

文章

[Qiao Peng](#) · 五月 15, 2022 阅读大约需 9 分钟

InterSystems互操作进阶 - 第二篇：规则引擎（第一部分）

在软件开发和业务集成中，规则无处不在：会员折扣的计算规则、根据消息类型和内容将其路由到不同目标系统的路由规则。还有一个规则发挥重要作用的地方-辅助决策规则，例如临床知识库和医疗质量指标规则。

规则经常需要随业务调整和知识积累进行调整，而规则的调整是业务和行业专家定的。如果规则是以代码硬编码的，这些调整需要程序员改动，一来不直观、需要业务专家与程序员大量的沟通成本，二来硬编码改动会对应用伤筋动骨，甚至带来风险，三来没法控制新规则生效的时间 - 总不能让程序员在新规则生效的那一刻去编译和部署吧。

InterSystems规则引擎可以帮助我们解决这些问题，于构建、执行和记录消息路由规则和普通的业务规则，带给应用和集成方案充分的灵活性和可用性。甚至业务专家和临床信息学家都可以通过低代码的、图形化的规则编辑器修改规则和指定规则生效和失效时间。

InterSystems规则引擎是InterSystems IRIS数据平台和Health Connect与Ensemble集成平台的组件。创建的规则可以被单独调用，也可以被业务流程调用。

本篇介绍规则的如何使用InterSystems规则编辑器创建规则和规则引擎执行规则的逻辑。

1. 规则基本概念

在设计规则前，先了解一下规则的基本概念。

1.1 上下文

上下文是用于规则定义的数据模型。用于临床决策支持的上下文通常需要患者信息、诊断、体征与观察、过敏、医嘱等信息来制定规则。例如糖化血红蛋白(观察)超过10%，要提示紧急警告。因为上下文提供了建立规则判据的主要*数据，所以它非常关键。

医疗行业的数据格式非常复杂，有HL7 V2这种分隔符分隔的字符串、有HL7 CDA这样的XML字符串/字符流、有新的基于JSON的FHIR、还有SQL二维表的记录。这些数据格式都可能会被用做建立规则的上下文模型。

*注：除了上下文中的数据，还可以使用规则临时变量的数据作为判据。

1.2 规则类型

规则可以用于很多场景，InterSystems规则引擎支持多种规则类型，包括：

- 用于消息路由的规则
 - 根据上下文（请求消息）信息类型和内容，决定消息路由的目标
 - InterSystems提供多种开箱即用的消息路由规则类，分别对应HL7 V2消息、行业虚拟文档消息（DICOM、X12等）、XML虚拟文档消息和普通对象消息
- 普通的业务判断规则
 - 根据上下文内容，决定返回结果
 - 上下文可以是普通对象、HL7消息、医疗行业虚拟文档、XML文档、JSON等
- 管理警告规则
 - 根据系统警告信息路由警告给不同的用户

我们会介绍路由规则和普通业务规则。

无论创建哪种规则，InterSystems技术平台都以类定义保存规则逻辑，也就是创建规则类，从而方便的导入/导出。

1.3 规则辅助类

规则辅助类约束

规则定义并决定规则可以使用哪

些操作，例如路由规则可以使用“**发送**”操作，而普通业务规则可以使用“**返回**”操作。

系统提供的每个规则类型都有对应的规则辅助类，例如虚拟文档路由规则的规则辅助类是：`En sLib.MsgRouter.VDocRuleAssist`。

在选定创建的规则类型后，规则编辑器会自动选择对应的规则辅助类，除非特殊场景，无需自己选择。通常规则辅助类不需要修改，我们将会在高级规则特性介绍文章中介绍用户自定义辅助规则类。

1.4 规则集、规则、条件、行为

规则定义由四层构成：

规则集：

一个规则定义中可以定义

一到多个规则集，同时只有一个规则集生效。

当有多个规则集时，通常它们代表不同的版本，通过规则集不同的生效起止日期进行分别。这样，可以事先生成好新的规则集、并定义好生效时间，规则引擎会在生效时间开始后自动启用新的规则集。

