOPICS
- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

The Maven2 Scripting Tools Overview Screen

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.

2. Specify *Maven2* in the *Type* field on the search panel.



   If required, use the other search criteria to refine the items displayed on the overview.
   The following options are available:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - the *Search* link to refresh the list based on the current search criteria,
   - the *Reset search* link to clear the search fields.

3. Verify the information on the *Scripting Tools Overview* panel.
   For a detailed description of the fields, refer to The Maven2 Scripting Tools Overview Screen (page 420).

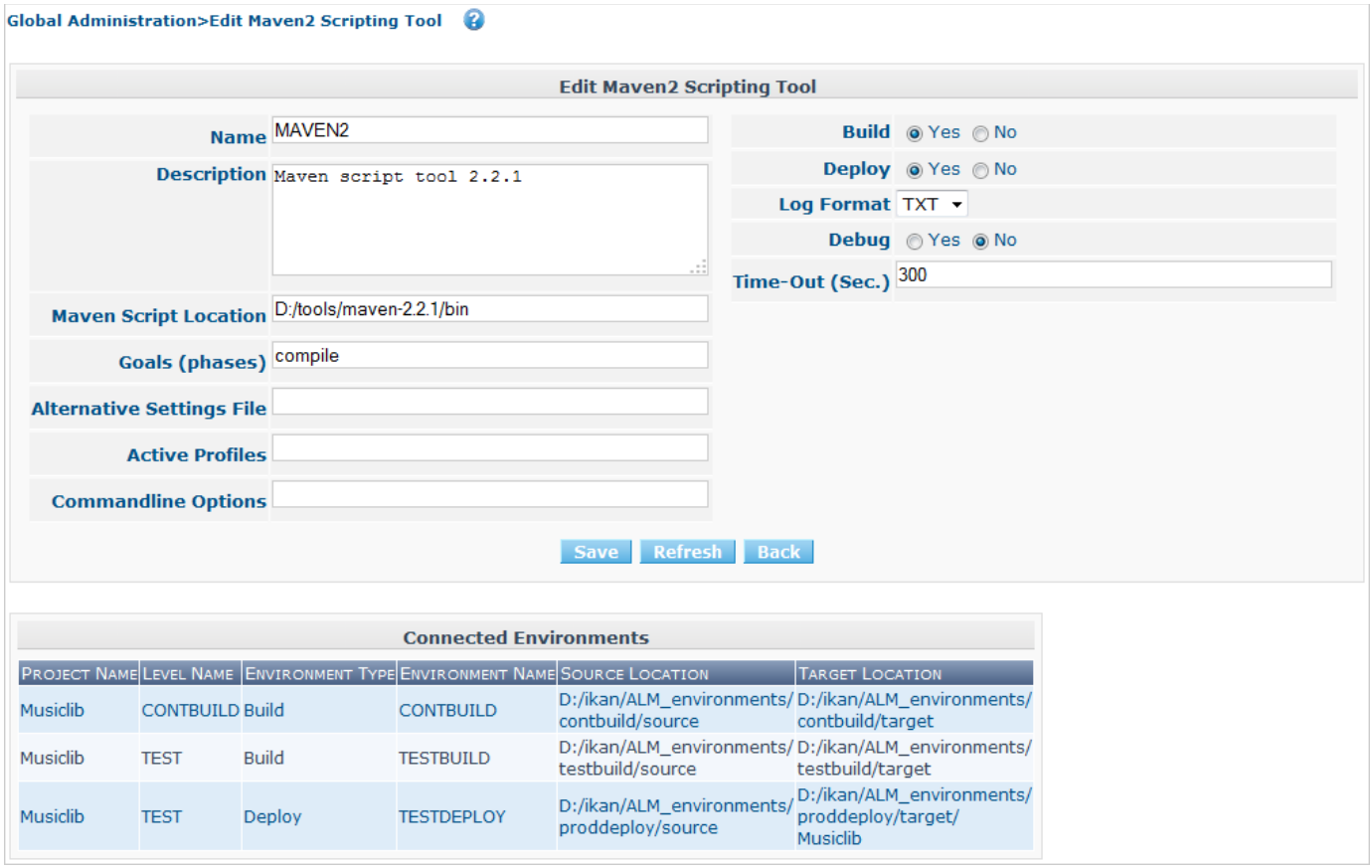
4. Depending on your access rights, the following links may be available on the *Maven2 Scripting Tools Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Scripting Tool definition.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Scripting Tool definition.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Scripting Tool

Editing a Maven2 Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the Maven2 Scripting Tools your are looking for.

2. Click the *Edit* link to change the selected Maven2 Scripting Tool.
   The following screen is displayed.



3. Edit the fields as required.
   For a description of the fields, refer to Creating a Maven2 Scripting Tool (page 417).

   Note: The *Connected Environments* panel displays the Environments the Scripting Tool is linked to.

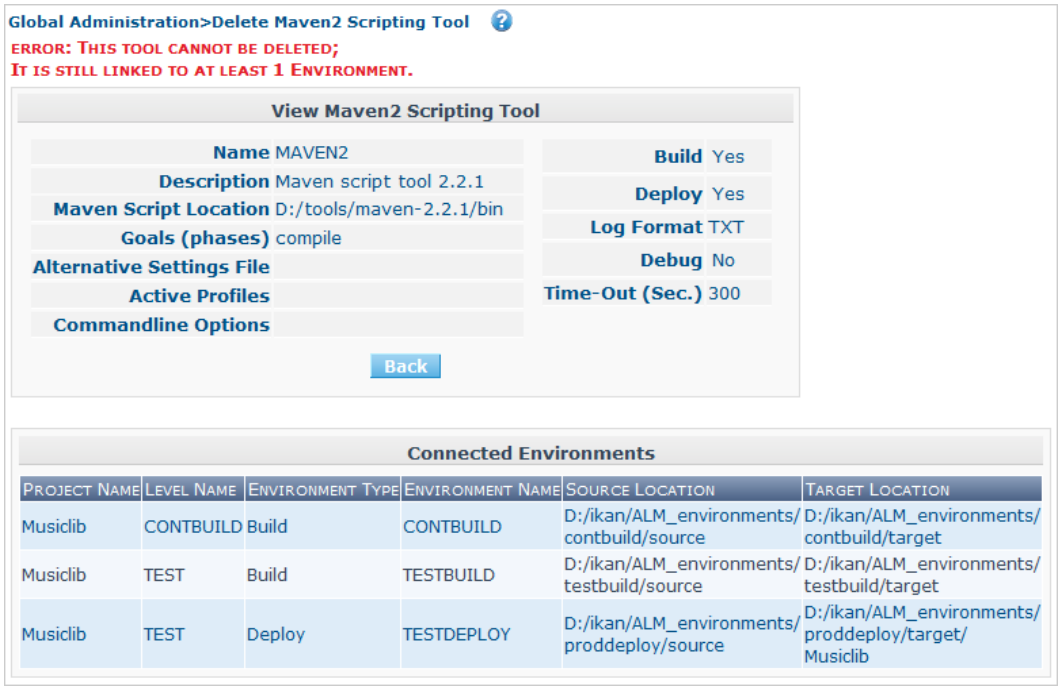
4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

Deleting a Maven2 Scripting Tool Definition

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the Maven2 Scripting Tools your are looking for.

2. Click the *Delete* link to delete the selected Maven2 Scripting Tool.
   The following screen is displayed:

3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the Maven2 Scripting Tool is linked to one or more Build or Deploy Environments, the following screen is displayed:



   You need to assign another Scripting Tool to these Environments, before deleting this Maven2 Scripting Tool.

Viewing the Maven2 Scripting Tool History

1. In the Global Administration context, select *Scripting Tools > Overview*.
   The overview of all defined Scripting Tools is displayed.
   Use the search criteria on the search panel to display the Maven2 Scripting Tools your are looking for.

2. Click the *History* link to display the *Maven2 Scripting Tool History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Scripting Tools Overview* screen.

RELATED TOPICS
- Scripting Tools (page 394)
- Editing Project Settings (page 149)
- Build Environments (page 209)
- Deploy Environments (page 225)

CHAPTER 32
Phases

When IKAN ALM is running Level Requests, Builds and Deploys, all actions are performed by executing a sequence of Phases. Those Phases are defined in the IKAN ALM database and can be consulted and manipulated in the *Phases* section of the *Global Administration* interface. Once they have been defined in *Global Administration*, Phases may be linked to Levels, Build or Deploy Environments in the *Project Administration* context.

The IKAN ALM core functionality is performed by so-called "Core" Phases. Those Core Phases can only be viewed, and cannot be altered nor deleted. Consider them an integral part of IKAN ALM.

You can extend this core functionality by creating your own Phases. There are two options:

1. Create a Phase from scratch using the "Create Phase" functionality. In that case, you first specify the name and the version of the Phase, and, next, you choose the script or scripts to be executed by the Phase.
2. Import a Phase that has already been created via the "Import Phase" functionality.

Refer to the following sections for detailed information:

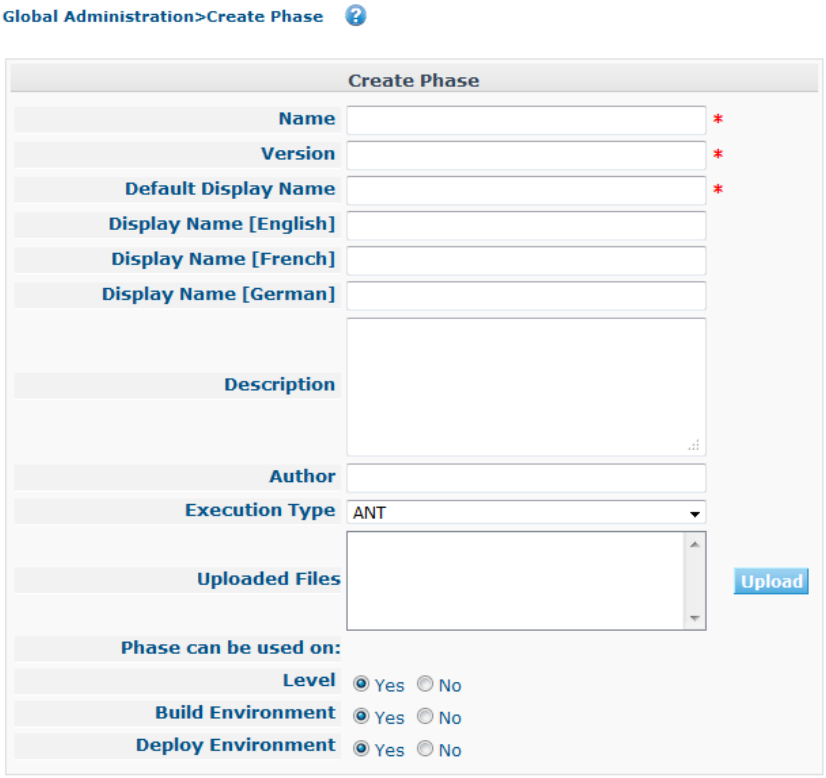
- Creating a Phase Definition (page 425)
- The Phases Overview Screen (page 428)
- Editing a Phase Definition (page 429)
- The Phase Parameters Overview Screen (page 434)
- Creating Phase Parameters (page 435)
- Editing Phase Parameters (page 437)
- Deleting Phase Parameters (page 440)
- Mass Editing Phase Parameters (page 440)
- Mass Replacing Phases (page 442)
- Mass Deleting Phases (page 445)
- Deleting a Phase Definition (page 447)
- Exporting a Phase Definition (page 447)
- Copying a Phase Definition (page 448)
- Viewing the Phase History (page 449)
- Importing a Phase Definition (page 450)

For a complete explanation of the concept of Phases, refer to Phases - General Information (page 520).

32.1. Creating a Phase Definition

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Phases* > *Create*.
   The following screen is displayed.



2. Fill out the fields in the *Create Phase* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	The name of the Phase. This must be a unique name so choose it carefully. A good practice is to use a "reverse DNS" notation, like for example: "com.mycompany.phases.ant.deployToTomcat". The name may contain the following characters: • letters • numbers • ._- All other characters (including whitespaces) are illegal and will result in a validation error. This field is mandatory and cannot be changed after the creation of the Phase.
Version	The version of the Phase. Together with the Phase name, this field forms the unique identifier of the Phase. As a result, there can only be one Phase defined with a certain name and version. The version needs to conform to the following format: `[0-9]+(\.[0-9]+(\.[0-9]+(\.[0-9A-Za-z_-]+)?)?)?` For example, the following versions are legal: • 1.0 • 1.0.0 • 1.0.0.0 • 1.0.0.0-beta2 The following versions are NOT legal: • 1. • beta2 • 1.0.0-beta2
Default Display Name	The name of the Phase as it will be displayed in the IKAN ALM user interface, e.g., on the *Level Request Detail* screen or on the *Build Environment Phases Overview* screen. This default name will be used when no language-specific display name is provided.This field is mandatory.
Display Name [English]	The English name of the Phase as it will be displayed in the IKAN ALM user interface for a User whose language is set to English in his or her *Personal Settings*. See Personal Settings on page 15. This field is optional.
Display Name [French]	The French name of the Phase as it will be displayed in the IKAN ALM user interface for a User whose language is set to French in his or her *Personal Settings*. See Personal Settings on page 15. This field is optional.
Display Name [German]	The German name of the Phase as it will be displayed in the IKAN ALM interface for a User whose language is set to German in his or her *Personal Settings*. See Personal Settings on page 15. This field is optional.
Description	The description for the new Phase. This field is optional.

Field	Meaning
Author	The author of the Phase. For example, the name of the User creating the Phase or the company he is working for. This field can be useful when searching for Phases. This field is optional.
Execution Type	Select the Scripting Tool type that will be used to execute the scripts that are contained in this Phase. The possible values are: • ANT • GRADLE • NANT • MAVEN2 This field is mandatory and cannot be changed after the creation of the Phase.

3. Once you have filled out the above mentioned fields, you need to upload the script file(s).

   Click the *Upload* button.

   A file selection window will open.

4. Select the scripts that will be used to execute the Phase.

   Note: You can select only one file.

   - If there is only one script file to be uploaded, you simply select that script file.
   - If the Phase needs multiple files for its execution, you must first archive those files into a .zip or a .tar.gz file, and then select that file. IKAN ALM will extract the archive file, and show its contents in the *Uploaded Files* field.

   When the upload has succeeded, the following message is displayed:

5. Select the "main" script.

   As the message suggests, you must now select the "main" script in the list of uploaded files. This is the script that will actually be called when the Phase is executed.

   When the upload has succeeded and the main script is selected, the *Create* and *Reset* buttons will become available.

6. Indicate where the Phase can be used.

   Select the appropriate option(s).

   There are three possibilities:

   - On Levels
   - On Build Environments
   - On Deploy Environments

7. Click *Create* to create the Phase.

When clicking the *Create* button, the Phase is created in the Phase Catalog and added to the *Phases Overview* panel.

Note: The location of the Phase Catalog is specified in the System Settings. See System Settings on page 255.

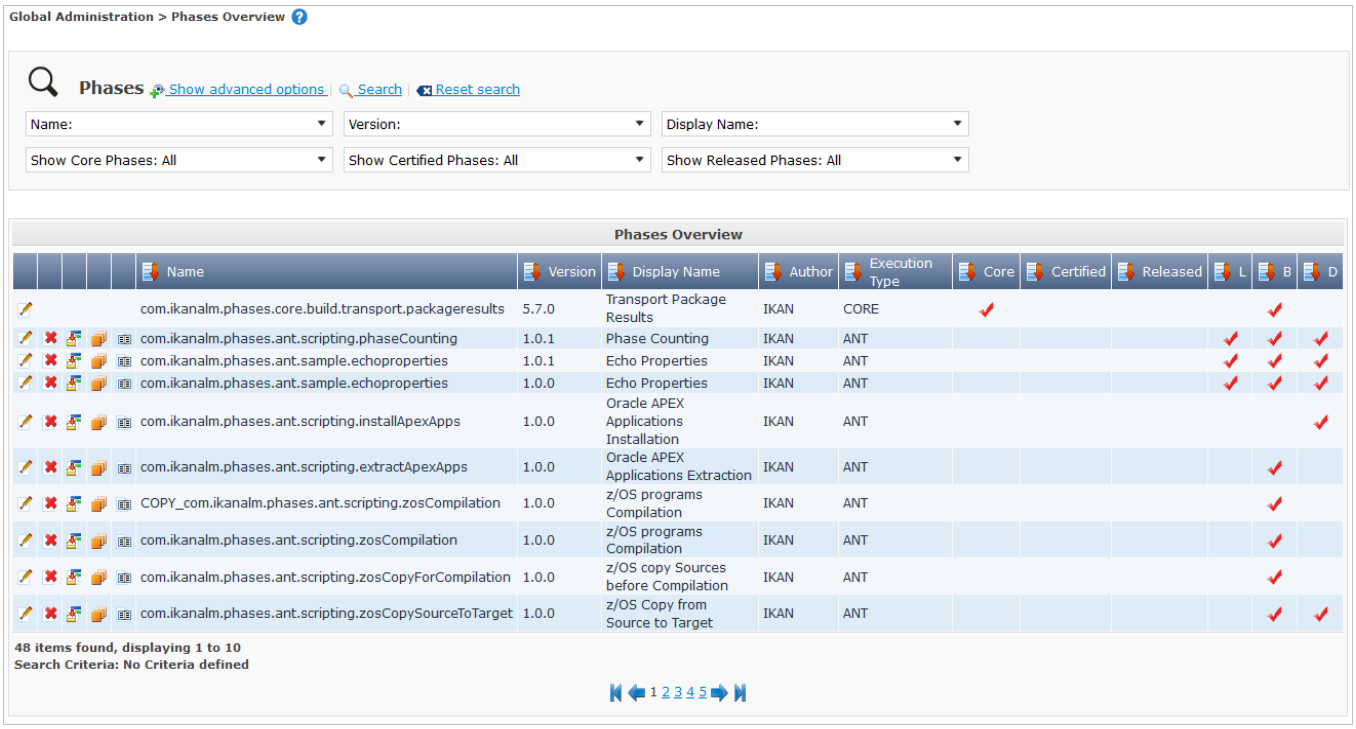
You can also click *Reset* to clear the fields and restore its initial values.

RELATED TOPICS
- Level Phases (page 195)
- Inserting a Level Phase (page 201)
- Build Environment Phases (page 215)
- Deploy Environment Phases (page 231)
- Phase Catalog settings. See System Settings on page 255.

32.2. The Phases Overview Screen

1. In the Global Administration context, select *Phases > Overview*.

The following screen is displayed:



2. Define the required search criteria on the search panel.

The list of items on the overview will be automatically updated based on the selected criteria.

You can also:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- click the *Search* link to refresh the list based on the current search criteria,
- click the *Reset search* link to clear the search fields.

3. Verify the information on the *Phases Overview* panel.

   For a detailed description of the fields, refer to Creating a Phase Definition (page 425)and Editing a Phase Definition (page 429).

4. Depending on your access rights, the following links may be available on the *Phases Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Phase. See Editing a Phase Definition on page 429.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Phase. See Deleting a Phase Definition on page 447.
	Export This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows exporting a Phase. See Exporting a Phase Definition on page 447.
	Copy This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows copying a Phase. See Copying a Phase Definition on page 448.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Phase. See Viewing the Phase History on page 449.

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

32.3. Editing a Phase Definition

The *Phase Info* panel lets you edit the definition of a Phase.

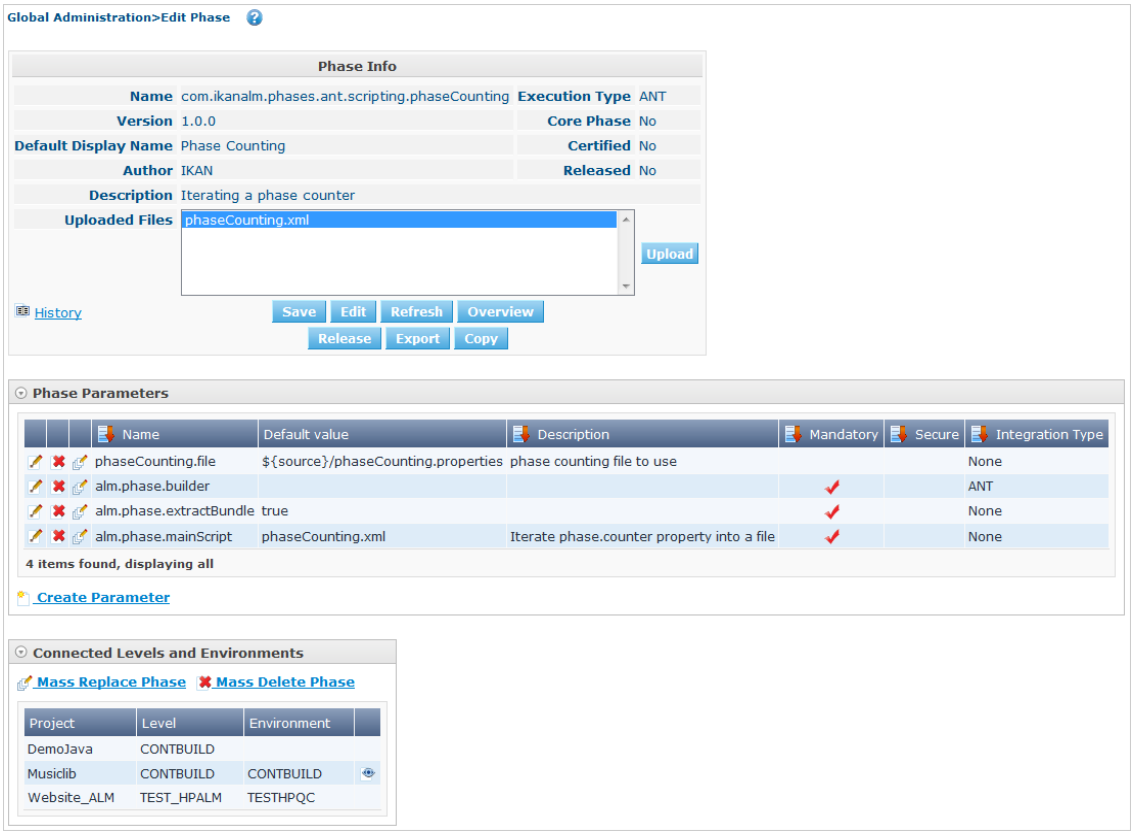
Underneath this panel, the *Phase Parameters* panel is displayed allowing you to create, edit, delete and mass edit Phase Parameters.

For more detailed information on Phase Parameters, refer to the following sections:

- The Phase Parameters Overview Screen (page 434)
- Creating Phase Parameters (page 435)
- Editing Phase Parameters (page 437)
- Deleting Phase Parameters (page 440)
- Mass Editing Phase Parameters (page 440)

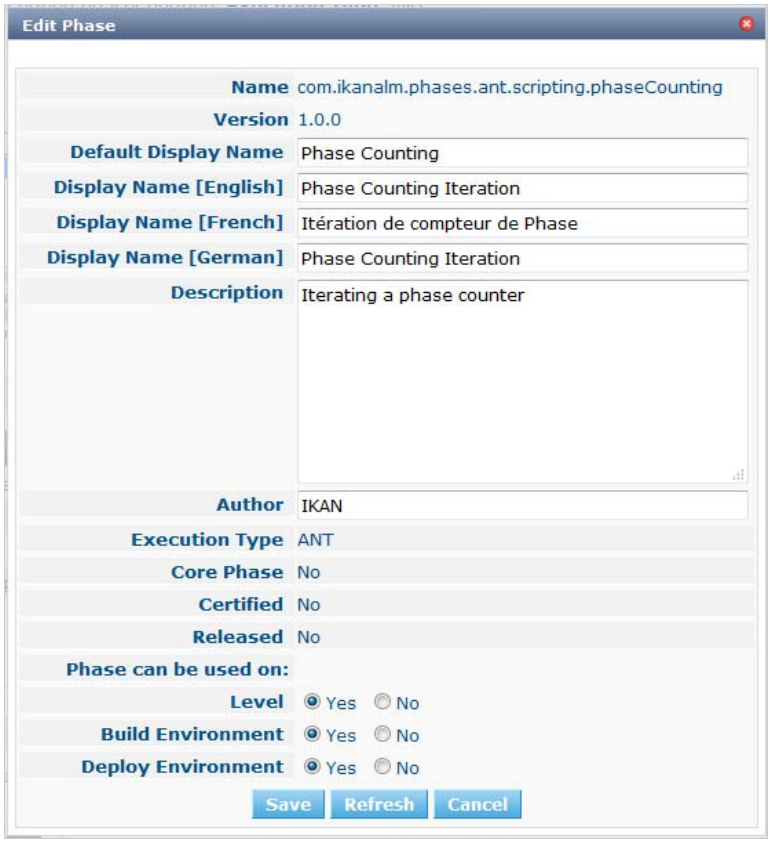
1. In the Global Administration context, select *Phases* > *Overview*.

2. On the *Phases Overview* panel, click the *Edit* link in front of the Phase you want to edit. The following screen is displayed:



3. Click the *Edit* button to modify the Phase.

The *Edit Phase* pop-up window is displayed.



For a description of the fields, refer to Creating a Phase Definition (page 425).

The following additional fields are displayed on this screen:

Field	Meaning
Core Phase	This field indicates whether a Phase is a Core Phase or not. A Core Phase is an internal IKAN ALM Phase that performs some core functionality (e.g., the *Retrieve Code* Phase). It cannot be edited nor deleted. For more information, refer to Phases - General Information (page 520).
Certified	This field indicates whether a Phase is Certified or not. A Certified Phase is a Phase that has been tested and approved by IKAN. It cannot be modified and its parameters cannot be deleted. For more information, refer to Phases - General Information (page 520).
Released	This field indicates whether a Phase has been Released or not. A Phase that has not been released is regarded as being in development, i.e., its script(s) and other containing files may be changed. To facilitate Phase development, IKAN ALM will automatically re-install a non-released Phase just before it is executed. Once a Phase has been released, its contents (scripts) cannot change anymore, so the *Upload* button will not be available. For more information, refer to Phases - General Information (page 520).

Note: The Name and Version fields are not editable. If you want to rename a Phase or change its version, you must first copy it, and then delete the original Phase.
For more information, refer to the section Copying a Phase Definition (page 448).

4. Verify the uploaded files

The *Uploaded Files* field lists the current contents of the Phase.

If you want to alter the contents, click the *Upload* button and choose a script file or an archive file. The new uploaded files will be shown in the *Uploaded Files* list.

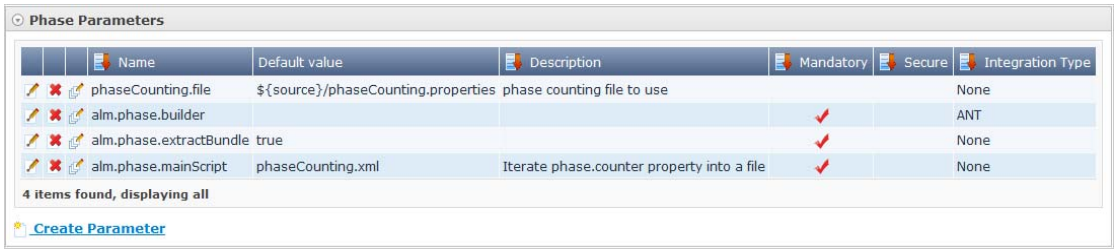
Note: The new uploaded files will **REPLACE** the old files; they are not added to the current contents of the Phase!

The new uploaded files will only be persisted when you click the *Save* button. To redisplay the originally uploaded files, click the *Refresh* button.

For more information on uploading files, refer to the section Creating a Phase Definition (page 425).

5. Verify the Phase parameters.

The *Phase Parameters* panel displays all the defined Parameters of the Phase.



For a detailed description of the fields, refer to the section Creating Phase Parameters (page 435).

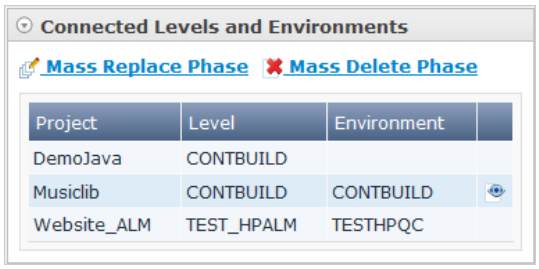
The following links are available on the *Phase Parameters* panel:

	Edit This option allows editing a Phase Parameter. See Editing Phase Parameters on page 437.
	Delete This option allows deleting a Phase Parameter. See Deleting Phase Parameters on page 440.
	Mass Edit This option allows editing the values of a Parameter in its connected Environments. See Mass Editing Phase Parameters on page 440.

You can also add a new parameter, by clicking the *Create Parameter* link underneath the *Phase Parameters* panel. For more information, refer to the section Creating Phase Parameters (page 435).

6. Verify the connected Environments.

The *Connected Levels and Environments* panel shows the Levels and Build or Deploy Environments where this Phase has been added.



Note: When the *Environment* field is empty, this means that the Phase has been added to the Level.

For more information on mass replacing and mass deleting Phases, refer also to the sections Mass Replacing Phases (page 442) and Mass Deleting Phases (page 445).

7. On the *Edit Phase* panel, click *Save* to save your changes.

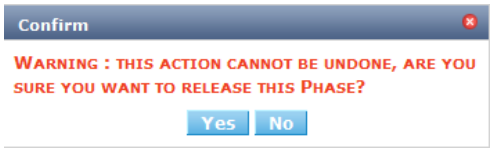
When clicking the *Save* button, the Phase's data are persisted and you will be redirected to the *Phases Overview* screen.

Meanwhile, IKAN ALM re-packages the uploaded files in a .jar file and replaces the existing .jar file in the Phase Catalog location (as defined in the System Settings (System Settings (page 255)) with the new .jar file. There, it is ready to be picked up by an IKAN ALM Server or Agent Daemon process when the Phase needs to be (re-)installed on an IKAN ALM Server or Agent.

You can also click:

- *Refresh* to retrieve the settings from the database.
- *Overview* to return to the previous screen without saving the changes.
- *Release* to release the Phase.

  When clicking the *Release* button, a confirmation pop-up window is displayed.



  Click *Yes* to confirm the release of the Phase.

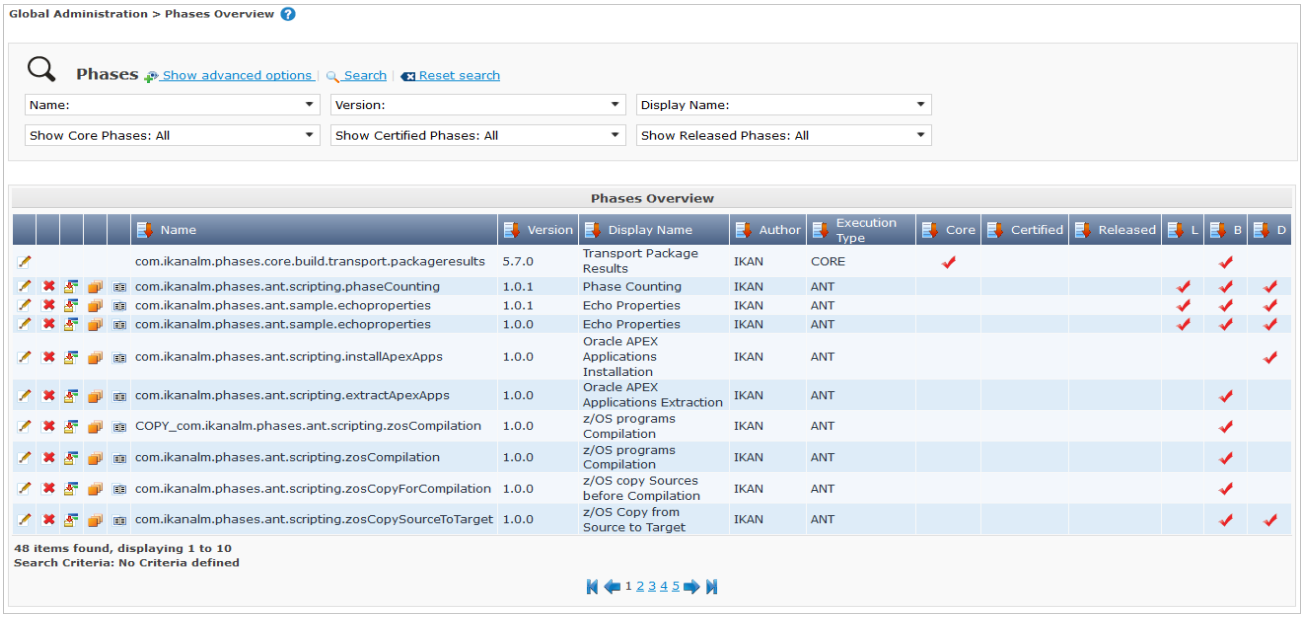
  As a result, the "Released" flag of the Phase will be set. Once a Phase has been released, its contents cannot be changed anymore, so the *Upload* button will no longer be available. The idea is that the behavior of the Phase is "frozen". Phase Parameters of a released Phase, however, can still be created, edited and deleted.

  The release of a Phase cannot be undone.
  If you need to upload new scripts for a Phase after it has been released, you must first copy the Phase and give the copy a different name and/or version, and then upload the new scripts for that new Phase. For more information, refer to the section Copying a Phase Definition (page 448).

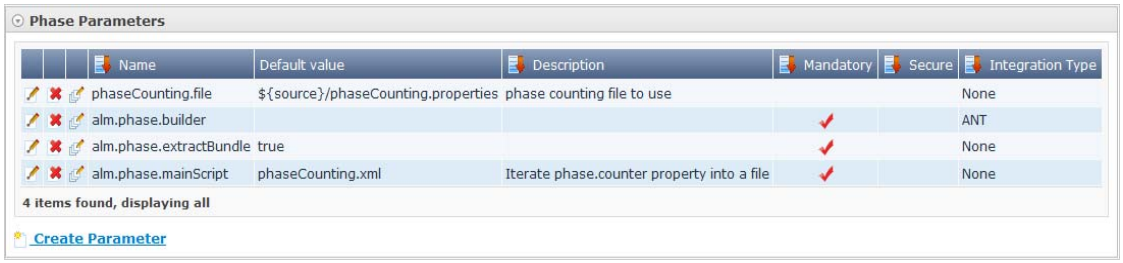
- *Export* to export the Phase. See Exporting a Phase Definition on page 447.
- *Copy* to copy the Phase. See Copying a Phase Definition on page 448.
- *History* to display the History of all create, update and delete operations performed on a Phase. See Viewing the Phase History on page 449.

