

文章

[Qiao Peng](#) · 三月 29, 2021 阅读大约需 4 分钟

CDC系列之四：使用DSTIME特性在InterSystems IRIS/Caché上实现CDC功能

前面介绍了通过mirroring或shadow，使用journal日志过滤器的方式，在不改动数据模型的情况下实现InterSystems IRIS/Caché上的CDC能力。

但如果你可以修改InterSystems IRIS/Caché上的数据模型，也可以考虑使用DSTIME特性实现变更数据捕获。

DSTIME特性

DSTIME特性是InterSystems IRIS/Caché的嵌入式实时BI工具DeepSee用于跟踪数据变更的。

InterSystems IRIS和2011版之后的Caché，都支持DSTIME特性。它会自动记录数据库中SQL表记录或持久化对象的变更，并将变更记录写入持久化的多维数组^OBJ.DSTIME中。

DSTIME特性是针对SQL表/持久化类的，因此需要为每个需要记录变化的表/持久化类开启。

当开启了DSTIME的表/持久化类，有记录插入、更新、删除时，InterSystems

IRIS/Caché引擎会自动在^OBJ.DSTIME中记录这些操作。其格式为：

^OBJ.DSTIME(类名,DSTIME,对象ID) = 执行的操作代码

DSTIME:

当SQL表/持久化类的参数DSINTERVAL未被设置时，DSTIME=0；

当SQL表/持久化类的参数DSINTERVAL被设置时，DSTIME=1840/12/31午夜12点到记录发生时的秒数。

执行的操作代码：

代码	操作
0	更新
1	插入
2	删除

例如：

`^OBJ.DSTIME(" Demo.Patient ",5673588714,2) = 1`

`^OBJ.DSTIME(" Demo.Patient ",5673588735,3) = 1`

`^OBJ.DSTIME(" Demo.Patient ",5673588784,4) = 1`

`^OBJ.DSTIME(" Demo.Patient ",5673588791,2) = 0`

因此，如果你没有使用DeepSee，就可以监控^OBJ.DSTIME即可获得IRIS/Caché的数据变更。

注：DeepSee在同步cube时，会自动查询、操作、删除^OBJ.DSTIME的数据，因此，将会对于直接使用它进行数据变更捕获产生影响。

如果你的确在使用DeepSee自动同步cube，也想使用^OBJ.DSTIME实现CDC，请联系InterSystems获得解决方案。这不在本文的讨论范围。

开启DSTIME特性

InterSystems IRIS/Caché可以在SQL表建模时、对象模型上开启DSTIME特性。

涉及2个类参数：DSTIME 和DSINTERVAL

DSTIME 类参数设置为AUTO时，InterSystems IRIS/Caché会自动记录对应SQL表/持久化类的数据变更。

DSINTERVAL类参数控制记录的数据变更如何分组。用于实现CDC时，建议设置该参数=1。

1. 通过SQL建模，新建SQL表

在使用SQL新建表时，可以增加对于类参数DSTIME 和DSINTERVAL的设置。

例如下面的SQL创建一个开启了DSTIME的表Demo.Patient：

```
CREATE TABLE Demo.Patient (
    %CLASSPARAMETER DSTIME = 'AUTO',
    EMPNUM      INT NOT NULL,
    NAMELAST    CHAR(30) NOT NULL,
    NAMEFIRST   CHAR(30) NOT NULL,
    CONSTRAINT EMPLOYEEPK PRIMARY KEY (EMPNUM))
```

其中%CLASSPARAMETER DSTIME = 'AUTO'开启表的DSTIME特性。

2. 通过面向对象建模，修改持久化的对象模型

对于对象模型，在建模时或之后修改模型，设置DSTIME类参数的方式开启：Parameter DSTIME = "AUTO"; :

```
Class Demo.Patient Extends %Persistent [ ClassType = persistent, DdlAllowed, Final, Owner = {_SYSTEM}, ProcedureBlock, SqlRowIdPrivate, SqlTableName = PATIENT ]
{
    Property NAME As %Library.String(MAXLEN = 1) [ SqlColumnNumber = 2 ];
    Property DOB As %Library.Date [ SqlColumnNumber = 3 ];
    Parameter USEEXTENTSET = 1;
    Parameter DSTIME = "AUTO";
    Parameter DSINTERVAL = 1;
}
```

3. 对批量对象模型增加DSTIME特性

如果想对整个SQL Schema下对所有表或一个包下的所有对象类增加DSTIME，也可以定义一个持久化的纯虚基类，在基类中定义DSTIME类参数，然后让所有的类继承于这个基类。示例代码如下：

定义基类：

```
Class Demo.Super Extends %Persistent [ Abstract, NoExtent ]
{
    Parameter DSTIME = "AUTO";
}
```

注意，基类应该设置NoExtent关键字，以避免编译器生成存储模型。

修改子类，使Demo.Super成为其主父类：

```
Class Demo.Patient Extends Demo.Super [ Language = objectscript ]
{
    Property NAME As %Library.String(MAXLEN = 100) [ SqlColumnNumber = 2 ];
```

```
Property DOB As %Library.Date [ SqlColumnNumber = 3 ];
```

测试

开启DSTIME后，就可以使用SQL或对象操作方式，在InterSystems IRIS/Caché中执行一些插入、更新、删除操作，并观察持久化多维数组^OBJ.DSTIME的记录。

总结

通过开启DSTIME特性，让InterSystems IRIS/Caché自动记录需要追踪的SQL表或持久化类的数据变更，可以方便的通过^OBJ.DSTIME捕获这些变更，并处理数据变更或将变更记录输出到下游系统。

其它注意事项：

应考虑在处理/输出完数据变更后，记录上次处理的DSTIME值，或删除^OBJ.DSTIME中已处理过的记录，以方便持续的、高效的捕获新的数据变更。

CDC系列

更多的CDC选项实现，请参考：

- [1. CDC系列之一：使用Dejournal Filter在InterSystems IRIS/Caché上通过Mirroring实现CDC功能](#)
- [2. CDC系列之二：使用Dejournaling filter routine在Caché上通过Shadow实现CDC](#)
- [3. CDC系列之三：建立InterSystems IRIS/Caché的Global数据变更与SQL表记录的对应关系](#)
- [4. CDC系列之四：使用DSTIME特性在InterSystems IRIS/Caché上实现CDC功能](#)

#Caché #Ensemble #InterSystems IRIS #InterSystems IRIS for Health

源

URL:

<https://cn.community.intersystems.com/post/cdc%E7%B3%BB%E5%88%97%E4%B9%8B%E5%9B%9B%EF%BC%9A%E4%BD%BF%E7%94%A8dstime%E7%89%B9%E6%80%A7%E5%9C%A8intersystems-iriscach%C3%A9%E4%B8%8A%E5%AE%9E%E7%8E%B0cdc%E5%8A%9F%E8%83%BD>