

# **AMS II**

## **Access Management System**

### **for language archive resources in IMDI-corpora**

**Manual for AMS II, version 1.3**



**Max Planck Institute for Psycholinguistics, Nijmegen, The Netherland  
December 2008**

---

**AMS II: Access Management System: for language archive  
resources in IMDI-corpora**

**Access Management System**

**for language archive resources in IMDI-corpora**

Micha Hulsbosch

Manual for AMS II, version 1.3

---

---

# Table of Contents

1. Introduction .....	5
1.1. Main Concepts .....	5
1.1.1. The corpus tree .....	5
1.1.2. AMS rules .....	6
1.1.3. Licenses .....	6
1.1.4. Roles .....	6
1.2. Getting started .....	7
2. User management .....	9
2.1. Creating a new user .....	9
2.1.1. Entering user data .....	9
2.1.2. Adding a new user to groups .....	10
2.2. Editing an existing user .....	11
2.3. Creating a new group .....	11
2.4. Editing an existing group .....	12
3. Rules .....	13
3.1. Basic principles .....	13
3.1.1. Examples of access rights calculation .....	15
3.2. Checking existing rules .....	16
3.2.1. All rules of a node .....	16
3.2.2. All rules of a node for a specific user or group .....	17
3.3. Managing rules .....	17
3.3.1. Adding rules .....	17
3.3.2. Editing and Revoking rules .....	18
3.3.3. Special groups: <i>Everybody</i> and <i>Registered Users</i> .....	19
3.4. Licenses .....	19
3.4.1. Accepting licenses .....	19
3.4.2. Linking a license to a node .....	20
4. Advanced usage and administration .....	21
4.1. Mailing .....	21
A. Example use case of roles .....	22
A.1. Corpus manager .....	22
A.2. Researcher .....	22
A.3. Research team .....	22
A.4. Research assistant .....	22
A.5. Interested colleague .....	22
A.6. Accidental visitor .....	22

---

# Chapter 1. Introduction

The Access Management System (AMS) allows you to manage the access rights of electronic language resources in the IMDI browser [<http://www.lat-mpi.eu/tools/imdi/browser/>] of the LAT [<http://www.lat-mpi.eu/>] framework. These resources consist of Info files, annotation or text files, images, audio or video files that were uploaded with LAMUS [<http://www.lat-mpi.eu/tools/lamus>] into a corpus. By setting the access rights you can both allow or explicitly deny access to resources for individual users or groups of users. Note that you are not able to set access rights to the nodes in the archive. The structure of the archive will always be visible. Therefore, the metadata will always be accessible, as required by the Open Archive Initiative [<http://www.openarchives.org/>].

AMS was developed at the Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands.

## 1.1. Main Concepts

### 1.1.1. The corpus tree

The base concept of AMS is the corpus tree. The corpus consists of nodes and arcs that form a tree-like structure representing the corpus hierarchy. Each node can group other nodes on the basis of, e.g., the geographical region, the discourse genre, the sex or age of the speaker, the dialect of the speaker, the target/source language etc. The lowest level in the hierarchy consist of the actual resources (see Section 1.1.1.2). Consider the following example:

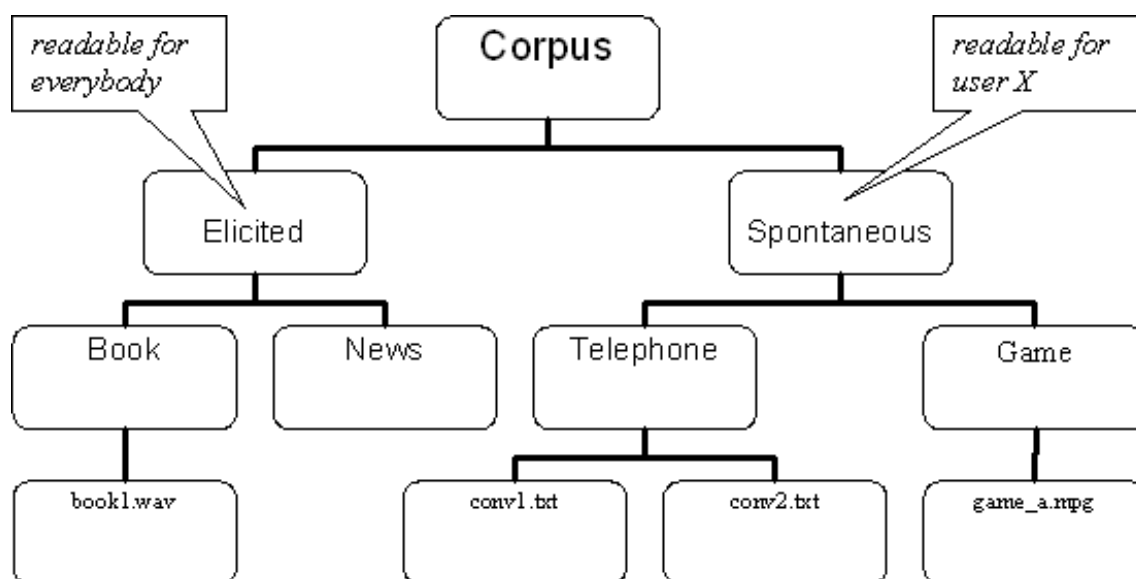


Figure 1.1. Corpus tree example

The node labeled Corpus is the top node. The nodes labeled Elicited and Spontaneous are subnodes. These subnodes are sometimes called 'children' of the node above them, in this case the top node Corpus. The nodes Book and News are children of the node Elicited and grandchildren of Corpus. Similarly, the nodes Telephone and Game are children of Spontaneous and also grandchildren of Corpus. The nodes labeled filenames like `book1.wav` and `game_a.mpg` are at the lowest level and represent some resources.