32.4. The Phase Parameters Overview Screen

1. In the Global Administration context, select *Phases > Overview*.
   The following screen is displayed.



2. Click the *Edit* link in front of the required Phase on the *Phases Overview* panel.
   The *Edit Phase* screen is displayed.
   Underneath the *Phase Info* panel, the *Phase Parameters* panel displays all defined parameters.



3. Verify the information on the *Phase Parameters* panel.
   For a description of the fields, see Creating Phase Parameters (page 435).

The following links are available:

Link	Description
	Edit This option is available to all Users with Global Administrator Access Rights. It allows editing the selected Phase Parameter definition. See Editing Phase Parameters on page 437.
	Delete This option is available to all Users with Global Administrator Access Rights. It allows deleting the selected Phase Parameter definition. See Deleting Phase Parameters on page 440.
	Mass Edit This option is available to all Users with Global Administrator Access Rights. It allows mass editing the selected Phase Parameter. See Mass Editing Phase Parameters on page 440.

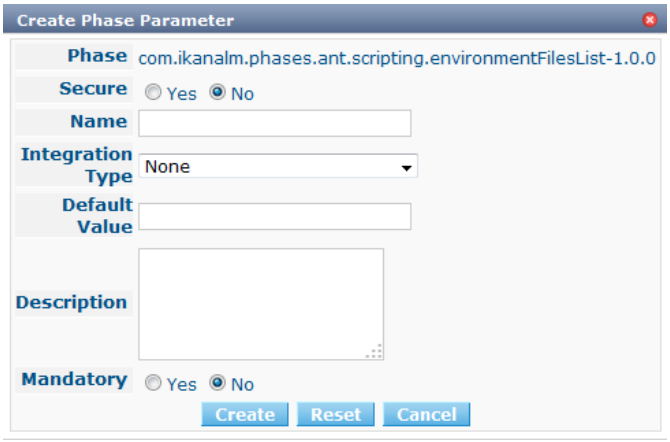
Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

RELATED TOPICS
- Creating Phase Parameters (page 435)
- Editing Phase Parameters (page 437)
- Deleting Phase Parameters (page 440)
- Mass Editing Phase Parameters (page 440)
- Viewing the Level Phase Parameters (page 198)
- Viewing the Build Environment Phase Parameters (page 220)
- Viewing the Deploy Environment Phase Parameters (page 235)

32.5. Creating Phase Parameters

1. In the Global Administration context, select *Phases > Overview*.
2. Click the *Edit* link in front of the required Phase on the *Phases Overview* panel.
   The *Edit Phase* screen is displayed.
3. Click the *Create Parameter* link at the bottom of the *Phase Parameters* panel.

The following pop-up window will be displayed:



4. Fill out the fields in the *Create Phase* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Phase	Name + version of the Phase the Parameter is being created for. This is a read-only field, displayed for informational purposes.
Secure	This field indicates whether the Parameter is secured or not. This field is mandatory and cannot be changed after the creation of the Parameter.
Name	The name of the Parameter. This field is mandatory.
Integration Type	This field indicates whether the value of the Parameter is a simple text value, or whether it represents a link (an integration) to an IKAN ALM object type. The possible values are: • None: the value is simple text • Transporter: link to a Transporter • VCR: link to a Version Control Repository • ITS: link to an Issue Tracking System • Scripting Tool: link to a Scripting Tool • ANT: link to an Ant Scripting Tool • GRADLE: link to a Gradle Scripting Tool • NANT: link to a NAnt Scripting Tool • MAVEN2: link to a Maven2 Scripting Tool When you select a type other than *None*, the *Default Value* field switches to a drop-down list where you can select a specific IKAN ALM object of that type. For example, if *ANT* is selected as Integration Type, the *Default Value* drop-down list will contain ANT Scripting Tools. This field is only relevant for non-secured Parameters. If the Parameter is set to secured, this field is hidden and an Integration Type of *None* is assumed.
Default Value	This is the default value the Parameter will get when the Phase is added to an Environment and no value has been explicitly set. This field is optional.
Repeat Default Value	Mandatory field for secured Parameters: repeat the secured default value.

Field	Meaning
Description	In this field, enter a description for the Parameter.
Mandatory	This field indicates whether the Parameter will always be created when adding the Phase to an Environment. When a Mandatory Parameter is created, it will be automatically created in the Environments where this Phase has been added to. When a non-Mandatory Parameter is set to Mandatory, it will also be automatically created in the Environments where this Phase has been added to.

5. Click *Create* to create the Phase Parameter.

   When clicking the *Create* button, the Phase Parameter is created and the pop-up window closes. The new Parameter is added to the *Phase Parameters* panel.

   You can also click:

   - *Reset* to clear the fields and restore its initial values.
   - *Cancel* to close the pop-up window without creating the Phase Parameter.

RELATED TOPICS
- Creating Phase Parameters (page 435)
- Editing Phase Parameters (page 437)
- Deleting Phase Parameters (page 440)
- Mass Editing Phase Parameters (page 440)
- Viewing the Level Phase Parameters (page 198)
- Viewing the Build Environment Phase Parameters (page 220)
- Viewing the Deploy Environment Phase Parameters (page 235)

32.6. Editing Phase Parameters

1. In the Global Administration context, select *Phases* > *Overview*.

2. Click the *Edit* link in front of the required Phase on the *Phases Overview* panel.

   The *Edit Phase* screen is displayed.

3. On the *Phase Parameters* panel, click the *Edit* link in front of the Parameter you want to edit.

The following pop-up window will be displayed:

4. Edit the fields as required.

   For a description of the fields, refer to Creating Phase Parameters (page 435).

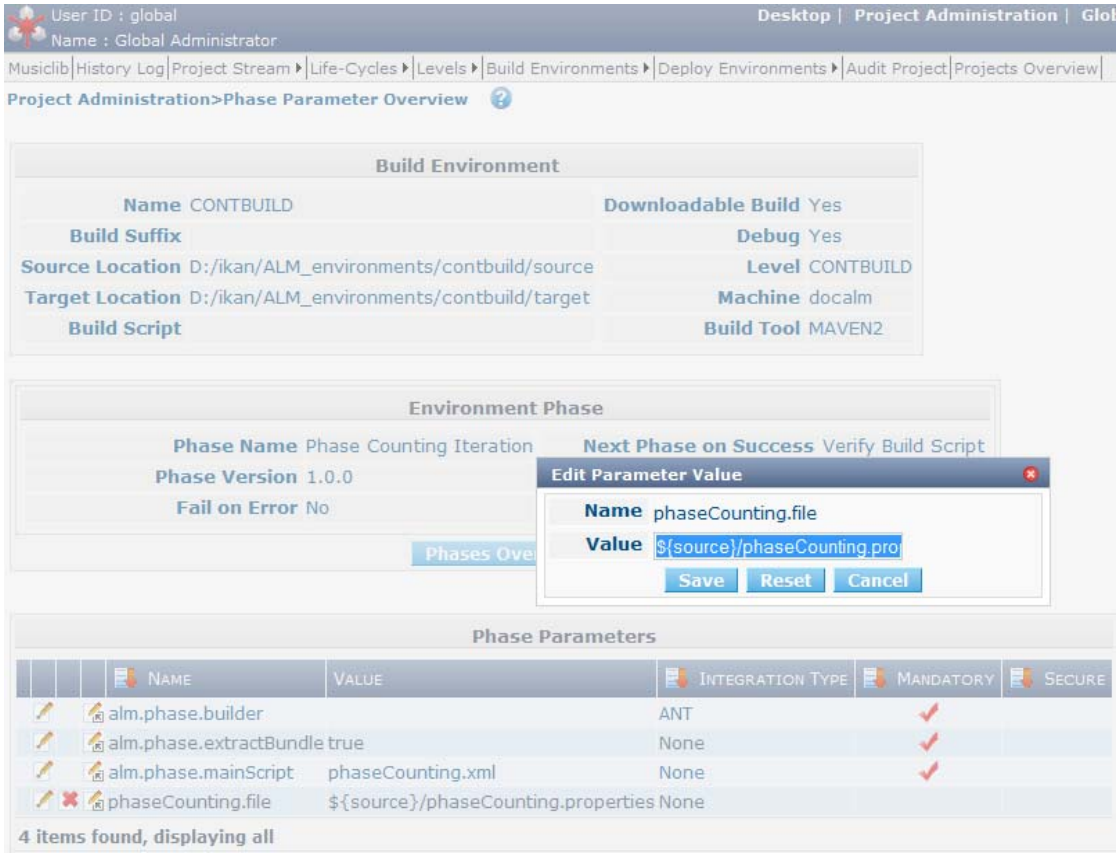
5. Verify the connected Environment Parameters.

   The *Connected Environment Parameters* panel shows the Environments where this Phase Parameter has been added to, and the values of the Parameter in those Environments.

   Note: An Environment is identified by its Project, Level, and, optionally, its Environment name. When the Environment field is empty, this means that the Phase has been added to the Level.

6. Click the *Edit Environment Phase Parameter* link next to an Environment Parameter.

The user will be redirected to the *Phase Parameter Overview* screen (in the Project Administration context) and the *Edit Parameter Value* pop-up window is opened.



7. Set the value of the Environment Parameter and click *Save* to save the value.

   You can also click:

   - *Reset* to retrieve the settings from the database.
   - *Cancel* to return to the *Phase Parameter Overview* screen without saving a value. See The Phase Parameters Overview Screen on page 434.

   To go back to the *Edit Phase Parameter* window (in the Global Administration context), click one of the *Edit Global Phase Parameter* links.

RELATED TOPICS
- The Phase Parameters Overview Screen (page 434)
- Creating Phase Parameters (page 435)
- Deleting Phase Parameters (page 440)
- Mass Editing Phase Parameters (page 440)
- Viewing the Level Phase Parameters (page 198)
- Viewing the Build Environment Phase Parameters (page 220)
- Viewing the Deploy Environment Phase Parameters (page 235)

32.7. Deleting Phase Parameters

1. In the Global Administration context, select *Phases > Overview*.

2. Click the *Edit* link in front of the required Phase on the *Phases Overview* panel.
   The *Edit Phase* screen is displayed.

3. On the Phase Parameter panel, click the *Delete* link in front of the Parameter you want to delete.
   The following pop-up window will be displayed:

If the Parameter has been created in 1 or more Environments, the following Warning message is shown:

4. Click *Delete* to confirm the deletion.
   The parameter will be removed from all connected Environments and from the Phase.
   You can also click *Cancel* to close the pop-up window without deleting the Parameter.

RELATED TOPICS
- The Phase Parameters Overview Screen (page 434)
- Creating Phase Parameters (page 435)
- Editing Phase Parameters (page 437)
- Mass Editing Phase Parameters (page 440)

32.8. Mass Editing Phase Parameters

1. In the Global Administration context, select *Phases > Overview*.

2. Click the *Edit* link in front of the required Phase on the *Phases Overview* panel.
   The *Edit Phase* screen is displayed.

3. On the *Phase Parameters* panel, click the *Mass Edit* link in front of the Parameter you want to edit.

   The following pop-up window will be displayed, showing the different connected Environment Phase Parameters with the Project and Level or Build/Deploy Environment and the Parameter value.

Note: If a label has been specified for a specific phase, you can display it by hovering the icon in the outer right column.
For more information on the usage of labels, refer to the section Inserting a Level Phase (page 201).

4. Select one or more items on the *Connected Environment Parameters* list.

5. Select or type a new value for the selected Parameters in the *Set value for selection* field, and click *Set*.

   After confirming, the values of the selected Environment Phase Parameters will be set to the specified value.

   If the Parameter is secured, the value has to be repeated in the *Repeat Value* field.

6. Click *Reset*.

   After confirming, the values of the selected Environment Phase Parameters will be set to the default value of the Phase Parameter.

7. Click *Delete*.

   After confirming, the selected Environment Phase Parameters will be removed from their Environments.

   The *Delete* action is only available for non-mandatory Parameters.

8. Click *Cancel* to close the pop-up window.

RELATED TOPICS
- The Phase Parameters Overview Screen (page 434)
- Creating Phase Parameters (page 435)
- Editing Phase Parameters (page 437)
- Viewing the Level Phase Parameters (page 198)
- Viewing the Build Environment Phase Parameters (page 220)
- Viewing the Deploy Environment Phase Parameters (page 235)

32.9. Mass Replacing Phases

The *Mass Replace Phase* option allows you to replace a Phase in several Environments of different Projects at once, which can be useful when installing a new version of a Phase in multiple Projects. That is a much easier process than having to remove the Phase and insert the Replacement Phase in each Level, Build/Deploy Environment Phases Overview.

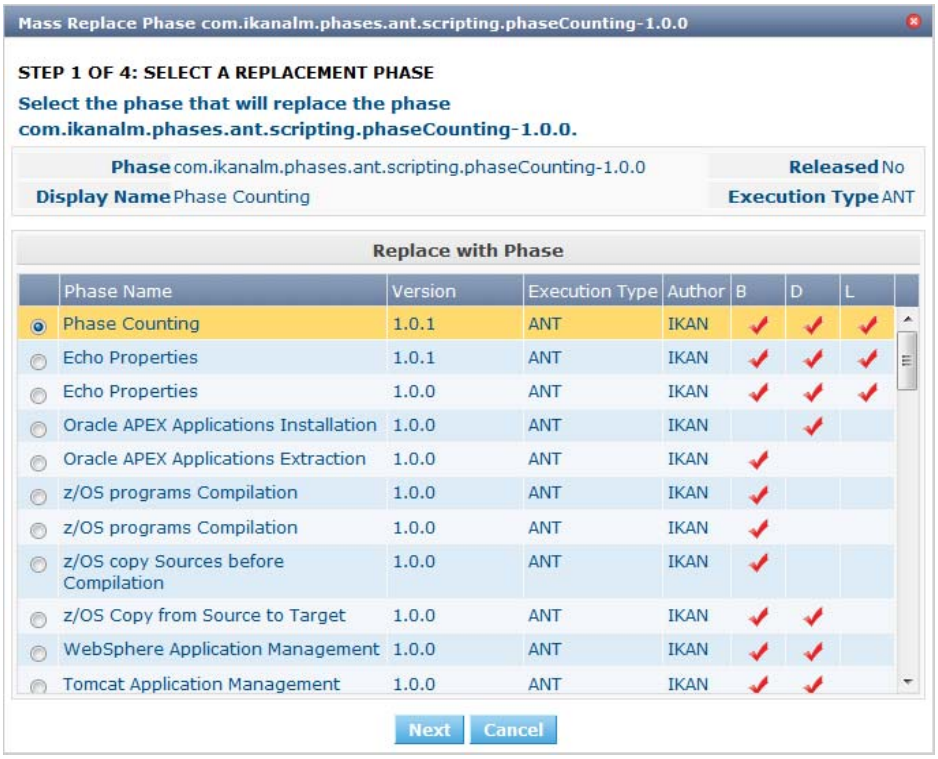
Note: You need Global Administration Security Rights in order to execute a *Mass Replace Phase*.

1. In the Global Administration context, select *Phases > Overview*.

2. Click the *Edit* link in front of the required Phase on the *Phases Overview* panel.
   The *Edit Phase* screen is displayed.

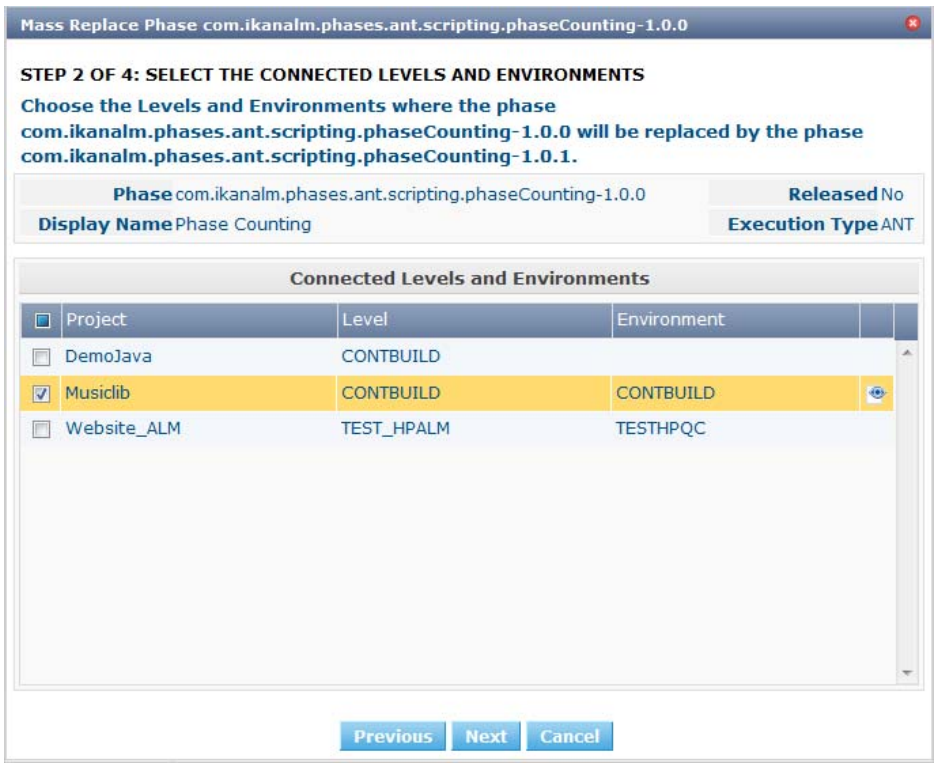
3. On the *Connected Levels and Environments* panel, click the *Mass Replace Phase* link.
   The Mass Replace Phase wizard is displayed. This wizard will guide you through the four steps of the Mass Replace Phase process.

3.1.STEP 1 - Select a Replacement Phase



Select the Phase that will replace the original Phase from the *Replace With Phase* table and click the *Next* button.

3.2.STEP 2 - Select the Connected Levels and Environments

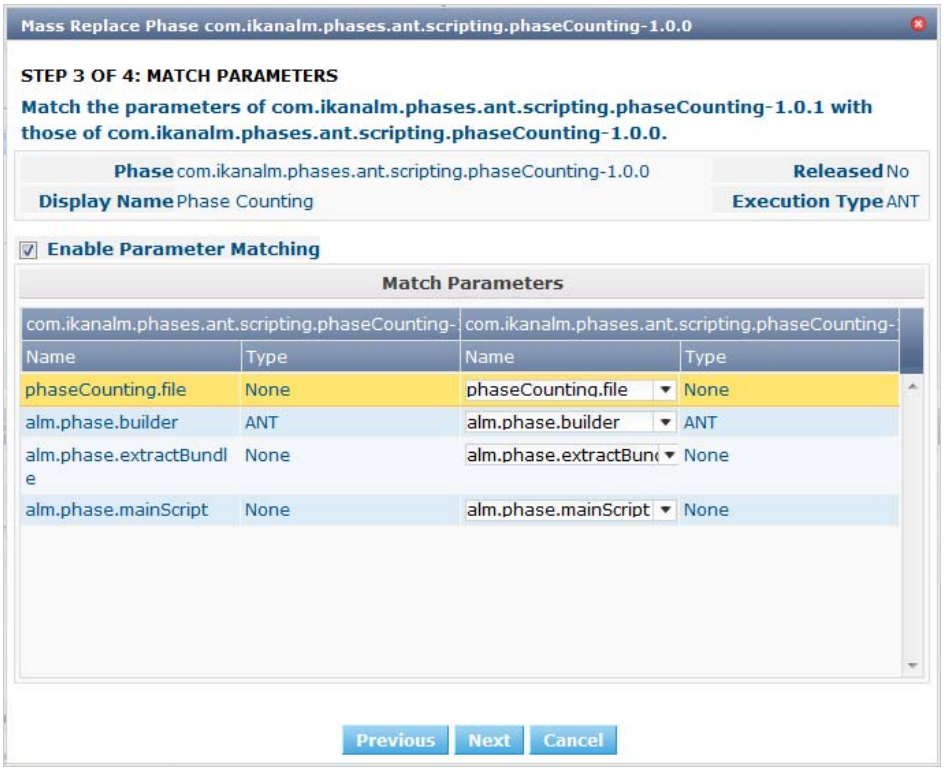


From the table of *Connected Levels and Environments*, select the Levels and Environments for which the original phase will be replaced by the phase you selected in step 1.

If you select the checkbox in the header, all Levels and Environments will be selected.

Note: If a Phase has a Label for a specific Environment, you can check its contents by hovering the icon.

3.3.STEP 3 - Match Parameters



In this step you have to match the parameters of the original Phase with those of the Replacement Phase.

IKAN ALM will try to match parameters with an identical name and type. If needed, you can always correct those automatic matches, or match unlinked parameters by selecting the appropriate parameter from the drop-down list.

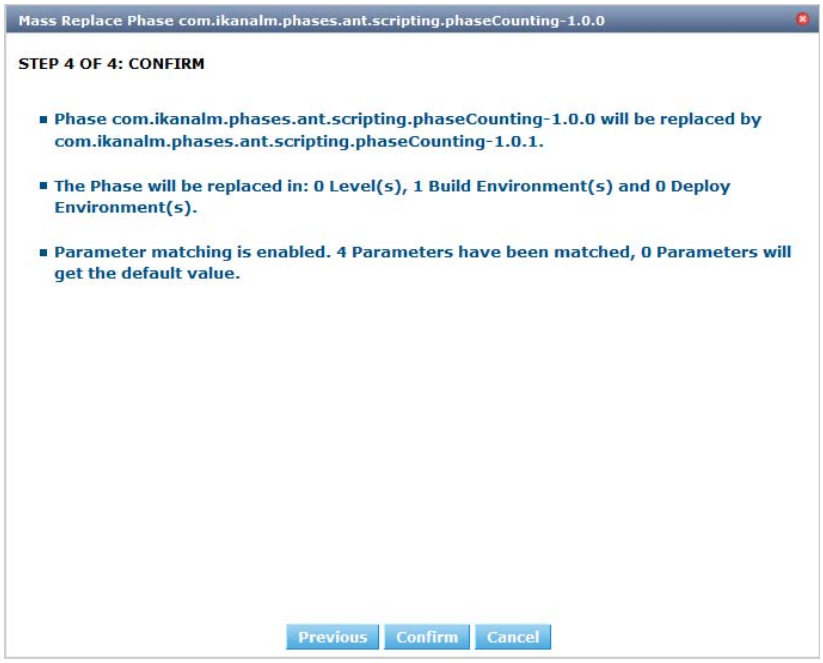
Matched parameters will get the value from the original Environment Parameter.

Unmatched parameters for which a default parameter has been specified, will be initialized using that parameter. If no default parameter has been specified, you can always specify it later. See Mass Editing Phase Parameters on page 440.

If you do not activate the *Enable Parameter Matching* option, the Environment Parameters will get the default value (if it has been set), or will stay empty.

Select *Next* if the parameter matching is OK.

3.4.STEP 4 - Confirmation Screen



On the confirmation screen you can verify all the choices before actually replacing the phase:

- the Phase that will replace the original Phase (selected in Step 1)
- in how many Levels, Build and Deploy Environments it will be replaced (selected in Step 2)
- how the Parameters will be matched (selected in Step 3)

Click the *Confirm* button to replace the Phase. Next, the Phase will be replaced in the different Environments.

RELATED TOPICS

- Viewing the Level Phase Parameters (page 198)
- Viewing the Build Environment Phase Parameters (page 220)
- Viewing the Deploy Environment Phase Parameters (page 235)
- Mass Deleting Phases (page 445)

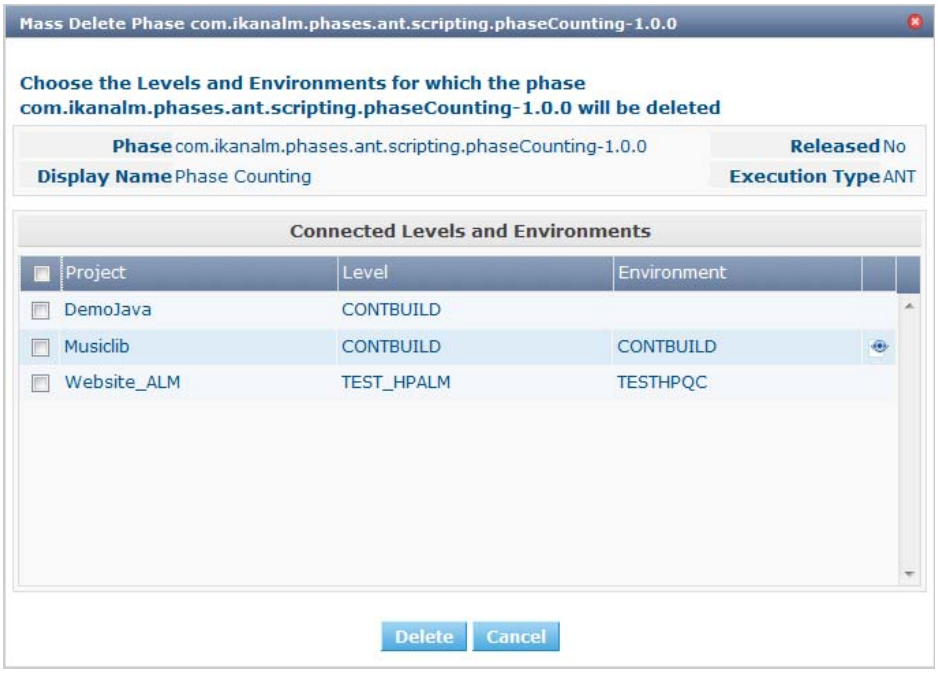
32.10.Mass Deleting Phases

The *Mass Delete Phase* option allows you to delete a Phase in several Environments of different Projects at once. This is a much easier process than having to remove the Phase in each Level, Build/Deploy Environment Phases Overview.

Note: You need Global Administration Security Rights in order to execute a *Mass Delete Phase*.

1. In the Global Administration context, select *Phases* > *Overview*.
2. On the *Phases Overview* panel, click the *Edit* link in front of the required Phase.
   The *Edit Phase* screen is displayed.
3. On the *Connected Levels and Environments* panel, click the *Mass Delete Phase* link.

The following pop-up window is displayed.



4. Select the Levels and/or Environments you want to delete the Phase from.
   If you select the checkbox in the header, all Levels and Environments will be selected.

Note: If a Phase has a Label for a specific Environment, you can check its contents by hovering the icon.

5. Click *Delete*.
   After confirming, the selected Phase(s) will be removed from their Levels and/or Environments.
   You can also click *Cancel* to return to the *Edit Phase* screen.

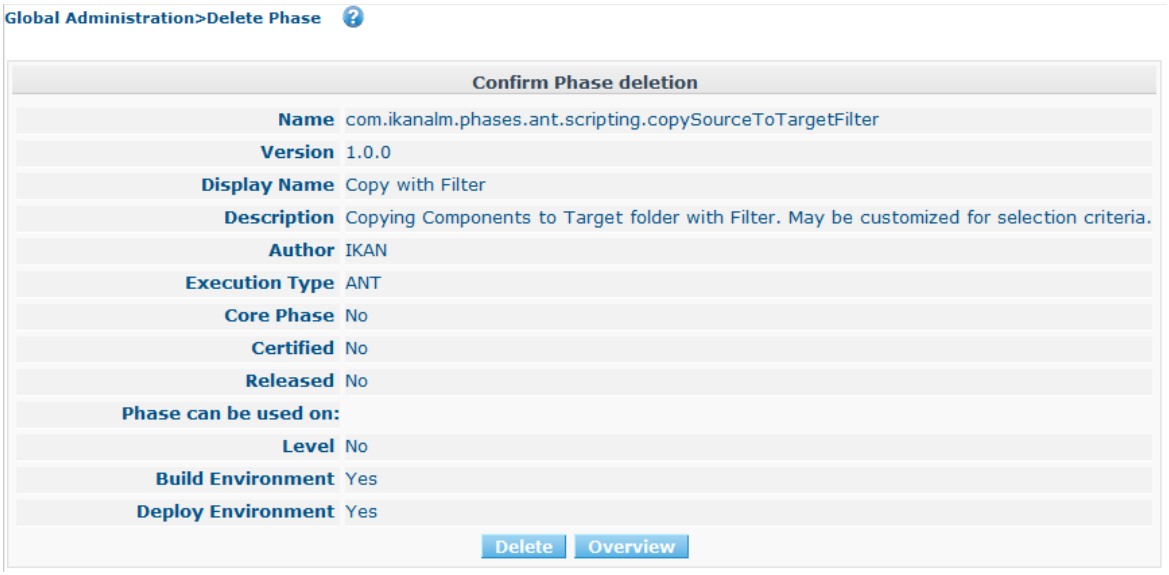
RELATED TOPICS

- Mass Replacing Phases (page 442)

32.11.Deleting a Phase Definition

1. In the Global Administration context, select *Phases* > *Overview*.

2. On the *Phases Overview* panel, click the *Delete* link in front of the Phase you want to delete.
   The *Confirm Phase deletion* screen is displayed.



3. Click *Delete* to confirm the deletion.
   You can also click *Overview* to return to the previous screen without deleting the Phase.
   Note: If you try to delete a Phase connected to at least one Environment, the following message is displayed:

   Before you can delete it, you must remove the Phase from all Environments it is connected to.

> Deleting a Phase definition will also delete the corresponding .jar file from the Phase Catalog location.

32.12.Exporting a Phase Definition

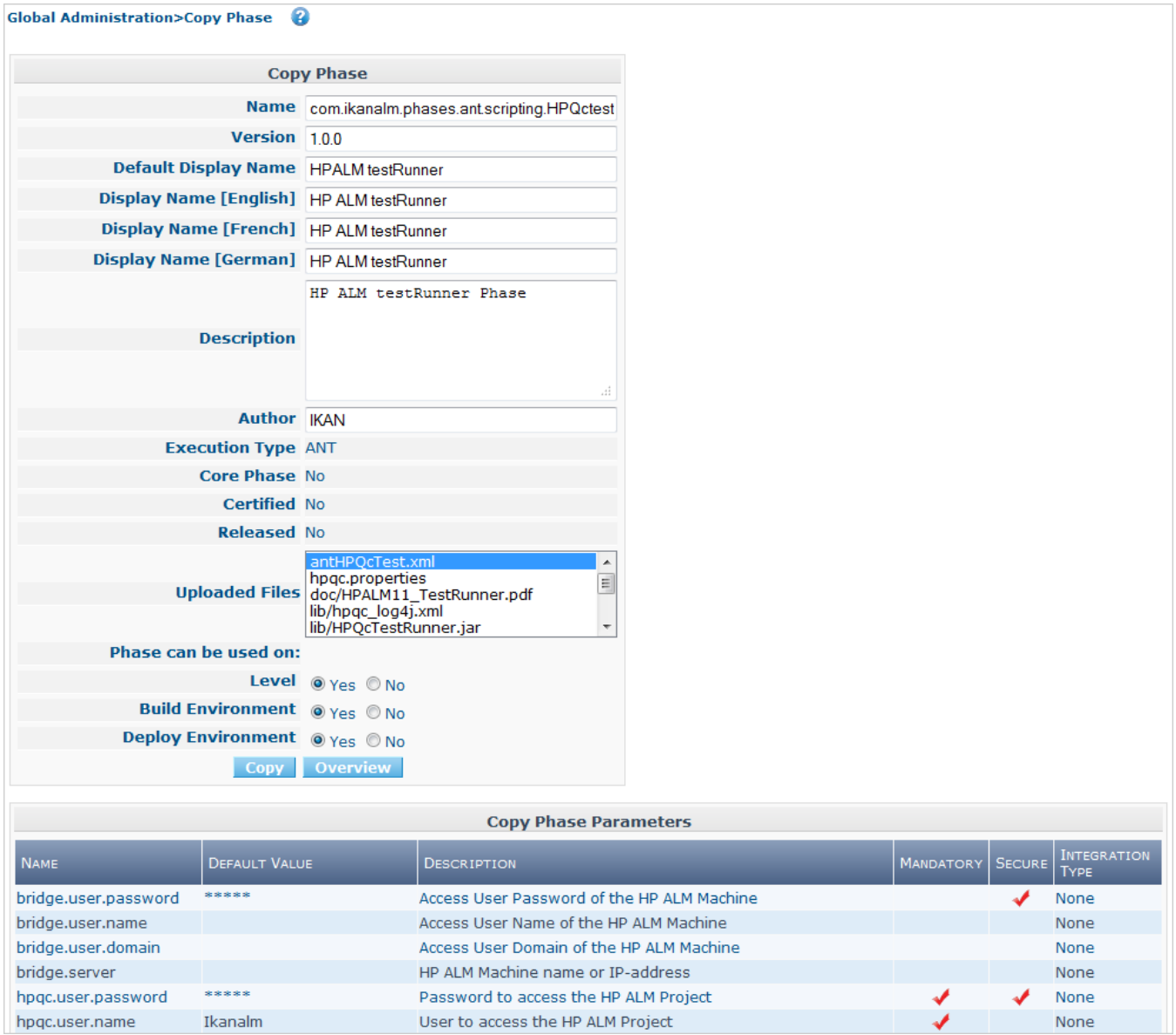
1. In the Global Administration context, select *Phases* > *Overview*.

2. On the *Phases Overview* panel, click the *Export* link in front of the Phase you want to export.
   IKAN ALM packages the Phase metadata and all the script files in a .jar file. A file dialog screen will be displayed, asking you where you want to save this .jar file.The exported .jar file can be used to import the Phase again in the future, possibly in another IKAN ALM installation. See Importing a Phase Definition on page 450.

32.13.Copying a Phase Definition

1. In the Global Administration context, select *Phases* > *Overview*.

2. On the *Phases Overview* panel, click the *Copy* link in front of the Phase you want to copy.
   The *Copy Phase* screen is displayed.



3. Modify the fields as required.
   For a description of the fields, refer to the section Creating a Phase Definition (page 425).

Note: The combination Name - Version must be unique, so at least one of those fields must be modified to be able to save the copy of the Phase. If you are upgrading the Phase, you (most likely) increment the *Version* value.

4. Verify the uploaded files.

   The *Uploaded Files* field lists the current contents of the Phase. You cannot upload new files on this screen, you must first complete the copy, and then edit the Phase. See Editing a Phase Definition on page 429.

5. Verify the Phase Parameters.

   The *Copy Phase Parameters* panel displays all the Parameters of the Phase that will be copied. All Parameters will be copied to the new Phase.

6. Click *Copy* to copy the Phase.

   When you click *Copy*, a new Phase will be created with the specified properties: all the displayed Phase Parameters will be created, and the user is redirected to the *Phases Overview* screen.

   You can also click *Overview* to return to the *Phases Overview* screen without saving the changes.

