GIFTI Surface Data Format Version 1.0

14 January 2011

John Harwell,[a] Hester Bremen,[b] Olivier Coulon,[c] Donna Dierker,[a] Richard C. Reynolds,[d] Nick Schmansky,[f] Claudio Silva,[e] Kevin Teich,[f] David C. Van Essen,[a] Simon K. Warfield,[g] and Ziad S. Saad[d]

[a] Department of Anatomy and Neurobiology, Washington University School of Medicine, St. Louis, MO, USA
[b] Brain Innovation B.V., The Netherlands
[c] Laboratoire LSIS, UMR 6168, CNRS, Marseille, France
[d] Scientific and Statistical Computing Core, National Institute of Mental Health, NIH, Department of Health and Human Services, Bethesda, Maryland, USA
[e] Scientific Computing and Imaging Institute and School of Computing, University of Utah, Salt Lake City, UT, USA
[f] Department of Radiology, Massachusetts General Hospital, Charlestown, MA, USA
[g] Computational Radiology Laboratory, Department of Radiology, Children's Hospital Boston, Boston, MA, USA

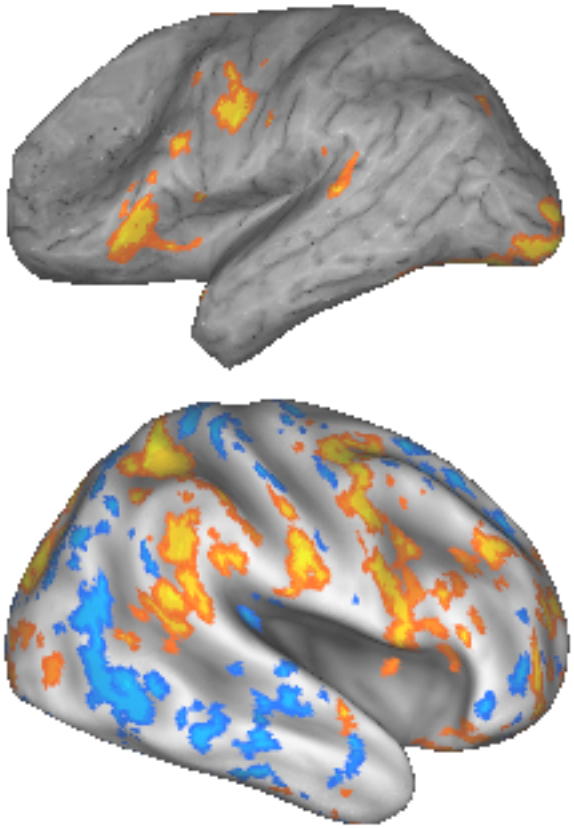


Table of Contents

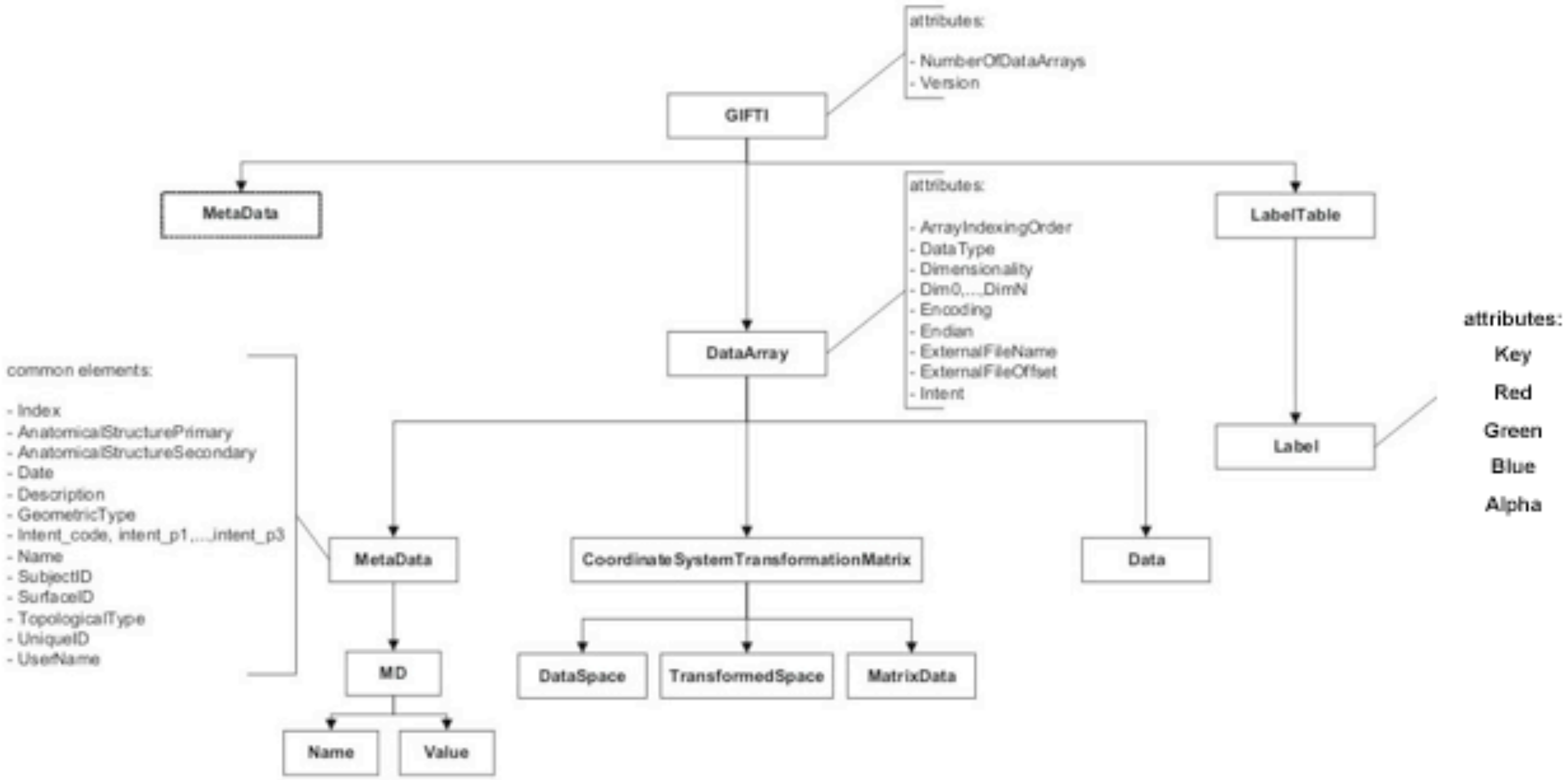
Copyright (c) 2011 John Harwell, Hester Bremen, Olivier Coulon, Donna Dierker, Rick Reynolds, Nick Schmansky, Claudio Silva, Kevin Teich, David C. Van Essen, Simon Warfield, and Ziad Saad

1.0 Introduction

This document describes a simple and flexible XML-compliant file for storing a variety of surface-based data. The effectiveness of the format is shown by its ability to store a variety of data types including surfaces, functional measurements, shape measurements, and labels. Its ability to store binary data or compressed binary data results in files that are smaller and noticeably faster to read than pure text XML files.

