

Build Forge Permissions

User Guide and Command Reference

Barton Akeley

November 29, 2008

INTRODUCTION	3
1 TYPES OF PERMISSION OBJECTS	4
1.1 USERS.....	4
1.2 GROUPS	4
1.3 PERMISSIONS.....	4
1.4 ROLES	4
1.5 APPLICATIONS.....	4
1.6 OWNERSHIP.....	4
1.7 INHERITANCE.....	4
2 EXAMPLE: SEPARATE ROLE AND APPLICATION GROUPS	5
2.1 SUMMARY	5
2.2 ADVANTAGES.....	5
2.3 DISADVANTAGE.....	5
3 EXAMPLE: COMBINED ROLE AND APPLICATION GROUPS	6
3.1 SUMMARY	6
3.2 ADVANTAGES.....	6
3.3 DISADVANTAGE.....	6
4 CASE STUDY: EXECUTING A PROJECT	7
5 PERMISSIONS LIST	8
5.1 ACCESS.....	8
5.2 SERVERS.....	8
5.3 CLASSES.....	9
5.4 ENVIRONMENT.....	9
5.5 INTERFACE.....	9
5.6 SOURCE	10
5.7 PROJECTS	10
5.8 REPORTS.....	11
5.9 EXECUTION.....	12
5.10 SECURITY	12
5.11 USERS	13
5.12 TUNING	13
6 DEFAULT PERMISSIONS	14
6.1 BUILD ENGINEER	14
6.2 DEVELOPER.....	15
6.3 GUEST.....	15
6.4 OPERATOR	15
6.5 SYSTEM MANAGER.....	16
SUMMARY	17
NOTES AND WARNINGS	17
REFERENCES	17

Introduction

This article describes the Build Forge Permission model, gives examples of best practices, and provides a complete reference to the permissions of Build Forge.

1 Types of Permission Objects

1.1 Users

A user represents a single account that can be used to view/modify/delete/execute objects in a Build Forge console. A user record may have zero (0) or many groups associated to it.

1.2 Groups

Groups provide both Ownership and Permission functionality within the Build Forge permissions model. A group can be associated with 0 or many subgroups.

All objects in a Build Forge console may only be associated with one group, with the following exceptions:

- **Schedule Entries** – These objects are owned by a user, and cannot be modified by any user other than the creating user. This object is not owned by a group, hence group access does not apply to these entries.
- **Jobs** – These objects are owned by both a user and a group. They are assigned at runtime by inheriting the Project's group and the executing user or schedule. The owning group and user cannot be changed once set.

1.3 Permissions

A permission describes a specific action within Build Forge. Permissions can only be associated to Group objects. A Group may have 0 or many Permissions associated with it.

1.4 Roles

A Role is a combination of Permissions. A User's Role is determined by an inclusive union of all the Permissions associated with all of the Groups the user is in.

1.5 Applications

For the purpose of this document an Application will be a combination of Groups representing a logical business unit. This can also be considered a site, product, or team depending on the deployment requirements.

1.6 Ownership

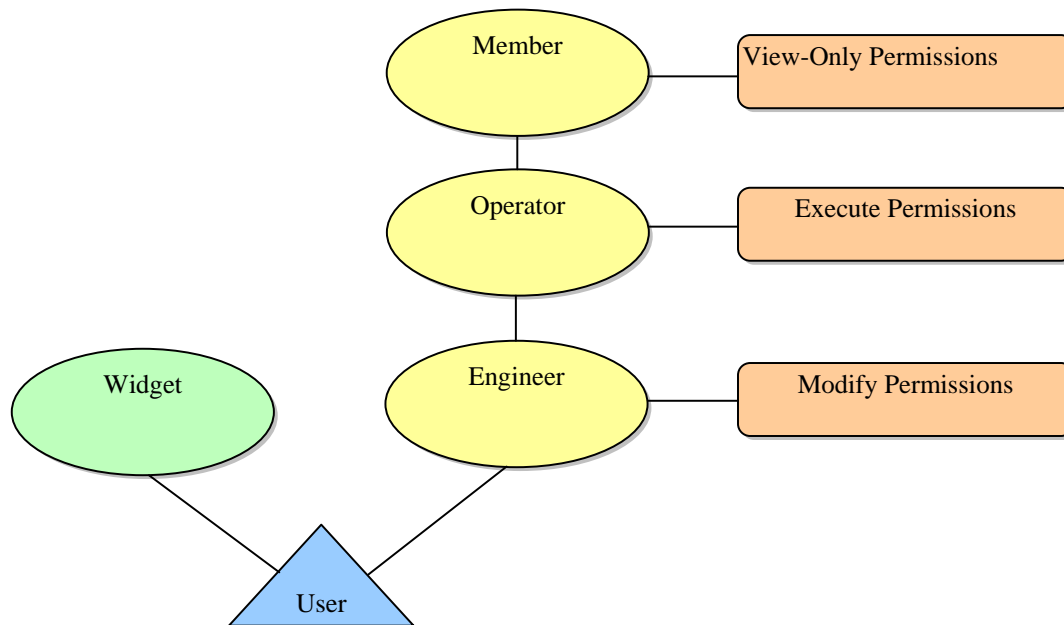
This is a type of authorization check that validates a User is a member of a Group. Ownership is considered “direct” if the User is a member of the Group. Ownership is not direct when a User is a member of a sub-group. Both direct and indirect Ownership qualify a successful ownership check.