规则：一个规则集中，可以有一到多个规则，这些规则同时生效。例如在HL7 V2路由规则定义中，每个规则可以处理不同类型的V2消息。

规则按其在规则集中的顺序执行，只要规则没有返回（`return`）行为，后面的规则会继续执行。

规则条件

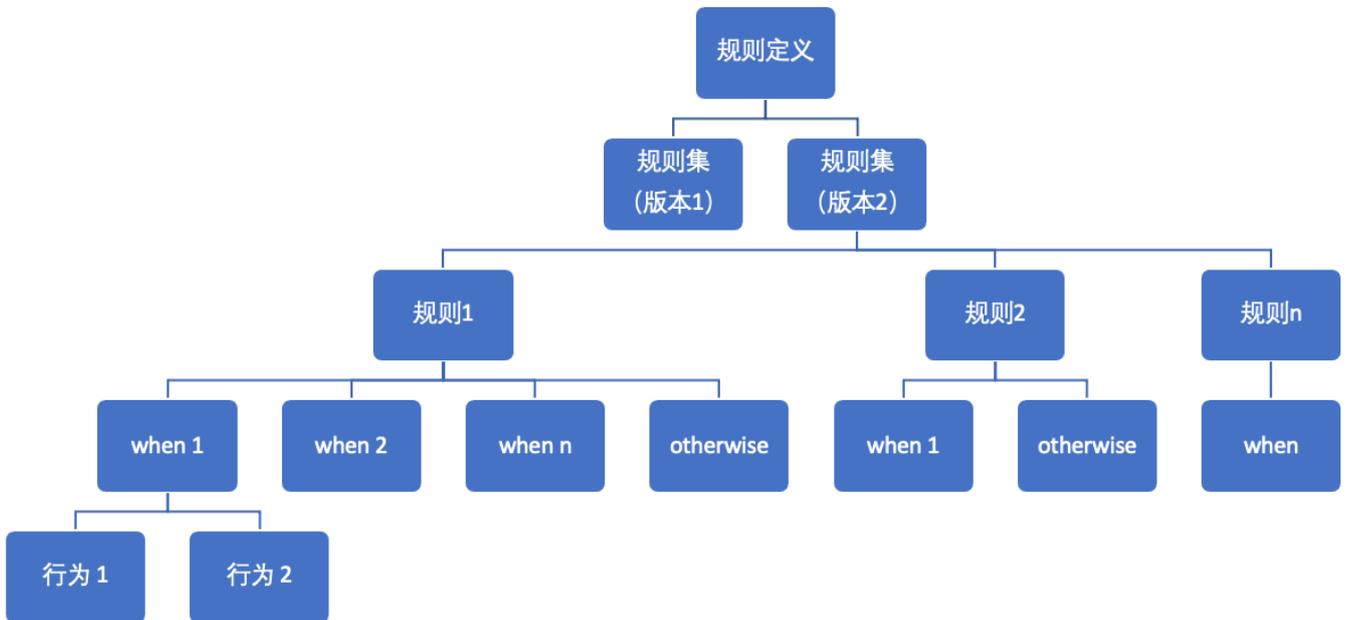
：在一个规则中，可以处理多个规则条件，用when和otherwise定义。规则条件定义需要满足的条件和满足条件时的行为(action)。同一级别条件下，when可以有多个，但otherwise最多有一个，代表所有when都不满足的条件。

