



1

提纲

- DT的创建
 - DTL, 图形化的界面生成XML Data, 继承Ens.DataTransformDTL
 - 使用ObjectScript创建Custom Code, 继承Ens.DataTransform
- DTL测试, 使用
- HL7v2, HL7v3消息的DT
- Utility Functions

2

DT基本概念

- DT的用途
 - Ensemble消息转变到另一种格式
 - 一种Object格式到另一个Object
 - 转变外部数据到Ensemble message
- DT可以在任意地方被调用(后面介绍)
- DT可以使用代码实现, 但通常的用法是使用DTL
- DTL和DT的类继承关系:

```
abstract class Ens.DataTransform extends %RegisteredObject, Ens.Util.FunctionSet
abstract class Ens.DataTransformDTL extends Ens.DataTransform
```

3

DT实现的方式

数据转换两种实现:

- 数据转换语言(DTL)
 - BPL的子集, 只包含数据转换相关的部分, xml格式
 - 编译后的类继承Ens.DataTransformDTL
 - DTL编译后立即生效, 不需要重启使用它的业务组件
 - 可以在Portal或者Studio里面操作, 通常使用图形界面方式, 也可以直接修改xml
- 用户自定义代码(COS)
 - 在Studio里面写COS代码
 - 继承Ens.DataTransform

4

使用ObjectScript自定义DT

可以使用COS来定义数据转化

- 需要在Studio里写代码
- 适用于较复杂的转化
 - Override transform()

例子: [Demo.HL7v3.Transformation.MFV](#)

This Data Transformation is implemented:

☐ Using the DTL Editor

☒ Using Custom Code

Source Class:

Source Doc Type:

Target Class:

Target Doc Type:

5

6

```

1 class Demo.X12.Sorter.DTL extends Ics.DataTransform { DependsOn = Demo.X12.Message }
2
3
4
5
6
7
8 // Get a HDTTemplate instance as a stream
9
10 // Set the template as a stream
11
12 // Get a stream of XML from the XML that is contained in this class
13
14 // Set the stream as a stream of XML from the XML that is contained in this class
15
16 // Create an HDTTemplate instance from the input stream of XML
17
18 // Set the template as a stream of XML from the XML that is contained in this class
19
20 // Get the patient's first name, upper-case it, and bind to the parameter
21
22 // Set the patient's first name, upper-case it, and bind to the parameter
23
24 // Get the patient's last name, upper-case it, and bind to the parameter
25
26 // Set the patient's last name, upper-case it, and bind to the parameter
27
28 // Transform the source according to the XML
29
30 // Set the target as a stream of XML from the XML that is contained in this class
31
32 // Set the target as a stream of XML from the XML that is contained in this class
33
34 // Copy over the name and the type, then create
35
36 // Set the target as a stream of XML from the XML that is contained in this class
37
38 // Set the target as a stream of XML from the XML that is contained in this class
39
40

```

7

Demo

- DTL界面
 - 拖拽完成简单属性对应关系
- DTL创建
 - 在SMP创建: Ensemble ☐ Build ☐ Data Transform
 - 在Studio创建: File -> New -> Production -> Data Transform
- DTL图形化格式和代码的对应

Source	Target
1. NAME	1. NAME
2. SEX	2. SEX
3. AGE	3. AGE
4. RACE	4. RACE
5. ETHNICITY	5. ETHNICITY
6. RELIGION	6. RELIGION
7. OCCUPATION	7. OCCUPATION
8. MARITAL STATUS	8. MARITAL STATUS
9. NUMBER OF CHILDREN	9. NUMBER OF CHILDREN
10. NUMBER OF SIBLINGS	10. NUMBER OF SIBLINGS
11. NUMBER OF PARENTS	11. NUMBER OF PARENTS
12. NUMBER OF GRANDPARENTS	12. NUMBER OF GRANDPARENTS
13. NUMBER OF GREAT-GRANDPARENTS	13. NUMBER OF GREAT-GRANDPARENTS
14. NUMBER OF GREAT-GRANDPARENTS	14. NUMBER OF GREAT-GRANDPARENTS
15. NUMBER OF GREAT-GRANDPARENTS	15. NUMBER OF GREAT-GRANDPARENTS
16. NUMBER OF GREAT-GRANDPARENTS	16. NUMBER OF GREAT-GRANDPARENTS
17. NUMBER OF GREAT-GRANDPARENTS	17. NUMBER OF GREAT-GRANDPARENTS
18. NUMBER OF GREAT-GRANDPARENTS	18. NUMBER OF GREAT-GRANDPARENTS
19. NUMBER OF GREAT-GRANDPARENTS	19. NUMBER OF GREAT-GRANDPARENTS
20. NUMBER OF GREAT-GRANDPARENTS	20. NUMBER OF GREAT-GRANDPARENTS

