New user management tool

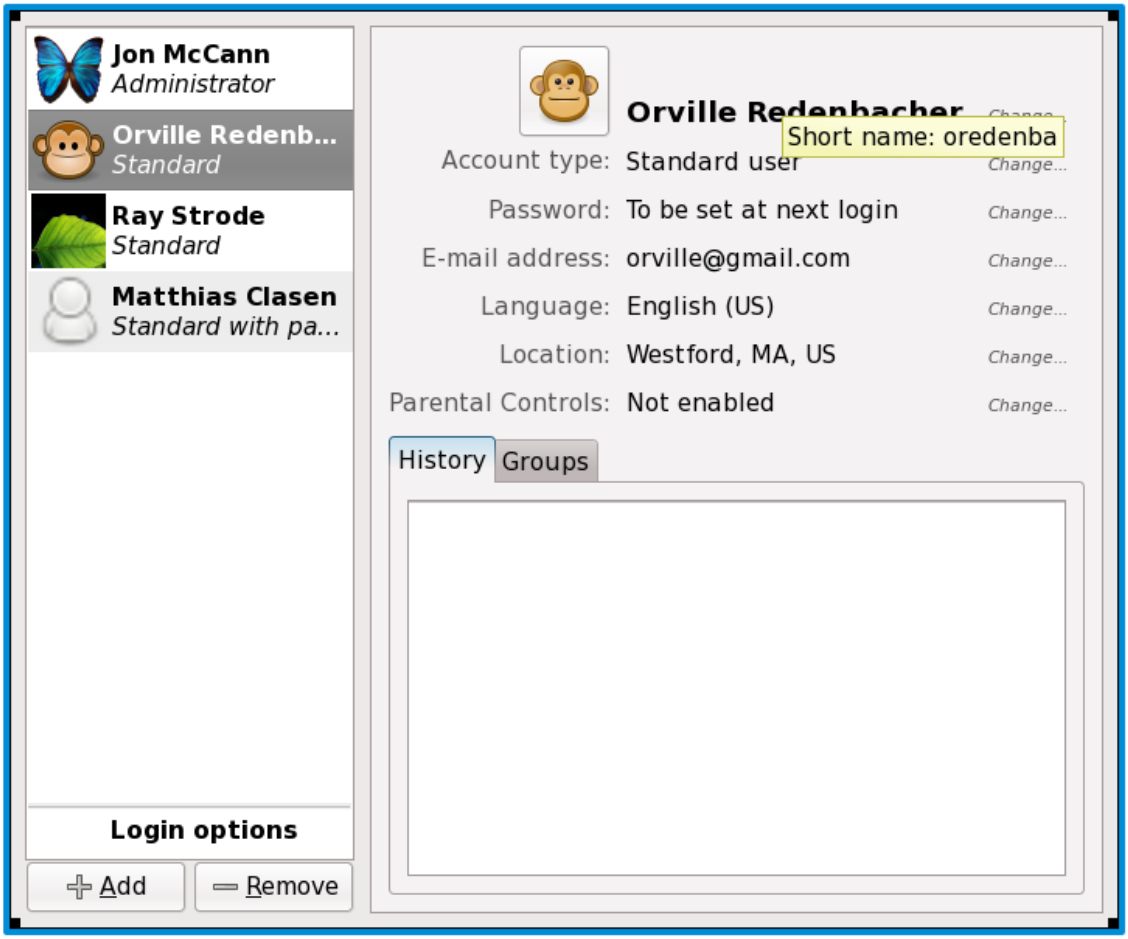
We started to explore what a new user management tool needs to offer in order to improve on the current state of affairs (system-config-users, gnome-about-me, gdmsetup, firstboot). The goal is to bring us on par with Windows and OS X in this area or at least narrow the gap.
In order to understand what the user service beneath this new tool needs to offer, we decided to first take a look at the user interface of a new user management tool.

