Article
Sean McKenna · Aug 5, 2016 8m read

HealthShare's new SDA extensions

Creating and working with the new SDA extensions for storage of custom data elements

In HSCore 15.01, there is a new way to store custom data elements. HealthShare now had the ability to use custom extensions on many SDA elements.

This article will:

1. Show how to set up your system to use SDA extensions
2. Create a new SDA extension property
3. Use the new SDA extension property in HL7 transactions
4. Interact with the new data
5. Show new SDA extension used in a customization of Patient Summary Report

Note: For this article, I am using build:
HS-2016.1.1.108.0-hscore15.01_hsaa15_hspi15_hsviewer15.01_linkage15-b2136-win_x64
I also created the system using “d ##class(HS.Util.Installer).InstallBusDemo()”

Set up your system for SDA extensions

This section will describe how to set up SDA extensions for a HSCore 15.01 environment.

Create a new Namespace

With the new SDA Extensions, the name of the Custom Namespace must be **HSCUSTOM**.

You can add this by going into: Management Portal->System Administration->Configuration->System Configuration->Namespaces.

Steps to create a new Namespace:

1. Hit the "Create New Namespace" button
2. Enter HSCUSTOM in the "Name of the namespace" field (Required)
3. Select the "Create New Database" button

- Enter HSCUSTOM in the "Enter the name of your database" field
- For the "Database directory" field:
    1. Hit the "Browse..." button.
    2. Create a new folder/directory, I named my directory "HSCUSTOM" which resides in the "mgr" directory.
    3. Hit the "OK" button
- Hit the "Next" button

- Accept the Defaults, and hit the "Next" button.
- Create a new Resource called %DB_HSCUSTOM and give it Read and Write permissions
- Hit the "Finish" button.

4. Back at the "New Namespace" screen, hit the "Save" button.

This will create a new namespace HSCUSTOM with all the default mappings.

Export the HS.Local package

One of the things that we need to do is copy the classes and code from the HSLIB database to the HSCUSTOM database.

You can do this a number of ways. I will talk about doing it from Studio or Terminal.

Exporting from Studio:

1. Log into Studio
2. Change the namespace to HSLIB namespace (note: this can be done from any namespace that has HS package mapped to HSLIB)
3. Go into Tools->Export menu.
4. Hit the "Add" button
5. Select HS/Local folder and Select everything and hit "Open" button.
6. This will select everything
7. Select a Local or Remote file to export these classes
    - In this example, I named the file "HSLocal.xml"

8. Hit the "OK" button.

Exporting from Terminal session:

1. Log into Terminal
2. Change the namespace to HSLIB namespace (note: this can be done from any namespace that has HS

package mapped to HSLIB)
3. HSLIB>d $system.OBJ.Export("HS.Local.*.cls","C: \ Intersystems \ Export \ HSLocal.xml")

Add a new package mapping

The package "HS.Local" needs to be referenced from the new namespace HSCUSTOM. When HSCUSTOM is created, it will have "HS" mapped to HSLIB. You will need to add "HS.Local" to the HSCUSTOM namespace, since it is currently pointed to HSLIB.

You can do this manually through the Management Portal or you can do this programmatically.

Management Portal:

1. Go to Management Portal->System Administration->Configuration->System Configuration->Namespaces
2. Look for HSCUSTOM under the namespace column and select "Package Mapping" link.
3. Hit the "New" button
4. Select **HSCUSTOM** from the "Pakage Database Location" drop down
5. Select "Specify a new package" radio button
6. Enter HS.Local into the "Package Name" field.
7. Hit "OK" button

Programmatically:

You can create code to add this mapping to a namespace.

1. Move to the %SYS namespace
    - HSCUSTOM> ZN "%SYS"
2. Define the database property
    - %SYS> set tProperties("Database")="HSCUSTOM"
3. Create the mapping
    - %SYS>w ##class(Config.MapPackages).Create("HSCUSTOM","HS.Local",.tProperties)

Note: You will need to do this for every namespace and instance that is a HealthShare namespace, with the exceptions of the Library namespaces. It is important to have these mappings so the other namespaces can access the HSCUSTOM code to use in their processing, like applications like Patient Index and Health Insight.

Import the HS.Local package

Now that the HS.Local packages are now pointing to HSCUSTOM, you can move the classes that we exported earlier into the HSCUSTOM namespace.

You can do this a number of ways. I will talk about doing it from Studio or Terminal.

Importing from Studio:

1. Log into Studio
2. Change the namespace to HSCUSTOM namespace.
3. Go into Tools->Import Local menu.
4. Select the file that you exported
5. Hit the "Open" button.

6. You should see all the classes checked and the "Compile Imported Items" checked.

7. Hit the "OK" Button.

Importing from Terminal session:

1. Log into Terminal
2. Change the namespace to HSCUSTOM
3. HSLIB>d $system.OBJ.Load("C: \ Intersystems \ Export \ HSLocal.xml",”ck”)

Summary

We now have the infrastructure for the new HSCore 15.01 custom SDA Extensions. We have the HS.Local classes defined in a new database HSCUSTOM and we have all the namespaces pointing to the proper location.

If you have more than one Cache Instance, the HSCUSTOM namespace and HS.Local mappings need to be on every instance that runs HealthShare.

Create a new custom SDA Extension

Now that we have the pieces in place, let's create a new custom property.

We will start by creating a custom property for the Patient SDA.

