**SEC系统EDI详细设计说明书**

**联想利泰软件有限公司 2014-4-1**

**更改记录：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 日期 | 修改章节 | 修改类型\* | 修改描述 | 修改人 | 版本 |
| 2014-3-31 | All | A | 新增 | 于大明 | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

\*修改类型分为 **A** – ADDED **M** – MODIFIED **D** – DELETED

# EDI设计

## 概述

SEC系统EDI，主要实现了SEC系统和分销商和物流商的系统的数据自动交换。SEC的EDI功能，主要是基于SQLServer的SSIS功能，结合存储过程实现。

对于SEC系统来说，EDI功能主要分为两部分，其一，生成EDI报文，通过AxWay中间传输平台，将报文发送给仓库3PL或物流商的系统；其二，接收通过AxWay中间平台传输过来的报文，并进行解析进入SEC数据库。

## SSIS包设计总体思路

SSIS包列表：

|  |  |  |  |
| --- | --- | --- | --- |
| 包名 | 包描述 | 执行次数 | out与In和对象 |
| CreateReplacementFiles.dtsx | EDI补发 | 1次/小时 | Out仓库（分销商,物流商） |
| ReadLogisticsStatusToSEC.dtsx | 物流反馈状态 | 1次/小时 | In SEC |
| ReadSNInfoToSEC.dtsx | WMSSN数据集成 | 1次/半小时 | In SEC |
| ReadCostFilesToSEC.dtsx | 费用反馈 | 1次/天 | In SEC |
| ReadOrderInfoToSEC.dtsx | EDI下单 | 1次/半小时 | In SEC |
| CreateCostInfoFiles.dtsx | 费用推送 | 1次/天 | Out仓库（分流商） |
| CreateStockInfoFiles.dtsx | 库存对账 | 1次/天 | Out仓库（分流商） |
| CreateDeliveryInfoFiles.dtsx | 交付状态 | 1次/6小时 | Out仓库（分流商） |
| CreateReturnInfoFiles.dtsx | 退货 | 1次/半小时 | Out仓库（物流商） |
| CreateSendOrderInFoFiles.dtsx | 出库 | 1次/1小时 | Out仓库（分销商,物流商） |
| CreateStorageInfoFiles.dtsx | 入库 | 1次/半小时 | Out仓库（分销商,物流商） |

SEC系统SSIS包共11个，

从报文的方向来看，分为两类：SEC生成的和SEC接收处理的；

从包的调度来看，也分为两类：系统应用程序实时调用的和SqlServer Job调度执行的。

### SECEDI报文生成机制

所有的EDI报文，都由对应的SSIS包运行生成。运行时间分为两种，即时和定时。

即时：用户在SEC系统中操作，某动作触发立即生成EDI报文；

定时：SEC数据库Job调度执行。

对于某报文是即时还是定时生成，在后文中包设计分述中有对应描述。（参见1.3）

### SEC生成报文

在上表中，标记为Out类型的为SEC生成的报文。

### SEC接收报文

在上表中，标记为In类型的为分销商或物流商生成，SEC系统接收处理的报文。

### 公共函数

-- Description: <输出订单关联错误信息>

-- =============================================

1，[dbo].[FN\_EDI\_GetErroMessage]

Description: <得到某一列的字符串>

-- =============================================

2,[dbo].[FN\_GetString](@ID VARCHAR(20))

### 公共存储过程

1. SP\_EDI\_SaveSSISRunRecord

功能描述：记录每个SSIS包每次的运行记录，包括运行开始时间、结束时间和成功状态

1. SP\_EDI\_GetSequenceByKeyword

功能描述：对不同的报文类型，生成文件名。

## 包设计分述

每个包中，有几个相同的步骤。

1. 获取GUID

功能描述：在SSIS包执行开始，先生成一个GUID，此GUID可以理解为当前包此次运行时对应的实例ID，在包运行过程中，更新数据和包执行记录都会依赖此ID。

1. 记录包运行日志中开始执行时间

功能描述：包开始执行时，先记录日志。

1. 获取参数

功能描述：获取生成报文时需要的参数，包括获取文件路径（从表[SYS\_Param]中读取，包括EDI报文Out和In的文件目录）、数据时间范围等。在各SSIS包中获取的参数不一样，但都会这样一个过程。

1. 创建Out文件夹下供应商对应的Pending和Processed目录

功能描述：原则上，供应商对应的此2个目录，是在系统开始就应该存在的，此目录将会配置到Axway中。但为了SSIS的稳定性和健壮性，加入了此判断，如果对应目录不存在，将会自动创建。

