



Noah 4

Programmers Reference

Noah 4 Audiogram Chart Control

Version 1.0, 8. December 2021
Written by the HIMSA Development Group

The information contained in this document is subject to change without notice.

HIMSA MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OR SUITABILITY FOR A PARTICULAR PURPOSE. HIMSA shall not be liable for errors contained herein or for incidental consequential damages in connection with the supply of, performance of, or use of this material.

This document contains proprietary information which is protected by copyright. All rights are reserved. No parts of this document may be photocopied, reproduced or translated to another language without the prior permission of HIMSA.

Copyright © 2021 HIMSA

REVISION HISTORY

Version	Status	Author	Date	Change (describe)
0.1	Draft	PFH	11/11/13	Initial document
1.0	Released	SH	8/12/21	Updated to handle e.g. format 502

Contents

1. Introduction.....	4
1.1 Purpose	4
2. The Audiogram Chart control.	5
2.1 The Control.	5
2.2 Setting the Audiogram to display in the chart	5
2.2.1 Properties:	6
2.3 Controlling which puretone measurements definitions to use.	8
3. Sample.....	9
4. Deployment.....	10
5. Glossary	11

1. Introduction

This document should describe the properties and use of the Noah 4 Audiogram chart control.

A sample usage of the Chart can be found in the Sample Business System.

The control is independent of the Module or Business System API.

1.1 Purpose

The purpose of this document is to enable the user to use the Audiogram Chart Control in their Module or Business System.

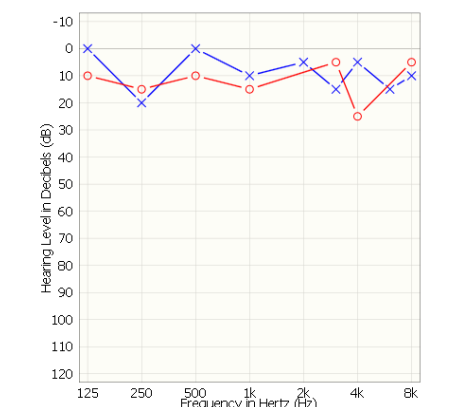
2. The Audiogram Chart control.

The control is a WPF usercontrol that can display a number of defined PureTone measurements as a chart. The PureTone measurements are predefined by HIMSA, and the display (shown measurements and symbols) depends on the RegionCode (LCID).

The layout of the chart control can be controlled in a number of ways. They are controlled via dependency properties on the control.

2.1 The Control.

The AudiogramChartControl is located in the assembly NoahExternalComponents.dll, in the namespace Himsa.Noah.ExternalComponents.



2.2 Setting the Audiogram to display in the chart

In order to display an audiogram chart the control needs an object of type TAudioSession200.

The action is either an Action object from the BusinessAPI or from the ModuleAPI.

```
TAudioSession200 audiogram = TAudioSession200.Create(
    action.DataType.Code,
    action.DataType.Format,
    action.PublicData);
```

The audiogram can now be set to the controls AudiogramAction property either via binding or through “code”.

NOTE: TAudioSession supports format 100, 200 and 500. If the format is e.g. 502 use ConvertData on the Action object.

2.2.1 Properties:

The chart control is primary controlled by a number of dependency properties.

2.2.1.1 DependencyProperties

Name	Type (default value)	Description
AudiogramAction	object (null)	Currently this must be an object of type TAudioSession200
AudiogramShadowAction	object (null)	Currently this must be an object of type TAudioSession200
AutoScaleSymbols	bool (false)	If the symbols should scale with the chart.
AutoSizeOnWidthHeight	bool (true)	Chart drawing option
BaseScaleSymbols	double (1.0)	The base scale of the symbols.
DrawLinesToSymbolCenter	bool (false)	How to draw the lines between the symbols.
GridContentBackgroundBrush	Brush (Brushes.PaleGoldenrod)	Color of the chart contents background.
GridLineBrush	Brush (Brushes.LightGray)	The Brush for the grid lines.
GridLineThickness	double (0.5)	The thickness of the gridlines.
LanguageCode	int (0)	Controlling the texts. (LCID)
LeftAlignYAxisValues	bool (false)	If the Y axis text should be left aligned.
MaxXValue	int (8000)	(8000,16000) Hertz
MaxYValue	int (120)	(120, 130) dB
RegionCode	int (0)	Controlling the selection of Audiogram measurements tests and symbols. (LCID)
SetXAxisLocation	XAxisLocation (XAxisLocation.Bottom)	The location on the X axis texts. See 2.2.1.3
ShowEar	Ear (Ear.None)	Which Ear to show on the chart (None means Both left and right) See 2.2.1.3
ShowGridBorder	bool (true)	If chart should draw the border rectangle.
ShowToneXYLabels	bool (true)	If the lables/axis texts should be displayed.
ToneUnderlayTemplate	ToneUnderlayTemplateType (ToneUnderlayTemplateType.No ne)	Controls which underlay to display. See 2.2.1.3
Use130DbIfPresent	bool (true)	If MaxYValue is 120 and a measurement has an intensity of 130db, 130 will be used.