不同条件间相当于if...elseif关系，因此一个条件满足时，后面的条件不会再进行判断。

行为

：在规则条件下，可以执行一个或多个行为(action)。

其中返回行为(return)是特殊行为，执行到这个行为会直接返回，进而中断后续的规则、条件和行为的执行。

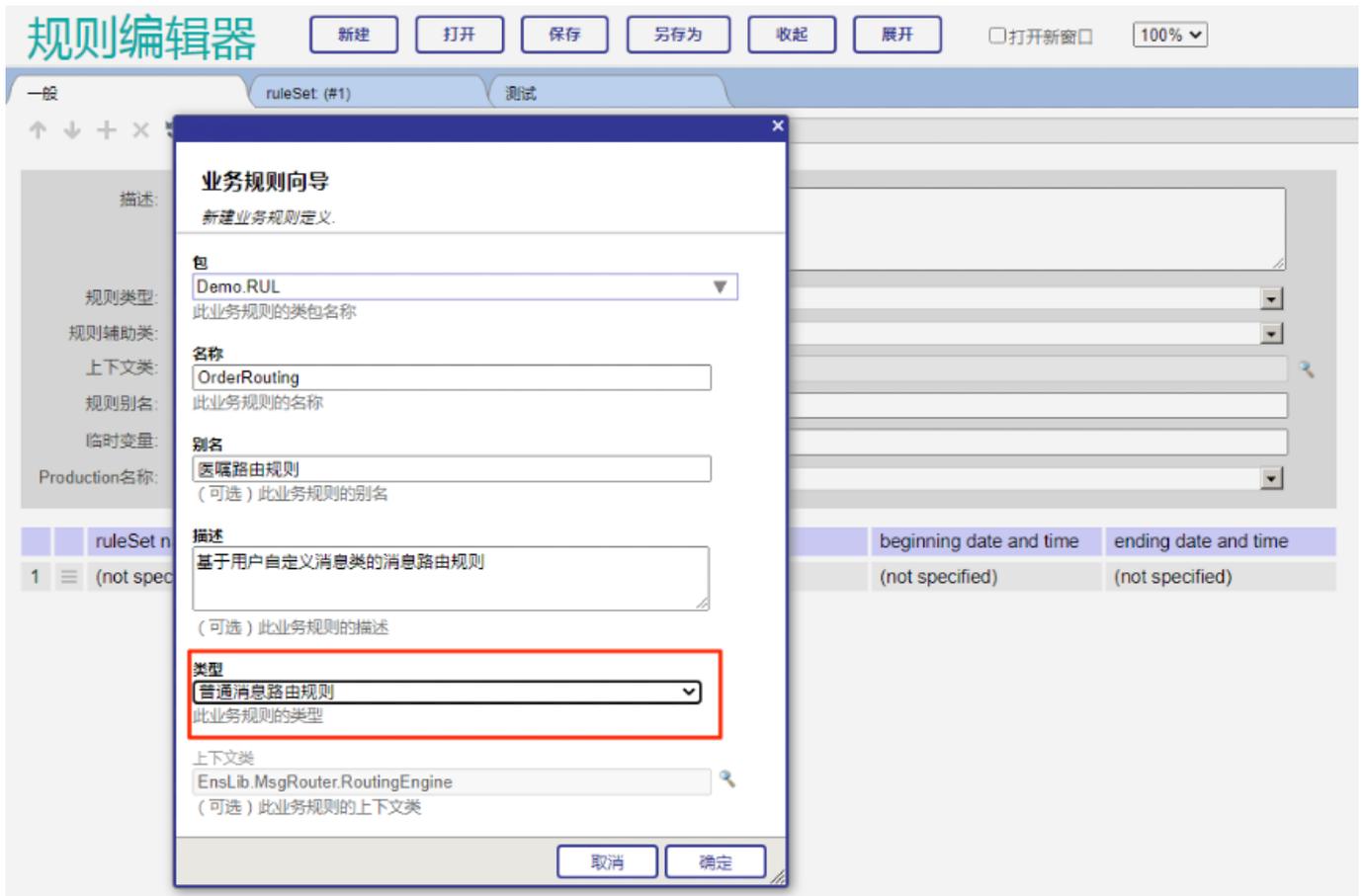


2. 使用规则编辑器

InterSystems规则引擎提供图形化的规则编辑器，通过 **管理门户**>Interoperability>**构建**>**业务规则** 访问它。

第一步要确定我们要建立什么类型规则：是消息路由规则，还是普通业务规则，亦或是管理警告规则？

同时，规则是对象类，因此需要指定它的类包名（包）和类名（名称）。



2.1 消息路由规则的创建与编辑

本章介绍消息路由规则的创建与编辑，如果您是在找普通业务规则的创建方式，请跳到下一章。

消息路由流程本身是一种简单的业务流程

，消息路由规则的上下文就是对应的业务流程对象，因此消息路由规则的上下文类型是固定的。

InterSystems提供4种开箱即用的消息路由规则，分别是：

路由规则类型	说明	上下文类
普通消息路由规则	用于用户自定义的对象类型消息的路由规则	EnsLib.MsgRouter.RoutingEngine
HL7 消息路由规则	用于HL7 V2消息的路由规则	EnsLib.HL7.MsgRouter.RoutingEngine
虚拟文档消息路由规则	用于XML、EDIFACT等虚拟文档类型消息的路由规则	EnsLib.MsgRouter.VDocRoutingEngine
基于消息段的虚拟文档消息路由规则	用于XML、EDIFACT等虚拟文档类型消息的路由规则，基于消息段，而非完整消息	EnsLib.EDI.MsgRouter.SegmentedRoutingEngine

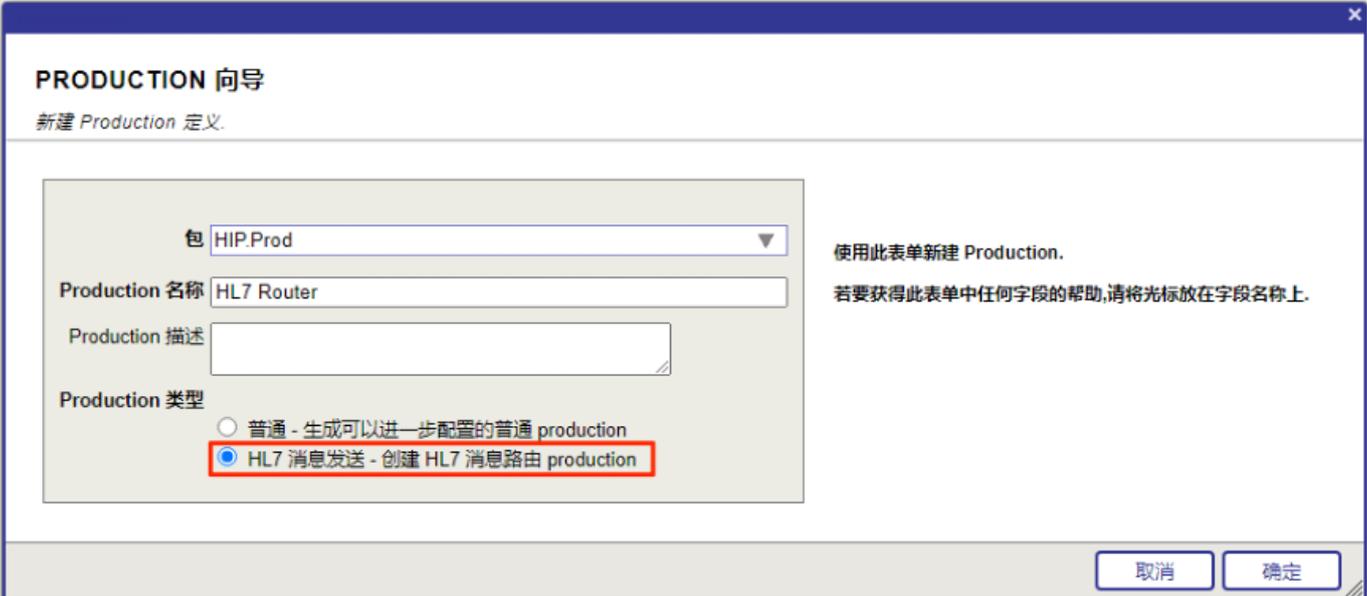
2.1.1 HL7 v2消息路由规则的创建和部署

HL7 v2消息是使用分割符分隔的字符串，在InterSystems数据平台上使用内建的虚拟文档来操作，因此可以将HL7 v2消息的段(Segment)和属性(Field)作为对象和属性方式操作，非常简单。同时，InterSystems数据平台提供开箱即用的路由production模版，直接基于模版编辑即可。