Target audience and use cases

The target usage scenarios are home and smb, not enterprise customers and big deployments, although we do need to make sure that the tools not totally break down when used in a big deployment scenario, since users should still be able to use it for changing their own account. But we do expect system administrators in enterprise settings to use a web interface to their directory server, not this tool.
One special use case is creating the first user when the system boots for the first time. This is pretty straightforward, and similar to the current situation. The only changes are that we plan on making firstboot a lot simpler in the next iteration, so the user creation dialog might be the only dialog seen. The user creation dialog in firstboot should be similar to the create user dialog that can be seen below.
Another use case is editing my own account data. The proposed workflow for this is to go to the user-switch applet (or bigboard) and select 'Edit personal information' from the menu. That will bring up the new tool, with the current user being selected.
Another use case is account maintenance. The proposed workflow for this is to go to the user-switch applet and select 'Manage accounts' from the menu. That will bring up the new tool with a list of current users, and with options to create, delete or modify users. The same tool will also be available from the System > Administration menu.
Another use case is configuring the login screen. Todo: does this need to be separately in the menus ?
Another use case is temporary access to a computer (guest account). The proposed workflow for this is to offer a "Guest" user on the login screen. Selecting this user will not ask for a password, the account will have limited privileges (Account type: Guest), and all data will be removed at the end of the session, unless the user chooses 'Convert to normal account' from the menu of the user-switch applet. Todo: this is not reflected in the screenshots below.

The main dialog

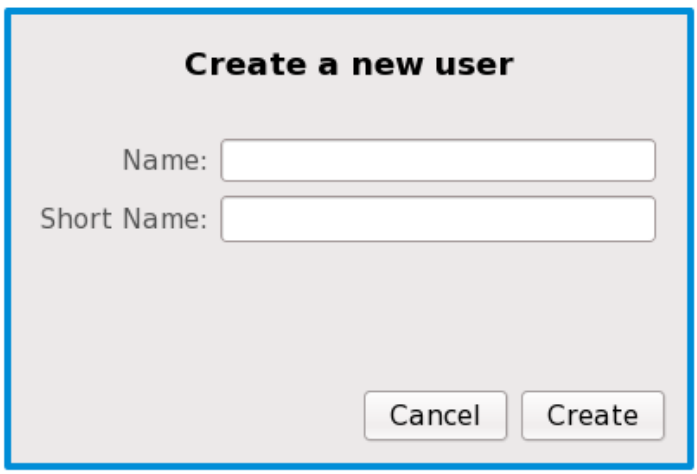
The tool requires different privileges for editing the current user vs modifying other users.



The box on the left is the list of existing users. It should show the same data as the login screen: face and full name, possibly also the account type. The right side shows detail information for the currently selected user.
The history tab can contain information such as the last login time, last time the password was changed, etc.

The Groups tab is a concession to Unix legacy. We can't get away without exposing group information, but at least we can put it out of sight on a separate tab. It is not very useful without some metadata for groups. One point of notice: just like we filter out system users out of the user lists, we should filter out uninteresting groups from the group list.

The Add button brings up a dialog that asks only for the very minimal information that is needed to create a new user.