Using this hierarchical data representation you can specify access rights in the form of rules (see Section 1.1.2) for a certain branch of the tree. A branch consists of a node and all of its descendants. So a rule does not only apply to an individual node, but also to all of its descendants. Since a node groups children, grandchildren and other descendants, a branch is called a *domain* in AMS.

To see how rules apply to a domain, consider Figure 1.1 again. Suppose we set the access right for the *domain* Elicited to something like 'readable for everybody'. This means that all resources in the domain of the

node Elicited are 'readable for everybody'. A resource (in fact the only one) in this domain is `book1.wav`, so this resource is 'readable for everybody'. Another example concerns the domain Spontaneous. Suppose we would like to make this domain 'readable for user X'. Since resources are not readable for any user by default, the only rule we have to set is 'readable for user X'. Now the resources `conv1.txt`, `conv2.txt` and `game_a.mpg` are readable for user X but not readable for any other user.

### 1.1.1.1. Corpus and session nodes

Nodes like Book, Telephone and Game in Figure 1.1 are called session nodes. They group all resources that are part of a meaningful unit of analysis. Nodes like Corpus, Elicited and Spontaneous are called corpus nodes. They group session nodes or other corpus nodes giving the archive its treelike structure.

### 1.1.1.2. Resources

*Resources* is a common name for all kinds of files that can be associated with session nodes. Resources are the content of the corpus (as opposed to the corpus tree and metadata). Resources can be placed in the following types:

- Images, e.g. JPEG
- Video files, e.g. MPEG
- Audio files, e.g. WAV
- Annotations/Text, e.g. EAF
- Info files (all kind of files), e.g. PDF

### 1.1.2. AMS rules

In AMS, the access rights of a certain domain are expressed by one or more rules. A rule gives an individual user or all users from a group the permission to open, download or read all resources of a certain type in a domain. A rule can also explicitly deny that permission. A rule contains the following elements:

- a user or group to which the rule applies;
- the type of resource;
- whether reading is allowed or denied;
- a priority;
- an expiration date (optional);

If there is a rule on a certain node (the rule then applies to the domain of that node) it is possible to create a rule on a descendant of that node. In that case there are two rules that apply to the domain of the descendant. Depending on the elements of these rules one of them is considered the 'strongest' and enforces its access rights. More on the calculation of the strongest rule as well as other topics can be found in Chapter 3.

### 1.1.3. Licenses

If necessary you can request a user to accept a license agreement before he/she can access resources in a part of the corpus tree. This typically contains agreements on ethical codes and responsibility for the use of the data. More about licenses in Section 3.4.

### 1.1.4. Roles

Roles are predefined templates that define in what way a user can use the corpus. This encompasses access rights, rights to change the content and the right to pass on all of these privileges. The following roles exist:

### Archive Managers

To each user the role of Archive Manager can be assigned. This means that all possible rights are granted to this user such as accessing all resources and changing access rules. An archive manager can in turn appoint other archive managers. It goes without saying that this possibility should be used with care. It is the only role that is not bound to a domain.

### Domain Curator

A domain curator:

- is bound to a domain (i.e. has power over a node and its descendants).
- can set and revoke rules on all of the nodes within that domain.
- can create, remove and alter users and groups. Altering and removing only works for users/groups created by the domain curator.
- can delegate his/her rights (except this delegation right itself) to a Domain Manager.

A domain can only have one Domain Curator.

### Domain Manager

A Domain Manager cannot assign other domain managers. This is the only difference with a domain curator. So a domain manager can (for a given domain) set and revoke read rules, create and change those self-created users/groups.

### Domain Editor

In contrast with the roles above a Domain Editor can add and remove corpus nodes and resources (again: for a specific domain). This right is closer related to LAMUS than AMS itself. Therefore one generally combines the Domain Editor role with that of Domain Manager or Curator. That way one user can both upload and change information in the corpus and set the access read rights afterwards.



#### Note

The domain-based roles (curator, manager, editor) can only be accessed and set via the Node Authorization Management as these are dependant on a certain domain. The Archive Manager role applies to the whole corpus and thus can be selected when creating or editing a user/group.

## 1.2. Getting started

To start AMS first open the IMDI browser via <http://corpus1.mpi.nl/>. Browse to the corpus node for which you want to create or revoke a rule. Right click that node and select **set access rights** in the context menu. You will subsequently be asked to enter a username and password. After entering these you will see the Node Authorization Overview (see Figure 1.2).