UseHighFreqIfPresent	bool (true)	If MaxXValue is set to 8000 and a measurement has Frequency above 8KHz, the max value will change to match the measurement frequency.
UseNoAudiogramMessage	bool (false)	If “No Audiogram” text should be displayed, if AudiogramAction object is null.
Ystep	int (10)	The dB stepping (5 or 10dB)

2.2.1.2 Other properties

Name	Type	Description
XOffset	int (20)	Offsets the chart left side.
YOffset	int (15)	Offsets the chart top.
XGridSpacing	int (32)	Spacing between the vertical grid lines.
YGridSpacing	int (8)	Spacing between the horizontal lines.
XGridSnippet	int (5)	Spacing between text and line.
YGridSnippet	int (5)	Spacing between text and line.

Other properties than those described above should be used!

2.2.1.3 Enum / Constant definitions

Location of the XAxes labels:

```
public enum XAxisLocation {
    None,
    Top,
    Bottom
}
```

Which underlay to display in the chart:

```
public enum ToneUnderlayTemplateType {
    None,
    SeverityOfLoss,
    SpeechSpectrum,
    AudibilityIndex
}
```

For which Ear to show measurements:

```
public enum Ear {
```

```
None,          (Show left and right Ear)
Left,
Right,
Binaural
};
```

2.3 Controlling which puretone measurements definitions to use.

Before instantiating the control, it can be controlled which Audiogram profile configuration to use. The profile controls the measurements and symbols set. The profile is embedded in the NoahAudiogramProfile.dll and LocalNoahAudiogramProfile.dll. It is controlled via the property:

```
Himsa.Noah.ExternalComponent.CommonConfig.UseSharedProfile bool (true)
```

If UseSharedProfile is true, the configuration loader will try to load the configuration from the NoahAudiogramProfile.dll places in the GAC.

If the configuration is not found or UseSharedProfile is false, the configuration loader will try to load the LocalNoahAudiogramProfile.dll in the NoahExternalComponents.dll location.

3. Sample

XAML code snippet:

```
<audi:AudiogramChartControl
  UseNoAudiogramMessage="True"
    RegionCode="1033"
    LanguageCode="1033"
  AudiogramAction="{Binding Path=LatestAudiogramActionTAudioSession200}">
</audi:AudiogramChartControl>
```

C# code snippet:

```
private TAudioSession200 _latestTAudioSession200 = null;
public TAudioSession200 LatestAudiogramActionTAudioSession200 {
    get {
        return _latestTAudioSession200;
    }
    set {
        if (_latestTAudioSession200 != value)
            _latestTAudioSession200 = value;
        OnPropertyChanged("LatestAudiogramActionTAudioSession200");
    }
}

// Find latest audiogram and set the property
private const int AudiogramDataTypeCode = 1;
public void SetLatestAudiogram() {
    Himsa.Noah.BusinessAPI.Action latestaction = null;
    if (_businessApi.CurrentPatient == null) return;
    for (int sessionidx = _businessApi.CurrentPatient.Sessions.Count; sessionidx > 0;
sessionidx--) {
        var session = _businessApi.CurrentPatient.Sessions[sessionidx-1];
        for (int actionidx = session.Actions.Count; actionidx > 0; actionidx--) {
            var action = session.Actions[actionidx-1];
            if (action.DataType.Code == AudiogramDataTypeCode) {
                // Chart supports 100,200 and 500
                // If other format it should be Converted
                action.ConvertData(500, latestaction);
                break;
            }
        }
        if (latestaction != null)
            break;
    }
    if (latestaction != null) {
        LatestAudiogramActionTAudioSession200 = TAudioSession200.Create(latestaction.DataType.Code,
latestaction.DataType.Format, latestaction.PublicData);
    }
}
}
```

4. Deployment

The Audiogram Chart control uses the following components

- NoahExternalComponents.dll
- LocalNoahAudiogramProfile.dll
- NoahAudiogramProfile.dll
- Log4net.dll

The assemblies NoahExternalComponents.dll and LocalNoahAudiogramProfile.dll are to be installed by the application, using private deployment, and this means that they must NOT be installed in the GAC.

The assemblies NoahAudiogramProfile.dll and Log4net.dll are installed by Noah 4, and are both installed in the GAC. These components must only be installed by Noah. These are shared components.

5. Glossary

Term/acronym	Description
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
WPF	Windows Presentation Foundation
MS	<u>M</u> icrosoft
N/A	<u>N</u> ot <u>A</u> vailable/ <u>N</u> ot <u>A</u> pplicable
SW	Software