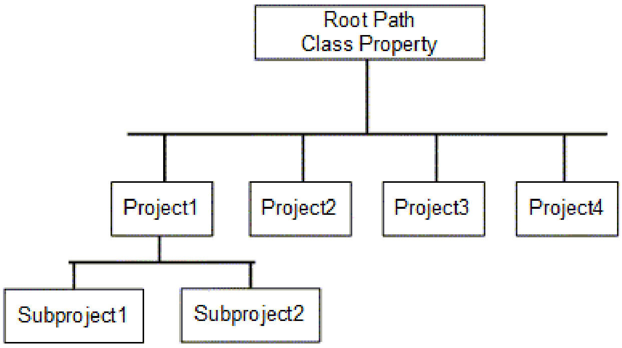
a TANGO class to manage settings : SettingsManager

- **Principle:**
  This idea is to standardize and centralize the settings management for the control system.
  It will have to be accessible from client applications and servers written in C++, Java or Python.
  The most flexible system to store the settings is to use files on disk. Users will be able to manage sub system, copies or edition (device/attribute renaming).
  All settings files will be store under a "RootPath" directory.
  A subdirectory will be created for each accelerator system.
  To be accessible from all languages, a TANGO class will manage read, write and apply settings for attributes.
  Each instance of this class will manage a system (SRRF, RIPS, LINAC, ...) files under its own subdirectory fixed by a TANGO property.

  This class will be able to:
  - Write a settings file with an optional list of attribute as input argument. This class will be responsible to do not write above its subdirectory.
  - Apply settings to attributes.
  - Get file content to preview file with string as output argument.
  - Read a settings file and return content in a pipe (att1 name, att1 value, att2 name, att2 value, ...)



- For Project4, the device **sys/settings/project4** will manage files found under **$RootPath/Project4**
- For Subproject2 the device **sys/settings/subproject2** will manages files found under **$RootPath/Project1/Subproject2**
- The device file path is given by **$RootPath** and **SettingsFilesPath** device property
- To generate a settings file, the attribute list could be taken from **DefaultAttributeList** device property or from input command argument.

- **How to control settings changes**
  When a settings file is applied a thread is started.
  this thread read the applied attributes (write part) and compare with the applied file content.
  If is different, the attribute **LastAppliedFile** is set to **ALARM**
  The list of changed attributes could be found in **AlarmAttributes** attribute

  The default period to read attributes and compare is fixed to 2 seconds.
  But it could be changed using **CheckChangePeriod** attribute.

  See:
  - How to use client API.
  - TANGO device class user's guide.
  - Client/Server classes reference.

Java Client Api for SettingsManager

- How to use API
  To generate or apply a settings file, read or write attributes will be done.
  The number and the response time of attributes is unknown and can be a quiet long.
  For this reason, the generate/apply calls will be executed and returned after a while.
  The generate/apply action will be done with a long TIMEOUT value.
  The default value is 5 seconds but it could be changed using:
  - **SETTINGS_TIMEOUT** VM property
  - **SETTINGS_TIMEOUT** environment variable

  The **JAVA client API** provides a listener (SettingsManagedListener) called after execution, giving an event (SettingsManagedEvent) containing execution information.

```
import org.tango.settingsmanager.client.SettingsManagerClient;
import org.tango.settingsmanager.client.SettingsManagedListener;
import org.tango.settingsmanager.client.SettingsManagedEvent;

    - - - - -
     private SettingsManagerClient settingsClient;
    - - - - -
    //  Create a settings manager client and add a listener
    settingsClient = new SettingsManagerClient(settingsProjectName);
    settingsClient.addSettingsAppliedListener(new SettingsManagedListener() {
        @Override
        public void settingsManaged(SettingsManagedEvent event) {
            settingsAppliedPerformed(event);
        }
    });
    - - - - -
```

