2.11.86 277 Batch Outbound - 3382

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
3382	Change Order		EDI and Claim Capture			

2.11.86.1 Desired Solution
277 Batch Outbound

Please reference EDI Appendix M of the External Data Sharing DSD- BizTalk Documentation - 277 Batchl

2.11.86.2 Business Impact
None.

2.11.86.3 Technical Specifications
Please reference EDI Appendix M of the External Data Sharing DSD- BizTalk Documentation - 277 Batch

2.11.86.4 Clarifications
No associated clarifications found.

2.11.86.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.86.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.86.7 Change Order Status

Status	Date
SE Assigned	08/15/2006
Ready for Model Office	08/21/2006
Model Office Implemented	09/20/2006
UAT Implemented	11/01/2006
Prod Implemented	06/14/2007

2.11.87 T_TP_ALLOWED_TRANSPORT Mod - 3468

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
3468	Change Order		EDI and Claim Capture			

2.11.87.1 Desired Solution
T_TP_ALLOWED_TRANSPORT needs to be added and configured to support resolution of transport types per Trading Partner

2.11.87.2 Business Impact
None.

2.11.87.3 Technical Specifications
None.

2.11.87.4 Clarifications
No associated clarifications found.

2.11.87.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.87.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.87.7 Change Order Status

Status	Date
SE Assigned	08/24/2006
Model Office Implemented	11/21/2006
UAT Implemented	06/14/2007
Prod Implemented	06/14/2007

2.11.88 Encounter Processing - 3479

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
3479	Change Order		EDI and Claim Capture			

2.11.88.1 Desired Solution
This CO will encompass configuration of Encounter Processing.

2.11.88.2 Business Impact
None.

2.11.88.3 Technical Specifications
None.

2.11.88.4 Clarifications
No associated clarifications found.

2.11.88.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.88.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.88.7 Change Order Status

Status	Date
SE Assigned	08/25/2006
Model Office Implemented	11/21/2006
UAT Implemented	06/14/2007
Prod Implemented	06/14/2007

2.11.89 EDI DB BBS File Tracking MOD - 3542

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
3542	Change Order		EDI and Claim Capture			

2.11.89.1 Desired Solution
Changes required for EDI File Tracking and BBS tracking tables

2.11.89.2 Business Impact
None.

2.11.89.3 Technical Specifications
None.

2.11.89.4 Clarifications
No associated clarifications found.

2.11.89.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.89.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.89.7 Change Order Status

Status	Date
Ready for DM Review	08/31/2006
DM Review Board Approved	09/07/2006
Ready for Construction Wthru	09/07/2006
Model Office Implemented	11/21/2006
UAT Implemented	06/14/2007
Prod Implemented	06/14/2007

2.11.90 XMLHandler File Tracking - 3741

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
3741	Change Order		EDI and Claim Capture			

2.11.90.1 Desired Solution
Add File Tracking to XMLHandler, so XMLHandler logs tracking and exceptions in the same way as .NET components

2.11.90.2 Business Impact
None.

2.11.90.3 Technical Specifications
XMLHandler File Tracking will implement all the tracking points on the Unix side that already exist on the .NET side. This will ensure uniform tracking/logging from either environment.

2.11.90.4 Clarifications
No associated clarifications found.

2.11.90.5 Associated Requirements

Requirement ID	Type
30.090.016.002.1	EDI

2.11.90.6 Associated System Objects

Technical Name	Object Type	Title
edi_271O_4010	Program	Outbound Translation X12 - 271 Outbound Elig Response
edi_276I_4010	Program	Inbound Translation X12 - 276 Clm Stat Inquiry

2.11.90.7 Change Order Status

Status	Date
SE Assigned	09/15/2006
Construction in Progress	09/21/2006
Ready for Model Office	09/26/2006
Model Office Implemented	09/29/2006
Ready for Model Office	10/25/2006
Model Office Implemented	11/21/2006
UAT Implemented	06/14/2007
Prod Implemented	06/14/2007

2.11.91 FTS v.2.0 - 3824

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
3824	Change Order		EDI and Claim Capture			

2.11.91.1 Desired Solution

FTS v. 2.0. Includes:

1) Secure transfers for FTPS, SFTP
2) High-speed file transfers
3) Improved configuration
4) Common Watcher/Writer framework
5) Streamlined code
6) Integrated commercial FTP library edtFTPNetPRO

Accept for items 3 and 4 we have included all the above functionality in our current FTS version including the new FTP library edtFTPNetPRO. Item 3 will be an ongoing task and Item 4 is no longer a desirable component of our FTS solution.

Please reference Appendix C - File Transfer Service Documentation

2.11.91.2 Business Impact

None.

2.11.91.3 Technical Specifications

FTS v.2.0 is a complete rewrite of File Transfer Service (FTS) with common componentry, improved configuration, improved transfer speed, and improved threading design. Functionally, this version will be identical to FTS v.1.0, except for the added facility of being able to do SFTP file transfers.

Accept for items 3 and 4 we have included all the above functionality in our current FTS version including the new FTP library edtFTPNetPRO. Item 3 will be an ongoing task and Item 4 is no longer a desirable component of our FTS solution.

Please reference Appendix C - File Transfer Service Documentation

2.11.91.4 Clarifications

Per Perry will be implemented later. Put in deferred. Donna Canady

2.11.91.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.91.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.91.7 Change Order Status

Status	Date
Construction in Progress	09/21/2006
Construction in Progress	11/21/2006
Deferred	12/11/2006

2.11.92 CH File Tracking/Exc. Handling - 4052

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
4052	Change Order		EDI and Claim Capture			

2.11.92.1 Desired Solution
This CO is for the File Tracking/Exception Handling that must be supported in Cold Harbor for complete FTS integration

2.11.92.2 Business Impact
None.

2.11.92.3 Technical Specifications
None.

2.11.92.4 Clarifications
No associated clarifications found.

2.11.92.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.92.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.92.7 Change Order Status

Status	Date
SE Assigned	10/05/2006
Model Office Implemented	11/21/2006
UAT Implemented	06/14/2007
Prod Implemented	06/14/2007

2.11.93 KY Amend - T_DOWNLOAD - 5275

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
5275	Change Order		EDI and Claim Capture			

2.11.93.1 Desired Solution
Change is needed to T_Download. This table needs a new column as stated in the document. Impact analysis is already done and it effects Internet and EDI. Both subsystems have relevant CO's to make the expected change in the table.

2.11.93.2 Business Impact
None.

2.11.93.3 Technical Specifications
Change is needed to T_Download. This table needs a new column as stated in the document.

2.11.93.4 Clarifications
No associated clarifications found.

2.11.93.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.93.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.93.7 Change Order Status

Status	Date
Change Order Written	12/19/2006
Ready for DM Review	12/20/2006
Ready for Model Office	01/04/2007
Model Office Implemented	04/05/2007
UAT Implemented	04/05/2007
Prod Implemented	06/14/2007

2.11.94 KY AMEND - T_FILE_TRACK - 5276

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
5276	Change Order		EDI and Claim Capture			

2.11.94.1 Desired Solution
Change the length of ID_USER in the table T_FILE_TRACK. This only effects EDI and the impact analysis is already done for this.

2.11.94.2 Business Impact
None.

2.11.94.3 Technical Specifications
Change the length of ID_USER in the table T_FILE_TRACK. This only effects EDI.

2.11.94.4 Clarifications
No associated clarifications found.

2.11.94.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.94.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.94.7 Change Order Status

Status	Date
Change Order Written	12/19/2006
Ready for DM Review	12/20/2006
Ready for Model Office	01/04/2007
Model Office Implemented	04/05/2007
UAT Implemented	04/05/2007
Prod Implemented	06/14/2007

2.11.95 Trans encounter 277 resp file - 5279

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
5279	Change Order		EDI and Claim Capture			2

2.11.95.1 Desired Solution
We need to create a custom NSF validation component for transportation encounters. This component will do syntax validation for the NSF file. A validation response file will be created to be sent back to the Transportation department and report all the syntax errors.

2.11.95.2 Business Impact
None.

2.11.95.3 Technical Specifications
None.

2.11.95.4 Clarifications
No associated clarifications found.

2.11.95.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.95.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.95.7 Change Order Status

Status	Date
Issue Identified	12/20/2006
Defect Corrected	03/08/2007

2.11.96 TPM UI - Role Based MEUPS – 6194

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
6194	Change Order		EDI and Claim Capture			

2.11.96.1 Desired Solution
Add role based MEUPS/SSO security to the TPM (Trading Partner Mgmt) and FTS (file tracking service) UI panels.

2.11.96.2 Business Impact
None.

2.11.96.3 Technical Specifications
Add role based MEUPS/SSO security to the TPM and FTS UI panels.

We should be able to provide limited access to users based on their roles.

Complete View and Update privileges or only View privileges.

2.11.96.4 Clarifications
Deferred decision based on the following: As a compromise we could add TPM/FTS to SSO just to control access. Scott says it would be a Web Configuration change and add a link to the Main MEUPS page. Then we will look to add role based security after GO-live.

Email will be attached with complete notification.

2.11.96.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.96.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.96.7 Change Order Status

Status	Date
SE Assigned	02/02/2007
Deferred	02/02/2007

2.11.97 provider number 8 or 10 solution - 6286

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
6286	Change Order		EDI and Claim Capture			2

2.11.97.1 Desired Solution

Solution to accommodate 8 or 10 digit provider numbers

Inbound Transactions:

Any Provider Number (not just billing provider)

If, the value is a Medicaid Provider Number and

Length of value equals 10 and

Position 1 and 2 of this value DOES NOT equals 71 and

Positions 9 and 10 of this value equals 00

EDI will only export Positions 1 - 8 to the appropriate attribute

Outbound Transactions:

Any Provider Number (not just billing provider)

If, the value is a Medicaid Provider Number and

Length of value equals 8 and

Position 1 and 2 of this value DOES NOT equals 71

EDI will add 00 to the end of the value making the length of the value 10

2.11.97.2 Business Impact

None.

2.11.97.3 Technical Specifications

Incoming 270 and Outgoing 271:

Incoming 276 and Outgoing 277:

Outgoing Unsolicited 277:

Outgoing 835:

Incoming 837D:

Incoming 837I:

Incoming 837P:

The below changes were made to all the above mentioned Transactions:

Inbound Transactions:

- Any Provider Number (not just billing provider)

- If, the value is a Medicaid Provider Number and
- Length of value equals 10 and
- Position 1 and 2 of this value DOES NOT equals 71 and
- Positions 9 and 10 of this value equals 00
- EDI will only export Positions 1 - 8 to the appropriate attribute

Outbound Transactions:

- Any Provider Number (not just billing provider)
- If, the value is a Medicaid Provider Number and
- Length of value equals 8 and
- Position 1 and 2 of this value DOES NOT equals 71
- EDI will add 00 to the end of the value making the length of the value 10

Incoming 270 and Outgoing 271:

- 2100B-NM108 (eligVerification \ eligRequest \ eligInfoReceiver \ receiverIdCodeQualifier

2.11.97.4 Clarifications

No associated clarifications found.

2.11.97.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.97.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.97.7 Change Order Status

Status	Date
Issue Identified	02/07/2007
Prod Implemented	09/19/2007

2.11.98 Update PKG_Tracking Package - 7932

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
7932	Change Order		EDI and Claim Capture			

2.11.98.1 Desired Solution

Need to change a stored procedure so we can log EDI specific tracking details for functional Acks and TA1 Acks.

2.11.98.2 Business Impact

None.

2.11.98.3 Technical Specifications

Need to change a stored procedure so we can log EDI specific tracking details for functional Acks and TA1 Acks.

Stored Procedure change made to the Data Model and implemented in production.

2.11.98.4 Clarifications

No associated clarifications found.

2.11.98.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.98.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.98.7 Change Order Status

Status	Date
Ready for Model Office	05/10/2007
Prod Implemented	09/19/2007

2.11.99 Create New Indexes - 7935

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
7935	Change Order		EDI and Claim Capture			

2.11.99.1 Desired Solution
Need this to clean up old records and for faster join with other tables having this as foreign key

2.11.99.2 Business Impact
None.

2.11.99.3 Technical Specifications
Need this to clean up old records and for faster join with other tables having this as foreign key

2.11.99.4 Clarifications
No associated clarifications found.

2.11.99.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.99.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.99.7 Change Order Status

Status	Date
DM Review Board Approved	05/10/2007

2.11.100 NCPDP Batch Document Tracking - 7936

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
7936	Change Order		EDI and Claim Capture			

2.11.100.1 Desired Solution

Add deferred CO to provide the Encounter batch stats on the MCO panel from FTS for NCPDP batches to be done after go-live.

Bill, Ponder to tell Jason Webster this will be added as a post go-live CO to be done as a deferred CO.

