



XPand Porting Guide

Porting to Qt 5 (for Windows only)

For Windows user only the Qt version moves from Qt 4 to Qt 5. Some changes of your code must be needed if you use Qt classes. To update your code to Qt 5 read the porting guide provided by Qt <http://doc.qt.io/qt-5/portingguide.html>.

New behavior on string encoding management

With Qt 5 the expected encoding when creating a QString from a char* has changed from Latin1 to Utf8.

Unfortunately Latin1 and Utf8 do not encode the characters in the same way except for all ASCII (encoded in the 0-127 interval) values.

To avoid encodings bugs, the char* parameters usage in QString methods has been deactivated because char cannot provide any encoding information.

Now, a compilation error occurs if you try to use the following calls:

```
char* myChar = ...;
QString myQString = myChar;
```

However QString(char[]) constructor for strings literals is allowed (even if QString(char *) is forbidden) if the string contains only ASCII (0-127) characters.

Thus you can continue to use the following calls:

```
QString myQString = "My value";
```

Be careful when writing code that only ASCII strings literals are used since they have exactly the same storage whatever the encoding system.

Moreover only ASCII characters (0-127) must be used in the source code even if your editor supports more extended characters.

If you have errors due to QString from char* creation, follows these simple rules:

1. Avoid useless conversions

Try to avoid useless conversion, check if there is a public method taking the type you have instead of converting it.

2. Conversions of strings from or to tcl encoding

You are in this context if:

- You are in a parse function and you read data from the tcl interpreter arguments.
- You send a result to the tcl interpreter.
- You need a string that will be read by the tcl interpreter (writing a project file for example, or in a save state method)

Tcl uses the utf8 encoding, so an explicit conversion using QString utf8 conversion methods is required.

To read a string from a tcl context, use the QString::fromUtf8() method.

To send a string to tcl use the QString::toUtf8() method.

Reading QString from parse argument example:

```
int
MyModule::parse(Tcl_Interp* t, int argc, char** argv)
{
    const QString cmd = QString::fromUtf8(argv[1]);

    if (cmdCheck(cmd, QLatin1String("loadReport"), QLatin1String("filename")))
    {
```



```
loadReport(QString::fromUtf8(argv[2]));
return TCL_OK;
}
// ...
}
```

Converting a QString in a saveState example:

```
void HxMyComputeModule::saveState(FILE* fp, const char* dataDir, SaveFlags saveFlags)
{
    HxCompModule::saveState(fp, dataDir, saveFlags);

    // Some QString to transform.
    QString customState = ....;

    fprintf(fp, "%s \n", customState.toUtf8().constData());
}
```

3. Conversion of strings from or to Avizo encoding

You are in this context if:

- You need to create a QString from a char* coming from a public Avizo method.
- You need to create a char* from a QString to send it as parameter to a public Avizo method.

You can keep the implicit translation only if the string literal contains only ASCII character (character encoded in the 0-127 interval).

If the compiler raise an error on a string literal (it occurs mostly on linux or on special cases) prefer the usage of QLatin1String to fix it instead of the other methods since it identifies all strings literals that contain only ASCII characters (0-127).

In the other cases, use the toQString() and toByteArray() functions from the hxqt/QxStringUtils.h to perform the conversion between QString and char* in the Avizo context.

Be careful with the toByteArray() function which returns a QByteArray, to get the char* you had to call the QByteArray::constData() method.

Converting char* to QString exemple:

```
...
int unitIndex = ...;
QString unitNameStr = toQString(HxUnitsHelpers::getUnitName(unitIndex));
QString unitSymbolStr = toQString(HxUnitsHelpers::getUnitSymbol(unitIndex));
...
char* name = ...;
QString string = toQString(name);
...
QString label = tr("MyLabel");
...
QString string = "MyString";
// or
QString string = QLatin1String("MyString");
```

Converting QString to char* exemple:

```
...
QString plotName;
PzCurve* m_plot = new PzCurve(toByteArray(plotName).constData());
...
```

Changes to custom tcl parse methods

Due to the encoding modifications some functions (cmdHelp, cmdCheck, cmdError) rise a warning if you use the command as a char*. To remove the warning store the command in a QString instead.

```
// Old
```



```
MyModule::parse(Tcl_Interp* t, int argc, char** argv)
{
    char* cmd = argv[1];
    ....
}
// New
MyModule::parse(Tcl_Interp* t, int argc, char** argv)
{
    const QString cmd = QString::fromUtf8(argv[1]);
    ...
}
```