Then to apply settings, the SettingsManagerClient object will call with apply method.
This method will propose a file chooser to select a file to apply.
This method will return the selected file name. If no selection, the file name will be null.
If the file name is not null, after a while, the specified listener will be called.
The **SettingsManagedEvent** event propose methods to get execution information.
See following example:

```
private void applySettingsItemActionPerformed(ActionEvent evt) {
    try {
        //  Apply settings
        String fileName = settingsClient.applySettings(this);
        if (fileName!=null) {
            System.out.println("Applying " + fileName);
        }
    }
    catch (DevFailed e) {
        ErrorPane.showErrorMessage(this, null, e);
    }
}
- - - - - -
private void settingsAppliedPerformed(SettingsManagedEvent event) {
    String fileName = event.getFileName();
    switch (event.getAction()) {
        case SettingsManagerClient.APPLIED:
            //  Display applied results
            if (event.hasFailed()) {
                ErrorPane.showErrorMessage(new JFrame(),
                    "Applying file " + fileName, event.getDevFailed());
            } else {
                JOptionPane.showMessageDialog(new JFrame(),
                    "Settings loaded from  " + event.getFileName());
            }
            break;
        case SettingsManagerClient.GENERATED:
            //  Display generated results
            if (event.hasFailed()) {
                ErrorPane.showErrorMessage(new JFrame(),
                    "Generated file " + fileName, event.getDevFailed());
            } else {
                JOptionPane.showMessageDialog(new JFrame(),
                    "Settings saved in  " + event.getFileName());
            }
            break;
    }
}
```

SettingsManager Tango Java Class

Contents :

SettingsManager Class Identification :

Contact	: at esrf.fr - accelerator-control
Class Family	: Archiving
Platform	: All Platforms
Bus	: Not Applicable
Manufacturer	: none
Manufacturer ref.	:

SettingsManager Class Inheritance :

- Tango::DeviceImpl
  - SettingsManager

SettingsManager Class Description :

This idea is to standardize and centralize the settings management for the control system.
It will have to be accessible from client applications and servers written in C++, Java or Python.

The most flexible system to store the settings is to use files on disk.
Users will be able to manage sub system, copies or edition (device/attribute renaming).
All settings files will be store under a $SETTINGS_ROOT directory.
A subdirectory will be created for each accelerator system.
To be accessible from all languages, a TANGO class will manage read, write and apply settings for attributes.
Each instance of this class will manage a system (SRRF, RIPS, LINAC, ...) files under its
own subdirectory fixed by a TANGO property.
This class will be able to:
- Write a settings file with an optional list of attribute as input argument. This class will be responsible
to do not write above its subdirectory.
- Get file content to preview file with string as output argument.
- Apply settings to attributes.
- Read a settings file and return content in a pipe (att1 name, att1 value, att2 name, att2 value, ...)

SettingsManager Properties :

Class Properties			
Name	**Description**	**Type**	**Default Value**
RootPath	Root path for all settings systems	String	none

Device Properties			
Name	**Description**	**Type**	**Default Value**
SettingsFilesPath	Path under the RootPath to read/write settings file.	String	none
DefaultAttributeList	Default attribute list to generate settings file if no attributes sent in WriteSettingsFile command	String[]	none
SettingsTimeout	timeout to write settings on specified devices in milliseconds.	int	3000
Debug	Add trace mode if true	boolean	false

SettingsManager Class Commands				
Name	**Input type**	**Output type**	**Level**	**Description**
State	DEV_VOID	DEV_STATE	OPERATOR	This command gets the device state (stored in its device_state data member) and returns it to the caller.
Status	DEV_VOID	CONST_DEV_STRING	OPERATOR	This command gets the device status (stored in its device_status data member) and returns it to the caller.
GetSettingsFileContent	DEV_STRING	DEV_STRING	OPERATOR	read the specified file content
GenerateSettingsFile	DEVVAR_STRINGARRAY	DEV_VOID	OPERATOR	the device will generate a settings file for specified attributes
ApplySettings	DEV_STRING	DEV_VOID	OPERATOR	Apply settings from specified file to attributes.
Reset	DEV_VOID	DEV_VOID	OPERATOR	Reset ALARM state on device
FileExists	DEV_STRING	DEV_BOOLEAN	OPERATOR	Check if specified file exists.
DeleteFile	DEV_STRING	DEV_VOID	OPERATOR	Delete specified file.
RenameFile	DEVVAR_STRINGARRAY	DEV_VOID	OPERATOR	Rename source file to target name.
GetFileList	DEV_STRING	DEVVAR_STRINGARRAY	OPERATOR	Get the file and directory list in specified path.

Command State :

This command gets the device state (stored in its device_state data member) and returns it to the caller.

State Definition		
Input Argument	Tango::DEV_VOID	none
Output Argument	Tango::DEV_STATE	Device state
DisplayLevel	OPERATOR	..
Inherited	true	..
Abstract	false	..
Polling Period	1000	..
Command allowed for	All states	..