1.7 Inheritance

Both Ownership and Permissions are inherited from a Group to its sub-group. To create a sub-group relationship between two groups, modify the parent Group's Subgroups tab to include children groups.

It is important to note that unlike the Windows ACL model, Build Forge uses an inclusive union during inheritance. It is not possible for a User to have write Permission in one Application, and read-only within another.

2 Example: Separate Role and Application Groups



2.1 Summary

In this example, The user is a member of two groups: “Widget” and “Engineer.” The user will have the inclusive union of all permissions of “Engineer” and its sub-groups. All of the Build Forge artifacts must then be assigned ownership of the “Widget” Group so that both ownership and permission checks succeed.

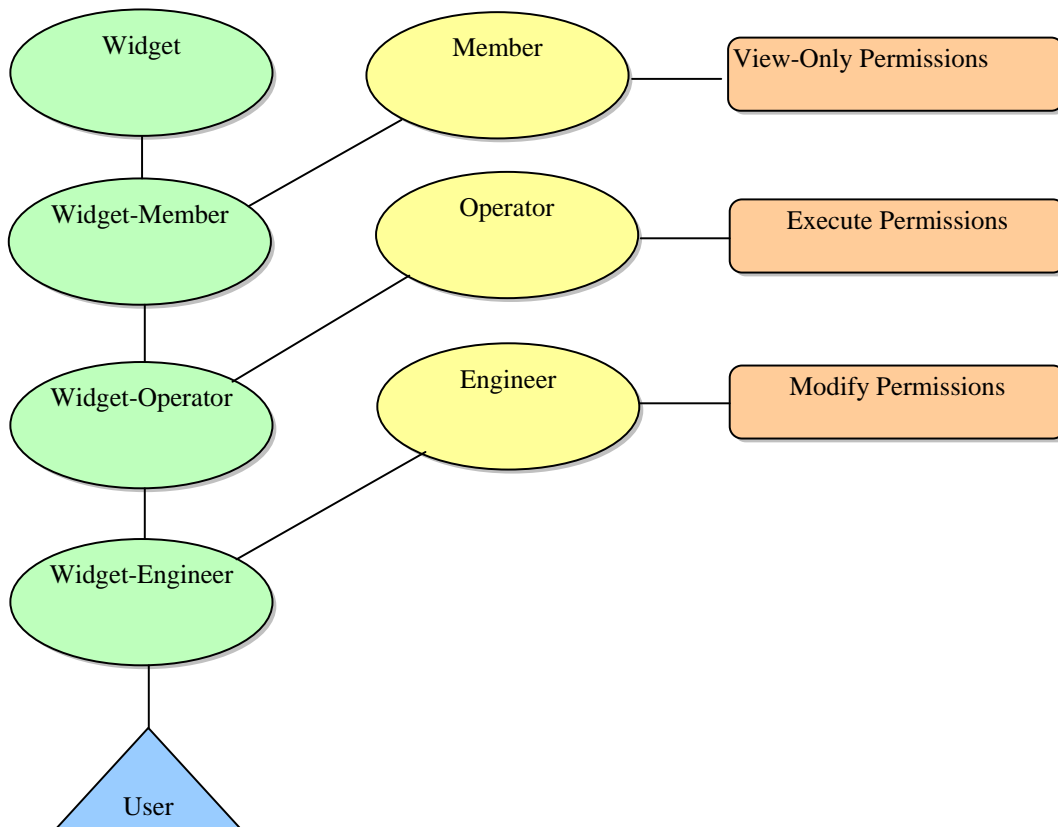
2.2 Advantages

- Role groups can be re-used across multiple Applications
- Ownership checks will always be made against a direct membership of “Widget” simplifying lookup
- User can be added to other Applications without modifying the Role

2.3 Disadvantage

- If both Role and Application change, then two groups must be modified to accommodate the changes

3 Example: Combined Role and Application Groups



3.1 Summary

In this example, the User is a direct member of only one group, “Widget-Engineer.” Both Ownership and Permission is inherited in this type of model. The User will have the inclusive union of all permissions inherited from the Widget-Engineer -> Engineer link. The User will also have indirect Ownership for Widget, Widget-Member, Widget-Member, and Widget-Operator.