1. 生成文件名

功能描述：根据获取的参数、报文类型、序列号和当前的仓库3PLCode或物流商Code，生成EDI报文名称。

1. 生成EDI报文

功能描述：根据当前参数，查询数据生成EDI报文，将数据按照报文定义的规则，写入CSV文件。

1. SSIS成功处理

当SSIS运行成功后，记录成功日志。

1. SSIS失败处理

当SSIS运行失败时，记录失败日志。

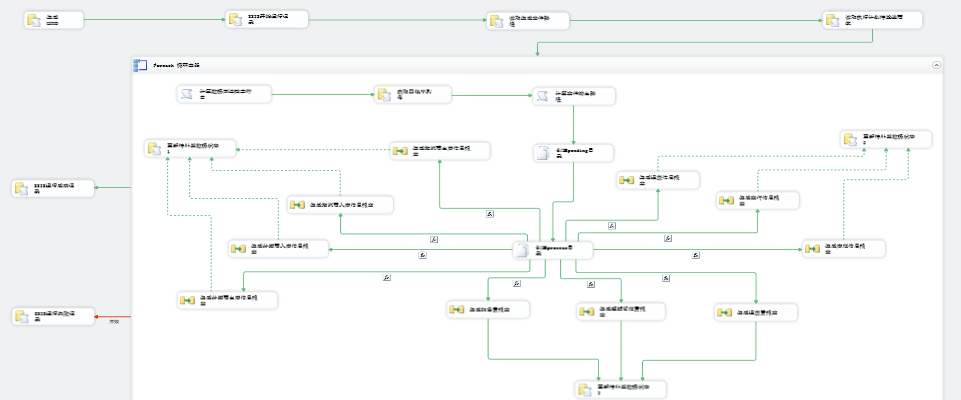
### CreateReplacementFiles EDI补发

#### 包名称

CreateReplacementFiles.dtsx

#### 运行时效

定时Job调度执行。1次/小时包逻辑描述



1. 读取执行计划待发送商家
2. SELECT [DS].[DBName],[DS].[DBIP],[DS].[UserID],[DS].[Password],[RP].[ID],[RC].[FileExtension],[RC].[BusinessType],[RP].[SourceCode],[RP].[ReplacementType],[RP].[OrderIDorDate],[RC].[ID],[RP].[FileName]

FROM [dbo].[EDI\_ReplacementPlan] AS [RP]

INNER JOIN [dbo].[EDI\_RouteCollocation] AS [RC] ON [RP].[SourceCode]=[RC].[SourceCode]

INNER JOIN [dbo].[DBSectors] AS [DS] ON [RC].[SectorCode] = [DS].[SectorCode]

WHERE [RP].[SendStatus]=0

1. 获取四位序列号

EXEC dbo.SP\_EDI\_GetSequenceByKeyword。

1. 生成报文后，记录订单发送日志

数据流任务

ReadLogisticsStatusToSEC 物流反馈状态

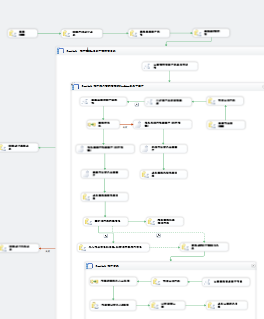
#### 包名称

ReadLogisticsStatusToSEC.dtsx

#### 运行时效

即时生成，1次/小时

#### 包逻辑描述



1，读取EDI物流商

SELECT DISTINCT [RC].[SourceCode],[RC].[FileExtension]

FROM [dbo].[DBSectors] AS [DS]

INNER JOIN [dbo].[EDI\_RouteCollocation] AS [RC] ON [DS].[SectorCode]=[RC].[SectorCode]

WHERE [BusinessType]=2

2，计算物流商文件夹及名称规范

3，初始化临时表

TRUNCATE TABLE [dbo].[Temp\_StateFeedbackTemp]

4，记录读取成功报文日志

INSERT INTO [dbo].[EDI\_ReadFileLog]

([SourceCode]

,[MessageName]

,[ExecutionTime]

,[ExecutionResult]

,[ISReplacement])

VALUES(?,?,GETDATE(),1,1)

5，更新临时表大区信息

UPDATE dbo.Temp\_StateFeedbackTemp

SET [DBIP]=DB.DBIP,

[DBName]=DB.DBName,

[UserID]=DB.UserID,

[DBPassword]=DB.[Password],

[SectorID]=DB.ID,

[FileNameAndPath]=?

FROM [dbo].[Temp\_StateFeedbackTemp] AS [TF]

INNER JOIN [dbo].[RDCs] AS [RDC] ON SUBSTRING([TF].BillNo,1, CHARINDEX('PSD',[TF].BillNo)-1)=[RDC].[RDCCode]

INNER JOIN [dbo].[SectorRDCs] AS [SRDC] ON [RDC].[ID]=[SRDC].[RDCID]

INNER JOIN [dbo].[DBSectors] AS [DB] ON ([SRDC].[SectorID]=[DB].[ID])

SELECT COUNT(1) FROM dbo.Temp\_StateFeedbackTemp WHERE ISNULL(SectorID,'')=''

6，汇总数据到主库临时表

INSERT INTO dbo.Temp\_StateFeedback

( SourceCode ,

BillNo ,

SignUser ,

SignDate ,

HappenTime ,

ExceptionCode ,

ExceptionName ,

ReasonOrAddr ,

PredictDelayDurationHour ,

GUID ,

CreateTime ,

CreateBy ,

UpdateTime ,

UpdateBy ,

Remark1 ,

Remark2 ,

Remark3 ,

Remark4 ,

Remark5 ,

DBIP ,

DBName ,

UserID ,

DBPassword ,

SectorID ,

FileNameAndPath

)

SELECT [SourceCode] ,

[BillNo] ,

[SignUser] ,

[SignDate] ,

[HappenTime] ,

[ExceptionCode] ,

[ExceptionName] ,

[ReasonOrAddr] ,

[PredictDelayDurationHour] ,

[GUID] ,

[CreateTime] ,

[CreateBy] ,

[UpdateTime] ,

[UpdateBy] ,

[Remark1] ,

[Remark2] ,

[Remark3] ,

[Remark4] ,

[Remark5] ,

[DBIP] ,

[DBName] ,

[UserID] ,

[DBPassword] ,

[SectorID] ,

[FileNameAndPath]