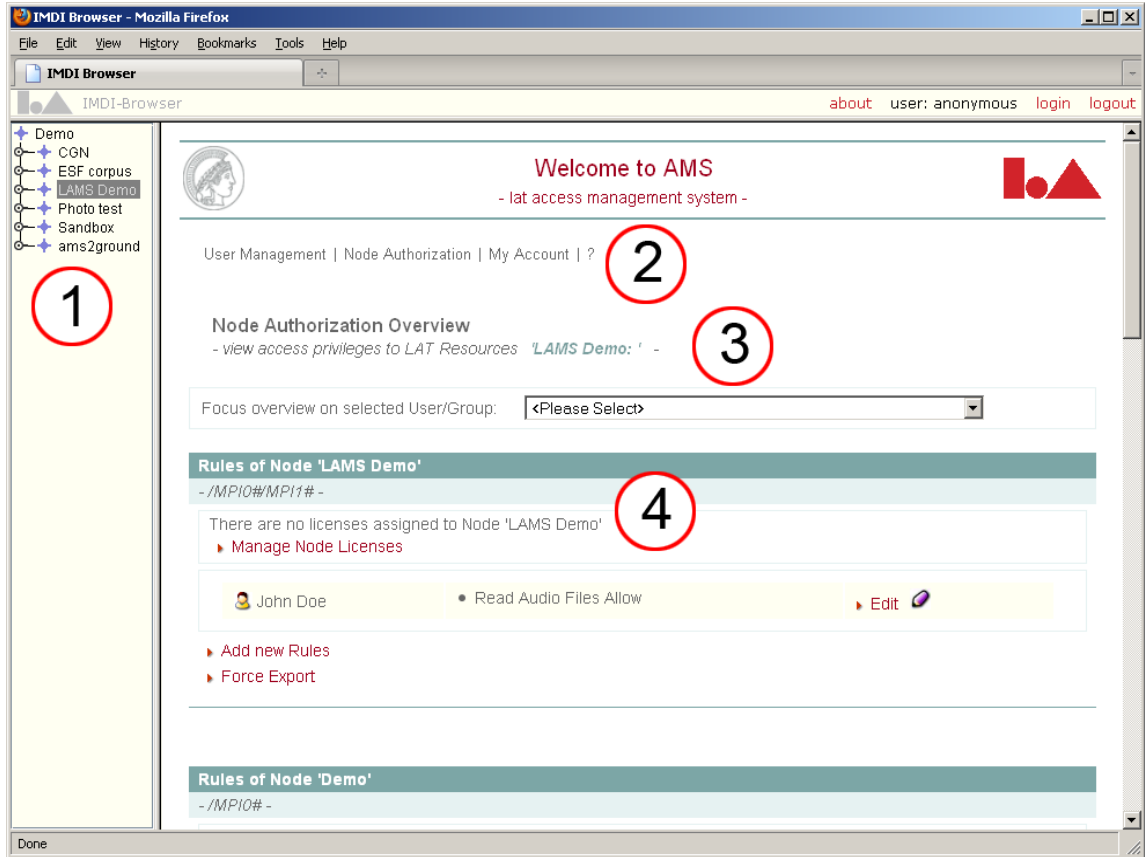
Beneath the welcome text you find the AMS menu bar containing the menu headings

- User Management
- Node Authorization
- My Account
- ?

Beneath the menu bar some information about the selected node is displayed:

- The name and title of the currently selected node (in this case "LAMS Demo" / "Lams Demo Title").
- All active rules for this node and its parent, grandparent and other ancestors, in bottom-up order (here "LAMS Demo" and its parent "Demo". For each rule the user/group and the granted permissions are shown.

From this point you can create or edit users and groups of users (more about user management in Chapter 2) or you can create or edit access rules (more about rules in Chapter 3).



The Node Authorization Overview with the corpus tree (1), the AMS menu (2), name and title of the current node (3) and all active rules (4).

### Figure 1.2. AMS introduction screen

*The Node Authorization Overview with the corpus tree (1), the AMS menu (2), name and title of the current node (3) and all active rules (4).*

---

# Chapter 2. User management

When granting access rights to a part of a corpus you obviously need to point out to whom these rights should be given. Unless you intend to open resources for the whole world it will be necessary to appoint a user or a group of users that should receive the access rights. This chapter describes how to create and modify users and groups.



## Note

To create or edit a user or group of users you need to have the role of Archive Manager, Domain Curator or Domain manager.

## 2.1. Creating a new user

To create a new user click User Management > Create New User in the AMS menu bar. A new page will appear as shown in Figure 2.1.

### 2.1.1. Entering user data

The first step is to fill in the required fields in the User Data section. Choose an appropriate Hosting Institution for the new user. UID is changed accordingly. Enter a unique user name in the User Name (UID) field.



## Note

If the chosen Hosting Institution is not the default (in the example in Figure 2.1 the default is MPI for Psycholinguistics) the username to be used at login is *User Name + @ + UID Domain*. In our example that would be *John Doe@mpi.nl* (if MPI for Psycholinguistics was not the default).

In the Archive-wide Roles field you can decide if the new user should become an Archive Manager by ticking the checkbox (only if you are an Archive Manager yourself). Save the user data by clicking on the Save button at the left bottom corner of the page. The creation of the new account will be confirmed with a message like User "John Doe" has been saved.

