

# Grouper installer non interactive mode

<a href="#">Wiki Home</a>	<a href="#">Grouper Release Announcements</a>	<a href="#">Grouper Guides</a>	<a href="#">Grouper Deployment Guide</a>	<a href="#">Community Contributions</a>	<a href="#">Internal Developer Resources</a>
---------------------------	---	--------------------------------	--	---	--

 For Grouper 2.5+ see the new approach for [Grouper 2.5+ packaging and versioning strategy](#)

- [Grouper installer non interactive apply patches run tests example](#)
- [Grouper installer non interactive auto apply patches example](#)

The grouper installer can run in non interactive mode and you can answer all the questions in the `grouper.installer.properties`.

To run with a `grouper.installer.properties` create the config file and run like this. Note on windows use a semicolon instead of a colon

```
UNIX
java -cp .:grouperInstaller.jar edu.internet2.middleware.grouperInstaller.GrouperInstaller

WINDOWS
java -cp .;grouperInstaller.jar edu.internet2.middleware.grouperInstaller.GrouperInstaller
```

See the use defaults as much as possible in the `grouper.installer.properties` (copy from `grouper.installer.example.properties` to the `grouper.installer.properties`, then look at the commonly changed items and move them to the `grouper.installer.properties` as needed)

```
#####
## Autorun properties
##
## If you uncomment one of these properties it will be used as empty, only uncomment to use
##
#####

#### set this to true to try to use defaults for everything. Only things without default values will need to
be set
grouperInstaller.autorun.useDefaultsAsMuchAsAvailable = false

##### AUTORUN PROPERTIES WITH NO DEFAULT OR ARE COMMONLY CHANGED
## Note: not all of them need to be filled out for all operations

# autorun Do you want to 'install' a new installation of grouper, 'upgrade' an existing installation
# or 'patch' an existing installation\n"
# (enter: 'install', 'upgrade', 'patch' or blank for the default)
# default is controlled in a config in this config file: grouperInstaller.default.installOrUpgrade
# grouperInstaller.autorun.actionEgInstallUpgradePatch =

# autorun, What do you want to install/upgrade/patch? api, ui, ws, or psp?
# note, default is configured in this file: grouperInstaller.default.appToUpgrade
# grouperInstaller.autorun.appToUpgrade =

# autorun, Where is the grouper api/ui/ws/psp installed for upgrades / patching?
# grouperInstaller.autorun.grouperWhereInstalled =

# autorun grouper system password (its not secure to have a plain text pass in a config file)
# grouperInstaller.autorun.grouperSystemPassword =
```

```

# autorun db url
# grouperInstaller.autorun.dbUrl =

# autorun db user
# grouperInstaller.autorun.dbUser =

# autorun db pass (note, it is not good security to have plaintext passwords in text config files)
# grouperInstaller.autorun.dbPass =

# autorun Do you want to init the database (delete all existing grouper tables, add new ones) (t|f)?
# grouperInstaller.autorun.deleteAndInitDatabase = f

# autorun, uncomment to make work, Enter in the Grouper install directory (note: better if no spaces or special
chars)
# default is controller in a config in this config file: grouperInstaller.default.installDirectory
# if that is blank or not configured, default is current working directory
# grouperInstaller.autorun.installDirectory =

# autorun, uncomment to make work, Enter in a Grouper temp directory to download tarballs
# default is controller in a config in this config file: grouperInstaller.default.tarballDirectory
# if that is blank or not configured, default is current working directory
# grouperInstaller.autorun.tarballDirectory =

# autorun patchCreator: Enter a description for this patch, e.g. GRP-123: fixes a problem with such and such:
[required]
# grouperInstaller.autorun.patchDescription =

# autorun patchCreator: Enter the risk level for the patch: (low|medium|high): [required]
# grouperInstaller.autorun.patchRiskLevel =

# autorun patchCreator: Is this a security patch? (t|f): [t]
# grouperInstaller.autorun.patchSecurity =

# autorun What is the location of your tomcat server.xml for the UI?
# Note, if you dont use tomcat just leave it blank or type 'blank':
# grouperInstaller.autorun.locationOfTomcatServerXml =

# autorun patch action: What do you want to do with patches (install, revert, status)
# grouperInstaller.autorun.patchAction =

# autorun patch Enter the comma separated list of files to make a patch from: [required]
# grouperInstaller.autorun.patchFilesCommaSeparated =

# autorun Enter any patch names that are dependencies that you know of (comma separated), or blank for none:
\n");
# grouperInstaller.autorun.patchNameDependenciesCommaSeparated =
# It is NOT detected that your patch requires restart, please confirm this, does it require restart (t|f)? [f]
# grouperInstaller.autorun.patchOverrideDoesntRequireRestart = t
# Enter a Jira key (e.g. GRP-123) for this patch: [required]
# (note, will try to get this from patch description, if its there, this can be blank)
# grouperInstaller.autorun.patchJiraKey =
##### AUTORUN PROPERTIES THAT HAVE A DEFAULT VALUE

# autorun rebuild ui after having been built
# grouperInstaller.autorun.rebuildUiAfterHavingBeenBuilt = t

# autorun rebuild client after having been built [t]
# grouperInstaller.autorun.rebuildClientAfterHavingBeenBuilt =

