

## IEcgAccess Open Interface for the EcgViewer Version 1.2

Implementing the open interface enables a proprietary or standard based ECG data to be viewed by the EcgViewer.

A sample solution for binary ECG data is provided in the "OEMEcg" sample solution. Output of the project is "OemLib.dll" .

After implementing your own solution, go to EcgViewer setup menu and select General->ECG Formats->Use OemLib.dll option.

Below is the C# code for the open interface IEcgAccess:

```
/// <summary>
/// IEcgAccess Interface to implement read/write API
/// </summary>
public interface IEcgAccess
{
    /// <summary>
    /// Reads entire record at the given path and returns true if
    /// successfull
    /// </summary>
    bool Read(string fileName);

    /// <summary>
    /// Writes entire record to the given path and returns true if
    /// successfull
    /// </summary>
    bool Write(string fileName);

    /// <summary>
    /// Patient related info is grouped here
    /// </summary>
    Person PatientInfo
    {
        get;
        set;
    }

    /// <summary>
    /// ECG related info, excluding ecg data, is grouped here
    /// </summary>
    SignalInfo EcgInfo
    {
        get;
        set;
    }
}
```

```

    /// <summary>
    /// Diagnose related info is grouped here
    /// </summary>
    Diagnostic DiagnoseInfo
    {
        get;
        set;
    }

    /// <summary> gets or sets ECG data as specified by IEcgAccess
    /// interface
    /// </summary>
    ArrayList EcgData
    {
        get;
        set;
    }
}

/// <summary>
/// Simple DateTime structure
/// </summary>
public struct DateAndTime
{
    public int Year;
    public int Month;
    public int Day;
    public int Hour;
    public int Minute;
    public int Second;
}

/// <summary>
/// Personal Details of the patient.
/// </summary>
public struct Person
{
    /// <summary>Patient Identification Number, reasonable length
    /// 40 characters
    /// </summary>
    public string ID;

    /// <summary>First Name </summary>
    public string FirstName;

    /// <summary>Second Last Name </summary>
    public string SecondLastName;

    /// <summary>Last Name </summary>
    public string LastName;

    /// <summary>
    /// 0.Not known 1.Male, 2.Female, 9.Unspecified
    /// </summary>
    public int Sex;
}

```

```

    /// <summary>
    /// 0.Unspecified, 1.Caucasian, 2.Black, 3.Oriental
    /// </summary>
    public byte Race;

    /// <summary> Weight in specified units </summary>
    public int Weight;

    /// <summary>
    /// 0.Unspecified, 1.Kilogram, 2.Gram, 3.Pound, 4.Ounce
    /// </summary>
    public byte WeightUnit;
    /// <summary> Height in specified units </summary>
    public int Height;

    /// <summary>
    /// 0.Unspecified, 1.Centimeters, 2.Inches, 3.Millimeters
    /// </summary>
    public byte HeightUnit;

    /// <summary>
    /// Date of birth specified in DateAndTime structure
    /// </summary>
    public DateAndTime Birth;

    /// <summary> Age in years </summary>
    public int Age;
}

/// <summary>
/// Details related to ECG signal characteristics and data
/// acquisition
/// </summary>

public struct SignalInfo
{
    /// <summary>
    /// Enter Lead names EXACTLY as described in Lead
    /// Identification Codes table.
    /// Do not put extra spaces or characters in between.
    /// Example: "I,II,III,aVR,aVL,aVF,V1,V2,V3,V4,V5,V6"
    /// </summary>
    public string Leads;

    /// <summary>
    /// If all lead recording lengths are not same, create and
    /// set this array to specify lead lengths in samples.
    /// If same, no need to use this field.
    /// </summary>
    public uint[] LeadLengths;

    /// <summary>
    /// A/D value corresponding to 1 mv at the input
    /// </summary>
    public double ADU;

    /// <summary> Sampling rate in samples per second </summary>
    public double SRATE;
}

```

```

/// <summary> Date and Time of Acquisition in DateAndTime
/// structure
/// </summary>
public DateAndTime Acq;

/// <summary>
/// "cut-off" frequency (-3 db) of the high pass baseline
/// filter in Hz
/// </summary>
public double BaseLineFilter

/// <summary>
/// "cut-off" frequency (-3db) of the low pass filter in Hz
/// </summary>
public double LowPassFilter;

/// <summary>
/// if other filters, which were not defined above, indicated
/// here.
/// Bit 0:60 Hertz notch filter, Bit 1:50 Hertz notch filter
/// Bit 2:Artifact filter Bit 3:Baseline filter (e.g.
/// adaptive filter or spline filter)
/// </summary>
public byte FilterBitMap;

/// <summary>
/// Text model description. Up to 5 bytes of text
/// </summary>
public string TextModel;

/// <summary>
/// Language Support Code (one byte). This bit map indicates
/// the supported character sets.
/// Refer to SCP-ECG standart for the explanation of codes
/// </summary>
public byte LanguageCode;

/// <summary>
/// Capabilities of the ECG Device (one byte bit map).
/// Bit 0-3 reserved, 4 printing, 5 analysis, 6 storage, 7
/// acquisition
/// </summary>
public byte CapabilitiesBitMap;

/// <summary>
/// Specifies frequency of the AC Mains
/// 0: unspecified, 1: 50Hz, 2: 60Hz
/// </summary>
public byte ACMainsFrequency;

/// <summary> Serial Number of Acquisition Device </summary>
public string DeviceSN;

/// <summary> Acquisition device SCP implementation software
/// identifier (maximum 24 characters) </summary>
public string DeviceSCP;

```

```

    /// <summary> Manufacturer of Acquisition Device </summary>
    public string DeviceMF;
}

/// <summary>
/// Details related to Diagnosis and Medical history
/// </summary>

public struct Diagnostic
{
    /// <summary> Systolic blood pressure in mmHg </summary>
    public ushort Systolic;

    /// <summary> Diastolic blood pressure in mmHg </summary>
    public ushort Diastolic;

    /// <summary> Acquiring Institution Description </summary>
    public string AcquiringInstitute;

    /// <summary> Referring Physician </summary>
    public string ReferringPhysician;

    /// <summary> Diagnosis or the referral indication</summary>
    public string Diagnosis;

    /// <summary> Medical History</summary>
    public string MedicalHistory;

    /// <summary> Remarks and free text</summary>
    public string FreeText;
}

```