2.11.100.2 Business Impact

None.

2.11.100.3 Technical Specifications

From: Taunton, Dean

Sent: Thursday, May 10, 2007 6:34 AM

To: Heer, Perry; Wajda, Mike

Cc: Ponder, Bill; Canady, Donna S

Subject: RE: KYDDI - HIGH- INFO - NCPDP batch files on Encounter Batch Panels

Importance: High

Perry/Mike,

Add deferred COs to provide the Encounter batch stats on the MCO panel from FTS for NCPDP batches to be done after go-live. Provide Bill the CO numbers.

Bill, you can tell Jason it will be added as a post go-live CO to be done as a deferred CO.

R,

Dean Taunton, PE, PMP

New KYMMIS DDI Implementation Manager

EDS

656 Chamberlin Avenue

Frankfort, KY 40601

Office: 502-209-3197

Cell: 502-320-4421

From: Ponder, Bill

Sent: Wednesday, May 09, 2007 11:22 PM

To: Taunton, Dean; Canady, Donna S

Subject: RE: KYDDI - HIGH- INFO - NCPDP batch files on Encounter Batch Panels

You can see the NCPDP batch in FTS but information in FTS has always be

2.11.100.4 Clarifications

No associated clarifications found.

2.11.100.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.100.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.100.7 Change Order Status

Status	Date
Deferred	05/10/2007
Assigned SE	08/29/2007

2.11.101 PA Txn synch code - 8874

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
8874	Change Order		EDI and Claim Capture			3

2.11.101.1 Desired Solution

An additional field is required for the error message on the SHPS PA transaction u_prior_auth_errors table. The synch code field is being added that links a specific line item on maxMC to a specific PA line item in interChange.

2.11.101.2 Business Impact

None.

2.11.101.3 Technical Specifications

Add new synch code field to the PA278 schema

Modify sql to update the synch code field when returning an error from interChange when there is a value in the field. There may not always be a synch code for an error message.

2.11.101.4 Clarifications

8874 is just an associated CO of 8686. The testing for this was done in 8686.

2.11.101.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.101.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.101.7 Change Order Status

Status	Date
Change Order Written	08/02/2007
UAT Implemented	08/30/2007
Ready for DM Review	08/30/2007

2.11.102 I3506 - 9173

Identifier	Type	Level	Subsystem	Computed	Estimated	Priority
9173	Change Order		EDI and Claim Capture			1

2.11.102.1 Desired Solution
Defect on AEVS regarding eligibility verification for program codes of Z, ZJ, Zk, ZL and ZQ which is not compliant to the user manual.

2.11.102.2 Business Impact
None.

2.11.102.3 Technical Specifications
This change required an additional field to be added to the evselig program to bring back the program code from the T_CDE_AID table. This field will then be used to determine what message to play in the eligibility speakback.

2.11.102.4 Clarifications
For example, members who are eligible and fall within these program codes, providers are getting a response message that the member is eligible in the Medicare Savings Benefit Plan. This does not specify if the member is eligible for Medicare co-insurance and deductible payment only. This is causing major billing issues.

Need AEVS to reflect what is written for the above program codes for steps 8.0, 9.0 and 10.0 in the VRES user manual. See attached copies.

Have requested a query for Z, ZJ, ZK, ZL and ZQ members.

MO AVR testing is complete. Waiting for EDI results

2.11.102.5 Associated Requirements

Requirement ID	Type
No associated Requirements found.	

2.11.102.6 Associated System Objects

Technical Name	Object Type	Title
No associated Change Orders found.		

2.11.102.7 Change Order Status

Status	Date
Change Order Written	09/10/2007
Assigned BA	09/10/2007
Assigned SE	09/11/2007
Construction in Progress	09/11/2007

Status	Date
Assigned SE	09/12/2007
UAT Testing in Progress	09/12/2007

2.12 Data Element Dictionary

Because of the length of some column names and the use of underscore characters, some names in the tables below may wrap to the next line.

2.12.1 T_BATCH

Associates a batch of transactions with a file identifier and a sender. It identifies the transaction type that is within the group and what day the group was processed. It also identifies if the group of transactions were successfully translated.

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE	Associates the transaction group with a file identifier that can be used to look up the recording of transaction.	NUMBER	9	0	Y
SAK_FILE_TRACK	This system generated key identifies the file and message information passing through EDI environment	NUMBER	9	0	N
NAM_FILEPATH	Path where the file is saved	VARCHAR2	255	0	N
DTE_PROCESSED	The date that the batch of transactions completed being processed by the transaction processor.	DATE	0	0	N
FILENAME	Filename is composed of: media: is the media type (web, disk, tape, cdrom, ftp, TCP/IP) sak_upload: is a 9-digit number that uniquely identifies this batch TPID: is a 10-digit ID that identifies the submitting entity txn is the transaction type (837P, 837I, 837D, 270, 276, NCPDP, and so on. -- claims would only be the 837x and NCPDP types).	VARCHAR2	255	0	N
CDE_SUBMIT	txn is the transaction type (837P, 837I, 837D, 270, 276, NCPDP, and so on. -- claims would only be the 837x and NCPDP types)	VARCHAR2	15	0	N
SENDER	TPID: is a 10-digit ID that identifies the submitting entity	VARCHAR2	50	0	N

Column Name	Description	Type	Length	Precision	Primary Key
MEDIA	media: is the media type (web, disk, tape, cdrom, ftp, TCP, RRI)	VARCHAR2	50	0	N
CDE_TXN	Describes the set of transactions. Values: A - Claim, B - Reversal, C - PA Inq., D - PA Rqst., E - Claim Inq., F - Elig Inq, G - Unknown.	CHAR	2	0	N
TXN_ACCEPTED	Number of transaction that could be translated	NUMBER	9	0	N
TXN_REJECTED	Number of transactions that failed to translate or were unsuccessful.	NUMBER	9	0	N

2.12.2 T_BBS_CORRELATION

Correlates providers with available document types

Column Name	Description	Type	Length	Precision	Primary Key
SAK_BBS_CORRELATION	Unique ID	NUMBER	9	0	Y
ID_PROVIDER	Id of the Provider being correlated	CHAR	15	0	N
ID_ISA_RECEIVER	ISA08 value in X12 document, ID being used as receiver id	VARCHAR2	15	0	N
SAK_SUPPORTED_DOC	Unique numeric for different document types.	NUMBER	9	0	N

2.12.3 T_CDE_EVENT

Lists the types of event that can be captured

Column Name	Description	Type	Length	Precision	Primary Key
SAK_EVENT	Unique numeric representing different codes of the event	NUMBER	9	0	Y
CDE_EVENT	Codes representing the type of event	VARCHAR2	50	0	N
DSC_EVENT	Description of the event	VARCHAR2	255	0	N

2.12.4 T_CDE_EVENT_KEY

This table has the codes that describe the business keys that can be tracked for an event.

Column Name	Description	Type	Length	Precision	Primary Key
SAK_EVENT_KEY	Unique codes for the Event Keys	NUMBER	9	0	Y
CDE_EVENT_KEY	Code representing the different Event keys	VARCHAR2	50	0	N
DSC_EVENT_KEY	Description for the Event Key codes	VARCHAR2	255	0	N

2.12.5 T_CDE_FILE_DIRECTION

Lists the direction of the file in respect to MMIS

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE_DIRECTION	Unique numeric code representing different file directions.	NUMBER	9	0	Y
CDE_FILE_DIRECTION	Codes representing the different file directions. Value values are: Inbound Outbound Solicited_Request Solicited_Response	VARCHAR2	50	0	N
DSC_FILE_DIRECTION	Description of the File Direction Code	VARCHAR2	255	0	N

2.12.6 T_CDE_FILE_STATUS

Lists all the possible status a file can have

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE_STATUS	Unique codes describing the file status	NUMBER	9	0	Y
CDE_FILE_STATUS	Code representing different file statuses.	VARCHAR2	50	0	N
DSC_FILE_STATUS	Description of the file status codes	VARCHAR2	255	0	N

2.12.7 T_CDE_FUNCTIONAL_STATUS

This table has the information about the 997 functional response that was sent or received for an EDI file.

Column Name	Description	Type	Length	Precision	Primary Key
CDE_FUNCTIONAL_STATUS	Codes representing the different AK5-01 values of 997 Valid values are: A E M R W X	CHAR	1	0	Y
DSC_FUNCTIONAL_STATUS	Description of codes representing the different AK5-01 values of 997. A - Accepted / ADVISED E - Accepted But Errors Were Noted M - Rejected, Message Authentication Code (MAC) / Failed R - Rejected / ADVISED W - Rejected, Assurance Failed Validity Tests X - Rejected, Content After Decryption Could Not Be Analyzed	VARCHAR2	255	0	N

2.12.8 T_CDE_PORT

Lists the description of all the ports

Column Name	Description	Type	Length	Precision	Primary Key
CDE_PORT	Unique key. Valid values are: T M P	CHAR	1	0	Y
DSC_PORT	Description describing the code valid values could be: Test Environment Model Office Environment Production Environment	VARCHAR2	255	0	N

2.12.9 T_CDE_SUPPORTED_DOC

Lists all the document types that are supported by EDI subsystem

Column Name	Description	Type	Length	Precision	Primary Key
SAK_SUPPORTED_DOC	Unique numeric for different document types.	NUMBER	9	0	Y
CDE_INDUSTRY	GS08 value of the EDI document	VARCHAR2	12	0	N
CDE_FUNCTIONAL	GS01 value of the EDI document	CHAR	2	0	N

Column Name	Description	Type	Length	Precision	Primary Key
CDE_TRANSACTION	ST01 value of the EDI document	CHAR	3	0	N
CDE_IDENTIFICATION	Identification code for the document	VARCHAR2	50	0	N
IND_ENCOUNTER	Indicator if this document type represent encounter claim	CHAR	1	0	N
DSC_SUPPORTED_DOC	Description of the document type	VARCHAR2	255	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N
DSC_SHORT_NAME	SHORT DESCRIPTION CODE	VARCHAR2	25	0	N

2.12.10 T_CDE_TP_STATUS

Lists all the status a trading partner can have

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TP_STATUS	Unique numeric code for different trading partner statuses.	NUMBER	9	0	Y
CDE_TP_STATUS	Unique code representing the different statuses of trading partners.	VARCHAR2	50	0	N
DSC_TP_STATUS	Description of the status code	VARCHAR2	255	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N

2.12.11 T_CDE_TP_TYPE

Lists the types of trading partner

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TP_TYPE	numeric code for different trading partner types	NUMBER	9	0	Y
CDE_TP_TYPE	Abbreviated code representing different trading partner types	VARCHAR2	50	0	N
DSC_TP_TYPE	Description of the trading partner type	VARCHAR2	255	0	N

2.12.12 T_CDE_TRANSPORT

Lists the transport mechanism through which we received a file

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TRANSPORT	Unique numeric code for different transport types	NUMBER	9	0	Y
CDE_TRANSPORT	Abbreviated codes representing different transport types FTP FTPS	VARCHAR2	50	0	N
DSC_TRANSPORT	Description of the transport types	VARCHAR2	255	0	N

2.12.13 T_DOCUMENT_TRACK

This table lists all the EDI header(ISA and GS) information for the X12 files received or sent by the system. This table also lists the envelope control number values which were added to EDI document during TPM

Column Name	Description	Type	Length	Precision	Primary Key
SAK_DOCUMENT_TRACK	This is the system generated key that identifies EDI header(ISA and GS) information for the X12 files received or sent by the system	NUMBER	9	0	Y
SAK_FILE_TRACK	Foreign key to T_FILE_TRACK	NUMBER	9	0	N
SAK_TP_CONFIG	Foreign key from T_TP_Config	NUMBER	9	0	N
NUM_ISA_CNTRL	ISA13 value from the EDI document	VARCHAR2	9	0	N
NUM_GROUP_CNTRL	GS06 value from EDI document	VARCHAR2	9	0	N
ID_SND_QUALIFY	ISA05 value from EDI document	VARCHAR2	2	0	N
ID_SND	ISA06 value from EDI document	VARCHAR2	15	0	N
ID_RECV_QUALIFY	ISA07 value from EDI document	VARCHAR2	2	0	N
ID_RECV	ISA08 value from EDI document	VARCHAR2	15	0	N

Column Name	Description	Type	Length	Precision	Primary Key
NUM_ORGN_ISA_CNTRL	Original ISA13 value from EDI document before TPM actions	VARCHAR2	9	0	N
NUM_ORGN_GROUP_CNTRL	Original GS06 value from EDI document before TPM actions	VARCHAR2	9	0	N
ID_ORGN_SND_QUALIFY	Original ISA05 value from EDI document before TPM actions	VARCHAR2	2	0	N
ID_ORGN_SND	Original ISA06 value from EDI document before TPM actions	VARCHAR2	15	0	N
ID_ORGN_RECV_QUALIFY	Original ISA07 value from EDI document before TPM actions	VARCHAR2	2	0	N
ID_ORGN_RECV	Original ISA08 value from EDI document before TPM actions	VARCHAR2	15	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N
DTE_LAST_UPDATED	Date time record was last updated	DATE	0	0	N

2.12.14 T_DOCUMENT_TRANSACTIONS

This table lists all the EDI transaction (ST-SE) information for the X12 files received or sent by the system

Column Name	Description	Type	Length	Precision	Primary Key
SAK_DOCUMENT_TRANSACTIONS	This is a system generated key that identifies EDI transaction (ST-SE) information for the X12 files received or sent by the system	NUMBER	9	0	Y
SAK_DOCUMENT_TRACK	Foreign key to T_DOCUMENT_TRACK table	NUMBER	9	0	N
NUM_TRANSACTION_CNTRL	ST02 value in the EDI document.	VARCHAR2	9	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N

2.12.15 T_DOWNLOAD

List of files available for download by web users.