**User Data**

- modify user core data -

1

Hosting Institution	<input type="text" value="MPI for Psycholinguistics"/>
User Name (UID)	<input type="text" value="John Doe"/>
UID Domain	<input type="text" value="mpi.nl"/>
First Name	<input type="text" value="John"/>
Last Name	<input type="text" value="Doe"/>
eMail	<input type="text" value="jdoe@mpi.nl"/>
Organization	<input type="text" value="MPI for Psycholinguistics"/>
Password	<input type="password" value="••••••••"/>
Repeat Password	<input type="password" value="••••••••"/>

Archive-wide Roles	<input type="checkbox"/> Archive Manager
--------------------	--

---

**User Groups**

- assign & revoke User to/from Groups -

Please save the new User initially before you assign him to Groups

Group Name	Group Name
<input type="checkbox"/> (none) ((none)@mpi.nl)	<input type="checkbox"/> Andes_group (Andes_group@mpi.nl)
<input type="checkbox"/> (none)-PROD ((none)-PROD@mpi.nl)	<input type="checkbox"/> annex_all (annex_all@mpi.nl)
<input type="checkbox"/> ac-Stoll (ac-Stoll@mpi.nl)	<input type="checkbox"/> annex_demo (annex_demo@mpi.nl)
<input type="checkbox"/> akhoe_group (akhoe_group@mpi.nl)	<input type="checkbox"/> aslep-tofa (aslep-tofa@mpi.nl)
<input type="checkbox"/> Amanda_group (Amanda_group@mpi.nl)	<input type="checkbox"/> aslep1 (aslep1@mpi.nl)

⏪ ⏴ ⏵ ⏩ 1 2 3 4 5 6 7 8 9 10 11 ⏴ ⏵ ⏩

Save
2
New
Cancel

**Figure 2.1. Creating a new user**

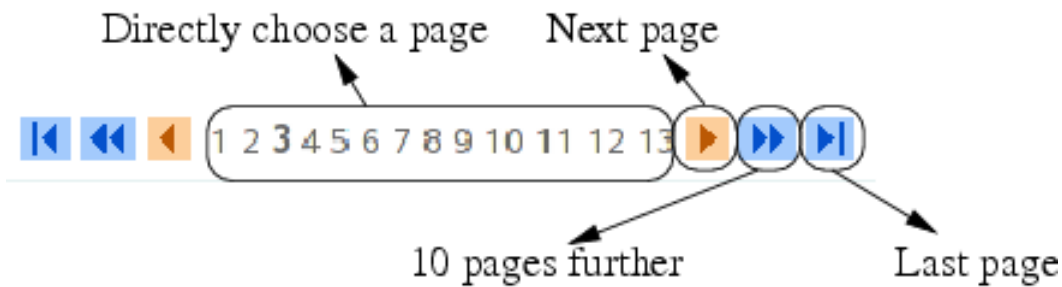
To create a new user first enter the user data (1), then click on Save (2) and finally select the groups you want the user to be added to (3) and click on Save again.

## 2.1.2. Adding a new user to groups

The second - optional - step is to add the newly created user to a group. In the *User Groups* section you find a list of all existing groups. All the groups a user is a member of will be listed first. If you want the new user to be added to one of them, tick the checkbox next to it and confirm again by clicking on *Save*. Note that if you did not save this new user data yet, the checkboxes cannot be ticked.

If the group for addition is not currently displayed, navigate through the group enumeration as shown in Figure 2.2. Alternatively, you can enter a string in the textfield in the lower right corner of the page and click *Filter groups*. As a result, the list is reduced to only contain those groups of which the name contains the string in the textfield.

10



**Figure 2.2.** Navigating through a list

To create another user click on New (to the right of Save) and go through the same steps as described in Section 2.1.1 and Section 2.1.2.

## 2.2. Editing an existing user

Changing the properties of existing users can be achieved via **User Management > Edit User**. This will show a list of all users. Look up a user in this list using the navigation shown in Figure 2.2. Alternatively, you can enter a string in the textfield in the lower right corner of the page and click **Filter users**. As a result, the list is reduced to only contain those users of which the name contains the string in the textfield. Click on the ID of a user to change its profile. The page that now appears is similar to that in Figure 2.1. Now you can:

- change all fields except the Hosting Institution, User Name (UID) and UID Domain. Confirm these changes by clicking on Save.
- delete this user by clicking on Delete in the bottom right corner.



### Note

When you delete a user all its data are removed including access rules. Restoring the data is not possible!

- change the group membership of the user by (un)checking the boxes before the group names in the User Groups section. Remember to confirm these changes by clicking on Save.

## 2.3. Creating a new group

A group is a collection of users. A rule for a group applies to all members of that group. Using groups saves you the hassle of setting rights for all individual users. One user can belong to multiple groups.

New groups can be created via **User Management > Create New Group**, which will result in a page like in Figure 2.3. Here you can enter an ID (a short identifier that cannot be changed afterwards) and a Name (a more extensive description of the group). Click on Save to store the new group or New to create another one.

LAT Group  
- manage LAT Groups -

**Group Data**  
- modify group core data -

Hosting Institution	MPI for Psycholinguistics
ID (group identifier)	<input type="text"/>
Name	<input type="text"/>

[Save](#)   [New](#)   [Cancel](#)

2007 [www.mpi.nl](http://www.mpi.nl)

**Figure 2.3. Creating a new group**

After clicking Save you are able to select users to be members of this newly created group. This is done in the section Members where you can check the users you want to add as members. Navigate through the list of members as shown in Figure 2.2. Alternatively, you can enter a string in the textfield in the lower right corner of the page and click **Filter users**. As a result, the list is reduced to only contain those users of which the name contains the string in the textfield.

