

Preferences in MITK

The screenshot displays the MITK (Medical Image Toolkit) interface. The main window shows a 3D visualization of a coronal CT scan slice. A 'Preferences' dialog box is open, showing the 'Hotkeys' section. The 'DataManager' window is also visible, showing a list of nodes. The 'Properties' panel on the right shows the properties of the selected node, 'patient_003_ct'.

Preferences Dialog - Hotkeys

Action	Hotkey
Delete selected nodes	Del
Global Reinit	Ctrl+, R
Load	Ctrl+, L
Make all nodes invisible	V
Reinit selected nodes	R
Save selected nodes	Ctrl+, S
Show Node Information	Ctrl+, I
Toggle visibility of selected nodes	D

DataManager - Selected Nodes

Name	Data Type	Visibility
1 widget1Plane	Geometry2D...	true
2 widget2Plane	Geometry2D...	true
3 widget3Plane	Geometry2D...	true
4 patient_003_ct	Image	true

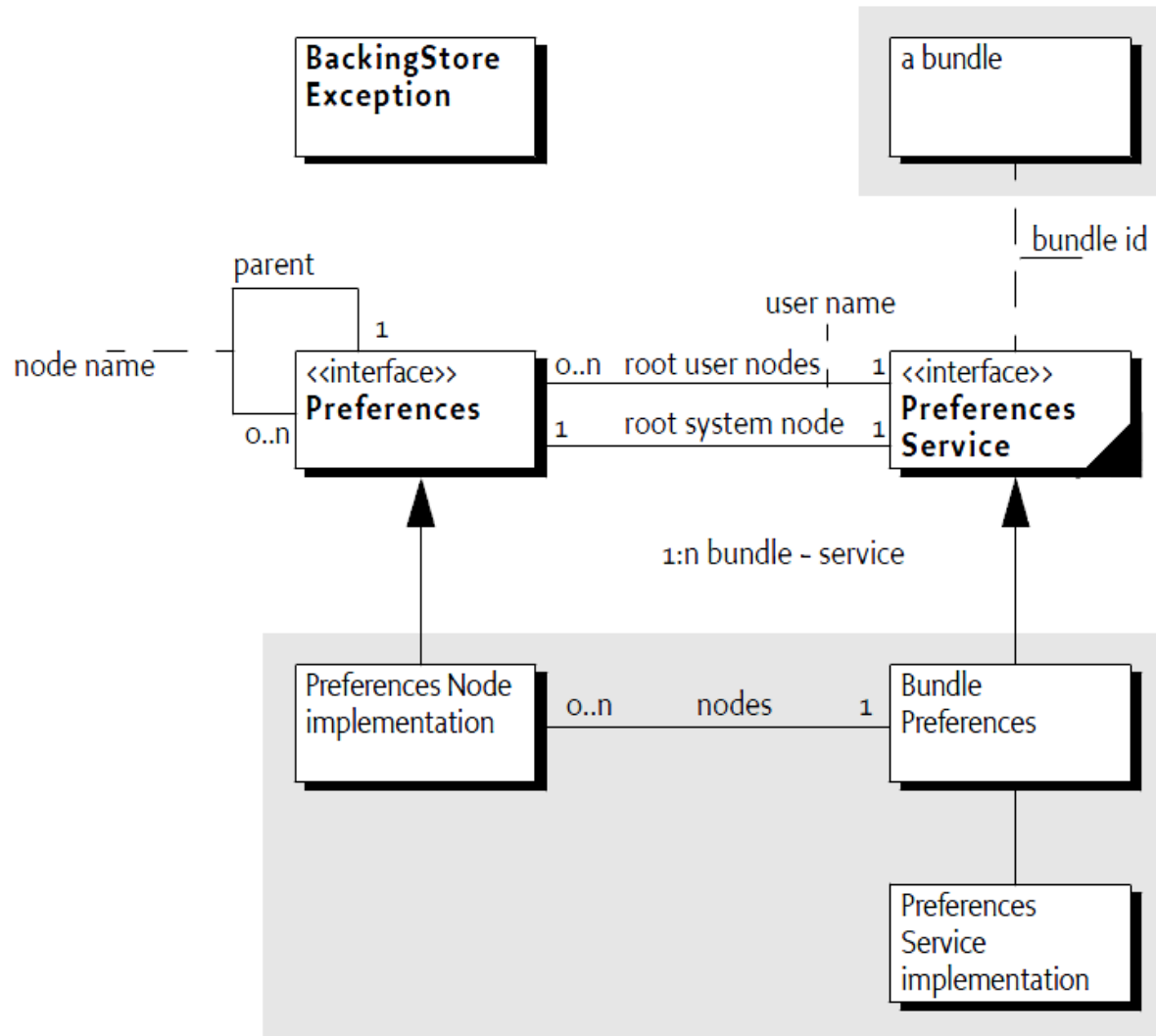
Properties Panel

Name	Value
LookupTable	n/a
TransferFunction	RTTI typeName: Reference Cour Modified Time: E Debug: Off Observers: none
binary	false
bounding box	false
color	
in plane resample extent by geometry	false
layer	0
levelwindow	L:665.5 W:3377
name	patient_003_ct
opacity	1