Column Name	Description	Type	Length	Precision	Primary Key
SAK_DOWNLOAD	The unique id for a file available to download for a web user.	NUMBER	9	0	Y
SAK_FILE_TRACK	SAK File Track	NUMBER	9	0	N
FILENAME	Name of the file.	VARCHAR2	256	0	N
SAK_SUPPORTED_DOC	system supported documentation	NUMBER	9	0	N
DTE_DOWNLOADED	Last date the file was downloaded by the web user.	DATE	0	0	N
DTE_AVAILABLE	Date and time that a file was available to be downloaded by the user.	DATE	0	0	N
DTE_PURGE	Date and time that a file was purged from the system.	DATE	0	0	N
CDE_MIME_TYPE	A file ending that denotes the file's content. Often called its "MIME" type. Some examples would be ".Z", ".txt", ".doc", and ".xls". The first character in this filed should always be a period.	CHAR	4	0	N
EXTRA_TEXT	Additional information.	VARCHAR2	256	0	N
NAM_FILEPATH	Path where file is physically available for download	VARCHAR2	255	0	N

2.12.16 T_EVENT_TRACK

This table lists all the events that have been captured by the system

Column Name	Description	Type	Length	Precision	Primary Key
SAK_EVENT_TRACKING	This is a system generated key that identifies events that have been captured by the system	NUMBER	9	0	Y

Column Name	Description	Type	Length	Precision	Primary Key
SAK_NAME	Name of the SAK. This should match the SAK_NAME value in T_SYSTEM_KEYS table	VARCHAR2	18	0	N
SAK	SAK Column from T_SYSTEM_KEYS table	NUMBER	9	0	N
SAK_EVENT	Unique numeric representing different codes of the event	NUMBER	9	0	N
DSC_EVENT_TRACK	Description or comments related to the event	VARCHAR2	250	0	N
DTE_CREATED	Date time record was created	TIMESTAMP	6	0	N

2.12.17 T_EVENT_TRACK_KEYS

This table has the business key related to an Event and that key value.

Column Name	Description	Type	Length	Precision	Primary Key
SAK_EVENT_TRACK_KEYS	This is a system generated key that identifies a business key related to an Event and that key value.	NUMBER	9	0	Y
SAK_EVENT_TRACKING	Foreign key from T_EVENT_TRACK table	NUMBER	9	0	N
SAK_EVENT_KEY	Code for the Event Keys	NUMBER	9	0	N
TXT_EVENT_KEY_VALUE	Value related to the Event Key	VARCHAR2	255	0	N
DSC_EVENT_TRACK_KEYS	Description or comments related to the event key	VARCHAR2	255	0	N
DTE_CREATED	Date time the record was created	TIMESTAMP	6	0	N

2.12.18 T_FILE_CORRELATION

This table correlate solicited (request response) files created or received by the system

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE_CORRELATION	This system generated key identifies correlations of solicited (request response) files created or received by the system	NUMBER	9	0	Y
SAK_FILE_TRACK_REQUEST	Foreign key from the T_FILE_TRACK record of the request	NUMBER	9	0	N
SAK_FILE_TRACK_RESPONSE	Foreign key from the T_FILE_TRACK record of the response	NUMBER	9	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N

2.12.19 T_FILE_SOLICITED_TRACK

This table holds the solicited EDI messages information passing through EDI environment

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE_SOLICITED_TRACK	This system generated key identifies the solicited EDI messages information passing through EDI environment	NUMBER	9	0	Y
SAK_FILE_TRACK	Foreign key from the T_FILE_TRACK	NUMBER	9	0	N
ADR_IP	IP address of the request	VARCHAR2	20	0	N
NUM_INBOUND_PORT_INSTANCE	Port instance from which request was received	NUMBER	11	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N

2.12.20 T_FILE_TRACK

This table hold the file and message information passing through EDI environment

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE_TRACK	This system generated key identifies the file and message information passing through EDI environment	NUMBER	9	0	Y
SAK_FILE_STATUS	Codes from T_CDE_FILE_STATUS	NUMBER	9	0	N
ID_TRANSACTION	This is a GUID generated to tie two File Tracking records. This identifier should be same for request response tracking records	VARCHAR2	50	0	N
CDE_CRC32	CRC 32 code unique for every file. This is used for duplicate check	VARCHAR2	8	0	N
NUM_FILESIZE	Number of bytes in the file	NUMBER	11	0	N
NAM_FILE	Original name of the file when it was received	VARCHAR2	255	0	N
NAM_FILEPATH	Directory path where a file was found or placed	VARCHAR2	255	0	N
IND_MESSAGE	Indicator if the record represents a message. Default is file	CHAR	1	0	N
SAK_TRANSPORT	Unique numeric code for different transport types	NUMBER	9	0	N
SAK_FILE_DIRECTION	Unique numeric code representing different file directions.	NUMBER	9	0	N
NAM_ORGN_SERVER	Name of server where tacking originated from	VARCHAR2	50	0	N
DTE_CREATED	Date time when record was created	DATE	0	0	N
DTE_LAST_UPDATED	Date time when the record was last updated	DATE	0	0	N

Column Name	Description	Type	Length	Precision	Primary Key
ID_USER	Unique ID for the account transferring the file	VARCHAR2	50	0	N
ID_ISA_IDENTIFIER	ISA06 or ISA08 value in X12 document, ID being used by trading partner as sender or receiver	VARCHAR2	15	0	N

2.12.21 T_FILE_TRACK_EXCEPTION

This table correlates file tracking record to handled exceptions if the file processing failed in EDI environment

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE_TRACK_EXCEPTION	This system generated key identifies the correlation of file tracking records to handled exceptions if the file processing failed in EDI environment	NUMBER	9	0	Y
SAK_FILE_TRACK	Foreign key from T_FILE_TRACK	NUMBER	9	0	N
ID_BATCH	ID_BATCHID from T_TRACK_EXCEPTIONHANDLE	VARCHAR2	50	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N

2.12.22 T_FUNCTIONAL_ACKS

List the status of all 997 acknowledgements received or sent by the system

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FUNCTIONAL_ACK	This system generated key identifies the status of all 997 acknowledgements received or sent by the system	NUMBER	9	0	Y

Column Name	Description	Type	Length	Precision	Primary Key
SAK_DOCUMENT_TRACK	This is the system generated key that identifies EDI header(ISA and GS) information for the X12 files received or sent by the system	NUMBER	9	0	N
CDE_FUNCTIONAL_STATUS	This represents the codes from T_CDE_FUNCTIONAL_STATUS codes	CHAR	1	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N

2.12.23 T_MEDIA

List of possible mediums for transactions.

Column Name	Description	Type	Length	Precision	Primary Key
CDE_MEDIA	Code.	CHAR	1	0	Y
DSC_MEDIA	Description.	VARCHAR2	32	0	N

2.12.24 T_PORT_KY

This table lists the ports available for VANs

Column Name	Description	Type	Length	Precision	Primary Key
SAK_PORT	This system generated key identifies ports available for VANs	NUMBER	9	0	Y
NUM_PORT	Port number assigned to a VAN	NUMBER	9	0	N
CDE_PORT	Code from T_CDE_PORT	CHAR	1	0	N

2.12.25 T_PORT

This table lists the ports available for VANs

Column Name	Description	Type	Length	Precision	Primary Key
SAK_PORT	This system generated key identifies ports available for VANs	NUMBER	9	0	Y
NUM_PORT	Port number assigned to a VAN	NUMBER	9	0	N
CDE_PORT	Code from T_CDE_PORT	CHAR	1	0	N

2.12.26 T_TELECOM_TP

This table contains Trading Partner Identification Codes that are associated with billable entities (those who have entered a telecom agreement). This table is updated by the TP maintenance window.

Column Name	Description	Type	Length	Precision	Primary Key
CDE_TP_ID	Trading Partner Identification Code	CHAR	15	0	Y
DTE_EFFECTIVE	Effective Date.	DATE	0	0	N

2.12.27 T_TP

Lists all the trading partners

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TP	This system generated key identifies the trading partners	NUMBER	9	0	Y
NUM_SEQ	Auto number to sequence the number of records of the same trading partner.	NUMBER	11	0	N
CDE_AUTHORIZATION_QUALIFIER	ISA01 value in X12 document	VARCHAR2	2	0	N
CDE_AUTHORIZATION	ISA02 value in X12 document	VARCHAR2	10	0	N
CDE_SECURITY_QUALIFIER	ISA03 value in X12 document	VARCHAR2	2	0	N
CDE_SECURITY	ISA04 value in X12 document	VARCHAR2	10	0	N
ID_ISA_IDENTIFIER_QUALIFIER	ISA05 or ISA07 value in X12 document, Qualifier being used by trading partner as sender or receiver	VARCHAR2	2	0	N
ID_ISA_IDENTIFIER	ISA06 or ISA08 value in X12 document, Id being used by trading partner as sender or receiver	VARCHAR2	15	0	N
CDE_GS_APPLICATION	GS02 value in X12 document	VARCHAR2	15	0	N
NAM_TP	Name of the trading partner	VARCHAR2	50	0	N

Column Name	Description	Type	Length	Precision	Primary Key
TXT_ENCRYPTION_USERID	Encryption id if file or message is encrypted	VARCHAR2	100	0	N
DTE_STATUS_EFFECTIVE_DATE	Status effective date	DATE	0	0	N
ID_PROVIDER	Provider ID in the interChange system	CHAR	15	0	N
CDE_NPI	NPI code of the healthcare providers	VARCHAR2	50	0	N
ADR_STREET_1	Address line 1 of the trading partner	VARCHAR2	55	0	N
ADR_STREET_2	Address line 2 of the trading partner	VARCHAR2	55	0	N
ADR_CITY	City of the trading partner	VARCHAR2	30	0	N
ADR_STATE	State of the trading partner	CHAR	2	0	N
ADR_ZIP	Zip of the trading partner	VARCHAR2	15	0	N
CDE_COUNTRY	Country of the trading partner	CHAR	3	0	N
NAM_CONTACT	Contact person name for the trading partner	VARCHAR2	60	0	N
NUM_PHONE	Contact person telephone number	VARCHAR2	15	0	N
NUM_PHONE_EXT	Contact person telephone number extension	VARCHAR2	10	0	N
NUM_FAX	Contact person facsimile number	VARCHAR2	15	0	N
ADR_EMAIL	Contact person email address	VARCHAR2	100	0	N
TXT_SEGMENT_TERMINATOR	Segment terminator to be used for X12 documents	VARCHAR2	10	0	N
TXT_COMPONENT_SEPARATOR	Component separator to be used for X12 documents	VARCHAR2	10	0	N
DTE_AUTHORIZATION	Authorization date	DATE	0	0	N

Column Name	Description	Type	Length	Precision	Primary Key
DTE_AGREEMENT_MAILED	Date agreement was mailed	DATE	0	0	N
DTE_AGREEMENT_RECV	Date agreement was received	DATE	0	0	N
TXT_NOTES	Notes or comments	CLOB	0	0	N
SAK_TP_TYPE	Foreign key from T_CDE_TP table	NUMBER	9	0	N
SAK_TP_STATUS	Unique numeric code for different trading partner statuses.	NUMBER	9	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N
DTE_LAST_UPDATED	Date time record was last updated	DATE	0	0	N
IND_VAN	Indicator for if trading partner is a vendor area network	CHAR	1	0	N

2.12.28 T_TP_ALLOWED_TRANSPORT

Table is used to restrict trading partners to allowed transport types

Column Name	Description	Type	Length	Precision	Primary Key
SAK_ALLOWED_TRANSPORT	This system generated key identifies allowed transport types for trading Partners.	NUMBER	9	0	Y
SAK_TP	Foreign key from T_TP table	NUMBER	9	0	N
SAK_TRANSPORT	Unique numeric code for different transport types	NUMBER	9	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N