3.2 Advantages

- Changing Role and Group can be done with a single Group change
- Role Groups can be reused for other Applications
- Multiple Ownership within the same Application is possible. For example, Steps can be protected as “Widget-Operator” such that “Widget-Member” cannot see them inside the project.

3.3 Disadvantage

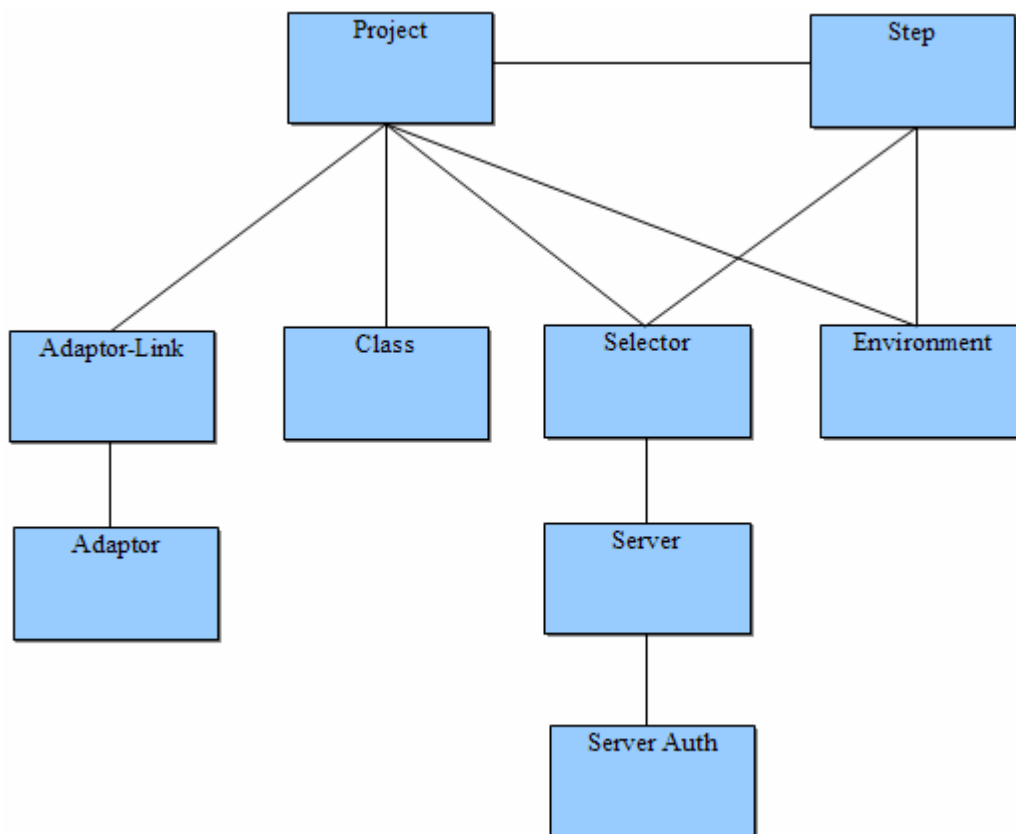
- Can lead to ambiguity across multiple Applications. For example, a User who is “Widget-Engineer” and also “Gadget-Operator” will still retain Modify permissions on the “Gadget” Application.

4 Case Study: Executing a Project

This chart traces the Ownership checks which are made when executing a Project in Build Forge.

At the root of the tree, a User must be within both the Project and (at least 1) Step owning group in order to execute a Build of the Project.

Both a Project and Step can have individual dependencies if the Step specifies non-default values. (i.e. It is not inherited from the Project)



5 Permissions List

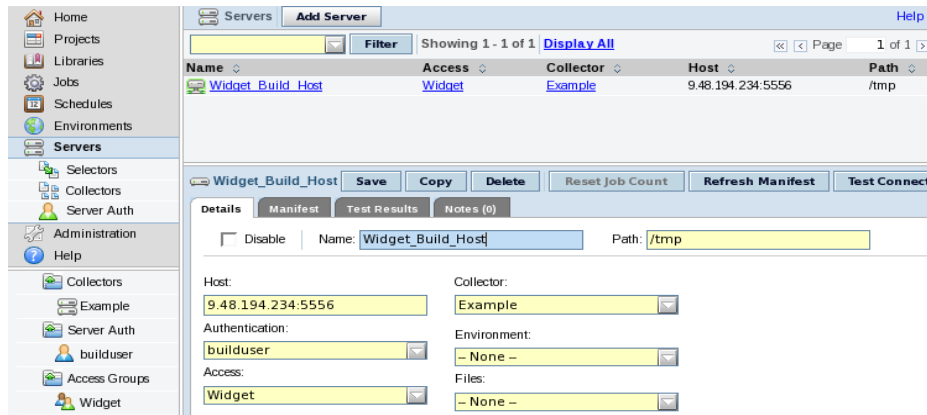
Build Forge Permissions are organized by groupings of access types. Below is the complete permission list as of 7.0.2-ifix3 arranged by grouping type.