2.0 XML File Tags

The GIFTI file format consists of 13 XML tags. Following the tag hierarchy are descriptions of each tag along with the valid parent and child tags. Each child tag is followed by a symbol or number in parenthesis indicating the number of children allowed in the parent tag. A "*" symbol indicates a minimum of zero with no maximum. A "+" symbol indicates a minimum of one with no maximum. A "?" indicates a minimum of zero and a maximum of one.


attributes:
- NumberOfDataArrays
- Version
GIFTI
MetaData
LabelTable
attributes:
- ArrayIndexingOrder
- DataType
- Dimensionality
- Dim0,...,DimN
- Encoding
- Endian
- ExternalFileName
- ExternalFileOffset
- Intent
DataArray
attributes:
Key
Red
Green
Blue
Alpha
Label
common elements:
- Index
- AnatomicalStructurePrimary
- AnatomicalStructureSecondary
- Date
- Description
- GeometricType
- Intent_code, intent_p1,...,intent_p3
- Name
- SubjectID
- SurfaceID
- TopologicalType
- UniqueID
- UserName
MetaData
CoordinateSystemTransformationMatrix
Data
MD
DataSpace
TransformedSpace
MatrixData
Name
Value

2.1. CoordinateSystemTransformMatrix

2.1.1 Description – For a DataArray with an Intent NIFTI_INTENT_POINTSET, this element describes the stereotaxic space of the data before and after the application of a transformation matrix. The most common stereotaxic space is the Talairach Space that places the origin at the anterior commissure and the negative X, Y, and Z axes correspond to left, posterior, and inferior respectively.

At least one CoordinateSystemTransformMatrix is required in a DataArray with an intent of NIFTI_INTENT_POINTSET. Multiple CoordinateSystemTransformMatrix elements may be used to describe the transformation to multiple spaces.

2.1.2 Children – DataSpace (1), MatrixData (1), TransformedSpace (1)

2.1.3 Parent – DataArray

2.2 Data

2.2.1 Description – Contains the data for a DataArray. The contents may be ASCII text, binary data encoded as base64 text, or zipped binary data encoded as base64 text. This element's contents are empty if the data is stored in an external file.

2.2.2 Parent – DataArray

2.3 DataArray

2.3.1 Description – This element contains the numeric data and its related metadata. The CoordinateSystemTransformMatrix child is only used when the DataArray's Intent is NIFTI_INTENT_POINTSET. FileName and FileOffset are required if the data is stored in an external file.

2.3.2 Children – Data (1), MetaData (?), CoordinateSystemTransformMatrix (+).

2.3.3 Parent – GIFTI

2.3.4 Attributes

2.3.4.1 ArrayIndexingOrder – This required attribute describes order of multi-dimensional data in a Data Array.

Value	Description
RowMajorOrder	The highest index increments fastest.
ColumnMajorOrder	The lowest index increments fastest such as in the FORTRAN and MATLAB languages.

Values in Data Element	1 2 3 4
Row Major Order	1 2 3 4
Column Major Order	1 3 2 4

2.3.4.2 DataType – This required attribute describes the numeric type of the data contained in a Data Array and are limited to the types displayed in the table.

Value	Description
NIFTI_TYPE_UINT8	Unsigned, 8-bit bytes.
NIFTI_TYPE_INT32	Signed, 32-bit integers.
NIFTI_TYPE_FLOAT32	32-bit single precision floating point.

2.3.4.3 Dimensionality – This required attribute contains the number of dimensions of the data contained in this data array. This is a single integer value.

2.3.4.4 Dim0, **Dim1**, …, **DimN** – The number of elements in the "N'th" dimension of the data. Dim0 is required. The last dimension shall never be one (The only exception is if the array contains one data value in which case Dim0 will be one). If the data is node based, the data should be stored so that Dim0 is the number of nodes.

2.3.4.5 Encoding – This required attribute describes the encoding of the data in the Data Array's Data element. Also see the sections of this document entitled Data Compression and Data Encoding.

Value	Description
ASCII	The Data element contains ASCII text with each value separated by whitespace.
Base64Binary	The Data element contains a string of consecutive ASCII characters that are a Base64 text representation of the binary data.
GZipBase64Binary	The Data element contains a string of consecutive ASCII characters that are a Base64 text representation of the gzipped binary data.
ExternalFileBinary	The data is located in an external file. The name of the file is contained in the ExternalFileName attribute and the offset of the data in the external file is contained in the ExternalFileOffset attribute.

2.3.4.6 Endian – When the Data Array contains Base64Binary or GzipBase64Binary data, this required attribute indicates the byte order of the binary data.

Value	Description
BigEndian	The most significant byte of a multi-byte value is stored at the beginning of the multi-bytes value's memory.
LittleEndian	The least significant byte of a multi-byte value is stored at the beginning of the mult-byte value's memory.

2.3.4.7 ExternalFileName – This optional attribute must be present if the data is stored in an external file. This attribute's value contains the name of the external file.

2.3.4.8 ExternalFileOffset – This optional attribute must be present if the data is stored in an external file. This attribute's value contains the offset of the data in the external file.

2.3.4.9 Intent – This required attribute lists the intent (category) of the data stored within the Data Array. These labels are those defined by the NIFTI volume file. When the data contains indices, the indices range from zero to the number of elements minus one as in programming languages such as C, C++, or Java.

Value	Description
NIFTI_INTENT_GENMATRIX	The data consists of tensor data.
Any value starting with "NIFTI_INTENT"	The data consists of statistical (functional) measurements.
NIFTI_INTENT_LABEL	The data consists of indices into the file's LabelTable.
NIFTI_INTENT_NODE_INDEX	The data consists of a list of node numbers.
NIFTI_INTENT_POINTSET	The data consists of three-dimensional coordinate data.
NIFTI_INTENT_RGB_VECTOR NIFTI_INTENT_RGBA_VECTOR	The data consists of red, green, blue, alpha color values.
NIFTI_INTENT_SHAPE	The data consists of shape measurements.
NIFTI_INTENT_TIME_SERIES	The data consists of time-series data.
NIFTI_INTENT_TRIANGLE	The data consists of consecutive triplets that are indices into a

	NIFTI_INTENT_POINTSET Array forming a triangle. The three nodes are listed in counter-clockwise winding (right-hand rule).
NIFTI_INTENT_NONE	Data intent not specified.
NIFTI_INTENT_VECTOR	The data consists of three-dimensional vectors.

2.4 DataSpace

2.4.1 Description – Contains the stereotaxic space of a DataArray's data prior to application of the transformation matrix. The stereotaxic space should be one of:
NIFTI_XFORM_UNKNOWN
NIFTI_XFORM_SCANNER_ANAT
NIFTI_XFORM_ALIGNED_ANAT
NIFTI_XFORM_TALAIRACH
NIFTI_XFORM_MNI_152

2.4.2 Parent – CoordinateSystemTransformMatrix

2.5 GIFTI

2.5.1 Description – This is the root element of a GIFTI file.

2.5.2 Children – In this order: MetaData (?), LabelTable (?), DataArray (+).

2.5.3 Attributes

2.5.3.1 NumberOfDataArrays – This required attribute indicates the number of data arrays contained within the GIFTI file.

2.5.3.2 Version – This required attribute indicates the version of the GIFTI file. For this specification, the version is “1.0”.

2.5.3.3 xmlns:xsi – An optional attribute that indicates the XML Schema namespace.