2.12.29 T_TP_CONFIG

Lists the configuration of each trading partner

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TP_CONFIG	This system generated key identifies configurations of trading partner	NUMBER	9	0	Y

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TP	Foreign key from T_TP table	NUMBER	9	0	N
SAK_SUPPORTED_DOC	Foreign key from T_CDE_SUPPORTED_DOCS	NUMBER	9	0	N
SAK_PORT	Foreign key from T_PORT	NUMBER	9	0	N
CDE_USAGE	ISA16 value of an X12 document. Valid values are T and P	CHAR	1	0	N
TXT_OUTBOUND_FILE MASK	File mask of the outbound file as required by the trading partner.	VARCHAR2	255	0	N
IND_EXPECT_INBOUND_997	Indicator if trading partner will send us 997 Acknowledgements for the EDI documents they send us.	CHAR	1	0	N
IND_SENT_OUTBOUND_997	Indicator if trading partner is expecting 997 Acknowledgement for the EDI documents they send us.	CHAR	1	0	N
DTE_CREATED	Date time record was created	DATE	0	0	N
DTE_LAST_UPDATED	Date time record was last updated	DATE	0	0	N
SAK_FILE_DIRECTION	Unique numeric code representing different file directions.	NUMBER	9	0	N

2.12.30 T_TRACK_EXCEPTION_HANDLE

Lists the exceptions handled by the system

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TRACK_EXCEPTION_HANDLE	This system generated key identifies exceptions handled by the system	NUMBER	9	0	Y

Column Name	Description	Type	Length	Precision	Primary Key
ID_BATCH	GUID value which ties all the exceptions and inner exceptions of the system.exception object	VARCHAR2	50	0	N
NUM_BATCH_SEQ	Sequence number of all the exceptions	NUMBER	11	0	N
TXT_EXCEPTION_TYPE	Object type of the exception	VARCHAR2	100	0	N
TXT_MESSAGE	Custom message associated to the exception	VARCHAR2	1000	0	N
NAM_ASSEMBLY	Assembly name where custom exception was created	VARCHAR2	100	0	N
NAM_METHOD	Method name where custom exception object was created	VARCHAR2	500	0	N
NAM_CLASS	Class name that had an exception	VARCHAR2	100	0	N
NAM_MACHINE	Machine name that had an exception	VARCHAR2	100	0	N
NAM_THREAD	Thread ID that had an exception	VARCHAR2	100	0	N
ID_PROCESS	Process ID that had an exception	VARCHAR2	10	0	N
TXT_STACK_TRACE	Stack trace associated to the exception	VARCHAR2	2000	0	N
DTE_EXCEPTION	Date time of the exception	DATE	0	0	N

2.12.31 T_TRACK_NOTIFICATION

Lists the notifications sent by the system

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TRACK_NOTIFICATION	This system generated key identifies the notifications sent by the system	NUMBER	9	0	Y
ADR_NOTIFICATION_TO	Email address notification was sent to	VARCHAR2	200	0	N

Column Name	Description	Type	Length	Precision	Primary Key
ADR_NOTIFICATION_FROM	Email notification which scents the exception	VARCHAR2	200	0	N
TXT_NOTIFICATION_SUBJECT	Subject line on the email notification	VARCHAR2	255	0	N
ID_BATCH	Batch ID of T_TRACK_EXCEPTIONHANDLE table which links notification to the set of exceptions	VARCHAR2	50	0	N
DTE_NOTIFICATION	Date time notification was sent	DATE	0	0	N

2.12.32 T_TRADING_PARTNER

Column Name	Description	Type	Length	Precision	Primary Key
CUSTNO		VARCHAR2	35	0	N
TYP_OWNER		VARCHAR2	1	0	N
NAME		VARCHAR2	35	0	N
IDQUAL		VARCHAR2	4	0	N
IDCODE		VARCHAR2	35	0	N
AUTH_QUAL		VARCHAR2	2	0	N
AUTH_CODE		VARCHAR2	10	0	N
SECU_QUAL		VARCHAR2	2	0	N
SECU_CODE		VARCHAR2	10	0	N
GSID		VARCHAR2	35	0	N
SHIPQUAL		VARCHAR2	2	0	N
SHIPIDEN		VARCHAR2	15	0	N
BILLQUAL		VARCHAR2	2	0	N
BILLIDEN		VARCHAR2	15	0	N
ADDR1		VARCHAR2	35	0	N
ADDR2		VARCHAR2	35	0	N
CITY		VARCHAR2	19	0	N

Column Name	Description	Type	Length	Precision	Primary Key
STATE		VARCHAR2	15	0	N
COUNTRY		VARCHAR2	25	0	N
ZIP		VARCHAR2	9	0	N
CONTACT1		VARCHAR2	35	0	N
TELEPHONE1		VARCHAR2	22	0	N
CONTACT2		VARCHAR2	35	0	N
TELEPHONE2		VARCHAR2	22	0	N
ISA_IN_NO		VARCHAR2	14	0	N
ISA_OUT_NO		VARCHAR2	14	0	N
SND_GSID		VARCHAR2	35	0	N
SND_IDQUAL		VARCHAR2	4	0	N
SND_IDCODE		VARCHAR2	35	0	N
SUB_DELIMT		VARCHAR2	3	0	N
ELE_DELIMT		VARCHAR2	3	0	N
SEG_DELIMT		VARCHAR2	3	0	N
RELEASE_CH		VARCHAR2	3	0	N
X12_REPEAT		VARCHAR2	3	0	N
DEL_CODE		VARCHAR2	1	0	N
EDIF_SUBDL		VARCHAR2	3	0	N
EDIF_ELEDL		VARCHAR2	3	0	N
EDIF_SEGDL		VARCHAR2	3	0	N
EDIF_RELCH		VARCHAR2	3	0	N
EDIF_REPEA		VARCHAR2	3	0	N
HL7_SEGDL		VARCHAR2	3	0	N
HL7_ELEDL		VARCHAR2	3	0	N
HL7_SUBDL		VARCHAR2	3	0	N
HL7_SUBSUB		VARCHAR2	3	0	N

Column Name	Description	Type	Length	Precision	Primary Key
HL7_RELCH		VARCHAR2	3	0	N
HL7_REPEAT		VARCHAR2	3	0	N
EXPORT_FLG		VARCHAR2	1	0	N
MBOX_NAME		VARCHAR2	35	0	N
MAILBOX		VARCHAR2	100	0	N
CURR_FMT		VARCHAR2	1	0	N
POS_LTR		VARCHAR2	1	0	N
SNDR_ROUTE		VARCHAR2	35	0	N
SNDR_SUBID		VARCHAR2	35	0	N
RCVR_ROUTE		VARCHAR2	35	0	N
RCVR_SUBID		VARCHAR2	35	0	N
APP_SND_QL		VARCHAR2	4	0	N
APP_RCV_QL		VARCHAR2	4	0	N
B_RECV_ID		VARCHAR2	24	0	N
B_SEND_ID		VARCHAR2	24	0	N
BIN_NUMB		NUMBER	19	0	N
PROC_NUMB		VARCHAR2	10	0	N
SERV_QUAL		VARCHAR2	2	0	N
SERV_ID		VARCHAR2	15	0	N
SOFT_ID		VARCHAR2	10	0	N
TPKEY		NUMBER	10	0	N

2.12.33 T_TRANSACTION

An instance of a transaction received by our system. Used for logging and reporting.

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TRANSACTION	Unique ID identifying a transaction.	NUMBER	9	0	Y

Column Name	Description	Type	Length	Precision	Primary Key
SAK_FILE	Associates the transaction group with a file identifier that can be used to look up the recording of transaction.	NUMBER	9	0	N
SAK_FILE_TRACK	This system generated key identifies the file and message information passing through EDI environment	NUMBER	9	0	N
SAK_CLAIM	sak claim	NUMBER	9	0	N
SAK_TRANSACTION_TYPE	Unique ID.	NUMBER	9	0	N
CDE_MEDIA	Code.	CHAR	1	0	N
CDE_VAN_ID	Van ID number. For VAN media type this could be the same or different than the submitter id. For WEB and NCP media types Van ID and Sub ID will be the same.	VARCHAR2	35	0	N
CDE_SUB_ID	Submitter ID number from the transaction header.	VARCHAR2	35	0	N
IND_TXN_ACK	Transaction Acknowledgement Indicator - Values: A - Transaction Acknowledged, X - Transaction not Acknowledged	CHAR	1	0	N
DTE_INBOUND	Inbound date/time stamp - transaction begin timestamp	TIMESTAMP	0	0	N
DTE_OUTBOUND	Outbound date/time stamp - transaction complete timestamp	TIMESTAMP	0	0	N

Column Name	Description	Type	Length	Precision	Primary Key
BILL_COUNT	Number of atomic transactions to be billed	NUMBER	9	0	N
BILL_INDICATOR	Values: Y - Yes, bill the transactions; N - No, don't bill the transactions. This indicator will be used to signal that the transaction count should not be billed (due to some system problem or other error encountered)	CHAR	1	0	N
PROCESS_ID		NUMBER	9	0	N
THREAD_ID		NUMBER	9	0	N
REQUEST		XMLTYPE	2000	0	N
RESPONSE		XMLTYPE	2000	0	N
FAULT		XMLTYPE	2000	0	N
SAK_WEB_USER		NUMBER	9	0	N
DTE_CREATED	Date and time when the record is created	TIMESTAMP	4	0	N

2.12.34 T_TRANSACTION_TYPE

List of possible transaction types are system can send or receive.

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TRANSACTION_TYPE	Unique ID.	NUMBER	9	0	Y
DSC_SHORT_NAME	Name of the Transaction.	CHAR	24	0	N
DSC_TRANSACTION_TYPE	Long Description.	VARCHAR2	64	0	N
CDE_INPUT_OUTPUT	I = Input O = Output B = Both	CHAR	1	0	N
IND_HIPAA	Y or N. Y if this transaction type is defined by HIPAA.	CHAR	1	0	N

Column Name	Description	Type	Length	Precision	Primary Key
MAP_TRAN	The Sybase tool uses a 6 character code to identify a HIPAA transaction.	VARCHAR2	6	0	N

2.12.35 T_TRANS_MEDIA_XREF

Some transactions are only available to a select group of medias.

Column Name	Description	Type	Length	Precision	Primary Key
SAK_TRANSACTION_TYPE	Unique ID.	NUMBER	9	0	Y
CDE_MEDIA	Code.	CHAR	1	0	Y

2.12.36 T_TRLOG

Column Name	Description	Type	Length	Precision	Primary Key
AFLD		NUMBER	10	0	N
RUN_ID		NUMBER	19	0	N
TYP		VARCHAR2	1	0	N
RUN_DATE		DATE	0	0	N
ACKBY_DATE		DATE	0	0	N
TRANS_CODE		VARCHAR2	2	0	N
TRANS_NAME		VARCHAR2	6	0	N
TPTNER_ID		VARCHAR2	35	0	N
VERSION		VARCHAR2	12	0	N
ISA_TYPE		VARCHAR2	5	0	N
INTERCHANG		VARCHAR2	35	0	N
GROUP_NO		VARCHAR2	35	0	N
TRANS_NO		VARCHAR2	35	0	N
APP_RCV_CD		VARCHAR2	35	0	N
APP_SND_CD		VARCHAR2	35	0	N
RECV_CODE		VARCHAR2	35	0	N
SEND_CODE		VARCHAR2	35	0	N

Column Name	Description	Type	Length	Precision	Primary Key
RECV_QUAL		VARCHAR2	4	0	N
SEND_QUAL		VARCHAR2	4	0	N
ERRORS		NUMBER	19	0	N
STAT		VARCHAR2	1	0	N
BYTE_COUNT		NUMBER	19	0	N
DIR		VARCHAR2	3	0	N
FLOW_LEVEL		VARCHAR2	5	0	N
RECORD_NAM		VARCHAR2	10	0	N
RECORD_NO		VARCHAR2	6	0	N
FIELD_NAME		VARCHAR2	15	0	N
SEGMENT		VARCHAR2	3	0	N
SEG_COUNT		NUMBER	10	0	N
ELEMENT		VARCHAR2	2	0	N
SUBELEM		VARCHAR2	2	0	N
SEV_CODE		VARCHAR2	2	0	N
MSG_NO		VARCHAR2	5	0	N
MSG_TEXT		VARCHAR2	100	0	N
FILENAME		VARCHAR2	160	0	N
FIELDVAL		VARCHAR2	30	0	N
USER_IDENT		VARCHAR2	35	0	N
ACK_EXPECT		VARCHAR2	1	0	N
TR_ACK_TYP		VARCHAR2	1	0	N
T_P_IND		VARCHAR2	1	0	N
TRANS_CNT		NUMBER	10	0	N
FILEOFFSET		NUMBER	19	0	N
RCOUNT		NUMBER	5	0	N
SNDR_ROUTE		VARCHAR2	35	0	N

Column Name	Description	Type	Length	Precision	Primary Key
SNDR_SUBID		VARCHAR2	35	0	N
RCVR_ROUTE		VARCHAR2	35	0	N
RCVR_SUBID		VARCHAR2	35	0	N
APPL_REF		VARCHAR2	14	0	N
PROC_PRIOR		VARCHAR2	1	0	N
COMM_AGM		VARCHAR2	35	0	N
APP_SND_QL		VARCHAR2	4	0	N
APP_RCV_QL		VARCHAR2	4	0	N
ASSOC_CODE		VARCHAR2	6	0	N
APP_PSWD		VARCHAR2	14	0	N
CLIST_VER		VARCHAR2	6	0	N
MSG_TYPE		VARCHAR2	6	0	N
RPT_NO		VARCHAR2	2	0	N
ST03		VARCHAR2	35	0	N

2.12.37 T_TXN_BILLING

This table contains X12 billing information and is used for the EDI transaction billings.