5.1 Access



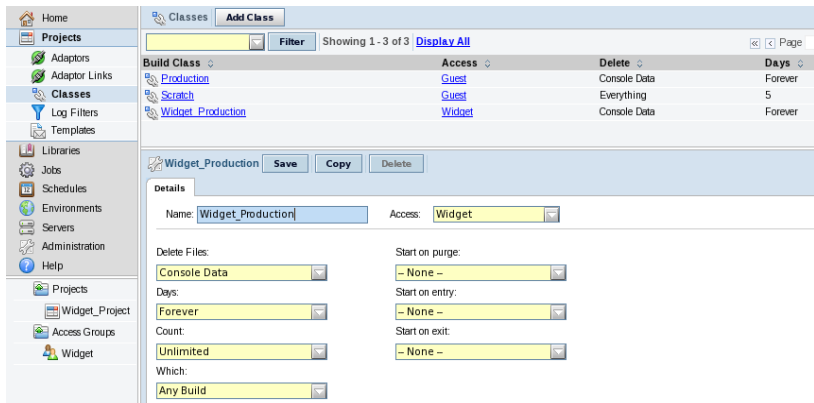
- DeleteAccess - Delete Access Groups
- AddAccess - Add Child Access Groups
- ToggleDefGroup - Toggle Whether a User can assign Access Groups to be a “default” group i.e. Assigned when the new Users logs in
- AddLevel - Add New Access Groups
- EditLevel - Edit Access Group Properties
- DeleteAccessMember - Remove Users From Access Groups
- AddAccessMember - Add Users To Access Groups

5.2 Servers



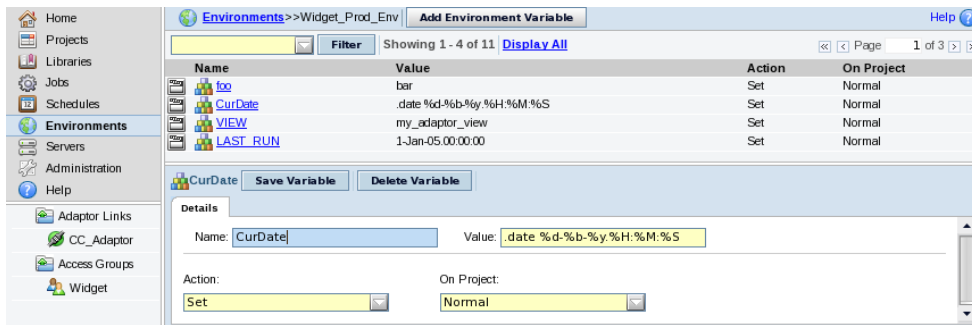
- EditSelector - Edit Selectors
- EditServer - Edit Servers
- AddCollector - Add New Collectors
- AddSelector - Add New Selectors
- DeleteServer - Delete Servers
- DeleteCollector - Delete Collectors
- DeleteSelector - Delete Selectors
- ResetServerUse – Controls the “Reset Job Count” Button on the properties view of a Server object. Without this permission the button will be grayed out.
- AddServer - Add New Servers
- EditCollector - Edit Collectors

5.3 Classes



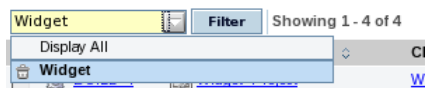
- DeleteClass - Delete Build Classes
- UpdateClass - Edit Build Classes
- AddClass - Add New Classes

5.4 Environment



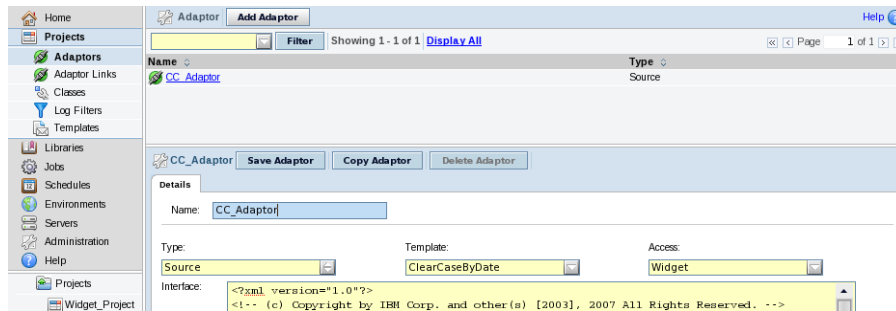
- EditEnvValue - Edit Environment Variables
- AddEnvGroup - Add New Environment Groups
- AddEnvGroupValue - Add Variables To Environment Groups
- DeleteEnvGroupValue - Remove Variables From Environment Groups
- MoveEnvValue - Move Environment Variables
- DeleteEnvGroup - Delete Environment Variable Groups
- ModifyEnvGroup - Edit Environment Group