FROM [dbo].[Temp\_StateFeedbackTemp]

7，初始化临时表

TRUNCATE TABLE dbo.Temp\_StateFeedback

8，状态反馈计算 EXEC [dbo].[SP\_Feedback\_SECRDCFB]

9，记录计算结果日志

INSERT INTO [dbo].[Temp\_StateFeedbackLog]

(

[SourceCode]

,[BillNo]

,[SignUser]

,[SignDate]

,[HappenTime]

,[ExceptionCode]

,[ExceptionName]

,[ReasonOrAddr]

,[PredictDelayDurationHour]

,[GUID]

,[Status]

,[ErroMessage]

,[ISReplacement]

,[Remark1]

,[Remark2]

,[Remark3]

,[Remark4]

,[Remark5])

SELECT

[SF].[SourceCode]

,[SF].[BillNo]

,[SF].[SignUser]

,[SF].[SignDate]

,[SF].[HappenTime]

,[SF].[ExceptionCode]

,[SF].[ExceptionName]

,[SF].[ReasonOrAddr]

,[SF].[PredictDelayDurationHour]

,[SF].[GUID]

,[FB].[IsUpdate]

,[FB].[Remark]

,[FB].[IsUpdate] AS [ISReplacement]

,[SF].[Remark1]

,[SF].[Remark2]

,[SF].[Remark3]

,[SF].[Remark4]

,[SF].[Remark5]

FROM [dbo].[Temp\_StateFeedback] AS [SF]

INNER JOIN [dbo].[Temp\_FBInfos] AS [FB] ON [SF].[ID]=[FB].[EDIGUID]

### ReadSNInfoToSEC WMSSN数据集成

#### 包名称

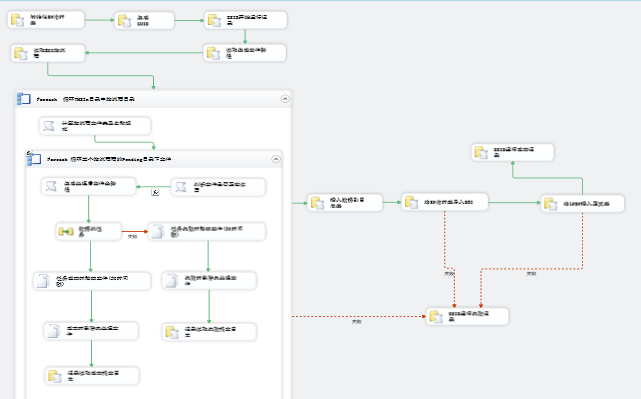
ReadSNInfoToSEC.dtsx

#### 运行时效

即时生成。1次/半小时。

#### 包逻辑描述

此SSIS包和TransPlanSecond报文为同一个包设计。SSIS包内会根据运单号是否已经生成过提交报文，自动生成二次提交报文。



1，读取生成文件路径

SELECT [FileReadPatch] FROM [dbo].[EDI\_FilesPatch]

2，读取EDI物流商

SELECT DISTINCT [ERC].[SourceCode],[ERC].[FileExtension]

FROM [dbo].[EDI\_RouteCollocation] AS [ERC]

INNER JOIN [dbo].[EDI\_MessageType] AS [MT]ON [ERC].[MessageTypeID] =[MT].[ID]

INNER JOIN [dbo].[DBSectors] AS [DS] ON [ERC].[SectorCode] = [DS].[SectorCode]

WHERE [MT].[MessageTypeCode]=70 AND [ERC].[BusinessType]=2

3，插入数据到日志表

INSERT INTO dbo.Temp\_WMSSNLog

( OrdNo ,

SapNo ,

SapSeq ,

InvoiceNo ,

PN ,

SN ,

BarCode ,

ScanTime ,

AdjPN ,

[FileName] ,

CreateTime ,

CreateBy

)

SELECT OrdNo ,

SapNo ,

SapSeq ,

InvoiceNo ,

PN ,

SN ,

BarCode ,

ScanTime ,

AdjPN ,

REPLACE([FileName],'Pending','Processed'),

CreateTime ,

CreateBy

FROM dbo.Temp\_WMSSN

4，将SN临时表导入SEC

------------清除临时表里面的在目标表里面scanState不为1的数目------------

DELETE [Temp\_WMSSN]

FROM [Temp\_WMSSN] AS [TP]

INNER JOIN [dbo].[WMSSNs] AS [WN] ON ( [TP].BarCode = [WN].BarCode

AND [TP].SapNo = [WN].SapNo

AND [TP].SapSeq = [WN].SapSeq

AND [TP].InvoiceNo = [WN].InvoiceNo

AND [WN].ScanState != 1

)

-----------------------------------------------------------------------------------

-----------------------插入更新数据-----------------------------------------------

MERGE INTO dbo.[WMSSNs] AS [t0]

USING

( SELECT DISTINCT [OrdNo] ,

[SapNo] ,

[SapSeq] ,

[InvoiceNo] ,

[PN] ,

[SN] ,

[BarCode] ,

[AdjPN] ,

1 AS [ScanState] ,

CreateTime ,

CreateBy ,

GETDATE() AS UpdateTime,

'EDI' AS UpdateBy

FROM [Temp\_WMSSN] AS [TP]