Column Name	Description	Type	Length	Precision	Primary Key
CDE_BILLING_REC_ID	MMIS Billing Record ID	CHAR	3	0	N
CDE_SEND_REC_ID_ISA	Interchange Sender ID or Interchange Receiver ID (depending on CDE_BILLING_REC_ID)	CHAR	15	0	N
CDE_ISA_RESERVED	Interchange Control Header Reserved Use	CHAR	2	0	N
DTE_ISA	Interchange Date and Time	DATE	0	0	N
CDE_VER_CTRL_NUM	Interchange Control Version Number	CHAR	5	0	N
CDE_CTRL_NUM	Interchange Control Number	CHAR	9	0	N

Column Name	Description	Type	Length	Precision	Primary Key
CDE_T_P	Usage Indicator: P (Production Data), T (Test Data)	CHAR	1	0	N
CDE_SEND_REC_ID_GS	Application Sender's Code or Application Receiver's Code (depending on CDE_BILLING_REC_ID)	CHAR	15	0	N
DTE_GS	Group Date and Time	DATE	0	0	N
SND_CTRL_NUM	Group Control Number	CHAR	9	0	N
SND_VER_NUM	Version/Release/Industry Identifier Code	CHAR	12	0	N
CDE_ST_ID	Transaction Set Identifier Code: 270 (Eligibility, Coverage or Benefit Inquiry), 271 (Eligibility, Coverage or Benefit Information), 276 (Health Care Claim Status Request)277 (Health Care Claim Status Notification)278 (Health Care Services Review Information)820 (Payment Order/Remittance Advice)834 (Benefit Enrollment and Maintenance)835 (Health Care Claim Payment/Advice)837 (Health Care Claim)	CHAR	3	0	N
ST_CTRL_NUM	Transaction Set Control Number	CHAR	9	0	N
ST_LEVEL1_ID	Identification Code.(T_TXN_BILLING_DESC.ST_LEVEL1_ID qualifies the identification code based on CDE_BILLING_REC_ID)	CHAR	15	0	N
ST_LEVEL1_NAME	Name.(T_TXN_BILLING_DESC.ST_LEVEL1_ID qualifies the name based on CDE_BILLING_REC_ID)	CHAR	35	0	N

Column Name	Description	Type	Length	Precision	Primary Key
ST_LEVEL2_ID	Identification Code. (T_TXN_BILLING_DESC.ST_LEVEL2_ID qualifies the identification code based on CDE_BILLING_REC_ID)	CHAR	15	0	N
ST_LEVEL2_NAME	Name.(T_TXN_BILLING_DESC.ST_LEVEL2_ID qualifies the name based on CDE_BILLING_REC_ID)	CHAR	35	0	N
ST_LEVEL2_ID2	Identification Code.(T_TXN_BILLING_DESC.ST_LEVEL2_ID2 qualifies the identification code based on CDE_BILLING_REC_ID)	CHAR	15	0	N
CNT_TXN_1	Counter. (Count of Subscribers, Organization Summary Remittances, Member Details, Payees or Claims depending on CDE_BILLING_REC_ID)	NUMBER	9	0	N
CNT_TXN_2	Counter.(Count of Subscriber Eligibility or Benefit Inquiries, Subscriber Eligibility or Benefit Information, Claim Submitter Trace Numbers, Service Lines, Organization Summary Remittance Details, Member Details or Claim Payment Information depending on CDE_BILLING_REC_ID)	NUMBER	9	0	N
BILLING_FILENAME	Billing File Name	VARCHAR	255	0	N
CDE_TP_ID	Trading Partner Identification Code.	CHAR	15	0	N
SAK_TXN_TYPE	Transaction Type System Assigned Key.	NUMBER	9	0	N
DTE_TXN		DATE	0	0	N

2.12.38 T_TXN_BILLING_DESC

This table contains the description of each field in T_TXN_BILLING that is below the ST level. This table will be used in SQL scripts and procedures that will be developed for account use in the event of billing disputes.

Column Name	Description	Type	Length	Precision	Primary Key
CDE_BILLING_REC_ID	MMIS Billing Record ID	CHAR	3	0	Y
ST_LEVEL1_ID	Qualifies T_TXN_BILLING.ST_LEVEL1_ID based on CDE_BILLING_REC_ID	CHAR	15	0	N
ST_LEVEL1_NAME	Qualifies T_TXN_BILLING.ST_LEVEL1_NAME based on CDE_BILLING_REC_ID	CHAR	35	0	N
ST_LEVEL2_ID	Qualifies T_TXN_BILLING.ST_LEVEL2_ID based on CDE_BILLING_REC_ID	CHAR	15	0	N
ST_LEVEL2_NAME	Qualifies T_TXN_BILLING.ST_LEVEL2_NAME based on CDE_BILLING_REC_ID	CHAR	35	0	N
ST_LEVEL2_ID2	ST_LEVEL2_ID2	CHAR	15	0	N

3 Appendix A – MSMQ Listener Service Documentation

3.1 Introduction

MSMQ Listener Service is a windows service component which is used to read the Messages published into MSMQ Queue using MSMQ Publisher. Pre Processor, Post Processor, FTS Service and BizTalk Orchestration call the MSMQ Publisher Class to publish Tracking Events and Balancing Count information into Defined public Messaging Queues. These features can be set and altered from MSMQListenerService.exe.config file.

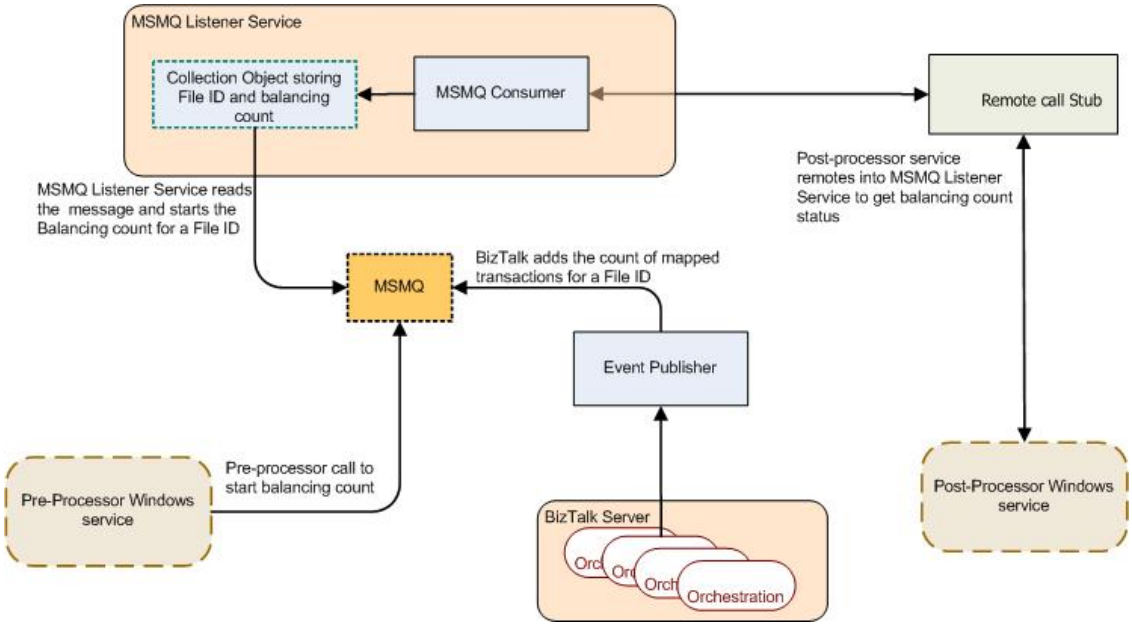
3.1.1 MSMQ Listener Service Features

3.1.1.1 Persistence of Tracking Data

One of the two main purposes of MSMQ Listener service is to persist tracking data published to the queue. Most of the tracking events passed by batch files and all the tracking data sent by real time solutions are inserted in a public queue rather than direct insertion in the database. This reduces transaction time and leads to off the transaction storing of tracking events. It is MSMQ Listener Service's responsibility to pick this data from the queue, study the message, make a valid stored procedure call, and then execute the stored procedure so that tracking data can be viewed in File Tracking Website.

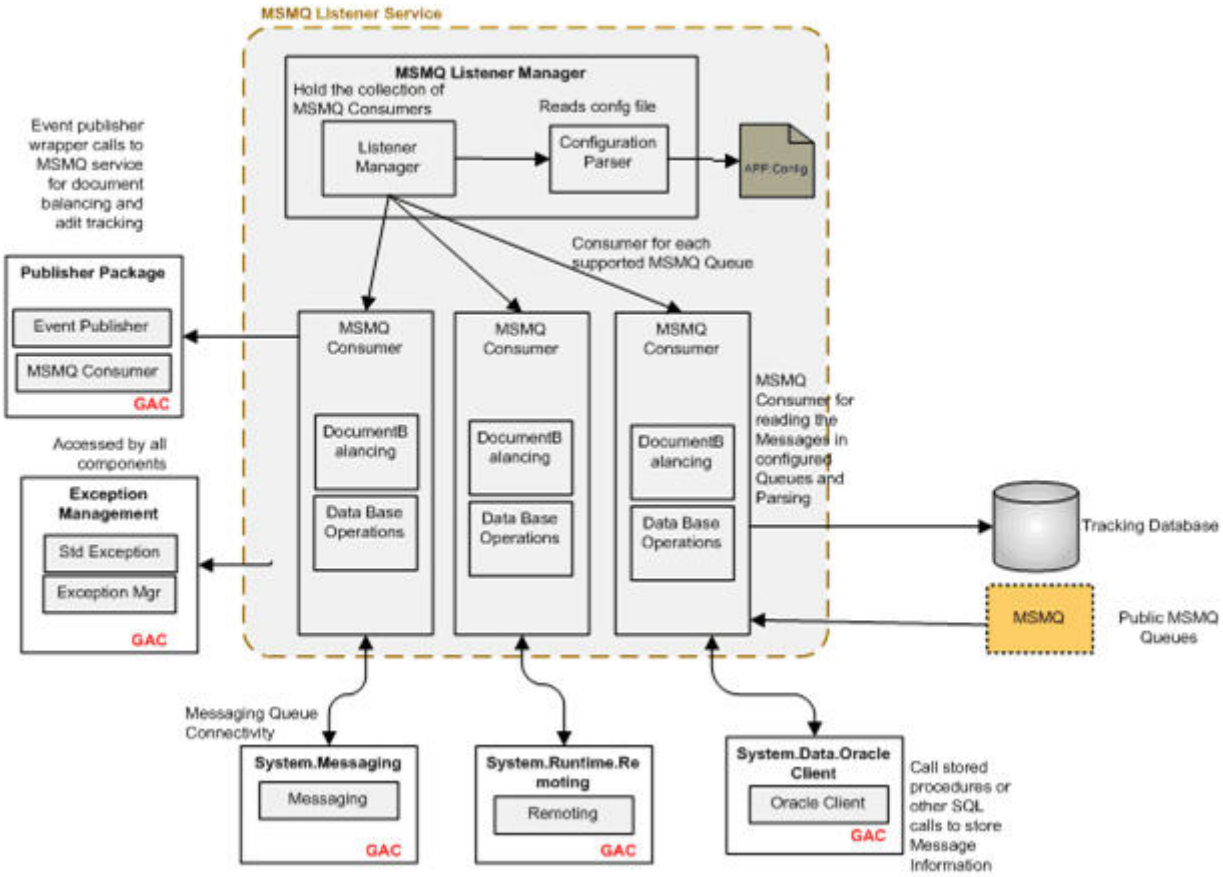
3.1.1.2 Document Balancing

Another main purpose of MSMQ Listener Service is to maintain the Balancing Count of all EDI files being processed through the ODC EDI environment. MSMQ Listener Service receives the call from Pre-processor service to start balancing count for a particular File ID. When the file with that File ID completes BizTalk mapping then orchestration appends to the balancing count record for this File ID created in MSMQ Listener Service's memory. After orchestration processing when BizTalk makes the file available to Post-processor service then Pos-processor service remotes into MSMQ Listener Service and gets the status of this File ID. If the balancing count received from BizTalk is less than the balancing count received from Pre-processor then Post-processor service gets an "Incomplete" status and it does not pick up the file. If the both balancing counts are same then Post-processor service gets a "Completed" status and it picks up the file and continues with its processing.