```

```

# autorun rebuild grouper api after having been built [t]
# grouperInstaller.autorun.rebuildGrouperApiAfterHavingBeenBuilt =

# autorun rebuild PSP after having been built [t]
# grouperInstaller.autorun.rebuildPspAfterHavingBeenBuilt =

# autorun rebuild subject API after having been built [t]
# grouperInstaller.autorun.rebuildSubjectApiAfterHavingBeenBuilt =

# autorun uncomment to make work, The Grouper WS has been built in the past, do you want it rebuilt? (t|f)
# note, this default can be set in this config file grouperInstaller.default.ws.rebuildIfBuilt, or default true
# grouperInstaller.autorun.rebuildWsIfBuiltAlready =

# autorun if the log directory of UI should be set
# grouperInstaller.autorun.setLogDirOfUi = t

# autorun log dir of UI
# grouperInstaller.autorun.uiLogDir =

# autorun path of sh command if not in a standard path
# grouperInstaller.autorun.pathOfShCommandIfNotFound =

# autorun if tomcat should be stopped anyways when detected not running
# grouperInstaller.autorun.stopTomcatAnyway = t

# autorun if tomcat should be starts anyways when detected running
# grouperInstaller.autorun.startTomcatAnyway = t

# autorun if the ports of tomcat should be checked when stopping
# grouperInstaller.autorun.stopTomcatCheckPorts = t

# autorun if the ports of tomcat should be checked when starting
# grouperInstaller.autorun.startTomcatCheckPorts = t

# autorun if continue after tomcat error
# grouperInstaller.autorun.continueAfterTomcatError = t

# autorun set grouper system password in tomcat users xml
# grouperInstaller.autorun.setGrouperSystemPasswordInTomcatUsers = t

# autorun conflicting jar continue (you can just uncomment, doesnt need a value)
# grouperInstaller.autorun.conflictingJarContinue =

# autorun branch to create a patch for, default to the one the version is above
# grouperInstaller.autorun.branchToCreatePatchFor =

# autorun branch to create a patch for PSP, default to the one the version is above
# grouperInstaller.autorun.branchForPspToCreatePatchFor =

# autorun backup your files (uncomment to make work): You should backup your files and database before you
start. Press <enter> to continue.
# grouperInstaller.autorun.backupFiles =
# any running processes: Are there any running processes using this installation? tomcats? loader? psp?
etc? (t|f)? [f]:
# grouperInstaller.autorun.runningProcesses = f

