

```
# encoding: utf-8
```

```
import gvsig
from gvsig import getResource
from gvsig import currentView, currentLayer

from java.io import File
from org.gvsig.scripting.app.extension import ScriptingExtension
from org.gvsig.app import ApplicationLocator
from org.gvsig.tools.swing.api import ToolsSwingLocator
from org.gvsig.andami import PluginsLocator

from InitApp import MYTOOL_NAME

class BlockInformationExtension(ScriptingExtension):
    def __init__(self):
        pass

    def isEnabled(self):
        layer = currentLayer()
        if layer == None:
            return False
        return layer.getProperty("MyApplicationLayer") == True

    def isVisible(self):
        view = currentView()
        if view == None:
            return False
        return view.getProperty("MyApplicationView") == True

    def execute(self, actionCommand, *args):
        actionCommand = actionCommand.lower()
        if actionCommand == "blocks-info":
            view = currentView()
            if view.getProperty("MyApplicationView") == True:
                view.getWindowOfView().getMapControl().setTool(MYTOOL_NAME)

    def selfRegisterToolButton():
        application = ApplicationLocator.getManager()

        icon = File(getResource(__file__, "images", "blocks-info.png")).toURI().toURL()

        iconTheme = ToolsSwingLocator.getIconThemeManager().getCurrent()
        iconTheme.registerDefault(
            "scripting.myfirstapp",
            "action",
            "blocks-info",
            None,
            icon
        )

        extension = BlockInformationExtension()
        actionManager = PluginsLocator.getActionInfoManager()
        action = actionManager.createAction(
            extension,
            "blocks-info", # Action name
            "Show blocks information", # Text
            "blocks-info", # Action command
            "blocks-info", # Icon name
            None, # Accelerator
            1009000000, # Position
            "Show blocks information 3" # Tooltip
        )
        action = actionManager.registerAction(action)
        application.addSelectableTool(action, "MyApplicationToolBar")

def main(*args):
    selfRegisterToolButton()
```