## 2.4. Editing an existing group

A group can be edited via User Management > Edit Group. First select a group by clicking it in the list. To find a group, navigate through the list of groups as shown in Figure 2.2. Alternatively, you can enter a string in the textfield in the lower right corner of the page and click **Filter groups**. As a result, the list is reduced to only contain those groups of which the name contains the string in the textfield.

Now change the groups name and/or add or remove users by (de)selecting the checkbox in front of the names of the users (use the navigation and filter to find users). Finally click Save to save the changes.

If you want to remove the group instead of changing it, click **Delete** in the bottom right corner of the page.



### Note

When you delete a group all its data are removed including access rules. Restoring the data is not possible!

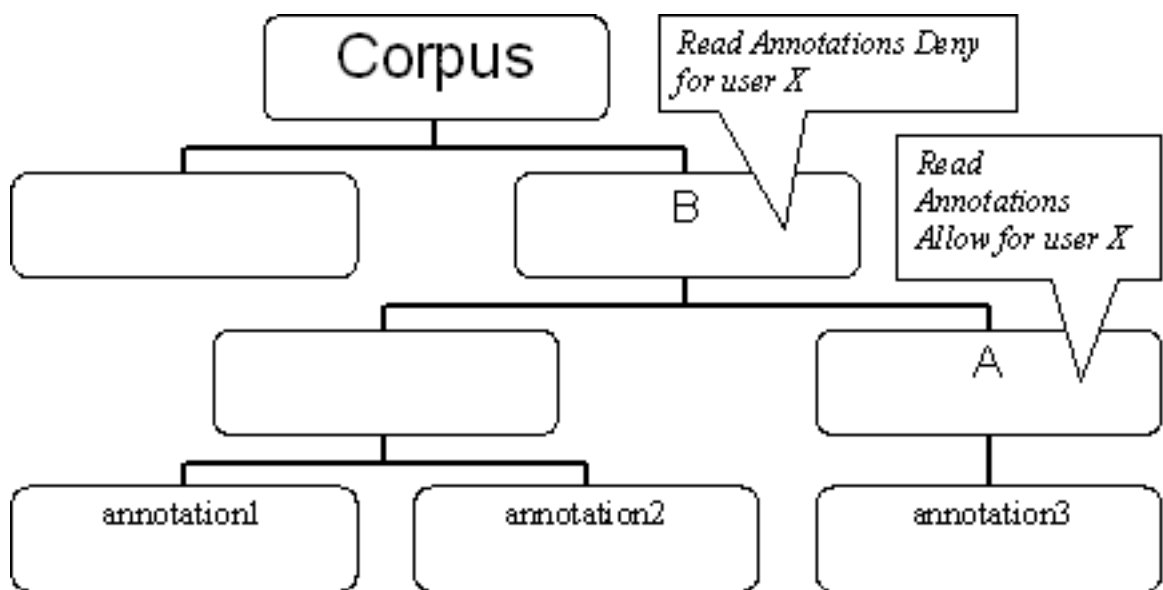
---

# Chapter 3. Rules

## 3.1. Basic principles

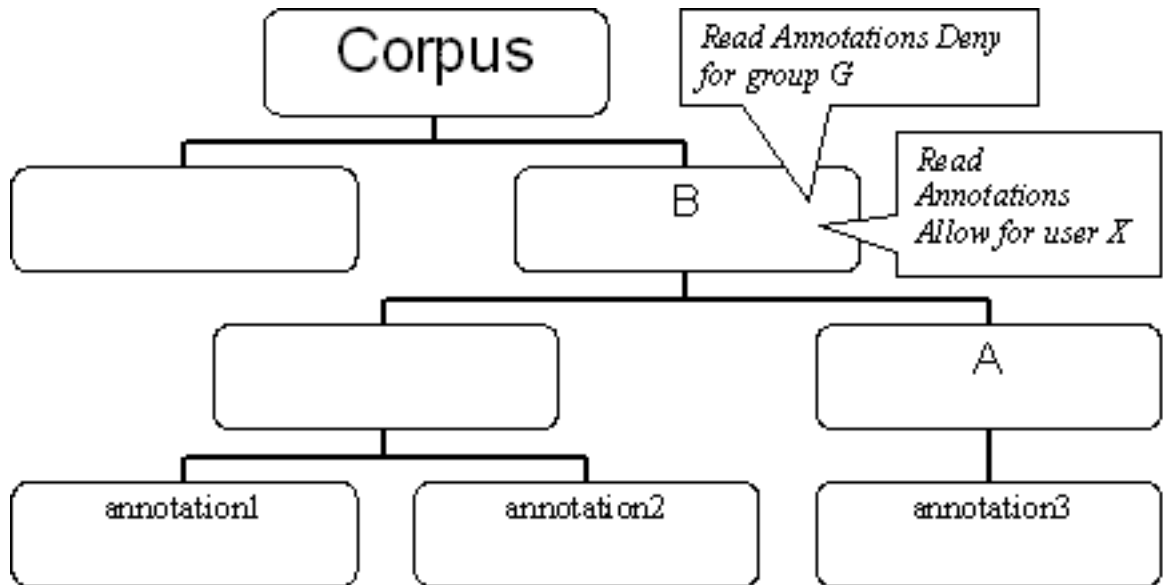
Before creating or editing rules, we will take a closer look at the principles behind the access policies in AMS. This is quite important given the potential for confusion if there is more than one rule that applies to a certain domain. There are two situations that are potentially confusing:

1. Lets say that node A has a rule allowing user X to read annotation files. That means that user X is allowed to read all annotation files in the domain of node A (i.e. annotation3 in Figure 3.1). Now say that one of the ancestors of node A, node B, has a rule denying user X to read annotation files. That means that user X is not allowed to read any annotation file in the domain of node B. If we consider that the domain of node A is in fact part of the domain of node B, we see that the rules of both nodes are in conflict.