2.1.1.1 创建HL7 v2路由规则的方法

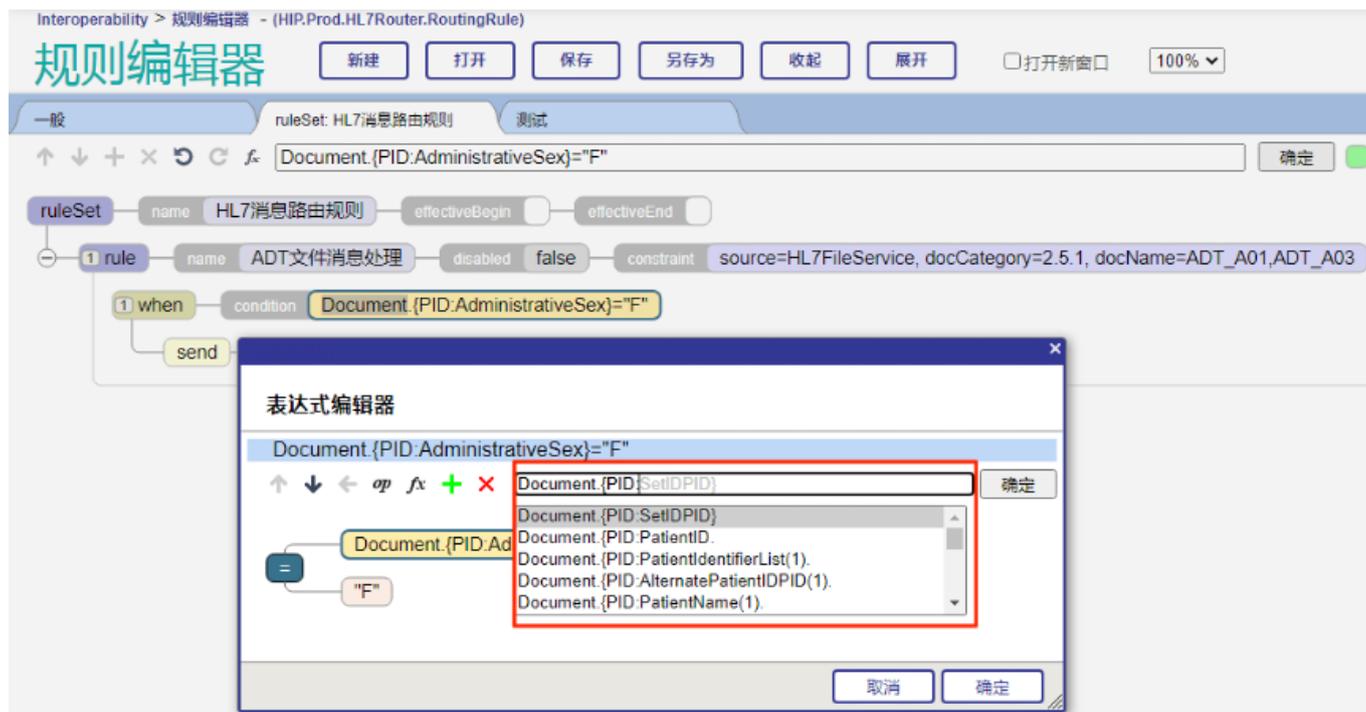
方法1: 使用HL7 v2 production模版向导创建

在production创建向导中，选择production类型为HL7 **消息发送**。这时同时创建用户HL7 v2路由的production和对应路由规则类。这种方法最简单，如果目标就是建立一个路由HL7 v2的产品，推荐使用这个模版向导。



向导会自动应用模版，创建如下的HL7 v2路由产品，它带处理输入HL7文件的BS和输出HL7文件的BO、一个路由规则BP：MsgRouter和一个异常警告BP：Ens.Alert。

选择MsgRouter，在右侧的业务规则名称中可以看到，向导自动创建了名为HIP.Prod.HL7Router.RoutingRule的路由规则。点击后面的放大镜图标，即可对该规则进行编辑。



方法2: 直接使用规则向导创建规则

在规则创建向导中，选择规则类型为：HL7 Message Routing Rule。

BUSINESS RULE WIZARD
Create a new Business Rule definition.

