I fought hard with this one.

Ross Dargahi wrote in the official [[http://blog.zimbra.com|Zimbra blog]] about the [[http://blog.zimbra.com/blog/archives/2006/09/look-ma-no-mouse-keyboard-navigation-and-shortcuts-in-the-zimbra-collaboration-suite-and-the-kabuki-ajax-toolkit.html|Keyboard navigation features]] in Kabuki, one of many names of the Zimbra javascript framework.

In case you didn't know: [[http://www.zimbra.com|Zimbra]] is a great groupware with open source building blocks and **a lot** of integration features. A „Zimlet" is one of them. Zimlets are „mashups", basically small javascript applications, that enhance the features of the wonderful web UI Zimbra has.

Coming a long way, the zimbra framework is horribly underdocumented. Zimbra engineers are heavily working on this, but in many cases you simply „don't get" how Zimbra's dealing with certain things.

Tab navigation is one of that things. Like I said, I fought hard with it, but am now able to tell you something about it.

Starting with Ross' blog post, the main thing we're talking about is a DwtTabGroup. The basic idea behind this is actually great: You have a root tab group (the very basic tabs in the main application) and there you can add new tab members (things like input boxes or complex Zimbra controls) and even new tab groups. So you're building up a tree structure. This tree is handled top down. So you're starting with a dialog at the frontmost position having some form fields that you tab through. When you get to the end of the dialog's tab group, the tree is traversed backwards and the next lower level is taken in account.

Using this you can provide a coherent tabbing throughout the whole application.

What we're gonna create is a dialog with form fields. Sounds simple enough.

Well, the first thing to wonder about is: How can I reach „my" tabgroup (the one, my dialog's bound to). There's the first thing, that's not handled well in Kabuki:

Your dialog's tabgroup is currently (as of 6.0.10) only accessible by

myDialog._tabGroup

Although I know, that JavaScript doesn't have private members like other languages do, but it still feels awkward accessing a property, that starts with an underscore.

Then another funny thing: Don't use the ZmZimletBase-method _createDialog or the ZmDialog-class. Although it's pretty obvious to use this function, use DWT's own DwtDialog-class. The reason for this is, that ZmDialog somehow removes any additional tabgroups you may have specified. I don't know, why they do it (that's probably one of these „I don't get it"-cases) but DwtDialog doesn't do this.

Let's start at the beginning. We pop up a dialog when someone single-clicks our panel item:

```
de_dieploegers_tabgroupHandlerObject.prototype.singleClicked = function (
canvas
) {
```

var myDialog,
myView;

myDialog = new DwtDialog({
parent: this.getShell(),
title: "Demo-Dialog",
standardButtons: [DwtDialog.OK_BUTTON, DwtDialog.CANCEL_BUTTON] });

myView = new DwtComposite(this.getShell());

// TODO Add the form

myDialog.setView(myView);

myDialog.popup();

};

That's the basic structure to open a dialog on single click in zimbra. So far, so good.

Now comes the interesting part: Adding the tabgroups. We just add a username and password field to the dialog as an example:

```
myUsername = new DwtInputField({
parent: myView
});
```

myPassword = new DwtPasswordField({
parent: myView

});

Now the tabgroup. We simply create one by issuing:

myTabGroup = new DwtTabGroup(„de_dieploegers_tabgroup_demoTabGroup“);

Then we add the two controls to it as tabgroup members:

myTabGroup.addMember(myUsername);
myTabGroup.addMember(myPassword);

We want the username to become focused when the dialog pops up, so we set the current focus member:

myTabGroup.setFocusMember(myUsername);

That is pretty obvious, right? Now the awkward stuff: Adding our tabgroup to the dialog’s tabgroup:

myDialog._tabGroup.addMember(myTabGroup);

That works. Here’s the complete example:

```
de_dieploegers_tabgroupHandlerObject.prototype.singleClicked = function (
canvas
) {
```

var myDialog,
myView;

myDialog = new DwtDialog({
parent: this.getShell(),
title: "Demo-Dialog",
standardButtons: [DwtDialog.OK_BUTTON, DwtDialog.CANCEL_BUTTON] });

myView = new DwtComposite(this.getShell());

myUsername = new DwtInputField({
parent: myView
});

myPassword = new DwtPasswordField({

```
parent: myView
});

myTabGroup = new DwtTabGroup("de_dieploegers_tabgroup_demoTabGroup");

myTabGroup.addMember(myUsername);
myTabGroup.addMember(myPassword);

myTabGroup.setFocusMember(myUsername);

myDialog._tabGroup.addMember(myTabGroup);

myDialog.setView(myView);

myDialog.popup();

};
```

When you look at it, it’s not as complicated as once thought. The main two things to note are:

* use DwtDialog
* use the private property myDialog._tabGroup to access the tabgroup you have to work with

My next task is to figure out keyboard shortcuts in zimlet. If I find something, I’ll keep you posted.