Email templates in opEvents

opEvents releases after 1.1.0 include a facility to define your own templates for email notifications (ie. emails as part of an event action or because of an escalation).

The setup is very simple and requires basically just two steps:

1. deciding on what information to present and in what order, and
2. associating a contact with a particular template.

Defining custom email templates

The configuration file `install/EventEmails.nmis` contains the definitions of all email templates. As shipped it includes one special template named `default`, which is chosen for all email contacts without a template association, and one example for a very concise report. You can of course modify these existing templates or define your own. Here are these two example templates:

```
'default' => {
        'subject'=> [ "event.date", ' ',"node.name", ' ', "event.event" ],
        'body' => [
                "Event Context:\n", "link.eventcontext", "\n\n",
                "Node Properties:\n","node.*","\n\n",
                "Event Properties:\n","event.*","\n","\n",
        ],
 },

 'veryshort' => {
         'subject' => [ "node.name", ' ', "event.event" ],
        'body' => [
                "The event \"","event.event", "\" for host \"", "node.name",
                "\" occurred at ", "event.date",
                ".\nAll event details can be found here: ", "link.eventcontext",
                "\n", "All node details can be found here: ", "link.nodecontext",
                "\n\n",
        ],
 },
```

The syntactic rules for templates are simple but flexible:

- An email template consists of two named lists of items to include, one for the mail subject and one for the body.
- The template lists define what order you want the items to be shown.
- Entries like `node.XYZ`, `event.ABC` and `link.EFG` (or `macro.HIJ`) are treated specially, but only if they are isolated separate items.
  For example, the text entry "The lazy fox likes node.group" is **not** special because here `node.group` is part of a longer bit of text.
- All non-special plain text items are included as-is, ie. for generating static header or footer text.
- You can write your texts either single- our double-quoted; within double-quoted strings all the standard backslash escapes work - e.g. "\t" for a tab character).
  As these are all perl strings, the "@", "$" and "%" characters are special: within double-quoted strings you'll need to prefix those with a backslash.
  No such problems arise when you use single-quoted strings, but then no backslash escapes whatsoever are recognized in single-quoted strings.
- Newlines are **NOT** created automatically. You have to include "\n" for that (within your double-quoted strings, or as separate item) whereever you want a line break.
- The line breaks in the actual template definition are not relevant.

- All special entries, of the form `node.XYZ` or `event.XYZ` are replaced by the node or event property `XYZ`.
  If the property is structured, with sub-elements (e.g. `event.script`), then these are shown on separate lines and indented with a tab character.
  The common node properties are documented here, and the event properties are listed on this page.
- From opEvents 2.2 onwards, special entries of the form `macro.XYZ` are replaced by the value of the macro named `XYZ` (which are defined in the `opCommon.nmis` configuration file, in the section `macro`).
- The entry `link.eventcontext` is replaced by a hyperlink to the event context page.
- The entry `link.nodecontext` creates a hyperlink to the node context page.
  Both of these hyperlink generators rely on the setting `opevents_url_base` in `opCommon.nmis` to set an appropriate schema and hostname for the hyperlink. if this setting is missing you'll see `http://localhost/` in the created hyperlinks.
- Entries of the form `node.*` are translated into a list of all "standard summary" node properties, which are defined in `opCommon.nmis`, key `opevents_gui_event_node_summary_list`.
  For this summary, the property name is shown first, followed by a colon and a space, the property value, and a newline; repeated for each property. There is a trailing newline.
- The form `event.*` works similarly, but for event properties and is not restricted to a default set. It also creates a trailing newline.
- Both special forms `node.*` and `event.*` are disabled in the subject template, because that must contain just a single line of text. You also must not create multiple lines of text in the subject template (using "\n").

Associating an email contact with a template

The configuration file `Contacts.nmis` contains the list of (email) contacts, and needs to be adjusted if you want templates other than `default` to be used. The example file as shipped shows where the email template is set:

```
'contact1' => {
        'Contact' => 'Contact1',
...
        'Email' => 'nobody@localhost',
        'EmailTemplate' => 'default',
```

Simply change the value after `EmailTemplate` to the name of your desired template. If opEvents cannot find the named template, or if no `EmailTemplat` setting is present, then the `default` template will be used. If that template doesn't exist or work, then you will receive your emails formatted with a hardcoded fallback template that includes a warning notice at the top.