Package
Demo.RUL
Class package name of this business rule

Name
HL7Router
Name of this business rule

Alias

(Optional) Alias for this business rule

Description

(Optional) Description of this business rule

Type
HL7 Message Routing Rule
Type of this business rule

Context Class
EnsLib.HL7.MsgRouter.RoutingEngine
(Optional) Context class for this business rule

Cancel OK

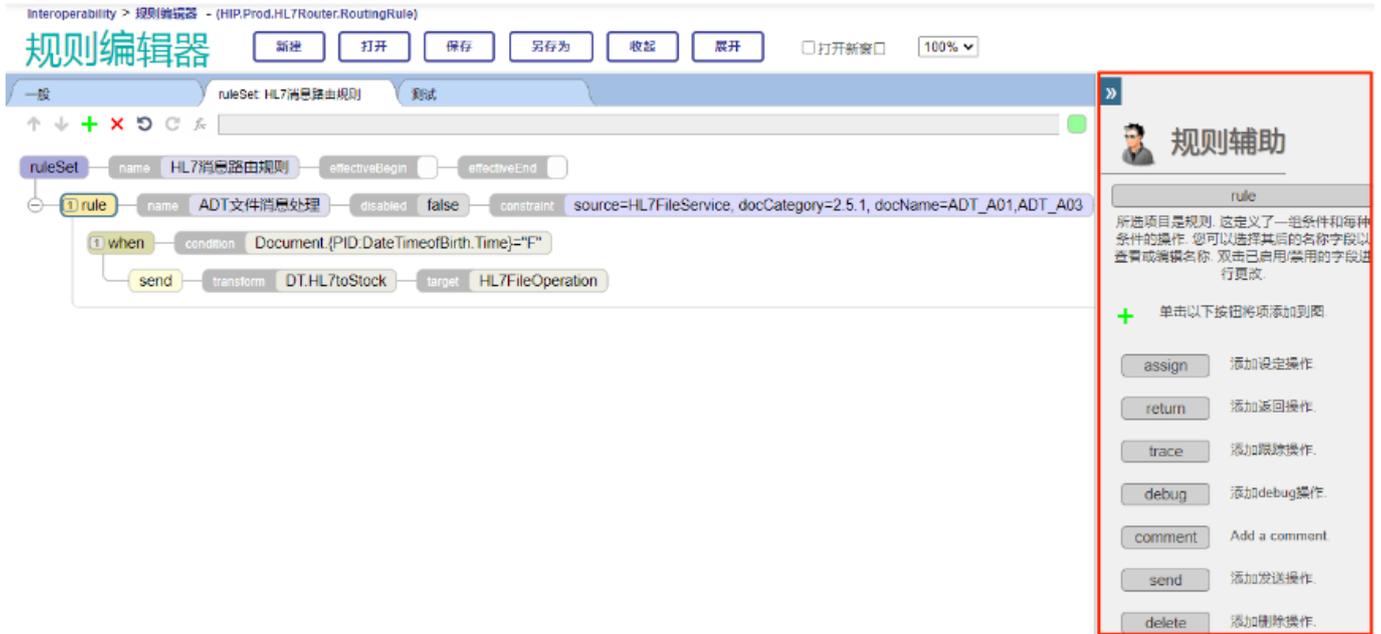
2.1.1.2 编辑规则

使用HL7 v2 production模版向导创建的规则定义已经包含了一个规则集(ruleSet)，其中有一

个规则(rule)，而该规则下只有一个条件(when)。使用规则向导创建的v2路由规则是完全空白的规则定义，只有一个规则集。

我们以HL7 v2 production模版向导创建的规则定义为例介绍规则的编辑。

在规则编辑器页面，右侧是**规则辅助**，由规则辅助类提供支持，列出可以执行的操作。选中ruleset、rule、条件(when/otherwise)和行为，都会显示不同的可以执行的操作和相应的解释。



1. 规则集(ruleSet)是一系列的规则的组合。

规则集可以设置生效时间范围:effectiveBegin和effectiveEnd。如果未定义生效时间范围，规则定义编译后即刻生效。当定义多个ruleSet时，应该定义每个ruleSet的生效时间范围；如果有生效时间范围的重叠，会使用第一个生效的规则集。