5.5 Interface



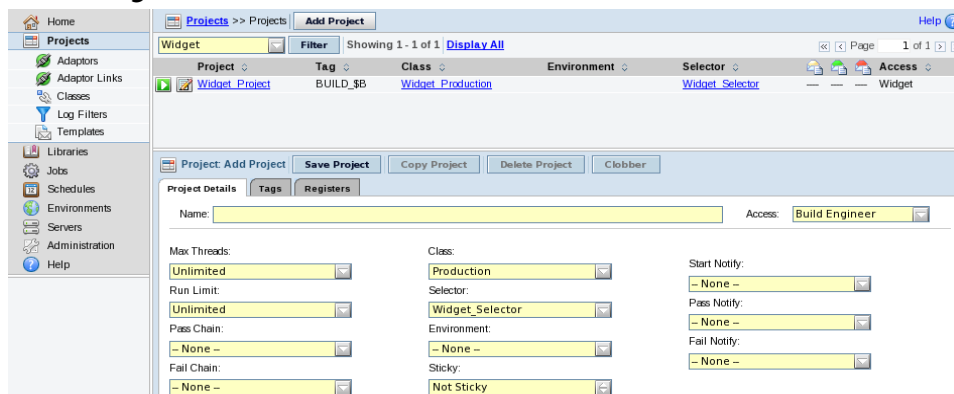
- AddView - Create UI Filters
- DeleteView - Delete UI Filters
- DeleteViewEntry – This is deprecated and no longer used by Build Forge
- AddViewEntry- This is deprecated and no longer used by Build Forge

5.6 Source



- DeleteSourceServer - Deprecated: No longer used by Build Forge
- DeleteLink - Delete Adaptor Links
- AddSourceServer – Deprecated: No longer used by Build Forge
- MakeRelease – Deprecated: No longer used by Build Forge
- AddInterface - Add Adaptors
- DeleteSourceModule – Deprecated: No longer used by Build Forge
- DeleteInterface - Delete Adaptors
- AddLink - Add Adaptor Links
- AddSourceModule – Deprecated: No longer used by Build Forge
- ToggleLink – Ability to set state of Adaptor Link (Active/Inactive/Debug)
- EditSourceServer - Deprecated: No longer used by Build Forge
- DeleteChange – Deprecated: No longer used by Build Forge
- EditInterface - Edit Adaptors

