The Neurophysiology Data Translation Format (NDTF) Specification – V1.2

Document Ref:	The Neurophysiology Data Translation Format (NDTF) Specification – V1.2
Author:	Bojian Liang, Jennifer Simonotto, Alastair Knowles, Martyn Fletcher
Date:	September 19, 2008
Pages:	32
Abstract:	The purpose of the Neurophysiology Data Translation Format (NDTF) is to provide a means of sharing neurophysiology experimental data and derived data between services and tools developed within the CARMEN project (www.carmen.org.uk). This document specifies the NDTF. The specification supports the types of data that are currently used by members of the CARMEN consortium and provides a capability to support future data types. It is capable of accommodating external data file formats as well as metadata such as user defined experimental descriptions and the history (provenance) of derived data.

Contents

1 Status of this document **4**

2 Introduction **4**

3 The NDTF data type **6**

3.1 The NDTF internal data types 6

3.2 NDTF external data support 8

3.3 The Annotation data file format 8

3.3.1 The Annotation data file XML elements example 9

4 The NDTF Configuration file **12**

4.1 Structure of the configuration file 12

4.2 General description element 12

4.3 DataSet element 13

4.3.1 The *DataInfo* element 14

4.4 Data type definitions 16

4.4.1 ImageData 16

4.4.2 TimeSeriesData 17

4.4.3 SegmentData 18

4.4.4 NeuralEventData 20

4.4.5 ExperimentalEventData 21

4.4.6 GenericMatrix 21

4.4.7 UserDefinedData 22

4.5 History element 23

5 Summary **24**

6 Technical Information **25**

6.1 Authors 25

6.2 Copyright Notice and License 25

Appendices **27**

A Annotation file XML file example **27**

B A simple NDTF configuration file **27**

C A full NDTF configuration file **28**

1 Status of this document

This is V1.2 of the Neurophysiology Data Translation Format (NDTF) Specification.

2 Introduction

CARMEN aims to create an environment for sharing neurophysiology experiment data and algorithms using distributed computing technology [Foster, Kesselman, 2004]. For this purpose, a variety of user contributed applications will be published, either from “wrapped” legacy implementations or from new implementations. CARMEN will need to cater for a wide range of incoming data types (from various acquisition systems, models, etc.). Analysis applications will also be executed on raw experimental data, model data and on derived data. There is a need for data translation to allow these applications to access a single, stable data format.

The NDTF specifies a uniform file and format structure for data sharing and inter-application[1] data communication. In the first instance this will be implemented and provided within the CARMEN system. An NDTF dataset consists of a configuration file which manages a variable number of host data files. The configuration file uses an XML format. It contains metadata and references to the associated host data files. Users and applications can embed additional information in the configuration file, for example to record the history of data processing activity, or to associate the dataset with other datasets. A client application can extract the metadata directly from the configuration file remotely without the need to look into the associated binary data file.

The NDTF also specifies a set of the most commonly used experimental data entities as “NDTF internal data types”. These may be equally applicable to other signal or image processing applications, and include continuous time series, fixed/arbitrary length time series segments and spike times (e.g. events). NDTF may be extended arbitrarily to accommodate other data entities. The applications developed within the CARMEN project will provide support for the NDTF specified data types. Further, third party format data files can be attached to an NDTF dataset as an “external format”. This provides backward compatibility for non-CARMEN developed applications as well as allowing CARMEN applications to, at least, detect the format of the data even if they cannot provide full support for it. Examples of third party data include images and image series, and proprietary and bespoke time series data formats.

The MAT data format [Mat-File, 2008] is used as the main data format to encode the host data files, providing a generic and open framework for encoding numerical entities and arrays. To optimise storage space and network load, it is recommended that original digitized data input sets in integer type are stored as such, rather than converting them to floating point data types.

All the data files comprising one set of NDTF data are, by default, placed in the same directory as the appropriate configuration file, but if desired links to data in other locations can be specified using URIs in the configuration file.

---

[1] For brevity the term application is used throughout to indicate tools, services, software packages and applications.

Section-3 specifies the NDTF data types and their file formats. Section-4 specifies the format of the XML configuration files.

3 The NDTF Data Types and the host data file format

For data storage purposes, an NDTF data set may contain any type of data, managed by an NDTF configuration file. However, the use of data types that are defined outside the CARMEN system (i.e. *external data formats*) may not have full support from the CARMEN applications. A set of "*CARMEN internal data types*" are defined for data sharing and inter-application communication within the CARMEN system. This section will briefly look at the internal data types, support for external data, and annotation files defined as part of an NDTF dataset for use in the CARMEN system.

3.1 The NDTF internal data types

Technical Report [BL,JS, 2008] identified the data entities that will be acquired and created by user groups within the CARMEN project. In addition to time series data, these include images and image series. As image handling is broadly standardized, the NDTF uses existing standard formats for image data. Images are therefore represented as third party data, specified by internal data type *ImageData*. Time series data are specified by the following types: continuous analog data (*TimeSeriesData*), fixed/arbitrary length waveform data segments (*SegmentData*) and neural event data (*NeuralEvent*). A further data type *ExperimentEventData* is defined for annotation/marker event data.

In addition, users can specify custom data types and associate instances of them with NDTF datasets. Custom types may only be supported by a subset of the services within CARMEN. It is anticipated that custom data types may be fully incorporated by the NDTF specification as they become prevalent. Table 3.1 lists the CARMEN internal data types and their host data file formats.

Table 1: NDTF Internal Data Types

Data Type Name	**Descriptions**	**Host data file format**
ImageData	Image and image sequence data	Industrial standard
TimeSeriesData	One session waveform data	MATLAB Mat file
SegmentData	Waveform data segment	MATLAB Mat file
NeuralEventData	Such as spike time data	MATLAB Mat file
ExperimentalEventData	Event marker or annotation data	XML file
GenericMatrix	Reserved for internal use	MATLAB Mat file

CARMEN does not define new file formats for the storage of the raw source data, these are stored in their original proprietary or bespoke formats.

1. **ImageData** files are stored in the appropriate industry standard formats. When associated with an NDTF dataset, metadata are extracted and copied into the configuration file. Image data are therefore fully defined by the host data file.
2. **TimeSeriesData** is the basic data type for storage of multichannel signal data.

Data can be of any numeric data types defined by the associated MAT file. Metadata is stored in the associated configuration file. The host MAT data file may or may not contain full metadata. This data type is therefore fully defined by the configuration file together with the host data files. Data elements used within the host MAT file can be of *numeric array* or *cell array* element with numeric array sub-elements. Data are stored as one channel per array for numeric array or one cell per channel for cell array. Channel names are defined as the array name for numeric array, or cell name for cell array. A copy of the channel names is also stored in the configuration file.

3. **SegmentData** is the type used to store time-series data segments of one or more channels. Data elements used within the host MAT file are *cell array* element with numeric array sub-elements. Data can be represented using any numeric data type. The cell name is a decimal string in seconds that represents the time offset of the start point of each data segment. One cell array can contain more than one channel of data. The cell array name is defined as a list of plain text *channelName:startCellIndex* pairs separated by commas, where *channelName* is the name of the data channel and *startCellIndex* is the index of the cell in which the first data segment of the channel data is stored. The *channelName:startCellIndex*list must be consistent with *ChannelLabels* and *MatElementLabels* defined in the configuration file. If data is stored on a one channel to one cell array basis, the *channelName:startCellIndex* can contain only the *channelName* since in this case the *startCellIndex* is always zero. Other metadata are defined in the configuration file. Where multiple channels of data are stored, data segments from the same channel must be stored in a block of contiguous cells. The data segments can be of the same length or have arbitrary lengths.

4. **NeuralEventData** is the data type used to store event data, where only time instances are of concern, such as spike time data. Neural event data can be encoded in any permissible numerical data type defined within the host MAT file. Metadata is stored in the associated configuration file. Data elements used within the host MAT file can be *numeric array* or *cell array* element with numeric array sub-elements. Data is stored as one channel per array for numeric array or one cell per channel for cell array. Array names correspond to channel names, while cell names are used for cell array elements. Channel names are also specified in the configuration file. For all data types the time unit must be defined in the configuration file by the XML attribute *timeResolution*.

5. **ExperimentalEventData** is the data type used to store "time stamp – annotation" data pair sequences. Each pair identifies a time instance and attaches a short text comment or description of the event. The host file for this type is in XML format. Data pairs are not necessarily ordered by time. Two data pairs that define the start and end time instances of a given event can be grouped together by the XML tab *interval*. Time stamps used in the annotation data must be consistent with the time course in the raw data. Details of the XML schema used by the host file can be found in Section-3.3.

6. **GenericMatrix** provides a means of creating application specific data types during CARMEN application development. Generic matrix can be of any data type supported by the MAT format. Instances are defined and used within CARMEN.

Instances of *GenericMatrix* are signed by the identifier *applicationID* in the configuration file, which defines a group of applications that can read the host data file format.

3.2 NDTF external data support

The NDTF supports any third party data formats including image data files and the other vendor created data formats such as Multi-Channel System files (mcd). Such files are stored as part of the data set (in host data files) and the configuration file contains an entry in *UserDefinedData* to describe the data.

3.3 The Annotation data file format

The CARMEN internal data type *ExperimentalEventData* supports data annotation. The host annotation file contains event markers (time stamps), paired with descriptions of the events that took place. The major advantage of separating annotation data from raw data is that annotations can be applied to secondary data derived from raw data without any additional changes. *ExperimentalEventData* host files are encoded in XML, which can be generated programmatically and/or modified manually.

The host data file consists of one *description* element with multiple *eventNote* and/or *interval* elements which define the time instances at which events occurred.

- The *version* element provides the version number of the annotation file.
- The *description* element provides general information in text format about the annotation that the data represents.
- The *timeMarker* element defines the type of markers. When *timeMarker* is true, the time instances of the marked events are the time in seconds. Otherwise, the value of *timeMarker* represents the number of the frame of the image sequence.
- The *timeResolution* element defines the time resolution of the time offset at each time stamp in seconds. This value overrides the default parameter defined in the data configuration file – see Section-4.4.5. When the *timeMarker* element is set as false, this element is not meaningful.
- The *eventNote* element contains four attributes and one text message.
    1. The *timeOffset* attribute represents the value of the time instance at which the event occurs. The unit of this attribute is defined by element *timeMarker*. This is a required attribute.
    2. The *group_id* attribute is optional and is reserved for identifying different event groups. This allows applications to only process events of specific types from the host data file.
    3. The *attachedFile* element allows an external data file (such as an audio recording of the sound of the event) to be attached to the event. This attribute is optional.

4. The *application* attribute can only be specified when *attachedFile* is also defined. It defines a recommended application for opening the file defined by the *attachedFile* attribute. If *application* is not specified, the associated file will be opened by the system default application for the file type. *application* is therefore optional.

An *eventNote* element can be empty, or can contain a text message to describe the event.

- The *interval* element contains two *eventNote* elements that specify the start and end time instances of the event in question. An optional *group_id* attribute can be applied to identifying events of different groups.

3.3.1 The Annotation data file XML elements example

This section explains the structure of the annotation data file in XML.

The *version* element declares the version number of the annotation file:

```
<version>1.0</version>
```

The *description* element contains a description of the annotation data:

```
<description>
    Description of the annotation data.
</description>
```

The *timeMarker* defines attribute *timeOffset* is time in seconds.

```
 <timeMarker>true</timeMarker>
```

The *timeResolution* element defines the time resolution of the marked time instances. For example, the following element defines a time resolution as 1 μs (10^{-6} second).

```
  <timeResolution>0.000001</timeResolution>
```

If the *timeOffset* attribute is set as decimal value 12345678.99, it represents that the time offset from the beginning of the data set is $12345678.99 \times 0.000001 = 12.34567899$ seconds.

The *interval* elements contains two *eventNote* elements that define the start and end time instances of the event:

```
  <interval group_id="04">
    <eventNote timeOffset="1237888.23"
               attachedFile="sound1.wmv"
```

```
              application="realplayer">
      Text message for the event start.
    </eventNote>
    <eventNote timeOffset="18958585.23">
      Text message for the event end.
    </eventNote>
  </interval>
```

An *eventNote* element must contain a *timeOffset* attribute. When *timeMarker* is set to false, *timeOffset* is an integer representing a frame number within an image series. As described in the previous section, the other three attributes (*attachedFile*, *application*, and *group_id*) are optional. *eventNote* can be an empty element, or can contain a descriptive text string:

```
<eventNote timeOffset="123456.789" group_id="04"
        attachedFile="sound1.wmv"
        application="realplayer">
   Pulse detected by the acquisition system
</eventNote>
```

These are also valid elements:

```
<eventNote timeOffset="123456.789"
        attachedFile="sound1.wmv" >
   Pulse detected by the acquisition system
</eventNote>

<eventNote timeOffset="123458.000" >
   Pulse detected by the acquisition system
</eventNote>

<eventNote timeOffset="123490.222" />
```

The last element is empty, illustrating that the *eventNote* marker can be used to specify the time instance of a predefined (or known by convention) event. The *eventNote* elements need not necessarily be ordered by time.

The following example demonstrates a complete annotation file combining the above components:

```
<?xml version="1.0" encoding="utf-8"?>
<NDTF_Annotation
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.carmen.org.uk
                                ndtfAnnotation.msd"
      xmlns="http://www.carmen.org.uk" >
  <description>
    A new annotation file.
  </description>
  <timeMarker>true</timeMarker>
  <timeResolution>0.000001</timeResolution>

  <interval group_id="04">
    <eventNote timeOffset="1237888.230"
               attachedFile="sound1.wmv"
               application="realplayer">
      Text message for the event start.
    </eventNote>
    <eventNote timeOffset="18958585.232">
      Text message for the event end.
    </eventNote>
  </interval>

  <eventNote timeOffset="123456.789" group_id="04"
        attachedFile="sound2.wmv"
        application="realplayer">
   Pulse detected by the acquisition system
  </eventNote>

  <eventNote timeOffset="123456.222"
                      attachedFile="sound1.wmv" >
       Pulse detected by the acquisition system
   </eventNote>

   <eventNote timeOffset="123458.000" >
      Pulse detected by the acquisition system
   </eventNote>
</NDTF_Annotation >
```

4 The NDTF Configuration file

One set of NDTF data consists of a configuration file (XML) and an arbitrary number of host data files. The configuration file contains metadata associated with the host data files. The configuration file can be considered as a separated header, providing a common means for applications to interpret the host data file(s). Within a data processing chain, the output of one application may become the input for another. The XML configuration file, which can be created and modified programmatically, provides a means to pass information about transformations that have been applied to data along the data processing chain. Only the most recent transformation need be described at any point in the chain, as earlier transformations can be inherited from the data associated with previous steps.

4.1 Structure of the configuration file

A NDTF configuration file consists of a *Version* element and three sections: *GeneralInfo*, *DataSet*, and *History*. Each section is enclosed within XML tags. The *Version* element records the version of the NDTF specification that has been used to structure the configuration file. The *GeneralInfo* section describes the general information about the NDTF data; the *DataSet* section contains metadata required for interpreting the included host data file(s), and the *History* section is used to record the data processing history. Figure-1 shows the tree diagram of a configuration file.

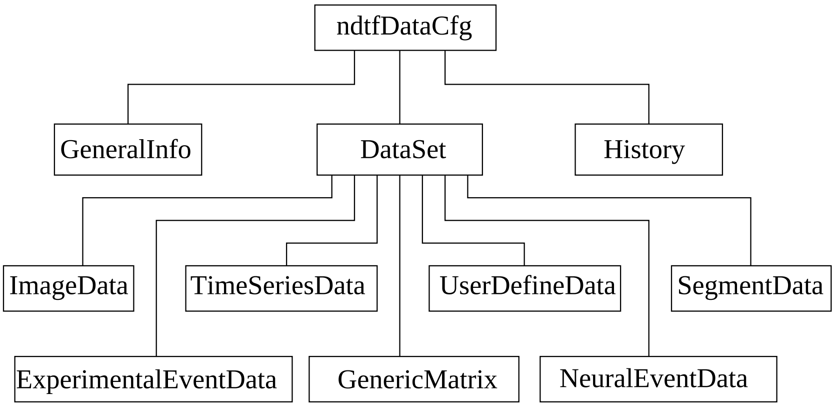

Figure 1: Configuration file tree diagram – summary

The following sub-sections describe the information that should be recorded inside the *GeneralInfo*, *DataSet*, and *History* sections.

4.2 General description element

This section provides general information about the NDTF dataset. It contains seven sub-elements, some of which are optional. The sub-elements need not conform to any particular ordering.

- *Description* contains a short textual description of the data. This is a required element.
- *Laboratory* is a text string which can be used to record a name or identifier for the laboratory where the data was generated. This is an optional element.
- *Investigator* is a text string of the name of the investigator. It is an optional element.
- *SpecimenID* is a text string which can be used to record a name or identifier for the biological specimen that was used for the recording (e.g. animal or tissue section). This is an optional element.
- *CreateDate* is a text string in YYYY-MM-DD format describing the date when the dataset was generated. This is a required element.
- *CreateTime* is a text string in hh:mm:ss format describing the time when the dataset was generated. This is an optional element.
- *RecordID* is an optional element reserved for the storage of the data ID.

An sample *GeneralInfo* section is as follows:

```
<GeneralInfo>
    <Description>
         This is a NDTF configuration file
       GeneralInfo section example.
    </Description>
    <Laboratory>CARMEN Lab</Laboratory>
    <Investigator>Smith Carmen</Investigator>
    <SpecimenID>anyname</SpecimenID>
    <CreateDate>2008-01-01</CreateDate>
    <CreateTime>12:38:38</CreateTime>
    <RecordID>abc123</RecordID>
</GeneralInfo>
```

4.3 DataSet element

The *DataSet* element is the core of the configuration file. It provides the information that applications require to interpret the dataset. The host data files contained in the dataset are defined as sub-elements of *DataSet*. Table-3.1 contains the six CARMEN internal data types. Within the *DataSet* element, each may be defined as a sub-element. *DataSet* can contain multiple sub-elements representing the same or different data types. For example, multiple *ExperimentalEventData* instances (e.g. annotations) may be created to describe different interpretations of the same data, or *ImageData* and *TimeSeriesData* instances may be combined to register concurrent recording modalities (e.g. imaging and multi-electrode array). Sub-elements within the *DataSet* section have an arbitrary order. Figure-1 shows the seven data types as the children of section *DataSet*.

In addition to the internal data types, an external data type *UserDefinedData* may be defined. This allows third party data format and other user defined extensions to be added, providing backward compatibility with proprietary or legacy software tools. The configuration file provides seamless access to these different representations based on the applications used to read the data. Users may arbitrarily define their own sub-elements within *UserDefinedData*.

While *UserDefinedData* may be extended by the user, two predefined methods are provided to access user defined datasets. One method defines an *applicationID* attribute, which can be used to specify an application or group of applications that can be used to access the data. The other defines a *recommendedApp* attribute, which can be used to specify a software package that is capable of accessing the data. This might be used to specify that a particular third-party software package provides support for the data format. If both *applicationID* and *recommendedApp* attributes are specified, *recommendedApp* is used by default.

4.3.1 The *DataInfo* element

The *DataInfo* element may be defined within several internal data types. Its contents and constraints are modified to make it specific. It contains metadata specific to neurophysiology experiment data such as sampling rate, filters and other equipment settings. Table-4.3.1 contains the full set of sub-elements for *DataInfo* and describes their respective constraints for different data types.

Table 2: The *Datainfo* sub-element and usages

Sub-elements of *DataInfo*	**NDTF DataTypes**			
	ImageData	*TimeSeriesData*	*SegmentData*	*NeuralEventData*
AcquisitionEquipment	Optional	Optional	Optional	Optional
EquipmentSettings	Optional	Optional	Optional	Optional
StartDatetime	Required	Required	Required	Required
EndDateTime	Optional	Optional	Optional	Optional
NumberOfChannels	N/A	Required	Required	Required
SamplingRate	N/A	Required	Required	Required
TransducerType	N/A	Optional	Optional	Optional
ADCSettings	N/A	Required	Required	Optional
LowPassFilter	N/A	Optional	Optional	Optional
HighPassFilter	N/A	Optional	Optional	Optional
ChannelLabels	N/A	Required	Required	Required
Trigger	N/A	N/A	Required	N/A

The sub-elements of *DataInfo* are defined as follows:

1. *AcquisitionEquipment* is a text string allowing a name or other identifier (e.g. model number, device identifier) for the data acquisition system to be specified. It is enclosed by a pair of XML tags:

   ```
   <AcquisitionEquipment>
   ```

```
   Name of the acquisition equipment
</AcquisitionEquipment>
```

2. *EquipmentSettings* is a short text string to describe the data acquisition system settings (only applicable if equipment is defined).

```
<EquipmentSettings>
   Message for equipment settings
</EquipmentSettings>
```

3. *StartDateTime* defines the time instance of the start of data recording. This sub-element consists of two attributes: the data-time string in CCYY-MM-DDThh:mm:ss format, and the decimal part of the second. The latter is stored as a floating point number, allowing different resolutions to be used.

```
<StartDateTime DateTime="2008-01-28T18:03:28"
                               decimalSeconds="0.000031"/>
```

4. *EndDateTime* defines the time instance of the end of the recording in the same format as *StartDateTime*.

```
<EndDateTime DateTime="2008-01-28T18:04:28"
             decimalSeconds="0.000031"/>
```

5. *NumberOfChannels* is the number of channels in the recording, e.g.

```
 <NumberOfChannels>64</NumberOfChannels>.
```

6. *SamplingRate* defines the data sampling rate in Hz.

```
<SamplingRate>25000</SamplingRate>
```

7. *TransducerType* defines the type of the signal transducer. It is a text string and specified as follows:

```
<TransducerType>active electrode</TransducerType>
```

8. *ADCSettings* describes the Analog-to-Digital conversion. Attribute *precision* is the precision of the Analog-to-Digital convertor, e.g. 12-bit. Attribute *zeroOffset* is the offset of the real zero level. *resolution* defines the quantization step. The unit of both *zeroOffset* and *resolution* is given by attribute *unit*. For example, given an Analog-to-Digital convertor of 12-bit precision, ADC resolution 0.00001221V and zero offset of -0.020V, the maximum input signal level will be $0.00001221 \times (2^{12} - 1) - 0.020 = 0.030$(V) and the minimum input signal level will be -0.020(V). If *ADCSettings* is not applicable, e.g. for a non-digitized signal, it may be omitted.

```
<ADCSettings precision="12" zeroOffset="0.020"
               resolution="0.0000121" unit="v"/>
```

9. *LowPassFilter* and *HighPassFilter* defines the settings for any low-pass and / or high-pass filters used. The attributes are self explanatory. *cutoffFrequency* is specified in Hz.

   ```
   <LowPassFilter cutoffFreqency="60000"
                filterType="Chebyshev" order="10"/>
   <HighPassFilter cutoffFreqency="5.10"
                filterType="Butterworth" order="10"/>
   ```

10. *ChannelLabels* defines a list of channel names in CSV (Comma Separated Value) format. The values in this list must correspond with the location of the data channels defined with the MAT file data element list defined on page 18 (item *StructInfo*).

    ```
    <ChannelLabels>ch11, ch12, ch15, ch86</ChannelLabels>
    ```

11. *Trigger* defines the trigger threshold and time span of the triggered data segments (where applicable). The unit of the threshold must match the unit of the signal values defined in *ADCsettings*. Left-span and right-span are in seconds.

    ```
    <Trigger threshold="0.0003"
             leftSpan="0.0025" rightSpan="0.0025"/>
    ```

4.4 Data type definitions

4.4.1 ImageData

An *ImageData* section containing all the possible sub-elements looks like this:

```
<ImageData filename="imagefilename">
    <Location>
      http://carmen/data/exampleData.mov
    </Location>
    <DataInfo>
       <AcquisitionEquipment>
         Name of the acquisition equipment
            </AcquisitionEquipment>
       <EquipmentSettings>
           Camera AGC turned off, Gamma=0.75
             </EquipmentSettings>
       <StartDateTime DateTime="2008-01-28T18:03:28"
                      decimalSeconds="0.000031"/>
       <EndDateTime DateTime="2008-01-28T19:03:28"
                    decimalSeconds="0.000031"/>
    </DataInfo>
    <Dimension width="100" height="288"/>
    <FrameInfo frameCnt="268" frameRate="15"/>
    <RecommendedApp>imageviewer</RecommendedApp>
</ImageData>
```

- The *filename* attribute specifies the host data file name. By default, the host data file is in the same directory as the configuration file. Depending on the application, the host data file may be downloaded from the remote site defined by the *Location* parameter.
- The *Location* sub-element defines the URI of the host data file wherever applicable. This need not be specified if the host data files are stored in the same directory as the configuration file.
- The *DataInfo* sub-elements are as defined in Section-4.3.1.
- The *Dimension* sub-element defines the dimensions of the image data (i.e. frame dimensions) in pixels.
- *FrameInfo* defines the total number of frames in the host file, and the frame rate (frames per second) if the host file is an image sequence.
- *RecommendedApp* is an optional sub-element. It provides the name of a recommend application to open the host data file.

4.4.2 TimeSeriesData

The *TimeSeriesData* sub-element defines one section of contiguous time-series recording. Both single and multiple channel recordings may be defined:

```
<TimeSeriesData filename="multipleChannelDataFilename"
                unit="mv">
  <Location>http://carmen/data/exampleData.mat</Location>
  <DataInfo>
     <AcquisitionEquipment>
        Name of the acquisition equipment
     </AcquisitionEquipment>
     <EquipmentSettings>
       Message for equipment settings
     </EquipmentSettings>
     <StartDateTime DateTime="2008-01-28T18:03:28"
                    decimalSeconds="0.000031"/>
     <EndDateTime DateTime="2008-01-28T19:03:28"
                  decimalSeconds="0.000031"/>
     <NumberOfChannels>128</NumberOfChannels>
     <SamplingRate>20000</SamplingRate>
     <TransducerType>active electrode</TransducerType>
     <ADCSettings precision="12" zeroOffset="0.0388"
                   resolution="0.000015" unit="mv"/>
     <LowPassFilter cutoffFreqency="60000"
                  filterType="Chebyshev" order="10"/>
     <HighPassFilter cutoffFreqency="5"
                  filterType="Butterworth" order="10"/>
     <ChannelLabels>
         ch11, ch12, ch15, ch86 ...
     </ChannelLabels>
  </DataInfo>
  <StructInfo>
     <MatElementLabels timeOffset="0.0000345">
  Index1:cell\_Index1, Index2:cell\_Index2, 5:1, 6, ...
     </MatElementLabels>
  </StructInfo>
  <ExtraInfo>
     Plain text message providing extra information such
     as sample type, position, etc that are not defined
     in the DataInfo element.
   </ExtraInfo>
   <RecommendedApp>Signal Data Explorer</RecommendedApp>
</TimeSeriesData>
```

In addition to the (*filename*, *Location*, *RecommendedApp* and *DataInfo*) parameters defined already, the following parameters may be set within the *TimeSeriesData* sub-element:

- Attribute *unit* defines the unit of the signal value. This is a required attribute, however, the attribute *ADCSettings::unit* will override this setting if both are defined.
- *StructInfo* provides information about the data location within the host data file. Currently, only one data structure type, *MatElementLabels* is defined for the host host data file, is defined. It may be extended to support other host data files in the future. The sub-element *MatElementLabels* defines a list of MAT elements in which the relevant channels of data defined in the section *DataInfo::ChannelLabels* are stored. When data is stored in a cell array, the element *MatElementLabels* list is defined as a set of data pairs representing the zero based indices of the cell array and the cell in which the data is stored. For example, 5:1 represents the 6th cell array in the file and the data is stored in the 2nd cell of the cell array. If the cell index is not presented, a numerical array or the first cell of a cell array is assumed. Each element in the list is separated by a comma. This list must correspond with the channel labels list defined in *DataInfo*. The attribute *timeOffset* within element *StructInfo* defines the time offset between the observed start of the recording and the first recorded datapoint.
- The *ExtraInfo* sub-element provides additional information about the data, in plain text.

4.4.3 SegmentData

The *SegmentData* sub-element defines the data structure of multiple channel, multiple segment time series data such as triggered MEA (Multi Electrode Array)data. Data is stored as a cell array within a MAT data file:

```
<SegmentData
```

```
   filename="multipleChannelTriggeredDataFilename"
   unit="mv">
   <Location>
      http://carmen/data/exampleData.mat
   </Location>
   <DataInfo>
     <AcquisitionEquipment>
        Name of the acquisition equipment
     </AcquisitionEquipment>
     <EquipmentSettings>
        Message for equipment settings
     </EquipmentSettings>
     <StartDateTime DateTime="2008-01-28T18:03:28"
                    decimalSeconds="0.000031"/>
     <EndDateTime DateTime="2008-01-28T19:03:28"
                 decimalSeconds="0.000031"/>
     <NumberOfChannels>64</NumberOfChannels>
     <SamplingRate>20000</SamplingRate>
     <TransducerType>active electrode</TransducerType>
     <ADCSettings precision="12" zeroOffset="0.0388"
                    resolution="0.000015" unit="mv"/>
     <LowPassFilter cutoffFreqency="60000"
                    filterType="Chebyshev" order="10"/>
     <HighPassFilter cutoffFreqency="5"
                  filterType="Butterworth" order="10"/>
     <Trigger threshold="0.0003" leftSpan="0.0025"
                                 rightSpan="0.0025"/>
   <ChannelLabels>ch11, ch12, ch15, ch86</ChannelLabels>
   </DataInfo>
   <StructInfo>
   <MatElementLabels timeOffset="0.0000345">
        0:0, 0:1, 5:1, 6, 7, 19...
   </MatElementLabels>
   </StructInfo>
   <ExtraInfo>
     Plain text message providing extra information
     such as sample type, position, etc that are
     not defined from the DataInfo element.
   </ExtraInfo>
   <RecommendedApp>Signal Data Explorer</RecommendedApp>
</SegmentData>
```

All parameters defined within *SegmentData* have the same specification as *TimeSeriesData*, with the exception of the *MatElementLabels* list. For each *Index:cell_Index* pair, *cell_Index* is defined as follows:

1. If data is stored as one channel per cell array, *cell_Index* is always 0, indicating that the record starts from the first cell. In this case parameter *cell_Index* need

not be specified and may be omitted.

2. If data is stored as multiple channels per cell array, *cell_Index* represents the cell index of the first data segment in the channel.

4.4.4 NeuralEventData

The *NeuralEventData* sub-element defines the data structure of neural event data such as spike times. The values of this data type represent the time instances of events:

```
<NeuralEventData filename="spikeTimeDataFilename"
                 timeResolution="0.00001">
  <Location>http://carmen/data/exampleData.mat</Location>
  <DataInfo>
        <AcquisitionEquipment>
            Name of the acquisition equipment
        </AcquisitionEquipment>
        <EquipmentSettings>
            Message for equipment settings
        </EquipmentSettings>
        <StartDateTime DateTime="2008-01-28T18:03:28"
                       decimalSeconds="0.000031"/>
        <EndDateTime DateTime="2008-01-28T19:03:28"
                     decimalSeconds="0.000031"/>
        <NumberOfChannels>128</NumberOfChannels>
        <SamplingRate>20000</SamplingRate>
        <TransducerType>active electrode</TransducerType>
        <ADCSettings precision="12" zeroOffset="0.0388"
                      resolution="0.000015" unit="mv"/>
        <LowPassFilter cutoffFreqency="60000"
                      filterType="Chebyshev" order="10"/>
        <HighPassFilter cutoffFreqency="5"
                    filterType="Butterworth" order="10"/>
        <ChannelLabels>
            ch11, ch12, ch15, ch86
        </ChannelLabels>
  </DataInfo>
  <StructInfo>
         <MatElementLabels timeOffset="0.0000345">
             Elm1:subElm1, Elm2:subElm2, 5:1, 6, 7, 19...
         </MatElementLabels>
  </StructInfo>
  <ExtraInfo>
          Plain text message providing extra information
          such as sample type, position, etc that are
          not defined from the DataInfo element.
  </ExtraInfo>
  <RecommendedApp>Signal Data Explorer</RecommendedApp>
</NeuralEventData>
```

The parameters defined within *neuralEventData* have the same specifications as those in *TimeSeriesData*, with the exception of *timeResolution*, which is specific to this data type. *timeResolution* defines the time resolution in seconds of the neural event time instance values. For example, given a time resolution of 0.000005 seconds and an event time value of 100000, the real time value will be $100000 \times 0.000005 = 0.5$ seconds.

4.4.5 ExperimentalEventData

As described previously, *ExperimentalEventData* (annotations) are fully defined by their host data files (XML), generated programmatically or manually. The *ExperimentalEventData* element contains cursory information required for the interpretation of these data and links out to their respective file locations:

```
<ExperimentalEventData
       filename="AnnotationFilename"
       timeResolution="0.00001">
       <Location>
          http://carmen/data/datafilename
       </Location>
</ExperimentalEventData>
```

The *timeResolution* attribute defines the time resolution (in seconds) of the time offset markers defined in the host data file. The format of the annotation file is described in Section-3.3

4.4.6 GenericMatrix

This data type is defined as an extensible data type, for the purpose of CARMEN application development. The syntax of the data type is as follows:

```
<GenericMatrix filename="InternalDataFilename"
                     applicationID="appID"
                     unit="mA">
     <Location>
         http://carmen/data/exampleData.mat
     </Location>
     <AppDefinedDataInfo>
        <DefineYourTagsHere>
             Parameters defined here describe the data structure.
        </DefineYourTagsHere>
     </AppDefinedDataInfo>
</GenericMatrix>
```

The *filename* attribute defines the host data file name. Attribute *applicationID* is the data type identifier, which in effect defines a group of applications that support the data format. The method for storing and referencing these applications is a system level consideration that is out of the scope of this document. The optional attribute *unit* defines the unit of the data values contained in the bespoke data, where this is applicable. The key sub-element of this data type is *AppDefinedDataInfo*. Applications may define any number of XML elements and attributes within this element, to arbitrarily specify their dataset. For example, the *DefineYourTagsHere* element in the above example is a user defined element, which has been added to the *GenericMatrix* definition to create a custom type. This data type has two purposes. First, there may be some application specific datasets such as intermediate data that need to be shared within groups of applications but are not already defined as internal data types. By making use of *GenericMatrix* it will be possible to reuse these data types, making algorithm development and deployment more straightforward. Second, it provides a migration path for new CARMEN internal data types to be added to the NDTF specification over time.

4.4.7 UserDefinedData

A *UserDefinedData* may be treated as the combination of *TimeSeriesData* and the *GenericMatrix* data type. It is used to reference a third party data file. The syntax of a *UserDefinedData* element is as follows,

```
<UserDefinedData filename="UserDefinedDataFilename"
                            applicationID="appID"
                            unit="mv">
   <Location>http://carmen/data/dataname</Location>
   <DataInfo>
      <AcquisitionEquipment>
              Name of the acquisition equipment
      </AcquisitionEquipment>
      <EquipmentSettings>
         Message for equipment settings
      </EquipmentSettings>
      <StartDateTime DateTime="2008-01-28T18:03:28"
                     decimalSeconds="0.000031"/>
      <EndDateTime DateTime="2008-01-28T19:03:28"
                   decimalSeconds="0.000031"/>
      <NumberOfChannels>128</NumberOfChannels>
      <SamplingRate>20000</SamplingRate>
      <TransducerType>active electrode</TransducerType>
      <ADCSettings precision="12" zeroOffset="0.0388"
                    resolution="0.000015" unit="mv"/>
      <LowPassFilter cutoffFreqency="60000"
                     filterType="Chebyshev" order="10"/>
      <HighPassFilter cutoffFreqency="5"
                   filterType="Butterworth" order="10"/>
      <ChannelLabels>
          ch11, ch12, ch15, ch86
      </ChannelLabels>
    </DataInfo>
    <ExtraInfo>
        Plain text message providing extra information
        such as sample type, position, etc that are
        not defined from the DataInfo element.
    </ExtraInfo>
    <RecommendedApp>preferred app</RecommendedApp>
    <UserInfo>
       <DefineYourTagsHere>
            Define your data with your new tags.
       </DefineYourTagsHere>
     </UserInfo>
</UserDefinedData>
```

Because this is a user defined data type, the sub-elements *DataInfo* and *ExtraInfo* are optional (in contrast to the typical *TimeSeriesData* element, where they are mandatory). The *UserInfo* element is similar to the *AppDefinedData* sub-element in *GenericMatrix*. Users can define an arbitrary range of tags here to specify the custom element of their data type. This approach allows both CARMEN and third party applications to understand the data. For *UserDefinedData* the *unit* attribute is also optional.

4.5 History element

The *History* element provides a mechanism to record the data processing history. Each record is stored within a pair of *Processor* XML tags. It is recommended that all CARMEN applications copy this section to the configuration file of the output data set and append their history to the end of the section as a new *Processor* sub-section. In this way, all datasets will be traceable within CARMEN.

```
<History>
   <Processor>
     <ProcessingDateTime
          StartDateTime="2008-01-28T18:03:28"
          EndDateTime="2008-01-28T18:03:30" />
     <CommandLine>
"spikedector.exe -f mydata.csd -a 5 -b 10 > this_data"
     </CommandLine>
     <ProcessingSettings>
        If it is not a command line application, use
        free form text message to describe the data
        processing parameters, application name,
        input filename and output filename, etc.
     </ProcessingSettings>
  </Processor>
</History>
```

The children of element *processor* are defined as follows,

- *ProcessingDateTime* defines the date/time instance when the process was initiated. The optional attribute *EndDateTime* can be used to record the date/time instance when the process concluded. This element is in plain text.
- *CommandLine* records the execution call(s) that invoked the process in plain text format. Command line recording is the recommended method for storing processing records. This makes it possible to read the processing history and apply the same process to other datasets automatically. More than one command line call may be recorded. Each is enclosed in quotes.
- *ProcessingSettings* is used when a command line record is not applicable. This sub-element allows the recording of a piece of script or a short text message to describe the data processing settings.

5 Summary

The NDTF is a wrapped data format consisting of one configuration file and one or more host data files. The NDTF specification does not define new formats for the host data files. Instead, established third party data formats that are open and provide mature tooling for data access (such as the MAT data format and XML) are used. The wrapped dataset allows applications in the CARMEN system to understand data and supports backward compatibility with third party tooling. The wrapped dataset contains an XML configuration file consisting of three sections. In addition to the *GeneralDescription* section, the *DataSet* section supports six predefined internal data types and one user defined data type which is extensible. The *History* element also provides a mechanism to retain knowledge of data processing steps along the processing chain. The NDTF configuration file provides a rich set of information about the experiments. Being in XML format, it is possible to quickly extract and display the contents of an NDTF configuration file of interest within a web page such as the CARMEN portal.

Appendices A-C provide XML templates for the annotation and configuration files, which can be used as examples to create new datasets. For more details of the NDTF configuration file definitions, the XML schema files can be found on the CARMEN SVN site. This specification is a work in progress.

6 Technical Information

6.1 Authors

Bojian Liang
Department of Computing Science, University of York,
York, YO10 5DD United Kingdom
bojian@cs.york.ac.uk

Jennifer Simonotto
School of Computing Science, Newcastle University,
Newcastle upon Tyne, NE1 7RU United Kingdom
Jennifer.Simonotto@newcastle.ac.uk

Alastair Knowles
School of Computing Science, Newcastle University,
Newcastle upon Tyne, NE1 7RU United Kingdom
alastair.knowles@newcastle.ac.uk

Martyn Fletcher
Department of Computing Science, University of York,
York, YO10 5DD United Kingdom
martyn.fletcher@cs.york.ac.uk

6.2 Copyright Notice and License

2007 Copyright by CARMEN Design Group. This is a recommendation document distributed under the terms of the complete Creative Commons Attribution License (`http://creativecommons.org/licenses/by/2.5`), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

References

[BL,JS, 2008] Liang, B. and Simonotto, J. (2008). Technical Report: Data Format in CARMEN V2.0 *http://metagenome.ncl.ac.uk/subversion/carmen/trunk/Documents*.

[Mat-File, 2008] The MathWorks, Inc.(2008). Mat-File Format 7.6 *http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/matfile_format.pdf*.

[Foster, Kesselman, 2004] Foster, I. and Kesselman, C. (2004). The Grid: Blueprint for a new computing infrastructure (2nd ed.). Morgan Kaufmann Publishers, ISBN: 1-55860-475-8.

Appendices

A Annotation file XML file example

```
<?xml version="1.0" encoding="utf-8"?>
<NDTF_Annotation
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.carmen.org.uk   ndtfAnnotation.msd"
    xmlns="http://www.carmen.org.uk" >
  <version>1.0</version>
  <description>
    A new testing CARMEN annotation file
  </description>
  <timeMarker>true</timeMarker>
  <timeResolution>0.000001</timeResolution>

  <eventNote timeOffset="18958585.23">
      Stopped
  </eventNote>
  </interval>
  <interval group_id="04">
    <eventNote timeOffset="1237888.23"
               attachedFile="sound1.wmv"
               application="realplayer">
      Setup data
    </eventNote>
    <eventNote timeOffset="18958585.23"
               attachedFile="sound2.wmv"
               application="realplayer">
      Stopped
    </eventNote>
  </interval>
  <interval group_id="02">
    <eventNote timeOffset="1237888.23">
      Setup data
    </eventNote>
    <eventNote timeOffset="18958585.23">
      Stopped
    </eventNote>
  </interval>
</NDTF_Annotation>
```

B A simple NDTF data configuration file example

```
<?xml version="1.0" encoding="utf-8"?>
<ndtfDataCfg
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.carmen.org.uk ndtfDataCfg.xsd"
     xmlns="http://www.carmen.org.uk">

  <Version>1.0</Version>

  <GeneralInfo>
      <Description>Experiment 1 </Description>
      <Laboratory>CARMEN Lab</Laboratory>
      <Investigator>John Smith</Investigator>
      <SpecimenID>anyname</SpecimenID>
      <CreateDate>2008-01-01</CreateDate>
      <CreateTime>12:38:38</CreateTime>
      <RecordID>abe123</RecordID>
  </GeneralInfo>

  <DataSet>
     <NeuralEventData filename="spikeTimeData.mat"
                                timeResolution="0.00001">
          <DataInfo>
              <StartDateTime dateTime="2008-01-01T18:03:28"
                             decimalSeconds="0.000031"/>
              <NumberOfChannels>4</NumberOfChannels>
              <SamplingRate>20000</SamplingRate>
              <ChannelLabels>ch11, ch12, ch15, ch86</ChannelLabels>
          </DataInfo>
          <StructInfo>
              <MatElementLabels timeOffset="0.0000345">
                   0:0, 0:1, 0:2, 5:1
              </MatElementLabels>
          </StructInfo>
          <ExtraInfo>
                 Example spike time data.
          </ExtraInfo>
     </NeuralEventData>
  </DataSet>

  <History>
      <Processor>
          <ProcessingDateTime StartDateTime="2008-04-01T18:03:28"/>
          <CommandLine>
                spikedector.exe -f mydata.mcd -a 5 -b 10
          </CommandLine>
      </Processor>
  </History>
</ndtfDataCfg>
```

C Full NDTF data configuration file template

```
<?xml version="1.0" encoding="utf-8"?>
<ndtfDataCfg
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.carmen.org.uk ndtfDataCfg.xsd"
     xmlns="http://www.carmen.org.uk">
  <Version>1.0</Version>
  <GeneralInfo>
      <Description>Experiment 1 </Description>
      <Laboratory>CARMEN Lab</Laboratory>
      <Investigator>True Man</Investigator>
      <SpecimenID>anyname</SpecimenID>
      <CreateDate>2008-01-01</CreateDate>
      <CreateTime>12:38:38</CreateTime>
      <RecordID>abe123</RecordID>
  </GeneralInfo>

  <DataSet>
      <ImageData filename="imagefilename">
          <Location>location URI</Location>
          <DataInfo>
              <AcquisitionEquipment>
```

```
           Name of the acquisition equipment
        </AcquisitionEquipment>
        <EquipmentSettings>
        Camera AGC turn off, Gamma=0.75
        </EquipmentSettings>
        <StartDateTime dateTime="2008-01-28T18:03:28"
                              decimalSeconds="0.000031"/>
        <EndDateTime dateTime="2008-01-28T19:03:28"
                              decimalSeconds="0.000031"/>
    </DataInfo>
    <Dimension width="100" height="288"/>
    <FrameInfo frameCnt="268" frameRate="15"/>
    <RecommendedApp>imageviewer</RecommendedApp>
</ImageData>

<TimeSeriesData filename="multipleChannelDataFilename"
                unit="mv">
    <Location>location URI</Location>
    <DataInfo>
        <AcquisitionEquipment>
            Name of the acquisition equipment
        </AcquisitionEquipment>
        <EquipmentSettings>
            Message for equipment settings
        </EquipmentSettings>
        <StartDateTime dateTime="2008-01-28T18:03:28"
                                    decimalSeconds="0.000031"/>
        <EndDateTime dateTime="2008-01-28T19:03:28"
                                    decimalSeconds="0.000031"/>
        <NumberOfChannels>128</NumberOfChannels>
        <SamplingRate>20000</SamplingRate>
        <TransducerType>active electrode</TransducerType>
        <ADCSettings precision="12" zeroOffset="0.0388"
                          resolution="0.000015" unit="mv"/>
        <LowPassFilter cutoffFreqency="60000"
                        filterType="Chebyshev" order="10"/>
        <HighPassFilter cutoffFreqency="5"
                      filterType="Butterworth" order="10"/>
        <ChannelLabels>
              ch11, ch12, ch15, ch86 ...
        </ChannelLabels>
    </DataInfo>
    <StructInfo>
        <MatElementLabels timeOffset="0.0000345">
             Elm1:subElm1, Elm2:subElm2, 5:1, 6, 7, 19...
        </MatElementLabels>
    </StructInfo>
    <ExtraInfo>
           Plain text message provide extra information such as
        sample type, position, etc that are not defined from the
        DataInfo element.
    </ExtraInfo>
    <RecommendedApp>Signal Data Explorer</RecommendedApp>
</TimeSeriesData>

<SegmentData filename="multipleChannelTriggeredDataFilename"
             unit="mv">
    <Location>location URI</Location>
    <DataInfo>
        <AcquisitionEquipment>
            Name of the acquisition equipment
        </AcquisitionEquipment>
        <EquipmentSettings>
           Message for equipment settings
        </EquipmentSettings>
        <StartDateTime dateTime="2008-01-28T18:03:28"
                                  decimalSeconds="0.000031"/>
        <EndDateTime dateTime="2008-01-28T19:03:28"
                                  decimalSeconds="0.000031"/>
        <NumberOfChannels>64</NumberOfChannels>
        <SamplingRate>20000</SamplingRate>
        <TransducerType>active electrode</TransducerType>
        <ADCSettings precision="12" zeroOffset="0.0388"
                       resolution="0.000015" unit="mv"/>
        <LowPassFilter cutoffFreqency="60000"
                      filterType="Chebyshev" order="10"/>
        <HighPassFilter cutoffFreqency="5"
                    filterType="Butterworth" order="10"/>
        <Trigger threshold="0.0003"
                   leftSpan="0.0025" rightSpan="0.0025"/>
        <ChannelLabels>
               ch11, ch12, ch15, ch86
        </ChannelLabels>
    </DataInfo>
    <StructInfo>
        <MatElementLabels timeOffset="0.0000345">
             Elm1:subElm1, Elm2:subElm2, 5:1, 6, 7, 19...
        </MatElementLabels>
    </StructInfo>
    <ExtraInfo>
           Plain text message provide extra information such as
        sample type, position, etc that are not defined from the
        DataInfo element.
    </ExtraInfo>
  <RecommendedApp>Signal Data Explorer</RecommendedApp>
</SegmentData>

<NeuralEventData filename="spikeTimeDataFilename"
                          timeResolution="0.00001">
    <Location>location URI</Location>
    <DataInfo>
        <AcquisitionEquipment>
            Name of the acquisition equipment
        </AcquisitionEquipment>
        <EquipmentSettings>
           Message for equipment settings
        </EquipmentSettings>
        <StartDateTime dateTime="2008-01-28T18:03:28"
                                      decimalSeconds="0.000031"/>
        <EndDateTime dateTime="2008-01-28T19:03:28"
                                      decimalSeconds="0.000031"/>
        <NumberOfChannels>128</NumberOfChannels>
        <SamplingRate>20000</SamplingRate>
        <TransducerType>active electrode</TransducerType>
        <ADCSettings precision="12" zeroOffset="0.0388"
                           resolution="0.000015" unit="mv"/>
        <LowPassFilter cutoffFreqency="60000"
                          filterType="Chebyshev" order="10"/>
        <HighPassFilter cutoffFreqency="5"
                        filterType="Butterworth" order="10"/>
        <ChannelLabels>ch11, ch12, ch15, ch86</ChannelLabels>
    </DataInfo>
    <StructInfo>
        <MatElementLabels timeOffset="0.0000345">
```

```
             Elm1:subElm1, Elm2:subElm2, 5:1, 6, 7, 19...
        </MatElementLabels>
    </StructInfo>
    <ExtraInfo>
           Plain text message provide extra information such as
        sample type, position, etc that are not defined from the
        DataInfo element.
    </ExtraInfo>
  <RecommendedApp>Signal Data Explorer</RecommendedApp>
</NeuralEventData>

<ExperimentalEventData filename="AnnotationFilename"
                            timeResolution="0.00001">
     <Location>location URI</Location>
</ExperimentalEventData>

<GenericMatrix filename="InternalDataFilename"
               applicationID="appID" unit="mA">
     <Location>location URI</Location>
     <AppDefinedDataInfo>
         <DefineYourTagsHere>
             Parameters place within the application define tags
         </DefineYourTagsHere>
     </AppDefinedDataInfo>
</GenericMatrix>

<UserDefinedData filename="UserDefinedDataFilename"
                 applicationID="appID" unit="mv">
     <Location>location URI</Location>
     <DataInfo>
        <AcquisitionEquipment>
            Name of the acquisition equipment
        </AcquisitionEquipment>
        <EquipmentSettings>
            Message for equipment settings
        </EquipmentSettings>
        <StartDateTime dateTime="2008-01-28T18:03:28"
                                    decimalSeconds="0.000031"/>
        <EndDateTime dateTime="2008-01-28T19:03:28"
                                    decimalSeconds="0.000031"/>
        <NumberOfChannels>128</NumberOfChannels>
        <SamplingRate>20000</SamplingRate>
        <TransducerType>active electrode</TransducerType>
        <ADCSettings precision="12" zeroOffset="0.0388"
                        resolution="0.000015" unit="mv"/>
        <LowPassFilter cutoffFreqency="60000"
                       filterType="Chebyshev" order="10"/>
        <HighPassFilter cutoffFreqency="5"
                     filterType="Butterworth" order="10"/>
        <ChannelLabels>
                  ch11, ch12, ch15, ch86</ChannelLabels>
     </DataInfo>
     <ExtraInfo>
           Plain text message provide extra information such as
        sample type, position, etc that are not defiined from the
        DataInfo element.
     </ExtraInfo>
     <RecommendedApp>preferred app</RecommendedApp>
     <UserInfo>
         <DefineYourTagsHere>
             defined your data for your new tags.
         </DefineYourTagsHere>
```

```
           </UserInfo>
      </UserDefineData>
  </DataSet>

  <History>
      <Processor>
          <ProcessingDateTime StartDateTime="2008-01-28T18:03:28"
                              EndDateTime="2008-01-28T18:03:30" />
          <CommandLine>
                spikedector.exe -f mydata.mcd -a 5 -b 10
          </CommandLine>
          <ProcessingSettings>
              If it is not a command line application, use free
              form text message to describe the data processing
              parameters, application name,  input filename and
              output filename, etc.
          </ProcessingSettings>
      </Processor>
  </History>

</ndtfDataCfg>
```