文章
Nicky Zhu · 五月 20, 2021 阅读大约需 7 分钟

互操作消息统一管理系列：SearchTable加速检索

在上一篇文章《互操作消息统一管理系列：Message Bank》中，我们了解到在Message Bank中，消息均以半结构化（XML）或非结构化（Stream）的形式保存，因此无法与客户端的结构化消息一样，直接支持基于索引的检索。为此，需要在Message Bank中定义Search Table以支持查询。关于Search Table的定义和作用，请查阅https://docs.intersystems.com/healthconnectlatest/csp/docbook/DocBook.UI...。

一. 在Message Bank中查询消息的特殊之处

大家如果使用过消息查看器，则能够了解IRIS自动持久化消息并提供界面让大家能够根据消息头中（如发生事件、来源、目标等）或消息体中（如患者姓名、诊断名称等消息的具体属性）来查询消息。
而在Message Bank上进行查询时，如果直接使用消息查看器，能够查询的是Message Bank的Production中传输的消息，而若不是在源系统中出现的消息，这一点一定不要混淆。
Message Bank提供了消息仓库查看器供大家查询源系统中的消息

查询界面尽管在风格与功能上与消息查看器非常相似，但查询的目标是在Message Bank中转储的消息（存储格式为虚拟文档或字符流，见上一篇文章）。因此，查询索引与原来的消息截然不同，需要单独创建。其中，虚拟文档在IRIS中有原生支持，我们先来看自定义消息类型的处理过程。

先看这样一个案例。

二. 测试系统

假设我们有一个REST接口实现的添加患者业务，如下所示：

实现了REST接口，接收如下格式的POST请求，在该实例数据库中创建患者：

注意，从外部JSON传入的报文，在经过IRIS处理的过程中，会被转换为XML格式的消息，如下：

在被转储到Message Bank后，以同样结构的XML字符流存储

三. 在消息转储过程中创建索引

Message Bank中转储的消息默认是没有索引的，需要在消息转储过程中创建。

通过回调干预消息转储过程

Message Bank的Production中的服务Ens.Enterprise.MsgBank.TCPService提供了回调函数入口，使用户可以在消息转储过程中执行自定义的逻辑实现需求。因此，可用于触发对Search Table索引的建立过程。
要定义回调函数，则需要继承类Ens.Enterprise.MsgBank.BankHelperClass，并实现方法OnBankMsg，如下所示：

```
Class MessageBank.MsgSerializationHelper Extends Ens.Enterprise.MsgBank.BankHelperCla
ss [ Language = objectscript ]
{

/// ??IndexClassList??????????????????????
Parameter IndexClassList = "LIS.REST.MSG.PatientReq";

/// ????????
ClassMethod OnBankMsg(pHeader As Ens.Enterprise.MsgBank.MessageHeader, pFullHeaderID
As %String, pBody As %RegisteredObject = "", pFullBodyID As %String, pService As Ens.
Enterprise.MsgBank.TCPService) As %Status [ Language = objectscript ]
{
        Set tSC = $$$OK
    do
    {
        set clientBodyClassName = pHeader.ClientBodyClassName

        /// ????????????????????????????????
if ..#IndexClassList[clientBodyClassName
        {
            /// ???????Message Bank??????????????????????
            do pService.SendRequestAsync("MessageBank.Process.ReceiveBankMsgBPL", pHe
ader)
        }
    } while 0

    Quit $$$OK
}

}
```

注意，尽管整个建立Search Table的过程可以在服务中完成，但由于服务不能开启多进程，对于构建索引这样的高延迟任务，容易变成性能瓶颈。因此，强烈建议将消息转发给BP和BO，并启用多进程处理提高处理速度。

定义好回调函数和Helper类执行，在BS中引用即可，如下：

在Business Process中分发建立Search Table
建立索引的过程需要应用特定的Search Table类实现，因此通过Business Service将消息分发到Business Process后，则可通过BP根据消息的类型再调用特定的Search Table实现，如下：

即根据消息类型的不同，使用不同的逻辑进行索引。其中，对于本例的消息类型LIS.REST.MSG.PatientReq

在switch分支中判定后即可调用对应的代码段处理。

在该段代码中，我们需要使用Message Bank消息（该BP的request）中记录的MessageBodyId，对于在源系统中的结构化消息，在Message Bank中被记录为流，该Id即为流记录的Id。根据Id加载流对象后，将该流对象通过IndexDoc方法传递给这类消息对象专用的Search Table实现类完成索引的建立。