规则集有名称 name，建议设置为有意义的规则集说明。

2. 规则(rule)是一系列条件的组合。

选中ruleSet，点击+号就可以添加规则。

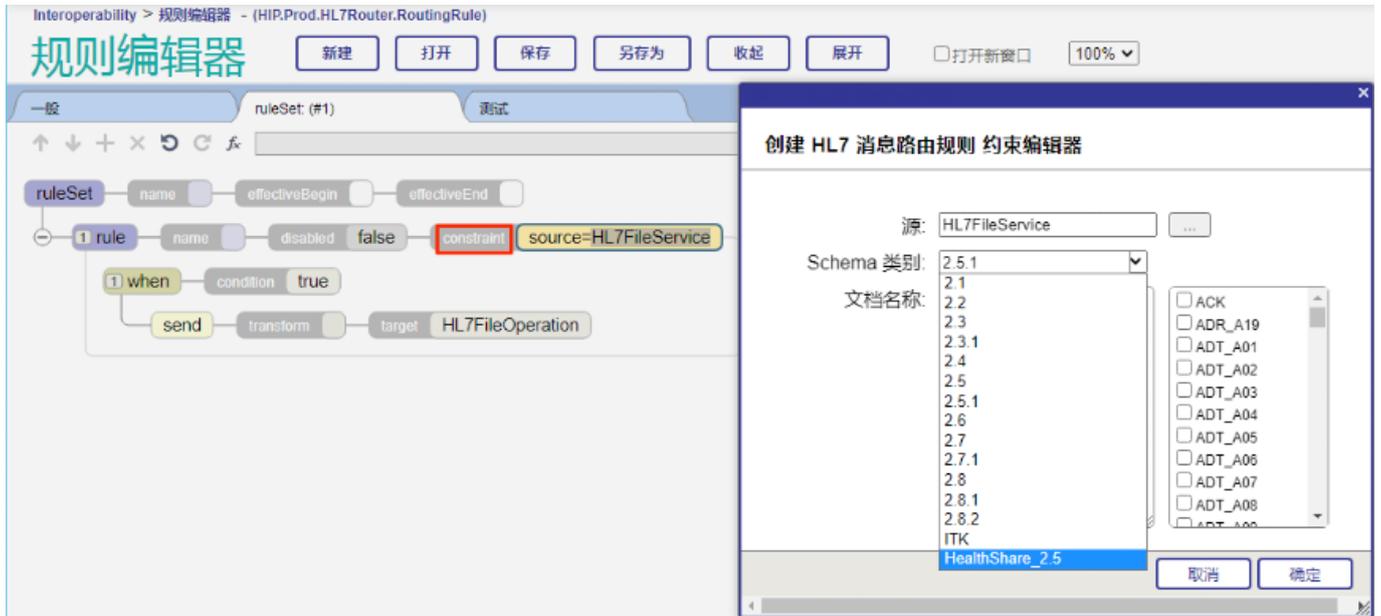
规则可以设置：

名称(name): 应该设置为有意义的字符串，例如 ” ADT文件消息处理 ”。

是否**禁用(disabled)**：默认是启用的(false)。

约束(constraint)：设置本规则的消息约束。双击constraint即可编辑。对于HL7 v2消息，可以对下列信息进行约束：

- **源(source)**
：对消息来源进行约束，本例中源被默认为HL7FileService，就是HL7文件源。如果这个路由规则需要处理多个HL7消息源，例如文件和TCP，可以把源约束删除掉。
- **Schema类别(Schema Category)**：对HL7 v2消息的Schema版本进行约束。如果路由规则只处理特定版本的消息，可以进行选择，例如2.5.1。
- **文档名称(Document Name)**：因为HL7 v2被作为虚拟文档对象进行处理，因此每种类型的v2消息都是一种虚拟文档。如果要对处理的v2消息类型进行约束，可以多选v2消息类型，例如ADTA01和ADTA03。