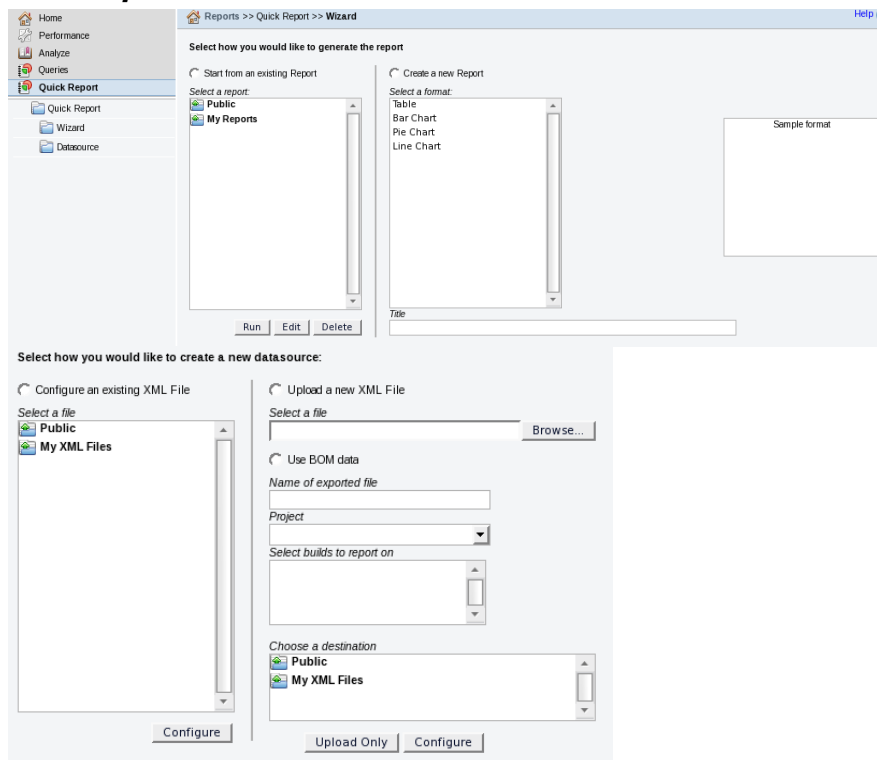
5.7 Projects



- EditProjectTagVar - Edit Project Tag Variables
- UpdateProjectStep - Edit Step Attributes
- AddFilterSetEntry – Create new regex entries in Log Filters (Projects> Log Filters)
- ClobberProject – Clobber button deletes a Project and all associated Jobs
- AddNote – Create a new entry in the “Notes” section listed in the bottom left pane of the Job details view
- AddProject - Add New Projects
- DeleteFilterSetEntry – Delete existing regex entries in a Log Filter (Projects> Log Filters)
- DeleteProjectStep - Delete Steps From Projects
- RestoreCheckpoint – Deprecated: No longer used by Build Forge
- MoveProjectStep - Move Project Steps
- DeleteCheckpoint - Deprecated: No longer used by Build Forge
- DelProjectReg - Delete Project Registers
- EditFilterSetEntry - Edit existing regex entries in a Log Filter (Projects> Log Filters)
- DeleteProjectTagVar - Delete Project Tag Variables

- EditProjectReg - Edit Project Registers (Register Tab on the Project details view)
- AddProjectReg - Add Register Data to Projects (Register Tab on the Project details view)
- AddProjectTagVar - Add Project Tag Variables
- UpdateProject - Edit Project Settings
- AddCheckpoint - Deprecated: No longer used by Build Forge.
- AddProjectStep - Add Steps To Projects
- AddFilterSet - Create Log Filters
- DeleteFilterSet - Delete Log Filters
- DeleteNote - Delete Build Notes
- ActivateProject - Enable and Disable Projects. This can only be done via the API and affects the field “bf_projects.bf_active”
- DeleteProject - Delete Projects

5.8 Reports¹



- EditPublicReports - Edit Public Reports.
- SavePublicReports - Save Public Reports.
- DeletePublicReports - Delete Public Reports.
- ConfigurePublicXML - Configure Public XML Files into Datasources
- DeletePublicXML - Delete Public XML Files. Users can create XML datasources
- ReadPublicReports - Read Public Reports
- SavePublicXML - Save Public XML Files
- RenamePublicXML - Rename Public XML Files

¹This group controls access to features which require an additional license key to access. Please see Online Help topic “Working With Reports> Quick Reports” for more details.

5.9 Execution

Tag	Projects and Libraries	Class	State	Result	Date	Runtime	Owner
BUILD_4	Widget_Project	Widget_Production	Completed	Failed	10/24/08 03:41PM	0:00:00	Joe Forge
BUILD_3	Widget_Project	Widget_Production	Completed	Failed	10/24/08 03:18PM	0:00:00	Joe Forge
BUILD_2	Widget_Project	Widget_Production	Completed	Failed	10/24/08 02:36PM	0:00:00	Joe Forge
BUILD_1	Widget_Project	Widget_Production	Completed	Failed	10/24/08 01:52PM	0:00:00	Joe Forge

- ExecStepPerm - Execute Steps for which the User is not within the owning Group of the Step
- DeleteBuildWaiting - Cancel Waiting Builds
- DeselectSteps - Select Steps At Runtime
- ActivateSched - Enable and Disable Schedules
- ExecServerAuthPerm – Ownership for Server Auth is skipped at Project Execution
- PauseStep - Pause Builds At Steps
- AddSchedule - Add Schedule Entries
- CancelStep - Cancel Running Steps
- RunBuild - Execute Builds
- ChangeBuildClass - Change Build Class
- UnlockBuild - Unlock Locked Builds
- ImportAsAdmin – Not currently implemented: Reserved for future use
- EditSchedule - Edit Schedule Entries
- PurgeBuild - Purge Builds
- ExportProject - Export Build Projects
- ImportProject - Import Build Projects
- ViewLog - View Build Logs
- LockBuild - Lock Builds
- CloneBuild - Clone Builds
- CancelRunning - Cancel Running Builds
- EditStartTagVars - Alter Tag Variables On Build Start
- RunStep - Resume Paused Steps
- SetState – Not currently implemented, reserved for future use.
- DeleteSchedule - Delete Schedule Entries
- ImportAsUser - Not currently implemented, reserved for future use
- RestartBuild - Restart Builds

5.10 Security

- SwitchUser - Switch to Another User
- DeleteGeo - Delete Geo Locations (GDD only)
- EditServerAuth - Edit Server Authentication
- EditHiddenTags – Modify a Environment Variables marked as “Hidden”
- ActivateServer - Enable and Disable Servers
- DeleteServerAuth - Delete Server Authentication
- EditSecurity – Deprecated: No longer used by Build Forge
- SaveGeo - Save Geo Locations (GDD only)
- UpdatePermMember – Assign Permissions to Access Groups
- ExpireUserPasswords – Set “Password Expires” Attribute on User record
- AddServerAuth - Add Server Authentication