2.5.3.4 xsi:noNamespaceSchemaLocation – An optional attribute that is the URL of an XML Schema for validation of the GIFTI XML file.

Note that when using an XML Schema to validate the GIFTI XML file, both xmlns:xsi and xsi:noNamespaceSchemaLocation are required. Otherwise, they should not be included as GIFTI element attributes.

2.6 Label

2.6.1 Description – Contains the name of a label and attributes indicating the key of the label and its color components.

When labels are created for nodes, there may be nodes that are “unassigned”. In this case, these “unassigned” nodes should correspond to a label whose alpha attribute is zero. An alpha attribute of zero indicates complete transparency.

Nodes that do not match a valid key in the LabelTable should not receive any coloring.

2.6.2 Parent – LabelTable

2.6.3 Attributes

2.6.3.1 Key – This required attribute contains a non-negative integer value. If a DataArray’s Intent is NIFTI_INTENT_LABEL and a value in the DataArray is ‘X’, its corresponding label is the label with the Key attribute containing the value ‘X’. In early versions of the GIFTI file format, the attribute Index was used instead of Key. If an Index attribute is encountered, it should be processed like the Key attribute.

2.6.3.2 Red – This optional, but recommended, attribute contains a floating-point value ranging from zero to one. It contains the red component of the label's color.

2.6.3.3 Green – This optional, but recommended, attribute contains a floating-point value ranging from zero to one. It contains the green component of the label's color.

2.6.3.4 Blue – This optional, but recommended, attribute contains a floating-point value ranging from zero to one. It contains the blue component of the label's color.

2.6.3.5 Alpha – This optional, but recommended, attribute contains a floating-point value ranging from zero to one. It contains the alpha component of the label's color.

2.7 LabelTable

2.7.1 Description – The label table is used by DataArrays whose values are an key into the LabelTable's labels. A file should contain at most one LabelTable and it must be located in the file prior to any DataArray elements.

2.7.2 Children – Label (*)

2.7.3 Parent – GIFTI

2.8 MatrixData

2.8.1 Description – Contains 16 text values representing a double-precision 4x4 transformation matrix. The values are listed as a one-dimensional array in row-major order which is used by most programming languages with the exception of FORTRAN and MATLAB. Note that OpenGL uses column-major order for its matrices.

For example, the matrix below would appear as "m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13 m14 m15 m16" is the MatrixData element.

m1	m2	m3	m4
m5	m6	m7	m8
m9	m10	m11	m12
m13	m14	m15	m16

2.8.2 Parent – CoordinateSystemTransformMatrix

2.9 MetaData

2.9.1 Description – When MetaData is a child of GIFTI, it contains metadata regarding the file. When MetaData is a child of DataArray, it contains metadata regarding the data array.

2.9.2 Children – MD (*)

2.9.3 Parent – GIFTI, DataArray

2.10 MD

2.10.1 Description – In a GIFTI file, metadata consists of pairs of Name and Value elements where Name is a descriptive name of the metadata item and Value contains the metadata. Metadata elements with names not recognized by the application software should be preserved or “passed through” if the file is updated.

2.10.2 Children – Name (1) and Value (1).

2.10.3 Parent – MetaData

2.11 Name

2.11.1 Description – Contains a descriptive name of a metadata item. Examples are Name, Description, Date, etc. Since the metadata name is often entered by the user, use of the CDATA directive is recommended to avoid problems caused by users entering a character such as ‘<’ that would otherwise confuse the XML parser.

2.11.2 Parent – MD

2.12 TransformedSpace

12.1 Description – Contains the stereotaxic space of a DataArray's data after application of the transformation matrix. See the DataSpace element for a list of stereotaxic spaces.

12.2 Parent – CoordinateSystemTransformMatrix

2.13 Value

2.13.1 Description – Contains the value of a metadata item. Since the metadata value is often entered by the user, use of the CDATA directive is recommended to avoid problems caused by users entering a character such as '<' that would otherwise confuse the XML parser.

2.13.2 Parent – MD

3.0 Standard MetaData

In a GIFTI data file, metadata consists of pairs of name and value elements. Developers are free to add any metadata they desire using the metadata elements of a GIFTI data file. To better facilitate the exchange of data, the committee has created several standard metadata elements that are listed below.

Metadata elements not recognized by an application will, in most cases, be discarded. This reason is that the application's operations may invalidate the unrecognized metadata.

3.1 AnatomicalStructurePrimary – Included in DataArray metadata for Intent NIFTI_INTENT_POINTSET to describe the structure that the point data geometrically models.

Values	Description
CortexLeft	The coordinates model the left cerebral cortex.
CortexRight	The coordinates model the right cerebral cortex.
CortexRightAndLeft	The coordinates model the right and left cortices.
Cerebellum	The coordinates model the cerebellum.
Head	The coordinates model the head.
HippocampusLeft	The coordinates model the left hippocampus.
HippocampusRight	The coordinates model the right hippocampus.

3.2 AnatomicalStructureSecondary – Included in DataArray metadata for

Intent NIFTI_INTENT_POINTSET to further describe the structure which the point data models.

Values	Description
GrayWhite	The white/gray boundary.
Pial	The pial (gray/CSF) boundary.
MidThickness	Middle layer of cortex (layer 4).

3.3 Date – Included in the GIFTI element's metadata and indicates the date, and possibly time, that the file was written.

3.4 Description – Included in GIFTI or Data Array metadata that provides a text description of the file or Data Array contents.

3.5 GeometricType - Included in NIFTI_INTENT_POINTSET DataArray metadata to describe the geometry of the point data.

Values	Description
Reconstruction	The surface as it appears immediately after reconstruction with a "blocky" appearance.
Anatomical	Represents an anatomical structure such as the white/gray boundary.
Inflated	A surface that has been inflated.
VeryInflated	A surface that has been highly inflated.
Spherical	A spherical surface.
SemiSpherical	A sphere with one-half flattened.
Ellipsoid	An ellipsoidal surface.
Flat	A flattened surface.
Hull	A hull surface (wrapping around a surface, ie: cortex with sulci filled.

3.6 Intent_code, intent_p1, intent_p2, intent_p3 – Included in Functional DataArray metadata and contains a NIFTI intent code and parameters describing the type of statistical test.

3.7 Name – Included in DataArray metadata and is used to provide a short name that often appears in the user-interface for selection by the user (similar to AFNI's BRICK_LABS).

3.8 SubjectID – Identifies a subject and is used to keep track of multiple surfaces for a subject.

3.9 SurfaceID – A unique string that identifies a surface.

3.10 TimeStep – Included in the file metadata of a data file that contains NIFTI_INTENT_TIME_SERIES DataArrays. TimeStep provides TR (repetition time). In a NIFTI volume file, this value is the "slice_duration" parameter.

3.11 TopologicalType – Included in NIFTI_INTENT_TRIANGLE DataArray metadata to describe the surface's topology.

Values	Description
Closed	A closed surface.
Open	Typically medial wall removed.
Cut	Cuts have been made for flattening

3.12 UniqueID – Included in a DataArray's metadata and provides a unique identifier for the DataArray. This value is best generated by a Universally Unique Identifier library function such as "uuid_generate()".