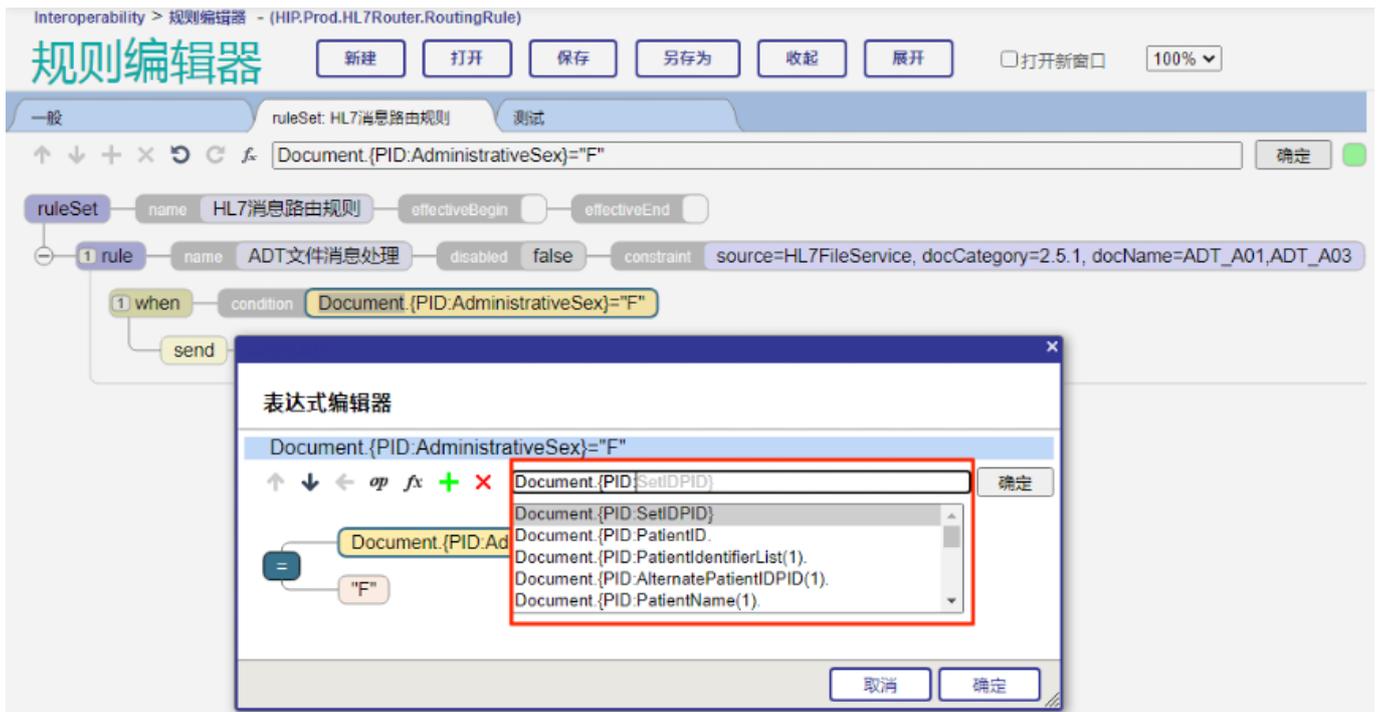


注意，这些约束项都是可选的。任何不符合约束的消息，都不会被路由，并且会被记录在规则日志(rule log)中。

3. 规则条件(when/otherwise)设定条件(condition)。

选中rule，点击+号，即可增加条件when或otherwise。增加后，双击condition即可编辑。可以用在条件的判据数据就是上下文中的属性(数据)，在表达式编辑窗口的文本输入栏会有属性提示，打首字母就会提示，也可以打任意字符并删除，从而显示所有上下文可用属性。

HL7消息是一种**虚拟文档**，因此有Document特殊变量用于访问其虚拟文档内的数据。如果在上一步的规则约束中选择了虚拟文档的类型，如ADTA01，那么引擎还会在 Document.{ 后提供所有对应HL7 v2虚拟文档的段和属性提示，如下图。



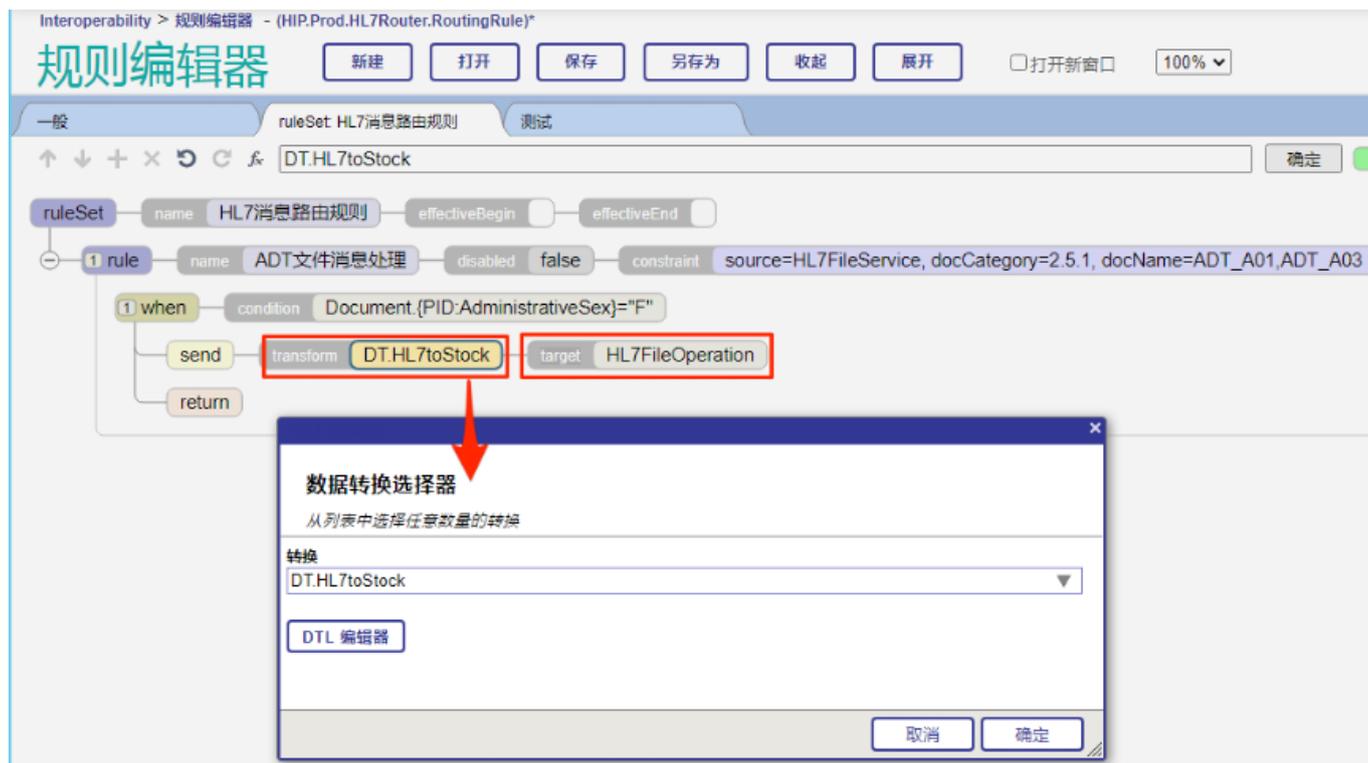
*注：关于虚拟文档，我将在下一篇文章中介绍。

在表达式编辑窗口中，点击op
打开操作符向导，选择操作符，
例如判等、大于、包含等。点击fx
打开函数向导，选择函数，例如ConvertDateTime函数可以将HL7的时间格式(YYYYMMDDhhmmss)转为ODBC格式(YYYY-MM-DD hh:mm:ss)。也可以直接写表达式，而不使用这些操作符和函数向导。用户可以扩展自定义函数，见第二部分的规则函数扩展章节。

