Admin Settings API

Many of the plugins you will develop will require some settings to change the behaviour of the plugin, e.g. number of result on pages, URLs to web services, and much more.
The *global* settings are defined in its ***settings.php*** file.

Admin Settings API (cont)

For most plugins, the global $settings$ variable is already defined and the script can add

```
29  if · ($ADMIN->fulltree) · {¶
30  · · · · ¶
31  · · · $settings->add(new·admin_setting_configselect¶
32  · · · · · · · · · (¶
33  · · · · · · · · · · · 'forum_displaymode',¶
34  · · · · · · · · · · get_string('displaymode', ·'forum'),¶
35  · · · · · · · · · · get_string('configdisplaymode', ·'forum'),¶
36  · · · · · · · · · · FORUM_MODE_NESTED,¶
37  · · · · · · · · · · forum_get_layout_modes()¶
38  · · · · · · · · · )¶
39  · · · · );¶
40  }¶
```

Admin Settings API (cont)

The admin settings are a kind of extension to the **Form API,** which follows a similar format and accepts similar inputs for the different elements.

For example, for a **checkbox**, you would define an `admin_setting_configcheckbox` object which relates to the `checkbox` element. The available classes are all defined in ***/lib/adminlib.php***

For some plugins, such as local plugins, the global `$settings` variable is not set and, in this case, the settings must be added to the global `$ADMIN` settings.

Admin Settings API (cont)

```
29  if ($hassiteconfig) {
30
31      $settings = new admin_settingpage(
32              'local_plugin',
33          new lang_string('pluginname', 'local_plugin')
34      );
35
36      // Enable plugin
37      $settings->add(new admin_setting_configcheckbox
38          (
39              'local_plugin/enabled',
40              new lang_string('enabled', 'local_plugin'),
41              new lang_string('enabled_help', 'local_plugin'),
42              0,
43              1,
44              0
45          )
46      );
47
48      $ADMIN->add('localplugins', $settings);
49  }
```

Admin Settings API (cont)

- Optionally, if you need either more control over the input form or to store the input separately, an external page can be used to create a link to a page that uses the Form API to manage the settings.

- You can still save the input as settings if you want to, by using the Moodlelib API's `set_config()` function.

- There is a disadvantage to this, however, as highlighted on the Moodle documentation's 'Admin Settings' page

    (https://docs.moodle.org/dev/Admin_settings).

Admin Settings API (cont)

```
29  if ($hassiteconfig) {
30      $ADMIN->add('courses', new admin_externalpage
31          (
32              'local_plugin',
33              get_string('pluginname', 'local_plugin'),
34              new moodle_url('/local/local_plugin/configsettings.php')
35          )
36      );
37  }
```

Admin Settings

- `admin_setting_configcheckbox`
- `admin_setting_configcheckbox_with_advanced`
- `admin_setting_configcheckbox_with_lock Form API`
- `admin_setting_configcolourpicker`
- `admin_setting_configdirectory`
- `admin_setting_configduration`
- `admin_setting_configduration_with_advanced`
- `admin_setting_configempty`
- `admin_setting_configexecutable`
- `admin_setting_configfile`
- `admin_setting_confightmleditor`
- `admin_setting_configiplist`

Admin Settings (cont)

- `admin_setting_configmixedhostiplist`
- `admin_setting_configmulticheckbox`
- `admin_setting_configmulticheckbox2`
- `admin_setting_configmultiselect`
- `admin_setting_configmultiselect_modules`
- `admin_setting_configpasswordunmask`
- `admin_setting_configpasswordunmask_with_advanced`
- `admin_setting_configportlist`
- `admin_setting_configselect`
- `admin_setting_configselect_with_advanced`
- `admin_setting_configselect_with_lock`

Admin Settings (cont)

- `admin_setting_configstoredfile`

- `admin_setting_configtext`

- `admin_setting_configtextarea`

- `admin_setting_configtext_with_advanced`

- `admin_setting_configtext_with_maxlength`

- `admin_setting_configthemepreset`

- `admin_setting_configtime`

'Admin Settings' page (https://docs.moodle.org/dev/Admin_settings).