32.14. Viewing the Phase History

1. In the Global Administration context, select *Phases* > *Overview*.

2. On the *Phases Overview* panel, click the *History* link in front of the Phase you want to display the history for.

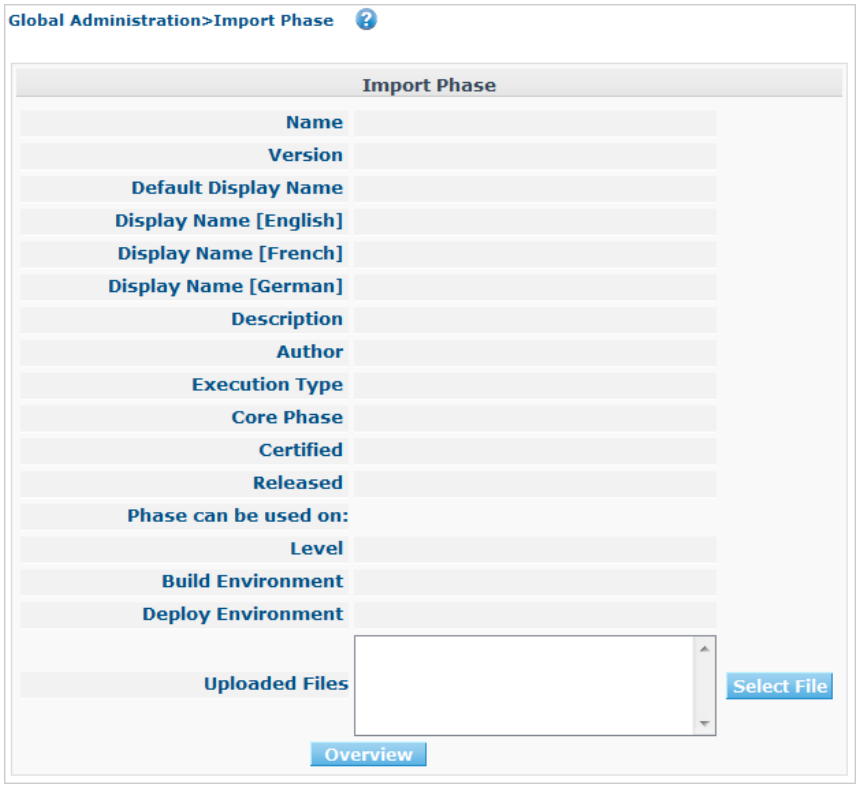
   The *Phase History View* screen is displayed.

   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Phases Overview* screen.

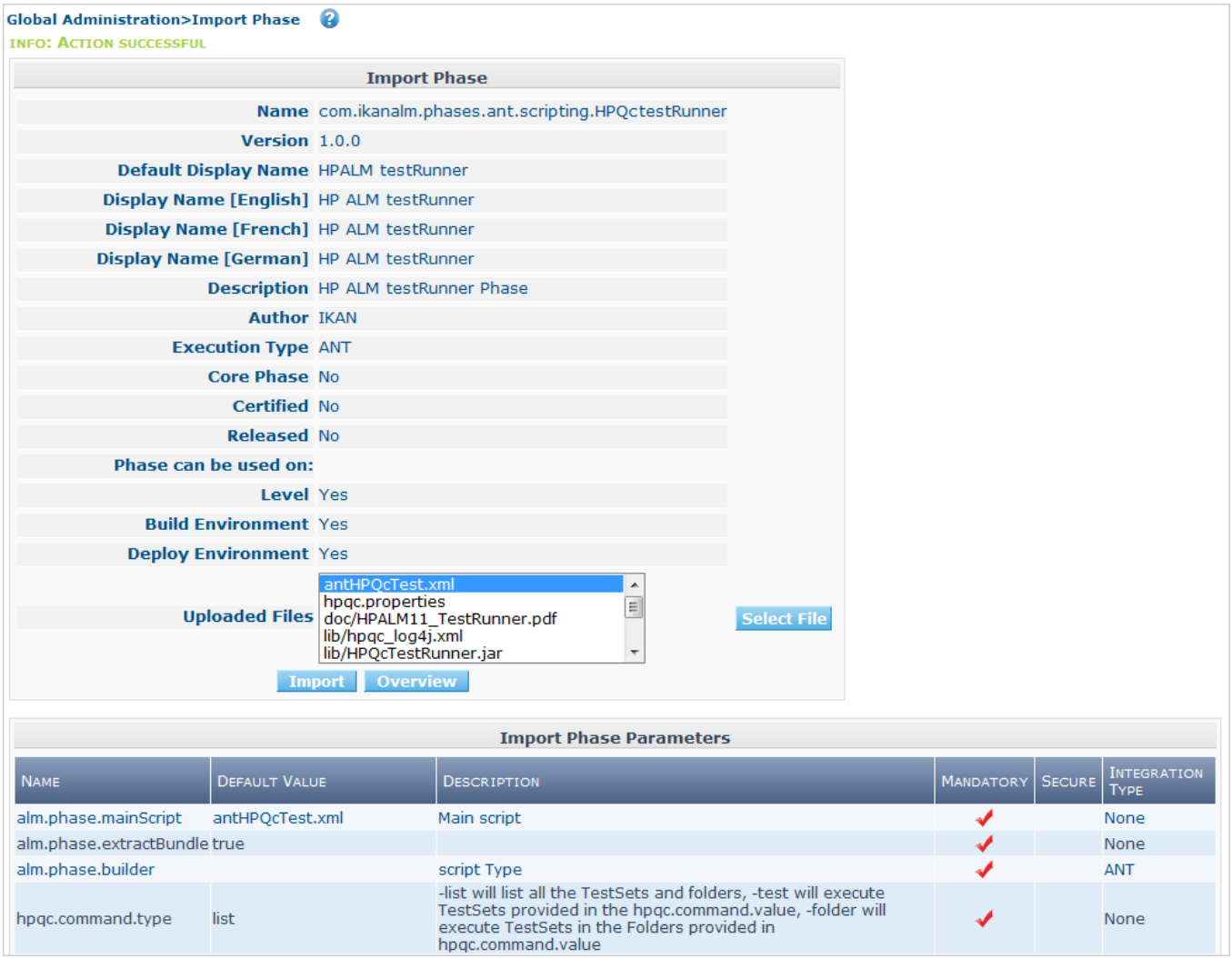
32.15.Importing a Phase Definition

1. In the Global Administration context, select *Phases* > *Import*.
   The *Import Phase* screen is displayed.



2. Click the *Select File* button to choose the Phase to be imported.
   A file dialog window opens, where you can choose a .jar file that contains a previously exported Phase.

Once you have selected a file, it will be uploaded and the Phase information contained in it will be read and then be displayed:



3. Verify the properties of the Phase to be imported.

   All the properties of the Phase will be shown in the fields. For a description of the fields, refer to the sections Creating a Phase Definition (page 425)and Editing a Phase Definition (page 429).

   The scripts and other files contained within the Phase are shown in the *Uploaded Files* field.

   The *Import Phase Parameters* panel shows the defined Parameters of the Phase that will be imported.

4. Click *Import* to import the Phase.

   When clicking *Import*, the Phase and its Parameters are created in the IKAN ALM database. The scripts and other files contained within the Phase are packaged into a .jar file and copied to the Phase Catalog location (as defined in the System Settings (page 255)).

   You can also click *Overview* to return to the *Phases Overview* screen without importing the Phase.

CHAPTER 33
Schedules

A Schedule definition enables to automate the creation of Level Request for a particular Build Level.

When a Schedule is linked to a Build Level, the IKAN ALM Scheduler Daemon will verify if there are changes in the Version Control Repository of the Project Stream the Level is linked to each time the Schedule interval expires.

If changes are found, a Level Request will automatically be created for the Build Level.

In other words, Schedules enable features like Build Automation or Continuous Integration.

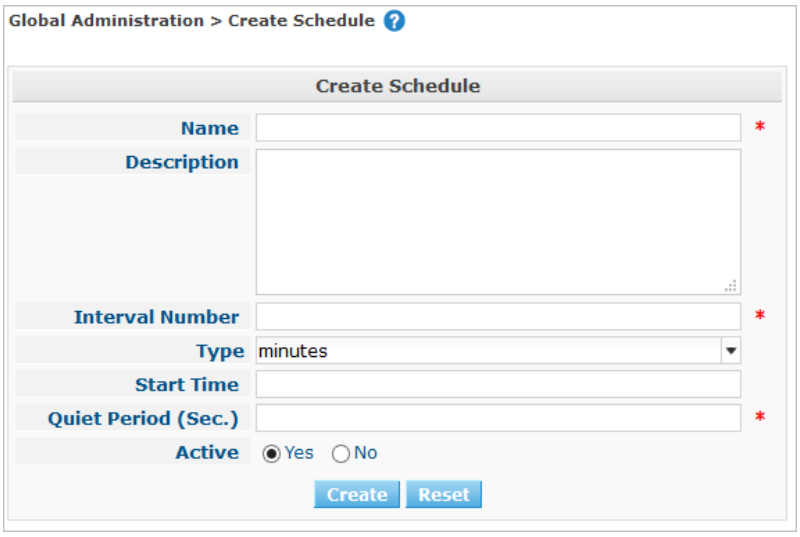
Refer to the following sections for detailed information:

- Creating a Schedule (page 452)
- The Schedules Overview Screen (page 456)
- Editing a Schedule Definition (page 458)
- Deleting a Schedule Definition (page 459)
- Viewing the Schedule History (page 460)

33.1. Creating a Schedule

Note: This option is only available if you have Global Administrator Access Rights in IKAN ALM.

1. In the Global Administration context, select *Schedules* > *Create*.
   The following screen is displayed:



2. Fill out the fields in the *Create Schedule* panel at the top of the screen. Fields marked with a red asterisk are mandatory:

Field	Meaning
Name	Enter the name of the new Schedule in this field. This field is mandatory.
Description	Enter a description for the new Schedule in this field. This field is optional.
Interval Number	Enter the number of interval units by which the Schedule must verify outstanding tasks. For instance, if the Interval Unit is set to MINUTES and the Interval Number to 10, the Schedule will check for changes in the Version Control Repository every 10 minutes.
Interval Unit	Select the required Interval Unit from the drop-down list. The following options are available: • minutes • hours • days
Start Time	Enter the time when the Schedule must start. This field is used once each time the Scheduler is restarted and when the Schedule is set to *Active*.
Quiet Period (Sec.)	Enter a Quiet Period value (between 0 and 3600 seconds) for the new Schedule. The Quiet Period avoids that a Build is performed while people are checking in files. When a Level Request is due to be started by the Scheduler, the system will check first if any source code activity took place during the defined number of seconds. Only if there was no source code activity, it will be assumed safe to update from the source control system and execute the Level Request.
Active	Select whether the Schedule is active or not.

Note: Avoid using a too "aggressive" Schedule that, for instance, will check for changes every 1 minute. This might stress the VCR, particularly when the schedule is attached to multiple Build Levels. Instead, use a "hook script" in the VCR that launches an IKAN ALM Level Request from the IKAN ALM Commandline Interface when a source has been committed. Please consult the documentation of your VCR for more information concerning hook scripts.

Note: A schedule that is attached to a Level of a tag-based Project Stream will not check for changes in the VCR and, hence, will have no effect, since it has no way of checking what has changed in the VCR since the last Level Request. Continuous Integration on a tag-based Project Stream is only possible by setting up VCR hook scripts that launch an IKAN ALM Level Request from the IKAN ALM Commandline Interface when a new tag has been created. Please consult the documentation of your VCR for more information concerning hook scripts.

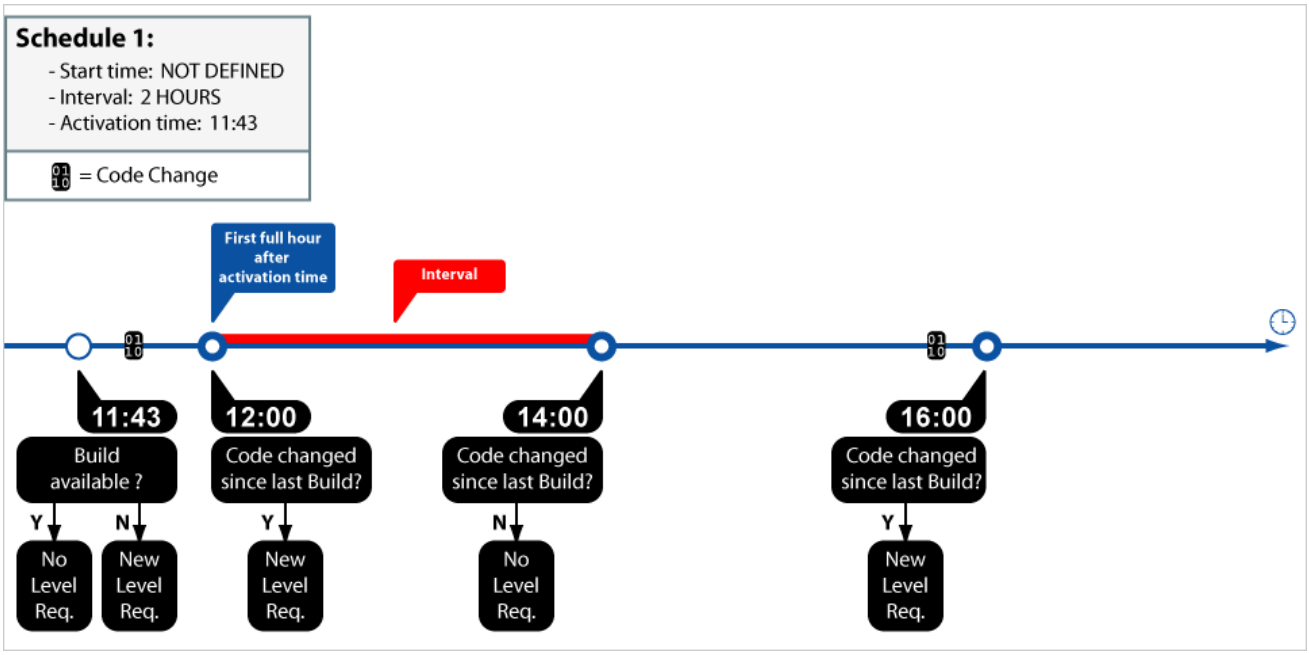
3. Once you filled out the fields as required, click *Create*.

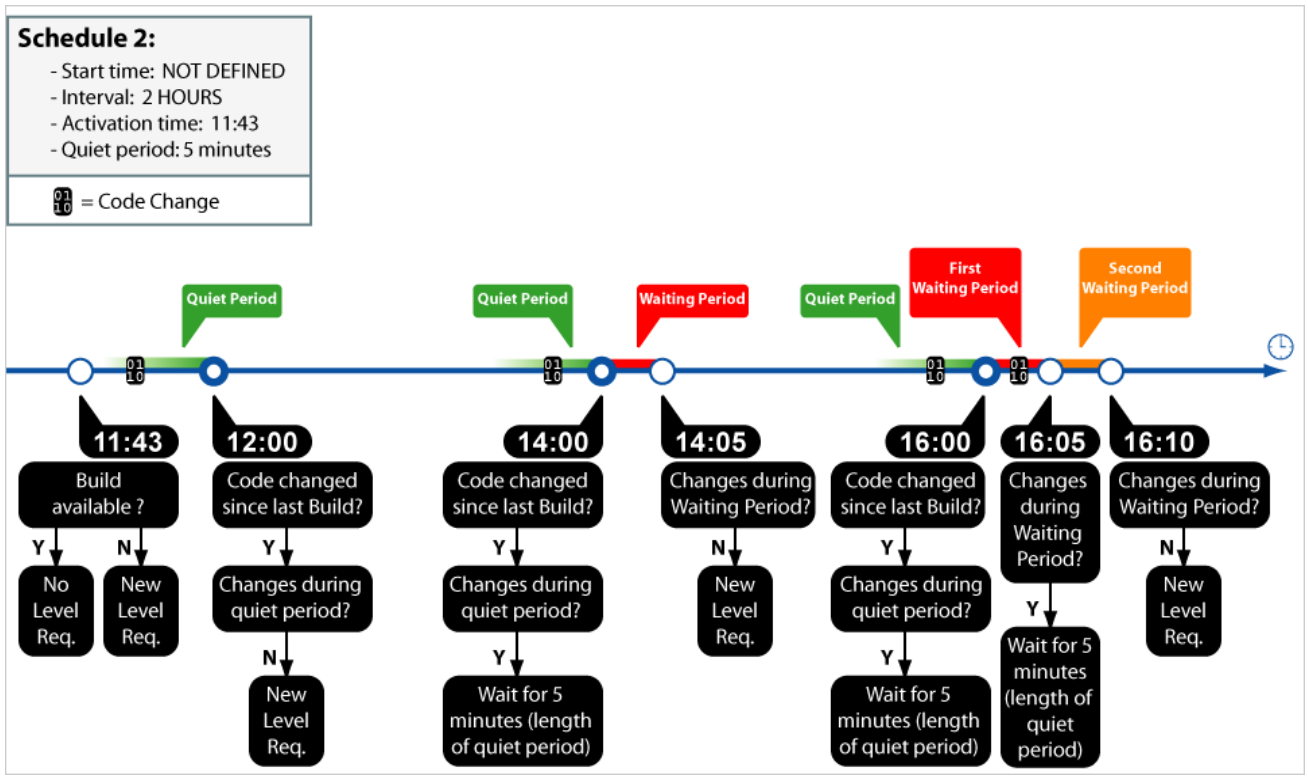
The newly created Schedule is added to the *Schedules Overview* at the bottom of the screen.

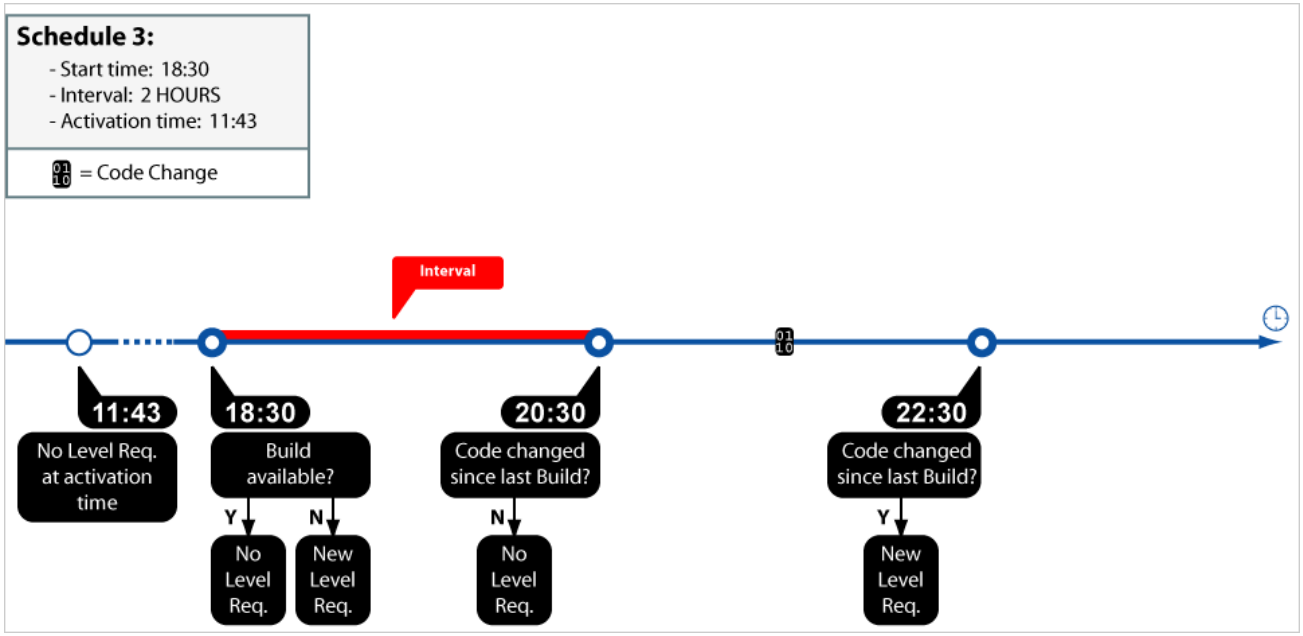
Samples:

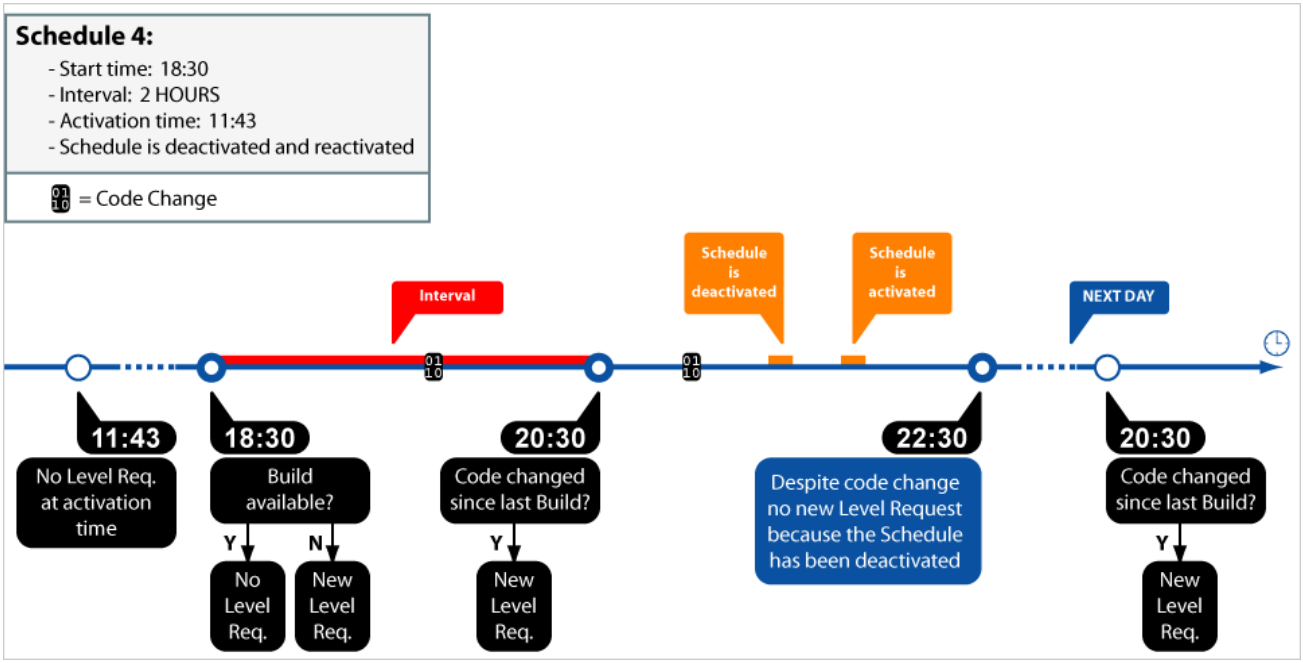
In the samples the question *"Build Available?"* means: *"Is there a Build Level Request for the Level linked to the Schedule?"*

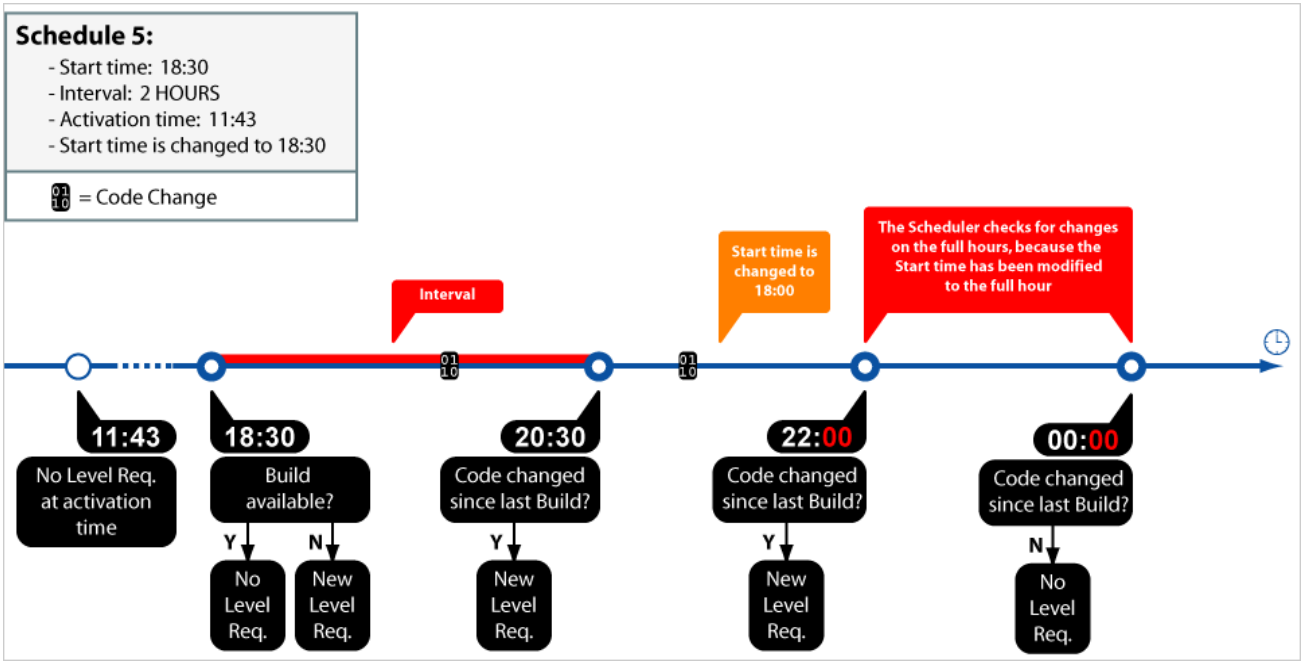
The question *"Code changed since last build?"* means: *"Have the resources in the VCR been changed since the last Build Level Request for the Level linked to the Schedule?*






Schedule 3:
- Start time: 18:30
- Interval: 2 HOURS
- Activation time: 11:43
= Code Change
Interval
11:43
No Level Req. at activation time
18:30
Build available?
Y
No Level Req.
N
New Level Req.
20:30
Code changed since last Build?
N
No Level Req.
22:30
Code changed since last Build?
Y
New Level Req.

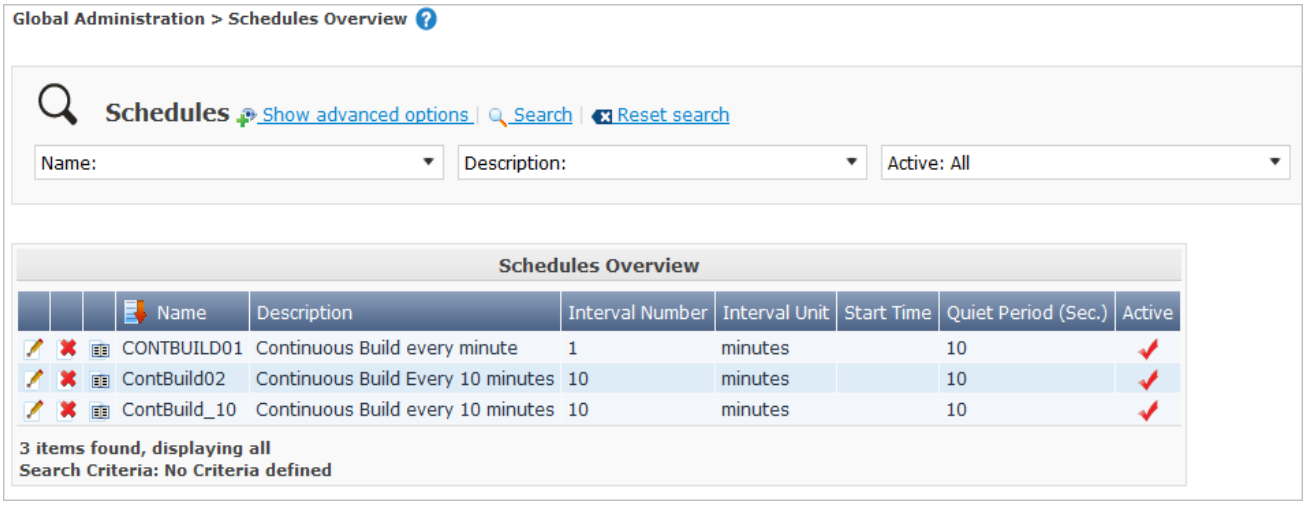

Schedule 4:
- Start time: 18:30
- Interval: 2 HOURS
- Activation time: 11:43
- Schedule is deactivated and reactivated
= Code Change
Interval
Schedule is deactivated
Schedule is activated
NEXT DAY
11:43
No Level Req. at activation time
18:30
Build available?
Y
No Level Req.
N
New Level Req.
20:30
Code changed since last Build?
Y
New Level Req.
22:30
Despite code change no new Level Request because the Schedule has been deactivated
20:30
Code changed since last Build?
Y
New Level Req.



RELATED TOPICS
- Schedules (page 452)
- Creating a Build Level (page 188)
- Creating Level Requests (page 45)

33.2. The Schedules Overview Screen

1. In the Global Administration context, select *Schedules > Overview*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.

You can also:

- click the *Show/hide advanced options* link to display or hide all available search criteria,
- click the *Search* link to refresh the list based on the current search criteria,
- click the *Reset search* link to clear the search fields.

3. Verify the information on the *Schedules Overview* panel.
   For a detailed description of the fields, refer to Creating a Schedule (page 452).

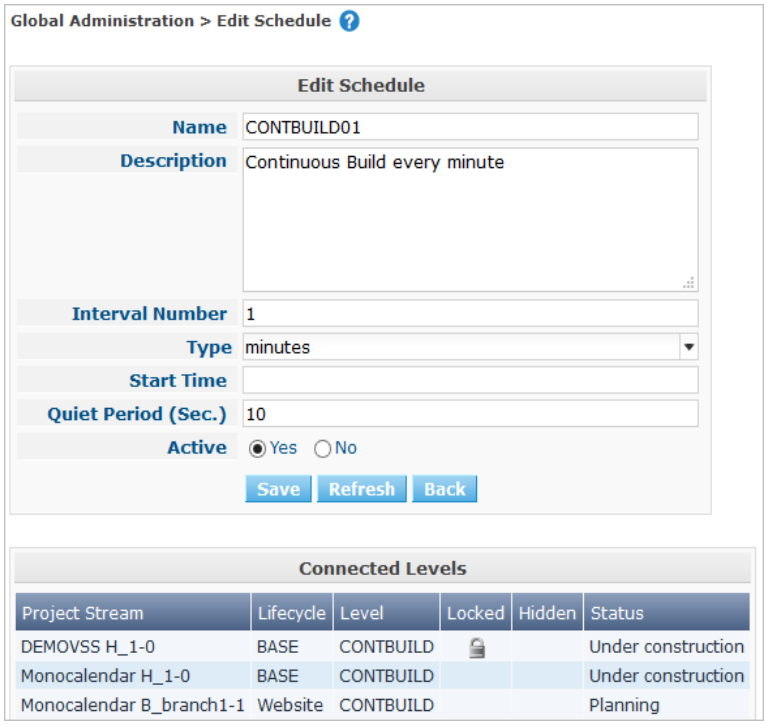
4. Depending on your access rights, the following links may be available on the *Schedules Overview* panel:

	Edit This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows editing a Schedule definition. See Editing a Schedule Definition on page 458.
	Delete This option is available to IKAN ALM Users with Global Administrator Access Rights. It allows deleting a Schedule definition. See Deleting a Schedule Definition on page 459.
	History This option is available to all IKAN ALM Users. It allows displaying the History of all create, update and delete operations performed on a Schedule See Viewing the Schedule History on page 460.

Editing a Schedule Definition

1. In the Global Administration context, select *Schedules* > *Overview*.

2. Click the *Edit* link to change the selected Schedule.
   The following screen is displayed:



3. Edit the fields as required.
   For a description of the fields, refer to Creating a Schedule (page 452).

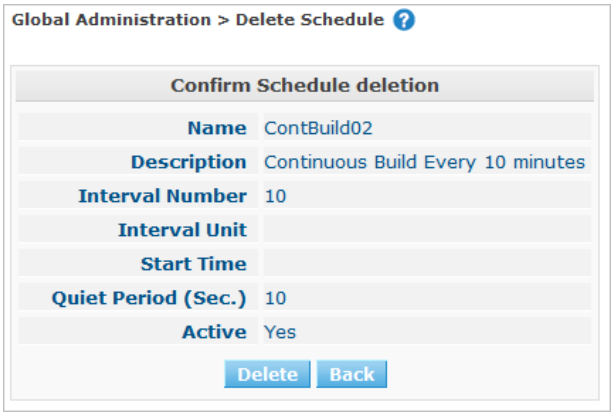
Note: The *Connected Levels* panel displays the Levels the Schedule is linked to.

4. Click *Save* to save your changes.
   You can also click:
   - *Refresh* to retrieve the settings from the database.
   - *Back* to return to the previous screen without saving the changes

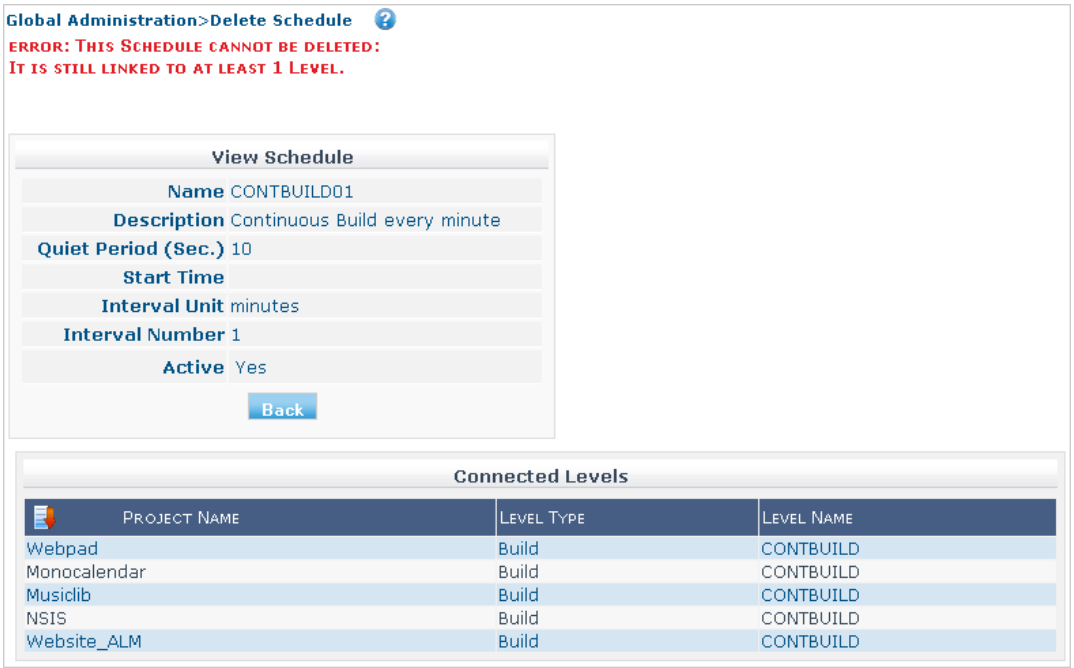
Deleting a Schedule Definition

1. In the Global Administration context, select *Schedules* > *Overview*.

2. Click the *Delete* link to delete the selected Schedule.
   If the Schedule is not linked to any Level, the following screen is displayed:



3. Click *Delete* to confirm the deletion.
   You can also click *Back* to return to the previous screen without deleting the entry.
   Note: If the Schedule is linked to one or more Levels, the following screen is displayed:



   You must remove the Schedule from the listed Levels before you can delete it.