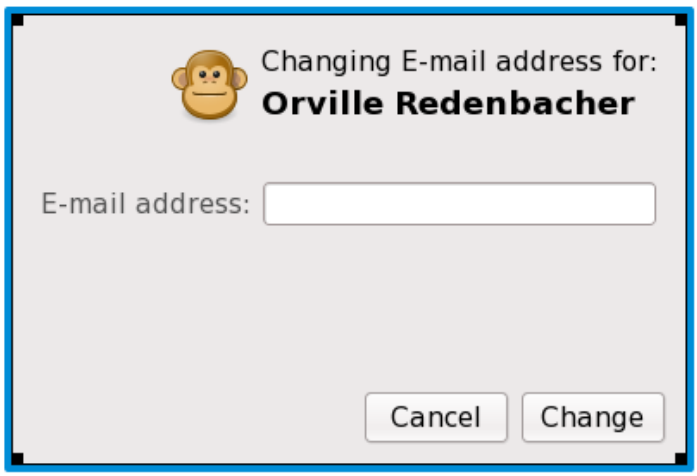


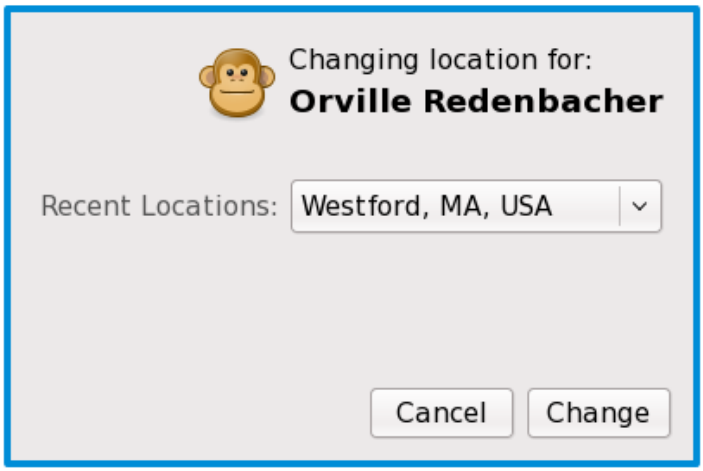
The Name field is first, since we expect the full name to be entered first. The tool then generates a Unix username from the full name and prefills the Short Name field. There was some idea to define a set of rules for this (e.g. Matthias Clasen → mclasen or Jon McCann → jonm) and update the currently used rule based on the corrections that are made to the pregenerated username. We do need to validate the Name and Short Name fields to ensure

that there are no conflicts with existing users. While it is technically possible to have two users with the same Name, we at last want to ask 'Did you really mean to do this ?'.
Likewise, the home directory, the uid and gid, the shell, and other uninteresting pieces of legacy information will be automatically determined by the tool and/or the service. People who have a strong interest in these will probably be better served by useradd anyway.
When the Create button is clicked, the user is created, and further editing can happen in the main window itself, using the Change links in each row. Changing some of the fields may require privileges, even when the current user modifies its own account information (such as changing the account type to Administrator). There should perhaps be some iconic indication when a change requires privileges.

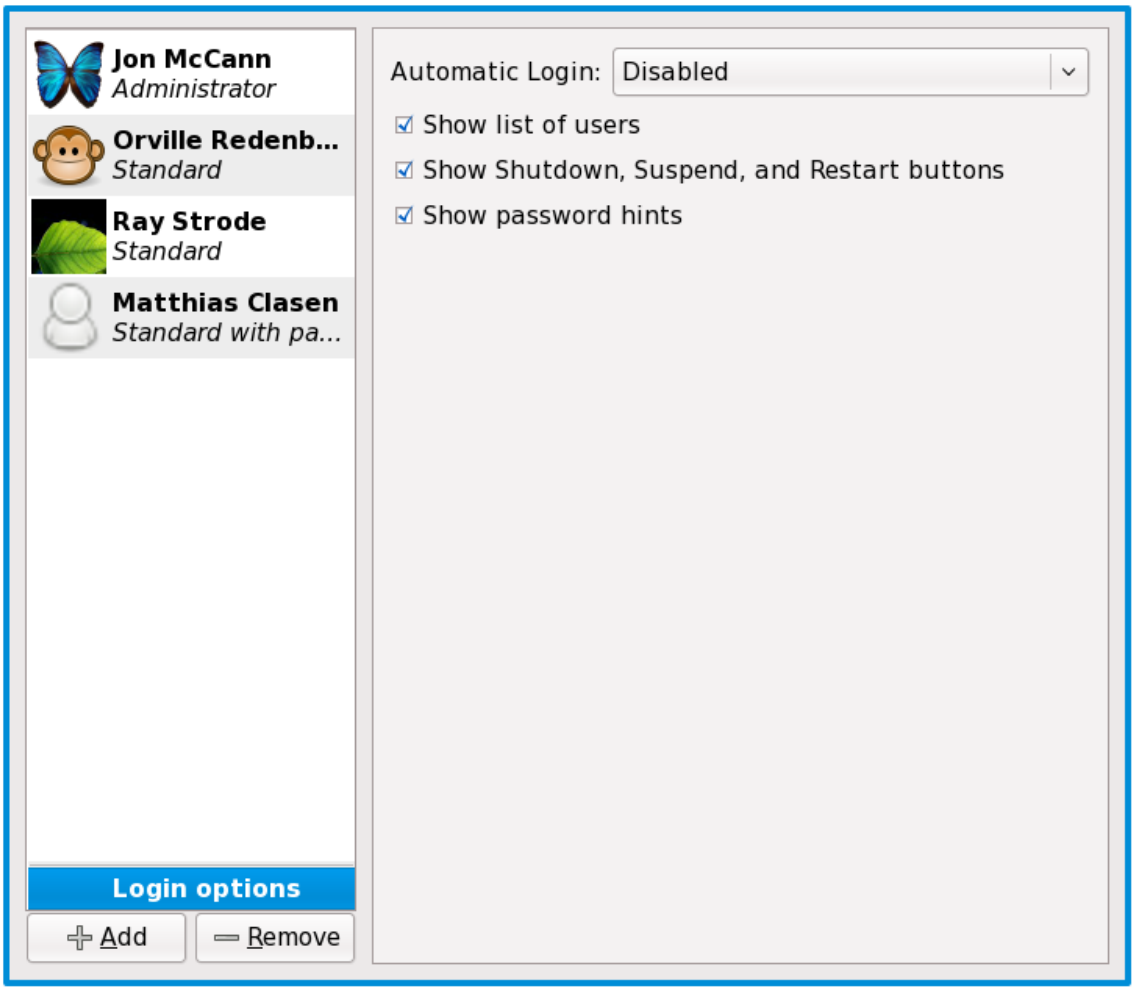
Clicking on the face image brings up a dialog for selecting the user image which offers a set of predefined images, as well as an option to use a webcam (if available), a simple drawing tool (such as MeMaker) or pick an image from the filesystem. Fine point: when showing the predefined faces, we should indicate which ones are already 'taken'. This dialog has not been mocked up yet.
When creating a new user, it initially gets a randomly picked image from the predefined images (excluding those that are already used for a different user)