8

创建DT可以使用什么类

- Ensemble 消息
- Persistent类
 - XML结构
- %RegisteredObject类
 - XML结构
- 虚拟文档

9

DTL组件设置

- 创建方式
 - New, Copy, Existing
- 源类, 目标类
- 源/目标文档类型(DocType)
 - 用于VDoc, 比如HL7消息
- 报告错误
 - 默认选中, 把转换中的错误存入Event log, as Warning
- 忽略缺失的源断(Ignore missing source segments)
 - 默认选择, 当源字段为空时不记录错误, 不执行相关的子转换
- 把空的重复字段视为null(Treat empty repeating fields as null)



12

创建方式: New, Copy

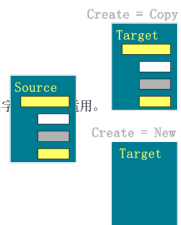
New

- 默认创建方式
- 目标开始是空的
- 需要一个一个的设置目标属性

Copy

- 必须源和目标类相同, 否则会编译出错。即便两个类内容一致只是名字不同
- 只需要设定修改的内容
- 常用于HL7消息从一个版本像另一个版本的转换(DocType不同)

example: Demo.X12.Sorter.DTL.ByGroup.Sorter.Transform



13

创建方式: Existing

创建一个转换, 把前面已存在的转换的对象的转换目标, 能提高效率。一般用于在code里实现数据转换, 或者执行一系列的转换。

You have three transformations that you want to perform in sequence:

- MyApp.ADTTransform — Uses the new option for Create.
- MyApp.MBTTransform — Uses the existing option for Create.
- MyApp.LABTransform — Uses the existing option for Create.

You invoke the transform as follows:

- do MyApp.ADTTransform.Transform(message, jump)
- do MyApp.MBTTransform.Transform(message, newtarget)
- do MyApp.LABTransform.Transform(message, newmessage)

其他例子:

- Demo.X12.Sorter.DTL.ByGroup.Sorter.Transform.dtl

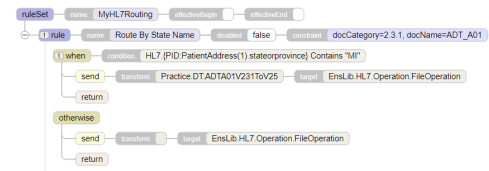
15

DTL, DT的使用

16

在消息规则里使用DT

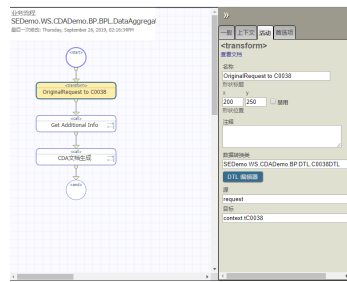
在定义路由规则Routing Rule的使用使用.



17

在BPL中使用DT

- BPL里面使用DT



18

在其他组件中使用DT

- 在任何业务组件里面调用DT使用以下classmethod
classmethod Transform(source As %RegisteredObject, ByRef target As %RegisteredObject, ByRef aux=("")) as %Status

//以下是在业务组件里用code调用DT的例子

```
Method PatientAdd(Request As HL7.V3.XML.Document, Output pResponse As WS.Service.Entity.Bundle) As %Status {
    set pRequest.DocType=SSSPatientAddTPl
    set pResponse=##class(WS.Service.Entity.Bundle).%New()
    set sC=##class(WS.Service.DT.V3.PatientAdd2Bundle).Transform(pRequest,pResponse)
    do pResponse.XML.Export()
    ...
}
```

19

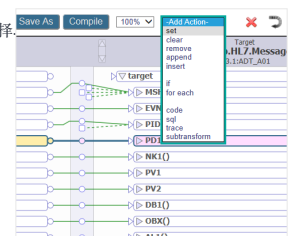
DTL的操作(Action)