**Figure 3.1.**

2. Lets say a node has two rules (see Figure 3.2). One rule allows user X to read annotation files. The other rule denies group G to read annotation files. Consider user X to be a member of group G. That means that the two rules concern user X, one allowing and the other denying user X to read annotation files. Again, the two rules are in conflict.



**Figure 3.2.**

Only rules that apply to the same resource type (annotations in the examples above) can conflict. If two rules do not apply to the same resource type they cannot conflict.

The fact that rules may conflict raises the question of how AMS determines whether a user is allowed to read or open a resource. To be able to answer that question we state the following basic principals:

- A rule either allows or denies access to a resource for a specific user or for all members of a group.
- The list containing a node and all of its ancestors is called the *canonical path* of that node. (Remember that each node has only one parent, so that a node and all its ancestors form a list from root node up to that node.)
- To calculate the access rights of a resource all rules in the canonical path are considered.
- A rule can have the following priority levels:
  - Highest
  - High
  - Normal



### Note

Only users with the Archive Manager role can use the highest level of priority when adding and editing rules.

Now that we have seen what basic principles are involved in the calculation of the access rights, we state the principles of the calculation itself. These principles are applied by AMS in the same order as we show them here:

1. The priority principle: The highest priority of the considered rules is determined and all rules with a priority lower than that are outvoted.
2. The closeness principle: From the nodes in the canonical path that have at least one rule, only the rules from the node that is closest to the resource is kept. All other rules are outvoted. Consider for example

the canonical path A-B-C-D-resource (top down). If both node A and node C have a rule, only the rule on C is kept. The rule on node A is outvoted.

3. The constraint principle: If there is more than one rule applying to the same resources type, the rules have equal priority and equal closeness and they are conflicting (at least one is denying and one is allowing a user to read a resource), access to the resource is denied. So denying outvotes allowing.

So in short, AMS first selects rules with the highest priority. Than it selects rules on nodes closest to the resource from the remaining rules. Finally, it looks whether one of the remaining rules denies the resource to be read. By applying the calculation principles this way AMS avoids conflicting situations as described at the beginning of this section.

### 3.1.1. Examples of access rights calculation

The following three situations exemplify how the access rights are calculated from all considered rules:

#### Example 1

Consider the following canonical path with the nodes listed in top down order and their corresponding rules:

Node A: Top Node	Rule 1: <i>Read Annotation Files Allow</i> (normal priority) for user X
Node B	Rule 2: <i>Read Annotation Files Deny</i> (normal priority) for user X
Node C: Resource - annotation file <code>test.txt</code>	-

Both rules concern the reading of annotation files by user X and both rules have the same priority. This results in a conflict. In this example rule 1 allows user X to read the annotation file `test.txt` on node C and rule 2 denies the file to be read by user X. Since both rules have the same priority, AMS cannot make a choice based on the priority principle. Next, AMS applies the closeness principle expressing that only rules from nodes that are closest to the resource are weighted heavier. Applying this principle results in the outvoting of rule 1 leaving only rule 2. Since rule 2 denies reading annotation files, AMS also denies user X to read the annotation file `test.txt`.

Putting rule 2 on node A and rule 1 on node B results in the discarding of rule 2 based on the closeness principle. The remaining rule 1 results AMS in allowing user X to read the annotation file `test.txt`.

#### Example 2

Another canonical path and corresponding rules:

Node A: Top Node	Rule 1: <i>Read Annotation Files Allow, highest priority</i> for user X
Node B	Rule 2: <i>Read Annotation Files Deny, high priority</i> for user X
Node C	Rule 3: <i>Read Annotation Files Deny</i> (normal priority) for user X
Node D: Resource - annotation file <code>test.txt</code>	-

One of the three rules in this example allows user X to read the annotation file `test.txt`, while the other two deny reading the file. Moreover, all three rules have a different priority level. Applying the priority principle results in the outvoting of rule 2 and 3 because only rule 1 has the highest priority found in the three rules. The final result is that AMS allows user X to read the annotation file of node D.

## Example 3

A slightly more complicated example is the following canonical path and corresponding rules:

Node A: Top Node	Rule 1: <i>Read Annotation Files Deny, high priority</i> for user X
Node B	Rule 2: <i>Read Annotation Files Deny, high priority</i> for user X
	Rule 3: <i>Read Annotation Files Allow, high priority</i> for group G - user X is a member of group G
Node C	Rule 4: <i>Read Annotation Files Deny</i> (normal priority) for user X
Node D: Resource - annotation file <code>test.txt</code>	-



Applying the priority principle discards rule 4 because the highest priority level is *high* and rule 4 has a lower priority level. This leaves rule 1, 2 and 3. Applying the closeness principle discards rule 1 leaving rule 2 and 3. Since we are interested in the access rights of user X and user X is a member of group G rule 2 and 3 are still in conflict. So, AMS applies the constraint principle. This means that if there is still a rule denying the right to read a resource, that rule is applied to calculate the final access right.