3.2 MSMQ Listener Service Component Diagram

Below diagram shows all the components used by MSMQ Listener Service and how to interact with each other. In later section this document will go over each project related to File Transfer Service solution in detail.

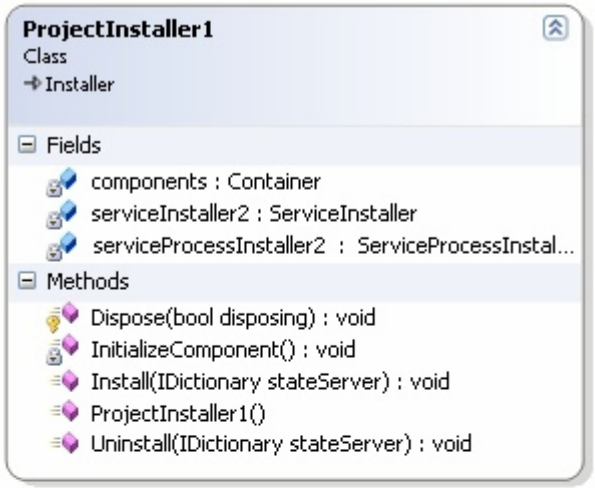


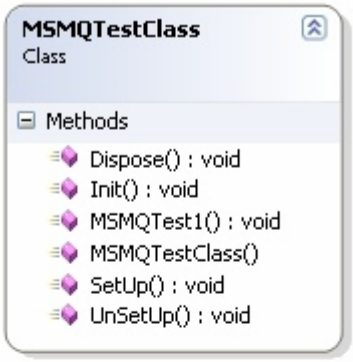
3.3 Projects Description

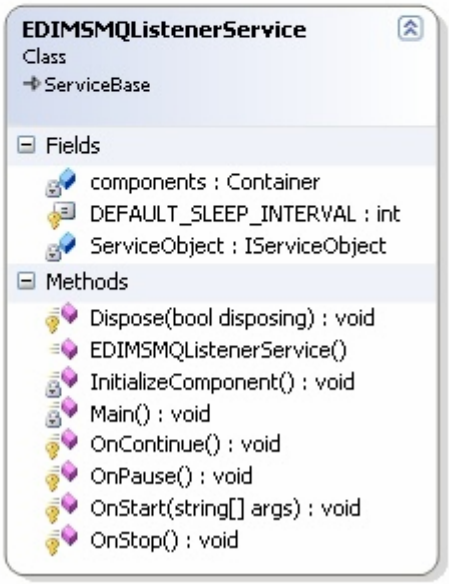
This section will explain all the projects involved in development of the File transfer Service. It will also explain the individual projects, their classes and the methods involved in supporting the required functionality of the File Transfer Service.

3.3.1 MSMQ Listener Service Project

MSMQ Listener service project is the main project which holds the actual windows service component. This windows service component is responsible for calling other assembly projects to perform its designed functions. This project also incorporates the test harness class (MSMQTestClass) which allows the service to be run in a debug mode using NUnit tool.







3.3.1.1 MSMQ Listener Service Class

MSMQ Listener Service is the file which has the main class, “EDIMSMQListenerService”, of the MSMQListenerService project where the windows service gets the control to Start, Stop, Pause or Restart itself.

File Name: FTSService.cs	**Class Name: EDIFTSService**
Method Name	**Description**
EDIMSMQListenerService	This is the default constructor of the class
Main	This is the entry point for the service control. MSMQ Listener Service is instantiated in this method and is set to run.
Initialize Component	This method is called by the constructor and is used to initialize the name of the class.
On Start	This is the method which is called to start the service. This method creates a new instance of MSMQ Listener manager class and then calls the Start method of the instantiated MSMQ Listener Manager class.
Unstop	Before stopping this method calls the Stop method of the above instantiated MSMQ Listener Manager object. If successful it stops running the MSMQ Listener Service.
Dispose	This method disposes of the service object.

3.3.1.2 Project Installer Class

This class is required to create a proper deployment (msi) package for the FTS service. This msi package is used to deploy the service in run time environment.

File Name: ProjectInstaller.cs	**Class Name: ProjectInstaller1**
Method Name	**Description**
ProjectInstaller1	This is the default constructor of the class and this calls Initialize Component.
Initialize Component	This is the designer generated code and should not be modified by developer using code editor.
Install	This method is called during install process of the windows service deployment package and responsible for installing the service component. This method is also used to setup the install properties of the service like service Name, service description and registry settings.
Uninstall	This method is called during uninstall process of the windows service. This method is used to remove the service related registry settings and finally uninstalls the service component.

File Name: ProjectInstaller.cs	**Class Name: ProjectInstaller1**
Method Name	**Description**
Dispose	This method disposes of the service object.

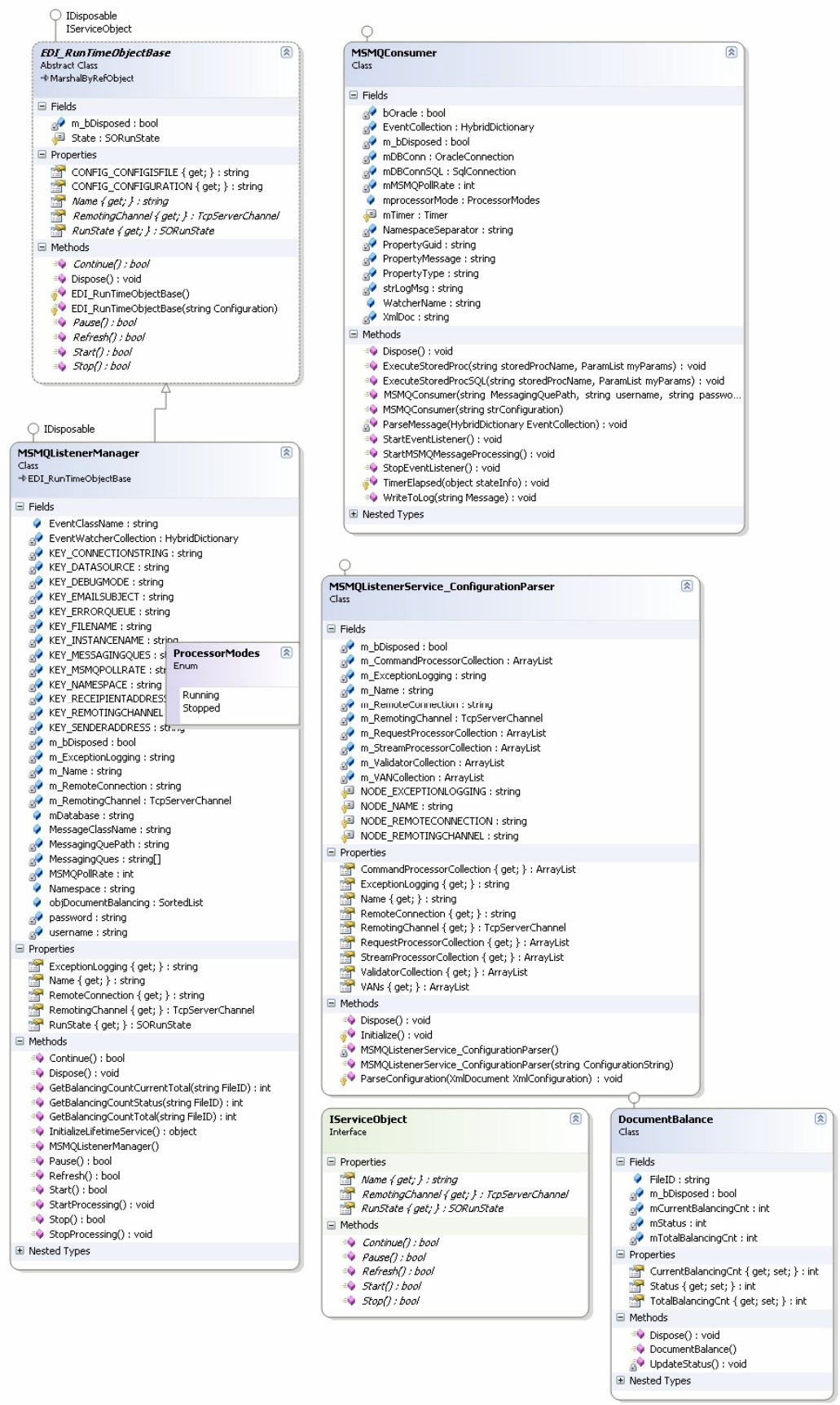
3.3.1.3 MSMQ Test Class

This class is a Text Fixture class of the NUnit tool and helps the developer to run the windows service in the debug mode.

File Name: MSMQTestClass.cs	**Class Name: MSMQTestClass**
Method Name	**Description**
MSMQTest1	This is the NUnit test that is run using the NUnit tool to start the MSMQ Listener Service in the Debug mode. This replaces the need to install the service component. It performs the same functionality of getting the configuration file and then it calls the MSMQ Listener Manager start class.

3.3.2 MSMQ Consumer Project

MSMQ consumer project is called by the MSMQListenerService class and is responsible for setting up the runtime parameters of the service. In the MSMQ Consumer project MSMQ Listener Service accesses the configuration files and reads the configuration attributes needed to initializes MSMQ Listener Manager. After reading the configuration values it starts each MSMQ Queue in a separate MSMQ Consumer class instance thread and starts them. This project is also responsible for cleaning up all the MSMQ Consumers before MSMQ Listener service shuts down.