20

设定操作 (Actions)

选定一个属性 □ 从Add Action下拉菜单选择.
操作的内容可以在右面的窗口修改

- set
- clear, remove, append, insert
- if
- for each
- codes
- sql
- trace
- subtransform



21

需要知道的技巧

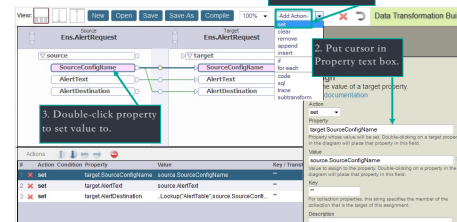
- 日期时间的处理
- 集合的处理
 - 对于List of %String
 - 对于list of object
- for each
- if
- for each的嵌套
- for each和if的嵌套



22

Set Actions

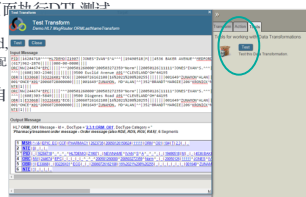
注意set的值是字符串时要用“”



23

测试DTL注意事项

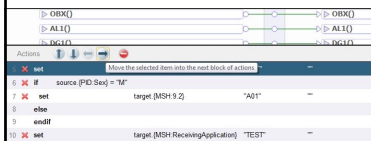
- 在管理界面或者Stuido都可以用专用页面运行DTL, 如图
- Tools tab □ Test.
- 测试的源和目标类必须是XML Enabled, 错误可能不会在DTL创建或修改时发现, DTL配Enabled
- 对象类和数字类的属性在测试时不能自



24

操作的顺序

- 从上至下执行
 - '设置'的内容有层关系时, 尤其要注意执行顺序。
- 下面窗口的上下箭头用于把一个设置上下移动
- 左右箭头把一个语句移入或者移出另一个语句, 比如把set移入一个if/else或者一个for each



25

Actions的顺序

Actions的执行顺序是从上到下

根据需要修改action的顺序

1. 首先, 完成MSH segment.

2. 接着, MSH property.

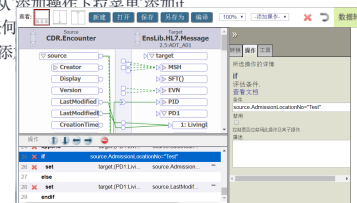
#	Action	Condition	Property	Value	Key / Transform
1	set		target (MSH)	source (MSH)	
2	set		target (MSH 9.1)	"ADT"	
3	set		target (MSH 9.2)	"AGT"	
4	set		target (MSH.ReceivingApplication)	"TEST"	
5	set		target (EVN)	source (EVN)	
6	set		target (PID)	source (PID)	
7	set		target (PID.5.1)	"NEWNAME"	



26

Action- If

- 先添加If语句(点击源元素, 从'添加操作'下拉菜单'添加if
- 补全if判断条件, 可以使用任何
- 在选中if或者else的情况下, 添



27

Action- for each

- 重复处理一系列动作
 - 属性集合 (Collection)
 - 重复的属性
 - 重复的部分, 不如VDoc中的部分

- 其他的使用例子:
 - Demo.ComplexMap.Transform.CreateClassBatch.dtl

foreach

Loop over the contents of a collection
View documentation
Property
source (PID.PatientIdentifierList())
Property to iterate over. Double-clicking on a property in this field.
Key
k1
Name of the iterator variable used for this loop.

#	Action	Condition	Property	Value	Key / Transform
1	for each		source (PID.PatientIdentifierList())		k1
2	if	source (PID.PatientIdentifierList(k1).id			
3	set		target (PID.PatientIdentifierList(k1).id	"NR"	
4	else				
5	endif				
6	if	source (PID.PatientIdentifierList(k1).id			

28

HL7 DTL

29

测试DTL-HL7消息



30

指定一个字段 Fields

- 默认, DTL中使用圆括号 "target.{...}"
- 可以使用下面任何一种方法指定字段field
 - MSH.9.1
 - MSH.MessageType.messageType

#	Action	Condition	Property	Value	Key / Transform
1	set		target (MSH)	source (MSH)	
2	set		target (MSH.9.1)	"ADT"	
3	set		target (MSH.9.2)	"A01"	
4	set		target (MSH.ReceivingApplication)	"TEST"	
5	set		target (EVN)	source (EVN)	
6	set		target (PID)	source (PID)	
7	set		target (PID.5.1)	"NEWNAME"	