WHERE ISNULL([TP].[OrdNo], '') != ''

AND ISNULL([TP].[SapNo], '') != ''

AND ISNULL([TP].[SapSeq], '') != ''

AND ISNULL([TP].[PN], '') != ''

AND ISNULL([TP].[SN], '') != ''

AND ISNULL([TP].[BarCode], '') != ''

) AS [t1]

ON ( [t0].BarCode = [t1].BarCode

AND [t0].SapNo=[t1].SapNo --根据这四个维度

AND [t0].SapSeq=[t1].SapSeq

AND [t0].InvoiceNo=[t1].InvoiceNo

AND [t0].ScanState =1

)

WHEN MATCHED

THEN

UPDATE SET

[t0].[OrdNo] = [t1].[OrdNo] ,

[t0].[SapNo] = [t1].[SapNo] ,

[t0].[SapSeq] = [t1].[SapSeq] ,

[t0].[InvoiceNo] = [t1].[InvoiceNo] ,

[t0].[PN] = [t1].[PN] ,

[t0].[SN]=[t1].[SN],

[t0].[AdjPN] = [t1].[AdjPN] ,

[t0].[UpdateBy] = [t1].UpdateBy ,

[t0].[UpdateTime] = [t1].[UpdateTime]

WHEN NOT MATCHED

THEN

INSERT VALUES

( [t1].[OrdNo] ,

[t1].[SapNo] ,

[t1].[SapSeq] ,

[t1].[InvoiceNo] ,

[t1].[PN] ,

[t1].[SN] ,

[t1].[BarCode] ,

NULL,

[t1].[AdjPN] ,

[t1].[ScanState],

CreateTime ,

CreateBy ,

NULL,

NULL

);

5，将lNSN插入正式表

exec SP\_Link\_InsertRecordLNAndSNs

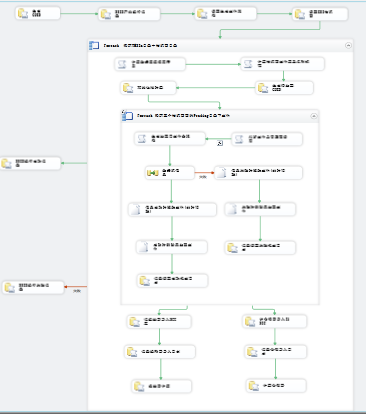
### ReadCostFilesToSEC 费用反馈包名称

ReadCostFilesToSEC.dtsx

#### 运行时效

即时生成。1次/天

#### 包逻辑描述



1，读取EDI物流商

SELECT DISTINCT [SourceCode],[FileExtension]

FROM [dbo].[EDI\_RouteCollocation]

WHERE [BusinessType]=2

2，初始化临时表

TRUNCATE TABLE [dbo].[Temp\_CostFeedback]

3，将运送费导入SCE库

INSERT INTO [dbo].[Fee\_TranFeeFeedbackUploadData]

([FeeResultID]

,[BillID]

,[BillCode]

,[FeeSource]

,[FeeItem]

,[RDC]

,[LogisticProvider]

,[LegalPerson]

,[TransportMode]

,[FeedbackAmount]

,[Total]

,[Volume]

,[Weight]

,[FeePackingNum]

,[ShipTime]

,[ServiceLevel]

,[County]

,[UploadDate]

,[IsProcessed]

,[Uploader]

,[UploadGUID]

,[CreateTime]

,[CreateBy])

SELECT

[TFR].[ID]

,[TFR].[BillID]

,[TP].[BillCode]

,[FS].[FeeSourceName]

,[FI].[FeeItemName]

,[RDC].[RDCName]

,[LP].[LogisticProviderName]

,[LE].[LegalPersonName]

,[TM].[TransportModeName]

,[TP].[FeedbackAmount]

,[SD].[Total]

,[SD].[Volume]

,[SD].[Weight]

,[SD].[FeePackingNum]

,[SD].[ConfirmTime] AS [ShipTime]

,[SL].[ServiceLevelName]

,[CT].[CountyName]

,GETDATE()

,0

,'EDI'

,[TP].[GUID]

,GETDATE()

,'EDI'

FROM [dbo].[Temp\_CostFeedback] AS [TP]

INNER JOIN [dbo].[Fee\_TranFeeResult] AS [TFR] ON [TP].[BillCode]=[TFR].[BillCode] AND [TP].[FeeSourceID]=[TFR].[FeeSourceID] AND [TP].[FeeItemID]=[TFR].[FeeItemID]

INNER JOIN [dbo].[Fee\_FeeSource] AS [FS] ON [TFR].[FeeSourceID]=[FS].[ID]

INNER JOIN [dbo].[Fee\_FeeItem] AS [FI] ON [TFR].[FeeItemID]=[FI].[ID]

INNER JOIN [dbo].[RDCs] AS [RDC] ON [TFR].[RDCID]=[RDC].[ID]

INNER JOIN [dbo].[LogisticProviders] AS [LP] ON [TFR].[LogisticProviderID]=[LP].[ID]

INNER JOIN [dbo].[LegalPersons] AS [LE] ON [TFR].[LegalPersonID]=[LE].[ID]

INNER JOIN [dbo].[SECDistributions] AS [SD] ON [TP].[BillCode]=[SD].[DistributionNo]