3.13 UserName – Included in GIFTI metadata and provides the name of the user that wrote the file.

4.0 Data Encoding

XML files by default use the UTF-8 character set (UTF-8 is a superset of the ASCII character set).. Several encodings for the Data's content are available including two that encode binary data as character data. The first encoding, ASCII, simply stores the numeric data in its text representation with each element separated by white space (spaces, line feeds, and tabs). The second encoding, Base64Binary (http://www.ietf.org/rfc/rfc3548.txt or http://email.about.com/cs/standards/a/base64_encoding.htm), encodes binary data into a sequence of ASCII characters. Base64 is used by email for the transmission of binary data. The third encoding,Base64GzipBinary, compresses the binary data using ZLIB and then converts the data to a Base64 representation.

The advantage of ASCII encoding is that it is human readable. The disadvantage is that the ASCII representation of numeric data may become quite large, particularly if high precision is needed for floating point data. In addition, converting ASCII data back to its binary representation is time consuming.

The advantage of Base64 encoding of binary data is that there is no loss of precision, its text representation is only 4/3 larger than its binary representation, and its conversion back to binary representation is faster than converting ASCII back to a binary representation. The disadvantage is that Base64 encoding is not human readable.

The GzipBase64Binary encoding possesses the same advantages as Base64 compared to ASCII. Its advantage over Base64 is reduced file size.

5.0 Data Compression

If the Data element's content is to be compressed, it is compressed using ZLIB (www.zlib.net). This is the algorithm used by the Unix gzip/gunzip utilities. In addition to the ZLIB C functions for compressing and uncompressing data in memory, VTK provides a class (vtkZLibDataCompressor) for C++ users and for Java users the java.util.zip class.

Our design does not compress the entire file. Instead, only the Data element contents are compressed. Compressing only the Data element contents and not the entire file allows quick examination of the file's contents. For example, the file's metadata can be extracted without having to perform any decompression. Likewise, the DataArray element names may be read and then a subset of the DataArrays read and decompressed.

Any data that is compressed must be Base64 encoded since XML files will not accept non-printable characters.

6.0 Sparse Data Storage

For storing a subset of node data, a DataArray with intent NIFTI_INTENT_NODE_INDEX may be used. In this case, the data array contains a list of node numbers and it should be the first data array in the file. The remaining data arrays in the file that contain data assigned to nodes must contain the same number of elements as the NIFTI_INTENT_NODE_INDEX array.

For example, suppose a NIFTI_INTENT_NODE_INDEX data array contains the elements 7, 13, 38, and 44. A NIFTI_INTENT_TTEST data array would contain only data for the node numbers listed in the NIFTI_INTENT_NODE_INDEX data array.

7.0 External Data File

In most cases, data will be stored within the XML file. However, some users, especially those with large files, may want to store the data in an external file. For external data storage, the Encoding attribute of the DataArray element will have a value of "ExternalFileBinary" indicating that the data is located in an external file and in binary format. The name of the external file and the offset of the data are contained in the DataArray attributes named ExternalFileName and ExternalFileOffset respectively. It is a requirement that the external data file is located in the same directory as the XML file to facilitate the transfer of the file to other users. Allowing each DataArray to specify the external file name allows the use of multiple external files that may be useful with large data sets and alleviate any single data file size limitations.

The DataArray's Endian attribute specifies the byte order of the data in the external file. The data in the external file is unencoded binary datay.

Due to XML limitations, the ExternalFileName may not contain '<' or '&' characters.

While external data storage is proposed, we recommend that it not be used unless it is absolutely necessary. There have been many times users have provided us with volume data but forget to provide both the header and the data file.

8.0 GIFTI XML Validation

XML files may be validated using either a DTD (Document Type Definition) or an XML Schema.

8.1 DTD –Document Type Definition

A DTD (Document Type Definition) describes the allowable elements and attributes in an XML file and may be used to assist in validating the contents of an XML file. Each ELEMENT entry in the DTD lists a valid element name and the children allowed for the element. An optional character may follow the name of each child element, "?", "+", or "*". A "?" character indicates there are zero or one instances of the child. A "+" character indicates that one or more of the instances of the child are required. A "*" indicates that there may be zero or more instances of the child. For example, the MetaData child is optional and there may at most one MetaData child of the GIFTI element so MetaData is followed by a "?".

As an example, in the DTD below, the GIFTI element has an optional child named MetaData, an optional child named LabelTable, and one or more children named DataArray. The children must appear in the specified order. LabelTable before DataArray is allowed but having LabelTable after DataArray would be disallowed by an XML validiator.

Each ATTLIST element in the DTD specifies the allowable attributes for an element. For example, the GIFTI element must have a Version attribute and the Label element must have an Key attribute.

8.1.1 Validate a GIFTI XML File With a DTD

On Linux and Macintosh systems, the program xmllint may be used to validate a GIFTI XML file with a DTD. The DTD must be declared within the GIFIT XML file. The declaration "<!DOCTYPE GIFTI SYSTEM "http://www.nitrc.org/frs/download.php/1594/gifti.dtd">" is placed between the XML element and the GIFTI element.

To validate the GIFTI XML file with a DTD, run the command "xmllint –noout –valid <gifti-file-name>". If the validation process finds errors, they are printed, otherwise, if the file successfully validates, nothing is printed.