The Account Type field associates (still to be implemented) PolicyKit 'roles' with the account. The Password field shows information about the kind of password that is currently set. The password change dialog is a bit more complicated than the other change dialogs, and is discussed in a separate section below.





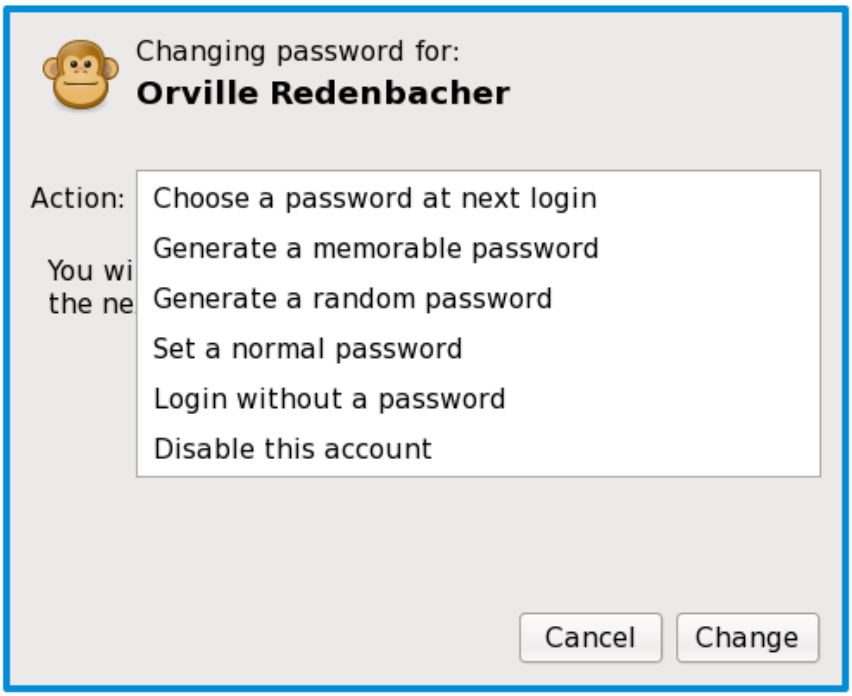
The Email, Language and Location fields are here because they are frequently useful (e.g. the language is needed on the login screen to relieve gdm from storing this information in .dmrc). Also, they are part of the standard LDAP user schema. Todo: these dialogs should perhaps provide some hints as to how these fields are used. E.g. entering an email address here does not create an email account or set up the mail client to use it.
The Parental Controls field is just an idea that needs to be fleshed out.



