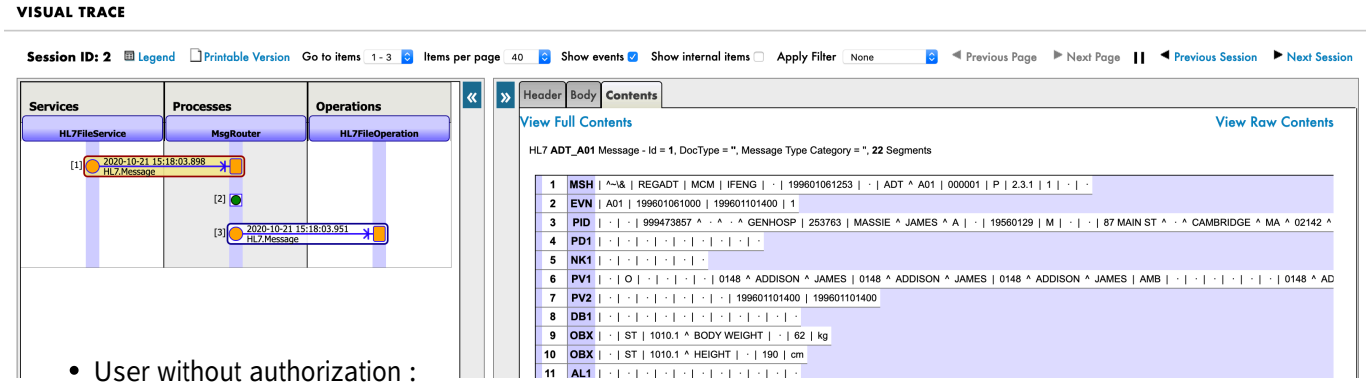
Article
Guillaume Rongier · Oct 23, 2020 5m read

# HL7 PID Obfuscation

The aim of my question is to achieve this :

- User with authorization :




- User without authorization :

You can notice that some information of the PID segment are obfuscated.

I do have a solution but it involve a modification of EnsLib.HL7.Message. **It's not a best practice**.

Do you have idea or a simple solution that doesn't involve overloading every HL7 classes ?

Here is my solution :

https://github.com/grongierisc/hl7-msg-anonymizer

# HL7-Msg-Anonymizer

This is a modification of EnsLib.HL7.Message to not display Patient demography information in visual trace.

This feature is toggle by a resource named NotAnonymize, if a user have this resource in USE, then he can see patient demography otherwise not.

# Remarques

First of all, this POC was build on IRIS for Health 2020.3 and it may not work on Ensemble.

In addition, this POC is not supported by InterSystems and is not intended to go into production.

This is not an overload of the EnsLib.HL7.Message class because if we create an overload of this class, the overloaded class could no longer pass through the HL7 routers.
Moreover, with an overloaded class we would need to recreate all the HL7 connectors (TCP, File, FTP, etc).

# How to install

Make a backup of EnsLib.HL7.Message.

The ENSLIB database must be mounted with write, to do so, uncheck realonly here :

## For Ensemble

Replace EnsLib.HL7.Message::OutputHTML() and EnsLib.HL7.Message::OutputHTMLZen() by the code in how it works.

## Form Iris for health

Import EnsLib.HL7.Message and the package EnsLib.HL7.Util in this git.

## End install

Dont forget to put back EnsLib in readonly mode

# How it's build

The project has been constructed in such a way that it is as least intrusive as possible in the original code.

This makes it possible to take up this idea and eventually apply it to Ensemble.

# How it works

The display of HL7 objects is done with the classes: EnsLib.HL7.Util.FormatHTMLv2 and EnsLib.HL7.Util.FormatHTMLv2Zen. The objective is to modify the behaviour of EnsLib.HL7.Message to use our modified classes to obfuscate PID segments.

