# Permission limit builtin implementations



This document shows the Grouper built-in implementations to Grouper permissions limits.

### Basics

Permission limits are pluggable, so if the one you needs isn't built in, then you can build your own. The Grouper team can help, ask the grouper-users list. If you create one that seems like it would be useful to others, contribute it back and we can make a comparable built-in. The built-in limits are are created automatically if you set this in the grouper.properties:

```
# root stem in grouper where built in attributes are put
grouper.attribute.rootStem = etc:attribute

# if the attribute loader attributes, and other attributes should be autoconfigured (created, etc)
grouper.attribute.loader.autoconfigure = true
```

Note, each institution sets their own grouper.attribute.rootStem, so the built in attributes might be in different places. If you are in the UI and search for the extension of the attribute you need, it will show you where it is, or you can ask your Grouper administrator. Note that each user who is using the built in attributes needs privileges to do so... Or the admin could make them public by setting ATTR_READ and ATTR_UPDATE to GrouperAll. You can make them public when created (default)

```
# if the permissions limits should be readable and updatable by GrouperAll (set when created)...
grouper.permissions.limits.builtin.createAs.public = true
```

Permission Limit Built-In Implementations are:

- Weekday 9 to 5 limit
- Amount less than limit
- Amount less than or equal limit
- Labels contain limit
- IP address on networks limit
- IP address on network realm limit
- Expression language (EL) limit

### Weekday 9 to 5 limit

Use this if you want the permission allowed from 9 to 5 on a weekday

```
AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

attributeAssign.getAttributeDelegate().assignAttributeByName(PermissionLimitUtils.limitWeekday9to5Name());

//check by time on client (6pm)
new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("(int)hourOfDay", "18").hasPermission()

//check by time on server
new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .hasPermission()
```

### Amount less than limit

This limit makes sure that the amount environment variable is less than a certain amount.  Note, if the amount environment variable is not passed in when the permissions query happens (assuming the call requests a permissions limit processing step), then there will be an exception

```
    AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

    attributeAssign.getAttributeValueDelegate().assignValueInteger(
        PermissionLimitUtils.limitAmountLessThanName(), 50000L);

    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("(int)amount", "49999").hasPermission();
```

### Amount less than or equal limit

This limit makes sure that the amount environment variable is less or equal to a certain amount.  Note, if the 'amount' environment variable is not passed in when the permissions query happens (assuming the call requests a permissions limit processing step), then there will be an exception

```
    AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

    attributeAssign.getAttributeValueDelegate().assignValueInteger(
        PermissionLimitUtils.limitAmountLessThanOrEqualName(), 50000L);

    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("(int)amount", "50000").hasPermission();
```

### Labels contain limit

This limit allows you to configure a comma separated list of labels where the user needs one of them. Note, if the 'labels' environment variable is not passed in when the permissions query happens (assuming the call requests a permissions limit processing step), then there will be an exception.  If the user has no labels, pass in the variable with an empty value.

```
    AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

    attributeAssign.getAttributeValueDelegate().assignValue(
        PermissionLimitUtils.limitLabelsContainName(), "twoFactor, certificate");

    //this will return true, since the user has two factor, and that is one of the required labels
    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("labels", "threeFactor, twoFactor, biometric").hasPermission()
```

### IP address on networks limit

Note, this assumes IPv4.  Note, if the caller is requesting to process limits, and does not pass the ipAddress env variable, then an exception will be thrown

```
    AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

    attributeAssign.getAttributeValueDelegate().assignValue(
        PermissionLimitUtils.limitIpOnNetworksName(), "1.2.3.0/24, 2.3.4.0/16");


    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("ipAddress", "1.2.3.40").hasPermission()
```

### IP address on network realm limit

Note, this assumes IPv4.  This allows you to centrally manage the ip networks being checked.  Register them by name in the grouper.properties.  Note, if the caller is requesting to process limits, and does not pass the ipAddress env variable, then an exception will be thrown

```
//put this in the grouper.properties: grouper.permissions.limits.realm.myInstitutionLocal2 = 4.1.6.0/24, 6.1.0.0
/16

    AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

    attributeAssign.getAttributeValueDelegate().assignValue(
        PermissionLimitUtils.limitIpOnNetworkRealmName(), "myInstitutionLocal2");

    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("ipAddress", "4.1.6.40").hasPermission()
```

### Expression language (EL) limit

This limit allows a scriptlet in expression language to do pretty much anything you want, though it can be complex for non technical people set it up.  There is documentation to help though.  We use Jakarta commons JEXL.

If you are doing numeric limits (e.g. amount < 50000), this is an easy one to use.  You can do boolean combinations, etc.