8.1.2 The GIFTI DTD

```
<!ELEMENT GIFTI (MetaData?,
                    LabelTable?,
                    DataArray+)>
   <!ELEMENT LabelTable (Label*)>
   <!ELEMENT DataArray (MetaData?,
                        CoordinateSystemTransformMatrix*,
                        Data)>
   <!ELEMENT Data (#PCDATA)>
   <!ELEMENT Label (#PCDATA)>
   <!ELEMENT Value (#PCDATA)>
   <!ELEMENT Name (#PCDATA)>
   <!ELEMENT MD (Name, Value)>
   <!ELEMENT MetaData (MD*)>
   <!ELEMENT CoordinateSystemTransformMatrix (DataSpace,

TransformedSpace,
                                              MatrixData)>
   <!ELEMENT DataSpace (#PCDATA)>
   <!ELEMENT TransformedSpace (#PCDATA)>
   <!ELEMENT MatrixData (#PCDATA)>
   <!ATTLIST GIFTI
      xmlns:xsi CDATA #IMPLIED
      xsi:noNamespaceSchemaLocation CDATA #IMPLIED
      NumberOfDataArrays  NMTOKEN #REQUIRED
      Version NMTOKEN #REQUIRED>
   <!ATTLIST Label
      Key     NMTOKEN #REQUIRED
      Red     NMTOKEN #IMPLIED
      Green   NMTOKEN #IMPLIED
      Blue    NMTOKEN #IMPLIED
      Alpha   NMTOKEN #IMPLIED>
   <!ATTLIST DataArray
      ArrayIndexingOrder  ( RowMajorOrder |
                            ColumnMajorOrder ) #REQUIRED
      DataType     (NIFTI_TYPE_UINT8 |
                    NIFTI_TYPE_INT32 |
                    NIFTI_TYPE_FLOAT32)  #REQUIRED
      Dimensionality   NMTOKEN #REQUIRED
      Dim0         NMTOKEN #REQUIRED
      Dim1         NMTOKEN #IMPLIED
      Dim2         NMTOKEN #IMPLIED
```

```
Dim3         NMTOKEN #IMPLIED
Dim4         NMTOKEN #IMPLIED
Dim5         NMTOKEN #IMPLIED
Encoding     (ASCII |
              Base64Binary |
              GZipBase64Binary |
              ExternalFileBinary)  #REQUIRED
Endian       (BigEndian |
              LittleEndian)  #REQUIRED
ExternalFileName    CDATA  #IMPLIED
ExternalFileOffset  CDATA  #IMPLIED
Intent (NIFTI_INTENT_NONE          |
      NIFTI_INTENT_CORREL          |
      NIFTI_INTENT_TTEST           |
      NIFTI_INTENT_FTEST           |
      NIFTI_INTENT_ZSCORE          |
      NIFTI_INTENT_CHISQ           |
      NIFTI_INTENT_BETA            |
      NIFTI_INTENT_BINOM           |
      NIFTI_INTENT_GAMMA           |
      NIFTI_INTENT_POISSON         |
      NIFTI_INTENT_NORMAL          |
      NIFTI_INTENT_FTEST_NONC      |
      NIFTI_INTENT_CHISQ_NONC      |
      NIFTI_INTENT_LOGISTIC        |
      NIFTI_INTENT_LAPLACE         |
      NIFTI_INTENT_UNIFORM         |
      NIFTI_INTENT_TTEST_NONC      |
      NIFTI_INTENT_WEIBULL         |
      NIFTI_INTENT_CHI             |
      NIFTI_INTENT_INVGAUSS        |
      NIFTI_INTENT_EXTVAL          |
      NIFTI_INTENT_PVAL            |
      NIFTI_INTENT_LOGPVAL         |
      NIFTI_INTENT_LOG10PVAL       |
      NIFTI_INTENT_ESTIMATE        |
      NIFTI_INTENT_LABEL           |
      NIFTI_INTENT_NEURONAME       |
      NIFTI_INTENT_GENMATRIX       |
      NIFTI_INTENT_SYMMATRIX       |
      NIFTI_INTENT_DISPVECT        |
      NIFTI_INTENT_VECTOR          |
      NIFTI_INTENT_POINTSET        |
      NIFTI_INTENT_TRIANGLE        |
      NIFTI_INTENT_QUATERNION      |
      NIFTI_INTENT_DIMLESS         |
      NIFTI_INTENT_TIME_SERIES     |
      NIFTI_INTENT_RGB_VECTOR      |
         NIFTI_INTENT_RGBA_VECTOR     |
         NIFTI_INTENT_NODE_INDEX      |
         NIFTI_INTENT_SHAPE) #REQUIRED
>
```

8.2 GIFTI XML Schema

Another way to validate a GIFTI XML file is to use an XML Schema. An XML Schema permits additional error checking that a DTD is not able to perform. For example, the GIFTI DTD can indicate that a Label element has an optional attribute Red, but nothing more. The GIFTI XML Schema not only indicates that a Label element has an optional attribute Red, but also that Red's value is a floating point-number ranging from 0.0 to 1.0.

The XML Schema is identified by adding two additional attributes to the GIFTI element. These attributes are xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" and xsi:noNamespaceSchemaLocation="http://brainvis.wustl.edu/caret6/xml_schemas/GIFTI_Caret.xsd".

8.2.1 Validate a GIFTI XML File with an XML Schema

One way to validate the GIFTI file with the GIFTI XML Schema is to use the website http://tools.decisionsoft.com/schemaValidate. On Linux and Macintosh systems, the program xmllint may be used. To validate using xmllint, run the command "xmllint -noout -schema http://www.nitrc.org/frs/download.php/1303/GIFTI_Caret.xsd <gifti-file-name>".

8.2.2 The GIFTI XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:simpleType name="TypeArrayIndexingOrder">
        <xs:restriction base="xs:string">
            <xs:enumeration value="RowMajorOrder"/>
            <xs:enumeration value="ColumnMajorOrder"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TypeColorComponent">
        <xs:restriction base="xs:float">
            <xs:minInclusive value="0.0"/>
            <xs:maxInclusive value="1.0"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TypeNiftiDataType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="NIFTI_TYPE_UINT8"/>
            <xs:enumeration value="NIFTI_TYPE_INT32"/>
            <xs:enumeration value="NIFTI_TYPE_FLOAT32"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TypeEncoding">
        <xs:restriction base="xs:string">
            <xs:enumeration value="ASCII"/>
            <xs:enumeration value="Base64Binary"/>
            <xs:enumeration value="GZipBase64Binary"/>
            <xs:enumeration value="ExternalFileBinary"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TypeEndian">
        <xs:restriction base="xs:string">
            <xs:enumeration value="BigEndian"/>
            <xs:enumeration value="LittleEndian"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TypeMatrixSpace">
        <xs:restriction base="xs:string">
            <xs:enumeration
value="NIFTI_XFORM_ALIGNED_ANAT"/>
            <xs:enumeration value="NIFTI_XFORM_MNI_152"/>
            <xs:enumeration
value="NIFTI_XFORM_SCANNER_ANAT"/>
            <xs:enumeration value="NIFTI_XFORM_TALAIRACH"/>
            <xs:enumeration value="NIFTI_XFORM_UNKNOWN"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TypeNiftiIntent">
        <xs:restriction base="xs:string">
            <xs:enumeration value="NIFTI_INTENT_BETA"/>
            <xs:enumeration value="NIFTI_INTENT_BINOM"/>
            <xs:enumeration value="NIFTI_INTENT_CHI"/>
            <xs:enumeration value="NIFTI_INTENT_CHISQ"/>
            <xs:enumeration
value="NIFTI_INTENT_CHISQ_NONC"/>
            <xs:enumeration value="NIFTI_INTENT_CORREL"/>
            <xs:enumeration value="NIFTI_INTENT_DIMLESS"/>
            <xs:enumeration value="NIFTI_INTENT_DISPVECT"/>
            <xs:enumeration value="NIFTI_INTENT_ESTIMATE"/>
            <xs:enumeration value="NIFTI_INTENT_EXTVAL"/>
            <xs:enumeration value="NIFTI_INTENT_FTEST"/>
            <xs:enumeration
value="NIFTI_INTENT_FTEST_NONC"/>
            <xs:enumeration value="NIFTI_INTENT_GAMMA"/>
            <xs:enumeration
value="NIFTI_INTENT_GENMATRIX"/>
            <xs:enumeration value="NIFTI_INTENT_INVGAUSS"/>
            <xs:enumeration value="NIFTI_INTENT_LABEL"/>
            <xs:enumeration value="NIFTI_INTENT_LAPLACE"/>
            <xs:enumeration
value="NIFTI_INTENT_LOG10PVAL"/>
            <xs:enumeration value="NIFTI_INTENT_LOGISTIC"/>
            <xs:enumeration value="NIFTI_INTENT_LOGPVAL"/>
            <xs:enumeration
value="NIFTI_INTENT_NEURONAME"/>
            <xs:enumeration value="NIFTI_INTENT_NONE"/>
            <xs:enumeration
value="NIFTI_INTENT_NONE_INDEX"/>
            <xs:enumeration value="NIFTI_INTENT_NORMAL"/>
            <xs:enumeration value="NIFTI_INTENT_POINTSET"/>
            <xs:enumeration value="NIFTI_INTENT_POISSON"/>
            <xs:enumeration value="NIFTI_INTENT_PVAL"/>
            <xs:enumeration
value="NIFTI_INTENT_QUATERNION"/>
            <xs:enumeration
value="NIFTI_INTENT_RGBA_VECTOR"/>
            <xs:enumeration
value="NIFTI_INTENT_RGB_VECTOR"/>
            <xs:enumeration value="NIFTI_INTENT_SHAPE"/>
            <xs:enumeration
value="NIFTI_INTENT_SYMMATRIX"/>
            <xs:enumeration
value="NIFTI_INTENT_TIME_SERIES"/>
            <xs:enumeration value="NIFTI_INTENT_TRIANGLE"/>
            <xs:enumeration value="NIFTI_INTENT_TTEST"/>
            <xs:enumeration
value="NIFTI_INTENT_TTEST_NONC"/>
            <xs:enumeration value="NIFTI_INTENT_UNIFORM"/>
            <xs:enumeration value="NIFTI_INTENT_VECTOR"/>
            <xs:enumeration value="NIFTI_INTENT_WEIBULL"/>
            <xs:enumeration value="NIFTI_INTENT_ZSCORE"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="TypeVersion">
```