INNER JOIN [dbo].[TransportModes] AS [TM] ON [SD].[TransportModeID]=[TM].[ID]

INNER JOIN [dbo].[ServiceLevels] AS [SL] ON [SD].[ServiceLevelID]=[SL].[ID]

INNER JOIN [dbo].[Counties] AS [CT] ON [SD].[CountyID]=[CT].[ID]

WHERE [TP].[FeeTypeID]=1

4，讲仓储费导入到SEC

INSERT INTO [dbo].[Fee\_StorageFeeFeedbackUploadData]

([FeeResultID]

,[BillID]

,[BillCode]

,[FeeSource]

,[FeeItem]

,[RDC]

,[LogisticProvider]

,[LegalPerson]

,[TransportMode]

,[FeedbackAmount]

,[Total]

,[Volume]

,[Weight]

,[FeePackingNum]

,[ShipTime]

,[ServiceLevel]

,[County]

,[UploadDate]

,[IsProcessed]

,[Uploader]

,[UploadGUID]

,[CreateTime]

,[CreateBy])

SELECT

[TFR].[ID]

,[TFR].[BillID]

,[TP].[BillCode]

,[FS].[FeeSourceName]

,[FI].[FeeItemName]

,[RDC].[RDCName]

,[LP].[LogisticProviderName]

,[LE].[LegalPersonName]

,[TM].[TransportModeName]

,[TP].[FeedbackAmount]

,[SD].[Total]

,[SD].[Volume]

,[SD].[Weight]

,[SD].[FeePackingNum]

,[SD].[ConfirmTime] AS [ShipTime]

,[SL].[ServiceLevelName]

,[CT].[CountyName]

,GETDATE()

,0

,'EDI'

,[TP].[GUID]

,GETDATE()

,'EDI'

FROM [dbo].[Temp\_CostFeedback] AS [TP]

INNER JOIN [dbo].[Fee\_StorageFeeResult] AS [TFR] ON [TP].[BillCode]=[TFR].[BillCode] AND [TP].[FeeSourceID]=[TFR].[FeeSourceID] AND [TP].[FeeItemID]=[TFR].[FeeItemID]

INNER JOIN [dbo].[Fee\_FeeSource] AS [FS] ON [TFR].[FeeSourceID]=[FS].[ID]

INNER JOIN [dbo].[Fee\_FeeItem] AS [FI] ON [TFR].[FeeItemID]=[FI].[ID]

INNER JOIN [dbo].[RDCs] AS [RDC] ON [TFR].[RDCID]=[RDC].[ID]

INNER JOIN [dbo].[LogisticProviders] AS [LP] ON [TFR].[LogisticProviderID]=[LP].[ID]

INNER JOIN [dbo].[LegalPersons] AS [LE] ON [TFR].[LegalPersonID]=[LE].[ID]

INNER JOIN [dbo].[SECDistributions] AS [SD] ON [TP].[BillCode]=[SD].[DistributionNo]

INNER JOIN [dbo].[TransportModes] AS [TM] ON [SD].[TransportModeID]=[TM].[ID]

INNER JOIN [dbo].[ServiceLevels] AS [SL] ON [SD].[ServiceLevelID]=[SL].[ID]

INNER JOIN [dbo].[Counties] AS [CT] ON [SD].[CountyID]=[CT].[ID]

WHERE [TP].[FeeTypeID]=2

5，运送费计算 EXEC [dbo].[SP\_TranFee\_FeedbackUploadDataProcess] ?

6, 计算仓储费EXEC [dbo].[SP\_StorageCosts\_FeedbackUploadDataProcess]

### ReadOrderInfoToSEC.dtsx EDI下单

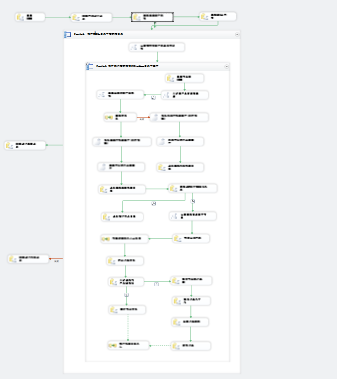
#### 包名称

ReadOrderInfoToSEC.dtsx

#### 运行时效

即时生成 1次/半小时。

#### 包逻辑描述



1，读取EDI分销商

SELECT DISTINCT [RC].[SourceCode],[RC].[FileExtension]

FROM [dbo].[DBSectors] AS [DS]

INNER JOIN [dbo].[EDI\_RouteCollocation] AS [RC] ON [DS].[SectorCode]=[RC].[SectorCode]

WHERE [BusinessType]=1

2，获取该RDC所在DB服务器

SELECT [DB].[DBIP],[DB].[DBName],[DB].[UserID],[DB].[Password],[DB].[ID]

FROM [dbo].[Temp\_EDIOrdersTemp] AS [TE]

INNER JOIN [dbo].[RDCs] AS [RDC] ON [TE].[RDCCode]=[RDC].[RDCCode]

INNER JOIN [dbo].[SectorRDCs] AS [SRDC] ON [RDC].[ID]=[SRDC].[RDCID]

INNER JOIN [dbo].[DBSectors] AS [DB] ON [SRDC].[SectorID]=[DB].[ID]

WHERE [TE].[BatchGUID]=?