Beyond direct user management, we also want the new tool to take up some login screen configuration (which is, after all, more or less related to users and passwords).

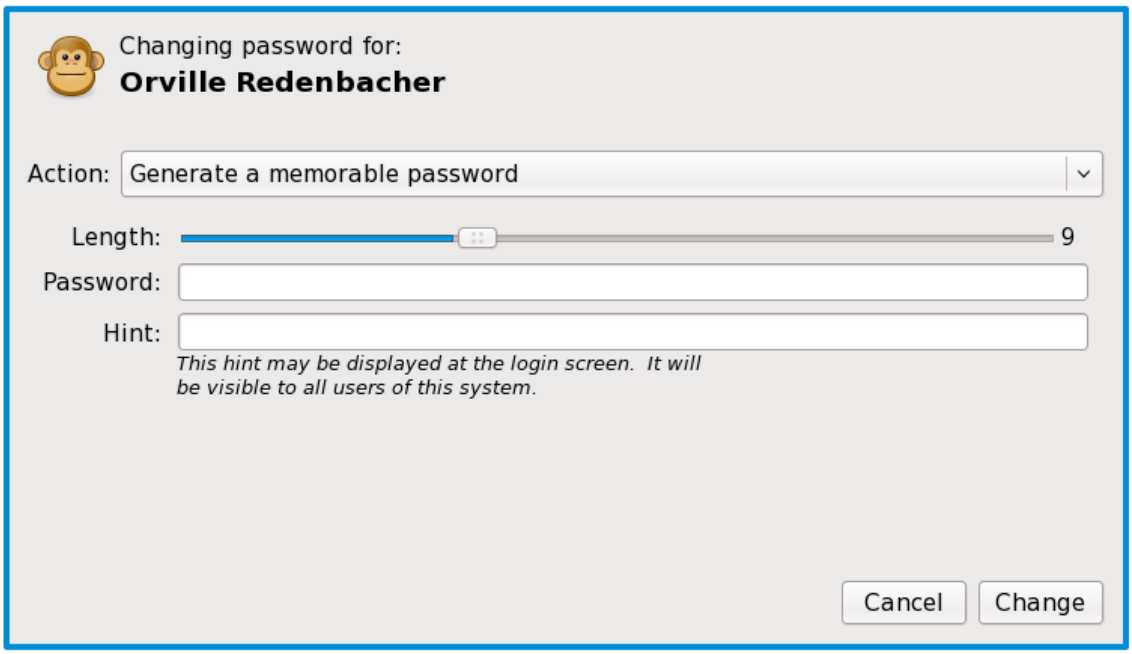
The password dialog

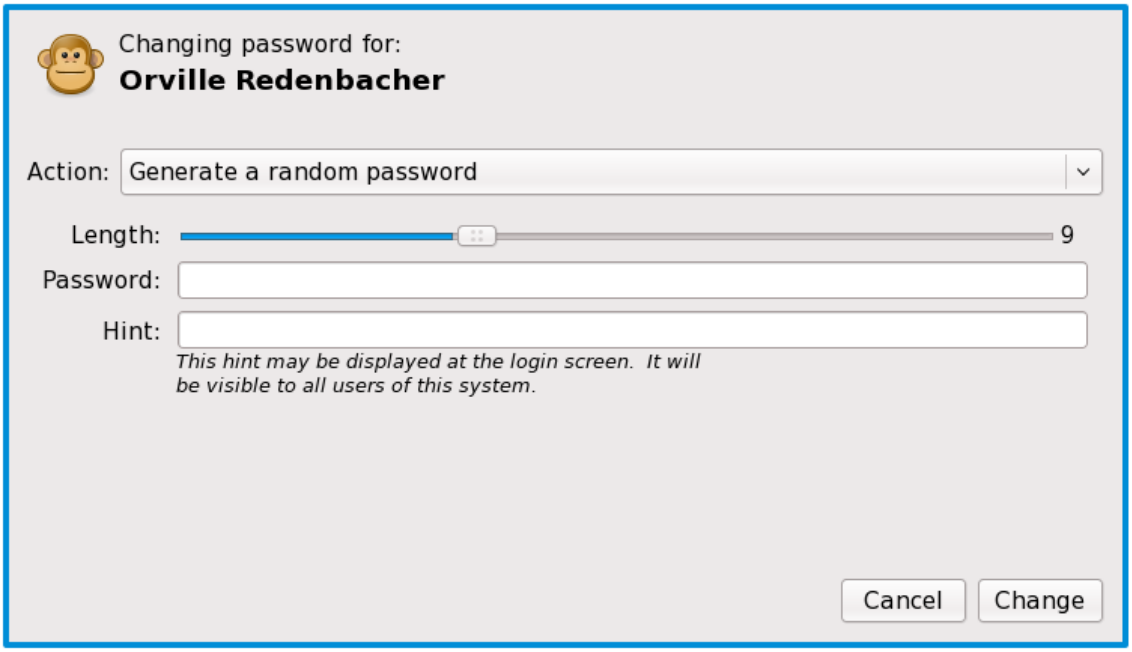
A good idea taken from some other systems is that the tool may offer to generate a password for the user, according to different criteria. One option is to generate a password that is (somewhat) memorable while still avoiding obvious weaknesses such as dictionary words. Another option is to generate a totally random password.