```
        <xs:restriction base="xs:float">
            <xs:minInclusive value="1.0"/>
            <xs:maxInclusive value="1.0"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:element name="Data"
                type="xs:string"/>
    <xs:element name="DataSpace"
                type="TypeMatrixSpace"/>
    <xs:element name="MatrixData"
                type="xs:string"/>
    <xs:element name="Name"
                type="xs:string"/>
    <xs:element name="TransformedSpace"
                type="TypeMatrixSpace"/>
    <xs:element name="Value"
                type="xs:string"/>

    <xs:element name="Label">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="Key"

type="xs:nonNegativeInteger"
                                  use="required"/>
                    <xs:attribute name="Red"
                                  type="TypeColorComponent"
                                  use="optional"/>
                    <xs:attribute name="Green"
                                  type="TypeColorComponent"
                                  use="optional"/>
                    <xs:attribute name="Blue"
                                  type="TypeColorComponent"
                                  use="optional"/>
                    <xs:attribute name="Alpha"
                                  type="TypeColorComponent"
                                  use="optional"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>

    <xs:element name="CoordinateSystemTransformMatrix">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="DataSpace"/>
                <xs:element ref="TransformedSpace"/>
                <xs:element ref="MatrixData"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="LabelTable">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Label"
                            minOccurs="0"
                            maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="MD">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="MetaData">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="MD"
                            minOccurs="0"
                            maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="GIFTI">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="MetaData"
                            minOccurs="0"
                            maxOccurs="1"/>
                <xs:element ref="LabelTable"
                            minOccurs="0"
                            maxOccurs="1"/>
                <xs:element ref="DataArray"
                            minOccurs="0"
                            maxOccurs="unbounded"/>
            </xs:sequence>
```

```
            <xs:attribute name="NumberOfDataArrays"
                          type="xs:nonNegativeInteger"
                          use="required"/>
            <xs:attribute name="Version"
                          type="TypeVersion"
                          use="required"/>
        </xs:complexType>
    </xs:element>

    <xs:element name="DataArray">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="MetaData"
                            minOccurs="0"
                            maxOccurs="1"/>
                <xs:element
ref="CoordinateSystemTransformMatrix"
                            minOccurs="0"
                            maxOccurs="unbounded"/>
                <xs:element ref="Data"
                            minOccurs="1"
                            maxOccurs="1"/>
            </xs:sequence>
            <xs:attribute name="Intent"
                          type="TypeNiftiIntent"
                          use="required"/>
            <xs:attribute name="DataType"
                          type="TypeNiftiDataType"
                          use="required"/>
            <xs:attribute name="ArrayIndexingOrder"
                          type="TypeArrayIndexingOrder"
                          use="required"/>
            <xs:attribute name="Dimensionality"
                          type="xs:positiveInteger"
                          use="required"/>
            <xs:attribute name="Dim0"
                          type="xs:positiveInteger"
                          use="required"/>
            <xs:attribute name="Dim1"
                          type="xs:positiveInteger"
                          use="optional"/>
            <xs:attribute name="Dim2"
                          type="xs:positiveInteger"
                          use="optional"/>
            <xs:attribute name="Dim3"
                          type="xs:positiveInteger"
                          use="optional"/>
            <xs:attribute name="Dim4"
```

```
                          type="xs:positiveInteger"
                          use="optional"/>
            <xs:attribute name="Dim5"
                          type="xs:positiveInteger"
                          use="optional"/>
            <xs:attribute name="Encoding"
                          type="TypeEncoding"
                          use="required"/>
            <xs:attribute name="Endian"
                          type="TypeEndian"
                          use="required"/>
            <xs:attribute name="ExternalFileName"
                          type="xs:string"
                          use="optional"/>
            <xs:attribute name="ExternalFileOffset"
                          type="xs:nonNegativeInteger"
                          use="optional"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

9.0 File Name Extensions

Initially, the extension “.gii” was suggested for all GIFTI files. However, it is desirable to have a unique extension for each intent type.

Intent	Extension
Generic GIFTI File	.gii
Coordinates	.coord.gii
Functional	.func.gii
Labels	.label.gii
RGB or RGBA	.rgba.gii
Shape	.shape.gii
Surface	.surf.gii
Tensors	.tensor.gii
Time Series	.time.gii
Topology	.topo.gii
Vector	.vector.gii

10.0 XML File Reading and Writing

There are two common APIs for reading XML files, DOM and SAX. DOM (Document Object Model) reads the XML file and creates a tree structure representing the XML file's contents. The tree structure is then traversed by the developer's software to obtain the desired content. SAX (Simple API for XML) is an event-based API that issues events to the developer's software as XML tags and content are encountered during file reading. The disadvantage of DOM is that as the size of the file grows, the memory required by the tree structure is even larger. SAX on the other hand, does not require the storage of the entire document in memory,

Since the GIFTI XML data files may be very large, we chose the SAX API for reading the XML files. Specifically, we used the QXmlDefaultHandler C++ class from Trolltech's QT Software. For writing XML files, QT only provides the DOM API. The simple format of our proposed XML files allows us to use text I/O to efficiently write the files.

11.0 GIFTI Website

The GIFTI website is be part of the Neuroimaging Informatics Tools and Resources Clearinghouse (http://www.nitrc.org/projects/gifti) website. This website provides the GIFTI specification document, example data files, software libraries for reading and writing GIFTI files, a mailing list, and a list of neuroimaging software packages that support the GIFTI data format.

12.0 Extending GIFTI

This document describes the initial version of the GIFTI Surface Data Format. The data array intents in this version of GIFTI are those categories of data commonly interchanged between the various neuroimaging software packages. Developers are encouraged to use the GIFTI file format for new data intent types that do not require the addition of new XML tags. It is not necessary for developers to get approval of the GIFTI committee to add a new type of data intent. However, the GIFTI mailing list should be used to announce the new data intent to assess interest in the new data intent becoming an official GIFTI data intent.