Command Status :

This command gets the device status (stored in its device_status data member) and returns it to the caller.

Status Definition		
Input Argument	Tango::DEV_VOID	none
Output Argument	Tango::CONST_DEV_STRING	Device status
DisplayLevel	OPERATOR	..
Inherited	true	..
Abstract	true	..
Polling Period	Not polled	..
Command allowed for	All states	..

Command GetSettingsFileContent :

read the specified file content

GetSettingsFileContent Definition		
Input Argument	Tango::DEV_STRING	Specified file name.
Output Argument	Tango::DEV_STRING	The specified file content
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command allowed for	All states	..

Command GenerateSettingsFile :

the device will generate a settings file for specified attributes

GenerateSettingsFile Definition		
Input Argument	Tango::DEVVAR_STRINGARRAY	This array must contain file name. If no attributes are defined as property, it must contain attribute names. It can also contain author names and comments. Each element needs to start with: FILE: for file name ATTRIBUTE: for each attribute name AUTHOR: for author name COMMENTS: for comments
Output Argument	Tango::DEV_VOID	
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command NOT allowed for	- MOVING	..

Command ApplySettings :

Apply settings from specified file to attributes.

ApplySettings Definition		
Input Argument	Tango::DEV_STRING	The specified file name
Output Argument	Tango::DEV_VOID	
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command NOT allowed for	- MOVING	..

Command Reset :

Reset ALARM state on device

Reset Definition		
Input Argument	Tango::DEV_VOID	

Output Argument	Tango::DEV_VOID	
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command NOT allowed for	• MOVING	..

Command FileExists :

Check if specified file exists.

FileExists Definition		
Input Argument	Tango::DEV_STRING	File name
Output Argument	Tango::DEV_BOOLEAN	true if specified file exists, false otherwise
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command allowed for	All states	..

Command DeleteFile :

Delete specified file.

DeleteFile Definition		
Input Argument	Tango::DEV_STRING	File name
Output Argument	Tango::DEV_VOID	
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command allowed for	All states	..

Command RenameFile :

Rename source file to target name.

RenameFile Definition		
Input Argument	Tango::DEVVAR_STRINGARRAY	Source file name Target file name
Output Argument	Tango::DEV_VOID	
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command allowed for	All states	..

Command GetFileList :

Get the file and directory list in specified path.

GetFileList Definition		
Input Argument	Tango::DEV_STRING	Path to be added to the RootPath
Output Argument	Tango::DEVVAR_STRINGARRAY	FILE: or DIR: followed by file or directory name
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command allowed for	All states	..

SettingsManager Class Attributes

Name	Inherited	Abstract	Attr. type	R/W type	Data type	Level	Description
SettingsPath	false	false	Scalar	READ	Tango::DEV_STRING	OPERATOR	Path where settings files are read and write.
LastGeneratedFile	false	false	Scalar	READ	Tango::DEV_STRING	OPERATOR	
LastAppliedFile	false	false	Scalar	READ	Tango::DEV_STRING	OPERATOR	
CheckChangePeriod	false	false	Scalar	READ_WRITE	Tango::DEV_LONG	OPERATOR	Period (in seconds) to check if settings have change since last appalied.
DefaultAttributes	false	false	Spectrum	READ	Tango::DEV_STRING	OPERATOR	The default attribute list from property.
AlarmAttributes	false	false	Spectrum	READ	Tango::DEV_STRING	OPERATOR	List of attributes in ALARM

There is no dynamic attribute defined.

Attribute SettingsPath :

Path where settings files are read and write.

Attribute Definition	
Attribute Type	Scalar
R/W Type	READ
Data Type	Tango::DEV_STRING
Display Level	OPERATOR
Inherited	false
Abstract	false
Polling Period	1000
Memorized	Not set
Read allowed for	All states

Attribute Properties	
label	Settings path
unit	
standard unit	
display unit	
format	
max_value	
min_value	
max_alarm	
min_alarm	
max_warning	
min_warning	
delta_time	
delta_val	

Attribute Event Criteria	
Periodic	Not set
Relative Change	Not set
Absolute Change	Not set
Archive Periodic	Not set
Archive Relative Change	Not set
Archive Absolute Change	Not set
Push Change event by user code	false
Push Archive event by user code	false
Push DataReady event by user code	false