Log Console

Time	Level	Message
274...	Info	Library C:\(home)\mueller\m...
275...	Info	Trying to load class QmitkDat...
275...	Info	Library C:\(home)\mueller\m...
276...	Info	Trying to load class QmitkDataManagerHotkeysPrefPage with manifest name cherryIQPreferencePage from bundle org.mitk.gui.qt.datamanager
276...	Info	Library C:\(home)\mueller\workspace\mitk\bin\mitk\ExtBundles\org.mitk.gui.qt.datamanager\bin\Release\liborg_mitk_gui_qt_datamanager.d...
283...	Info	Preferences successfully imported from X:\settings.xml

Preferences Class Diagram



```
#include "cherryIPreferencesService.h"
#include "cherryPlatform.h"

...

// get the preferences service
IPreferencesService::Pointer preferencesService =
    Platform::GetServiceRegistry().GetServiceById<IPreferencesService>
    ( IPreferencesService::ID );

...

// get a preferences NODE
IPreferences::Pointer generalPreferences
    = preferencesService->GetSystemPreferences()->Node( "/General" );
```

```
// put some values in a preferences NODE
generalPreferences->PutBool("a bool variable", true);
generalPreferences->Put("a string", "blaaaa");
generalPreferences->PutInt("int", 3);

// get some values from a preferences NODE
// always deliver a default value in case the key does not exist
bool boolVar = generalPreferences->GetBool("a bool variable", false);
std::string stringVar = generalPreferences->Get("a string", "");
int intVar = generalPreferences->GetInt("int", 0);

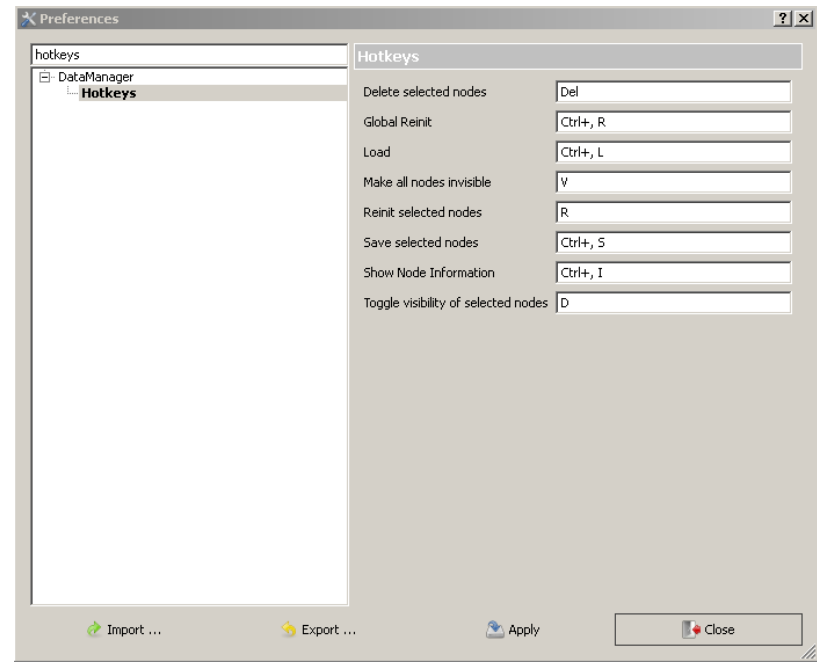
// force immediate save of preferences node
generalPreferences->Flush();
```

1. Inherit from IQtPreferencesPage

```
#include „cherryIQPreferencePage.h“
...
class QmitkMyPreferencePage : public cherry::IQPreferencePage
{
    Q_OBJECT
    /// Build UI in constructor
    QmitkGeneralPreferencePage(QWidget* parent = 0, Qt::WindowFlags f = 0);
    /// Invoked when the OK button was clicked in the preferences dialog
    virtual bool PerformOk();
    /// Invoked when the Cancel button was clicked in the preferences dialog
    virtual void PerformCancel();
    ...
};
```

2. Add to files.cmake

```
SET( INTERNAL_CPP_FILES
    QmitkMyPreferencePage.cpp
)
SET( MOC_H_FILES
    src/internal/QmitkMyPreferencePage.h
)
```



3. Add to manifest.cpp

```
#include "src/internal/QmitkMyPreferencePage.h"

POCO_BEGIN_NAMED_MANIFEST(cherryIQtPreferencePage, cherry::IQtPreferencePage)

    POCO_EXPORT_CLASS(QmitkMyPreferencePage)

POCO_END_MANIFEST
```

4. In plugin.xml:

```
<extension point="org.opencherry.ui.preferencePages">
    <page id="org.mitk.gui.qt.<mybundle>.QmitkMyPreferencePage" name="My Preferences"
        class="QmitkMyPreferencePage">
        <keywordreference id="org.mitk.gui.qt.<mybundle>.QmitkMyPreferencePageKeywords">
            </keywordreference>
        </page>
</extension>
```

5. (Adding keywords in plugin.xml:)

```
<extension point="org.opencherry.ui.keywords">  
  <keyword id="org.mitk.gui.qt.<mybundle>.QmitkMyPreferencePageKeywords"  
    label="my preferences keywords separated by spaces"></keyword>  
</extension>
```