Search Table类的实现

该类需要继承Ens.CustomSearchTable并实现OnIndexDoc回调方法，如下：

```
Class MessageBank.SearchTable.Client.LISReqTable Extends Ens.CustomSearchTable [ Lang
uage = objectscript ]
{

///?Search Table????????????????????????????????
Property FullName As %String(MAXLEN = 36);

Index IdxFullName On FullName;

///??????Message Bank??????????????????????????????????????????????????
Parameter DOCCLASS = "%Stream.GlobalCharacter";

///?????????
ClassMethod OnIndexDoc(pDocObj As %Persistent, ByRef pSearchTable As MessageBank.Sear
chTable.Client.LISReqTable) As %Status
{
    set st = $$$OK
    do
    {
        set text = pDocObj.Read($$$MaxLocalLength)
        set text = $ZSTRIP(text, "*C",, $C(9,10,13,133))
///?????????????
        set vDoc = ##class(EnsLib.EDI.XML.Document).ImportFromString(text, .st) quit:
$$$ISERR(st)
///???????????????????Search Table????
        set pSearchTable.FullName = vDoc.GetValueAt("/PatientReq/FullName")

    } while 0

    if $$$ISERR(st) $$$LOGWARNING("Failed to index message LIS.REST.MSG.PatientReq:"_
pDocObj.%Id()_" "_st)

    quit $$$OK
}

}
```

应当注意的是，使用Search Table支持查询，则意味着查询的目标是Search Table对象而不是直接在Message Bank转储的消息上进行查询。平台的查询功能会负责维护查询逻辑，在Search Table上执行查询并关联到消息上去。
因此，Search Table的属性可以使用消息中的属性组合，大家也可以思考和改进本例实现这样一个场景，源系统中不传FullName全名，代之以分在两个属性中的姓和名。但在Message Bank中却可以在Search Table上建立全名FullName并建立索引，使用户可以通过全名查询消息。

还应当注意到的是，在解析字符流获取所需的属性值时，用户可以选择使用虚拟文档处理，如果能够获得原始消息的类定义，也可以将字符流转换为消息对象再取值。考虑到在实际的生产环境下在Message Bank并不能确保获取源消息定义，采用虚拟文档更为通用且不受类型限制，推荐大家使用。

四. 实现效果

定义好上述代码后启动Message Bank的Production，并测试源系统的接口：

此时查看Message Bank的production上的消息：

可看到在Message Bank有新消息触发了处理流程
通过SQL查询Search Table中的记录

则可看到最后一行中记录的全名
再转到消息仓库查看器，查询时选择查询表域，选择好对应的SearchTable类型，选择我们刚才使用的全名Biden

则可查询到消息和对应的Message Trace

对虚拟文档的处理过程

源系统的消息类型如为虚拟文档，例如EnsLib.EDI.XML.Document，则Message Bank的转储消息类型也是EnsLib.EDI.XML.Document，此时的处理流程与上述流程一致，但处理细节上会简单一些。体现在

MessageBank消息对象的获取

如上图所示，不直接使用字符流，通过Id直接获得虚拟文档对象即可。

Search Table的定义

```
Class MessageBank.SearchTable.DocSearchTable Extends EnsLib.EDI.XML.SearchTable [ Lan
guage = objectscript ]
{

XData SearchSpec [ XMLNamespace = "http://www.intersystems.com/EnsSearchTable" ]
{
<Items>
   <Item DocType="example:shiporder" PropName="OrderPerson" PropType="String:CaseSens
itive">{orderperson}</Item>
   <Item DocType=":root" PropName="d_e" PropType="String:CaseSensitive">{/root/d/e}</
Item>
   <Item DocType=":root" PropName="li_la_c" PropType="String:CaseSensitive">{/root/li
s/li}</Item>
</Items>
}

}
```

如上所示，EnsLib.EDI.XML.Document有系统原生的Search
Table模版EnsLib.EDI.XML.SearchTable，继承该类后在XData中定义好Search
Table的属性和取值路径（在消息XML中的路径）即可。
关于EnsLib.EDI.XML.Document的Search
Table，可通过https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls...获取更多信息。

#API #InterSystems 业务解决方案和架构 #互操作性 #企业消息库 #消息搜索 #监视 #系统管理 #InterSystems IRIS #InterSystems IRIS for Health

---

源URL: https://cn.community.intersystems.com/post/%E4%BA%92%E6%93%8D%E4%BD%9C%E6%B6%88%E6%81%AF%E7%BB%9F%E4%B8%80%E7%AE%A1%E7%90%86%E7%B3%BB%E5%88%97%EF%BC%9Asearchtable%E5%8A%A0%E9%80%9F%E6%A3%80%E7%B4%A2