Viewing the Schedule History

1. In the Global Administration context, select *Schedules* > *Overview*.

2. Click the *History* link to display the *Schedule History View*.
   For more detailed information concerning this *History View*, refer to the section History and Event Logging (page 497).

3. Click *Back* to return to the *Schedules Overview* screen.

RELATED TOPICS
- Schedules (page 452)
- Creating a Build Level (page 188)
- Creating Level Requests (page 45)

CHAPTER 34
Projects

Note: The *Project* menu is only available to users with Global Administrator Access Rights.

The Project menu options are explained in the following procedures:

- Creating a Project (page 461)
- Cloning an Existing Project (page 468)

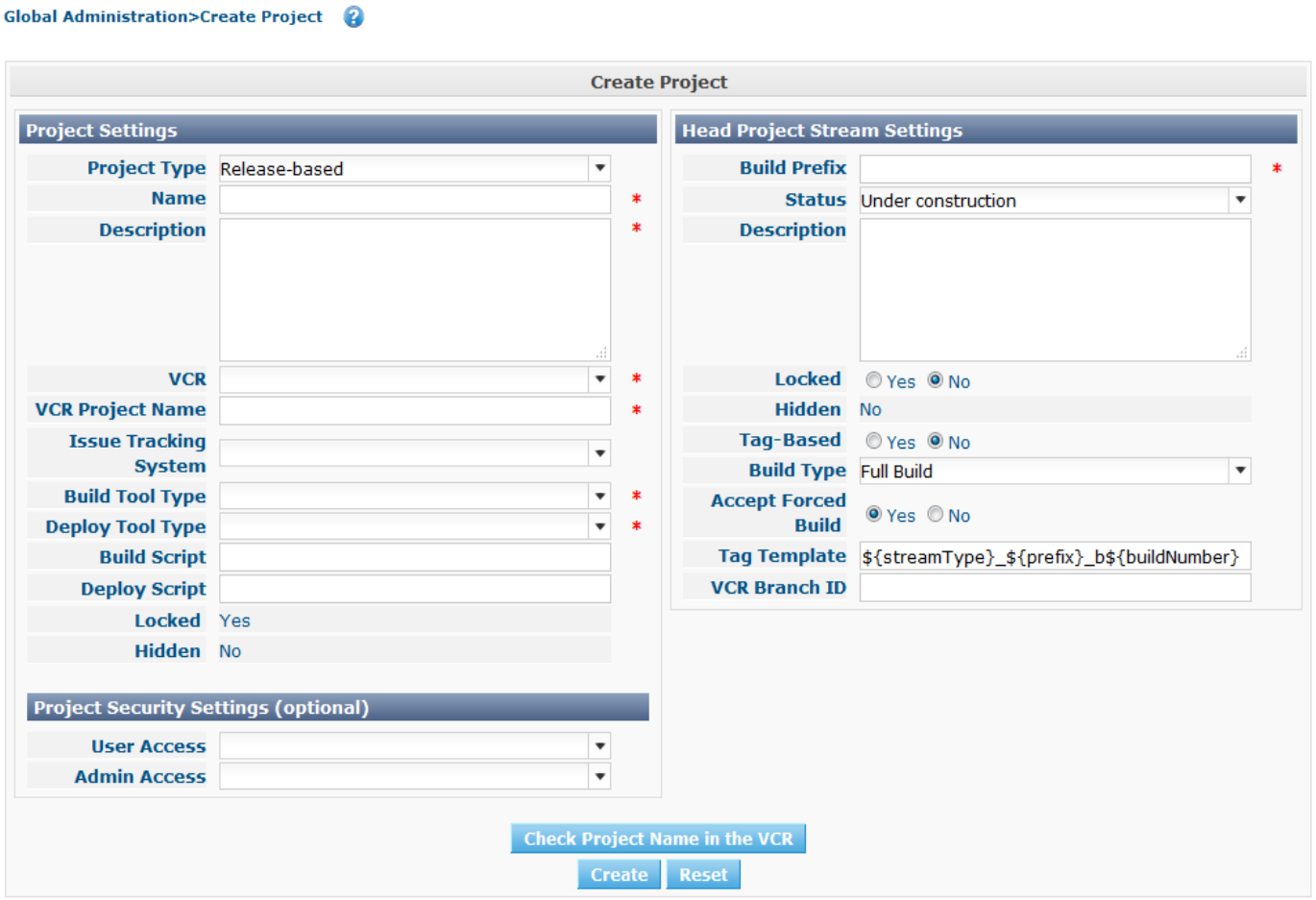
34.1. Creating a Project

There are 2 types of Projects:

- Release-based Projects: IKAN ALM will work with the existing structure in the VCR (Version Control Repository) system, so that the objects to be extracted from the VCR will be retrieved automatically when starting the build process.
- Package-based Projects: IKAN ALM introduces the Package concept, which enables to work with isolated files from the VCR system. Objects must be selected manually in a Package structure created in IKAN ALM before starting the Build process.

Note: Package-based Projects must connect to a VCR of type Subversion, TFVC or ClearCase. Other VCR types do not support working with isolated files and thus not with Packages in IKAN ALM.

1. In the Global Administration context, select *Project > Create Project.*
   The following screen is displayed:



2. Fill out the fields in the different sections.
   Fields marked with a red asterisk are mandatory.
   - Project Settings

Field	Meaning
Project Type	Select the type of Project you want to create: Release-based or Package-based. Depending on the type of Project you select, other fields will be available for editing on the *Head Project Stream Settings* panel.
Name	Enter the name of the new Project in this field.
Description	Enter a description for the new Project in this field.
VCR	From the drop-down list, select the VCR to be used for this IKAN ALM Project.

Field	Meaning
VCR Project Name	Enter the name of the VCR Project matching this IKAN ALM Project. It is possible that the VCR Project Name does not match the IKAN ALM Project Name. You can verify whether the project with the specified VCR Project Name exists in the linked VCR by clicking the *Check Project Name in the VCR* button. **Note:** This field stays empty for Subversion VCRs with single-project layout and for Git VCRs.
Issue Tracking System	Select one of the available Issue Tracking Systems from the drop-down menu. If an Issue Tracking System is specified in this field, IKAN ALM will always automatically create an Issue Tracking Level Phase when a new Level is created for the Project. This field is only visible if external Issue Tracking Systems have been defined in IKAN ALM. For more information on Issue Tracking systems, refer to the chapter Issue Tracking (page 381)
Build Tool Type	From the drop-down list, select the Build Tool Type (Ant, Gradle, Nant or Maven2) to be used for this IKAN ALM Project. Only Scripting Tools of the selected Build Tool Type may be linked to Build Environments that will be created in the Project.
Deploy Tool Type	From the drop-down list, select the Deploy Tool Type (Ant, Gradle, NAnt or Maven2) to be used for this IKAN ALM Project. Only Scripting Tools of the selected Deploy Tool Type may be linked to Deploy Environments that will be created in the Project.
Build Script	In this field, enter the name of the Build Script to be used for this IKAN ALM Project. This script will be used by default when executing Builds in the Project, but may be overwritten when defining a Build Environment.
Deploy Script	In this field, enter the name of the Deploy Script to be used for this IKAN ALM Project. This script will be used by default when executing Deploys in the Project, but may be overwritten when defining a Deploy Environment.
Locked	This read-only field is set to *Yes*, because an IKAN ALM Project can only be unlocked when is audited and fully configured.
Hidden	This read-only field is defaulted to *No*, because an IKAN ALM Project cannot be hidden when it is still being created.

- Project Security Settings (Optional)

Field	Meaning
User Access	From the drop-down list, select the User Group that will have User Access Rights for the new IKAN ALM Project. If defined, this setting overrides the *User Group with User Access Rights* defined in the *System Settings* dialog. See System Settings on page 255. This group is the default group that is set as Requester when creating a Level Request.
Admin Access	From the drop-down list, select the User Group that will have Administrator Access Rights for the new IKAN ALM Project. If defined, this setting overrides the *User Group with Admin Access Rights* defined in the *System Settings* dialog. See System Settings on page 255.

- Head Project Stream Settings

Field	Meaning
Build Prefix	Enter the Build Prefix to be used for the HEAD Project Stream of the new IKAN ALM Project.
Status	Select the required status for the HEAD Project Stream of the new IKAN ALM Project.
Description	Enter a description for the HEAD Project Stream of the new IKAN ALM Project.
Locked	Select whether or not the HEAD Project Stream of the new IKAN ALM Project must be locked. By default, this option is set to *No*.
Hidden	This read-only field is defaulted to *No*, because an IKAN ALM Project Stream cannot be hidden when it is still being created.
Tag-Based	When creating a Release-based Project, select whether or not the HEAD Project Stream of the new IKAN ALM Project must be Tag-based. By default, this option is set to *No*. In a Tag-Based Project Stream, the Builds on the Build Level will be executed on sources with a pre-applied tag in the VCR, whereas in non Tag-Based projects these Builds will be executed on the latest sources. When creating a Package-based Project, this option is unchangeably set to *No*.
Build Type	When creating a Release-based Project, select the required Build Type from the drop-down menu. The following options are available: • *Full Build* • *Partial Build: based on a user-selected tag* • *Production-based Partial Build: based on the tag of the Build currently in production* When creating a Package-based Project, this option is unchangeably set to *Full Build*.

Field	Meaning
Accept Forced Build	When creating a Release-based Project, select whether or not the HEAD Project Stream of the new IKAN ALM Project must accept Forced Builds, i.e., allow Builds to bypass the Schedule set on the Build Level. By default, this option is set to *Yes*. When creating a Package-based Project, this option is unchangeably set to *No*.
Tag Template	This field contains the Template that should be used for tagging Builds generated in this Project Stream. The Tags will be used to display these Builds in the VCR. A template can be a combination of *constants* and *variables*. The following *constants* are allowed: • letters • numbers • underscores • hyphens • Some other characters (such as , ; $ @) may be illegal, depending on the VCR type. The following *variables* are available. They all have the format *${name}*: • *${prefix}*: This variable returns the Build Prefix defined for the Project Stream. • *${suffix}*: This variable returns the Build Suffix defined for the Branch Stream. (If this variable is included in a Template for a Head Stream, it is ignored.) • *${streamType}*: This variable returns the Type indication for a Stream: *H* for Head Stream or *B* for Branch Stream. • *${buildNumber}*: This variable returns the number of the Build, so that the Build can be matched with its Tag. • *${projectName}*: This variable returns the name of the IKAN ALM Project. • *${vcrProjectName}*: This variable returns the VCR name of the Project. • *${dateTime(dateformat)}*: This variable returns the timestamp of the Building action in the indicated format. *yyyy-MM-dd* is a valid format. For other valid formats, refer to http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html • ${packageName}: This variable returns the name of the Package associated to the Level, when the Build is executed (only used for Package-based Projects).
VCR Branch ID	This field contains the VCR Branch ID. The same ID must be available in the VCR linked to the Project. This field is required in case of a ClearCase UCM (mapped to the Integration Stream) or Team Foundation Projects(mapped to Main Branch ID). For Head Streams belonging to other VCR Project types, this field remains empty.

The following table contains some examples of Tag Templates and the resulting tags:

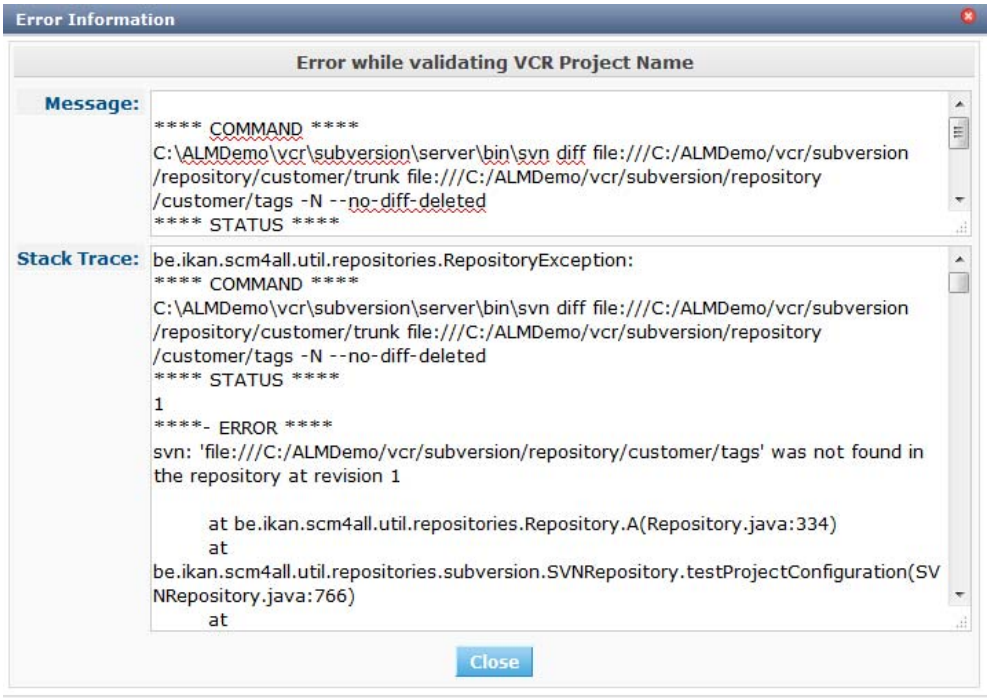
Tag Template	Resulting VCR Tag
${streamType}_${prefix}_${suffix}_b${buildNumber}	*H_1.0_b5* where: • *H* = Head Stream Type • *1.0* = Build Prefix • *5* = Build Number **Note:** The Suffix variable was ignored for the Head Stream! *B_1.0_bugfix_b5* where: • *B* = Branch Stream Type • *1.0* = Build Prefix • *bugfix* = Build Suffix • *5* = Build Number
${projectName}_${packageName}_b ${dateTime(yyyy-MM-dd)}_b${buildNumber}	*DEMOCVS_ Package2_2006-03-27_b5* where: • *DEMOCVS* = Project Name • *2006-03-27* = Creation timestamp of the Build • Package2 = Name of the Package used for the Build • *5* = Build Number

3. Click the *Project Name in the VCR* button to check the availability of the VCR Project Name.

   If the test is successful, the following message is displayed:

   Info: The Project Name exists in the VCR.

   If the test is not successful, the following pop-up window is displayed:



Correct the errors reported in the Stack Trace field and perform the test again.

4. Once the test is successful, click *Create*.

The Project will be created and you are forwarded to the *Project Info* screen in the Project Administration context. You can now administer the newly created Project by creating Levels etc. For more information, refer to the section Editing Project Settings (page 149).

Underneath the *Project Info* panel, the following links and buttons are available:

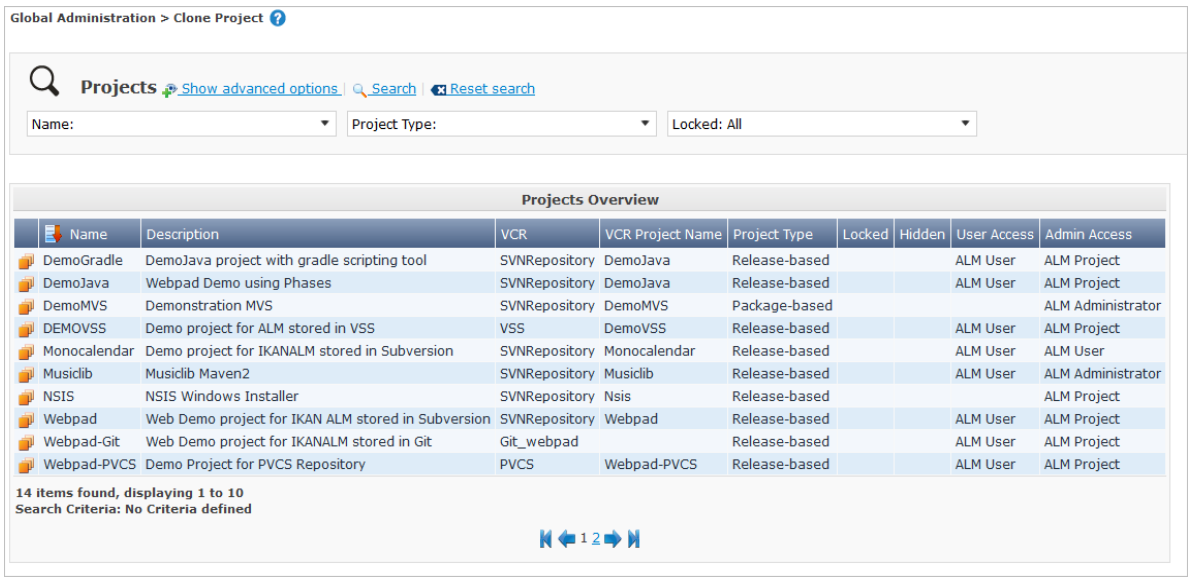
- *History*. This link will display the *Project History View* screen.
- *Unlock* to lift the blocking of all activity on this Project, like manual or scheduled Level Request. You must unlock a Project after having completed the maintenance.
- *Edit* to edit the Project Settings.
- *Refresh* to retrieve the settings from the database.

RELATED TOPICS

- Editing Project Settings (page 149)
- Project Streams (page 161)
- Lifecycles (page 174)
- Levels (page 187)
- Build Environments (page 209)
- Deploy Environments (page 225)

34.2. Cloning an Existing Project

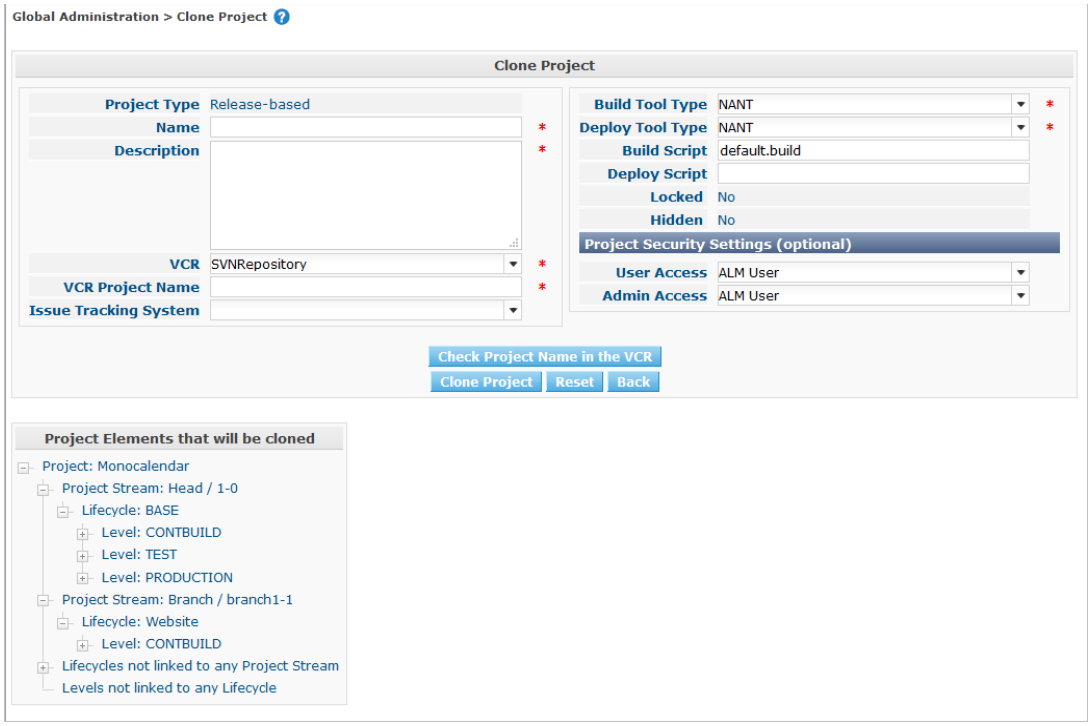
1. In the Global Administration context, select *Project > Clone Project*.
   The following screen is displayed:



2. Define the required search criteria on the search panel.
   The list of items on the overview will be automatically updated based on the selected criteria.
   You can also:
   - click the *Show/hide advanced options* link to display or hide all available search criteria,
   - click the *Search* link to refresh the list based on the current search criteria,
   - click the *Reset search* link to clear the search fields.

3. Next, click the *clone* link in front of the required Project.

The following screen is displayed:



4. Fill out or edit the available fields.

The following fields are available:

Field	Meaning
Project Type	The Project Type used for the cloned Project is displayed and cannot be changed.
Name	Enter the name of the new Project in this field.
Description	Enter a description for the cloned Project in this field.
VCR	By default, the VCR used for the cloned Project is displayed. You may select another VCR from the drop-down list, if required.
VCR Project Name	Enter a VCR Project name for the new Project in this field.
Issue Tracking System	By default, the Issue Tracking System used for the cloned Project will be selected. You can select another Issue Tracking System from the drop-down list, if required.
Build Tool Type	By default, the Build Tool Type used for the cloned Project is displayed. You may select another Build Tool Type from the drop-down list, if required.
Deploy Tool Type	By default, the Deploy Tool Type used for the cloned Project is displayed. You may select another Deploy Tool Type from the drop-down list, if required.
Build Script	By default, the Build Script used for the cloned Project is displayed. You may select another Build Script from the drop-down list, if required.
Deploy Script	By default, the Deploy Script used for the cloned Project is displayed. You may select another Deploy Script from the drop-down list, if required.

Field	Meaning
Locked	This read-only field is set to *No*, because a new IKAN ALM Project is configured accordant to the cloned Project.
Hidden	This read-only field is defaulted to *No*, because an IKAN ALM Project cannot be hidden when it is still being created.
Project Security Settings - User Access	By default, the User Group with User Access Rights used for the cloned Project is displayed. You may select another User Group from the drop-down list, if required. If defined, this setting overrides the *User Group with User Access Rights* defined in the *System Settings* dialog. See System Settings on page 255. This group is the default group that is seen as Requester when creating a Level Request.
Project Security Settings - Admin Access	By default, the User Group with Admin Access Rights used for the cloned Project is displayed. You may select another User Group from the drop-down list, if required. If defined, this setting overrides the *User Group with Admin Access Rights* defined in the *System Settings* dialog. See System Settings on page 255.

5. Expand the tree structure of the *Project Elements to be cloned* and check the validity of the Elements.

   It contains the following object types:

   - Project Streams
   - Lifecycles
   - Levels
   - Level Phases
   - Build Environments
   - Deploy Environments
   - Environment Phases
   - Build Parameters
   - Deploy Parameters
   - Phase Parameters
   - Approvals

6. Fill out the fields as required and click *Check Project Name in the VCR*.

7. If the Project exists in the VCR, click *clone Project*.

   The Project will be cloned.

8. Edit the definition of the required Objects.

All objects belonging to the new Project are exact copies of their counterparts in the cloned Project. It is the responsibility of the User to verify which objects need to be changed in the new Project. For instance, the Target Location of Deploy Environments in the new Project most likely need to be changed.

9. Audit the Project to unlock it.
   See Auditing Projects on page 247.

CHAPTER 35
Miscellaneous

Note: The *Miscellaneous* menu is only available to users with Global Administrator Access Rights.

The Miscellaneous menu options are explained in the following procedures:

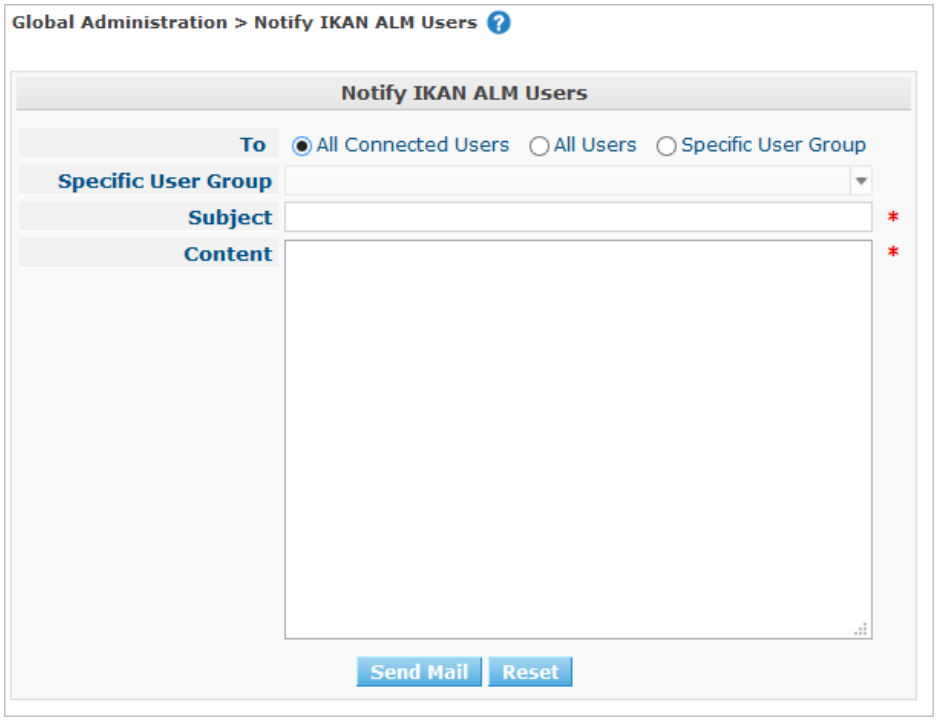
- Notifying IKAN ALM Users (page 472)

35.1. Notifying IKAN ALM Users

This option allows sending e-mail messages to other IKAN ALM users from within IKAN ALM.

This makes it possible, for example, to send a warning to connected users, or to notify users belonging to a certain User Group that some settings have been changed. It is also a mean to test the Mail settings specified under *System Settings*.

1. In the Global Administration context, select *Miscellaneous > Notify IKAN ALM Users*.
   The following screen is displayed:



The following fields are available:

Field	Meaning
To	Select whether you want to send your message to the connected IKAN ALM Users, to the members of a specific User Group or to all IKAN ALM Users.
Specific User Group	If you send your message to a specific User Group, select the required User Group from the drop-down list. If you do not send your message to a specific User Group, leave this field blank.
Subject	Enter a Subject description in this field.
Content	Enter the message body text in this field.

2. Make the required selections and enter your message. Then click *Send Mail*.

The IKAN ALM Users matching your selection will receive your message. The *SMTP From User* defined in the System Settings will be marked as the Sender of the message.

You can also click the *Reset* button to clear the fields.

RELATED TOPICS

- System (page 255)

Command Line Interface

CHAPTER 36
Command Line Interface (Optional)

IKAN ALM can be extended with an optional Command Line interface, allowing the creation of Level Requests and the generation of Reports by means of a shell interface instead of the IKAN ALM web application.

Possible uses:

- Manual creation of Level Requests (CLR Command): See Creating Level Requests by Specifying Options and Parameters on page 475.
- Manual generation of Reports (Report Command): See Generating Reports by Specifying Options and Parameters on page 482.
- Creation of Level Requests in an external IDE. This guide describes the integration procedure for JDeveloper (page 490) and Eclipse (page 492). Refer to the indicated sections.
- Automated creation of Level Requests using a Scripting Tool. A scripting tool is used to define the triggers (e.g., server backup) for Level Request creation. This guide describes the use of an ANT script (page 494). Refer to the indicated section.

The IKAN ALM Command Line interface can be installed on each client machine requiring its functionality. It requires a connection (TCP/IP) to the central IKAN ALM server. The connection with the IKAN ALM server is established using XML-RPC.

36.1. Manual Usage of the Command Line interface

Refer to the following procedures for detailed information:

- Creating Level Requests by Specifying Options and Parameters (page 475)
- Creating Level Requests Using a Properties File (page 479)
- Generating Reports by Specifying Options and Parameters (page 482)
- Generating Reports Using a Properties File (page 488)

Creating Level Requests by Specifying Options and Parameters

1. Prepare the Level Request creation.

   On a Windows platform:

   - Open a Command prompt.
   - Browse to the IKAN ALM Command Line working folder. After a standard installation, this is: *C:\alm\commandline*

You may add this location to the PATH Environment variable, so that it becomes accessible through other working directories.

On a Unix platform:

- Open a terminal console.
- Browse to the IKAN ALM Command Line working directory.

  After a standard installation, this is: */opt/ikan/alm/commandline*

  You may create a symbolic link to the `alm.sh` script so that it becomes accessible through other working directories.

2. Enter the required Level Request creation instruction.

Format:

alm -serverhost:$HOSTNAME [-serverport:$PORTNUMBER] -user:$USERID -password:$PASSWORD [-secured=true|false] clr parameters

The following elements are available:

Element	Description	Mandatory
alm	This element calls the *alm.cmd* (Windows) or *alm.sh* (Linux) file.	Yes
serverhost	This element allows defining the DHCP name or IP address of the host running the IKAN ALM Server.	Yes
serverport	This element allows defining the port number used to establish the connection to the IKAN ALM Server via XML-RPC. If this element is not provided, the default port number (20021) will be used.	No
user	This element allows defining a User ID with IKAN ALM access rights.	Yes
password	This element allows defining the password associated with the User ID used to access IKAN ALM.	Yes
secured	This element allows determining whether an encrypted connection must be used or not. If the IKAN ALM Server is using an encrypted connection, set the *secured* element to *true*: *-secured=true* If the IKAN ALM Server is not using an encrypted connection, set the *secured* element to *false* or omit the *secured* parameter out.	No
clr	This command stands for *Create Level Request*.	Yes
parameters	The Level Request creation parameters are explained in the following step.	Yes

3. Define the Level Request creation parameters.

Format:

clr -project:[project name] [-projectstream:[build prefix[-build suffix]]] [-package:[package name]] -level:[level name] [-description:[description]] [-buildnumber:[buildnumber]] [-tag:[vcr tag]] [-redeliver=true|false] [-startdate:[startdate]] [-paramfile:[parameter file]]

Parameter	Description	Mandatory
project	This parameter allows defining the Project for which you create a Level Request. Use the Project name as defined in IKAN ALM.	Yes
projectstream	This parameter allows defining the Project Stream for which you create a Level Request. Use the Build Prefix (and Build Suffix) as defined in IKAN ALM to identify the Project Stream to create a Level Request for. If this parameter is not provided, the Level Request will be created for the Head Project Stream.	No
package	This parameter allows defining the name of the Package for which a Level Request must be created.	Yes (only for Package-based projects)
level	This parameter allows defining the Level name for which you create a Level Request. Use the Level Name as defined in IKAN ALM.	Yes
description	This parameter allows defining the description of the Level Request.	No
tag	This parameter allows defining the Tag with which the Build will be tagged in the VCR. Only has effect on Level Requests of a Build Level If omitted, a tag name will be generated using the Tag Template of the Project Stream.	No
redeliver	This parameter allows redelivering previously delivered Build Results on Test and Production Levels. By default, this parameter is set to *false*: if no Level Request is available on the previous Level in the Lifecycle with a higher build number, the current active Level Request will NOT be redelivered via the commandline. In case you want to allow a redeliver, you must explicitly set the *redeliver* parameter to *true*.	No
startdate	This parameter allows defining the requested starting date and time of the Level Request. The accepted format is *dd/MM/yyyy HH:mm*. Only has effect on Level Requests of a Test or Production Level. If omitted, the Level Request will run as soon as possible.	No
paramfile	This parameter allows defining the name of the property file containing Build and Deploy Parameters in key=value format.	No

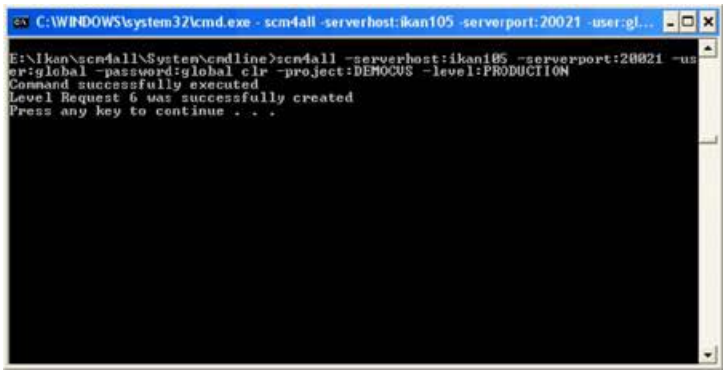
Note that you do not need to define the Level Request Type. The Level Request Type is determined automatically:

- For Build Levels with a Schedule, a Force Build Level Request will be created.
- For Build Levels without a Schedule, a Request Build Level Request will be created.
- For Test and Production Levels, a Deliver Level Request will be created, which will deliver the latest successful Level Request on the previous Level in the Lifecycle (whereas in the web application, you can select the Build to be delivered).

4. Once you have entered the complete command, press *Return*.

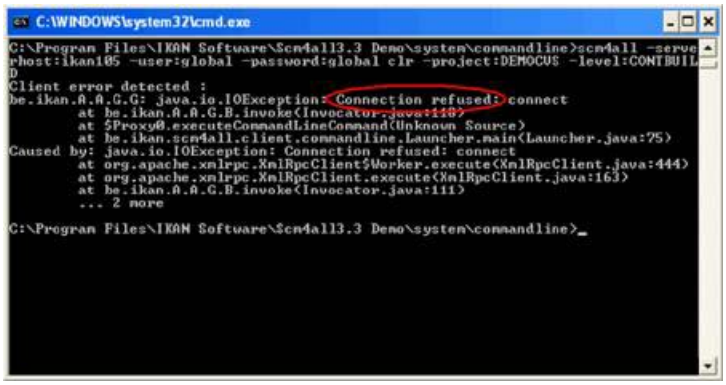
   Result:

   - If the Level Request is created successfully, the following screen is displayed:

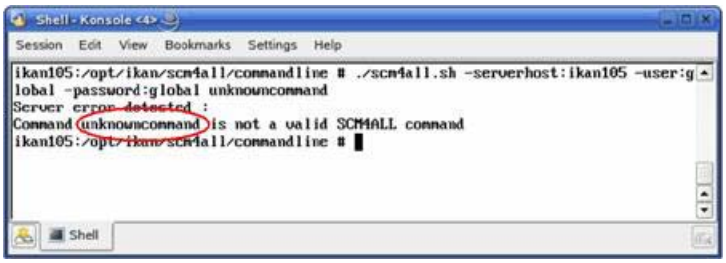


These messages only indicate that the Level Request was *created* successfully. Refer to the Level Requests Overview (page 58)in the web application to verify if the Level Request was *executed* successfully as well.