Some other things that the Action field lets us handle elegantly are to defer setting the password until the user logs in the next time, login without a password, or locking the account. Note that some of the actions may be forbidden by system policy, or may require privileges. The dialog adapts to changes in the Action field.

Generated passwords



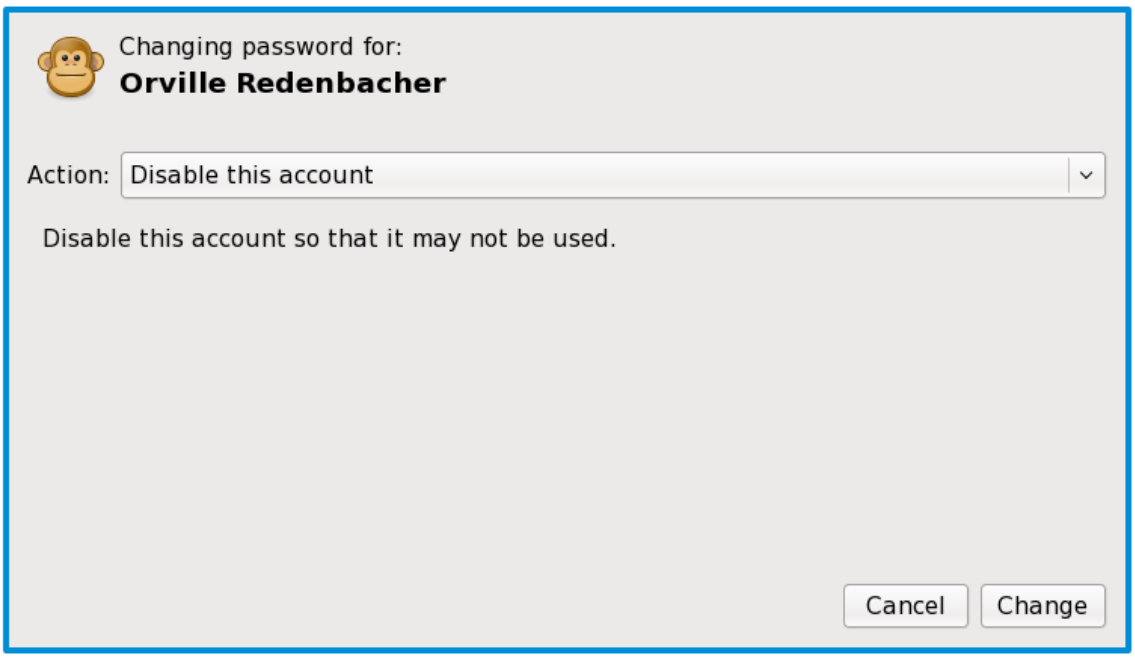


When a password is generated, the Password entry shows the generated password in clear text. The Length field allows to control the length of generated passwords (within system-defined limits).
The Hint field allows to enter a hint that can be shown when the user can't remember the password. We've discussed ways to generate useful hints to go along with generated passwords, including somewhat crazy ideas like computer-generated poetry (cf gnoetry).