3, 拆分订单信息

INSERT INTO [dbo].[Temp\_EDIOrders]

([BizTypeID]

,[DeliveryStandardID]

,[ResellerCode]

,[AgencyCode]

,[AgencyAddressCode]

,[RDCCode]

,[ChannelNo]

,[PayMentID]

,[PaymentAmount]

,[Remark]

,[ProductGroupNo]

,[SECAgentOrdNo]

,[BatchGUID])

SELECT TOP 1 [BizTypeID]

,[DeliveryStandardID]

,[ResellerCode]

,[AgencyCode]

,[AgencyAddressCode]

,[RDCCode]

,[ChannelNo]

,[PayMentID]

,[PaymentAmount]

,[Remark]

,[ProductGroupNo]

,[SECAgentOrdNo]

,[BatchGUID]

FROM [dbo].[Temp\_EDIOrdersLog]

WHERE [BatchGUID]=?

---------子表数据----------

INSERT INTO [dbo].[Temp\_EDIOrderDetails]

([OrderID]

,[ProductCode]

,[BizTypeID]

,[ProductGroupNo]

,[ChannelNo]

,[RDCCode]

,[ResellerCode]

,[TMSOrdNo]

,[TMSAgentOrdNo]

,[OrdSign]

,[LO].[LogID]

,[LO].[Total])

SELECT

[SO].[ID] AS [OrderID]

,[LO].[ProductCode]

,[LO].[BizTypeID]

,[LO].[ProductGroupNo]

,[LO].[ChannelNo]

,[LO].[RDCCode]

,[LO].[ResellerCode]

,[LO].[TMSOrdNo]

,[LO].[TMSAgentOrdNo]

,[LO].[OrdSign]

,[LO].[GUID]

,[LO].[Total]

FROM [dbo].[Temp\_EDIOrdersLog] AS [LO]

INNER JOIN [dbo].[Temp\_EDIOrders] AS [SO] ON [SO].[BatchGUID]=[LO].[BatchGUID]

WHERE [LO].[BatchGUID]=?

4,判断通路和产品组权限

exec SP\_EDI\_ValidateChannelNo

5, 查询待处理订单ID

SELECT [ID] FROM [dbo].[Temp\_EDIOrders] WHERE [BatchGUID]=?

6, 处理订单数据

EXEC [dbo].[SP\_EDI\_InsertOrdersInfo]

7,　 审核订单 EXEC [dbo].[SP\_EDI\_CheckOrderApproval]

### CreateCostInfoFiles 费用推送

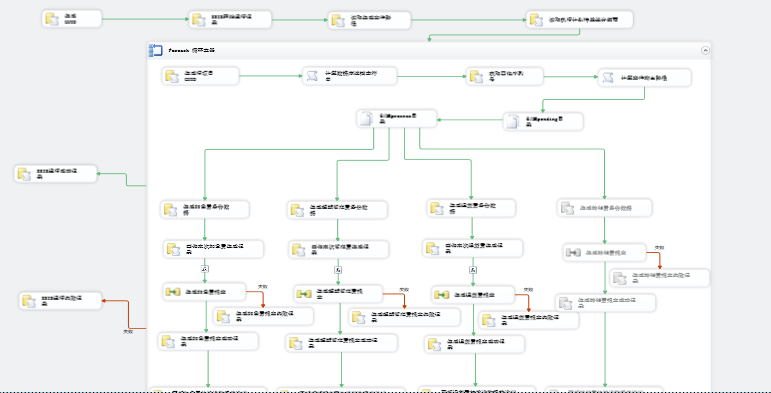
#### 包名称

CreateCostInfoFiles.dtsx

#### 运行时效

即时生成, 1次/天

#### 包逻辑描述



1, 生成行项目GUID SELECT CONVERT(VARCHAR(36),NEWID())

2, 获取四位序列号

EXEC dbo.SP\_EDI\_GetSequenceByKeyword

3, 生成退货费备份数据

EXEC dbo.SP\_EDI\_InsertReturnFeesByReseller

4, 生成超期暂存费备份数据

EXEC dbo.SP\_EDI\_InsertOverdueFeesByReseller

5, 生成加急费备份数据

EXEC dbo.SP\_EDI\_InsertExpeditedFeesByResellers

6, 生成退货费报文成功记录

INSERT INTO [dbo].[EDI\_SendFileLog]

([SourceCode]

,[MessageName]

,[ExecutionTime]

,[ExecutionResult])

VALUES(?,?,GETDATE(),1)

7,

### CreateStockInfoFiles库存对账

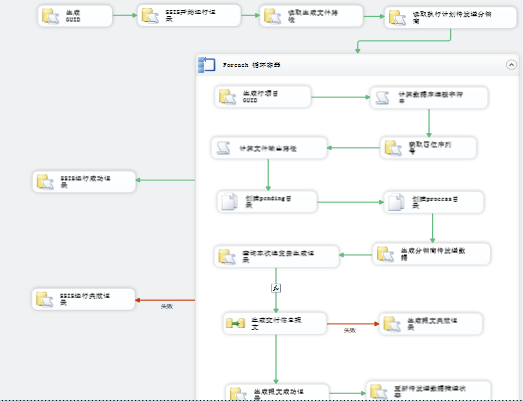
#### 包名称

CreateStockInfoFiles.dtsx

#### 运行时效

及时执行。1次/天。

#### 包逻辑描述



1. 获取四位序列号 EXEC dbo.SP\_EDI\_GetSequenceByKeyword
2. 生成分销商待发送数据 EXEC dbo.SP\_EDI\_InsertStockInfoByReseller
3. 查询本次退货费生成记录 SELECT

COUNT(ID)

FROM [dbo].[EDI\_ReturnFees]

WHERE [GUID]=?