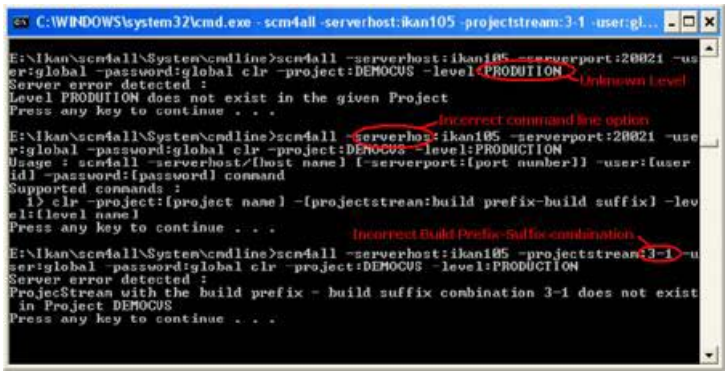
   - If the Level Request cannot be created, because there is no connection with the IKAN ALM Server, the following screen is displayed:



   - If the user entered an unknown command, the following screen is displayed:

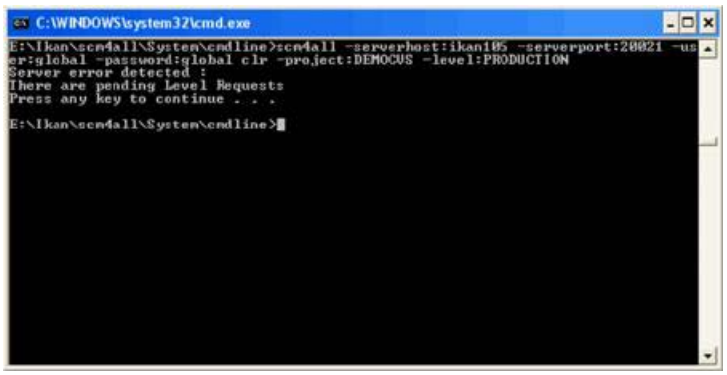


   - If the user entered unknown or incorrect command options, the screen similar to the following is displayed:



Similar error messages are provided for unknown Project or Package names and incorrect User ID/ Password combinations.

- If the Level Request cannot be created because of pending Level Requests for the Level, the following screen is displayed:



Similar error messages are shown if there is no suitable Build result to be delivered, the Level is locked, the Project Stream is locked or if there is an authorization error.

Creating Level Requests Using a Properties File

It is possible to save frequently used settings in a Properties file, so that you do not have to enter the complete Level Request creation parameters. After a standard installation, one such properties file, called *clr.properties* is available in the Command Line installation folder.

Open the file in a text editor to display its content:

```
#-------------------------------------------------
# SAMPLE property file for the IKAN ALM clr-command
#-------------------------------------------------

#-----------------
# GLOBAL OPTIONS :
#-----------------

# HOSTNAME OF THE IKAN ALM SERVER
serverhost=localhost

# PORT NUMBER OF THE IKAN ALM SERVER
#serverport=20021

# USER ID TO CONNECT TO THE IKAN ALM SERVER
user=global

# PASSWORD TO CONNECT TO THE IKAN ALM SERVER
password=global

# USE ENCRYPTED CONNECTION OR NOT (TRUE|FALSE)
secured=false

#-----------------------------------
# SPECIFIC OPTIONS FOR clr COMMAND :
#-----------------------------------

# NAME OF THE PROJECT
project=DEMOCVS

# [BUILD PREFIX]-[BUILD-SUFFIX] OF THE PROJECTSTREAM
projectstream=1-0
```

You can edit this standard file so that the settings match your requirements. Refer to the Creating Level Requests by Specifying Options and Parameters (page 475)for a description of the options and parameters. You can also create any number of specific properties files by copying the standard file, editing the copies and saving them under logical names for later usage.

1. Prepare the Level Request creation.

   On a Windows platform:

   - Open a Command prompt.
   - Browse to the IKAN ALM Command Line working folder. After a standard installation, this is: *C:\alm\commandline*

     You may add this location to the PATH Environment variable, so that it becomes accessible through other working directories.

   On a Unix platform:

   - Open a terminal console.
   - Browse to the IKAN ALM Command Line working directory. After a standard installation, this is: */opt/ikan/alm/commandline*

     You may create a symbolic link to the `alm.sh` script so that it becomes accessible through other working directories.

2. Make sure that the properties file to be used is available and that the settings match the requirements.

   If not, create the properties file and/or edit the settings with a text editor.

3. Create the Level Request by entering a command in the following format:

   alm clr -propertyfile:$PROPERTYFILENAME [-options] [-parameters]

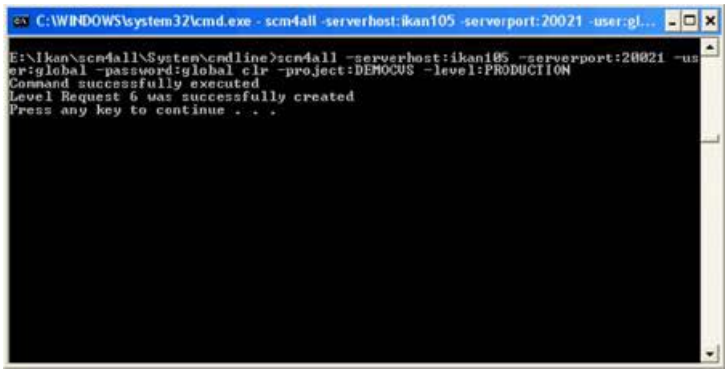
The following elements are available:

Element	Description	Mandatory
alm	This element calls the *alm.cmd* (Windows) or *alm.sh* (Linux) file.	Yes
clr	This element indicates that you want to create a Level Request.	Yes
propertyfile	This element allows selecting the properties file that must be used to create the Level Request.	Yes
options or parameters	Any option or parameter defined after the properties file **overrides** the setting in the selected properties file.	No

4. Once you have entered the complete command, press *Return*.

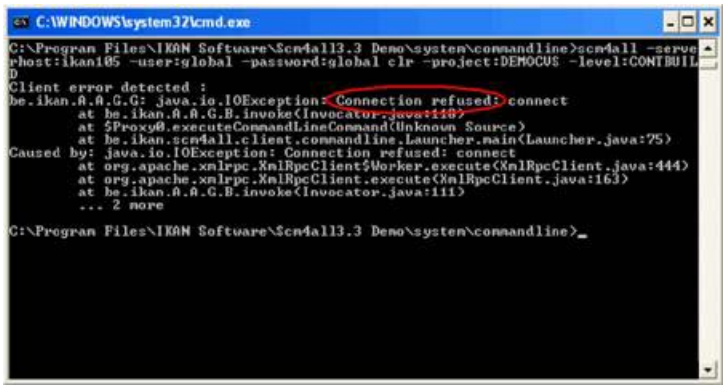
Result:

- If the Level Request is created successfully, the following screen is displayed:

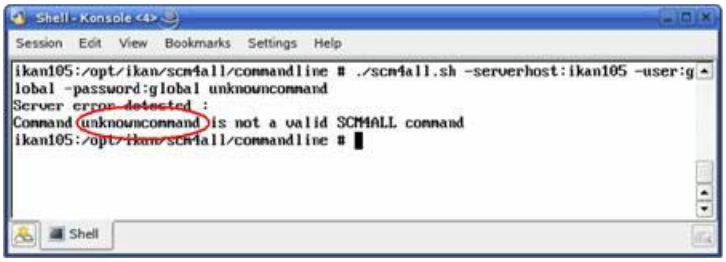


These messages only indicate that the Level Request was *created* successfully. Refer to the Level Requests Overview (page 58)in the web application to verify if the Level Request was *executed* successfully as well.

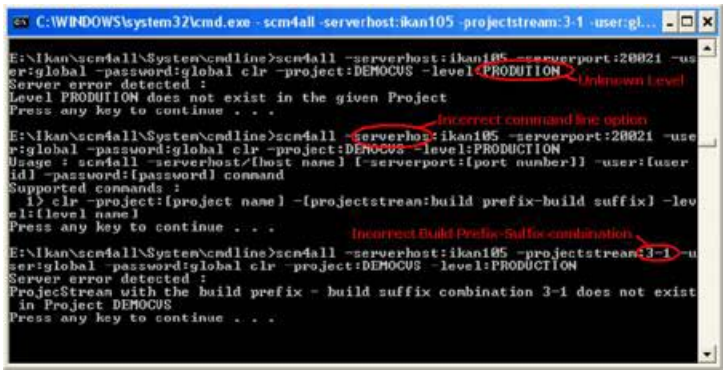
- If the Level Request cannot be created, because there is no connection with the IKAN ALM Server, the following screen is displayed:



- If the user entered an unknown command, the following screen is displayed:

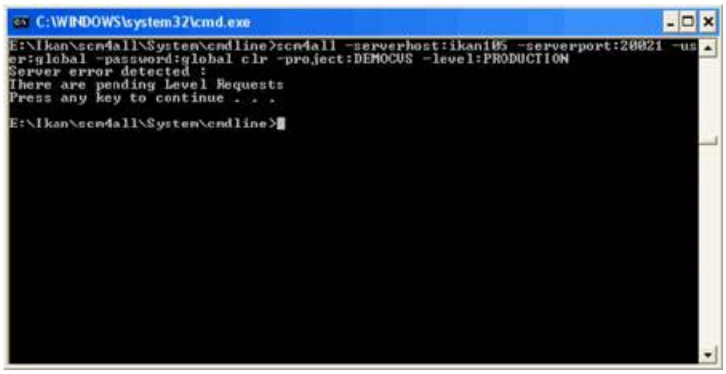


- If the user entered unknown or incorrect command options, the screen similar to the following is displayed:



Similar error messages are provided for unknown Project names and incorrect User ID/Password combinations.

- If the Level Request cannot be created because of pending Level Requests for the Level, the following screen is displayed:



Similar error messages are shown if there is no suitable Build result to be delivered, the Level is locked, the Project Stream is locked or if there is an authorization error.

Generating Reports by Specifying Options and Parameters

1. Prepare the Report generation.

   On a Windows platform:

   - Open a Command prompt.
   - Browse to the IKAN ALM Command Line working folder.

     After a standard installation, this is: *C:\alm\commandline*

     You may add this location to the PATH Environment variable, so that it becomes accessible through other working directories.

On a Unix platform:

- Open a terminal console.
- Browse to the IKAN ALM Command Line working directory.

  After a standard installation, this is: */opt/ikan/alm/commandline*

  You may create a symbolic link to the `alm.sh` script so that it becomes accessible through other working directories.

2. Enter the required Report generation instruction.

Format:

alm -serverhost:$HOSTNAME [-serverport:$PORTNUMBER] -user:$USERID -password:$PASSWORD [-secured=true|false] report PARAMETERS

The following elements are available:

Element	Description	Mandatory
alm	This element calls the *alm.cmd* (Windows) or *alm.sh* (Linux) file.	Yes
serverhost	This element allows defining the DHCP name or IP address of the host running the IKAN ALM Server.	Yes
serverport	This element allows defining the port number used to establish the connection to the IKAN ALM Server via XML-RPC. If this element is not provided, the default port number (20021) will be used.	No
user	This element allows defining a User ID with IKAN ALM access rights.	Yes
password	This element allows defining the password associated with the User ID used to access IKAN ALM.	Yes
secured	This element allows determining whether an encrypted connection must be used or not. If the IKAN ALM Server is using an encrypted connection, set the *secured* element to *true*: *-secured=true* If the IKAN ALM Server is not using an encrypted connection, set the *secured* element to *false* or omit the *secured* parameter.	No
report	This command allows generating Reports.	Yes
PARAMETERS	The Report generation parameters are explained in the following step.	Yes

3. Define the Report generation parameters.

Format:

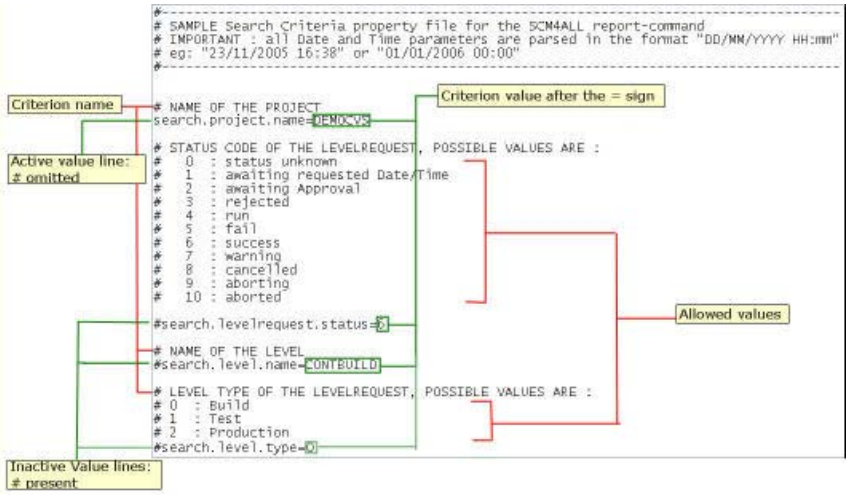
report -design:$REPORTDESIGNFILE [-dest:$DESTINATIONFILE] -format:$FORMAT [-lang:$LANGUAGE] [-max:MAXRESULT] [-filter:$SEARCHCRITERIAFILE] [-group:[$GROUPINGVALUE]] [-order:[$ORDERINGVALUE]]

Parameter	Description	Mandatory
design	This parameter allows selecting the required Jasper Reports design file (file extension is *.jrxml*). After a standard installation, the files are located in the directory *IKAN ALM_HOME/commandline/classes/reports/design*.	Yes
dest	This parameter allows defining the destination file name for the Report. Do not provide the extension, as IKAN ALM will append the format indication as extension. If this destination name is not provided, the Report will get a default name (*levelrequestoverview_[format].[format]*) and it will be saved at the default location (*IKAN ALM_HOME/commandline/classes/reports/generated_reports*).	No
format	This parameter allows defining the Report format. The following formats are allowed: • pdf • htm • xml • csv • rtf • txt • xls	Yes
lang	This parameter allows defining the Report language. The following values are allowed: • en (English) • fr (French) • de (German) If the language parameter is omitted, the Report will be generated in English.	No
max	This parameter allows defining the maximum number of Level Requests to be included in the Report. If more Level Requests are available than the defined maximum, only the most recent Level Requests will be included in the Report.	No
filter	This parameter allows selecting a property file containing search criteria. Only Level Requests matching all defined criteria will be included in the Report. After a standard installation, one such property file, called *search.properties* is available in the Command Line installation directory. You can edit this standard file so that the settings match your requirements. See the description in the following step. You can also create any number of specific search criteria properties files by copying the standard file, editing the copies and saving them under logical names for later usage.	No

Parameter	Description	Mandatory
group	This parameter allows defining how the reported Level Requests should be grouped together. The following values are allowed: • projectname : group by Project Name • levelname : group by Level Name If this parameter is omitted or left empty, no grouping of Level Requests will occur.	No
order	This parameter allows defining how the reported Level Requests should be ordered. The following values are allowed: • asc : order ascending (=default) • desc : order descending	No

4. If required, edit the search criteria properties file in a text editor.

The file has the following structure:



The following selection criteria are available:

Criteria	Description
Project name	Property: `search.project.name` Enter a Project name, if you want to limit the Report to Level Requests of that Project.
Package name	Property: `search.package.name` Enter a Package name, if you want to restrict the Report to Level Requests for that Package.
Search hidden packages	Property: `search.package.hidden` Enter one of the possible values, if you want to limit the Report to Level Requests for hidden Packages: • yes = show Level Requests for hidden (archived) Packages or Level Requests having no Packages associated • no = show Level Requests for visible (non-archived) Packages or Level Requests having no Packages associated • all = no restriction regarding the Package archived status
Level Request Status code	Property: `search.levelrequest.status` Enter one of the possible status codes, if you want to limit the Report to Level Requests with that status: • 0 = unknown • 1 = awaiting requested date/time • 2 = awaiting Approval • 3 = rejected • 4 = run • 5 = fail • 6 = success • 7 = warning • 8 = cancelled • 9 = aborting • 10 = aborted
Level Name	Property: `search.level.name` Enter the name of the Level, if you want to limit the Report to Level Requests for that Level.
Level Type	Property: `search.level.name` Enter one of the possible Level Types, if you want to limit the Report to Level Requests pertaining to that Level Type: • 0 = Build • 1 = Test • 2 = Production

Level Request Start Time interval	Properties:
	`search.levelrequest.startdatetime.from`
	`search.levelrequest.startdatetime.to`
	Enter the start and end timestamp of the *Level Request Start Time* interval, if you want to limit the Report to Level Requests having started within this interval.

Level Request End Time interval	Properties:
	`search.levelrequest.enddatetime.from`
	`search.levelrequest.enddatetime.to`
	Enter the start and end timestamp of the *Level Request End Time* interval, if you want to limit the Report to Level Requests having ended within this interval.

Criteria	Description

Level Request Request Time interval	Properties:
	`search.levelrequest.requestdatetime.from`
	`search.levelrequest.requestdatetime.to`
	Enter the start and end timestamp of the *Level Request Request Time* interval, if you want to limit the Report to Level Requests requested within this interval.

Requester Name	Property: `search.username` Enter the name of the Requester, if you want to limit the Report to Level Requests requested by a specific User.
Level Request Action Type	Property: `search.levelrequest.actiontype` Enter one of the possible Level Request Action Types, if you want to limit the Report to Level Requests with this Action Type: • 0 = Build initiated by Scheduler • 1 = Force Build • 2 = Request Build • 3 = Deliver Build • 4 = Rollback Build • 5 = Dependency Build • 6 = Redeliver Build
Level Request Type	Property: `search.levelrequest.type` Enter one of the possible Level Request Types, if you want to limit the Report to Level Requests with this Type: • 0 = Build based on latest code • 1 = Builds based on tagged code • 2 = Builds and Deploys on latest code • 3 = Builds and Deploys on tagged code • 4 = Deploys of archived Build • 5 = No Builds or Deploys
VCR Tag	Property: `search.vcrtag` Enter a VCR tag, if you want to limit the Report to Level Requests pertaining to that VCR Tag.
Project Stream Status	Property: `search.projectstream.status` Enter one of the possible Project Stream Status indications, if you want to limit the Report to Level Requests with this Project Stream Status: • 0 = under construction • 1 = planning • 2 = development • 3 = testing • 4 = stable • 5 = general available • 6 = frozen • 7 = closed
Project Stream Prefix	Property: `search.projectstream.buildprefix` Enter a Project Stream Prefix, if you want to limit the Report to Level Requests pertaining to that Project Stream Prefix.
Project Stream Build Suffix	Property: `search.projectstream.buildsuffix` Enter a Project Stream Build Suffix, if you want to limit the Report to Level Requests pertaining to that Project Stream Build Suffix.

Criteria	Description
Search Hidden Project Stream	Property: `search.projectstream.hidden` Enter one of the possible values, if you want to limit the Report to Level Requests for hidden Project Streams: • yes = show only hidden project streams • no = do not show hidden project streams (default value) • all = show all project streams

Note: Do not forget to remove the # sign in order to activate a search criterion.

5. Once you have entered the complete command, press *Return*.

   The report will be generated.

Generating Reports Using a Properties File

It is possible to save frequently used settings in a Properties file, so that you do not have to enter the complete Report generation parameters. After a standard installation, one such properties file, called *report.properties* is available in the Command Line installation folder.

Open the file in a text editor to display its content:

```
#-------------------------------------------------------------
# SAMPLE property file for the SCM4ALL report-command
# This file should be in the same folder as the scm4all executable
# IMPORTANT : for all paths : use '/' or '\\'
#-------------------------------------------------------------

#-----------------
# GLOBAL OPTIONS :
#-----------------

# HOSTNAME OF THE SCM4ALL SERVER
serverhost=localhost

# PORT NUMBER OF THE SCM4ALL SERVER
#serverport=20021

# USER ID TO CONNECT TO THE SCM4ALL SERVER
user=global

# PASSWORD TO CONNECT TO THE SCM4ALL SERVER
password=global

# USE ENCRYPTED CONNECTION OR NOT (TRUE|FALSE)
secured=false

#--------------------------------------
# SPECIFIC OPTIONS FOR report COMMAND :
#--------------------------------------

# JASPERREPORTS DESIGN FILE (.JRXML)
design=reports/design/levelrequestoverview_pdf.jrxml

# FORMAT OF THE REPORT. MUST BE ANY OF THE FOLLOWING VALUES : PDF,HTM,XML,CSV,RTF,TXT OR XLS
format=pdf

# LANGUAGE OF THE REPORT. CURRENTLY SUPPORTING EN,FR AND DE
#lang=en

# MAXIMUM NUMBER OF LEVELREQUESTS TO REPORT
#max=5

# DESTINATION FILE NAME OF THE GENERATED REPORT. SPECIFY FILENAME WITHOUT EXTENSION AS SCM4ALL
WILL ALWAYS APPEND THE FORMAT AS EXTENSION
dest=./classes/reports/generated_reports/levelrequestoverview_pdf
```

You can edit this standard file so that the settings match your requirements. Refer to the section Creating Level Requests by Specifying Options and Parameters (page 475)for a description of the options and parameters. You can also create any number of specific properties files by copying the standard file, editing the copies and saving them under logical names for later usage.

1. Prepare the Report generation.

   On a Windows platform:

   - Open a Command prompt.
   - Browse to the IKAN ALM Command Line working folder. After a standard installation, this is: *C:\alm\commandline*

You may add this location to the PATH Environment variable, so that it becomes accessible through other working directories.

On a Unix platform:

- Open a terminal console.
- Browse to the IKAN ALM Command Line working directory. After a standard installation, this is: */opt/ikan/alm/commandline*

  You may create a symbolic link to the `alm.sh` script so that it becomes accessible through other working directories.

2. Make sure that the properties file to be used is available and that the settings match the requirements.

   If not, create the properties file and/or edit the settings with a text editor.

3. Generate the Report by entering a command in the following format:

   alm report -propertyfile:[$PROPERTYFILENAME] [-options] [-parameters]

   The following elements are available:

Element	Description	Mandatory
alm	This element calls the *alm.cmd* (Windows) or *alm.sh* (Linux) file.	Yes
report	This element indicates you want to generate a Report.	Yes
propertyfile	This element allows selecting the properties file that must be used to generate the Report.	Yes
options or parameters	Any option or parameter defined after the properties file **overrides** the setting in the selected properties file.	No

4. Once you have entered the complete command, press *Return*.

   The Report will be generated.

36.2. Integrating IKAN ALM in an External IDE

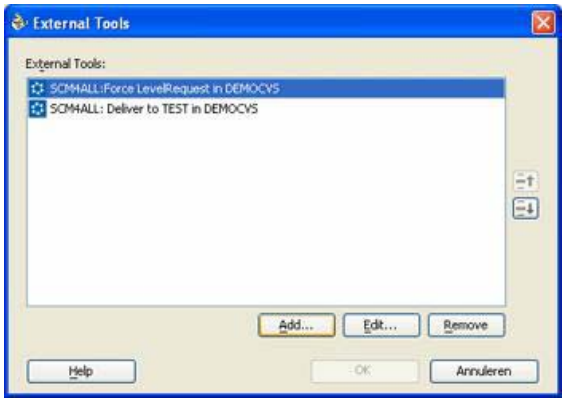
Refer to the following procedures for detailed information:

- Integrating IKAN ALM in JDeveloper (page 490)
- Integrating IKAN ALM in Eclipse (page 492)

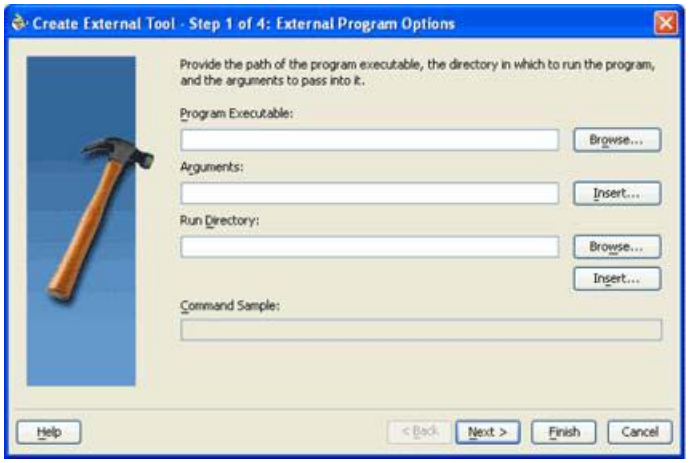
Integrating IKAN ALM in JDeveloper

This procedure describes how to set up IKAN ALM as an external tool in JDeveloper, so that you can create Level Requests (Forced Build, Requested Build or Deliver Build) from within this IDE.

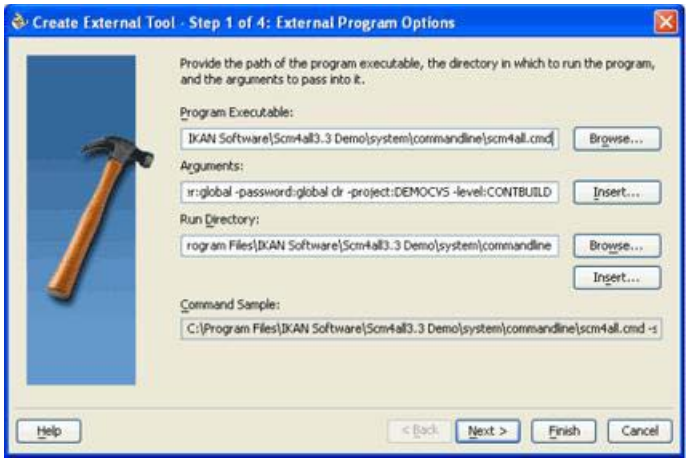
1. On the JDeveloper main menu, select *Tools* | *External Tools...*
   The following dialog is displayed:



2. Click *Add...*
   The following dialog is displayed:



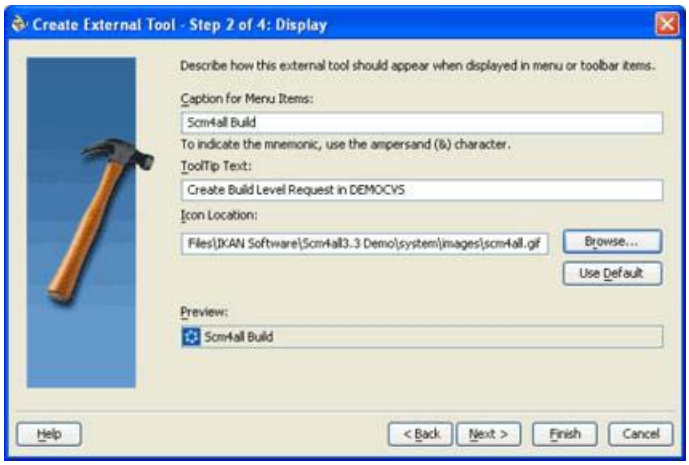
3. Provide the correct parameters to create a Level Request on the wanted level:



In the example above, the options are given to create a Level Request on the *CONTBUILD* Level in the *DEMOCVS* Project.

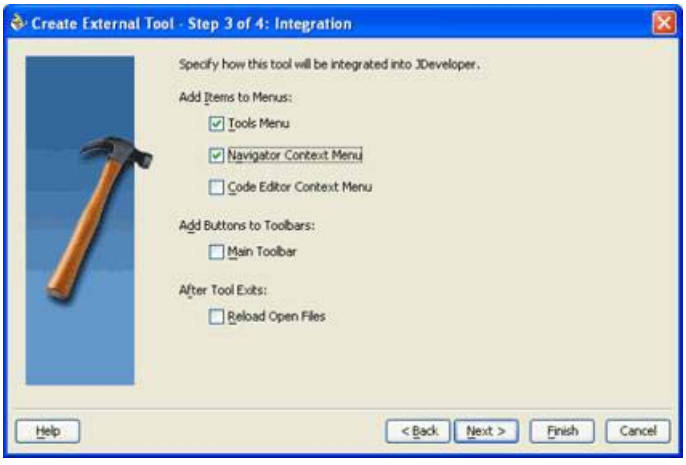
4. Click *Next*.

The following dialog is displayed:



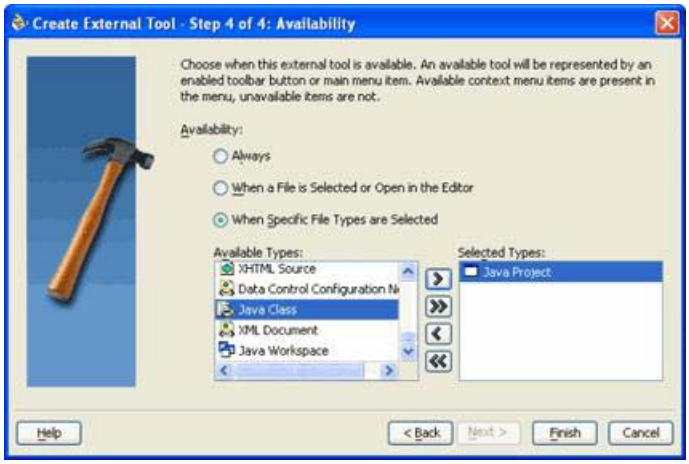
5. Provide the display properties and click *Next*.

The following dialog is displayed:



6. Specify where you want to integrate this new tool and click *Next*.

   The following dialog is displayed:



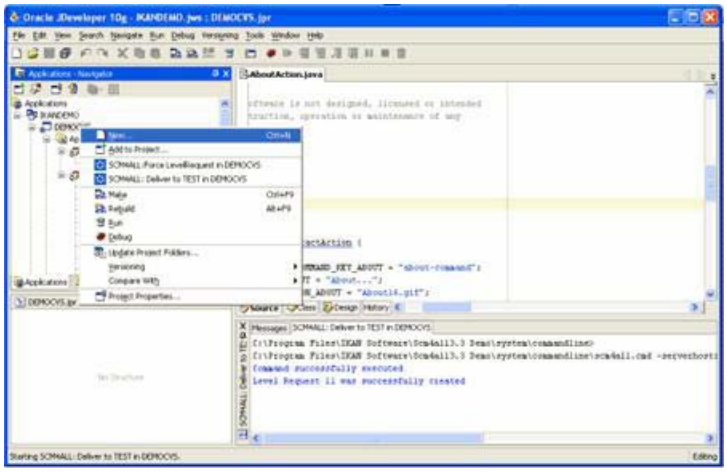
7. Specify the availability of the external tool.

   In the example above, it was specified that the external tool will be integrated in the shortcut menu, if a java project is selected.

8. Select *Finish*.

   Result:

   You can now create Level Requests when selecting a java project in JDeveloper. The output of the IKAN ALM Command Line will be visible in a message box:



Integrating IKAN ALM in Eclipse

This procedure describes how to set up IKAN ALM as an external tool in Eclipse, so that you can create Level Requests (Forced Build, Requested Build or Deliver Build) from within this IDE.

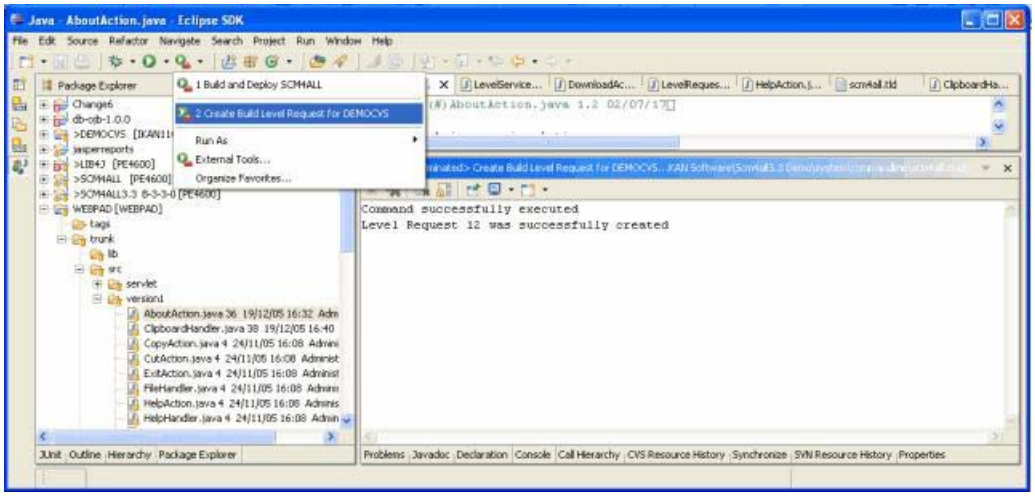
1. On the Eclipse main menu, select *Run | External Tools Configuration...*
   The following dialog is displayed:

2. Select the *New launch configuration* icon.
   The following dialog is displayed:

3. Provide the correct parameters to create a Level Request on the wanted level:
   In the example above, the options are given to create a Level Request on the *CONTBUILD* Level in the *DEMOCVS* Project.

4. Click *Run*.
   The external tools will be tested and saved.

The output from the command is visible in an Eclipse console:



36.3. Automating the Creation of Level Requests Using an ANT script

This section provides a sample ANT script that

- Creates an Level Request using the Command Line interface
- Reports the status of this action

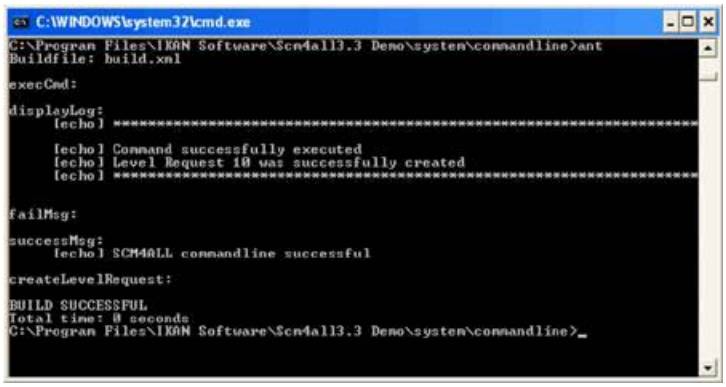
If you want to use this ANT Script yourself, customize it by:

- Adapting the values for the IKAN ALM Command Line options (values for serverhost, user, password, etc.)
- Saving it as *build.xml* in the IKAN ALM_COMMANDLINE root directory
- Launching it via the standard ANT command.