IDisposable
IServiceObject
EDI_RunTimeObjectBase
Abstract Class
MarshalByRefObject
Fields
m_bDisposed : bool
State : SORunState
Properties
CONFIG_CONFIGISFILE { get; } : string
CONFIG_CONFIGURATION { get; } : string
Name { get; } : string
RemotingChannel { get; } : TcpServerChannel
RunState { get; } : SORunState
Methods
Continue() : bool
Dispose() : void
EDI_RunTimeObjectBase()
EDI_RunTimeObjectBase(string Configuration)
Pause() : bool
Refresh() : bool
Start() : bool
Stop() : bool
MSMQConsumer
Class
Fields
bOracle : bool
EventCollection : HybridDictionary
m_bDisposed : bool
mDBConn : OracleConnection
mDBConnSQL : SqlConnection
mMSMQPollRate : int
mprocessorMode : ProcessorModes
mTimer : Timer
NamespaceSeparator : string
PropertyGuid : string
PropertyMessage : string
PropertyType : string
strLogMsg : string
WatcherName : string
XmlDoc : string
Methods
Dispose() : void
ExecuteStoredProc(string storedProcName, ParamList myParams) : void
ExecuteStoredProcSQL(string storedProcName, ParamList myParams) : void
MSMQConsumer(string MessagingQuePath, string username, string passwo...
MSMQConsumer(string strConfiguration)
ParseMessage(HybridDictionary EventCollection) : void
StartEventListener() : void
StartMSMQMessageProcessing() : void
StopEventListener() : void
TimerElapsed(object stateInfo) : void
WriteToLog(string Message) : void
Nested Types
IDisposable
MSMQListenerManager
Class
EDI_RunTimeObjectBase
Fields
EventClassName : string
EventWatcherCollection : HybridDictionary
KEY_CONNECTIONSTRING : string
KEY_DATASOURCE : string
KEY_DEBUGMODE : string
KEY_EMAILSUBJECT : string
KEY_ERRORQUEUE : string
KEY_FILENAME : string
KEY_INSTANCENAME : string
KEY_MESSAGINGQUES : s
KEY_MSMQPOLLRATE : str
KEY_NAMESPACE : string
KEY_RECEIPIENTADDRESS
KEY_REMOTINGCHANNEL
KEY_SENDERADDRESS : str
m_bDisposed : bool
m_ExceptionLogging : string
m_Name : string
m_RemoteConnection : string
m_RemotingChannel : TcpServerChannel
mDatabase : string
MessageClassName : string
MessagingQuePath : string
MessagingQues : string[]
MSMQPollRate : int
Namespace : string
objDocumentBalancing : SortedList
password : string
username : string
Properties
ExceptionLogging { get; } : string
Name { get; } : string
RemoteConnection { get; } : string
RemotingChannel { get; } : TcpServerChannel
RunState { get; } : SORunState
Methods
Continue() : bool
Dispose() : void
GetBalancingCountCurrentTotal(string FileID) : int
GetBalancingCountStatus(string FileID) : int
GetBalancingCountTotal(string FileID) : int
InitializeLifetimeService() : object
MSMQListenerManager()
Pause() : bool
Refresh() : bool
Start() : bool
StartProcessing() : void
Stop() : bool
StopProcessing() : void
Nested Types
ProcessorModes
Enum
Running
Stopped
MSMQListenerService_ConfigurationParser
Class
Fields
m_bDisposed : bool
m_CommandProcessorCollection : ArrayList
m_ExceptionLogging : string
m_Name : string
m_RemoteConnection : string
m_RemotingChannel : TcpServerChannel
m_RequestProcessorCollection : ArrayList
m_StreamProcessorCollection : ArrayList
m_ValidatorCollection : ArrayList
m_VANCollection : ArrayList
NODE_EXCEPTIONLOGGING : string
NODE_NAME : string
NODE_REMOTECONNECTION : string
NODE_REMOTINGCHANNEL : string
Properties
CommandProcessorCollection { get; } : ArrayList
ExceptionLogging { get; } : string
Name { get; } : string
RemoteConnection { get; } : string
RemotingChannel { get; } : TcpServerChannel
RequestProcessorCollection { get; } : ArrayList
StreamProcessorCollection { get; } : ArrayList
ValidatorCollection { get; } : ArrayList
VANs { get; } : ArrayList
Methods
Dispose() : void
Initialize() : void
MSMQListenerService_ConfigurationParser()
MSMQListenerService_ConfigurationParser(string ConfigurationString)
ParseConfiguration(XmlDocument XmlConfiguration) : void
IServiceObject
Interface
Properties
Name { get; } : string
RemotingChannel { get; } : TcpServerChannel
RunState { get; } : SORunState
Methods
Continue() : bool
Pause() : bool
Refresh() : bool
Start() : bool
Stop() : bool
DocumentBalance
Class
Fields
FileID : string
m_bDisposed : bool
mCurrentBalancingCnt : int
mStatus : int
mTotalBalancingCnt : int
Properties
CurrentBalancingCnt { get; set; } : int
Status { get; set; } : int
TotalBalancingCnt { get; set; } : int
Methods
Dispose() : void
DocumentBalance()
UpdateStatus() : void
Nested Types

3.3.2.1 MSMQ Listener Manager Class

MSMQ Listener Manager Class implements EDI_RunTimeObjectBase class and IDisposable interface.

File Name: MSMQListenerManager.cs	**Class Name: MSMQListenerManager**
Method Name	**Description**
MSMQListenerManager	This is the default constructor of the class. This method reads the App.config to get the configuration file and passes it to Directory manager class.
On Start	This is the method which is called to start the service. This method in turn calls the Start Processing Method.
On Stop	This is the method which is called to stop the service. This method in turn calls the Stop Processing Method. Before stopping this method calls the ReleaseDataComponent method and then calls the Stop method of the above instantiated Directory Manager object. If successful it stops running the File transfer Service.
Start Processing	This method creates an instance of MSMQ Consumer class for each configured MSMQ Queue in App.Config file and calls StartEventListener method of MSMQ Consumer class. All the MSMQ Consumer Class instances are collected in Hybrid Dictionary for reference. This method also creates objDocumentBalancing Sorted List for holding the Document Balancing information and deserializes the Document Balancing information from .dat configured file to sorted list.
Stop Processing	This method calls the StopEventListener method of above created MSMQ Consumer class instances referred through Hybrid Dictionary. This method also serializes the objDocumentBalancing sorted list to .dat configured file in App.Config.
GetBalancingCountStatus	This method takes FileID has input parameter and creates instances of Document Balance class and returns the Balancing Count status from Complete / Failed / Processing options.
GetBalancingCountCurrentTotal	This method takes FileID has input parameter, Makes a Search in above created objDocumentBalancing sorted list, if found creates instances of Document Balance class and returns the Current Balancing Count.
GetBalancingCountTotal	This method takes FileID has input parameter, Makes a Search in above created objDocumentBalancing sorted list, if found creates instances of Document Balance class

File Name: MSMQListenerManager.cs	Class Name: MSMQListenerManager
Method Name	**Description**
	and returns the Total Balancing Count.
Dispose	This method disposes of the service object.

3.3.2.2 MSMQ Consumer Class

MSMQ Consumer class implements IDisposable Interface. For Each Configured Queue in App.Config configuration MSMQ Listener Manager creates an instance of MSMQ Consumer class.

File Name: MSMQConsumer.cs	Class Name: MSMQConsumer
Method Name	**Description**
MSMQConsumer	MSMQConsumer is overloaded constructor method accepts MessagingQuePath, username, password, Namespace, EventClassName, EventClassCondition, Event Collection and MSMQPollRate as input Parameters. This Constructor initializes the required variables for Consuming Messages from the Defined Messaging Queues in App.Config file.
StartEventListener	This method initializes the MSMQ Consumer class Processor mode to Running and creates a timer Call Back Delegate class instance and Associates the Event handler for handling the Timer call back event.
StopEventListener	This method initializes the MSMQ Consumer class processor mode to Stopped and Disables the Timer class polling MSMQ Queues.
Timer Elapsed	This is the Event Handler which will be executed for ever time interval configured as polling rate in App.Config configuration file. This method calls the StartMSMQMessageProcessing and waits until StartMSMQMessageProcessing method Execution completes.
StartMSMQMessageProcessing	This method connects to the configured Messaging Queue, creates an array of System.Messaging.Message by calling MessageQueue.GetAllMessages() method. Once the Array of Messages fetched into array each message is type caste to Notification Message class, converted to object and passed to Parse Message method.
Parse Message	This method parses the received message from Queue and depending upon DocumentBalance.DocumentBalanceStages option received in the message, executes the appropriate

File Name: MSMQConsumer.cs	**Class Name: MSMQConsumer**
Method Name	**Description**
	Document balancing code. This method also calls the ExecuteStoredProc and ExecuteStoredProcSQL methods.
ExecuteStoredProc	This method accepts storedProcName, myParams as input parameters and creates Command. Parameters from the received myparams, executes the received Stored Procedure using ExecuteNonQuery() method of Command class.
ExecuteStoredProcSQL	This method accepts storedProcName, myParams as input parameters and creates Command. Parameters from the received myparams, executes the received Stored Procedure using ExecuteNonQuery() method of Command class.
WriteToLog	This method writes the Messages from Messaging Queue to configured log file if debug mode is configured. The Debug mode can be configured in MSMQ Listener App.Config file.
Dispose	This method disposes of the service object.

3.3.2.3 MSMQListenerService_ConfigurationParser Class

This is a static class which is specifically made to read the configuration file and pass it to the MSMQ Listener Manager class.

File Name: MSMQListenerService_ConfigurationParser.cs	**Class Name: MSMQListenerService_ConfigurationParser**
Method Name	**Description**
MSMQListenerService_ConfigurationParser	This is the default constructor of the class.
MSMQListenerService_ConfigurationParser	This is an overloaded constructor of the class and accepts path of configuration file as an input parameter. This method loads the configuration xml file into xmldocument object and calls the Parse Configuration method.
Parse Configuration	This method accepts xmldocument object as input parameter and returns the required node name.
Dispose	This method disposes of the service object.

3.3.2.4 IService Object Interface

This is the interface which is implemented by EDI_RunTimeObjectBase Abstract Base Class. This class is present in other EDI services also so the implementation of all EDI services is consistent.

File Name: IServiceObject.cs	**Class Name: IServiceObject**
Method Name	**Description**
Start	Interface method.
Stop	Interface method.
Continue	Interface method.
Refresh	Interface method.
Pause	Interface method.

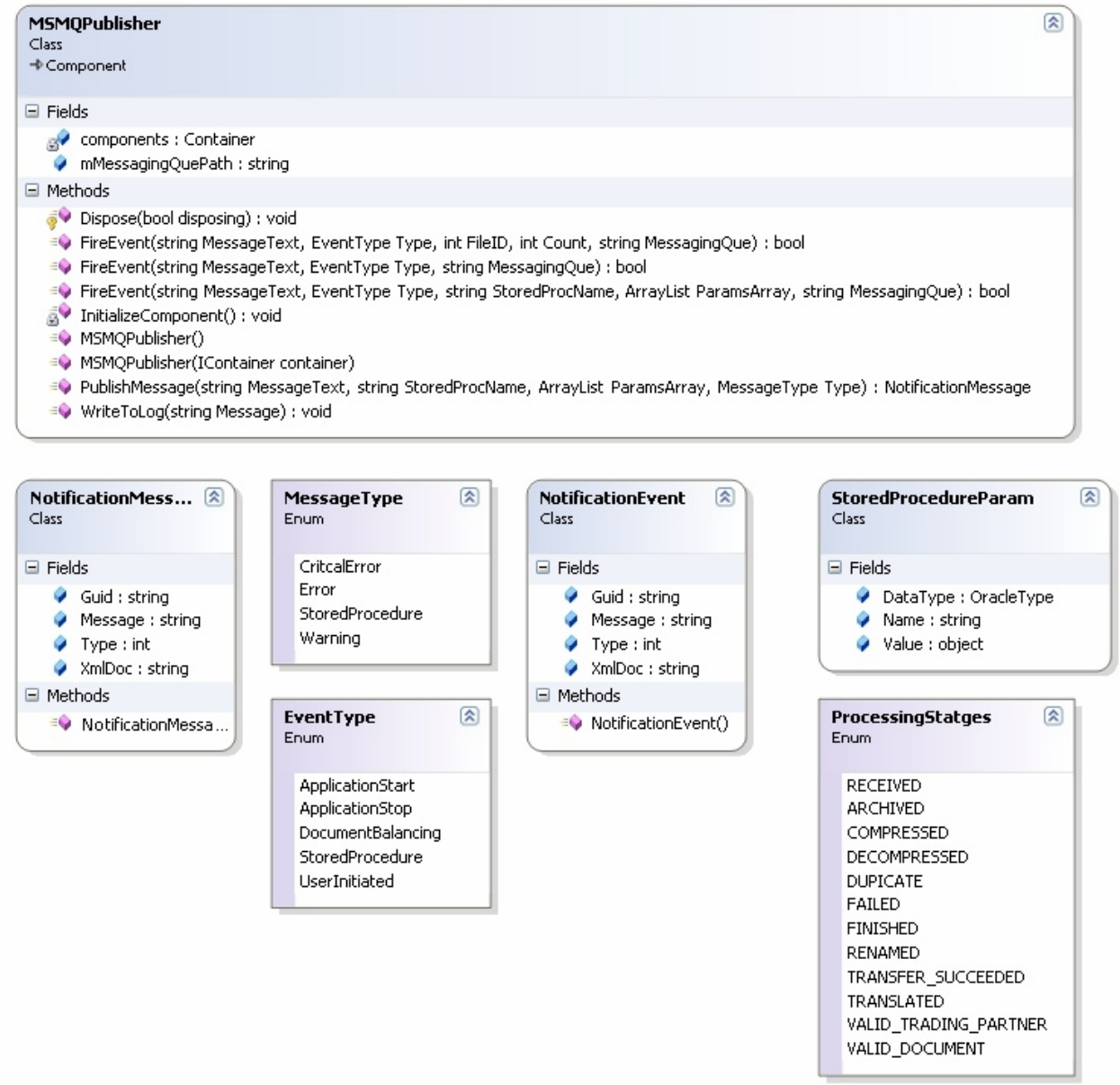
3.3.2.5 EDI_RunTimeObjectBase Abstract Base Class

This is an abstract class which implements MarshalByRefObject class, IDisposable interface and IServiceObject interface. The purpose of this class is to give a layer to add EDI subsystem specific code.

File Name: EDI_RunTimeObjectBase.cs	**Class Name: EDI_RunTimeObjectBase**
Method Name	**Description**
Start	Abstract method.
Stop	Abstract method.
Continue	Abstract method.
Refresh	Abstract method.
Pause	Abstract method.

3.3.3 MSMQ Publisher Project

MSMQ publisher project is called by the MSMQConsumer class and is responsible for setting up the runtime parameters of the MSMQ Listener service.



3.3.3.1 MSMQ Publisher Class

MSMQ Publisher class creates the Notification messages and publishes to configure Messaging Queue.

File Name: MSMQPublisher.cs	**Class Name: MSMQPublisher**
Method Name	**Description**
MSMQPublisher	This is the default constructor of the class.