4, 更新待发送数据推送状态 UPDATE [dbo].[EDI\_StockInformation] SET [SendStatus]=1 WHERE [GUID]=?

### CreateDeliveryInfoFiles 交付状态

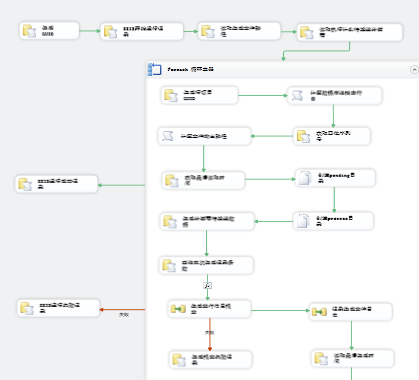
#### 包名称

CreateDeliveryInfoFiles.dtsx

#### 运行时效

Job调度执行。1次/6小时

#### 包逻辑描述



1, 读取执行计划待发送分销商 SELECT [DS].[DBName],[DS].[DBIP],[DS].[UserID],[DS].[Password],[ERC].[ID],[ERC].[SourceCode],[ERC].[FileExtension],[ERC].[BusinessType]

FROM [dbo].[EDI\_RouteCollocation] AS [ERC]

INNER JOIN [dbo].[EDI\_MessageType] AS [MT]ON [ERC].[MessageTypeID] =[MT].[ID]

INNER JOIN [dbo].[DBSectors] AS [DS] ON [ERC].[SectorCode] = [DS].[SectorCode]

WHERE [MT].[MessageTypeCode]=40 AND [ERC].[BusinessType]=1

2，生成行项目GUID SELECT CONVERT(VARCHAR(36),NEWID())

3，获取四位序列号 EXEC dbo.SP\_EDI\_GetSequenceByKeyword

4，生成分销商待发送数据 EXEC dbo.SP\_EDI\_InsertDeliveryInfoByReseller

5，查询本次生成记录条数 SELECT

COUNT(ID)

FROM [dbo].[EDI\_DeliveryInformation]

6，读取最后生成时间 SELECT (CASE WHEN MAX([OperateTime]) IS NULL THEN ? ELSE MAX([OperateTime])END) FROM [dbo].[EDI\_DeliveryInformation]

7，更新最后生成时间 UPDATE [dbo].[EDI\_Sequence] SET

[LastReadTime]=?

WHERE [BusinessID]=? AND [SourceType]=40

8，更新待发送数据推送状态 UPDATE [dbo].[EDI\_DeliveryInformation] SET [SendStatus]=1

### CreateReturnInfoFiles.dtsx退货

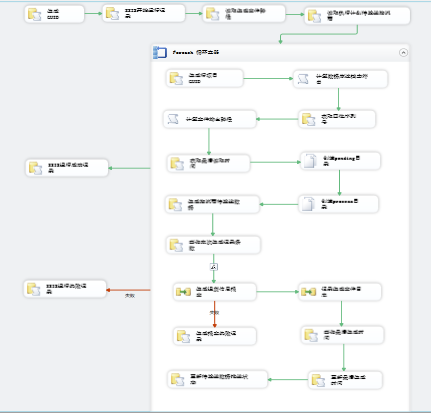
#### 包名称

CreateReturnInfoFiles.dtsx

#### 运行时效

Job调度执行。1次/半小时。

#### 包逻辑描述



1，读取执行计划待发送物流商

SELECT [DS].[DBName],[DS].[DBIP],[DS].[UserID],[DS].[Password],[ERC].[ID],[ERC].[SourceCode],[ERC].[FileExtension],[ERC].[BusinessType]

FROM [dbo].[EDI\_RouteCollocation] AS [ERC]

INNER JOIN [dbo].[EDI\_MessageType] AS [MT]ON [ERC].[MessageTypeID] =[MT].[ID]

INNER JOIN [dbo].[DBSectors] AS [DS] ON [ERC].[SectorCode] = [DS].[SectorCode]

WHERE [MT].[MessageTypeCode]=30 AND [ERC].[BusinessType]=2

2，获取四位序列号 EXEC dbo.SP\_EDI\_GetSequenceByKeyword

3，获取最后读取时间

SELECT CONVERT(VARCHAR,[LastReadTime],121) FROM [dbo].[EDI\_Sequence] WHERE [BusinessID]=? AND [SourceType]=30

4，生成物流商待发送数据 EXEC dbo.SP\_EDI\_InsertReturnInfoByLogisticProvider

5，查询本次生成记录条数 SELECT

COUNT(ID)

FROM [dbo].[EDI\_ReturnInformation]

6，更新最后生成时间

UPDATE [dbo].[EDI\_Sequence] SET

[LastReadTime]=?

WHERE [BusinessID]=? AND [SourceType]=30

7，更新待发送数据推送状态

UPDATE [dbo].[EDI\_ReturnInformation] SET [SendStatus]=1 WHERE [GUID]=?