Changes to HxParseUtils.h (Package amiramesh)

- » The TCL_PARSE_INIT macro is removed.
- » The CMD macro is removed. Use the cmdCheck method (from HxParseUtils.h header) instead.

```
void
// Old
if (CMD("setScale")) {
    ...
}
// New
if (cmdCheck(cmd, "setScale")) {
    ...
}
```

Changes to HxParamBundle (Package amiramesh)

- » The HxParamBundle::duplicate() virtual method signature is changed to return HxParamBundle type instead of HxParamBase type.

Changes to HxParameter (Package amiramesh)

- » The HxParameter::duplicate() virtual method signature is changed to return HxParameter type instead of HxParamBase type.

Changes to HxColormapGradient (Package hxcolor)

- » The HxColormapGradient::duplicate() virtual method signature is changed to return HxColormapGradient type instead of HxData type.

Changes to HxColormapHue (Package hxcolor)

- » The HxColormapHue::isTransparent() virtual method is removed.
- » The HxColormapHue::getRGBAu() virtual method is removed.

Changes to HxMain (Package hxcommon)

- » The HxMain::tabWidget() method signature is changed to return QxToolPanelWidget type instead of QxToolPanelTabWidget type.

Changes to hxapplication (Package hxcore)

- » The hxapplication::handlePendingEvents() method signature is changed to use void handlePendingEvents (int allowViewerRedraw=1, QEventLoop::ProcessEventsFlags flags=QEventLoop::AllEvents) instead of void handlePendingEvents (int allowViewerRedraw=1).
- » The hxapplication::registerCustomOivType() method is removed.



Changes to HxEditor (Package hxcore)

- » The HxEditor::attachData() method signature is changed to use void attachData (HxData *data, bool createGUI= true) instead of void attachData (HxData *data).
- » The HxEditor::createPortWidgets() method signature is changed to use void createPortWidgets (HxData *data, bool createGUI= true) instead of void createPortWidgets (HxData *data).

Changes to HxFileDialog (Package hxcore)

- » The HxFileDialog::getExistingDirectory() method signature is changed to use QString getExistingDirectory (const QString &caption=QString(), const QString &dir=QString()) instead of QString getExistingDirectory (const QString &caption=QString()).

Changes to HxInterpreter (Package hxcore)

- » The HxInterpreter::productCompatibility() method is removed.
- » The HxInterpreter::loadScript() method signature is changed to use static int loadScript (const QString &filename) instead of static int loadScript (const char *filename) .

Changes to HxObject (Package hxcore)

- » The HxObject::parseAllInterfaces() method is removed.

Changes to HxViewer (Package hxcore)

- » The HxViewer::viewX() method signature is changed to use void viewX (bool reverse=false, bool doViewAll= true) instead of void viewX (bool reverse=false).
- » The HxViewer::viewY() method signature is changed to use void viewY (bool reverse=false, bool doViewAll= true) instead of void viewY (bool reverse=false).
- » The HxViewer::viewZ() method signature is changed to use void viewZ (bool reverse=false, bool doViewAll= true) instead of void viewZ (bool reverse=false).

Changes to QxViewer (Package hxcore)

- » The QxViewer::snapshot() method signature is changed to use void snapshot (const char *filename, int numTilesX, int numTilesY, int nrSamples=1.0, int offscreenWidth=0, int offscreenHeight=0, bool stereo=false, const char *filename2=0, bool calledFromGUI=false, QxSnapshotDialog::OutputType output=QxSnapshotDialog::TO_FILE) instead of void snapshot (const char *filename, int numTilesX, int numTilesY, int nrSamples=1.0, int offscreenWidth=0, int offscreenHeight=0, bool stereo=false, const char *filename2=0, bool calledFromGUI=false) .
- » The QxViewer::viewX() method signature is changed to use void viewX (bool reverse=false, bool doViewAll= true) instead of void viewX (bool reverse=false).
- » The QxViewer::viewY() method signature is changed to use void viewY (bool reverse=false, bool doViewAll= true) instead of void viewY (bool reverse=false).
- » The QxViewer::viewZ() method signature is changed to use void viewZ (bool reverse=false, bool doViewAll= true) instead of void viewZ (bool reverse=false).