If you are calculating limits, and you do not pass in a variable used in the EL, then it will throw an exception

```
attributeAssign.getAttributeValueDelegate().assignValue(PermissionLimitUtils.limitElAttributeDefName().
getName(), "amount < 50000");
new PermissionFinder().addSubject(this.subj0).addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS).hasPermission();
//note: throws ExpressionLanguageMissingVariableException
//the exception message says: variable 'amount' is not defined in script: 'amount < 50000'
```

You can use built-in variables (which you can override by passing them in the env var map:

- calendar: the current calendar
- dayOfWeek: from calendar.SUNDAY to calendar.SATURDAY.  Note that these are integers are 1-7
- hourOfDay: from 0 to 23
- minuteOfHour: from 0 to 59
- minuteOfDay: from 0 to 1439
- monthOfYear: from calendar.JANUARY to calendar.DECEMBER.  Note that these are integers from 0 to 11

These are built in variables that you cannot override:

- limitElUtils: instance of LimitElUtils
- limitAssignmentId: assignmentId of the limit

- permissionAction: action we are checking
- permissionMemberId: memberId we are checking if permission is assigned to member
- permissionRoleId: role id of the permission we are checking
- permissionRoleName: role name of the permission we are checking
- permissionAttributeDefNameId: attribute def name id of the permission
- permissionAttributeDefNameName: attribute def name of the permission

You can put in your own custom variables in the grouper.properties to run your own logic:

```
grouper.permissions.limits.el.classes = some.fully.qualified.SomeClassName, some.fully.other.AnotherClassName
```

Those classes will be registered with the variable names: someClassName and anotherClassName, note they will be instances (but can still use static methods), and need a default public constructor

So, if you want to check time, you can do this with time on client, or server.  Here is an example of time on client (at 10:13am), allowed between 9am and 5pm:

```
    this.adminRole.getPermissionRoleDelegate().assignSubjectRolePermission(
        this.readString, this.artsAndSciences, this.subj0, PermissionAllowed.ALLOWED);

    AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

    attributeAssign.getAttributeValueDelegate().assignValue(PermissionLimitUtils.limitElAttributeDefName().
getName(),
      "hourOfDay >= 9 && hourOfDay <= 17");


    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("(int)hourOfDay", "10").hasPermission()
```

Here is an example of time on server (just dont pass in a variable, and it wont override whats there)

```
    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .hasPermission()
```

Here is an example of checking an ip address to be on a network (note, if processing limits on the server, the env var ipAddress must be passed in):

```
AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

attributeAssign.getAttributeValueDelegate().assignValue(PermissionLimitUtils.limitElAttributeDefName().
getName(),
        "limitElUtils.ipOnNetwork(ipAddress, '1.2.3.0', 24)");

new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("ipAddress", "1.2.3.40").hasPermission();

//check to see if on multiple networks:
attributeAssign.getAttributeValueDelegate().assignValue(PermissionLimitUtils.limitElAttributeDefName().
getName(),
        "limitElUtils.ipOnNetworks(ipAddress, '1.2.3.0/24, 2.3.4.0/16')");

new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("ipAddress", "1.2.3.40").hasPermission();

//centrally manage the networks:
attributeAssign.getAttributeValueDelegate().assignValue(PermissionLimitUtils.limitElAttributeDefName().
getName(),
        "limitElUtils.ipOnNetworkRealm(ipAddress, 'myInstitutionLocal')");

//note: there is a grouper.properties entry: grouper.permissions.limits.realm.myInstitutionLocal = 4.5.6.0/24,
6.7.0.0/16new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("ipAddress", "4.5.6.40").hasPermission();
```

You can test a label set to see if the user has that label

```
    AttributeAssign attributeAssign = new PermissionFinder().addSubject(this.subj0).addAction(this.readString)
      .addPermissionName(this.artsAndSciences).addRole(this.adminRole).assignImmediateOnly(true).findPermission
(true).getAttributeAssign();

    attributeAssign.getAttributeValueDelegate().assignValue(PermissionLimitUtils.limitElAttributeDefName().
getName(),
        "limitElUtils.labelsContain(authnAttributes, 'twoFactor, certificate')");

    //this is true since one of the strings is twoFactor, and that is one of the ones checking for...
    new PermissionFinder().addSubject(this.subj0)
        .addAction(this.readString).addPermissionName(this.english)
        .assignPermissionProcessor(PermissionProcessor.
FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS)
        .addLimitEnvVar("authnAttributes", "twoFactor, threeFactor, biometric").hasPermission()
```

sdf