5.11 Users

The screenshot displays the 'Users' management page. At the top, it indicates '1 of 1000 License Seats Used'. A table lists the following users:

Name	User name	Email	Limit	Activity	TZ
Joe Forge	jf	jforge@bf.com	0	457:28:03	CDT
Root User	root	root@localhost	0	0:00:03	CDT

Below the table, the details for 'Joe Forge' are shown. The 'Name' field is 'Joe Forge' and the 'Email' is 'jforge@bf.com'. Other fields include 'User name' (jf), 'Password' (empty), 'Limit' (Unlimited), 'Time Zone' (Central Daylight Time), 'Verified' (empty), 'Priority Login' (No), 'Date Format' (01/01/03 08:30PM), 'Language' (English US), 'Password Expires' (Yes), 'Uses screen reader' (No), 'Calendar Start Day Of Week' (Sunday), and 'Truncation' (Default Value).

- DeleteUser - Delete Users
- UpdateAccessMember – Add/Remove Users from Access Groups
- AddUser - Add New Users
- UpdateUser - Edit User Settings

5.12 Tuning

The screenshot displays the 'Templates' management page. At the top, it indicates 'Showing 1 - 4 of 12'. A table lists the following templates:

Description	Project	Step	Subject	Language
Change Artifact Match Message			\$(PROJECTNAME)-\$.facts \$(ARTIFACTS)	English US
Filter Match Message			\$(PROJECTNAME)-\$.n\$(FILTERPATTERN)	English US
Job Break Message			\$(PROJECTNAME)-\$. Stopped at .break	English US
Job Fail Message			\$(PROJECTNAME)-\$.TAG] Failed	English US

Below the table, the details for 'Filter Match Message' are shown. The 'Key' is 'Filter Match Message', 'Description' is 'Filter Match Message', and 'Language' is 'English US'. Other fields include 'Subject' (\$(PROJECTNAME)-\$.TAG]-\$.STE), 'From' (Build Forge Management Conso), 'Project' (- All -), 'Step' (- All -), and 'Body' (Job \$(TAG) of project \$(PROJECTNAME) \$(RUNACTION) on \$(SERVER) at \$(START). Step \$(STEPNAME) triggered filter pattern \$(FILTERPATTERN) of filter \$(FILTER). You can view

- DeleteTemplate - Delete Notification Templates
- UpdateTemplate - Edit Notification Templates
- UpdateSysconfig - Edit System Parameters
- EditTimeZones - Edit Time Zone Information (via API only via TimeZone object)
- DeleteTimeZones – Deprecated: Previously controlled API deletion of time zones in BF database
- AddTimeZones – Deprecated: Previously controlled API creation of new time zones in BF Database
- AddTemplate - Add Notification Templates

6 Default Permissions

6.1 Build Engineer

ActivateProject	DeleteChange	EditServerAuth
ActivateSched	DeleteCheckpoint	EditStartTagVars
ActivateServer	DeleteClass	ExecStepPerm
AddAccess	DeleteCollector	ExpireUserPasswords
AddAccessMember	DeleteEnvGroup	ExportProject
AddCheckpoint	DeleteEnvGroupValue	ImportAsAdmin
AddClass	DeleteFilterSet	ImportAsUser
AddCollector	DeleteFilterSetEntry	ImportProject
AddEnvGroup	DeleteInterface	LockBuild
AddEnvGroupValue	DeleteLink	MakeRelease
AddFilterSet	DeleteNote	ModifyEnvGroup
AddFilterSetEntry	DeleteProject	MoveEnvValue
AddInterface	DeleteProjectStep	MoveProjectStep
AddLevel	DeleteProjectTagVar	PauseStep
AddLink	DeletePublicReports	PurgeBuild
AddNote	DeletePublicXML	ReadPublicReports
AddProject	DeleteSchedule	RenamePublicXML
AddProjectReg	DeleteSelector	ResetServerUse
AddProjectStep	DeleteServer	RestartBuild
AddProjectTagVar	DeleteServerAuth	RestoreCheckpoint
AddSchedule	DeleteTemplate	RunBuild
AddSelector	DeleteUser	RunStep
AddServer	DeleteView	SavePublicReports
AddServerAuth	DeleteViewEntry	SavePublicXML
AddTemplate	DelProjectReg	SetState
AddUser	EditCollector	ToggleDefGroup
AddView	EditEnvValue	ToggleLink
AddViewEntry	EditFilterSetEntry	UnlockBuild
CancelRunning	EditHiddenTags	UpdateAccessMember
CancelStep	EditInterface	UpdateClass
ChangeBuildClass	EditLevel	UpdatePermMember
ClobberProject	EditProjectReg	UpdateProject
CloneBuild	EditProjectTagVar	UpdateProjectStep
ConfigurePublicXML	EditPublicReports	UpdateSysconfig
DeleteAccess	EditSchedule	UpdateTemplate
DeleteAccessMember	EditSelector	UpdateUser
DeleteBuildWaiting	EditServer	ViewLog