## 3.2. Checking existing rules

Before creating and editing rules, we first take a look at how existing rules are displayed in AMS.

### 3.2.1. All rules of a node

As we saw in Section 1.2 you start AMS by right clicking a node in the tree structure of the IMDI browser and clicking on **set access rights** in the context menu. The page that now appears contains the Node Authorization Overview. If you are working in a different part of AMS, this overview can also be displayed by clicking **Node authorization > Overview** in the AMS menu.

The Node Authorization Overview shows the rules of all nodes in the canonical path from the selected node to the top node of the corpus tree in a bottom-up fashion. In the example shown in Figure 3.3, only the active node - LAMS Demo - contains rules: one for the user (marked by the icon ) *John Doe* and one for the group () *John Doe's research group*. If you let your mouse cursor hover over a user or group icon, some details for these items will be displayed. For both the user and group the existing rules are displayed in a bulleted list to the right of their name.



The screenshot displays the 'Node Authorization Overview' page. At the top, it shows the title 'Node Authorization Overview' and a subtitle '- view access privileges to LAT Resources 'LAMS Demo: Lams Demo Title' -'. Below this is a dropdown menu for 'Focus overview on selected User/Group:' with the value '<Please Select>'. The main content area is divided into two sections: 'Rules of Node 'LAMS Demo'' and 'Rules of Node 'Demo''. The 'Rules of Node 'LAMS Demo'' section shows a list of assigned licenses, including 'Webcode (webcode.html)', and a list of users and groups with their respective privileges. The 'Rules of Node 'Demo'' section shows that there are no licenses or privileges assigned to this node. A tree view on the left side of the interface shows the hierarchy of nodes, with 'LAMS Demo' selected.

**Figure 3.3. Overview of the rules for the selected node**

Besides the access rules Figure 3.3 also shows that a license, called Webcode, has been linked to node LAMS Demo. More about licenses in Section 3.4.

### 3.2.2. All rules of a node for a specific user or group

To be able to manage access rules more easily AMS enables you to narrow down the list of rules of a node to only the rules that concern a specific user or group. To do so select a user or group from the drop down list next to *Focus overview on selected User/Group*. Now only the rules that pertain to the selected person or group will be displayed.

If you have selected a user you are able to view the effective user privileges for all resources directly under the current node for the selected user. To do so click on View next to *View effective privileges on selected User* or click Node authorization > Effective User Privileges in the AMS menu. A list of all resources will appear. Depending on the rights a user has,  (access) or  (no access) will be displayed for each resource. Both read and write access rights are shown. The latter can only be obtained by users with the Domain Editor role.

## 3.3. Managing rules

### 3.3.1. Adding rules

As we saw in Section 1.1.4 only Archive Managers, Domain Curators and Domain Managers can add, edit or revoke rules. So make sure you have an appropriate role for the considered domain when carrying out the steps in this section. Also make sure you have agreed upon the node licenses (see Section 3.4), as not doing so might block you from setting up new rules.

To add a rule carry out the following steps:

1. Select a domain i.e. a node from the corpus tree in the IMDI browser.
2. Right-click on the node and select **Set access rights**. The page that now appears is the Node Authorization Overview (see also Figure 1.2).

3. Click on *Add new Rules*. The page that now appears is the Node Authorization Management.
4. Select a user or group from the dropdown menu in the section *Rule Settings*. This section has now two additional boxes: Specify Rules and Assign Roles.




### Note

If you had already selected a user or group in the Node Authorization Overview, it is not necessary to select a user or group in the Node Authorization Management.

## Specify Rules

The following steps will specify rules:

5. Specify a rule for one or more of the following resource types by selecting the appropriate checkboxes:
  - Info files (all kind of files), e.g. PDF
  - Annotations/Text, e.g. EAF
  - Images, e.g. JPEG
  - Video files, e.g. MPEG
  - Audio files, e.g. WAV
6. Allow or deny reading the resource by selecting the appropriate option from the dropdown menu in the Type column.
7. If you have the Archive Manager role, a priority dropdown menu will be shown as well next to each of these resource types. See Chapter 4 for an explanation on this.
- 8.

Optionally you can specify an expiration date - on which the rule will be deactivated. Click on the  icon to get a small calendar for the selection of a day.

## Assign Roles

If you are an Archive Manager or Domain Curator the Assign Roles boxes will be activated. This allows you to appoint a user/group as Domain Curator/Manager/Editor. Only the possibilities that come with your privileges are available, the other ones are grayed out.

9. Check the roles you wish to assign to the selected user/group.
10. Optionally roles can be time-limited using an expiration date as well.
11. The Domain Editor role must be assigned a maximum storage space for resources.
12. Click on *Save* or discard the settings using the *Cancel* button on the bottom of the screen.

When saved, the new rules/roles will be displayed next to the users they are assigned to.

## 3.3.2. Editing and Revoking rules

Editing and revoking rules can be done in much the same way as adding rules. Besides the possibility to click *Add new Rules* in the Node Authorization Overview it is also possible to click on *Edit* to the left of an existing rules. This shows the Node Authorization Management where you can edit and revoke rules. Revoking is done by deselecting the checkbox of an existing rule and clicking on *Save*.

### 3.3.3. Special groups: *Everybody* and *Registered Users*

There are two special groups in the list of users and groups: *Everybody* and *Registered Users*. The difference between the two is that *Everybody* includes both unregistered and registered users while *Registered Users* only contains registered users.


A rule for *Everybody* outvotes any other rule in a domain, independent of the priority, and all users are a member of *Everybody*. The implications are as follows: suppose a rule states that access to a resource type in a certain domain is denied for user X and another rule states that access is allowed for the same resource type in the same domain for *Everybody*. Since a rule for *Everybody* outvotes all other rules within its domain, the *Everybody* rule is effective and access is allowed for everybody including user X despite a rule denying access. The opposite also holds: a rule stating that *Everybody* is denied access will always result in denying access to each individual user despite the access allowing rules that might exist.


A rule for *Registered Users* is very much the same as for *Everybody*. The only difference is that a rule for *Everybody* also outvotes licenses and a rule for *Registered Users* does not. To put it another way, if there is a license set for a resource (see also Section 3.4) and a rule saying that access is allowed for *Everybody*, the license does not have to be accepted first. In the same situation except that the rule is set for *Registered Users*, the license must be accepted first.

## 3.4. Licenses

A license is a text a user should agree upon before being able to access certain resources. It always applies to a domain, i.e. a node and all its descendants.

### 3.4.1. Accepting licenses

The first time you want to access a resource to which a license applies, you have to accept that license explicitly. To do so, right click the resource, select **Set Access Rights** from the context menu and then go to **Node Authorization > Licenses Required**. An overview of all licenses that apply to the resource you are interested in will be displayed. Click on  *View* to read the license agreement text.


To accept a license, tick the checkbox in front of it and click on *Accept*. The accepted licenses will now be marked with a  icon. The date of the acceptance will be shown as well.



#### Note

If you accept a license, you accept it for all the domains for which the license is set.

It is possible that there are more than one licenses assigned to nodes within a domain. You could accept each of these licenses once you come across a resource for which a license that you did not accept is set. However, it can be useful to accept all licenses that are assigned to nodes within a domain at once. To do so, right click the top node of the domain and select **Set Access Rights** from the context menu. If you now select **Node Authorization > Licenses Required** in the AMS menu, you will see not only the licenses assigned to the selected node, but also the licenses assigned to all the nodes in the domain of the node, i.e. all nodes that are below the selected node in the tree. Moreover, you can accept them all at once by checking them all and clicking *Accept*. Naturally, you can also accept only part of the licenses in the list.

To view all the licenses you have accepted so far, click **My Account > My Accepted Licenses** in the AMS menu. This overview shows the licenses and the date and time you have accepted it. You can also view the license by clicking  *View*. The overview is independent of the selected node.

### 3.4.2. Linking a license to a node

Similarly to setting up rules for a node, right click on a node and select Set Access Rights. Now click on *Manage Node Licences* or select **Node Authorization > Node License Assignment**; a list with all existing license agreements will appear. Check the boxes in front of one or more licenses and click on *Save*. Only users that have accepted the license can access the resources in the domain of this node. By deselecting and saving the changes a license requirement can be revoked.



#### **Note**

At the moment the list of licenses is fixed. In future versions of AMS it will be possible to specify your own.

---

# Chapter 4. Advanced usage and administration

This chapter is intended to give some explanation about miscellaneous features.

## 4.1. Mailing

To send an email to a user click on **User Management > Edit User** and click on the envelope/email address of a user. Now you can enter a **Subject** and a message. When finished writing a message, click **Send** to send the email.

---

# Appendix A. Example use case of roles

To illustrate the typical use of roles we will now describe an example setup. It features some often encountered user types and the roles that fit within their profiles.

## A.1. Corpus manager

A corpus manager is responsible for the whole corpus. As such, he/she has full read and write access to all nodes and their resources and can delegate rights to other users.

**Roles:** Archive Manager

## A.2. Researcher

The prototypical researcher is responsible for building a subcorpus and has thus all rights on it. Later on he/she can appoint assistants to assist with changing the content and granting access rights.

**Roles:** Domain Curator + Domain Editor (for the specific subcorpus)

## A.3. Research team

Multiple researchers want to share a corpus, so that they can simultaneously access and change it.

**Roles:** one Domain Curator + a Domain Manager for each subcorpus, to be assigned to a group to which all team members belong

## A.4. Research assistant

The assistant of the researcher can grant or deny access to all nodes of the relevant subcorpus. If necessary he/she can also make changes to the corpus.

**Roles:** Domain Manager (+ Domain Editor if changing the corpus is required)

## A.5. Interested colleague

A colleague asks a certain researcher to get read access to a non-public part of his/her corpus.

**Roles:** none, a username and password for the colleague is required (which can be created by a Domain Manager or Curator).

Of course, there must be some rules allowing this user to read some of the resources.

## A.6. Accidental visitor

A visitor wants to access a publicly opened part of the corpus.

**Roles:** none, even obtaining a user name is not necessary.

The domain must be open for *everybody*.