This is done by this modification in EnsLib.HL7.Message :

```
/// Display Segments as HTML, using DocType info if available
Method OutputHTML() As %Status
{

    If ($System.Security.Check("NotAnonymize", "USE")) {
        quit ..OutputHTMLOriginal()
    } else {
        quit ..OutputHTMLAnonymize()
    }
}

/// Display Segments as HTML, using DocType info if available
Method OutputHTMLZen() As %Status
{
    If ($System.Security.Check("NotAnonymize", "USE")) {
        quit ..OutputHTMLZenOriginal()
    } else {
        quit ..OutputHTMLZenAnonymize()
```

```
    }
}

/// Display Segments as HTML, using DocType info if available
Method OutputHTMLOriginal() As %Status
{
    Set tSC=$$$OK
    Set tSeparators=..Separators  Set:""=tSeparators tSeparators=$$$HL7DefSeparators
Set:""=$$$SEGTERM(tSeparators) tSeparators=tSeparators_..SegmentTerminator  Set $$$SE
GTERM(tSeparators)=$ZStrip($$$SEGTERM(tSeparators),"*CW")
    Quit ..OutputToDevice(,tSeparators,"EnsLib.HL7.Util.FormatHTMLv2")
}

/// Display Segments as HTML, using DocType info if available
Method OutputHTMLZenOriginal() As %Status
{
    Set tSC=$$$OK
    Set tSeparators=..Separators  Set:""=tSeparators tSeparators=$$$HL7DefSeparators
Set:""=$$$SEGTERM(tSeparators) tSeparators=tSeparators_..SegmentTerminator  Set $$$SE
GTERM(tSeparators)=$ZStrip($$$SEGTERM(tSeparators),"*CW")
    Quit ..OutputToDevice(,tSeparators,"EnsLib.HL7.Util.FormatHTMLv2Zen")
}

/// Display Segments as HTML, using DocType info if available
Method OutputHTMLAnonymize() As %Status
{
    Set tSC=$$$OK
    Set tSeparators=..Separators  Set:""=tSeparators tSeparators=$$$HL7DefSeparators
Set:""=$$$SEGTERM(tSeparators) tSeparators=tSeparators_..SegmentTerminator  Set $$$SE
GTERM(tSeparators)=$ZStrip($$$SEGTERM(tSeparators),"*CW")
    Quit ..OutputToDevice(,tSeparators,"EnsLib.HL7.Util.FormatHTMLv2Anonymize")
}

/// Display Segments as HTML, using DocType info if available
Method OutputHTMLZenAnonymize() As %Status
{
    Set tSC=$$$OK
    Set tSeparators=..Separators  Set:""=tSeparators tSeparators=$$$HL7DefSeparators
Set:""=$$$SEGTERM(tSeparators) tSeparators=tSeparators_..SegmentTerminator  Set $$$SE
GTERM(tSeparators)=$ZStrip($$$SEGTERM(tSeparators),"*CW")
    Quit ..OutputToDevice(,tSeparators,"EnsLib.HL7.Util.FormatHTMLv2ZenAnonymize")
}
```

EnsLib.HL7.Util.FormatHTMLv2Anonymize and EnsLib.HL7.Util.FormatHTMLv2ZenAnonymize look like this :

```
Class EnsLib.HL7.Util.FormatHTMLv2Anonymize Extends EnsLib.HL7.Util.FormatHTMLv2
{

ClassMethod OutputSegment(pSegObj As EnsLib.EDI.Segment, Output pStatus As %Status, p
IOStream As %IO.I.CharacterStream, pSeparators As %String, pSegNum As %String, pSegPa
th As %String, pParentDoc As EnsLib.EDI.Document, ByRef pSequenceNumber As %String) A
s %Boolean
{
    set tSegObj = pSegObj.%ConstructClone()

    if tSegObj.Name = "PID" {
```

```
        // Names
        $$$ThrowOnError(..Anonymize(tSegObj,"5.3"))
        $$$ThrowOnError(..Anonymize(tSegObj,"5.2"))
        $$$ThrowOnError(..Anonymize(tSegObj,"5.1"))

        // Birthday
        $$$ThrowOnError(..Anonymize(tSegObj,"7"))

        // Adress
        $$$ThrowOnError(..Anonymize(tSegObj,"11.1"))
        $$$ThrowOnError(..Anonymize(tSegObj,"11.2"))
        $$$ThrowOnError(..Anonymize(tSegObj,"11.3"))
        $$$ThrowOnError(..Anonymize(tSegObj,"11.4"))
        $$$ThrowOnError(..Anonymize(tSegObj,"11.5"))
        $$$ThrowOnError(..Anonymize(tSegObj,"11.6"))

        //SSN
        $$$ThrowOnError(..Anonymize(tSegObj,"19"))

    }

    quit ##super(tSegObj , .pStatus , pIOStream , pSeparators , pSegNum , pSegPath ,
pParentDoc, .pSequenceNumber)
}

ClassMethod Anonymize(pSegment As EnsLib.EDI.Segment, pPosition As %String) As %Statu
s
{

    set sc = $$$OK

    set tLen = $LENGTH(pSegment.GetValueAt(pPosition))

    set value = ""
    for i=1:1:tLen {
        set value = value _ "*"
    }

    if value '= "" {
        set sc = pSegment.SetValueAt(value,pPosition)
    }


    Quit sc
}

}
```

Here we can notice two things:

One, we use the ##super() method in order to reuse the logic of the overloaded method.

Secondly, the segments are anonymized by their positions and not by their names because not all HL7 objects have doctypes.

#HL7 #InterSystems IRIS for Health

**Source URL:**https://community.intersystems.com/post/hl7-pid-obfuscation