Attribute LastGeneratedFile :

Attribute Definition	
Attribute Type	Scalar
R/W Type	READ
Data Type	Tango::DEV_STRING
Display Level	OPERATOR
Inherited	false
Abstract	false
Polling Period	1000
Memorized	Not set

Attribute Properties	
label	Last generated file
unit	
standard unit	
display unit	
format	
max_value	
min_value	
max_alarm	

Attribute Event Criteria	
Periodic	Not set
Relative Change	Not set
Absolute Change	Not set
Archive Periodic	Not set
Archive Relative Change	Not set
	Not

Read allowed for	All states

min_alarm	
max_warning	
min_warning	
delta_time	
delta_val	

Archive Absolute Change	set
Push Change event by user code	false
Push Archive event by user code	false
Push DataReady event by user code	false

Attribute LastAppliedFile :

Attribute Definition	
Attribute Type	Scalar
R/W Type	READ
Data Type	Tango::DEV_STRING
Display Level	OPERATOR
Inherited	false
Abstract	false
Polling Period	1000
Memorized	Not set
Read allowed for	All states

Attribute Properties	
label	Last applied file
unit	
standard unit	
display unit	
format	
max_value	
min_value	
max_alarm	
min_alarm	
max_warning	
min_warning	
delta_time	
delta_val	

Attribute Event Criteria	
Periodic	Not set
Relative Change	Not set
Absolute Change	Not set
Archive Periodic	Not set
Archive Relative Change	Not set
Archive Absolute Change	Not set
Push Change event by user code	false
Push Archive event by user code	false
Push DataReady event by user code	false

Attribute CheckChangePeriod :

Period (in seconds) to check if settings have change since last appalied.

Attribute Definition	
Attribute Type	Scalar
R/W Type	READ_WRITE

Attribute Properties	
label	Check Change Period

Attribute Event Criteria	
Periodic	Not set
Relative Change	Not

Data Type	Tango::DEV_LONG
Display Level	OPERATOR
Inherited	false
Abstract	false
Polling Period	1000
Memorized	true
Write hardware at init.	true
Read allowed for	All states
Write allowed for	All states

unit	sec.
standard unit	
display unit	
format	
max_value	300
min_value	1
max_alarm	
min_alarm	
max_warning	
min_warning	
delta_time	
delta_val	

	set
Absolute Change	1
Archive Periodic	Not set
Archive Relative Change	Not set
Archive Absolute Change	Not set
Push Change event by user code	false
Push Archive event by user code	false
Push DataReady event by user code	false

Attribute DefaultAttributes :

The default attribute list from property.

Attribute Definition	
Attribute Type	Spectrum (512)
R/W Type	READ
Data Type	Tango::DEV_STRING
Display Level	OPERATOR
Inherited	false
Abstract	false
Polling Period	1000
Memorized	Not set
Read allowed for	All states

Attribute Properties	
label	
unit	
standard unit	
display unit	
format	
max_value	
min_value	
max_alarm	
min_alarm	
max_warning	
min_warning	
delta_time	
delta_val	

Attribute Event Criteria	
Periodic	Not set
Relative Change	Not set
Absolute Change	Not set
Archive Periodic	Not set
Archive Relative Change	Not set
Archive Absolute Change	Not set
Push Change event by user code	false
Push Archive event by user code	false
Push DataReady event by user code	false

Attribute AlarmAttributes :

List of attributes in ALARM

Attribute Definition	
Attribute Type	Spectrum (512)
R/W Type	READ
Data Type	Tango::DEV_STRING
Display Level	OPERATOR
Inherited	false
Abstract	false
Polling Period	1000
Memorized	Not set
Read allowed for	All states

Attribute Properties	
label	Alarm attributes
unit	
standard unit	
display unit	
format	
max_value	
min_value	
max_alarm	
min_alarm	
max_warning	
min_warning	
delta_time	
delta_val	

Attribute Event Criteria	
Periodic	Not set
Relative Change	Not set
Absolute Change	Not set
Archive Periodic	Not set
Archive Relative Change	Not set
Archive Absolute Change	Not set
Push Change event by user code	false
Push Archive event by user code	false
Push DataReady event by user code	false

SettingsManager Class States	
Name	**Description**
ON	The device is ready to apply settings.
ALARM	The device has applied settings, but at least one attribute apply has failed. Or Settings have been applied but changed by someone.