

Coding style for Python source code

Main specification of HCE API for python language based on Chromium project specifications:

<http://www.chromium.org/chromium-os/python-style-guidelines>

<http://google-styleguide.googlecode.com/svn/trunk/pyguide.html>

<http://www.chromium.org/developers/coding-style>

Differences from Chromium specifications:

1. Unit tests and functional tests placed in the directory on the same level in to the “tests” directory.
2. Test source code file name “test_<module_name>_<test_name>”.
3. Exception class name “<moduleNamePrefix>Err”.
4. Number of empty lines between methods definition – two.
5. Number of lines before class definition – two.
6. Constants import from module definition:
import transport.Consts as TRANSPORT_CONSTS
i = TRANSPORT_CONSTS.ID_GENERATOR
7. Line length max 120 characters.
8. Functional tests placed in to the “ftests” directory.
9. Extended comments of complex logic in code branches.
10. Comments and headers:

```
'''  
HCE project, Python bindings, Distributed Tasks Manager application.  
Event objects definitions.
```

```
@package: dtm  
@author bgv bgv.hce@gmail.com  
@link: http://hierarchical-cluster-engine.com/  
@copyright: Copyright &copy; 2013-2014 IOIX Ukraine  
@license: http://hierarchical-cluster-engine.com/license/  
@since: 0.1  
'''
```

```
import zlib
```

```
##Brief descriptions  
#  
#More detailed description  
#More detailed description  
class MyClass(object):  
  
    ##Class constants description  
    #Detailed description  
    #  
    CLASS_CONSTANT1 = 1  
    CLASS_CONSTANT2 = 2  
  
    ##constructor  
    #constructor description  
    #  
    def __init__(self, parameter1, parameter2=None):  
        ##@var field1 brief description  
        #field1 detailed description  
        self.field1 = parameter1  
  
    ##Method brief description  
    #  
    #@param parameter1 Parameter 1 description  
    #@param parameter2 Parameter 2 description  
    #@return nothing  
    def myMethod(self, parameter1, parameter2):  
        self.field1 = parameter1
```