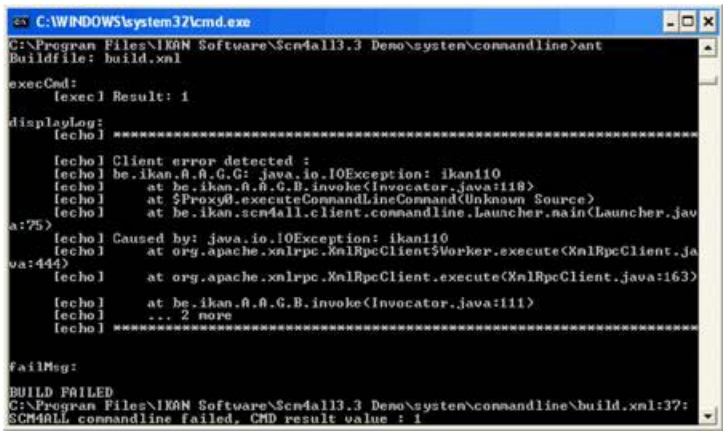
Sample script:

```
<project name="SCM4ALL Commandline" default="createLevelRequest" basedir=".">
  <description>launch scm4all commandline via an ANT script</description>

  <!-- set  properties for SCM4ALL commandline options -->
  <property name="serverhost" value="ikan110"/>
  <property name="user" value="global"/>
  <property name="password" value="global"/>
  <property name="command" value="clr"/>
  <property name="project" value="DEMOCVS"/>
  <property name="level" value="CONTBUILD"/>

  <target name="createLevelRequest" depends="execCmd,displayLog,failMsg,successMsg"/>

  <target name="execCmd">
    <exec dir="."executable="scm4all.cmd"timeout="6000"
failonerror="false" resultproperty="scm4allResult" outputproperty="scm4allLog">
      <arg line="-serverhost:${serverhost} -user:${user} -password:${password} ${command}
        project:${project} -level:${level}"/>
    </exec>
    <condition property="scm4allSuccess">
      <equals arg1="${scm4allResult}" arg2="0"/>
    </condition>
    <condition property="scm4allFail">
      <not>
        <equals arg1="${scm4allResult}" arg2="0"/>
      </not>
    </condition>
  </target>

  <target name="displayLog">
    <echo message="*********************************************************************************"/>
    <echo message="${scm4allLog}"/>
    <echo message="*********************************************************************************"/>
  </target>

  <target name="failMsg" if="scm4allFail">
    <fail message="SCM4ALL commandline failed, CMD result value : ${scm4allResult}"/>
  </target>

  <target name="successMsg" if="scm4allSuccess">
    <echo message="SCM4ALL commandline successful"/>
  </target>
</project>
```

Output on success:



Output on failure:



APPENDICES

APPENDIX A

History and Event Logging

All create, update and delete operations of Project and Global Administration objects and components are logged in the IKAN ALM Database.

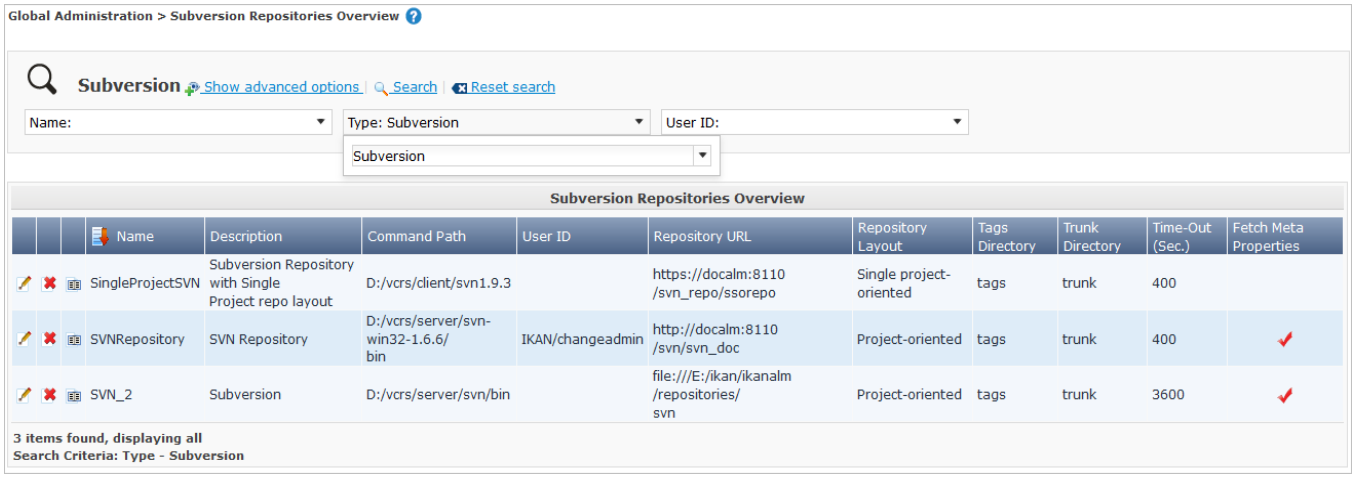
To display the History Log of those operations, you can use the icon (or *History* link) which is available on all *Overview* screens as well as on some of the edit screens, like the *System Settings* screen or the *Edit Level* screen.

Delete operations that lead to cascaded deletions are only logged for the aggregate component. For example: when deleting a Project, only the deletion of the Project is logged, and not the deletion of the associated Project Streams, Lifecycles, Levels, Build and Deploy Environments, ….

1. Switch to the appropriate *Overview* screen.

   For more information on how to switch to an *Overview* screen, refer to the specific section.

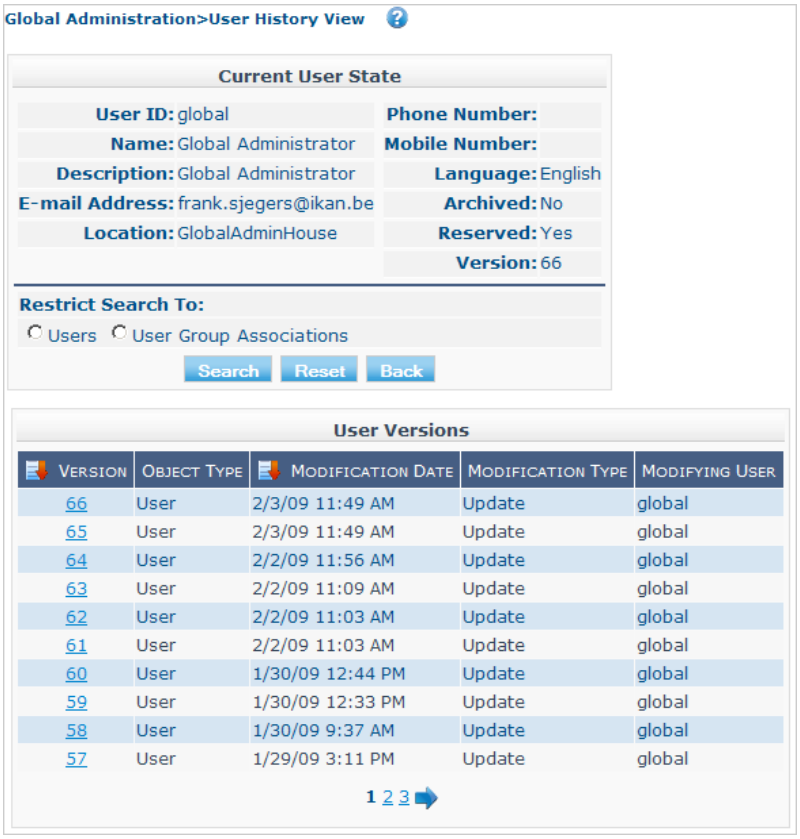
   A screen similar to this one, will appear.



2. Click the *History* link, underneath the *Overview*, to display the *History View* screen.

Note: As there is no *Overview* screen for system settings, the *History* link for the system settings is available on the *System Settings* screen itself.

A screen similar to the one below will be displayed:



It contains the following sections:

- The *Current State* Panel displaying the current state of the Object
- A subpanel allowing you to limit the number of versions to a specific Object Type.

  This subpanel is only available for object types having an association (like Users or User Groups in System Settings, Levels and Lifecycles in Project Administration) or for fine grained object types which history is shown in their aggregate object type (like Build Parameters, which history are shown with the Build Environment).
- The *Versions* Panel

3. Restrict the list of items displayed on the *Versions* panel.

For certain objects, a subpanel to restrict the number of displayed items is available.

Specify the required selection criteria and click the *Search* button. You can also click the *Reset* button to clear the Search fields.

The following table lists the different selection options per object type:

Object Type	Available Selection Criteria
Build Environment	Build Environments Build Environment Phases Build Parameters
Deploy Environment	Deploy Environments Deploy Environment Phases Deploy Parameters

Object Type	Available Selection Criteria
Level	Levels Level Environment Phases Lifecycle Associations Pre-Approval User Group Associations Post-Approval User Group Associations
Lifecycle	Lifecycles Lifecycle Associations
Machines	Machines Machine Parameters
Packages	Packages Modified Files
Package Build Groups	Package Build Groups Package Build Group Associations
Phases	Phases Phase Parameters
Project Streams	Project Streams Master Dependencies Child Dependencies
User	Users User Group Associations
User Group	User Groups User Associations

4. Verify the information on the *Versions* panel.

Underneath the *Current State* panel, the *Versions* panel displays the list of modifications that have been made to the object, or to associated objects. The first non-association version in the list should always have the same state as the current state of the object.

Note: It is possible that the *Versions* panel is empty or does not list all previous versions. Objects created during the installation process will not have a Create Version. Objects created or modified before the SCM4ALL 4.3 Release will not have an event log in the IKAN ALM database.

The following information fields are available:

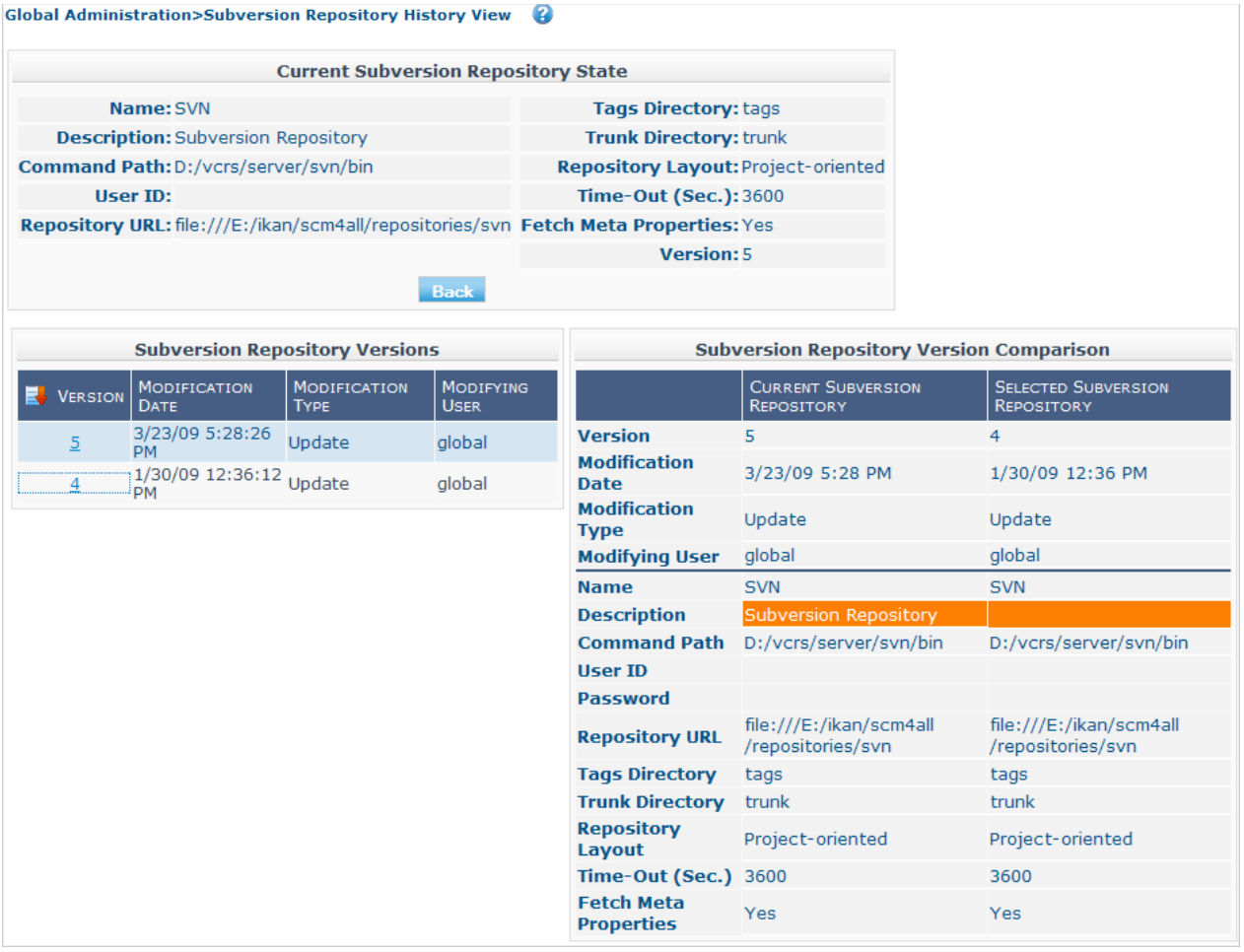
Field	Description
Version	The version the object had at the moment the modification was saved.
[Object OID]	The OID number of the object (only visible for Levels, Build Environments and Deploy Environments).
[Name]	The name of the Environment or the Parameter (only visible for Build and Deploy Environments and their parameters).
[Object type]	The type of the object that was modified (only visible for object types that have an association).
[Level Name]	The name of the Level (only visible for Lifecycles).
Modification Date	The date/time the modification was saved.
Modification Type	The modification type can be one of the following: • *Create* The object was created • *Update* The object was updated • *Add* (for an association type) The association has been added • *Remove* (for an association type) The association has been removed • *Modified Files* (for a Package) Files have been added to or removed from the Package, or their revision has been changed.
Modifying User	The User ID of the User who performed the modification.

Note: Columns marked with the icon can be sorted alphabetically (ascending or descending).

5. Compare a specific version with the current version of the object.

   Select the version you want to compare by clicking its number on the *Versions* panel.

The *Version Comparison* panel is added to the right of the *Versions* panel.



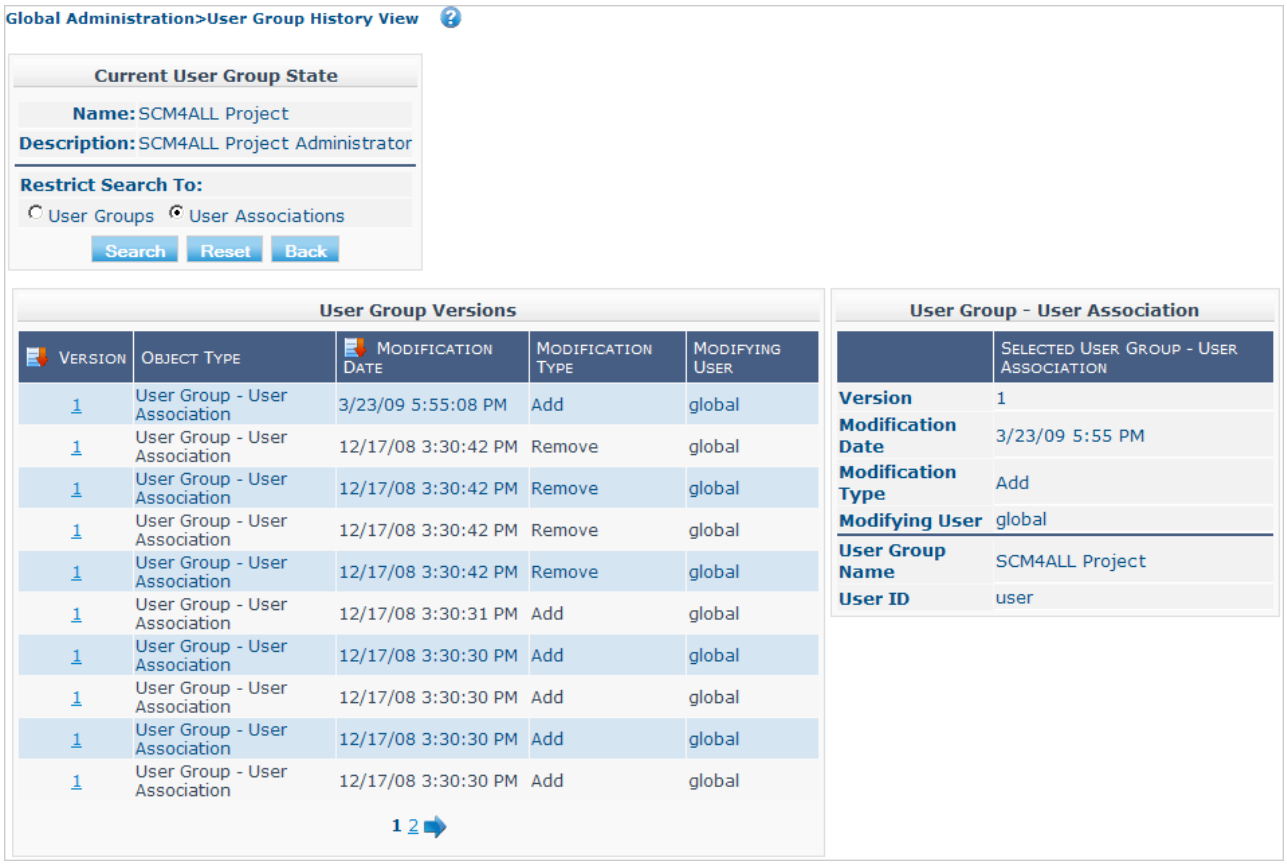
The *Version Comparison* panel displays the differences between the current version of the object (shown in the "Current" column) and the selected version of the object (shown in the "Selected" column)

- The upper part of the *Version Comparison* panel displays the Version, Modification Date, Modification Type and Modifying User of the two versions that are being compared
- The lower part displays the object-specific values of the two versions that are being compared, highlighting the fields where there are differences.

Note: The Version Comparison panel is not displayed for User Group - User Associations nor for Packages. See the example screens below.

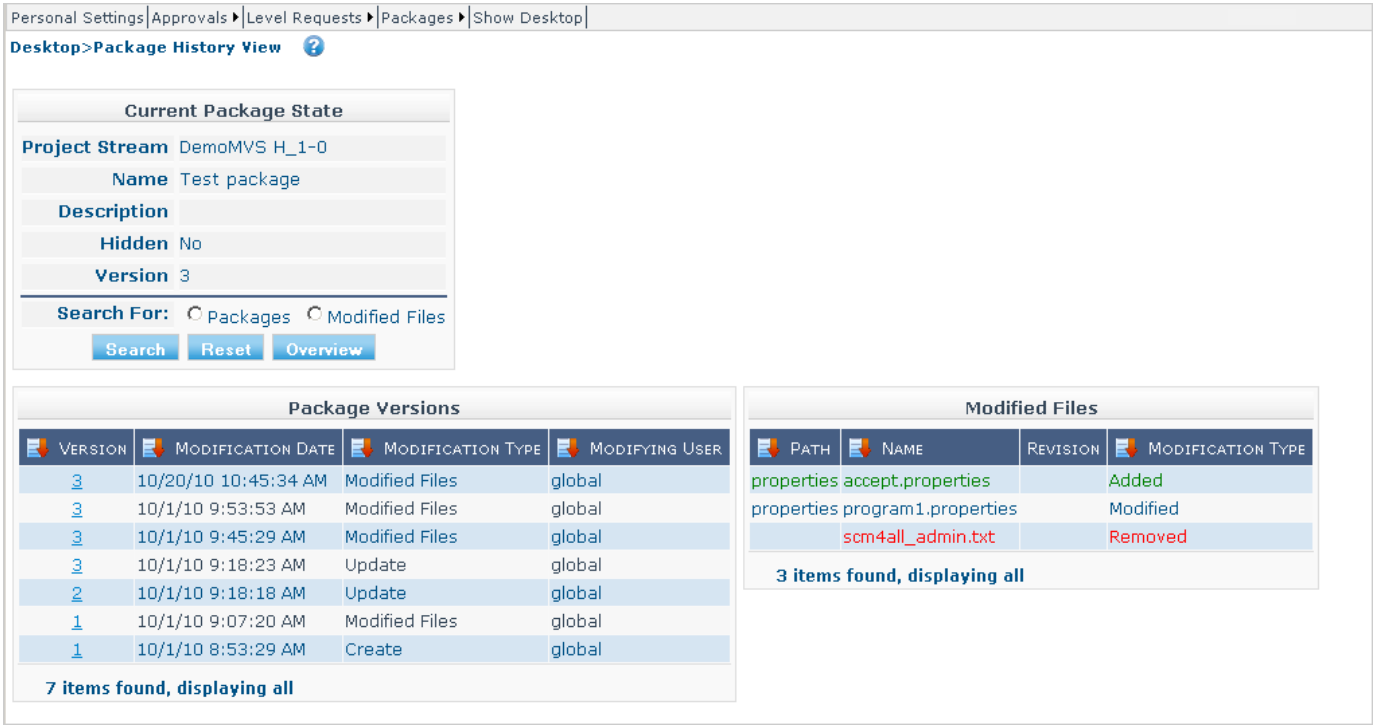
Example 1 - User Group - User Association

The panel is replaced with the *User Group – User Association* panel.



Example 2 - Modified Files (Packages)

The panel is replaced with the *Modified Files* panel, displaying which files have been added (green), modified (black) or deleted (red).



APPENDIX B

Predefined Level Parameters

The following table provides an overview of the predefined Level Parameters:

Parameter	Description
alm.levelRequest.actionType	This parameter provides the type of the Level Request: The possible types are: • Scheduled Build • Forced Build • Requested Build • Deliver Build • Rollback Build • Dependency Build
alm.levelRequest.buildNumber	This parameter provides the Build Number.
alm.levelRequest.buildType	This parameter provides the Build Type of the Level Request. The possible types are: • Full Build • Partial Build • Production-based Partial Build • Tag-based Build
alm.levelRequest.levelName	This parameter provides the name of the Level where the Level Request is executed.
alm.levelRequest.oid	This parameter provides the unique number of the Level Request.
alm.levelRequest.requester	Only available if there is a Requesting User. This parameter provides the name of the Requester.
alm.levelRequest.rootLocation	A directory under the Work Copy Location on the IKAN ALM Server where the "Retrieve Code" Phase places the checked out sources. The Work Copy Location is expanded with the Level Request OID (e.g., `E:\Ikan\alm\system\system\workCopy\1170`)
alm.levelRequest.startDateTime	This parameter provides the start date and time of the current Level Request. The format is `date.time`. The date is in the format used on the server where IKAN ALM runs: • `YYYY/MM/DD` for Europe • `MM/DD/YYYY` for the USA

Parameter	Description
alm.levelRequest.vcrTag	This parameter provides the tag of the Level Request. The tag is defined as `projectstream_prefix_b+buildoid_suffix` where: `projectstream` is replaced by `H` for the Head stream and `B` for a Branch stream.
alm.package.oid	Only available in case of a Package. This parameter provides the unique number of the Package.
alm.project.buildToolTypeName	This parameter provides the Build Tool type used: • ANT • GRADLE • NANT • MAVEN2
alm.project.deployToolTypeName	This parameter provides the Deploy Tool type used: • ANT • GRADLE • NANT • MAVEN2
alm.project.description	This parameter provides the description of the IKAN ALM Project.
alm.project.name	This parameter provides the IKAN ALM Project name. This name can be different from the Project name as known within the used VCR (see alm.project.vcrProjectName).
alm.package.name	This parameter is only available for Package-based Projects. It provides the name of the Package associated with the Level Request.
alm.project.vcrName	This parameter provides the name of the VCR (as defined in the Global Administration) to which this Project is linked.
alm.project.vcrProjectName	This parameter provides the name of the Project as known within the VCR. This name can be different from the IKAN ALM Project name (see alm.project.name).
alm.projectStream.buildPrefix	This parameter provides the Build Prefix defined for the Project Stream.
alm.projectStream.buildSuffix	This parameter provides the Build Suffix defined for the Project Stream.
alm.projectStream.description	This parameter provides the Description of the Project Stream.
alm.projectStream.type	This parameter provides the Project Stream Type: • H = Head • B = Branch
alm.projectStream.vcrBranchId	This parameter provides the Branch ID in the VCR defined in the IKAN ALM Project Stream (only for Branches).
source	This parameter provides the Source Location which is derived from the System Settings Local Work Copy Location. This location will be expanded with the Level Request OID, "workcopy", the Project Name and the Project VCR Name. (e.g., `E:\Ikan\alm\system\system\workCopy\1170\workcopy\Webpad\webpad`)

APPENDIX C

Predefined Build Parameters

The following table provides an overview of the predefined Build Parameters:

Parameter	Description
alm.build.environmentName	This parameter provides the name of the Build Environment where the Build Script is running.
alm.build.filename	This parameter provides the name of the Build File that will be stored in the Build Archive Location. See System Settings on page 255.
alm.build.machineName	This parameter provides the name of the Machine where the Build Script is running.
alm.build.number	This parameter provides the unique Build Number. This is a sequential number that is incremented by IKAN ALM when a Build Level Request is created.
alm.build.oid	This parameter provides the unique identifier of the current Build as used to store the build information in the database.
alm.build.startDateTime	This parameter provides the start date and time for the Build (i.e., the timestamp indicating the start of the Build Script). Format is date.time. The date is in the format used by the server hosting IKAN ALM: • *YYYY/MM/DD* for Europe • *MM/DD/YYYY* for the USA
alm.levelRequest.actionType	This parameter provides the Level Request type. Possible types: • Scheduled Build • Forced Build • Requested Build • Deliver Build • Rollback Build • Dependency Build
alm.levelRequest.buildType	This parameter provides the Build Type of this Level Request. Possible types: • Full Build • Partial Build • Production-based Partial Build • Tag-based Build
alm.levelRequest.levelName	This parameter provides the name of the Level where the Build is executed.
alm.levelRequest.oid	This parameter provides the unique identifier of the current Level Request.

Parameter	Description
alm.levelRequest.requester	Only available if there is a Requester User. This parameter provides the name of the Requester.
alm.levelRequest.startDateTime	This parameter provides the start date and time of the current Level Request. Format is *date.time*. The date is in the format used by the server hosting IKAN ALM: • *YYYY/MM/DD* for Europe • *MM/DD/YYYY* for the USA
alm.levelRequest.vcrTag	This parameter provides the Tag used during the current build to tag the sources in the VCR.
alm.package.oid	Only available in case of a Package. This parameter provides the unique number of the Package.
alm.previous.build.filename	This parameter provides the name of the last successfully zipped Build File, i.e., the result of the previous Build: e.g., *DEMOCVS_B_b_1_test_b2_CONTBUILD.zip* This is the last build file for project DEMOCVS.
alm.previous.build.number	This parameter provides the Build Number of the previous successful Build on this Level.
alm.previous.levelRequest.vcrTag	This parameter provides the VCR tag of the previous Level Request on this Level.
alm.project.buildToolTypeName	This parameter provides the Build Tool type. Possible values: • ANT • GRADLE • NANT • MAVEN2
alm.project.deployToolTypeName	This parameter provides the Deploy Tool type. Possible values: • ANT • GRADLE • NANT • MAVEN2
alm.project.description	This parameter provides the description of the IKAN ALM Project.
alm.project.name	This parameter provides the IKAN ALM Project name. This name can be different from the Project name in the VCR.
alm.package.name	This parameter is only available in Package-based Projects. It provides the name of the Package associated with the Level Request.
alm.project.vcrName	This parameter provides the name of the VCR (as defined in the Global Administration) to which this Project is linked.
alm.project.vcrProjectName	This parameter provides the name of the Project as defined in the VCR. This name can be different from the IKAN ALM Project name
alm.projectStream.buildPrefix	This parameter provides the Build Prefix defined in the Project Stream definition.
alm.projectStream.buildSuffix	This parameter provides the Build Suffix as defined for the Project Stream (no entry for a Head Project Stream).

Parameter	Description
alm.projectStream.description	This parameter provides the description defined for the Project Stream.
alm.projectStream.type	This parameter provides Project Stream type of the project: • H=Head type • B=Branch type
alm.projectStream.vcrBranchId	This parameter provides the Branch ID in the VCR, defined in the IKAN ALM Project Stream, in case of a Branch Project Stream.
source	This parameter provides the name of the Source Location as defined for the current Build Environment. The name will be expanded with the number of the *alm.build.oid* and the name of the project as known within the VCR (*alm.project.vcrProjectName*)
sourceroot	This parameter provides the name of the source location as defined for the current Build Environment. The name will be expanded with the number of the *alm.build.oid.* This property will only be set if the current project depends on another project. The named directory in this property will contain all the sources from the parent project.
target	This parameter provides the name of the target location as defined in the current build environment definition. The name will be expanded with the number of the *alm.build.oid.*

APPENDIX D

Predefined Deploy Parameters

The following table provides an overview of the predefined Deploy Parameters:

Parameter	Description
alm.build.environmentName	This parameter provides the name of the Build Environment where the Build Result that is deployed, was built on.
alm.build.filename	This parameter provides the name of the Build file that will be unpacked in the Source Location and deployed. See System Settings on page 255.
alm.build.machineName	This parameter provides the name of the Machine where the Build Result was built on.
alm.build.number	This parameter provides the number of the Build that will be deployed. The Build Number is a sequential number that is incremented by IKAN ALM when a Build Level Request is created.
alm.build.oid	This parameter provides the unique ID of the Build that will be deployed.
alm.build.startDateTime	This parameter provides the start date and time for the Build, that will be deployed. This is the timestamp indicating the start of the Build Script. The format is *date.time*. The date is in the format of the server where the IKAN ALM runs: • *YYYY/MM/DD* for Europe • *MM/DD/YYYY* for the USA
alm.deploy.environmentName	This parameter provides the name of the Deploy Environment where this Deploy will be executed.
alm.deploy.machineName	This parameter provides the name of the Machine where the Deploy will be executed.
alm.deploy.oid	This parameter provides the unique OID of the Deploy.
alm.deploy.startDateTime	This parameter provides start date and time for the Deploy (i.e., when the Deploy Script started). Format is *date.time*. The date is in the format of the server where IKAN ALM runs: • *YYYY/MM/DD* for Europe • *MM/DD/YYYY* for the USA

Parameter	Description
alm.levelRequest.actionType	This parameter provides the type of the Level Request: • Scheduled Build • Forced Build • Requested Build • Deliver Build • Rollback Build • Dependency Build
alm.levelRequest.levelName	This parameter provides the name of the Level where the Level Request is executed.
alm.levelRequest.oid	This parameter provides the unique number of this Level Request.
alm.levelRequest.requester	Only available if there is a Requester User. This parameter provides the name of the Requester.
alm.levelRequest.startDateTime	This parameter provides the start date and time of the current Level Request. Format is *date.time*. The date is in the format used on the server where IKAN ALM runs: • *YYYY/MM/DD* for Europe • *MM/DD/YYYY* for the USA
alm.levelRequest.vcrTag	This Parameter provides the tag of the Level Request. The tag is defined as *projectstream_prefix_b+buildoid_suffix* where: *projectstream* is replaced by *H* for the Head stream and *B* for a Branch stream.
alm.package.oid	Only available in case of a Package. This parameter provides the unique number of the Package.
alm.previous.build.filename	This parameter provides the name of the previous Build Result that was deployed on this Level. Example: DEMOCVS_B_b_1_test_b2_CONTBUILD.zip
alm.previous.build.number	This parameter provides the number of the Previous Build Result that was deployed on this Level.
alm.previous.levelRequest.vcrTag	This parameter provides the Tag of the previous successful Level Request on this Level.
alm.project.buildToolTypeName	This parameter provides the Build Tool type used: • ANT • GRADLE • NANT • MAVEN2
alm.project.deployToolTypeName	This parameter provides the Deploy Tool type used: • ANT • GRADLE • NANT • MAVEN2
alm.project.description	This parameter provides the description of the IKAN ALM Project.
alm.project.name	This Parameter provides the IKAN ALM Project name. This name can be different from the Project name as known within the used VCR (see alm.project.vcrProjectName)

Parameter	Description
alm.package.name	This parameter is only available in Package-based Projects. It provides the name of the Package associated with the Level Request.
alm.project.vcrName	This parameter provides the name of the VCR (as defined in the Global Administration) to which this Project is linked.
alm.project.vcrProjectName	This parameter provides the name of the Project as known within the VCR. This name can be different from the IKAN ALM Project name (see alm.project.name).
alm.projectStream.buildPrefix	This parameter provides the Build Prefix defined for the Project Stream.
alm.projectStream.buildSuffix	This parameter provides the Build Suffix defined for the Project Stream.
alm.projectStream.description	This parameter provides the Description of the Project Stream.
alm.projectStream.type	This parameter provides the Project Stream Type: • H = Head • B = Branch
alm.projectStream.vcrBranchId	This Parameter provides the Branch ID in the VCR defined in the IKAN ALM Project Stream (only for Branches).
source	This parameter provides the Source Location as defined in the current Deploy Environment definition. The name will be expanded with the number of the alm.deploy.oid to create the complete working directory.
target	This parameter provides the name of the Target location as defined in the current Deploy Environment definition, the target of the deploy action.

APPENDIX E

Subversion - General Information

This appendix provides general information on the connection and definition of the Subversion Repository, including some concrete examples and a section on Meta Properties.

- Repository Connection (page 511)
- Repository Definition (page 511)
- Fetching Meta Properties (page 517)

E.1. Repository Connection

IKAN ALM connects with the repository using the SVN client which must be installed on the IKAN ALM Server machine. The Command Path refers to the path where this SVN client is located.

