

MimeTypes

Caspar J. Goch

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer

Welcome to the DKFZ!

dkfz.

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer

What is a MimeType?

- In general a standard identifier indicating the type of data
- In MITK they provide meta-data about a specific data format
- Rule of thumb
 - If it has the same binary representation but different extensions it is the same MimeType
 - If it has a different binary representation it is not (even if you are using the same reader/writer for it)

Why do I need it?

- Every reader/writer instance needs an associated `MimeType`
- Used for categorizing data formats/GUI strings

How do I use it?

```
mitk::PointSetReaderService::PointSetReaderService()  
: AbstractFileReader(IOMimeTypes::POINTSET_MIMETYPE_NAME(), "MITK Point Set Reader")  
{  
  RegisterService();  
}
```

- Use `X_MIMETYPE_NAME()` instead of `X_MIMETYPE()` to use a registered one
- Can also use `SetMimeType(const CustomMimeType& mimeType)` but most Reader/Writer register in constructor

How to create my own?

```
CustomMimeType DiffusionIOMimeTypes::CONNECTOMICS_MIMETYPE()  
{  
    CustomMimeType mimeType(CONNECTOMICS_MIMETYPE_NAME());  
    std::string category = "Graphs";  
    mimeType.SetComment("Connectomics Networks");  
    mimeType.SetCategory(category);  
    mimeType.AddExtension("cnf");  
    return mimeType;  
}
```

```
std::string DiffusionIOMimeTypes::CONNECTOMICS_MIMETYPE_NAME()  
{  
    static std::string name = IOMimeTypes::DEFAULT_BASE_NAME() + ".graphs.connectomics-networks";  
    return name;  
}
```

```
std::vector<CustomMimeType*> DiffusionIOMimeTypes::Get()  
{  
    std::vector<CustomMimeType*> mimeTypes;  
    // order matters here (descending rank for mime types)  
    mimeTypes.push_back(CONNECTOMICS_MIMETYPE().Clone());  
    return mimeTypes;  
}
```

You do not need your own IOMimeTypes class, your reader/writer/IO can also create it itself

How do create my own?

```
class DiffusionModuleActivator : public us::ModuleActivator
{
public:

    void Load(us::ModuleContext* context)
    {
        us::ServiceProperties props;
        props[ us::ServiceConstants::SERVICE_RANKING() ] = 10;

        m_MimeTypes = mitk::DiffusionIOMimeTypes::Get();
        for (std::vector<mitk::CustomMimeType*>::const_iterator mimeTypeIter = m_MimeTypes.begin(),
             iterEnd = m_MimeTypes.end(); mimeTypeIter != iterEnd; ++mimeTypeIter)
        {
            context->RegisterService(*mimeTypeIter, props);
        }
        /* Reader/Writer creation here */
    }

    void Unload(us::ModuleContext*)
    {
        for (unsigned int loop(0); loop < m_MimeTypes.size(); ++loop)
        {
            delete m_MimeTypes.at(loop);
        }
        /* Reader/Writer uncreation here */
    }
}
```

High ranking = override previously registered MimeTypes

But I need to do really complicated stuff because my files are only valid if it is Tuesday and there are at least three of them and you have internet connection via low-orbit satellite. So the extension alone won't work for me. Please help?

```
class DiffusionImageNrrdMimeType : public CustomMimeType
{
public:
    DiffusionImageNrrdMimeType();
    virtual bool AppliesTo(const std::string &path) const;
    virtual DiffusionImageNrrdMimeType* Clone() const;
};
```

You can derive your own MimeType which can check whether it applies

A photograph of the German Cancer Research Center (DKFZ) building, a modern multi-story structure with a central glass tower and balconies. In the foreground, there is a paved plaza with several water fountains and yellow benches. The sky is blue with some clouds.

Thank you
for your attention!

Further information on www.dkfz.de

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer