



REGULATORY REVIEW OF CONTINUOUS ECGs

Annotated ECG Implementation Guide

-

Continuous Waveforms Supplement

Edited by: Barry Brown

2-December-2013

In partnership with:



Table of Contents

- Scope..... 3
- Background 3
- Waveforms in XML..... 3
- Annotations in XML..... 4
- Challenge..... 5
- Need for Alternate Solution..... 6
- Need for ECG Warehouse v2 Format..... 6
- HL7 Standardization of aECG v2..... 6
- ECG Waveform File Formats 7
- Referencing a Lead in an Interleaved File 7
- ECG Waveforms Example..... 9
- Beat-by-Beat Annotations in Binary Files 11
- Beat-by-Beat Annotations in Text Files..... 12
- Beat-by-Beat Annotation Example 13
- All Other Waveform Annotations 13
- Time Windows 14
 - Time Point Windows 14
 - Analysis Windows 14
- XML Structure for Windows and Annotations..... 15
- Time Point and Analysis Windows Example 16
- Baseline Windows..... 20
- Protocol Event Annotations 21

Scope

This *Continuous Waveforms Supplement* to HL7's existing *aECG Implementation Guide* presents an alternate way to encode ECG waveforms and their beat-by-beat annotations while maintaining the current aECG message structure used by the FDA to review ECG recordings.

Background

The HL7 v3 Annotated ECG (aECG) standard was developed and approved in the 2002-2005 timeframe. At that time, FDA and industry were focused on traditional 10-second 12-lead ECG recordings that were used to study drug-induced QT effects in healthy volunteers. The aECG information model was developed according to HL7's Version 3 message development process in place at that time. It involved the creation and approval of an information model. Once approved, the information model was compiled into XML Schema by tools maintained by HL7.

The V3 information modeling constructs are general enough to handle all types of healthcare information (registration, financial, scheduling, medical, people, devices, results, etc.). Therefore, when the models are automatically compiled into XML Schema, some parts can be inefficient in how information is encoded and structured. The structure can be several orders of magnitude larger than the actual information being communicated. Most of the time this inefficiency is not important because HL7 messages tend to carry relatively small amounts of data, and computer memory is plentiful, network speeds are fast, and disk space is cheap.

A 10-second, 12-lead ECG, sampled at 1000 samples-per-second contains 120,000 sample values. Typical ECGs in a thorough QT study may have annotations such as P-onset, QRS-onset, R-peak, QRS-offset, and T-offset for 3 beats. The typical ECG may have 20 to 30 annotations. Typical aECG XML files are roughly 0.5 MB each (uncompressed). Although inefficient when compared to binary storage of ECG information, these XML files do not tax modern computers, networks, and software architecture.

Waveforms in XML

ECG waveforms are currently encoded directly in the XML as a listing of integer sample values separated by white space. A single digit value and a space takes 2 characters (bytes) to encode, a negative 3-digit value takes 5 characters (bytes) to encode, and so on. Compared to the waveform samples, there is relatively little XML overhead per ECG waveform, although it takes 2 to 5 bytes to encode each sample.

```
<component>
  <sequence classCode="OBS">
    <code code="MDC_ECG_LEAD_I" etc./>
    <value xsi:type="SLIST_PQ">
      <origin value="0" unit="uV"/>
      <scale value="3.75" unit="uV"/>
      <digits>-5 -5 -3 -1 -3 -4 -4 -4 etc.</digits>
    </value>
  </sequence>
</component>
```

Annotations in XML

Below is an example of a QRS-onset annotation at 345 ms from the beginning of the recording. There are two important pieces of information encoded: 1) it's a QRS annotation, 2) its onset is 345ms from the beginning of the ECG waveform. The rest of the XML is mostly structural overhead generated by the HL7 v3 information model.

```
<annotation classCode="OBS">
  <code code="MDC_ECG_WAVC" />
  <value xsi:type="CE" code="MDC_ECG_WAVC_QRSWAVE" />
  <support typeCode="SPRT">
    <supportingROI classCode="ROIBND">
      <code code="ROIPS" />
      <component>
        <boundary classCode="OBS">
          <code code="TIME_RELATIVE" />
          <value xsi:type="PQ">
            <low value="345" unit="ms" />
          </value>
        </boundary>
      </component>
    </supportingROI>
  </support>
</annotation>
```

Challenge

Some studies use continuous ECG recording devices (e.g. Holter or telemetry) to capture a subject's ECG. The duration of the recordings will vary, but FDA expects to receive studies that use 30 and 48 hour recordings. Looking at the extreme case of a 12-lead, 1000 samples-per-second, 48 hour recording, the ECG will have:

$$\begin{array}{l} 1000 \text{ (samples/sec)} \\ \times 60 \text{ (sec/min)} \\ \times 60 \text{ (min/hr)} \\ \times 48 \text{ (hr/lead)} \\ \times 12 \text{ (lead/recording)} \\ \hline = 2,073,600 \text{ (samples/recording) values} \\ \text{(This is 17,280 times more samples than the typical 10-second recording.)} \end{array}$$

Because it is impractical for a person to review and annotate an entire continuous recording, these recordings are often analyzed by signal processing algorithms. These algorithms may annotate several features of each beat. As an example, assuming 60 beats-per-minute and 5 annotations per beat (P-on, P-off, QRS-on, QRS-off, T-off), a recording will have:

$$\begin{array}{l} 5 \text{ (annotations/beat)} \\ \times 60 \text{ (beats/min)} \\ \times 60 \text{ (min/hr)} \\ \times 48 \text{ (hr/recording)} \\ \hline = 864,000 \text{ (annotations/recording) in 48 hours.} \\ \text{(This is 57,600 times more annotations than the typical "3-beat" analysis of a 10-second recording.)} \end{array}$$

The ~500 kb of disk space used by a typical 10-second aECG XML files is generally split in the following way:

- Rhythm waveforms: ~480 (kb/file) for 12 leads = ~40 (kb/10 second lead) = 96% of file space
- Annotations: ~15 kb for ~20 annotations = ~0.75 (kb/annotation) = 3% of file space
- Structural XML: 8 (kb/file) = 1.6% of file space

The structure XML for waveforms will be the same no matter how long the recording is, so it can essentially be ignored when estimating the size of a 48 hour aECG XML file. A 48 hour ECG recording will therefore take:

$$\begin{array}{l} 480 \text{ (kb/10-sec-file)} \\ \times 17,280 \text{ (10-sec-file/48-hr-file)} \\ \hline = 8,294 \text{ MB of XML waveforms} \end{array}$$

Plus

0.75 (kb/annotation)
X 864,000 (annotation/48-hr-file)

= 648 MB of XML annotations

For a total of **9 GB** per 48-hour aECG XML file.

Need for Alternate Solution

Many XML parsers read the entire XML structure into memory before making the individual pieces of information available to the hosting application. Although a few of today's computers may have enough processing power and memory to parse 9 GB files in the current aECG XML format, it is unreasonable to expect all ECG labs and drug sponsors to employ such computers for managing their ECG studies.

Removing the waveform samples and annotations from the aECG XML file and storing them into separate binary files will greatly improve the efficiency of XML processing. The aECG XML can still provide all the structure and context for the data, but the actual sample values and annotation locations can be more efficiently encoded as arrays of integers in binary files.

Need for ECG Warehouse v2 Format

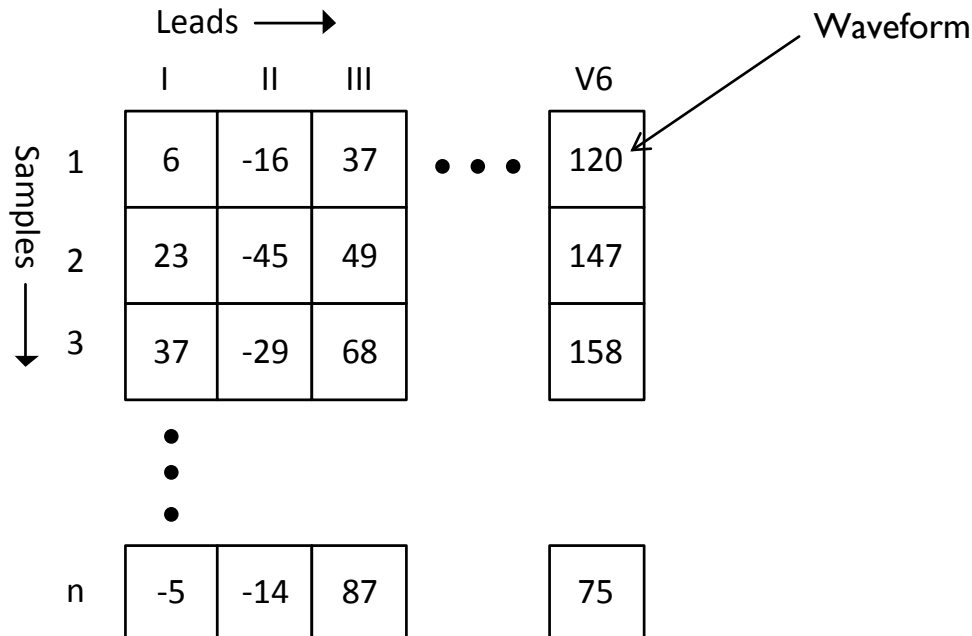
Mortara, working with FDA, updated the ECG Warehouse so it could import, analyze, and display continuous ECG recordings. As part of this project, Mortara added support for external file references in the aECG XML. These changes are minimal, and they support use of existing ECG waveform file formats commonly used for Holter data, including ISHNE. The goal was to make minimal changes to existing software that already supports the aECG format, and to minimize reformatting of waveform data that is already produced by Holter systems.

HL7 Standardization of aECG v2

Because the ECG Warehouse continuous waveform project needed to be completed before HL7 had a chance to standardize changes to aECG, the ECG Warehouse already supports the v2 format described hereafter, and will continue to do so even if HL7 approves a different form. Efforts are currently underway in HL7 to formalize the changes to aECG so it can reference external waveform and annotation files, but that process may take another one to two years. In the meantime, software developers producing data for FDA can move forward with the v2 format already supported by ECG Warehouse.

ECG Waveform File Formats

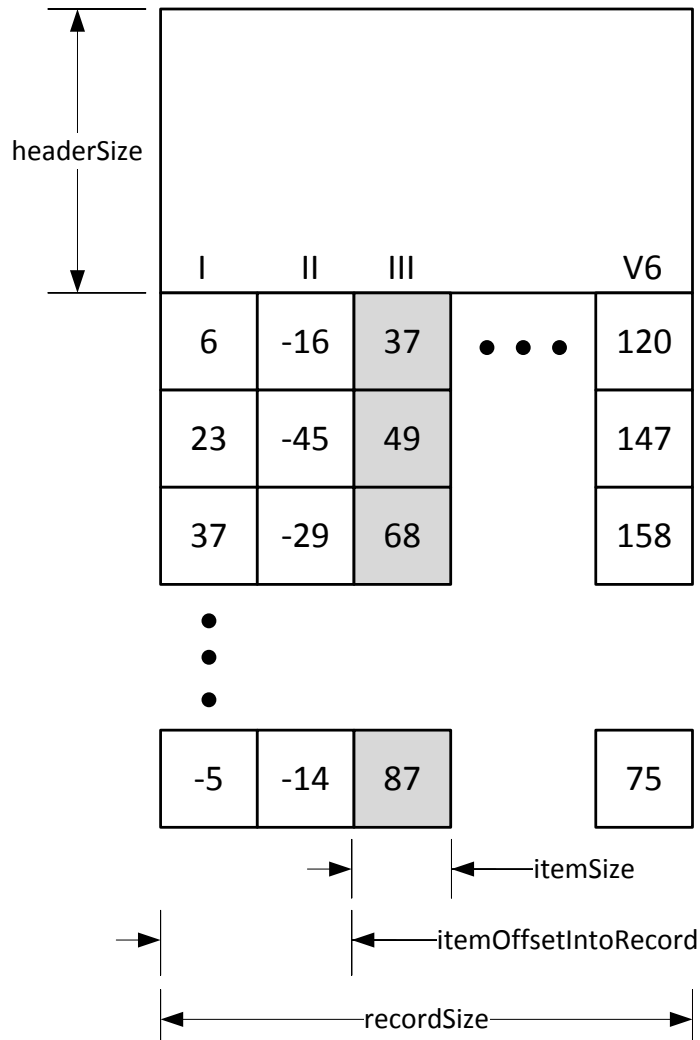
Continuous ECG recording devices simultaneously record all leads. The most common way to save the samples is in an “interleaved” format. That is, the leads are interleaved with one another. All leads for sample 1 are written to the file, and then all leads for sample 2, and so on. In the diagram below, the recording device would write a row at a time. If the device samples the leads 500 times a second, it would write 500 rows to the file every second.



Referencing a Lead in an Interleaved File

The aECG format describes and encodes each lead separately. A change being introduced is to allow a reference to an external file in place of listing all the samples directly in the XML file. Because a lead’s samples are interleaved with the other leads, the external reference must describe each row, or “record”, and the lead’s position within each record.

The diagram below shows how the structure of the file is described so the samples of a lead can be found. There may be a header that precedes the waveform data, so the size of the header must be specified in bytes as “headerSize”. The lead’s sample size is specified in bytes as “itemSize”. The location of the lead’s sample within each record is specified in bytes as “itemOffsetIntoRecord”. And finally, the size of each record is specified in bytes as “recordSize”.



Besides describing the location of the lead's samples within a file, information must be provided on how to interpret the sample values read from the file. ECG samples are always stored as integers along with a real scale factor to convert them into physical quantities, like microvolts. ECG Warehouse v2 supports 2, 4, and 8 byte integers. The integers can be signed (2's complement) or unsigned, and can be stored in the file in big-endian or little-endian format. aECG has always had a place for the DC offset to be subtracted from the values, and the scale factor for conversion into physical quantities. The lead identification, DC offset, and scale factor specifications remain unchanged in the aECG XML.

ECG Waveforms Example

The following example shows how lead I would be referenced in an external file called “waveforms.bin”. Each sample is a 2-byte signed integer stored in little-endian format. The samples need to be scaled by 3.75 to convert them into uV. There are 9 leads in the file, and the maximum integer value “32,767” is used to represent a null (e.g. lead fail). Lead I is the first lead in each record, and the records are 18 bytes long (2 bytes * 9 leads). 43,200,000 records should be read from the file (500 samples-per-second * 24 hrs).

```
<component>
  <sequence classCode="OBS">
    <code code="MDC_ECG_LEAD_I" etc./>
    <value xsi:type="SLIST_PQ">
      <origin value="0" unit="uV"/>
      <scale value="3.75" unit="uV"/>
      <externalFile filePath="waveforms.bin" fileFormat="LE_BINARY"
        itemType="INT" itemSize="2" headerSize="0" recordSize="18"
        itemOffsetIntoRecord="0" recordCount="43200000" nullValue="32767" />
    </value>
  </sequence>
</component>
```

Attribute	Binary Files	Text Files
filePath	Path and name of the data file relative to referencing file (e.g. the aECG XML file).	Path and name of the data file relative to referencing file (e.g. the aECG XML file).
fileFormat	<p>The format of the integer values.</p> <ul style="list-style-type: none"> LE_BINARY = little endian binary integers. BE_BINARY = big endian binary integers. 	<p>The type of text file. Records must be separated by a carriage return (0x0d) and/or line feed (0x0a).</p> <ul style="list-style-type: none"> CSV = comma separated ASCII text values. Note: values cannot have commas in them. TSV = tab separated ASCII text values. Note: values cannot have tabs in them.
itemType	<p>Data type of the item.</p> <ul style="list-style-type: none"> INT = signed integer (2's compliment) UINT = unsigned integer STRING = ASCII encoded string. Must be null-terminated. 	<p>Note: data types are inferred from the annotation type:</p> <ul style="list-style-type: none"> Beat Types are expected to be MDC_ECG_BEAT_? code strings. Physical Quantities are expected to be numbers (integer or real).

Attribute	Binary Files	Text Files
itemSize	For integers, this must be 2, 4, or 8 bytes. For strings, the number of bytes allocated for the null-terminated string value.	
headerSize	The number of bytes to skip over to get to the first record.	The number of records to skip to get to the first record, presumably to skip over header rows and column labels.
recordSize	The number of bytes per record.	The number of items in each record.
itemOffsetIntoRecord	The byte offset of the item within the record.	The number of items to skip over to get to this item. <ul style="list-style-type: none"> • 0 = first item • 1 = second item • Etc.
recordCount	Number of records to read from the file.	Number of records to read from the file.
nullValue	The value used to indicate an integer item has no value (e.g. lead fail). The maximum value is recommended, e.g. 0x7FFF = 32,767 for 2-byte signed integers.	Note: a null value is implied when no value is included (two comma or tab separators in a row).

Beat-by-Beat Annotations in Binary Files

Beat-by-beat annotations can be structured in external files much like ECG waveforms. Instead of representing samples, the records represent beats. Each beat record can include beat annotations such as the beat type, P-onset, P-offset, QRS-onset, QRS-offset, T-offset, etc. Since the external files do not handle binary floating point data, integer timing information should be expressed as relative time in milliseconds from the beginning of the waveform. Absolute times using HL7 time stamps can be encoded as strings in binary files. Beat types must be encoded as strings using the MDC_ECG_BEAT codes. Below is a representation of a binary file with beat records containing beat type codes and several fiducial locations (as relative times from the beginning of the recording).

Annotations →

	Type	P-on	QRS-on	...	T-off
1	MDC_ECG_BEAT_NORMAL	400	457	• • •	920
2	MDC_ECG_BEAT_NORMAL	1356	1402		1894
3	MDC_ECG_BEAT_V_P_C	2238	2303		2517
	•				
	•				
	•				
n	MDC_ECG_BEAT_NORMAL	398722	398774		399158

Beats ↓

ECG Warehouse currently recognizes the following MDC codes for beat types:

MDC_ECG_BEAT_NORMAL	MDC_ECG_BEAT_ABNORMAL
MDC_ECG_BEAT_SV_P_C	MDC_ECG_BEAT_ATR_P_C
MDC_ECG_BEAT_JUNC_P_C	MDC_ECG_BEAT_ATR_P_C_ABERR
MDC_ECG_BEAT_ATR_PWAVE_BLK	MDC_ECG_BEAT_V_P_C
MDC_ECG_BEAT_V_P_C_FUSION	MDC_ECG_BEAT_V_P_C_RonT
MDC_ECG_BEAT_SV_ESC	MDC_ECG_BEAT_ATR_ESC
MDC_ECG_BEAT_JUNC_ESC	MDC_ECG_BEAT_V_ESC
MDC_ECG_BEAT_BB_BLK	MDC_ECG_BEAT_LBB_BLK_COMP
MDC_ECG_BEAT_LBB_BLK_INCOMP	MDC_ECG_BEAT_RBB_BLK_COMP
MDC_ECG_BEAT_RBB_BLK_INCOMP	MDC_ECG_BEAT_BLK_ANT_L_HEMI
MDC_ECG_BEAT_BLK_POS_L_HEMI	MDC_ECG_BEAT_BLK_BIFAS
MDC_ECG_BEAT_BLK_TRFAS	MDC_ECG_BEAT_BLK_BILAT
MDC_ECG_BEAT_BLK_IVCD	MDC_ECG_BEAT_PREX
MDC_ECG_BEAT_WPW_UNK	MDC_ECG_BEAT_WPW_A
MDC_ECG_BEAT_WPW_B	MDC_ECG_BEAT_LGL
MDC_ECG_BEAT_PACED	MDC_ECG_BEAT_PACED_FUS
MDC_ECG_BEAT_UNKNOWN	MDC_ECG_BEAT_LEARN
MDC_ECG_BEAT_TOC	

Beat-by-Beat Annotations in Text Files

In addition to binary files, beat-by-beat annotations can also be encoded as text files in comma- and tab-separated-values formats. The same "record" structure applies, and the records must be separated with the standard CR (0x0d) and/or LF (0x0a) characters. Text files can support fiducial point locations by absolute time in the HL7 "yyyyMMddHHmmss.fff" format, where fff is milliseconds.

Beat-by-Beat Annotation Example

The external file reference can be used in place of the beat type annotations and ROI boundary values. The following example shows how beat-by-beat annotations are referenced in a tab-separated-values text file having 3 columns: beat type, QRS-onset, and T-offset.

```
<annotation classCode="OBS">
  <code codeSystemName="MDC" codeSystem="2.16.840.1.113883.6.24"
    code="MDC_ECG_BEAT"/>
  <value xsi:type="CE_ext_file" codeSystemName="MDC"
    codeSystem="2.16.840.1.113883.6.24">
    <externalFile recordCount="3923" itemOffsetIntoRecord="0" headerSize="3"
      fileFormat="TSV" filePath="BeatLabels.tsv" />
  </value>
  <component contextConductionInd="true" typeCode="COMP" contextControlCode="OP">
    <annotation classCode="OBS">
      <code codeSystemName="MDC" codeSystem="2.16.840.1.113883.6.24"
        code="MDC_ECG_WAVC_TYPE"/>
      <value xsi:type="CE" codeSystemName="MDC"
        codeSystem="2.16.840.1.113883.6.24" code="MDC_ECG_WAVC_QRSTWAVE"/>
      <support typeCode="SPRT">
        <supportingROI classCode="ROIBND">
          <code codeSystemName="ActCode" codeSystem="2.16.840.1.113883.5.4"
            code="ROIPS"/>
          <component typeCode="COMP">
            <boundary classCode="OBS">
              <code codeSystemName="ActCode"
                codeSystem="2.16.840.1.113883.5.4" code="TIME_RELATIVE"/>
              <value xsi:type="IVL_PQ">
                <low unit="ms">
                  <externalFile recordCount="3923"
                    itemOffsetIntoRecord="1" headerSize="3"
                    fileFormat="TSV" filePath="BeatLabels.tsv" />
                </low>
                <high unit="ms">
                  <externalFile recordCount="3923"
                    itemOffsetIntoRecord="2" headerSize="3"
                    fileFormat="TSV" filePath="BeatLabels.tsv" />
                </high>
              </value>
            </boundary>
          </component>
        </supportingROI>
      </support>
    </annotation>
  </component>
</annotation>
```

All Other Waveform Annotations

Although other waveform annotations can technically be encoded in external files, it is recommended that external file annotations be used only for beat-by-beat annotations. Annotation of rhythms, for example, should be done directly in the aECG XML as in the past.

Time Windows

When small windows of time within a longer continuous ECG are used to assess ECG measurements, those ECG waveform “extractions” should not be submitted separately from the original continuous ECG waveforms. Instead, timing windows should be added to the continuous aECG XML file to indicate key protocol measurement locations.

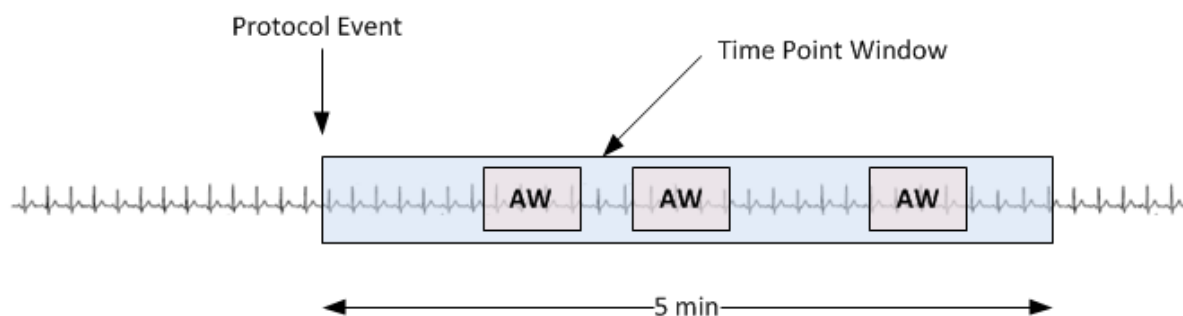
Time Point Windows

Many protocols are designed to assess key ECG measurements at time points relative to other key events, such as a dosage. These protocol time points should be indicated in the aECG XML as “Time Point Windows”. These are windows of time where the protocol allows ECG measurements to be made and associated with that protocol time point. For example, a Time Point Window could be five minutes in duration, located 30 minutes after a dosage was given.

Analysis Windows

Many ECG measurement tools are designed to operate on short periods of ECG waveforms, like ten seconds. It is common to form a representative beat from the ten-second window and make careful timing measurements with it. These short windows of time should be indicated as “Analysis Windows” within the larger Time Point Windows. These Analysis Windows would, for example, indicate the locations of the classic “ten-second extractions” already used in thorough QT studies.

The diagram below shows the location of a protocol event, like 30 minutes post-dose. It shows a five-minute Time Point Window where measurements for this protocol event can be made. Within the Time Point Window are three Analysis Windows where careful ECG timing measurements were made.

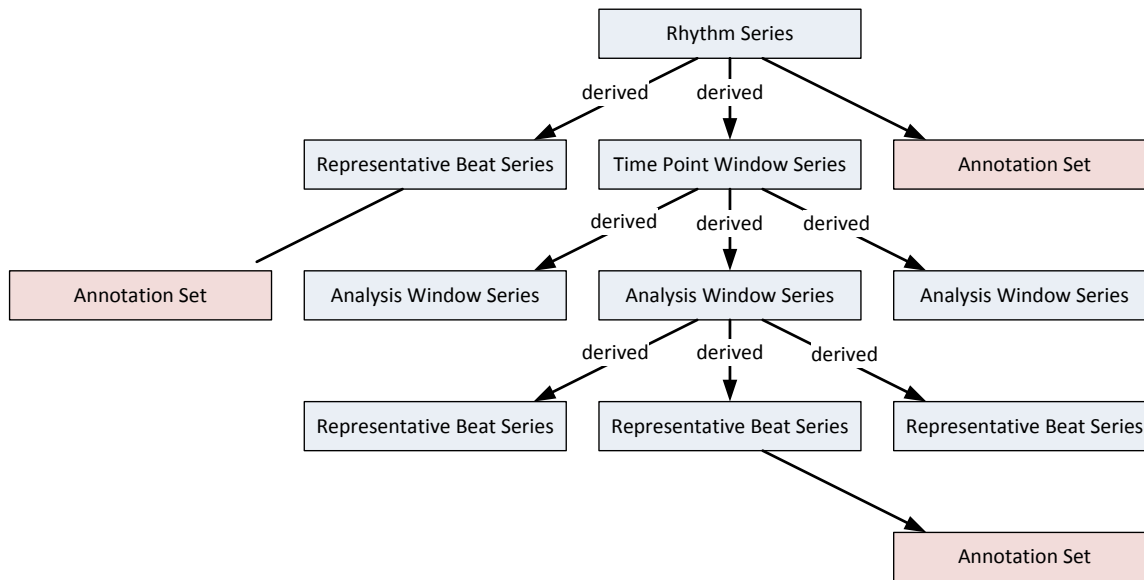


XML Structure for Windows and Annotations

aECG XML supports a Series type RHYTHM for the waveforms directly recorded from the subject. It also supports a derived Series type of REPRESENTATIVE_BEAT for waveforms algorithmically derived from several beats of the parent Rhythm Series. ANALYSIS_WINDOW and TIME_POINT_WINDOW are also derived Series types.

For continuous aECG XML files, ECG Warehouse expects to receive Time Point Windows as derived series from the Rhythm series. Analysis Windows are expected to be derived from Time Point Windows. A Representative Beat can be derived from an Analysis Window.

ECG Warehouse only expects to find waveforms within a Rhythm Series or a Representative Beat Series. ECG Warehouse also expects to find annotations attached to a Rhythm Series or a Representative Beat.



Time Point and Analysis Windows Example

The example below shows a Rhythm Series with its waveforms in an external file. This Series has an Annotation Set with beat-by-beat annotations in another external file. The beat-by-beat annotations include a beat label and QRS-onset.

The Rhythm Series has a derived Time Point Window Series. The Time Point Window has a derived Analysis Window Series. The Analysis Window has a derived Representative Beat, and its waveforms are directly included in the XML. The Representative Beat has annotations for a QT measurement and the QRST onset and offset locations.

```
<series classCode="OBSSER" moodCode="EVN">
  <id root="37D85819-C1ED-415D-89D0-2ACCF8D48E22" />
  <code code="RHYTHM" codeSystem="2.16.840.1.113883.5.4" codeSystemName="ActCode" />
  <effectiveTime>
    <low value="20011219074300.000" inclusive="true" />
    <high value="20011220072100.000" inclusive="false" />
  </effectiveTime>
  <component typeCode="COMP" contextConductionInd="true">
    <sequenceSet classCode="OBSCOR" moodCode="EVN">
      <component typeCode="COMP" contextConductionInd="true">
        <sequence classCode="OBS" moodCode="EVN">
          <code code="TIME_ABSOLUTE" codeSystem="2.16.840.1.113883.5.4"
            codeSystemName="ActCode" />
          <value xsi:type="GLIST_TS">
            <head value="20011219074300.000" />
            <increment value="0.002" unit="s" />
          </value>
        </sequence>
      </component>
      <component typeCode="COMP" contextConductionInd="true">
        <sequence classCode="OBS" moodCode="EVN">
          <code code="MDC_ECG_LEAD_I" codeSystem="2.16.840.1.113883.6.24"
            codeSystemName="MDC" />
          <value xsi:type="SLIST_PQ">
            <origin value="0.000" unit="uV" />
            <scale value="6.25" unit="uV" />
            <externalFile filePath="leads.ecg" fileFormat="LE_BINARY" itemType="INT"
              itemSize="2" headerSize="522" recordSize="18" itemOffsetIntoRecord="0"
              recordCount="42480000" nullValue="-32768" />
          </value>
        </sequence>
      </component>
      <component typeCode="COMP" contextConductionInd="true">
        <sequence classCode="OBS" moodCode="EVN">
          <code code="MDC_ECG_LEAD_II" codeSystem="2.16.840.1.113883.6.24"
            codeSystemName="MDC" />
          <value xsi:type="SLIST_PQ">
            <origin value="0.000" unit="uV" />
            <scale value="6.25" unit="uV" />
            <externalFile filePath="leads.ecg" fileFormat="LE_BINARY" itemType="INT"
              itemSize="2" headerSize="522" recordSize="18" itemOffsetIntoRecord="2"
              recordCount="42480000" nullValue="-32768" />
          </value>
        </sequence>
      </component>
    </sequenceSet>
    <!-- more leads -->
  </component>
</series>
```



```

<subjectOf typeCode="SUBJ">
  <annotationSet classCode="OBS" moodCode="EVN">
    <component typeCode="COMP">
      <annotation classCode="OBS" moodCode="EVN">
        <code code="MDC_ECG_BEAT" codeSystem="2.16.840.1.113883.6.24"
codeSystemName="MDC" />
        <value xsi:type="CE_ext_file" codeSystem="2.16.840.1.113883.6.24"
codeSystemName="MDC">
          <externalFile filePath="BeatLabels.txt" fileFormat="TSV" headerSize="3"
recordSize="2" itemOffsetIntoRecord="1" recordCount="0" />
        </value>
        <component typeCode="COMP" contextControlCode="OP"
contextConductionInd="true">
          <annotation classCode="OBS" moodCode="EVN">
            <code code="MDC_ECG_WAVC_TYPE" codeSystem="2.16.840.1.113883.6.24"
codeSystemName="MDC" />
            <value xsi:type="CE" code="MDC_ECG_WAVC_QWAVE"
codeSystem="2.16.840.1.113883.6.24" codeSystemName="MDC" />
            <support typeCode="SPRT">
              <supportingROI classCode="ROIBND" moodCode="EVN">
                <code code="ROIPS" codeSystem="2.16.840.1.113883.5.4"
codeSystemName="ActCode" />
                <component typeCode="COMP">
                  <boundary classCode="OBS" moodCode="EVN">
                    <code code="TIME_RELATIVE" codeSystem="2.16.840.1.113883.5.4"
codeSystemName="ActCode" />
                    <value xsi:type="IVL_PQ">
                      <low unit="ms">
                        <externalFile filePath="BeatLabels.txt" fileFormat="TSV"
headerSize="3" recordSize="2" itemOffsetIntoRecord="0"
recordCount="0" />
                      </low>
                    </value>
                  </boundary>
                </component>
              </supportingROI>
            </support>
          </annotation>
        </component>
      </annotationSet>
    </subjectOf>
    <derivation typeCode="DRIV" contextControlCode="OP" contextConductionInd="true">
      <derivedSeries classCode="OBSER" moodCode="EVN">
        <id root="91BE2AF2-E08D-4E6F-8CAF-7F5ACCB106AC" />
        <code code="TIME_POINT_WINDOW" codeSystem="2.16.840.1.113883.5.4"
codeSystemName="ActCode" />
        <effectiveTime>
          <low value="20011219074330.000" inclusive="true" />
          <high value="20011219074730.000" inclusive="false" />
        </effectiveTime>
        <component typeCode="COMP" contextConductionInd="true">
          <sequenceSet classCode="OBSCOR" moodCode="EVN">
            <component typeCode="COMP" contextConductionInd="true">
              <sequence classCode="OBS" moodCode="EVN">
                <code />
                <value xsi:type="ST" />
              </sequence>
            </component>
          </sequenceSet>
        </component>
      </derivedSeries>
    </derivation>
  </subjectOf>

```

```

<derivation typeCode="DRIV" contextControlCode="OP" contextConductionInd="true">
  <derivedSeries classCode="OBSSER" moodCode="EVN">
    <id root="2B822245-003F-44FF-AA3F-CCEA786EB124" />
    <code code="ANALYSIS_WINDOW" codeSystem="2.16.840.1.113883.5.4"
    codeSystemName="ActCode" displayName="-15min" />
    <effectiveTime>
      <low value="20011219074700.000" inclusive="true" />
      <high value="20011219074710.000" inclusive="false" />
    </effectiveTime>
    <component typeCode="COMP" contextConductionInd="true">
      <sequenceSet classCode="OBSCOR" moodCode="EVN">
        <component typeCode="COMP" contextConductionInd="true">
          <sequence classCode="OBS" moodCode="EVN">
            <code />
            <value xsi:type="ST" />
          </sequence>
        </component>
      </sequenceSet>
    </component>
  </derivedSeries>
  <derivation typeCode="DRIV" contextControlCode="OP"
  contextConductionInd="true">
    <derivedSeries classCode="OBSSER" moodCode="EVN">
      <id root="0F675158-C35F-450C-9172-971D8B0CDDF9" />
      <code code="REPRESENTATIVE_BEAT" codeSystem="2.16.840.1.113883.5.4"
      codeSystemName="ActCode" displayName="ECG representative beat" />
      <effectiveTime>
        <low value="20011219074700.000" inclusive="true" />
        <high value="20011219074710.000" inclusive="false" />
      </effectiveTime>
      <component typeCode="COMP" contextConductionInd="true">
        <sequenceSet classCode="OBSCOR" moodCode="EVN">
          <component typeCode="COMP" contextConductionInd="true">
            <sequence classCode="OBS" moodCode="EVN">
              <code code="TIME_RELATIVE" codeSystem="2.16.840.1.113883.5.4"
              codeSystemName="ActCode" displayName="Relative Time Sequence" />
              <value xsi:type="GLIST_PQ">
                <head value="0.000" unit="s" />
                <increment value="0.002" unit="s" />
              </value>
            </sequence>
          </component>
          <component typeCode="COMP" contextConductionInd="true">
            <sequence classCode="OBS" moodCode="EVN">
              <code code="MDC_ECG_LEAD_I" codeSystem="2.16.840.1.113883.6.24"
              codeSystemName="MDC" displayName="I" />
              <value xsi:type="SLIST_PQ">
                <origin value="0" unit="uV" />
                <scale value="6.25" unit="uV" />
                <digits>4 4 4 4 3 3 3 etc.</digits>
              </value>
            </sequence>
          </component>
          <component typeCode="COMP" contextConductionInd="true">
            <sequence classCode="OBS" moodCode="EVN">
              <code code="MDC_ECG_LEAD_II" codeSystem="2.16.840.1.113883.6.24"
              codeSystemName="MDC" displayName="II" />
              <value xsi:type="SLIST_PQ">
                <origin value="0" unit="uV" />
                <scale value="6.25" unit="uV" />
                <digits>6 6 6 6 6 5 5 etc.</digits>
              </value>
            </sequence>
          </component>
        </sequenceSet>
      </component>
    </derivedSeries>
  </derivation>

```

```

        </component>
        <!-- more leads -->
    </sequenceSet>
</component>
<subjectOf typeCode="SUBJ">
    <annotationSet classCode="OBS" moodCode="EVN">
        <component typeCode="COMP">
            <annotation classCode="OBS" moodCode="EVN">
                <code code="MDC_ECG_TIME_PD_QT"
                    codeSystem="2.16.840.1.113883.6.24" codeSystemName="MDC" />
                <value xsi:type="PQ" value="402" unit="ms" />
                <component typeCode="COMP" contextControlCode="OP"
                    contextConductionInd="true">
                    <annotation classCode="OBS" moodCode="EVN">
                        <code code="MDC_ECG_WAVC_TYPE"
                            codeSystem="2.16.840.1.113883.6.24" codeSystemName="MDC" />
                        <value xsi:type="CE" code="MDC_ECG_WAVC_QRSTWAVE"
                            codeSystem="2.16.840.1.113883.6.24" codeSystemName="MDC" />
                        <support typeCode="SPRT">
                            <supportingROI classCode="ROIBND" moodCode="EVN">
                                <code code="ROIPS" codeSystem="2.16.840.1.113883.5.4"
                                    codeSystemName="ActCode" />
                                <component typeCode="COMP">
                                    <boundary classCode="OBS" moodCode="EVN">
                                        <code code="TIME_RELATIVE"
                                            codeSystem="2.16.840.1.113883.5.4"
                                            codeSystemName="ActCode" />
                                        <value xsi:type="IVL_PQ">
                                            <low value="356" unit="ms" />
                                            <high value="758" unit="ms" />
                                        </value>
                                    </boundary>
                                </component>
                            </supportingROI>
                        </support>
                    </annotation> <!-- QRST Waveform Component -->
                </component>
            </annotation> <!-- QT Measurement -->
        </component>
    </annotationSet>
</subjectOf>
</derivedSeries>
</derivation> <!-- Representative Beat -->
</derivedSeries>
</derivation> <!-- Analysis Window -->
</derivedSeries>
</derivation> <!-- Time Point Window -->
</series> <!-- Rhythm Series -->

```

Baseline Windows

ECG studies assess the effects of drugs on the subjects' cardiac cycle. A way to study those effects is to make measurements while the subject is on drug, and compare those measurements to ones taken off drug. The off-drug measurements are often called "baseline" measurements.

The ECG Warehouse has features for comparing on-drug Analysis Windows to baseline Analysis Windows. The on-drug Analysis Window can reference its baseline using the baseline window's unique ID (UID). ECG Warehouse interprets the root portion of the window's ID as the window's UID, and it interprets the extension portion of the window's ID as a reference to the baseline window's UID.

Here is an example of a baseline Analysis Window. It's UID is 2B822245-003F-44FF-AA3F-CCEA786EB124. Any Analysis Window that's compared to this baseline will put this baseline window's UID into its ID extension.

```
<derivedSeries classCode="OBSSER" moodCode="EVN">
  <id root="2B822245-003F-44FF-AA3F-CCEA786EB124" />
  <code code="ANALYSIS_WINDOW" codeSystem="2.16.840.1.113883.5.4"
    codeSystemName="ActCode" displayName="-15min" />
```

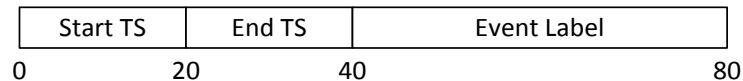
Here is an example of an Analysis Window that references the previous Analysis Window as its baseline.

```
<derivedSeries classCode="OBSSER" moodCode="EVN">
  <id root="3E822246-113E-44FF-AA3F-DDFB897FC235" extension="2B822245-003F-44FF-AA3F-
  CCEA786EB124" />
  <code code="ANALYSIS_WINDOW" codeSystem="2.16.840.1.113883.5.4"
    codeSystemName="ActCode" displayName="-15min" />
```

Protocol Event Annotations

Since continuous ECG waveforms can span a long period of time, it can be helpful to give FDA reviewers some context. ECG Warehouse can display protocol event annotations when displaying long-term trends. The event annotations can be key events like when a meal was eaten, a dosage given, blood drawn, etc. Currently ECG Warehouse can only import protocol event annotations from external files.

Below is an example of how protocol event annotations are referenced in an external file. The protocol event labels are 40-character strings, 40 bytes into each record. The label location start time stamps are found at the beginning of each record, and the label location ending time stamps are found 20 bytes into each record.



```
<component>
  <annotation classCode="OBS">
    <code code="MDC_ECG_PROTOCOL_EVENT" codeSystem="2.16.840.1.113883.6.24"
      codeSystemName="ProtocolEvent"/>
    <value xsi:type="ST_ext_file">
      <externalFile filePath="Annotations.dat" fileType="LE_BINARY" itemType="STRING"
        itemSize="40" recordSize="80" itemOffsetIntoRecord="40" recordCount="22"
        nullValue="" />
    </value>
    <support typeCode="SPRT">
      <supportingROI classCode="ROIBND">
        <code code="ROIPS" codeSystem="2.16.840.1.113883.5.4" codeSystemName="HL7V3"/>
        <component>
          <boundary classCode="OBS">
            <code code="TIME_ABSOLUTE" codeSystem="2.16.840.1.113883.5.4"
              codeSystemName="ActCode"/>
            <value xsi:type="IVL_TS">
              <low>
                <externalFile filePath="Annotations.dat" fileType="LE_BINARY"
                  itemType="STRING" itemSize="20" recordSize="80"
                  itemOffsetIntoRecord="0" recordCount="22" nullValue="" />
              </low>
              <high>
                <externalFile filePath="Annotations.dat" fileType="LE_BINARY"
                  itemType="STRING" itemSize="20" recordSize="80"
                  itemOffsetIntoRecord="20" recordCount="22" nullValue="" />
              </high>
            </value>
          </boundary>
        </component>
      </supportingROI>
    </support>
  </annotation>
</component>
```

Below is a partial screen shot from ECG Warehouse that shows some “Dominated Rhythm Change” events detected by a Holter analysis algorithm. Other protocol events, like dosages, would also be displayed as tick marks on the X-axis.