```

```
# autorun, uncomment to make work, If you have edited the Owasp.CsrfGuard.overlay.properties please merge the
changes to the new file
# grouperInstaller.autorun.continueAfterEditedOwaspCsrfGuard =

# autorun, uncomment to make work, If you customized the web.xml please merge your changes back in
# grouperInstaller.autorun.continueAfterMergeWebXml =

# autorun, uncomment to make work, Merge nav.properties
# grouperInstaller.autorun.continueAfterNavProperties =

# autorun, uncomment to make work, Merge media.properties
# grouperInstaller.autorun.continueAfterMediaProperties =

# autorun, Remove old keys nav.properties
# grouperInstaller.autorun.removeOldKeysFromNavProperties = t

# autorun, Remove old keys media.properties
# grouperInstaller.autorun.removeOldKeysFromMediaProperties = t

# autorun, view list of files changed in copy
# grouperInstaller.autorun.viewListOfFilesChangedInCopy = f

# autorun, uncomment to make work, continue after merging sources.xml
# grouperInstaller.autorun.continueAfterMergingSourcesXml =

# autorun, You are upgrading from pre API version 2.0.0, do you want to run Unresolvable Subject Deletion
Utility (USDU) (recommended) (t|f)? [t]
# grouperInstaller.autorun.runUsduPre2.0.0 = t

# autorun, You are upgrading from after API version 2.0.0, so you dont have to do this, but do you want to run
Unresolvable Subject Deletion Utility (USDU) (not recommended) (t|f)? [f]
# grouperInstaller.autorun.runUsduPost2.0.0 = f

# autorun, ERROR, cannot find grouper version in grouper jar file, do you want to continue? (t|f)? [f]:
# grouperInstaller.autorun.shouldContinueAfterNoGrouperVersionFound = f

# autorun, You are upgrading from pre API version 2.0.0, do you want to resolve all group subjects
(recommended) (t|f)? [t]: ");
# grouperInstaller.autorun.resolveGroupSubjectsPre2.0.0 = t

# autorun, You are upgrading from after API version 2.0.0, so you dont have to do this,
# but do you want to resolve all group subjects (not recommended) (t|f)? [f]:
# grouperInstaller.autorun.resolveGroupSubjectsPost2.0.0 = f

# autorun You are upgrading from pre API version 2.1.0, do you want to
# see if you have rules with ruleCheckType: flattenedPermission* (recommended) (t|f)? [t]: ");
# grouperInstaller.autorun.seeRulesFlattenedPermissionsPre2.1.0 = t

# autorun You are upgrading from after API version 2.1.0, so you dont have to do this,
# but do you want to see if you have rules with ruleCheckType: flattenedPermission* (not recommended) (t|f)?
[f]:
# grouperInstaller.autorun.seeRulesFlattenedPermissionsPost2.1.0 = f

# autorun Error getting count of rules, would you like to continue (t|f)? [t]:
```

```

# grouperInstaller.autorun.shouldContinueAfterErrorCountFlattenedRules = t

# autorun You have flattenedPermission rules that need to be removed. You need to look in the view
# grouper_rules_v and notify the owners and remove these rules. Do you want to continue (t|f)? [t]:
# grouperInstaller.autorun.shouldContinueAfterFoundFlattenedRules = t

# autorun You are upgrading from pre API version 2.2.0,
# do you want to run the 2.2 upgrade GSH script (recommended) (t|f)? [t]:
# grouperInstaller.autorun.run2.2gshUpgradeScriptPre2.2.0 = t

# autorun You are upgrading from after API version 2.2.0, so you dont have to do this,
# but do you want to run the 2.2 upgrade GSH script (not recommended) (t|f)? [f]:
# grouperInstaller.autorun.run2.2gshUpgradeScriptPost2.2.0 = f

# autorun You are upgrading from pre API version 2.2.1, do you want to
# run the 2.2.1 upgrade GSH script (recommended) (t|f)? [t]:
# grouperInstaller.autorun.run2.2.1gshUpgradeScriptPre2.2.1 = t

# autorun You are upgrading from after API version 2.2.1, so you dont have to do this
# but do you want to run the 2.2.1 upgrade GSH script (not recommended) (t|f)? [f]:
# grouperInstaller.autorun.run2.2.1gshUpgradeScriptPost2.2.1 = f

# autorun Error: we tried to upgrade the database but it didnt work, would you like to continue skipping DDL
# (t|f)?
# grouperInstaller.autorun.shouldContinueIfErrorUpgradingDatabase = f

# autorun Review the script(s) above if there are any, do you want the upgrader to run it to upgrade the DDL
# for you (t|f)? [t]:
# grouperInstaller.autorun.shouldRunDdlScript = t

# autorun uncomment to make work There are multiple related jars for *jarName* GrouperInstallerUtils.
# toStringForLog(relatedJars));
# There should be only one, remove the others, if this is a mistake, then you need to\n rename the prefix so
# they are different, and report to the Grouper team
# grouperInstaller.autorun.continueAfterFoundMultipleJars =

# autorun uncomment to make work Compare you old ehcache.xml with the new ehcache.xml file
# grouperInstaller.autorun.continueAfterCompareEhcache =

# autorun remove redundant properties from grouper client
# grouperInstaller.autorun.removeRedundantPropetiesFromGrouperClient = t

# autorun remove redundant properties from grouper properties
# grouperInstaller.autorun.removeRedundantPropetiesFromGrouperProperties = t

# autorun remove redundant properties from grouper hibernate properties
# grouperInstaller.autorun.removeRedundantPropetiesFromGrouperHibernateProperties = t

# autorun remove redundant properties from grouper loader properties
# grouperInstaller.autorun.removeRedundantPropetiesFromGrouperLoaderProperties = t

# autorun remove redundant properties from grouper ws properties
# grouperInstaller.autorun.removeRedundantPropetiesFromGrouperWsProperties = t

# autorun remove redundant properties from subject properties
# grouperInstaller.autorun.removeRedundantPropetiesFromSubjectProperties = t