A Subversion Repository is identified via its Repository URL, e.g., *file:///path/to/repos* or *svn://servername* or *http://my.domain/repos*. For more information, see the on-line book about Subversion "Version Control with Subversion, Chapter 2 - Basic Concepts" at http://svnbook.red-bean.com/en/1.7/svn.basic.version-control-basics.html. In chapter 7 of this book you will find a table about repository URLs and a topic about the Subversion Repository.

In most cases a User ID and Password must be provided, unless the repository is not protected (e.g., for the *svn* protocol, *anon-access* = *write* in the `svnserver.conf file`) or unless the credentials of the user running the IKAN ALM web server or application server process are stored in this user's private runtime config area (*~/.subversion/auth/* or *%APPDATA%/Subversion/Auth/*). See also http://svnbook.red-bean.com/en/1.7/svn.serverconfig.netmodel.html#svn.serverconfig.netmodel.credcache.

E.2. Repository Definition

Specific fields in the Subversion definition are necessary to support the flexible setup of a Subversion Repository Layout (see http://svnbook.red-bean.com/en/1.2/svn.reposadmin.projects.html#svn.reposadmin.projects.chooselayout).

This sections provides the following information:

- Specific Fields (page 512)
- Layout Examples (page 512)

Specific Fields

The following specific fields have been added to the Subversion definition:

- Trunk Directory
- Tags Directory
- Repository Layout

Trunk Directory

The Trunk Directory is the directory of the “head” or “main” development stream. Its default name is *trunk*, but other names are allowed, e.g., *main*, *head*.

This directory may reside directly under the repository path, or be defined per project, depending on the settings of the Repository Layout field described here below.

Note: The Trunk field may have an empty value only if the Repository Layout field has the "Repository-oriented" value. In the case of a Project-oriented or Single Project-oriented Repository Layout, the Trunk directory must be specified to avoid infinite continuous builds if a schedule is set on such a repository definition (because the Tags directory would be part of the trunk, tagging would mean changing the trunk, and you would get an infinite loop).

Tags Directory

The Tags Directory is the directory where IKAN ALM creates the tags (also called labels) after a successful build. Its default name is *tags*, but other names are allowed, e.g., *tag*.

This directory may reside directly under the repository path, or be defined per project, depending on the settings of the Repository Layout field.

Repository Layout

The default Repository Layout is "Project-oriented". Other possible values are: "Repository-oriented" and "Single Project-oriented".

See the Layout Examples (page 512) for the configuration of the 4 possible Layouts.

Layout Examples

This section provides some possible repository project layouts followed by the possible solution in IKAN ALM for a Subversion Repository.

Note: The directory names used (trunk, tags, branches) are fully arbitrary. If you decided to call the trunk "main" or "head", or the branches "stream" or …, you are free to do so.

Layout 1 (Project-oriented)

Structure in the Subversion Repository:

```
/
/calc
/calc/trunk
/calc/branches
/calc/tags
/paint
/paint/trunk
/paint/branches
/paint/tags
```

Definition in IKAN ALM:

Note: URL = Repository URL defined in the Subversion definition.

	Description	Value
Trunk Directory (Subversion Repository)	The Trunk Directory for the Subversion Repository.	trunk
Tags Directory (Subversion Repository)	The Tag Directory for the Subversion Repository.	tags
Repository Layout (Subversion Repository)	One of the 3 possible Repository Layouts.	Project-oriented
VCR Project Name (Project Definition)	The name of the Project as known within the VCR.	calc
Validate Project (Project Definition)	When clicking "Check Project Name in the VCR" the existence of these directories is checked.	URL/calc/trunk URL/calc/tags
Check-out Main (Level Request Handling)	Directory checked out from the head stream of the project in the "Retrieve Code" Phase.	URL/calc/trunk
Tag Sample Main (Level Request Detail)	Sample of a tag after a successful build in the head stream.	URL/calc/tags/H_1-0_b1
VCR Branch ID (Branch Project Stream Definition)	The Branch ID defined within the VCR.	/calc/branches/B_1-2
Validate Branch (Branch Project Stream Definition)	When clicking "Check Branch ID in the VCR" the existence of these directories is checked.	URL/calc/branches/B_1-2
Check-out Branch (Level Request Handling)	Directory checked out from the Branch of the project in the "Retrieve Code" Phase.	URL/calc/branches/B_1-2
Tag Sample Branch (Level Request Detail)	Sample of a tag after a successful build in a branch stream.	URL/calc/tags/B_1-2_b5

Layout 2 (Repository-oriented)

Structure in the Subversion Repository:

```
/
/trunk
/trunk/calc
/trunk/paint
/tags/paint
/tags/calc
/branches
```

Definition in IKAN ALM:

Note: URL = Repository URL defined in the Subversion definition.

	Description	Value
Trunk Directory (Subversion Repository)	The Trunk Directory for the Subversion Repository.	trunk
Tags Directory (Subversion Repository)	The Tag Directory for the Subversion Repository.	tags
Repository Layout (Subversion Repository)	One of the 3 possible Repository Layouts.	Repository-oriented
VCR Project Name (Project Definition)	The name of the Project as known within the VCR.	calc
Validate Project (Project Definition)	When clicking "Check Project Name in the VCR" the existence of these directories is checked.	URL/trunk/calc URL/tags/calc
Check-out Main (Level Request Handling)	Directory checked out from the head stream of the project in the "Retrieve Code" Phase.	URL/trunk/calc
Tag Sample Main (Level Request Detail)	Sample of a tag after a successful build in the head stream.	URL /tags/calc/H_1-0_b1
VCR Branch ID (Branch Project Stream Definition)	The Branch ID defined within the VCR.	/branches/calc/B_1-2
Validate Branch (Branch Project Stream Definition)	When clicking "Check Branch ID in the VCR" the existence of these directories is checked.	URL/branches/calc/B_1-2
Check-out Branch (Level Request Handling)	Directory checked out from the Branch of the project in the "Retrieve Code" Phase.	URL/branches/calc/B_1-2
Tag Sample Branch (Level Request Detail)	Sample of a tag after a successful build in a branch stream.	URL /tags/calc/B_1-2_b5

Layout 3 (One Repository = One Project)

Structure in the Subversion Repository:

```
/
/trunk
/tags
/branches
```

Definition in IKAN ALM:

Note: URL = Repository URL defined in the Subversion definition.

	Description	Value
Trunk Directory (Subversion Repository)	The Trunk Directory for the Subversion Repository.	trunk
Tags Directory (Subversion Repository)	The Tag Directory for the Subversion Repository.	tags
Repository Layout (Subversion Repository)	One of the 3 possible Repository Layouts.	Single Project-oriented
VCR Project Name (Project Definition)	The name of the Project as known within the VCR.	Empty
Validate Project (Project Definition)	When clicking "Check Project Name in the VCR" the existence of these directories is checked.	URL/trunk URL/tags
Check-out Main (Level Request Handling)	Directory checked out from the head stream of the project in the "Retrieve Code" Phase.	URL/trunk
Tag Sample Main (Level Request Detail)	Sample of a tag after a successful build in the head stream.	URL /tags/H_1-0_b1
VCR Branch ID (Branch Project Stream Definition)	The Branch ID defined within the VCR.	/branches/B1-2
Validate Branch (Branch Project Stream Definition)	When clicking "Check Branch ID in the VCR" the existence of these directories is checked.	URL/branches/B1-2
Check-out Branch (Level Request Handling)	Directory checked out from the Branch of the project in the "Retrieve Code" Phase.	URL/branches/B1-2
Tag Sample Branch (Level Request Detail)	Sample of a tag after a successful build in a branch stream.	URL /tags/B_1-2_b5

Layout 4 (Repository-oriented, no trunk)

Structure in the Subversion Repository:

```
/
/calc (= trunk directory)
/paint (= trunk directory)
/tags/paint
/tags/calc
/branches
```

Definition in IKAN ALM:

Note: URL = Repository URL defined in the Subversion definition.

	Description	Value
Trunk Directory (Subversion Repository)	The Trunk Directory for the Subversion Repository.	Empty
Tags Directory (Subversion Repository)	The Tag Directory for the Subversion Repository.	tags
Repository Layout (Subversion Repository)	One of the 3 possible Repository Layouts.	Repository-oriented
VCR Project Name (Project Definition)	The name of the Project as known within the VCR.	calc
Validate Project (Project Definition)	When clicking "Check Project Name in the VCR" the existence of these directories is checked.	URL/calc URL/tags/calc
Check-out Main (Level Request Handling)	Directory checked out from the head stream of the project in the "Retrieve Code" Phase.	URL/calc
Tag Sample Main (Level Request Detail)	Sample of a tag after a successful build in the head stream.	URL /tags/calc/H_1-0_b1
VCR Branch ID (Branch Project Stream Definition)	The Branch ID defined within the VCR.	/branches/calc/B_1-2
Validate Branch (Branch Project Stream Definition)	When clicking "Check Branch ID in the VCR" the existence of these directories is checked.	URL/branches/calc/B_1-2
Check-out Branch (Level Request Handling)	Directory checked out from the Branch of the project in the "Retrieve Code" Phase.	URL/branches/B1-2
Tag Sample Branch (Level Request Detail)	Sample of a tag after a successful build in a branch stream.	URL /tags/calc/B_1-2_b5

E.3. Fetching Meta Properties

Subversion provides interfaces for adding, modifying, and removing versioned metadata on versioned directories and files. Those metadata are referred to as *properties*, see the "Version Control with Subversion, Chapter 7 - Advanced Properties" book at http://svnbook.red-bean.com/en/1.2/svn.advanced.props.html, which explains how to use and set these properties on files and directories.

IKAN ALM makes it possible to use those properties during the build and deploy processes. If the "Fetch Meta Properties" value is set to *yes*, the following will happen during the build process:

1. Creation of a properties file (vcr.properties)

   At the end of the "Retrieve Code" Phase during the handling of a Level Request a properties file `vcr.properties` will be created in the root directory of the checked out code. This file will contain all meta properties of the files in the checked out URL (trunk or branch) in the following form:

   ```
   path.to.file.filename.propertyname=propertyvalue
   ```

   If a file has multiple properties linked to it, this will of course result in multiple entries in the `vcr.properties` file.

   Some samples

   Sample 1: A property with property name *register* and property value *true* on the file */bin/Musicbiz.dll* will result in the following propertyline: `bin.Musicbiz.dll.register= true`

   Sample 2: The file */MusicLib/WEB-INF/web.xml* has two properties linked to it: *servletspecs=2.5* and *unchangeable=true*. This will result in following property lines in the `vcr.properties` file:

   ```
   MusicLib.WEB-INF.web.xml.servletspecs= 2.5
   MusicLib.WEB-INF.web.xml.unchangeable= true
   ```

2. Transportation of the vcr.properties file to the Source Directory of the Build Environment

   During the "Transport Source" Phase of a Build process the `vcr.properties` file will be transported to the source directory of the Build Environment together with the sources checked out from your Subversion trunk or branch directory.

   In order to use the properties, the build script should be adapted, e.g., when using an ANT scripting tool, the properties may be imported in the script via a *<property file="vcr.properties"/>* task.

3. Inclusion of the vcr.properties file in the Build result

   When these properties should also be available during a following deploy process(es), i.e., on a deploy environment that is linked to this build environment, the `vcr.properties` file should be included in the Build result.

   This can be achieved by including a copy instruction in the build script which copies the `vcr.properties` file from the source (predefined build parameter *source*) to the target (predefined build parameter *target*) directory of the Build Environment.

   In that way the `vcr.properties` file will be included in the build result, and as a consequence it will be available to be used in the deploy script during a deploy action of this build result.

APPENDIX F

RSS URL Details

The IKAN ALM URL has the following structure:

protocol://almhost:almhostport/alm/displayRss.actionrequestparams

This is equal to the System Settings of the IKAN ALM URL (Global Administration > System Settings):

- "protocol" = http or https
- "almhost:almhostport" = the IKAN ALM server domain,e.g., localhost:8080
- "requestparams" =

The list of (optional) criteria will be added, separated with an "&" sign. The following table gives you an overview of the optional criteria.

Optional Criteria	Description
Title=value	This is the title that will be shown in the RSS Feed. If it is not provided, it will be set automatically when the RSS feed is constructed. This way it is possible to place more than one IKAN ALM RSS Feed in the RSS reader (e.g., for different projects). **Note:** white spaces in the request will be provided via a "+" sign.
projectName=value	The Project Name. For example: projectName=DEMOCVS, or projectName=D*
status=value	The Level Request status. Multiple statuses may be provided. For example: statuses=1&statuses=1 Valid statuses are 0-10, with 0=unknown, 1=awaiting requested Date/time, 2=awaiting Approval, 3=rejected, 4=run, 5=fail, 6=success, 7=warning, 8=cancelled, 9=aborting, 10=aborted
projectStreamBuildPrefix=value	The Project Stream Build Prefix as defined in a Project Stream. For example: ProjectStreamBuildPrefix=1
projectStreamBuildSuffix=value	The Project Stream Build Suffix as defined in a Project Stream. For example: ProjectStreamBuildSuffix=1
levelName	The level name. For example:levelName=contbuild
levelType=value	The level type. For example: levelType=0 Valid values are 0-2, with 0=Build, 1=Test, 2=Production
lang=value	Valid locale indicating the user's language (default = none = en). For example: lang=en, lang=de, lang=fr (the three currently supported languages in IKAN ALM)

Valid sample URLs:

http://localhost:8080/alm/displayRss.action

http://localhost:8080/alm/ displayRss.action?lang=de

https://pe4600:8080/alm/
displayRss.action?projectName=DEMOCVS&projectStreamBuildPrefix=1&projectStreamBuildSuffix=1&levelType=0&lang=fr

https://pe4600:8080/alm/
displayRss.action?projectName=DEMOCVS&projectStreamBuildPrefix=1&projectStreamBuildSuffix=1&levelType=0&title=IKAN ALM+builds+in+project+DEMOCVS

APPENDIX G

Phases - General Information

This appendix provides more detailed information on the IKAN ALM concept of “Phases”.

When IKAN ALM is running Level Requests, Builds and Deploys, all actions are performed by executing a sequence of Phases.

Phases represent specific tasks or actions that must be performed by the system.

Phases are defined in the IKAN ALM database and can be consulted and manipulated in the Phases section of the Global Administration interface. Once they have been defined in Global Administration, Phases may be linked to Levels, Build or Deploy Environments in the Project Administration context.

Phases are categorized into different types depending on who created them (“Core” and “Custom” Phases) and where they are used (Level, Build and Deploy Phases).

The IKAN ALM core functionality is performed by so-called "Core" Phases. Those Core Phases can only be viewed, and cannot be altered nor deleted. Consider them an integral part of IKAN ALM. When you create a new Level, Build or Deploy Environment, its default workflow will be created and will completely consist of a sequence of Core Phases. Next, you may change this default workflow by removing Core Phases, by changing the sequence order or by adding your own Phases.

Those own Phases are also called “Custom” Phases. You may create them from scratch in Global Administration, based on one or more working scripts and resources, or you can import an existing Custom Phase, using the “Import Phase” functionality.

We also distinguish 3 types of Phases in IKAN ALM, depending on where they are being used:

- Level Phases (page 522): phases inserted into the Level workflow. They will be executed on the IKAN ALM Server.
- Build Phases (page 526): phases inserted into the Build Environment workflow. They will be executed on an IKAN ALM Agent.
- Deploy Phases (page 529): phases inserted into the Deploy Environment workflow. They will be executed on an IKAN ALM Agent.

A Level Request represents the execution of a Level. A Level is split up into Level Phases. Those Phases will be executed sequentially, and when all Phases have completed, the Level Request has ended. One of the Phases of a Level Request is the Build Phase, which will execute the Builds related to this Level Request. A Build represents the execution of a Build Environment. These Builds, for their part, are split up into Build Phases. When all the defined Build phases in the Build have executed successfully, the Build was successful. The Deploys of a Level Request are handled similarly.

Note: From IKAN ALM 5.5 onwards, there is no strict boundary anymore between those three types of Phases. The Core “Execute Script” Phase, is as well a Level as a Build or Deploy Phase, since you may use it on a Level, Build or Deploy Environment. Also, when you create a Custom Phase, you define where the Phase may be used: on a Level, on a Build Environment or on a Deploy Environment (or on all three Environments). In this context, we consider the Level to be also an "Environment".

All types of Phases inserted into the workflow of a Level/Build/Deploy Environment have the following general fields:

Field	Description
Phase Name	This field corresponds to the Display Name of the Phase definition and, at the same time, defines its function.
Phase Version	This field, together with the Phase name, forms the unique identifier of the Phase. The version of the Core Phases matches the IKAN ALM version (e.g., 5.5.0). The version of Custom Phases conforms to the format as explained in the section Creating a Phase Definition (page 425).
Next Phase On Success (Phase Position)	This field defines which Phase should be executed next if the current Phase completes successfully. It can't be edited directly; its value is inferred by the order in the Phase sequence of the Level/Build Environment/Deploy Environment.
Next Phase On Error	This field defines which Phase should be executed next if the current Phase has gone in Error. Usually this Phase is a Cleanup Phase.
Fail On Error	This field defines whether the status of the enclosing Level Request, Build or Deploy should be set to Fail or to Warning when this Phase goes in Error. Possible values: • *Yes*: set this to *Yes* when the Phase is critical (e.g., a Retrieve Code Phase, an Execute Build Script Phase, a Custom Compile Sources Phase, ...). The Level Request, Build or Deploy status will be set to *Fail*. • *No*: set this to *No* for a non-critical Phase (e.g., a Cleanup Build Result Phase, a Custom Test Phase, ...). The Level Request, Build or Deploy status will be set to *Warning*.

For more information on defining and configuring Phases, refer to the following sections: Editing a Level Phase (page 197), Editing a Build Environment Phase (page 219), Editing a Deploy Environment Phase (page 234), Creating a Phase Definition (page 425)and Editing a Phase Definition (page 429).

When executing a Phase, it can have the following status:

Status	Description
Success	The Phase finished successfully
Warning	The Phase finished with a warnings
Fail	The Phase failed
Running	The Phase is currently being executed
Not executed	The Phase has not been executed
Aborted	The Phase was aborted

The execution of a Phase can be viewed on the Phase Logs (page 77) tab page on the *Level Request Detail* screen.

G.1. Level Phases

Level Phases are the actions that must be performed to complete a Level Request.

A vanilla IKAN ALM installation only contains Level Phases of the "Core" type. You can search for them in the *Phases Overview* by restricting the search to Core Phases that can be used on a Level. Next to those Core Phases, you may enrich the functionality of IKAN ALM by creating your own Custom Phases which can be used on a Level.

Level Phases may be inserted into the workflow of a Level (Level Phases (page 195)) and are logged on the *Phase Logs* panel of the *Level Request Detail* screen (Level Phases (page 80)).

They are executed by the Monitor Thread of the IKAN ALM Server, so they run on the IKAN ALM Server Machine!

Their exact behavior may depend on the Level Type, and on the status of previously executed Level Phases.

The following section describes each of the Core Level Phases in detail:

- Retrieve Code Phase (page 522)
- Build Phase (page 523)
- Tag Code Phase (page 523)
- Deploy Phase (page 523)
- Issue Tracking Phase (page 524)
- Link File Revisions (page 524)
- Cleanup Work Copy Phase (page 525)
- Execute Script Phase (page 525)

Next to the Core Level Phases, you can create your own Custom Level Phases:

- Custom Level Phase (page 525)

Retrieve Code Phase

The *Retrieve Code* Phase is usually the first Phase executed in a Level Request. It retrieves (checks out) the source code from the VCR and copies it to the Transport Location (sub folder of the Work Copy Location) where it is accessible for the Agents running the Builds of the Level Request.

When the Level Request is for a Build Level, the latest source code is checked out; when it is for a Test Level, the tagged source code is retrieved.

When the Level Request is for a Build Level attached to a Tag-based Project Stream, the source code is retrieved that was tagged with the tag specified in the *VCR Tag* field when the Level Request was created. See the description of the *VCR Tag* field in the section Creating a Build Level Request (page 48).

The *Retrieve Code* Phase also retrieves the source code and/or the build results of all Child Builds this Level Request is depending on. See Build Dependencies (page 100).

In the case of a Partial Build, (Creating a Branch Project Stream (page 161)), only the modified source code will be retrieved and made available to the Agents running the Builds of the Level Request.

The *alm.phase.retrieve.source.partialBuild.partialCheckout* (Environment) Phase Parameter defines how this is done: if it is set to *true* (default value) and if the VCR supports it (currently only Subversion), this is done by a partial checkout of the modified sources. Otherwise, all sources will be checked out, but only the modified sources will be transported to the source location of the Build Environment.

When the Level Request has no Builds associated with it, the *Retrieve Code* Phase does nothing and exits with status *Success*. In that case, you could remove the *Retrieve Code* Phase from the Level's workflow.

Build Phase

The *Build* Phase activates and monitors the execution of the Builds of the Level Request.

When it starts, it activates the Builders running on the Agent Machines to start all the waiting Builds of the Level Request. Then, it waits for all the Builds to finish.

Meanwhile, when a Level Request is aborted while in the Build Phase, the Build Phase notifies and stops the executing Builders.

When all Builds have finished, the status of the Build Phase is set, depending on the statuses of the Builds:

- When a Build has failed, the status of the Build Phase is set to *Error*,
- When no Builds have failed, but one Build ended with status *Warning*, the status of the Build Phase is set to *Warning*
- When all Builds executed successfully, the status of the Build Phase is set to *Success*.

When no Builds have been defined for the Level Request, the *Build* Phase does nothing and exits with status *Success*. In that case, you could remove the *Build* Phase from the Level's workflow.

Tag Code Phase

The *Tag Code* Phase tags the code that was checked out with the VCR Tag associated with the Level Request. When the tag already exists in the VCR, the tag is moved.

The *Tag Code* Phase only tags when the checked-out code was the latest code of the VCR branch. So, it won't tag for a Build Level in a Tag Based Project Stream, and it won't (re-)tag for a Deliver Level Request. In both of those cases, tagged code was checked out, so no tagging was needed, and you could remove the *Tag Code* Phase from the Level's workflow.

Deploy Phase

The *Deploy Phase* activates and monitors the execution of the Deploys of the Level Request.

When it starts, it activates the Deployers running on the Agent Machines to start all the waiting Deploys of the Level Request with Sequence Number `0`. Next, it waits for all those Deploys to finish. Next, when those Deploys all ended with status *Success* or *Warning*, it activates the Deploys that have Sequence Number `1`, and so on until there are no more Deploys or a Deploy has failed.

Meanwhile, when a Level Request is aborted while in the Deploy Phase, the Deploy Phase notifies and stops the executing Deployers.

When all Deploys have finished, the status of the Deploy Phase is set, depending on the statuses of the Deploys:

- When a Deploy has failed, the status of the Deploy Phase is set to *Error*,
- When no Deploys have failed, but one Deploy ended with status *Warning*, the status of the Deploy Phase is set to *Warning*
- When all Deploys executed successfully, the status of the Deploy Phase is set to *Success*.

When no Deploys have been defined for the Level Request, the *Deploy* Phase does nothing and exits with status *Success*. In that case, you could remove the *Deploy* Phase from the Level's workflow.

Issue Tracking Phase

The *Issue Tracking* Phase links Issues, managed in an external Issue Tracking System, with a Level Request, by searching for references to the Issues in the commit comments of the VCR.

In the case of a Build Level Request, the Issue Tracking Phase parses the commit comments that have been entered since the last successful Level Request for that Level and tries to match the Issue Pattern of the attached Issue Tracking System (Creating an Issue Tracking System (page 382)). All found Issues will be attached to the Level Request.

For an Atlassian JIRA, HP Quality Center or CollabNet TeamForge Issue Tracking System, the Issue Tracking Phase will also connect to and try to identify the issues in JIRA, HP ALM or TeamForge. For each identified Issue, it will try to get additional information from JIRA, HP ALM or TeamForge (like description, status, owner and priority) and store it in IKAN ALM.

When the Level Request is a Deliver Level Request, the Issue Tracking Phase enumerates all the Issues that have been attached to the previous successful Build Level Requests that have occurred since the last successful Deliver Level Request on this Level, and adds all of them to this Level Request.

For example:

Suppose we have built the following Builds: Build 3 with Issue 3, Build 4 with Issue 4, Build 5 with Issue 5 and 6. Previously, Build 2 was delivered. If we now deliver Build 5, Issues 3,4,5 and 6 will be attached to the Deliver Level Request.

For an Atlassian JIRA, HP ALM or TeamForge Issue Tracking System, the Issue Tracking Phase will also synchronize all the issues attached to the Deliver Level Request: it will compare the info about the issue in IKAN ALM with the current information in JIRA, HP ALM or TeamForge and update description, status, owner or priority if necessary.

If the Level Request was not successful, the Issue Tracking Phase does nothing, and exits with status *Success*, reporting that it did not process any Issues.

When no Issue Tracking System was attached to the Project of this Level Request, the Issue Tracking Phase does nothing, and exits with status *Success*.

Note: When you attach an Issue Tracking System to a Project after it has been created, you must manually add the Issue Tracking Phase to the Levels you want Issue Tracking to be performed on.

Link File Revisions

The *Link File Revisions* Phase links the involved File Revisions to the Level Request.

For a Build Level Request this is done based on the File Revisions that have been checked out from the VCR during the *Retrieve Code* Phase.

For a Deliver or Rollback Level Request, this is done based on the File Revisions linked to the Level Request (from the previous Level) that will be delivered or rollbacked. Although these File Revisions are also linked to the Package, this Phase is necessary in order to take a snapshot of the Package content at Level Request execution time, since this content will probably change in time.

As this Phase is only applicable to Level Requests for Packages, it will only appear in the Levels of Package-based Projects.

Cleanup Work Copy Phase

The *Cleanup Work Copy* Phase cleans up the Work Copy Location where the sources of the Level Request were checked out.

It fails when it can't find the folder. Typically, this Phase's Fail On Error flag is set to 'No', causing the Level Request to end with status *Warning* instead of *Fail* when this Phase goes in error.

If the Level has its *Debug* flag set to "Yes", the *Cleanup Work Copy* Phase does nothing, and exits with status *Error*, reporting that the *Debug* flag was set for the Level.

Execute Script Phase

The *Execute Script* Phase executes a script on the IKAN ALM Server Machine using the specified Scripting Tool and the pre-defined Level Parameters. Both the script (alm.phase.mainScript) and the Scripting Tool (alm.phase.builder) must be defined by a mandatory Phase Parameter after inserting this Phase into a Level.

The *Execute Script* Phase has been introduced on the Level from IKAN ALM 5.5 onwards, together with the Custom Phase. The log generated by the script is saved in the IKAN ALM database. Note that this Phase is never inserted into the default workflow of a Level (i.e., when creating a new Level from scratch).

When the script is executed successfully, the *Execute Script* Phase exits with status *Success*. If not, it exits with status *Error* and logs the errors on the *Phase Logs* panel of the *Level Request Detail* screen (Level Phases (page 80)).

Next to the Core Phases, you may define your own Phases in Global Administration (Creating a Phase Definition (page 425)) and specify that they may be used on a Level. Once inserted into the workflow of a Level, we call them Custom Level Phases.

Custom Level Phase

Note: The Display Name of a Custom Level Phase, as defined in Global Administration and provided by the creator of the Custom Phase, is used in the ALM interface when inserting it into a Level or viewing the log on the *View Level Request Log* screen. So, the name displayed could be something like "Retrieve from Archive" or "Filter Sources".

A *Custom Level* Phase executes a script on the IKAN ALM Server Machine using the specified Scripting Tool and the pre-defined Level Parameters. The Display Name of this Phase and the executed script (alm.phase.mainScript) are specified in the definition of this Custom Phase in Global Administration. The Scripting Tool (alm.phase.builder) that executes the script depends on the Execution Type of the Phase definition and its value must be set after inserting this Phase into a Level.

The Custom Level Phase has been introduced from IKAN ALM 5.5 onwards, together with the *Execute Script* Phase. The log generated by the script is saved in the IKAN ALM database. Note that this Phase is never inserted into the default workflow of a Level (i.e., when creating a new Level from scratch).

When the script is executed successfully, the *Custom Level* Phase exits with status *Success*. If not, it exits with status *Error* and logs the errors on the *Phase Logs* panel of the *Level Request Detail* screen (Level Phases (page 80)).

Note: A Custom Level Phase may also be a Custom Build or Deploy Phase: the definition in Global Administration can also specify that it may be used on a Build or Deploy Environment.

G.2. Build Phases

Build Phases are the actions that must be performed to complete a Build. A vanilla IKAN ALM installation only contains Build Phases of the "Core" type. You can search for them in the *Phases Overview*, by restricting the search to Core Phases that can be used on a Build Environment. Next to those Core Phases, you may enrich the functionality of IKAN ALM by creating your own Custom Phases that can be used on a Build Environment.

Build Phases may be inserted into a Build Environment (Build Environment Phases (page 215)). Their actions during the handling of a Build are logged on the *Phase Logs* tab page of the *Level Request Detail* screen (Build Actions (page 82)).

They are executed by the Builder Thread of the IKAN ALM Agent, so they run on an IKAN ALM Agent Machine!

The following section describes each of the Core Build Phases in detail:

- Transport Source Phase (page 526)
- Verify Build Script Phase (page 526)
- Execute Script Phase (page 527)
- Transport Deploy Script Phase (page 527)
- Transport Package Results Phase (page 527)
- Compress Build Phase (page 527)
- Archive Result Phase (page 528)
- Cleanup Source Phase (page 528)
- Cleanup Result Phase (page 528)

Next to the Core Build Phases, you can create your own Custom Build Phases:

- Custom Build Phase (page 528)

Transport Source Phase

The *Transport Source* Phase transports the Source code and, possibly, the build results of Child Projects from the Work Copy Location on the IKAN ALM Server Machine to the Build Environment Source Location on the IKAN ALM Agent Machine, using the Transporter associated with the IKAN ALM Agent Machine.

When doing a Partial Build, the Transport Source Phase may also transport the build results of the previous Build from the Build Archive Location on the IKAN ALM Server Machine to the Environment's Source Location on the IKAN ALM Agent Machine. Set the *alm.phase.transport.source.partialBuild.copyPreviousBuildResult* (Environment) Phase Parameter to *true* to obtain this behavior. Note that the default value of this Parameter is *false*.

See also Retrieve Code Phase (page 522).

Verify Build Script Phase

The *Verify Build Script* Phase tries to locate the defined Build Script, and fails if it cannot.

First, it determines what Build Script to look for. If there's a Build Script defined on the Build Environment (Creating a Build Environment (page 210)), it will try to find that. If not, it will look for the Build Script defined on the Project (Editing Project Settings (page 149)).

Then, it tries to find the Build Script in the Build Environment's Source Location.

If not found, it tries to copy the Build Script from the IKAN ALM Script Location as defined in the System Settings (System Settings (page 255)).

If not found there either, the Verify Build Script Phase exits with status *Error*.

If found, the Verify Build Script Phase exits with status *Success*, and reports where it located the Build Script.

Execute Script Phase

The *Execute Script* Phase executes the Build Script on the defined Machine using the specified Scripting Tool and the defined Build Parameters.

It saves the Build log generated by the Build Script in the IKAN ALM database.

When the Build Script is executed successfully, the *Execute Script* Phase exits with status *Success*. If not, it exits with status *Error* and logs the errors on the *Phase Logs* panel of the *Level Request Detail* screen (Build Actions (page 82)).

Transport Deploy Script Phase

The *Transport Deploy Script* Phase copies the Deploy Scripts that are defined in the Deploy Environments linked to the Build Environment of this Build from the Build Environment's Source Location to the Target Location.

This action is performed, so that the Deploy Scripts are included in the compressed Build File created by the Compress Build Phase.

Transport Package Results Phase

This Phase is only relevant for Package Builds.

If the Package is part of a Package Build Group, it will retrieve the latest Build Results of some (or all, dependent on the configuration of the Package Build Group) of the Packages in the Package Build Group. It will use the Transporter defined for the Agent to transport the Results from the IKAN ALM Build Archive on the IKAN ALM Server to the ${sourceRoot}/packages directory on the Build Environment. The Phase also creates a *PackageBuildGroup.xml* file in the ${sourceRoot}/packages directory on the Build Environment that can be used as input in later Phases, e.g., for the Mainframe Compilation workflow, to transfer these Build Results and build up the correct PDS structure on the Mainframe.

The *Retrieve All Build Results* attribute of the Package Build Group, and the setting of the Dependency Level of the Packages in the Package Build Group determine which Build Results will be retrieved: the latest build results of all Packages in the Package Build Group in case *Retrieve All Build Results* has been set to *true*, or only the latest Build Results of Packages with a lower Dependency Level in case *Retrieve All Build Results* has been set to *false*.

Compress Build Phase

The *Compress Build* Phase compresses the Build result files in the Build Environment's Target Location.

If the Agent Machine runs a Windows OS, the Compress Build Phase creates a `.zip` file, otherwise it creates a `.tar.gz` file

Archive Result Phase

The *Archive Result* Phase transports the compressed Build file from the Build Environment's Target Location on the IKAN ALM Agent Machine to the Build Archive Location on the IKAN ALM Server Machine, using the Transporter associated with the IKAN ALM Agent Machine.

Cleanup Source Phase

The *Cleanup Source* Phase deletes all files in the Build Environment's Source Location.

If the Build Environment has its Debug flag set to 'Yes', the Cleanup Source Phase does nothing, and exits with status *Error*, reporting that the Debug flag was set in the Build Environment.

Cleanup Result Phase

The *Cleanup Result* Phase deletes all files in the Build Environment's Target Location.

If the Build Environment has its Debug flag set to 'Yes', the Cleanup Result Phase does nothing, and exits with status *Error*, reporting that the Debug flag was set in the Build Environment.

Next to the Core Phases, you may define your own Phases in Global Administration (Creating a Phase Definition (page 425)) and specify that they may be used on a Build Environment. Once inserted into the workflow of a Build Environment, we call them Custom Build Phases.

Custom Build Phase

Note: The Display Name of a Custom Build Phase, as defined in Global Administration and provided by the creator of the Custom Phase, is used in the ALM interface when inserting it into a Build Environment or viewing the log on the *View Build Phases Log* screen. So, the name displayed could be something like "Generate Documentation" or "Run Unit Tests".

The *Custom Build* Phase executes a script on the IKAN ALM Agent Machine using the specified Scripting Tool and the defined Build Parameters. The Display Name of this Phase and the executed script (alm.phase.mainScript) are specified in the definition of this Custom Phase in Global Administration. The Scripting Tool (alm.phase.builder) that executes the script depends on the Execution Type of the Phase definition. When this Execution Type differs from the Scripting Tool linked to the Build Environment, its value must be set after inserting this Phase into a Build Environment.