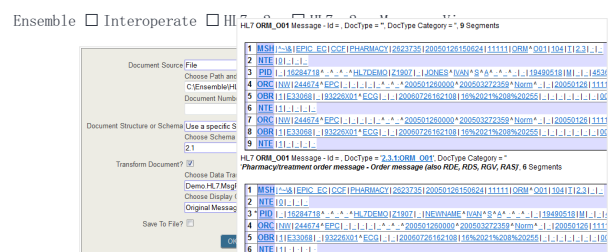
31

设定重复的段 Segments和组 Groups

- 重复的段 segment, 指定相应的数字在 () 中
target. {NK1 (1):10}
- 对于一组的段 (group of segments), 在组的第一个段后附加 "grp" 的字段
source. {ORCgrp (1). ORC:5}
- 对于嵌套组, 可同样附加 "grp".
target. {PIDgrpgrp (1). ORCgrp (1). OBR:Set IDOBR}

32

从HL7 Message Viewer测试



33

其他

34

转换函数(Utility Functions)

DTL的内置函数，在Action里面使用，用于值的转换，比如数学函数，字符串函数等。

具体函数的使用细节参见以下文档:Developing Business Rules □ Ensemble UI

- 注意下面函数使用时的 前缀



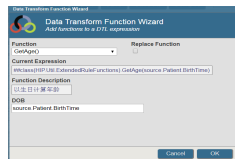
35

转换函数(Utility Functions)

转换函数的扩展:

- 创建新类继承Ens.Rule.FunctionSet, 编译后会扩展Portal中的转换函数列表

```
Class HIP.Util.ExtendedRuleFunctions Extends Ens.Rule.FunctionSet
{
    /// 以生日计算年龄
    ClassMethod GetAge(DOB As %String) As %Integer{
        quit (1+HH-$ZDTH(DOB,X)) \ 365.25)
    }
}
```



36

命名规范

由名字了解是什么转换

内容包括:

- System.
- Message type.
- “DTL”.

例如, EmergencyORM001ToPharmORM001DTL.

参考以下文档: Getting Started With Ensemble □ Ensemble Best Practices □

Design Model for a Routing Production □ Naming Conventions.

http://localhost:57777/csp/docbook/DocBook.UI.Page.cls?KEY=EGBP_routing_best_practices#EGBP_naming_conventions

37

DTL在线文档

参考 Ensemble Business Process Language and Data Transformation Language XML Reference for DTL details.

Reference DTL Elements

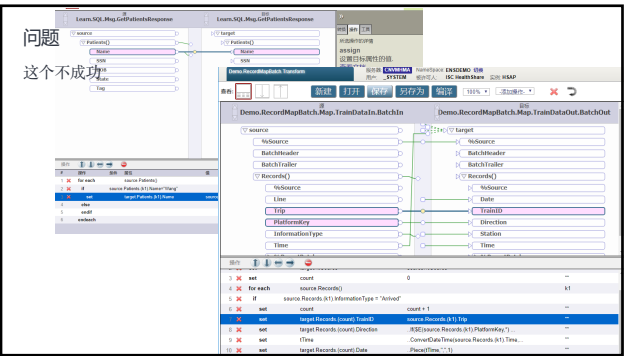
- DTL <assignment> — Provide a descriptive comment for a DTL element.
- DTL <assign> — Assign a value to a property within an object.
- DTL <code> — Execute lines of custom code.
- DTL <false> — Perform a set of activities when the condition for an <if> element is false.
- DTL <foreach> — Define a sequence of activities to be executed iteratively.
- DTL <if> — Evaluate a condition and perform one action if true, another if false.
- DTL <log> — Generate an embedded SQL, SQL27 statement within a data transformation.
- DTL <subtransform> — Invoke another data transformation.
- DTL <trace> — Write a message to the foreground Terminal window.
- DTL <transform> — Transform an object of one type into an object of another type.
- DTL <true> — Perform a set of activities when the condition for an <if> element is true.

38

总结

- DTL或者自定义DT
- DTL的操作从上往下执行
- DTL中可以使用代码块, SQL,另一个DTL.
- DTL可以使用if/else, for each
- 使用转换函数
- DTL有专属的测试页面

39



40