Changes to HxAnnaScalarField3 (Package hxfield)

- » The HxAnnaScalarField3::duplicate() virtual method signature is changed to return HxAnnaScalarField3 type instead of HxData type.

Changes to HxAnnaVectorField3 (Package hxfield)



» The `HxAnnaVectorField3::duplicate()` virtual method signature is changed to return `HxAnnaVectorField3` type instead of `HxData` type.

Changes to `HxColorField3` (Package `hxfield`)

» The `HxColorField3::duplicate()` virtual method signature is changed to return `HxColorField3` type instead of `HxData` type.

Changes to `HxComplexScalarField3` (Package `hxfield`)

» The `HxComplexScalarField3::duplicate()` virtual method signature is changed to return `HxComplexScalarField3` type instead of `HxData` type.

Changes to `HxComplexVectorField3` (Package `hxfield`)

» The `HxComplexVectorField3::duplicate()` virtual method signature is changed to return `HxComplexVectorField3` type instead of `HxData` type.

Changes to `HxField3` (Package `hxfield`)

» The `HxField3::duplicate()` virtual method signature is changed to return `HxField3` type instead of `HxData` type.

Changes to `HxHexaComplexScalarField3` (Package `hxfield`)

» The `HxHexaComplexScalarField3::duplicate()` virtual method signature is changed to return `HxHexaComplexScalarField3` type instead of `HxData` type.

Changes to `HxHexaComplexVectorField3` (Package `hxfield`)

» The `HxHexaComplexVectorField3::duplicate()` virtual method signature is changed to return `HxHexaComplexVectorField3` type instead of `HxData` type.

Changes to `HxHexaField3` (Package `hxfield`)

» The `HxHexaField3::duplicate()` virtual method signature is changed to return `HxHexaField3` type instead of `HxData` type.

Changes to `HxHexaGrid` (Package `hxfield`)

» The `HxHexaGrid::duplicate()` virtual method signature is changed to return `HxHexaGrid` type instead of `HxData` type.

Changes to `HxHexaScalarField3` (Package `hxfield`)

» The `HxHexaScalarField3::duplicate()` virtual method signature is changed to return `HxHexaScalarField3` type instead of `HxData` type.

Changes to `HxHexaVectorField3` (Package `hxfield`)

» The `HxHexaVectorField3::duplicate()` virtual method signature is changed to return `HxHexaVectorField3` type instead of `HxData` type.

Changes to `HxLabelLattice3` (Package `hxfield`)

» The `HxLabelLattice3::duplicate()` virtual method signature is changed to return `HxLabelLattice3` type instead of `HxData` type.

Changes to `HxLattice3` (Package `hxfield`)



» The `HxLattice3::copyData()` method signature is changed to return `bool` type instead of `void` type.

Changes to `HxRegColorField3` (Package `hxfield`)

» The `HxRegColorField3::duplicate()` virtual method signature is changed to return `HxRegColorField3` type instead of `HxData` type.

Changes to `HxRegField3` (Package `hxfield`)

» The `HxRegField3::duplicate()` virtual method signature is changed to return `HxRegField3` type instead of `HxData` type.

Changes to `HxRegScalarField3` (Package `hxfield`)

» The `HxRegScalarField3::duplicate()` virtual method signature is changed to return `HxRegScalarField3` type instead of `HxData` type.

Changes to `HxRegVectorField3` (Package `hxfield`)

» The `HxRegVectorField3::duplicate()` virtual method signature is changed to return `HxRegVectorField3` type instead of `HxData` type.

Changes to `HxScalarField3` (Package `hxfield`)

» The `HxScalarField3::duplicate()` virtual method signature is changed to return `HxScalarField3` type instead of `HxData` type.

Changes to `HxSurface` (Package `hxfield`)

» The `HxSurface::duplicate()` virtual method signature is changed to return `HxSurface` type instead of `HxData` type.

Changes to `HxSurfaceField` (Package `hxfield`)

» The `HxSurfaceField::duplicate()` virtual method signature is changed to return `HxSurfaceField` type instead of `HxData` type.