Looking at the HL7 Annotations, it looks like "Veterans Military Status", which is PID, piece 27 is not used in SDA, so let's try and create this as our Custom SDA Extension.

Because PID piece 27 is a coded entry field, we will show that the new custom SDA extensions are more than the previous name/value pair, it can now be a more complex data type. In this example, we are creating a Custom property type.

Edit HS.Local.SDA3.PatientExtension.cls

We need to add the new property to the HS.Local.SDA3.PatientExtension.cls

1. Log into Studio
2. Change namespace to HSCUSTOM
3. Edit HS.Local.SDA3.PatientExtension.cls
4. Add a Custom Class
    - This class represents a complex data type that will have:
        - Code field
        - Description Field

5. Add property VeteransMilitaryStatus

- Property VeteransMilitaryStatus As CUSTOM.SDA3.CodeTableDetail.VeteransMilitaryStatus;

6. Compile HS.Local.SDA3.PatientExtension.cls
7. Compile HS.SDA3.Patient class
8. Compile HS.Registry.Patient class

Now this property is available to add/edit/delete from the SDA streamlet.

Use the new SDA extension property in HL7 transactions

Before we can use the SDA extension property, we need to create a new Custom Class that will extend the HS.Gateway.HL7.HL7ToSDA3 class. This code will run on the EDGE gateway.

Here is a code sample of the new custom class:

```
Class CUSTOM.Gateway.HL7.HL7ToSDA3 Extends HS.Gateway.HL7.HL7ToSDA3 [ Not ProcedureBl
ock ]
{

/// Callback method for custom processing of Patient streamlet
ClassMethod OnPatient()
{
                do ..write(cr_"<Extension>")
                set tVMSCode = $$$xml($g(^||d(s,27,1)))
                set tVMSDescription = $$$xml($g(^||d(s,27,2)))
                if tVMSCode'="" {
                                do ..write(cr_"<VeteransMilitaryStatus>")
                                do ..write(cr_"<Code>"_tVMSCode_"</Code>")
                                do ..write(cr_"<Description>"_tVMSDescription_"</Desc
ription>")
                                do ..write(cr_"</VeteransMilitaryStatus>")
                }
                do ..write(cr_"</Extension>")
                Quit
}
```

Update the Edge Gateway Ensemble Production.

Edit the Operation: HS.Gateway.HL7.InboundProcess and change the setting "HL7ToSDA3Class" to use the new class we just created.

Hit the "Apply" Button to save the changes.

We use the following HL7 message: (Note, that PID piece 27 has a value of "V^Veteran")

```
MSH|^~\&||HC6|||||ADT^A04|||2.5
EVN|A04|20160711094500
PID|||STM123^^^HC6^MR||Bolton^George||19271014|M|||1 Memorial Drive^^Cambridge^MA^021
42||||||||028345081||||||||V^Veteran
```

We now process this HL7 on an Edge Gateway.

Interact with the new data

Now we have the data stored in SDA that is stored on the Edge (and the Registry), we can look at it in several ways.
Now if we look at the trace, we can see our data in the <Patient> SDA.

On the Edge, I can run the following code:

```
ClassMethod ListVeteransStatus()
{
                #dim tPatient as HS.SDA3.Patient
                set sql="Select ID from HS_SDA3_Streamlet.Patient"
                set statement = ##class(%SQL.Statement).%New()
                set tSC = statement.%Prepare(sql)
                quit:$$$ISERR(tSC) tSC
                set result = statement.%Execute()
                while result.%Next() {
                      set tStreamletID = result.ID
                      set tSC = ##class(HS.SDA3.Container).LoadSDAObject(tStreamletID
,.tPatient)
                      w !,tPatient.Name.GivenName_" "_tPatient.Name.FamilyName
                      w ": Veteran Status = "_tPatient.Extension.VeteransMilitaryStat
us.Description
                }
                Quit
}
```

And get the following:

```
HSEDGE1>d ##class([sample class]).ListVeteransStatus()
George Bolton: Veteran Status = Veteran
```

You can see that we are using the property "Extension.VeteransMilitaryStatus.Description" to display the new custom data.

On HSREGISTRY, we can look at the following:

```
HSREGISTRY>s oPat=##class(HS.Registry.Patient).%OpenId(1)
HSREGISTRY>w oPat.Extension.VeteransMilitaryStatus.Code
V
HSREGISTRY>w oPat.Extension.VeteransMilitaryStatus.Description
Veteran
```

For Patient information, we can see that the custom property is available not only in SDA that is stored on the Edge gateway, but also is available in the Patient Registry.

Using Custom SDA in a customization of Patient Summary Report.

This is just a quick look how you can use the new SDA extension in custom code.

In this example, we will create a custom Patient Summary Report.

I created a CUSTOM.Reports.Patient.Summary class, I can just add a new item to display using the following code:

```
<item field="Extension/VeteransMilitaryStatus/Description" >
        <caption value="Military Status" style="padding:5px;"/>
</item>
```

Now, when we bring up the patient and view Patient Summary Reports we can see the following:

Summary

This article showed how to set up the environment to use the new SDA extensions. It showed how to populate the new property using HL7. We showed how to access the data. In this case, we were using patient data, so we were able to access it from both the Edge gateway and Registry. We also used this new information in an example using a custom patient summary report. Now that this data is in the SDA, it can be used by Health Insight and Patient Index as well as any other place that uses SDA.

#Studio #Terminal #HealthShare #Mapping

Source URL: https://community.intersystems.com/post/healthshares-new-sda-extensions