6.2 Developer

AddEnvGroup	DeleteEnvGroup	ExportProject
AddEnvGroupValue	DeleteEnvGroupValue	ImportAsUser
AddFilterSet	DeleteFilterSet	ModifyEnvGroup
AddFilterSetEntry	DeleteFilterSetEntry	MoveEnvValue
AddNote	DeleteNote	MoveProjectStep
AddProject	DeleteProject	PauseStep
AddProjectReg	DeleteProjectStep	PurgeBuild
AddProjectStep	DeleteTemplate	ReadPublicReports
AddProjectTagVar	DeleteView	RestartBuild
AddTemplate	DeleteViewEntry	RunBuild
AddView	DelProjectReg	RunStep
AddViewEntry	DeselectSteps	UpdateProject
CancelRunning	EditEnvValue	UpdateProjectStep
CancelStep	EditFilterSetEntry	UpdateTemplate
ChangeBuildClass	EditProjectReg	ViewLog
CloneBuild	EditProjectTagVar	
DeleteBuildWaiting	EditStartTagVars	

6.3 Guest

CancelRunning
CloneBuild
PurgeBuild
RestartBuild
RunBuild
ViewLog

6.4 Operator

CancelRunning	DeleteAccess	ExpireUserPasswords
CloneBuild	DeleteAccessMember	ImportAsAdmin
PurgeBuild	DeleteChange	ImportAsUser
RestartBuild	DeleteInterface	MakeRelease
RunBuild	DeleteLink	ReadPublicReports
ViewLog	DeleteNote	SetState
Security	DeleteServer	SwitchUser
AddAccess	DeleteUser	ToggleDefGroup
AddAccessMember	DeselectSteps	ToggleLink
AddInterface	EditHiddenTags	UpdateAccessMember
AddLevel	EditInterface	UpdatePermMember
AddLink	EditLevel	UpdateUser
AddNote	EditSecurity	
AddUser	EditServer	

6.5 System Manager

ActivateProject	DeleteAccessMember	EditServerAuth
ActivateSched	DeleteBuildWaiting	EditStartTagVars
ActivateServer	DeleteChange	ExecStepPerm
AddAccess	DeleteClass	ExpireUserPasswords
AddAccessMember	DeleteCollector	ExportProject
AddClass	DeleteEnvGroup	ImportAsAdmin
AddCollector	DeleteEnvGroupValue	ImportAsUser
AddEnvGroup	DeleteFilterSet	LockBuild
AddEnvGroupValue	DeleteFilterSetEntry	MakeRelease
AddFilterSet	DeleteGeo	ModifyEnvGroup
AddFilterSetEntry	DeleteInterface	MoveEnvValue
AddInterface	DeleteLink	MoveProjectStep
AddLink	DeleteNote	PauseStep
AddNote	DeleteProject	PurgeBuild
AddProject	DeleteProjectStep	ReadPublicReports
AddProjectReg	DeleteProjectTagVar	ResetServerUse
AddProjectStep	DeleteSchedule	RestartBuild
AddProjectTagVar	DeleteSelector	RunBuild
AddSchedule	DeleteServerAuth	RunStep
AddSelector	DeleteTemplate	SaveGeo
AddServer	DeleteView	SetState
AddServerAuth	DeleteViewEntry	ToggleLink
AddTemplate	DelProjectReg	UnlockBuild
AddUser	DeselectSteps	UpdateAccessMember
AddView	EditCollector	UpdateClass
AddViewEntry	EditEnvValue	UpdateProject
CancelRunning	EditFilterSetEntry	UpdateProjectStep
CancelStep	EditInterface	UpdateSysconfig
ChangeBuildClass	EditProjectReg	UpdateTemplate
ClobberProject	EditProjectTagVar	ViewLog
CloneBuild	EditSchedule	
DeleteAccess	EditSelector	

Summary

Build Forge combines permissions and ownership checks, where both can be inherited to a subgroup. Combining both checks together can enable a role-based permissions model supporting multiple groups and geographies.

Notes and warnings

- [Technote 1304886](#) - How to flush permissions and group cache in Build Forge
- [Technote 1294408](#) - About the selection process for servers (includes information about access checks)

References

The following was used in reference or as other sources of information:

- Other than the source code, refer to the [IBM Rational Build Forge Online Help](#) > **Administering the Administration Console** > **Security Overview**