Changes to `HxSurfaceScalarField` (Package `hxfield`)

» The `HxSurfaceScalarField::duplicate()` virtual method signature is changed to return `HxSurfaceScalarField` type instead of `HxData` type.

Changes to `HxSurfaceVectorField` (Package `hxfield`)

» The `HxSurfaceVectorField::duplicate()` virtual method signature is changed to return `HxSurfaceVectorField` type instead of `HxData` type.

Changes to `HxTetraField3` (Package `hxfield`)

» The `HxTetraField3::duplicate()` virtual method signature is changed to return `HxTetraField3` type instead of `HxData` type.

Changes to `HxTetraGrid` (Package `hxfield`)

» The `HxTetraGrid::duplicate()` virtual method signature is changed to return `HxTetraGrid` type instead of `HxData` type.

Changes to `HxTetraScalarField3` (Package `hxfield`)

» The `HxTetraScalarField3::duplicate()` virtual method signature is changed to return `HxTetraScalarField3` type instead of `HxData` type.



Changes to HxTetraVectorField3 (Package hxfield)

» The HxTetraVectorField3::duplicate() virtual method signature is changed to return HxTetraVectorField3 type instead of HxData type.

Changes to HxUniformCoord3 (Package hxfield)

» The HxUniformCoord3::duplicate() virtual method signature is changed to return HxUniformCoord3 type instead of HxData type.

Changes to HxUniformLabelField3 (Package hxfield)

» The HxUniformLabelField3::copyData() method is removed.

Changes to HxUniformVectorField3 (Package hxfield)

» The HxUniformVectorField3::duplicate() virtual method signature is changed to return HxUniformVectorField3 type instead of HxData type.

Changes to HxVectorField3 (Package hxfield)

» The HxVectorField3::duplicate() virtual method signature is changed to return HxVectorField3 type instead of HxData type.

Changes to HxField3 (Package hxfield)

» The HxField3::computeMemSize() virtual method signature is changed to use virtual mcint64 computeMemSize (const McBox3f &box, const McDim3l &dims) const type instead of virtual mcint64 computeMemSize (const McBox3f &box, const HxLattice3 &lattice) const type.

Changes to McInterface (Package mclib)

» The McInterface::getInterface() method signature is changed to use McTypedObject * getInterface (const McTypeInfo *t, bool forceAttachInterfaces= true) const instead of McTypedObject * getInterface (const McTypeInfo *t) const.

Changes to McInterfaceOwner (Package mclib)

» The McInterfaceOwner::getInterface() method signature is changed to use McTypedObject * getInterface (const McTypeInfo *t, bool forceAttachInterfaces= true) const instead of McTypedObject * getInterface (const McTypeInfo *t) const .

Changes to McVec4 (Package mclib)

» The McVec4::operator* method signature is changed to use MCLIB_API friend McVec4 < realT > operator* (const McVec4 < realT > &va, realT fac) instead of McVec4 < realT > operator* (realT fac) .

Changes to McRawData2D (Package mclib)

» The McRawData2D::McRawData2D() method signature is changed to use McRawData2D (int nx, int ny, int elemSize, int bytesPerPixel, mcuint64 bytesPerLine, void *data) type instead of McRawData2D (int nx, int ny, int elemSize, int bytesPerPixel, mculong bytesPerLine, void *data) type.

» The McRawData2D::McRawData2D() method signature is changed to use McRawData2D (int nx, int ny, int elemSize, mcuint64 bytesPerLine, void *data) type instead of McRawData2D (int nx, int ny, int elemSize, mculong bytesPerLine, void *data) type.

» The McRawData2D::getBytesPerLine() method signature is changed to return mcuint64 type instead of mculong type.

Changes to McRawData3D (Package mclib)



- » The McRawData3D::getNumElems() method signature is changed to return muint64 type instead of mculong type.
- » The McRawData3D::getOffset() method signature is changed to return muint64 type instead of mculong type.
- » The McRawData3D::accessSlice() method signature is changed to use void accessSlice (Orientation orientation, int sliceNumber, void *&first, int &bytesPerPixel, muint64 &bytesPerLine) const type instead of void accessSlice (Orientation orientation, int sliceNumber, void *&first, int &bytesPerPixel, int &bytesPerLine) const type.