### CreateSendOrderInFoFiles.dtsx 出库

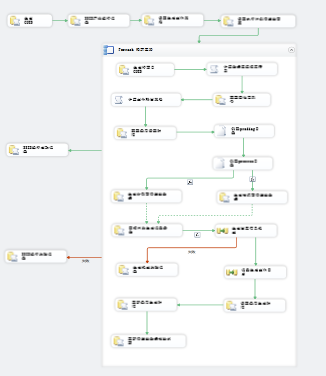
#### 包名称

CreateSendOrderInFoFiles.dtsx

#### 运行时效

Job调度执行。目前是1次/1小时。

#### 包逻辑描述



1，读取执行计划待发送商家 SELECT [DS].[DBName],[DS].[DBIP],[DS].[UserID],[DS].[Password],[ERC].[ID],[ERC].[SourceCode],[ERC].[FileExtension],[ERC].[BusinessType]

FROM [dbo].[EDI\_RouteCollocation] AS [ERC]

INNER JOIN [dbo].[EDI\_MessageType] AS [MT]ON [ERC].[MessageTypeID] =[MT].[ID]

INNER JOIN [dbo].[DBSectors] AS [DS] ON [ERC].[SectorCode] = [DS].[SectorCode]

WHERE [MT].[MessageTypeCode]=20

2，获取四位序列号 EXEC dbo.SP\_EDI\_GetSequenceByKeyword

3，获取最后读取时间 SELECT CONVERT(VARCHAR,[LastReadTime],121) FROM [dbo].[EDI\_Sequence] WHERE [BusinessID]=? AND [SourceType]=20

4，生成分销商待发送数据 EXEC dbo.SP\_EDI\_InsertDistributionInfoByReseller

5, 生成物流商待发送数据 EXEC dbo.SP\_EDI\_InsertDistributionInfoByLogisticProvider

6, 查询本次生成记录条数 SELECT

COUNT(ID)

FROM [dbo].[EDI\_DistributionInformation]

7, 更新待发送数据推送状态 UPDATE [dbo].[EDI\_DistributionInformation] SET [SendStatus]=1

### CreateStorageInfoFiles 入库

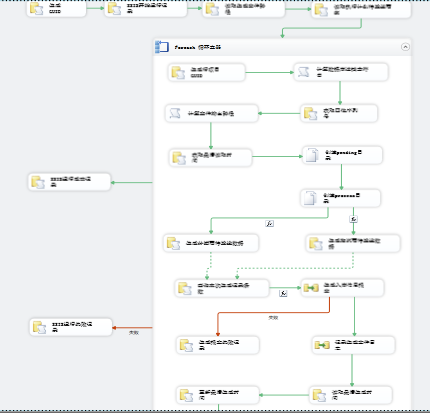
#### 包名称

CreateStorageInfoFiles.dtsx

#### 运行时效

Job调度执行。1次/半小时

#### 包逻辑描述



1, 读取执行计划待发送商家 SELECT [DS].[DBName],[DS].[DBIP],[DS].[UserID],[DS].[Password],[ERC].[ID],[ERC].[SourceCode],[ERC].[FileExtension],[ERC].[BusinessType]

FROM [dbo].[EDI\_RouteCollocation] AS [ERC]

INNER JOIN [dbo].[EDI\_MessageType] AS [MT]ON [ERC].[MessageTypeID] =[MT].[ID]

INNER JOIN [dbo].[DBSectors] AS [DS] ON [ERC].[SectorCode] = [DS].[SectorCode]

WHERE [MT].[MessageTypeCode]=10

2, 获取四位序列号 EXEC dbo.SP\_EDI\_GetSequenceByKeyword

3, 获取最后读取时间 SELECT CONVERT(VARCHAR,[LastReadTime],121) FROM [dbo].[EDI\_Sequence] WHERE [BusinessID]=? AND [SourceType]=10

4, 生成分销商待发送数据 EXEC dbo.SP\_EDI\_InsertStorageInfoByReseller

5, 生成物流商待发送数据 EXEC dbo.SP\_EDI\_InsertStorageInfoByLogisticProvider

6, 更新待发送数据推送状态 UPDATE [dbo].[EDI\_StorageInformation] SET [SendStatus]=1

# EDI SSIS部署

## 部署SSIS包

运行部署文件“edi.SPECt”，将SSIS包以文件形式部署在DB服务器上。

## 配置文件修改

打开配置文件SEC.dtsConfig。

修改以下值：

1. Data Source
2. User ID
3. Password
4. Initial Catalog

## 创建数据库Job并调度

根据1.3中运行时效描述，将Job调度执行的包，创建Job并进行调度。

## 修改EDI\_MeeageType

初始化该表，配置各个报文的类型

## 修改数据库表EDI\_FilePath中数据值

该表中存放的是配置参数，主要是FileGenerationPatch

和FileReadPatch

两个值，分别指EDI报文的存放位置。此两个值必须是和AxWay中的设置统一。

# EDI各记录查询

## 包执行记录及运行成功失败结果查询

SELECT \* FROM dbo.EDI\_SSISRunRecord

查询结果总，RunResult列为包对应的运行结果。凡是出现“0”的，为执行失败。多个“1”

## 推送的报文查询状态

SELECT \* FROM dbo.EDI\_SendFileLog

## EDI补发查询

SELECT \* FROM EDI\_ReplacementPlan

## SEC系统各报文初始化查询

ELECT \* FROM dbo.EDI\_Sequence

## SEC 读取报文日志记录

SELECT \* FROM dbo.EDI\_ReadFileLog

ExecutionResult 读取结果，ISReplacement 是否补发