
ImageNetworkReader: an ITK Class for Loading Images Across a Network

Release 1.00

Daniel Blezek¹

November 24, 2008

¹Mayo Clinic, Rochester, MN, USA

Abstract

This document describes a new class for the Insight Toolkit (ITK www.itk.org) for reading images across a network. Network access is handled by CURL (<http://curl.haxx.se/libcurl/>), an Open Source library for client-side URL transfer. CURL supports the FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE protocols.

This paper is accompanied with source code and a test to load a NRRD file across the network and verify that it is read properly.

Latest version available at the [Insight Journal](http://hdl.handle.net/1926/1338) [<http://hdl.handle.net/1926/1338>]
Distributed under [Creative Commons Attribution License](http://creativecommons.org/licenses/by/4.0/)

Contents

1	Basic Usage	1
2	Advanced Usage	2
3	Software Requirements	2

The rationale for this contribution to the Insight Journal is to meet a need for a cross-platform, integrated ImageSource for reading images across networks. The class is straightforward, modeled after ImageFileReader and uses CURL, a well known library for URL transfer.

1 Basic Usage

ImageNetworkReader is a templated class. The first step to usage is including the header, and defining a typedef for ease of use.

```
#include "itkImageNetworkReader.h"

...

typedef itk::Image<short, 3> ImageType;
typedef itk::ImageNetworkReader<ImageType> ReaderType;

ReaderType::Pointer reader = ReaderType::New();
```

The reader class requires a URI to be set. This URI may have any of the common forms, the CURL documentation provides details on the supported protocols (<http://curl.haxx.se/docs/>).

```
reader->SetURI ( "http://www.na-mic.org/Wiki/images/f/f6/Test.nhdr" );
```

The class functions like any other `ImageSource` and may be included in pipelines.

2 Advanced Usage

In addition to the same methods available to `ImageFileReader`, `ImageNetworkReader` has additional options available. Images downloaded from the URI are saved in a temporary directory. By default, the temporary directory is the current working directory, but this may be changed through a call to `SetTemporaryDirectory`. All files will be downloaded into this directory before reading. Files will be named according to the tail of the URI. If an `ImageIO` object is set, the downloaded file will be read using that IO class, otherwise the standard `ImageIO` factory methods will be used.

To properly allocate the output of the filter, `ImageNetworkReader` downloads the image header, under the assumption that most image formats have header information stored at the beginning of the file. The size of this download is controlled by `SetHeaderFetchSize`. This method takes an unsigned long int specifying the number of bytes to read from the file in the `GenerateOutputInformation` method. NB: CURL does not allow termination of the download on exact byte boundaries. The code is guaranteed to transfer at least `HeaderFetchSize` bytes, but may read more. When reading a full file, the header is read twice, once during `GenerateOutputInformation` and again during `GenerateData`. This is a limitation of the class. However, the default `HeaderFetchSize` is 4K, and should not incur much overhead.

Since several file formats have separate header and data files, provisions have been made to specify a data file extension. When set, this extension is used to replace the extension of the `SetURI` call and download the data file as well.

3 Software Requirements

You need to have the following software installed to compile and use `ImageNetworkReader`:

- Insight Toolkit 3.0 or greater

- CMake 2.4

Note that other versions of the Insight Toolkit are also available in the testing framework of the Insight Journal. Please refer to the following page for details

<http://www.insightsoftwareconsortium.org/wiki/index.php/IJ-Testing-Environment>

4 CURL License

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2008, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.