UNIXvisual Quick Start
UNIXvisual requires a root directory from which to start. UNIXvisual will use the underlying file system or a user specification and there then are two ways to specify the root directory: through the change root directory option or through a specification file (.unix).

A notional system can be defined through a specification file. The file contains text that indicates the root directory, users, user-to-group assignment, objects and their permissions (information listed by using command “ls -l yourobject”). Below is an example:

```
UNIX
root: ./pseudoroot
user: tony, mike, lucy, mary
user: jim
group: tester       mike
group: developer    tony, lucy
group: manager      mary, jim
object: drwsrwxr--  tony:developer  -r   /code
object: rwsrw-r--   tony:developer  -r   /code/program1
object: drwxrw-r--  mary:manager   -r   /document
object: drwxrw-r--  mary:manager   -r   /document/projectA
object: drwxrw-r--  mary:manager   -r   /document/projectA/testers
object: drwxrw-r--  jim:manager    -r   /document/projectB
object: drwxrw-r--  jim:manager    -r   /document/projectB/developers
object: drwxrwx--x  mike:tester    -r   /test
```
User and Group View
(1.1) Import a specification file where root directory is specified (Based on file system defined in specification file and on the real system)

or (1.2) Set root directory (Only based on real file system)

User and Group View allows you to explore the set of objects that are accessible to a single user or to a specific group through the group permission bits

Blue boxes have the sequence of operations. Green boxes explain the functionality.
Choose a user or a group

(2) Choose a user or a group

(3) Choose permissions and conjunction or disjunction

(4) Refresh
When a user is selected, a full permissions check is made. When access is allowed through any of the user, group, or world bits, UNIXvisual will indicate access is allowed.

Red text means the access is blocked at the step.

(5) Files accessible with set permission shown in black. Select one for more information.
When a group is selected, only the group permission bits are checked.
Object View
(0) Import a policy file or specify root directory if neither has been done

(1) Enter an object path and click OK

Object View allows an examination of all access allowed to a single object
(2) Choose a user or all users

(3) Click on a user to see detailed access with checkmark and crossmark

A list of all directories on the path between the root and the selected object

User "root" does not have access to the object.
Click on checkmark or crossmark to see analysis of permissions

Shows the groups clicked user belongs to

User "root" does not have access to the object.
The permission information for objects are clickable.

Permission calculator shows which permission bits are set based on the octal notation of the object clicked.

Hover on a "user(group) permission" tuple in the last row for detailed analysis.
Examining a specific user is available too

<table>
<thead>
<tr>
<th>Object:</th>
<th>/c/pseudoroot/code/program1</th>
<th>OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group:</td>
<td>manager</td>
<td></td>
</tr>
<tr>
<td>User:</td>
<td>jim</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Owner bits</th>
<th>Group bits</th>
<th>Other bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- `/code`
  - User: tony
  - Group: developer
  - Permission bits: 04774 (rwxr-xr-x)

- `/program1`
  - User: tony
  - Group: developer
  - Permission bits: 04764 (rwxr-xr-x)

User "jim" does not have access to the object.
Permission Calculator
(1) Click to toggle Permission Calculator

(2) Switch to converting permission to octal and letter notations

(3) Configure permission here
(1) Switch to decoding octal notation to letter notation and permission bit setting

(2) Enter permission in octal here and click update

(3) Click update

(4) See permission result here