```

```

# autorun, uncomment to make work, Error: cant find directory for patch, press <enter> to continue
# grouperInstaller.autorun.continueAfterCantFindPatchDir =
# This patch requires all processes that user Grouper to be stopped.
# Please stop these processes if they are running and press <enter> to continue...
# grouperInstaller.autorun.continueAfterStoppingGrouperProcesses =

# autorun, when reverting Would you like to revert patch " + keyBase + " (t|f)? [f]:
# grouperInstaller.autorun.revertPatch = f

# autorun, uncomment to make work Cannot install patch " + keyBase + " since it is dependent on a patch
# which is not installed: " + dependency continue?
# grouperInstaller.autorun.continueAfterPatchDependencyFails =

# autorun Would you like to install patch " + keyBase + " (t|f)? [t]:
# grouperInstaller.autorun.installPatch = t

# autorun uncomment to make work This patch requires all processes that user Grouper to be stopped.
# Please stop these processes if they are running and press <enter> to continue...
# grouperInstaller.autorun.continueAfterPatchStopProcesses =

# autorun Would you like to be prompted about this patch next time? (t|f)? [t]:
# grouperInstaller.autorun.promptAboutPatchNextTime = t

# autorun When creating patch if grouper source is already downloaded, use local source?
# grouperInstaller.autorun.createPatchDownloadSourceUseLocalIfExist = f

# autorun WS uncomment to make work, Since patches were applied/reverted, you should delete files
# in your app server work directory,
# in tomcat it is named 'work'. Hit <enter> to continue:
# grouperInstaller.autorun.continueAfterDeleteWsWorkDirectory =

# autorun UI uncomment to make work, Since patches were applied/reverted, you should delete files
# in your app server work directory,
# in tomcat it is named 'work'. Hit <enter> to continue:
# grouperInstaller.autorun.continueAfterDeleteUiWorkDirectory =

# autorun (blank is default) Enter the default IP address for checking ports (just hit enter to accept the
# default
# unless on a machine with no network, might want to change to 127.0.0.1): [0.0.0.0]:
# grouperInstaller.autorun.defaultIpAddressForPorts =

# autorun (blank is default) Do you want to use the default and included hsqldb database (t|f)? [t]:
# grouperInstaller.autorun.useBuiltInHsql =

# autorun uncomment to make work, Note: you need to change the search sql in the
# jdbc source in the grouperApi/conf/sources.xml... the change is in the comments in that file
# Ready to continue? (t|f)? [t]
# grouperInstaller.autorun.dbContinueAfterChangeSourcesXmlForPostgresSqlServer =

# autorun uncomment to make work, Do you want to install the provisioning service provider (t|f)? [t]:
# grouperInstaller.autorun.installPsp =

# autorun uncomment to make work, Do you want to set " + specify + "gsh script to executable (t|f)? [t]:
# grouperInstaller.autorun.setGshScriptsToExecutable =