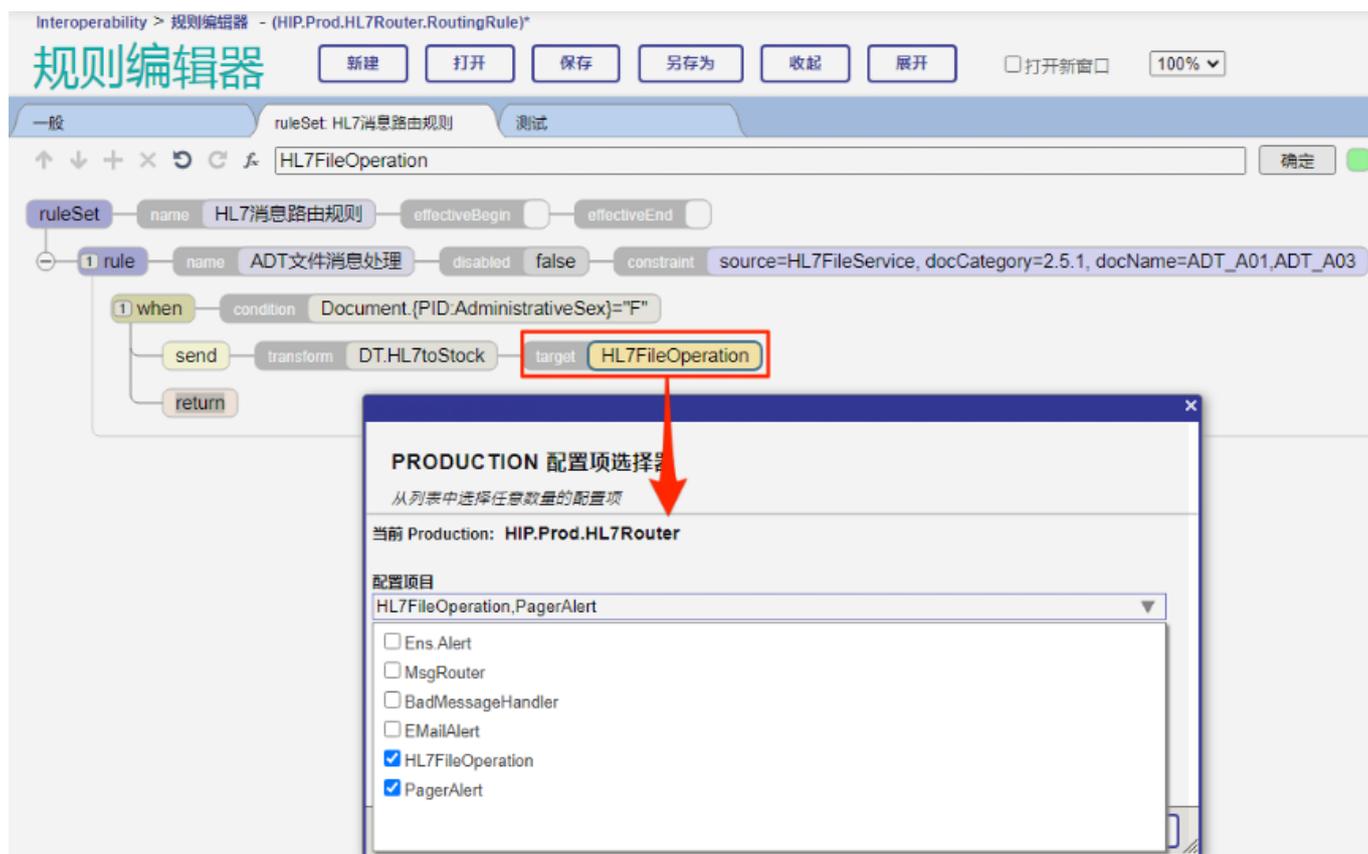
4. 行为(action)定义了满足条件时的行为。

例如满足条件，将消息做模型转换后发送给外部HL7接口。在右侧的**规则辅助**窗口列出可以执行的行为。注意在同一个**规则条件**下，可以有多个行为，例如向多个目标发送不同的消息。

对于路由规则，通常的行为是**发送**(send)。发送前，选中**转换**(transform)可以选择发送前要执行的数据转换；



发送目标(target)会列出production中的所有业务操作和业务流程，可以多选。



注意1：规则(rule)也是一种行为，即满足条件时，可以执行子规则。

注意2：返回(return)行为会中断当前规则条件下的后续行为。

<<未完待续>>

[#业务规则](#) [#持续集成](#) [#提示和技巧](#) [#Ensemble](#) [#HealthShare](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

源

URL:

<https://cn.community.intersystems.com/post/intersystems%E4%BA%92%E6%93%8D%E4%BD%9C%E8%BF%9B%E9%98%B6-%E7%AC%AC%E4%BA%8C%E7%AF%87%E7%BC%9A%E8%A7%84%E5%88%99%E5%BC%95%E6%93%8E-%E7%BC%88%E7%AC%AC%E4%B8%80%E9%83%A8%E5%88%86%E7%BC%89>