When naming a new data intent, a process similar to that used for OpenGL extensions (http://en.wikipedia.org/wiki/OpenGL#Extensions) should be followed. In the case of GIFTI, the name of the new data intent should begin with the name of the neuroimaging software package creating the new data intent. For example, if the developers of Caret wanted to store latitude/longitude data in a GIFTI data file, the data intent would initially be named "CARET_INTENT_LATITIUDE_LONGITUDE". If the GIFTI committee supported the addition of this new data intent, its name would change to "NIFTI_INTENT_LATITUDE_LONGITUDE".

13.0 Data File Layout

13.1 Coordinate File

The coordinate file contains one DataArray with Intent set to NIFTI_INTENT_POINTSET and DataType set to NIFTI_TYPE_FLOAT32. Dimensionality is two with the first dimension set to the number of nodes and the second dimension set to three. This DataArray contains the X-, Y-, and Z-Coordinate for each node.

The application of a **CoordinateSystemTransformMatrix**, places the coordinates into the system shown in the table below. All coordinates are in millimeters.

Coordinate Axis	Orientation
Negative X	Left
Positive X	Right
Negative Y	Posterior
Positive Y	Anterior
Negative Z	Inferior
Positive Z	Superior

13.2 Functional File

The functional file contains one or more DataArrays with Intent set to NIFTI_INTENT_NONE or one of the statistical intent values. Each DataArray has DataType set to NIFTI_TYPE_FLOAT32. Dimensionality is one with the first dimension set to the number of nodes.

13.3 Label File

The label file contains a LabelTable followed by one or more DataArrays with Intent set to NIFTI_INTENT_LABEL and DataType set to NIFTI_TYPE_INT32. Dimensionality is one with the first dimension set to the number of nodes. The value at each node is a key into the LabelTable. Nodes that are “unassigned” should be assigned a label whose color components’ alpha value is zero. An RGBA color with an alpha value of zero is fully transparent. Also note that the keys of the Labels are not necessarily sequential.

13.4 Shape File

The shape file contains one or more DataArrays with Intent set to NIFTI_INTENT_SHAPE and DataType set to NIFTI_TYPE_FLOAT32. Dimensionality is one with the first dimension set to the number of nodes.

13.5 Surface File

The surface file contains two DataArrays. The first DataArray is identical to the DataArray in a Coordinate File and the second DataArray is identical to the DataArray in a Topology File.

13.6 Time-Series File

The Time-Series file contains one or more DataArrays with Intent set to NIFTI_INTENT_TIME_SERIES and DataType set to NIFTI_TYPE_FLOAT32. Dimensionality is one with the first dimension set to the number of nodes.

13.7 Topology File

The Topology File contains one DataArray with Intent set to NIFTI_INTENT_TRIANGLE and DataType set to NIFTI_TYPE_INT32. Dimensionality is two with the first dimension set to the number of triangles and the second dimension set to three. Each triplet consists of node indices that, along with the corresponding Coordinate File, form the triangles of the surface model.

14.0 Example Data Files

The example data files are available at "http://www.nitrc.org/frs/?group_id=75". The release ***caret_gifti_files.zip*** in the package **caret gifti files** contains the data listed in the following table.

Subdirectories Created when Zip File is Unzipped	Encoding of Data Files
gifti_files/LEFT_HEM/ascii	ASCII
gifti_files/LEFT_HEM/base64	Base64 encoded binary
gifti_files/LEFT_HEM/gzip_base64	Gzipped base64 encoded binary
gifti_files/RIGHT_HEM/ascii	ASCII
gifti_files/RIGHT_HEM/base64	Base64 encoded binary
gifti_files/RIGHT_HEM/gzip_base64	Gzipped base64 encoded binary
Volumes used for creating surface and creating functional files.	

14.1 Descriptions of Right Hemisphere Example Data Files

The right hemisphere surfaces are part of the “colin” atlas supplied by the Van Essen laboratory.

Human.colin.Cerebral.R.FIDUCIAL.TLRC.711-2B.71723.surf.gii	A mid-thickness surface.
Human.colin.Cerebral.R.FLAT.CartSTD.71723.surf.gii	A flat surface.
Human.colin.Cerebral.R.VERY_INFLATED.71723.surf.gii	A Very Inflated surface.
Human.colin.R.FUNCTIONAL.71723.func.gii	A functional data file.
Human.colin.R.activations.label.gii	A label data file.
Human.colin.R.shape.71723.shape.gii	A shape data file.

14.2 Descriptions of Left Hemisphere Example Data Files

The example volume files (aoantonym_gen_t25.nii, hospgr.nii, and srV_ts_al12.nii) were provided by Simon Warfield. Ziad Saad created the surfaces. Caret software was used to create the GIFTI surface files and map the functional and time-series volumes to a surface. There is a lesion located in the lateral side of the frontal lobe. At this time, the generic GIFTI file name extension is used until a decision is made about the file type specific extensions.

gifti.case1.aoantonym_functional.L.func.gii	Functional data file containing the mapping of the functional volume aoantonym_gen_t25.nii to the surface.
gifti.case1.inflated.L.surf.gii	The inflated surface.

Gifti_case1.midthick.L.surf.gii	The “mid-thickness” (layer4 approximation) surface
gifti.case1.pial.L.surf.gii	The pial surface.
gifti.case1.shape.L.shape.gii	Shape file containing curvature generated using the pial surface.
gifti.case1.smoothwm.L.surf.gii	The white matter surface.
gifti.case1.sphere.L.surf.gii	The spherical surface.
gifti.case1.time_series.L.time.gii	Functional data file containing the mapping of the time-series volume srV_ts_al12.nii to the surface.

14.3 File Reading Time and Size Comparisons

The left hemisphere surface files contain 143479x3 coordinates, 143479x3 surface normals, and 28654x3 triangles. The functional file contains 143479 functional values (one per node). The time-series file contains 136 time-points for each of 143479 nodes. The file reading times are each an average of reading the file five times.

Timing for reading each of the three data encoding are shown in the table below. For files representing the data in an ASCII encoding, six digits of precision were used. These comparisons were run on a Dell PC with a 3.2 Ghz Pentium Dual Core processor and 2 GB RAM running Windows XP Professional.

Table 1: File Reading Time Comparisons in Seconds

File Type	File Name	Ascii	Base64	Gzip Base64
Surface	gifti.case1.pial.L.surf.gii	3.7	0.625	0.547
Func-tional	gifti.case1.aoantonym_functional.L.func.gii	0.302	0.052	0.01
Time-Series	gifti.case1.time_series.L.time.gii	60.52	6.9	2.61

Table 2: File Size Comparisons in MegaBytes

File Type	File Name	Ascii	Base64	Gzip Base64
Surface	gifti.case1.pial.L.surf.gii	21.2	8.76	5.4
Functional	gifti.case1.aoantonym_functional .L.func.gii	3.02	0.73	0.03
Time Series	gifti.case1.time_series.L.time.gii	476.1	98.6	19.2

