Grant Cancel Permission for user and group that have Build permission

Purpose

This script is developed for JENKINS-14713.
Pre JENKINS-14713, Build permission implicitly grants Cancel permission; post JENKINS-14713, all user/group granted Build permission will lose Cancel permission unless Cancel permission has been given explicitly. This is a configuration incompatibility and can cause few dev start jumping up and down. This script will help Jenkins admin grant Cancel Permission to everyone that has Build permission, therefore maintain the configuration compatibility.

Overview

This script will go through all groups and users in both Global security and per job security settings.
If the group/user has Build permission, Cancel permission will also be granted. If the user/group has Cancel permission already, it's a NOOP.
This script doesn't require JENKINS-14713 to be run.
Pre JENKINS-14713, this script is NOOP

I actually recommend you to run the script before you apply patch for JENKINS-14713, so you are can change and review your ACL setting without any impact to user.

It can also be easily modified for generic Permission bulk changes.

Usage

- go to your Jenkins - > Manage Jenkins - > Script Console
- copy and paste the code snippet from next section
- review it and click Run, this will be a dryrun by default
- follow the output to make real change

Code

```
import hudson.security.*
import jenkins.security.*
import jenkins.model.Jenkins


boolean dryrun=true

if (dryrun) {
  println ''.center(100,'!')
  println 'This is a dryrun, nothing will be changed'.center(100,'!')
  println 'Change this line: boolean dryrun=false to boolean dryrun=true to make the real change'.center
(100,'!')
  println ''.center(100,'!')
}

switch (Jenkins.instance.authorizationStrategy){
  case GlobalMatrixAuthorizationStrategy:
    println '\nGlobal Matrix Strategy defined. Fixing Cancel permissions...\n'
    def sids = Jenkins.instance.authorizationStrategy.getAllSIDs().plus('anonymous')
    for (sid in sids){
      if (Jenkins.instance.authorizationStrategy.hasPermission(sid,hudson.model.Item.BUILD)){
        println '----'+sid+' has Build permission and Cancel permission will be add'
        if (!dryrun) Jenkins.instance.authorizationStrategy.add(hudson.model.Item.CANCEL,sid)
      }
    }
    if (!dryrun) Jenkins.instance.save()
  case ProjectMatrixAuthorizationStrategy:
    println '\nProject Matrix Strategy defined. fixing Cancel permissions...\n'
    def jobs = Jenkins.instance.items
    jobs.each {
      println it.name.center(80,'-')
      def authorizationMatrixProperty = it.getProperty(AuthorizationMatrixProperty.class)
      def sids = authorizationMatrixProperty?.getAllSIDs().plus('anonymous')
      for (sid in sids){
        if (authorizationMatrixProperty?.hasPermission(sid,hudson.model.Item.BUILD)){
          println ''+sid+' has Build permission and Cancel permission will be add'
          if (!dryrun) authorizationMatrixProperty?.add(hudson.model.Item.CANCEL,sid)
        }
      }
      if (!dryrun) it.save()
    }
    break
  default:
    println "No permission need to be mofdified gloabally"
    break
}

return
```

Disclaimer

The script is developed, tested and applied on Jenkins LTS 1.509.2. Use at your own risk!