The *Custom Build* Phase has been introduced from IKAN ALM 5.5 onwards. The log generated by the script is saved in the IKAN ALM database. Note that this Phase is never inserted into the default workflow of a Build Environment (i.e., when creating a new Build Environment from scratch).

When the script is executed successfully, the *Custom Build* Phase exits with status *Success*. If not, it exits with status *Error* and logs the errors on the *Phase Logs* panel of the *Level Request Detail* screen (Build Actions (page 82)).

Note: A Custom Build Phase may also be a Custom Level or Deploy Phase: the definition in Global Administration can also specify that it may be used on a Level or Deploy Environment.

G.3. Deploy Phases

Deploy Phases are the actions that must be performed to complete a Deploy. A vanilla IKAN ALM installation only contains Deploy Phases of the "Core" type. You can search for them in the *Phases Overview*, by restricting the search to Core Phases that can be used on a Deploy Environment. Next to those Core Phases, you may enrich the functionality of IKAN ALM by creating your own Custom Phases that can be used on a Deploy Environment.

Deploy Phases may be inserted into a Deploy Environment (Deploy Environment Phases (page 231)) and their actions during the handling or a Deploy are logged on the *Phase Logs* tab page of the *Level Request Detail* screen (Deploy Actions (page 85)).

They are executed by the Deployer Thread of the IKAN ALM Agent, so they run on an IKAN ALM Agent Machine!

Note: The number of running Deploys on an IKAN ALM Agent is managed by specifying the *Concurrent Deploy Limit* attribute for the Machine representing the IKAN ALM Agent. By default, this limit is set to *0*, meaning that all Deploys on the Agent will run concurrently (i.e., in parallel).

If another number is specified, a Deploy can only be started if there are not more running Deploys as indicated. So if the number is limited to 1, this means that all deploys will run sequentially on the Agent. If the number is set to 2, only 2 deploys can run concurrently, meaning that if there is a third Deploy with status *Ru,n* , this third one will be added to a "Waiting queue" and it will only be started if one of the other (running) Deploys has finished.

The following section describes each of the Core Deploy Phase in detail:

- Transport Build Result Phase (page 529)
- Decompress Build Result Phase (page 530)
- Verify Deploy Script Phase (page 530)
- Execute Script Phase (page 530)
- Cleanup Build Result Phase (page 530)

Next to the Core Deploy Phases, you can create your own Custom Deploy Phases:

- Custom Deploy Phase (page 531)

Transport Build Result Phase

The *Transport Build Result* Phase transports the Build result from the Build Archive Location on the IKAN ALM Server Machine to the Deploy Environment Source Location on the IKAN ALM Agent Machine, using the Transporter associated with the IKAN ALM Agent Machine.

When doing a Partial Deploy, only the modified and added files in the Build result are transferred. See the description of the *Partial Deploy* field in the section Creating a Deploy Environment (page 225).

Decompress Build Result Phase

The *Decompress Build Result* Phase decompresses the Build result file that was transported by the Transport Build Result Phase into the Deploy Environment's Source Location, and afterwards, deletes the Build result file.

Verify Deploy Script Phase

The *Verify Deploy Script* Phase tries to locate the defined Deploy Script, and fails if it cannot.

First, it determines what Deploy Script to look for. If there's a Deploy Script defined on the Deploy Environment (Creating a Deploy Environment (page 225)), it will try to find that. If not, it will look for the Deploy Script defined on the Project (Editing Project Settings (page 149)).

Then, it tries to find the Deploy Script in the decompressed Build result available in the Deploy Environment's Source Location.

If not found, it tries to copy the Deploy Script from the IKAN ALM Script Location as defined in the System Settings (System Settings (page 255)).

If not found there either, the Verify Deploy Script Phase exits with status *Error*.

If found, the Verify Deploy Script Phase exits with status *Success*, and reports where it located the Deploy Script.

Execute Script Phase

The *Execute Script* Phase executes the Deploy Script on the defined Machine using the specified Scripting Tool and the defined Deploy Parameters.

It saves the Deploy log generated by the Deploy Script in the IKAN ALM database.

When the Deploy Script is executed successfully, the *Execute Script* Phase exits with status *Success*. If not, it exits with status *Error* and logs the errors on the *Phase Logs* panel of the *Level Request Detail* screen (Deploy Actions (page 85)).

Cleanup Build Result Phase

The *Cleanup Build Result* Phase deletes all files in the Deploy Environment's Source Location.

If the Deploy Environment has its Debug flag set to 'Yes', the Cleanup Build Result Phase does nothing, and exits with status *Error*, reporting that the Debug flag was set in the Deploy Environment.

Next to Core Phases, you may define your own Phases in Global Administration (Creating a Phase Definition (page 425)) and specify that they may be used on a Deploy Environment. Once inserted into the workflow of a Deploy Environment, we call them *Custom Deploy* Phases.

Custom Deploy Phase

Note: The Display Name of a Custom Deploy Phase, as defined in Global Administration and provided by the creator of the Custom Phase, is used in the ALM interface when inserting it into a Deploy Environment or viewing the log on the *View Deploy Phases Log* screen. So, the name displayed could be something like "Update Database" or "Deploy to web server".

The *Custom Deploy* Phase executes a script on the IKAN ALM Agent Machine using the specified Scripting Tool and the defined Deploy Parameters. The Display Name of this Phase and the executed script (alm.phase.mainScript) are specified in the definition of this Custom Phase in Global Administration. The Scripting Tool (alm.phase.builder) that executes the script depends on the Execution Type of the Phase definition. When this Execution Type differs from the Scripting Tool linked to the Deploy Environment, its value must be set after inserting this Phase into a Deploy Environment.

The *Custom Deploy* Phase has been introduced from IKAN ALM 5.5 onwards. The log generated by the script is saved in the IKAN ALM database. Note that this Phase is never inserted into the default workflow of a Deploy Environment (i.e., when creating a new Deploy Environment from scratch).

When the script is executed successfully, the *Custom Deploy* Phase exits with status *Success*. If not, it exits with status *Error* and logs the errors on the *Phase Logs* panel of the *Level Request Detail* screen (Deploy Actions (page 85)).

Note: A *Custom Deploy* Phase may also be a Custom Level or Build Phase: the definition in Global Administration can also specify that it may be used on a Level or Build Environment.

APPENDIX H

Glossary of Terms

Approval	A *Pre-Approval* enables a verification moment before delivering a Level Request to a Test or Production Level. It adds an extra condition to the execution of the Level Request: not only the Requested Date/Time must be expired, but the Pre-Approval must also be granted by an IKAN ALM User. As long as both conditions are not fulfilled, the Level Request will not be handled. When a Pre-Approval is rejected, the Level Request will never be handled. A *Post-Approval* enables a verification moment after the execution of the Level Request on a Test or Production Level. It adds an extra condition to the determination of the end status of a Level Request: not only must all Build(s) or Deploy(s) have been ended successfully, but the Post-Approval must also be granted by an IKAN ALM User. As long as the Post-Approval is not handled, the Level Request status will not be set to warning or success. When a Post-Approval is rejected, the Level Request can never be delivered to the next Level in the Lifecycle.
Auditing a Project	After having created a new Project, that Project is locked. Auditing the Project will perform a certain number of checks and if all checks succeed the Project will be unlocked and ready to be used.
Branch Project Stream	A Branch Project Stream is a development Stream in the Project different from the Head (also referred to as main or trunk) Project Stream. It maps to the sources controlled in a Branch of the VCR. A Branch Project Stream is typically used for Releases, parallel development, patches or other ALM practices to support development next to the main development line. Just as a Head Project Stream, the Branch Project Stream must be linked to a Lifecycle in order to be active (Builds and Deploys).
Build	A Build is an action on a Build Environment which involves several sub processes, called Build Phases. It is always part of a Level Request, which may also contain other Builds or Deploys. [A Build is handled by the IKAN ALM Builder]. It starts from sources that were retrieved from the VCR to the Build Environment. A Build script is executed on those sources by a Scripting Tool and gives a Build Result that is transferred to the Build Archive.
Build Archive	The physical location (path) on the IKAN ALM Server where the Build Results are stored in a compressed and archived format (*.zip or *.tar.gz). The Build Results are organized by Project and by Project Stream.
Build Environment	A physical environment on a Machine where sources retrieved from the VCR may be transformed by a Build script executed by a Scripting Tool. A Build Environment is always part of a Level.
Build History	The Build History provides a historical overview of the Build Level Requests for a specific Lifecycle of a Project Stream. It allows you to verify the flow of a certain Build in the Lifecycle: was it ever delivered to a Level higher in the Lifecycle then the first Build Level? If yes, did it reach the highest Level (e.g., Production)?
Build Number	Each Build on the Build Level in a Project Stream has a unique Build Number. This is a sequential number that is incremented by IKAN ALM when a Build Level Request is created. The highest Build Number is stored on the Project Stream.

Build OID	The Build OID is the unique identifier of a Build used to store the build information in the database.
Build Parameter	Build Parameters are parameters that can be used during the execution of the Build Script. They are defined on a Build Environment, and can have a predefined value or be modified during the creation of a Level Request.
Build Phase	A Build Phase is a sub process that must be performed to complete a Build action. Different Build Phases form the workflow of a Build and are inserted into a Build Environment. They are executed by the IKAN ALM Builder Thread of the IKAN ALM Agent. A Build Phase may be a Core Phase (e.g., Verify Build Script), or a Custom Build Phase created or imported by the User in the Phase Catalog.
Build Prefix/Suffix	A series of numbers or a character set to uniquely differentiate a Project Stream in the Project, also referred to as a "Release Number". Samples: 4.2, or Head, or 1-0. A Head Project Stream is only be identified by a Build Prefix. A Branch Project Stream will combine the Build Prefix and Suffix to obtain its release number.
Build Tool	Scripting Tool installed on a Build Environment
Child Project Stream	A Project Stream who's sources or Build Result are used to create Builds in one (or more) Parent Project Streams. The Parent Project Stream has then a dependency link to the sources, or the Build Result of the Child Project Stream. Example: a child project stream may contain a common library that is used in several Parent Project Streams.
Continuous Build Level	A Build Level for which a Schedule has been defined verifying the latest sources in the VCR after a specified number of minutes (e.g., every 5 or 10 minutes). If the code has changed in the VCR, the Scheduler will notify the changes after the defined interval and a Level Request will be generated automatically.
Core Phase	Core Phases form the IKAN ALM "Core" functionality. They can only be viewed, and can be neither altered nor deleted. Consider them an integral part of IKAN ALM. When a new Level, Build or Deploy Environment is created, its default workflow will be created and will completely consist of a sequence of Core Phases. This default workflow may be changed by removing Core Phases, by changing the sequence order, or by adding Custom Phases.
Custom Phase	A Phase added by the User is also called a "Custom" Phase. It may be created from scratch in Global Administration based on one or more working scripts and resources, or it may be imported, using the "Import Phase" functionality. Once defined in Global Administration, a Custom Phase may be inserted into (and consequently change) the default work flow of a Level, Build or Deploy Environment. All Custom Phases are stored in the Phase Catalog on the IKAN ALM Server, and are transported automatically to the IKAN ALM Server (Level Phase) or IKAN ALM Agent (Build or Deploy Phase) when they are to be executed.
Deliver [Level Request]	A manually (via the Command Line or web interface) created Level Request to deliver sources or a Build Result to the next higher Test or Production Level in the Lifecycle of a Project Stream. The Level Request may contain Build(s) and/or Deploy(s).
Dependency	Dependencies are defined on Project Streams. This functionality makes it possible to reuse common libraries or sources from Child Projects Streams The project that reuses the common library is called a Parent Project Stream. Projects can be reused in two ways: as sources retrieved from the Versioning System or as a Build result retrieved from the Build Archive.
Deploy	A Deploy is an action on a Deploy Environment which involves several sub processes, called Deploy Phases. It is always part of a Level Request, which may also contain (an)other Build(s) or Deploy(s). [A Deploy is handled by the IKAN ALM Deployer]. It starts from a Build Result which is retrieved from the Build Archive. A Deploy script is executed on this Build Result by a Scripting Tool.

Deploy Parameter	Deploy Parameters are parameters that can be used during the execution of the Deploy Script They are defined on a Deploy Environment, and can have a fixed value or be modified during the creation of a Level Request.
Deploy Phase	A Deploy Phase is a sub process that must be performed to complete a Deploy action. Different Deploy Phases form the workflow of a Deploy and are inserted into a Deploy Environment. They are executed by the IKAN ALM Deployer Thread of the IKAN ALM Agent. A Deploy Phase may be a Core Phase (e.g., Transport Build Result) or a Custom Deploy Phase created or imported by the User in the Phase Catalog.
Deploy Tool	A scripting Tool installed on a Deploy Environment
Deploy Environment	A physical environment on a Machine where a Build result retrieved from the Build Archive on the IKAN ALM Server may be deployed by a Deploy script executed by a Scripting Tool. A Deploy Environment is always part of a Level.
Desktop	The Desktop lists current information about selected Project Streams and Levels. Users can personalize their Desktop adding the items they are interested in. Furthermore the personal Desktop provides shortcuts for creating Level Requests.
Environment Parameter	Environment Parameters are parameters that can be used during the *Execute Script* Phase which runs a Build/Deploy Script. They may also be used during the execution of a Custom Phase.
Environment Phase Parameter	Phases may have their own Phase Parameters. Once a Phase is linked to an Environment, specific values can be specified for these Phase Parameters, which are then referred to as Environment Phase Parameters.
Forced Build [Level Request]	If a Continuous Build Process has been defined for the Build Level by means of a Schedule and this Schedule is bypassed by generating a Build [Level Request] manually via the web interface or Command Line, this action is called a "Forced Build [Level Request]". The Level Request must at least contain one Build and may contain one or several Deploys.
Head Project Stream	A Stream in the Project mapped to the main development line (referred to as main, head or trunk) of the VCR. A Head Project Stream is typically used for the ongoing development of the next Release. Just as a Branch Stream, the Head Stream must be linked to a Lifecycle in order to be active (Builds and Deploys).
IKAN ALM Agent	A process (daemon) running on a Machine with sub processes to handle Build or Deploys. An Agent running on the same Machine as the IKAN ALM Server is also referred to as "local", whereas running on a different Machine it is indicated as "remote". During a Build or Deploy the IKAN ALM Agent interacts remotely with the IKAN ALM Monitor, and locally with a Transporter and with a Scripting Tool that must be correctly configured on the Machine.
IKAN ALM Server	The Machine hosting the IKAN ALM web application and the IKAN ALM Monitor and Scheduler processes.
IKAN ALM Monitor	A process (daemon) running on the IKAN ALM Server to handle Level Requests. During the proceeding of a Level Requests it interacts with a VCR client installed on the IKAN ALM Server and with a local or remote IKAN ALM Agent.
IKAN ALM Scheduler	A process (daemon) running on the IKAN ALM Server. In case a Schedule (= a predefined interval, e.g., each 5 or 10 minutes, each night or each week,…) is linked to a Build Level, the IKAN ALM Scheduler will verify, in the Version Control Repository, whether changes have been made to the sources in the VCR each time the Schedule interval expires. This enables Continuous Integration or Nightly Builds.
IKAN ALM Builder	A sub process (daemon) of the IKAN ALM Agent that will handle the Builds of a Level Request in the Build Environment on an Agent Machine.
IKAN ALM Deployer	A sub process (daemon) of the IKAN ALM Agent that will handle the Deploys of a Level Request in the Deploy Environment on an Agent Machine.

Issue Tracking	A system external to IKAN ALM, where Issues (defects, enhancements, tasks, …) may be defined for a Project. Samples are Atlassian JIRA, HP Quality Center, Collabnet TeamForge, Bugzilla or Trac. IKAN ALM can plug in to such a System and keep up with the Issues that were handled for a Level Request. The integration with JIRA, HP Quality Center and TeamForge is more advanced: Issues are automatically synchronized through the Lifecycle, and it is possible to keep a link with the Level Requests in the JIRA Issue, HP Quality Center Defect or TeamForge Artifact.
Level	A Level is a stage in the Lifecycle, a conceptual step in the process of promoting sources and build results from development to production. A Level is linked in a specific order to a Lifecycle. A Level must contain at least one (physical) Build and/or Deploy Environment in order to be active. It may have more than one Build and/or Deploy Environments to support parallel Builds or Deploys on multiple Machines.
Level Phase	A Level Phase is a sub process that must be performed to complete a Level Request. The execution of a Level Request is split up in Level Phases which will be executed sequentially. Different Level Phases form the workflow of a Level Request and are inserted into a Level. They are executed by the IKAN ALM Monitor Thread of the IKAN ALM Server. A Level Phase may be a Core Phase (e.g., Retrieve Code) or a Custom Level Phase created or imported by the User in the Phase Catalog.
Level Request	A Level Request is an action on a Level which involves several sub processes, called Level Phases. In most cases, a Level Request will contain at least one Build or Deploy action, which will be executed on local or remote Machines. A Level Request may be created manually by the user via the Web interface or the Command Line interface, or automatically by the Scheduler Thread of the IKAN ALM Server. A Level Request is handled by the Monitor Thread of the IKAN ALM Server.
Lifecycle	A Lifecycle is a sequence of Levels that is linked to a Project Stream. It enables to set up the step-by-step process to promote sources and build results from development, to test, QA, ... to end up into production. One Project may have different Lifecycles, e.g., for development on the next release, for maintenance or urgency fixes on the release currently in production, for parallel development,… A Lifecycle may be reused in more than one Project Stream.
Machine	A representation of a concrete Server. Builds and Deploys may be done on a Machine, when it is linked to a Build respectively Deploy Environment. Other conditions are that the IKAN ALM Agent is installed and running on the Machine, and that a Scripting Tool is installed on the Machine. The IKAN ALM Server is a special Machine, containing the web application and the running IKAN ALM Monitor and Scheduler Threads.
Machine Parameter	Machine Parameters are defined for a Machine instead of for a specific Environment. Parameters defined for a specific Machine, will automatically be available for all Environments using that Machine. This avoids having to (re)define Build and/or Deploy Parameters for each Environment linked to that Machine. If an Environment Parameter and a Machine Parameter have the same name, the Environment Parameter takes precedence.
Master Project Stream	The Master Project Stream is a Project Stream to which a Dependency (a Child Project Stream) is added. Builds in the Master Project Stream may use the sources or Build Result (e.g., a common library) from the Child Project Stream.
Notification	A message to a User defined in IKAN ALM via mail or netsend. Notifications may be sent when a Level Request fails or succeeds, when an Approval must be granted for a Level Request, when an Approval for a Level is rejected, when a Level Request is delivered to or from a certain Level, when the IKAN ALM Administrator wants to notify certain Users.
OID (Level Request OID / Build OID / Deploy OID /Level Approval OID)	Object Identifier. Unique number to identify a Level Request/ Build / Deploy / Level Approval from other Level Requests/ Builds / Deploys / Level Approvals

Package	A Package allows moving one or more individual files selected manually from a VCR Stream (Head or Branch) through the IKAN ALM Lifecycle. One or multiple Packages may be created in each Project Stream of a Package-based Project. This is different from the original way of working in the Project Streams of Release-based projects, where a configurable automated process defines which file revisions are retrieved from the head (trunk) or branch of the versioning system and moved in the Lifecycle.
Phase Catalog	The physical location (path) on the IKAN ALM Server where the Custom Phases (created from scratch or imported) are stored in an archived format (Phase.name-Phase.version.jar, e.g., com.ikanalm.echoproperties-1.0.0.jar). When an IKAN ALM Server or Agent needs to install a missing Custom Phase, it will be retrieved from that location. That will be done using the Transporter linked to the Server or Agent Machine.
Project	An IKAN ALM Project maps to a project or subproject in a versioning system (VCR) which bundles related sources. An IKAN ALM Project is a shell for one or more Project Streams in which the real actions (Level Requests, Builds, Deploys) are done. It is possible to set up dependencies between different Projects, also through the Project Streams. There are 2 types of Projects: • Release-based Projects: IKAN ALM will work with the existing structure in the VCR system, so that the objects to be extracted will be retrieved automatically when starting the build process. • Package-based Projects: this concept enables to work with isolated files from the VCR system. Objects must be selected manually in a Package structure created in IKAN ALM before starting the Build process.
Project Stream	The Project Stream concept enables to control different active Lifecycles within an IKAN ALM Project. In general, all Projects have a main Project Stream called "Head", in which development for the next upcoming release happens. In addition to this there will probably be one or more Branch Project Streams. A Branch Project Stream can be used for maintaining Project releases which are currently in Production, so that fixes can be rolled out automatically or urgent fixes can be promoted along a shorter (and faster) Lifecycle to production. A Branch Project Stream may also be used to allow parallel development, or to test upcoming Test and Production environments with different settings (new operating system, new compiler or database version, …).
Requested Build [Level Request$	A manually (via the Command Line or the web interface) created Level Request on a Build Level that has no Schedule linked to it. The Level Request must contain at least one Build and may contain (a) Deploy(s).
Rollback [Level Request]	A manually (via the Command line or web interface) created Level Requests to reset previously Delivered sources or Build results on a Test or Production Level in the Lifecycle of a Project Stream. The Level Request may contain Build(s) and/or Deploy(s)
Scripting Tool	A system external to IKAN ALM which can execute user-created scripts and which is installed on a Machine. IKAN ALM integrates with Ant, Gradle, NAnt and Maven2. When the Scripting Tool is linked to a Build respectively Deploy Environment it is also referred to as a Build respectively Deploy Tool. The script for executing a Build or Deploy must be stored in the VCR (together with the sources) or in the Script Location on the IKAN ALM Server.
Tag-based Build	A tag-based Build will be executed on sources with a pre-applied tag (manually by a user) in the VCR, whereas a non-Tag-Based Build will be executed on the latest sources, also called the "tip", of a branch or head (trunk/main) stream from the VCR.
Transporter	A Transporter is used for transporting files and directories between the IKAN ALM server and a local or remote Agent handling the Build or Deploy processes. Therefore, a Transporter must be defined for a specific Machine that is linked to the Build or Deploy Environment. IKAN ALM supports the local FileCopy, remote FileCopy, SecureCopy and FTP Transporters. A Transporter may transport checked-out sources from the Versioning System, a Build result from the Build Archive, but also Custom Phases from the Phase Catalog.

User	A person that may log on to IKAN ALM. The membership to User Groups determines his or her "Access Rights, i.e., what actions (Global or Project administration, creation of Level Request, verification of Projects,…) a User can do in IKAN ALM. Users are not created manually in IKAN ALM, but in an external security system (like LDAP or Active Directory). If the User belongs to the correct User Group in this security system, he or she may log on to IKAN ALM and will be created automatically.
User Group	An entity grouping Users with the same "Access Rights". Actions in IKAN ALM (Global or Project administration, creation of Level Request, verification of Projects,…) are protected by a User Group. User Groups must be created in IKAN ALM. There are two types of User Groups: external and internal. The membership of Users to an external User Group is set in the external Security System. Each time a User logs on to IKAN ALM, his or her memberships to the different User Groups will be synchronized with the external security system. Internal User Groups, however, are not synchronized with an external security system: they are intended for notification and approval purposes and are managed manually through the IKAN ALM interface.
VCR Branch ID	A unique identifier for a Branch in the external VCR system.
VCR Tag	After a successful Level Request on a Build Level the IKAN ALM Monitor applies a tag in the Version Control Repository (VCR) system. This VCR Tag matches a Build [Level Request] with its source code in the VCR The format of the VCR Tag normally matches the Tag Template defined for the Project Stream.
Version Control Repository (VCR)	An external versioning system holding different versions of sources. Related sources are bundled in a Project or subproject (sometimes also called a Module). A VCR Project may contain different development streams, called head (=main or trunk) or branch streams. IKAN ALM integrates with the following VCRs: Subversion, Git, CVS, Microsoft Visual SourceSafe, IBM ClearCase, Serena PVCS and TFVC. In order to connect to the VCR, a VCR Client must be installed on the IKAN ALM Server and correctly configured. The IKAN ALM Monitor interacts with the VCR by retrieving or tagging sources. The web interface interacts with the VCR to show revision numbers, modified sources, … related to a Level Request.
Work Copy	A physical location (path) on the IKAN ALM Server to which the Monitor retrieves the source code from the VCR or Build Results from the Build Archive.

Index

A

Ant
    Overview Screen 400
ANT Scripting Tools
    Creating 397
    Deleting 402
    Editing 401
    History 403
Approvals 21, 91
    Approving 23
    Level Approval Sequence 205
    Level Approvals 205
    Outstanding Approvals 22
    Overview Screen 26
    Rejecting 24
Auditing 248
    Error and Warning Messages 249
    Projects 248

B

Build Dependencies 100
Build Environments 209
    Cloning 215
    Creating 203, 210
    Deleting 214
    Editing 213
    History 224
    Overview Screen 212
    Phases 215
        Changing the order 219
        Deleting 223
        Editing 219
        Inserting 217
        Overview 216
        Viewing the Build Environment Phase Parameters 220
    View Parameters 223
Build History 104
Build Parameters
    Predefined Build Parameters 505
    Predefined Deploy Parameters 508
    Predefined Level Parameters 503
Build Phases Log 77
Build Result 89
    Download 89
Build Usage 101
Builds and Deploys Overview 107

C

ClearCase 360
    Creating 360
    Deleting 365
    Editing 364
    History 366
    Overview Screen 363
Cloning an Existing Project 468
Command Line Interface 475
    Parameters 475
Creating an IKAN ALM Project 461
    Package-based 461
    Release-based 461
CVS 346
    Creating 346
    Deleting 351
    Editing 350
    History 353
    Overview Screen 349

D

Dependencies 99
Deploy Environments 225
    Cloning 230
    Creating 204, 225
    Deleting 230
    Editing 229
    History 239
    Overview 228
    Phases 231
        Changing the order 234
        Deleting 238
        Editing 234
        Inserting 233
        Overview Screen 232
        Viewing the Deploy Environment Phase Parameters 235
    View Parameters 239
Deploy Phases Log 77
Desktop 14
    Adding a Desktop Tab Page 139
    Adding Elements to a Desktop Tab Page 140
    Approvals 21
    Build History 104

Editing the Desktop Tab Page Label 139
Introduction 14
Level Requests 29
Managing Your Desktop 135
Overview 136
Package Build Groups 125
Packages 110
Personal Settings 15
Rearranging Desktop Elements 143
Removing a Desktop Tab Page 140
Removing Desktop Elements 143
Search Filters 16
Download Build Result 89

E

Environment Parameters 240
Copying 245
Creating 242
Deleting 245
Editing 244
History 244
Overview Screen 241
Event Logging 497

F

FileCopy Transporters 310
Creating 311
Deleting 314
Editing 314
History 315
Overview Screen 313
FTP Transporters 316
Creating 316
Deleting 321
Editing 320
History 322
Overview Screen 319

G

Git
Creating 340, 373
Deleting 345, 379
Editing 344, 378
History 346, 380
Overview Screen 343, 376
Global Administration 253
History Log 271
Housekeeping 262
Housekeeping History 265
Issue Tracking 381
Machines 289
Miscellaneous 472
Phases 424
Projects 461
Schedules 452
Scripting Tools 394
Synchronize Build Archive 268
System Settings 255
Transporters 307
User Groups 280
Users 275
Version Control Repositories 330
Gradle
Overview Screen 407, 420
Gradle Scripting Tools
Creating 404
Deleting 409
Editing 408
History 410

H

History 497
History Log 158, 271
Housekeeping 262
Housekeeping History 265

I

IKAN ALM
Installing a New License 5
Introduction 1
Logging on 3
Logging out 5
Installed Phases Overview 298
Issue Tracking 381
Issue Tracking System Properties 390
Issue Tracking Systems
Creating 382
Deleting 389
Editing 387
History 390
Overview Screen 385
Properties
Creating 390
Deleting 390
Editing 390
Overview Screen 390
Issues 93
Adding 96
Deleting 95
Editing 94
Synchronizing 95

L

Level Parameters 77
Level Phases 195
Level Request Detail 65
Approvals 91
Automatic Refresh 65

Build Dependencies 100
Build Usage 101
Dependencies 99
Issues 93
Level Request Usage 103
Modifications 98
Phase Logs 77
Results 89
Sources 97
Summary 66
Level Request Usage 103
Level Requests 29
Approvals 91
Build 30
Build and Deploy 35
Build Dependencies 100
Build Phases Log 77
Build Usage 101
Creating 45
Selecting a Level 45
Deliver Build 52
Dependencies 99
Deploy 42
Deploy Phases Log 77
Force Build 48
Issues 93
Level Request Usage 103
Modifications 98
Overview
Builds and Deploys 107
Level Requests 58
Phase Logs 77
Rebuild 30
Rebuild and Deploy 35
Reports 63
Request Build 48
Results 89
Rollback Build 55
Sources 97
Summary 66
Actions 66
Approvals 66
Builds and Deploys 66
Error Log 66
Info 66
Issues 66
Warning Log 66
Levels 187
Changing the order 177
Creating a Level 188
Creating a Production Level 190
Creating a Test Level 190
Deleting 206
Editing 184, 194
History 208
Inserting 177
Optional Levels 177
Overview Screen 193
Phases 195, 522
Viewing the Level Phase Parameters 198
Required Levels 177
License
Install 5
Lifecycles 174
Creating 175
Deleting 182
Editing 177
History 184
Levels
Changing the order 177
Editing 184
Inserting 177
Optional Levels 177
Removing 186
Required Levels 177
Overview Screen 176
Locking
Project Streams 168
Projects 153
Logging on 3
Logging out 5

M

Machine Parameters 300
Copying 306
Creating 303
Deleting 305
Editing 304
History 306
Overview Screen 301
Machines 289
Creating 289
Deleting 296
Editing 294
History 300
Overview Screen 292
Parameters 300
Status 297
Managing Your Desktop 135
Mass deleting phases 445
Mass replacing phases 442
Maven2 Scripting Tools 416
Creating 417
Deleting 422
Editing 421
History 423
Miscellaneous 472
Notifying IKAN ALM Users 472
Modifications 98

N

NANT
Overview Screen 413

NANT Scripting Tools 410
Creating 411
Deleting 415
Editing 414
History 416
Notifying IKAN ALM Users 472

O

Outstanding Approvals 22
Approving 23
Overview Screen 26
Rejecting 24

P

Package Build Groups 125
Adding Packages 130
Creating 126
Dependency Levels 130
Editing 129
History 134
Overview Screen 127
Packages Sequence Order 130
Packages 110
Creating 111
Editing 116
Hiding 117, 133
History 117
Overview Screen 113
Showing 117, 133
View Settings 115, 130
Parameters 80
Build Environment 223
Build Environment Phase 220
Deploy Environment 239
Deploy Environment Phase 232, 235
Environment 240
Level 80
Level Phases 198
Machine 300
Phase 434
Personal Settings 15
Phase Logs 77
Phase Parameters 434
Creating 435
Deleting 440
Editing 437
Exporting 447
Mass editing 440
Overview Screen 434
Phases 424, 520
Build Environment 215
Build Phases 526
Archive Result 528
Cleanup Result 528
Cleanup Source 528
Compress Build 527
Custom Build Phase 528
Execute Script 527
Transport Deploy Script 527
Transport Source 526
Verify Build Script 526
Copying 448
Creating 425
Deleting 447
Deploy Environment 231
Deploy Phases 529
Cleanup Build Result 530
Custom Deploy Phase 531
Decompress Build Result 530
Execute Script 530
Transport Build Result 529
Verify Deploy Script 530
Editing 429
General Information 520
History 449
Importing 450
Level Phases 522
Build 523
Cleanup Work Copy 525
Custom Level Phase 525
Deploy 523
Execute Script Phase 525
Issue Tracking 524
Link File Revisions 524
Retrieve Code 522
Tag Code 523
Logs 77
Mass deleting phases 445
Mass replacing phases 442
Overview Screen 428
Phase Parameters 434
Creating 435
Deleting 440
Editing 437
Exporting 447
Mass editing 440
Overview Screen 434
Predefined Parameters 505
Build 505
Deploy 508
Level 503
Project Administration 145
Auditing Projects 247
Build Environments 209
Deploy Environments 225
Environment Parameters 240
History Log 158
Levels 187
Lifecycles 174
Project Management Options 155
Project Name 157
Project Streams 161
Projects 146

Project Management Options 155
Auditing Projects 247
Build Environments 209
Deploy Environments 225
Levels 187
Lifecycles 174
Project Name 157
Project Streams 161
Project Name 157
Project Stream Dependencies 169
Project Streams 161
Creating a Branch Project Stream 161
Deleting 168
Dependencies 169
Adding 171
Deleting 173
Editing 173
Editing 167
Hiding 168
Locking 168
Overview Screen 165
Showing 168
Unlocking 168
Viewing the Settings 166
Projects 146, 461
Cloning an Existing Project 468
Deleting 153
Editing 149
Hiding 154
History 154
Locking 153
Overview Screen 146
Package-based 461
Release-based 461
Showing 154
Unlocking 153
Viewing 148
PVCS 367
Creating 367
Deleting 372
Editing 371
History 373
Overview Screen 369

R

Reports
Level Requests 63
RSS Functionality 62
RSS URL Details 518

S

Schedules 452
Creating 452
Deleting 459
Editing 458
History 460
Overview Screen 456
Scripting Tools 394
ANT 396
Creating 394
Gradle 403
Maven2 416
NANT 410
Overview Screen 395
Search Filters 16
Creating 17
Defining 16
Editing 18
Managing 18
Modifying Search Criteria 20
Selecting 18
Secure Shell Transporters 323
Creating 323
Deleting 328
Editing 327
History 329
Overview Screen 326
Sources 97
Subversion 511
Creating 332
Deleting 338
Editing 337
Fetching Meta Properties 517
General Information 511
History 339
Overview Screen 336
Repository Connection 511
Repository Definition 511
Synchronize Build Archive 268
System Settings 255

T

Transporters 307
Creating 308
FileCopy 310
FTP 316
Overview Screen 309
Secure Shell 323

U

Unlocking
Project Streams 168
Projects 153
User Groups 280
Creating 281
Deleting 287
Editing 283
History 288
Overview Screen 282
User Interface 8
Navigation 8
Screens 9

Users 275
Creating 275
Disconnecting 279
Editing 278
History 280
Overview Screen 276

V

VCR
Creating 330
Overview Screen 331
Version Control Repositories 330
ClearCase 360
CVS 346
Git 340
PVCS 367
Subversion 332
TFVC 373
VSS (Visual Source Safe) 353
VSS 353
Creating 353
Deleting 358
Editing 357
History 359
Overview Screen 356