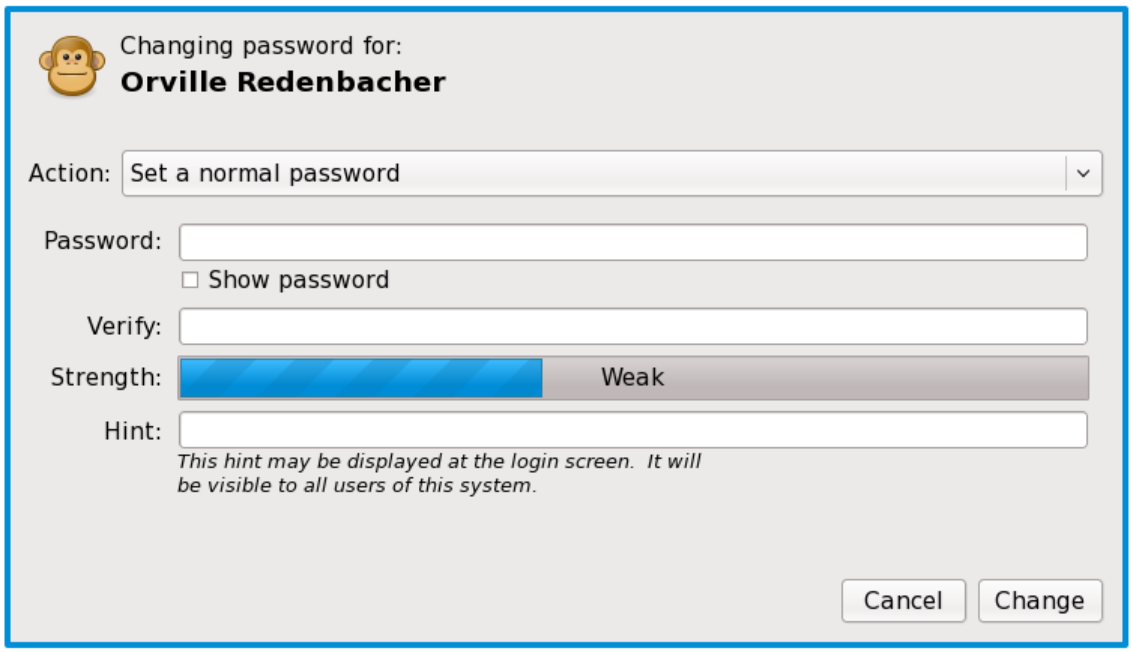
No password

For actions that don't set a password, the dialog will just show an explanatory text.


Changing password for:
Orville Redenbacher
Action:
Login without a password
This will remove the password protection from this account. Are you sure this is what you want to do?
Cancel
Change


Changing password for:
Orville Redenbacher
Action:
Disable this account
Disable this account so that it may not be used.
Cancel
Change

Normal passwords



The Password and Verify entries don't show clear text unless Show password is checked.
The Verify field needs to be identical to the Password field.
The Strength field shows the 'quality' of the password using some yet-to-be-defined algorithm. There is a system-defined minimum quality that a password needs to be acceptable.
The Hint field allows to enter a hint that can be shown when the user can't remember the password.
The Change button is disabled unless the password has sufficient strength (according to some system-defined minimum).
The mockup is missing a Help button, but one point we noted is that Vista has a very good help page explaining how to choose a good password.

Architecture

The basic architecture will consist of a dbus service on the system bus that offers an interface for user management. The service will use PolicyKit to ensure that callers have the necessary privileges for requested operations.

From the UI presentation above, it is clear that we need some more information about users than passwd offers. Additional fields include face, full name, account type (might be covered by groups), email, language, password hint. It would also be nice to get some more transient information, such as 'is the user enrolled in the fingerprint database ?', 'can the user log in without network ?'.

The service will certainly have the expected Create, Delete, Modify functions dealing with individual users. It is well-known that it is a bad idea to have a enumerate-all-users function, since the cost may be prohibitive and user interfaces that rely on such a function will simply not work in large deployments (cf fast-user-switch-applet vs NIS). By the same token, exposing every user as a dbus object will not work very well in such situations. One idea (inspired by LDAP again) is to have a Query function that allows querying for users by certain criteria.