14.4 Example Surface File (gifti.case1.pial.L.surf.gii ASCII version)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE GIFTI SYSTEM "http://www.nitrc.org/frs/download.php/1594/gifti.dtd">
 <GIFTI
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://www.nitrc.org/frs/download.php/1303/GIFTI_Caret.xsd"
    Version="1.0"
    NumberOfDataArrays="3">
  <MetaData>
    <MD>
      <Name><![CDATA[date]]></Name>
      <Value><![CDATA[Thu Nov 15 09:05:22 2007]]></Value>
    </MD>
  </MetaData>
  <LabelTable/>
  <DataArray Intent="NIFTI_INTENT_POINTSET"
          DataType="NIFTI_TYPE_FLOAT32"
          ArrayIndexingOrder="RowMajorOrder"
          Dimensionality="2"
          Dim0="143479"
          Dim1="3"
          Encoding="ASCII"
          Endian="LittleEndian"
          ExternalFileName=""
          ExternalFileOffset="">
    <MetaData>
      <MD>
        <Name><![CDATA[AnatomicalStructurePrimary]]></Name>
        <Value><![CDATA[CortexLeft]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[AnatomicalStructureSecondary]]></Name>
        <Value><![CDATA[Pial]]></Value>
      </MD>
      <MD>
```

```
        <Name><![CDATA[GeometricType]]></Name>
        <Value><![CDATA[Anatomical]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[UniqueID]]></Name>
        <Value><![CDATA[{565f3bd1-c0b1-49da-af3d-707cd6a8ccd0}]]></Value>
      </MD>
    </MetaData>
    <CoordinateSystemTransformMatrix>
      <DataSpace><![CDATA[NIFTI_XFORM_TALAIRACH]]></DataSpace>
      <TransformedSpace><![CDATA[NIFTI_XFORM_TALAIRACH]]></TransformedSpace>
      <MatrixData>
        1.000000 0.000000 0.000000 0.000000
        0.000000 1.000000 0.000000 0.000000
        0.000000 0.000000 1.000000 0.000000
        0.000000 0.000000 0.000000 1.000000
      </MatrixData>
    </CoordinateSystemTransformMatrix>
    <Data>
      -16.072010 -66.187515 21.266994
      -16.705893 -66.054337 21.232786
      -17.614349 -65.401642 21.071466
    </Data>
  </DataArray>
  <DataArray Intent="NIFTI_INTENT_TRIANGLE"
          DataType="NIFTI_TYPE_INT32"
          ArrayIndexingOrder="RowMajorOrder"
          Dimensionality="2"
          Dim0="286954"
          Dim1="3"
          Encoding="ASCII"
          Endian="LittleEndian"
          ExternalFileName=""
          ExternalFileOffset="">
    <MetaData>
      <MD>
        <Name><![CDATA[TopologicalType]]></Name>
        <Value><![CDATA[CLOSED]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[UniqueID]]></Name>
        <Value><![CDATA[{3160d119-081a-4b67-b5c8-bc0765783341}]]></Value>
      </MD>
    </MetaData>
    <Data>
      0 1 3
      4 3 1
      0 30 1
    </Data>
  </DataArray>
</GIFTI>
```

14.5 Example Functional File (gifti.case1.aoantonym_functional.L.func.gii ASCII version)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE GIFTI SYSTEM "http://www.nitrc.org/frs/download.php/1594/gifti.dtd">
 <GIFTI
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="http://www.nitrc.org/frs/download.php/1303/GIFTI_Caret.xsd"
      Version="1.0"
      NumberOfDataArrays="1">
  <MetaData>
    <MD>
      <Name><![CDATA[date]]></Name>
      <Value><![CDATA[Thu Nov 15 09:05:11 2007]]></Value>
    </MD>
  </MetaData>
  <LabelTable/>
  <DataArray Intent="NIFTI_INTENT_TTEST"
          DataType="NIFTI_TYPE_FLOAT32"
          ArrayIndexingOrder="RowMajorOrder"
          Dimensionality="2"
          Dim0="143479"
          Dim1="1"
          Encoding="ASCII"
          Endian="LittleEndian"
          ExternalFileName=""
          ExternalFileOffset="">
    <MetaData>
      <MD>
        <Name><![CDATA[Description]]></Name>
        <Value><![CDATA[CARET v5.511
Mapped to surface: fiducial.coord
Mapped from volume: aoantonym_gen_t25.nii
Subvolume: 0
Algorithm: Metric Enclosing Voxel
intent: T-statistic  DOF=8.00
intent_code: 3
intent_p1: 8.000
intent_p2: 0.000
intent_p3: 0.000
slice_duration: 2

Date Mapped: Thu Nov 15 08:55:32 2007
]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[Name]]></Name>
        <Value><![CDATA[aoantonym_gen_t25.nii[1]]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[UniqueID]]></Name>
        <Value><![CDATA[{929bf088-a838-4b29-877d-6fb5bdbfacbf}]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[intent]]></Name>
        <Value><![CDATA[T-statistic  DOF=8.00]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[intent_code]]></Name>
        <Value><![CDATA[3]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[intent_p1]]></Name>
        <Value><![CDATA[8.000]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[intent_p2]]></Name>
        <Value><![CDATA[0.000]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[intent_p3]]></Name>
        <Value><![CDATA[0.000]]></Value>
      </MD>
      <MD>
        <Name><![CDATA[slice_duration]]></Name>
        <Value><![CDATA[2.000000]]></Value>
      </MD>
    </MetaData>
    <Data>
      0.000000
      0.000000
      1.000000
    </Data>
  </DataArray>
</GIFTI>
```

14.6 Example Label File (Human.colin.R.activations.label.gii)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE GIFTI SYSTEM "http://www.nitrc.org/frs/download.php/1594/gifti.dtd">
 <GIFTI
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="http://www.nitrc.org/frs/download.php/1303/GIFTI_Caret.xsd"
     Version="1.0"
     NumberOfDataArrays="1">
  <MetaData>
    <MD>
     <Name><![CDATA[date]]></Name>
     <Value><![CDATA[3 Nov 2009 11:08:27]]></Value>
    </MD>
  </MetaData>
  <LabelTable>
    <Label Key="0" Red="0.667" Green="0.667" Blue="0.667"
                          Alpha="1.000"><![CDATA[???]]></Label>
```

```
      <Label Key="1" Red="1.000" Green="1.000" Blue="0.000"
                              Alpha="1.000"><![CDATA[Positive]]></Label>
      <Label Key="2" Red="0.008" Green="1.000" Blue="1.000"
                              Alpha="1.000"><![CDATA[Negative]]></Label>
   </LabelTable>
   <DataArray Intent="NIFTI_INTENT_LABEL"
     DataType="NIFTI_TYPE_INT32"
     ArrayIndexingOrder="RowMajorOrder"
     Dimensionality="1"
     Dim0="71723"
     Encoding="ASCII"
     Endian="LittleEndian"
     ExternalFileName=""
     ExternalFileOffset="">
    <MetaData>
      <MD>
         <Name><![CDATA[Description]]></Name>
         <Value><![CDATA[]]></Value>
      </MD>
      <MD>
         <Name><![CDATA[Name]]></Name>
         <Value><![CDATA[Activations]]></Value>
      </MD>
      <MD>
         <Name><![CDATA[UniqueID]]></Name>
         <Value><![CDATA[{06058ed8-b782-4431-99ac-b760a7410080}]]></Value>
      </MD>
      <MD>
         <Name><![CDATA[tagStudyMetaDataLinkSet]]></Name>
         <Value><![CDATA[]]></Value>
      </MD>
    </MetaData>
    <Data>
         0
         2
         0
         1
         1
         0
         2
         0
         1
         0
      </Data>
   </DataArray>
</GIFTI>
```