```

```
# autorun uncomment to make work, Do you want to run dos2unix on gsh.sh (t|f)? [t]:
# grouperInstaller.autorun.dos2unixOnGsh =

# autorun uncomment to make work, Do you want to set the tomcat memory limit (t|f)? [t]:
# grouperInstaller.autorun.setTomcatMemoryLimit =

# autorun uncomment to make work, Do you want to set tomcat scripts to executable (t|f)? [t]:
# grouperInstaller.autorun.setTomcatScriptsToExecutable =

# autorun uncomment to make work, Do you want to run dos2unix on tomcat sh files (t|f)? [t]:
# grouperInstaller.autorun.runDos2unixOnTomcatFiles =

# autorun uncomment to make work, What ports do you want tomcat to run on (HTTP, JK, shutdown): (will default
to whats in config file)
# grouperInstaller.autorun.tomcatPorts =

# autorun uncomment to make work, The tomcat HTTP/JK port is in use or unavailable: do you want to pick
different ports? (t|f)
# grouperInstaller.autorun.pickDifferentPortIfInUse = f

# autorun uncomment to make work, Do you want to set URIEncoding to UTF-8 in tomcat server.xml <Connector>
elements (t|f)? [t]
# note this default can be set in this config file, grouperInstaller.default.ui.setTomcatUriEncoding, or
defaults to true if blank
# grouperInstaller.autorun.setUriEncodingToUtf8inServerXml =

# autorun uncomment to make work, do you want to use local UI download, tar, etc, default true
# grouperInstaller.autorun.useLocalUiDownloadTarEtc =

# autorun uncomment to make work, do you want to use local WS download, tar, etc, default true
# grouperInstaller.autorun.useLocalWsDownloadTarEtc =

# autorun uncomment to make work, do you want to use local API download, tar, etc, default true
# grouperInstaller.autorun.useLocalApiDownloadTarEtc =

# autorun uncomment to make work, do you want to use local PSP download, tar, etc, default true
# grouperInstaller.autorun.useLocalPspDownloadTarEtc =

# autorun uncomment to make work, do you want to use local ant/tomcat/etc download, tar, etc, default true
# grouperInstaller.autorun.useLocalToolsDownloadTarEtc =

# autorun uncomment to make work, Do you want to add quickstart subjects to DB (t|f)? [t]:
# grouperInstaller.autorun.addQuickstartSubjectsToDb =

# autorun uncomment to make work, Do you want to add quickstart data to registry (t|f)? [t]
# grouperInstaller.autorun.addQuickstartData =

# autorun Do you want to start the Grouper loader (daemons)?
# grouperInstaller.autorun.startGrouperDaemons = f

# autorun Do you want this script to start the hsqldb database (note, it must not be running in able to start)
(t|f)? [t]:
# grouperInstaller.autorun.startHsqlDatabase = f

# autorun This port does not seem available, even after trying to stop the DB! continue?
# grouperInstaller.autorun.continueAfterPortNotAvailable = t
```

```
# autorun Enter the URL path for the UI [grouper]:
# grouperInstaller.autorun.urlPathForUi = t

# autorun uncomment to make work, Enter the WS log dir: (defaults to the tomcat logs/grouperWs dir
# grouperInstaller.autorun.wsLogDir =

# autorun Do you want to set the log dir of WS (t|f)? [t]:
# grouperInstaller.autorun.setWsLogDir =

# autorun Enter the URL path for the WS [grouper-ws]
# grouperInstaller.autorun.wsUrlPath =

# autorun uncomment to make work, do you want to use local client download, tar, etc, default true
# grouperInstaller.autorun.useLocalClientDownloadTarEtc =

# autorun uncomment to make work, Is it ok to run a script that copies change log temp records to the change
log (recommended) (t|f)
# default is in this file in: grouperInstaller.default.api.runChangeLogToChangeLogTemp
# grouperInstaller.autorun.runChangeLogTempToChangeLog =

# autorun, Download patch if already exists on filesystem, unzip, etc? (default true)
# grouperInstaller.autorun.useLocalPatchIfExists =

# autorun, uncomment to make work, if the version that is detected to be next is ok (t|f) [t]
# grouperInstaller.autorun.patchIndexIsOk =

# autorun, uncomment to make work, Local patch dir exists, is it ok to be automatically deleted? (t|f)? [t]
# grouperInstaller.autorun.deleteLocalPatchFile =

# autorun, unnncomment to make work, continue after copying wiki markup
# grouperInstaller.autorun.patchContinueAfterWikiMarkup =

# autorun, Do you want to name this file as a test version so you can test it without affecting other users?
(t|f) [t]:
# grouperInstaller.autorun.patchNameFileAsTestVersion =
```

sdf