File Name: MSMQPublisher.cs	**Class Name: MSMQPublisher**
Method Name	**Description**
Publish Message	This method creates instance of Notification Message class, populates the instance with even information, and Publishes the message to Configured Messaging Queue.
Fire Event	This method creates instance of Notification Message class, populates the instance with even information, and Publishes the message to Configured Messaging Queue.
Dispose	This method disposes of the service object.

3.3.3.2 StoredProcedureParam Class

Store Procedure param class is used in MSMQ Publisher Class. This class defines variables required for calling Database calls.

3.3.3.3 Notification Message Class

Notification Message Class is used in MSMQ publisher, MSMQ Consumer classes. This class defines variables required for creating the MSMQ Message.

File Name: Notification Message.cs	**Class Name: Notification Message**
Method Name	**Description**
Notification Message	This is the default constructor of the class.

3.3.3.4 Notification Event Class

Notification Event Class is used in MSMQ publisher, MSMQ Consumer classes. This class defines variables required for creating the MSMQ Message.

File Name: NotificationEvent.cs	**Class Name: Notification Event**
Method Name	**Description**
Notification Event	This is the default constructor of the class.

3.4 Configuration Description

3.4.1 App.Config

```
<configuration> <appSettings>

        <add key="RemotingChannel" value="9090" />

        <add key="InstanceName" value="EDS.MSMQ.TCP" />

        <add key="Namespace" value="root\FileTrackingMSMQNotification" />

        <add key="MessageQueuePaths" value=".\Private$\SimplestExamplePrivateQueue" />

        <add key="DataSource" value="ORACLE" />

        <add key="Configuration" value="C:\Program Files\EDS\MSMQ Listener
Service\MSMQConfig.xml" />

        <add key="DBConnection" value="Data Source=testsource;User ID=testuser; Password=xyz;
Integrated Security=no;" />

        <add key="ReceipientAddress" value="tester-eds@eds.com" />

        <add key="SenderAddress" value="tester-eds@eds.com" />

        <add key="EmailSubject" value="MSMQ Listener SBG" />

        <add key="FileName" value="DocumentBalance.dat" />

        <add key="MSMQPollRate" value="1000" />

        <add key="DebugMode" value="true" />

        <add key="ErrorQueue" value=".\private$\archivefailedqueue" />

</appSettings> </configuration>
```

4 Appendix B – NCPDP Batch and Translator

4.1 Introduction

The National Council for Prescription Drug Programs (NCPDP) defines standards for transmitting data. The Commonwealth of Kentucky has adopted the NCPDP Batch Standard version 1.1 and NCPDP Telecommunication Standard version 5.1 to transmit pharmacy data. While the NCPDP Telecommunication Standard defines the usage of many transaction types the Commonwealth of Kentucky only requires the Billing (B1), Reversal (B2) and Rebill (B3) transactions.

The NCPDP Batch Processing component is composed of the NCPDP Batch Submitter Service and the NCPDP Translator to provide the functionality to translate and process a Batch NCPDP Standard 1.1 request file containing payloads in the NCPDP 5.1 Telecommunication Standard format. Some of the NCPDP Batch Processing component features are explained below.

4.1.1 NCPDP Batch Processing Features

4.1.1.1 Exception Logger

If there are any technical or business exceptions while processing, all errors are tracked and captured in events.

4.1.1.2 NCPDP Batch Standard Version 1.1 Compliance

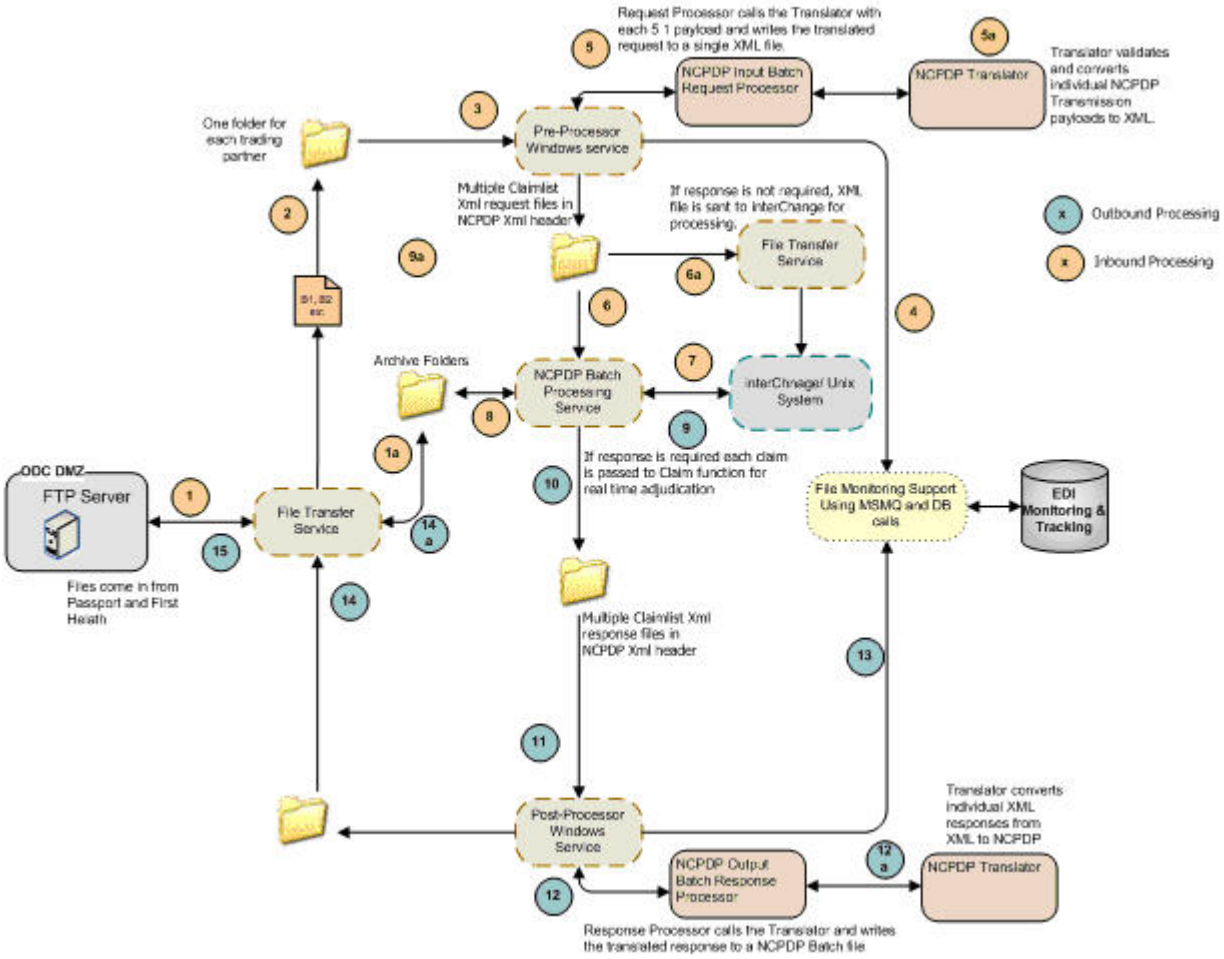
The NCPDP Batch Standard version 1.1 is made up of fields in the Batch Header, Batch Detail and Batch Trailer. A configuration defines all values that are acceptable for each field. These checks are applied against all NCPDP Batch Standard files submitted. In addition, the code has other checks to verify that the Batch file is in compliance.

4.1.1.3 NCPDP Telecommunication Standard Version 5.1 Compliance

The NCPDP Telecommunication Standard version 5.1 is made up transactions, groups, segments and fields. Configuration files define which transactions, segments and fields are used. In addition, a configuration defines all the values that are acceptable for each field.

4.2 NCPDP Batch Processing Solution Diagram

Below diagram shows all the components used in the NCPDP Batch Processing Solution. This document addresses the NCPDP Batch Processing Solution objects in detail.

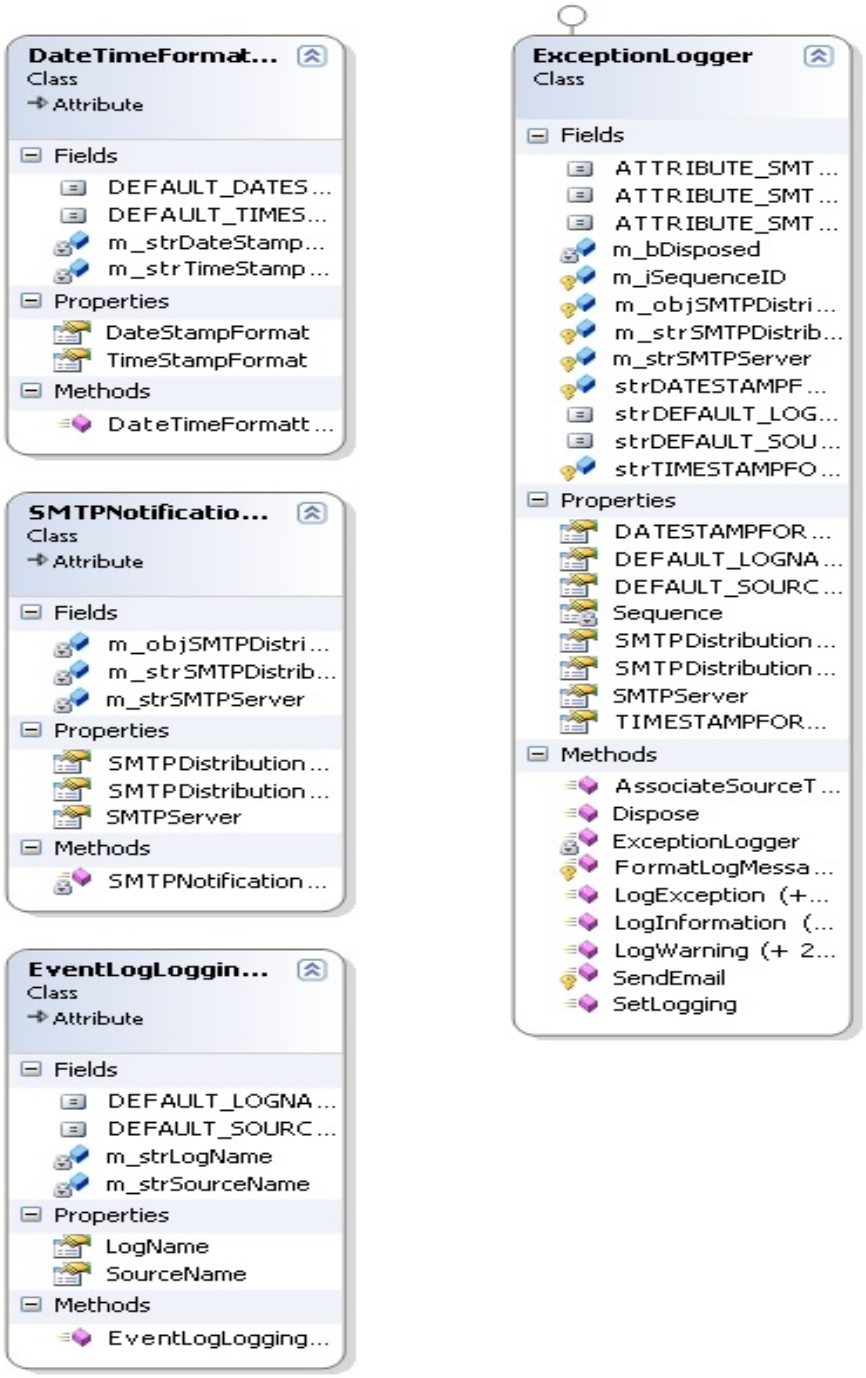


4.3 Projects Description

This section will explain all the projects involved in the development of the NCPDP Batch Processing Solution. It will also further explain the individual projects, classes and the methods involved in supporting the required functionality of the NCPDP Translator Service.

4.3.1 Exception Logger Project

The Exception Logger project contains the logic to log batch information as an event. All events will write to a unique Batch Event Log.



4.3.1.1 Event Log Logging Attribute Class

The Exception Logger File contains multiple classes made up to create a event log and write events to the log. This class defines the name for all Batch Events to write to, and provides the mechanism to write logging information.

File Name: ExceptionLogger.cs	**Class Name: EventLogLoggingAttribute**
Method Name	**Description**
EventLogLoggingAttribute	This is the default constructor of the class and is overloaded with different options in the signature to allow a call to specify the log name or log name and source name.

4.3.1.2 Date Time Formatting Attribute Class

The Exception Logger File contains multiple classes made up to create a event log and write events to the log. This class defines how event times are written.

File Name: ExceptionLogger.cs	**Class Name: DateTimeFormattingAttribute**
Method Name	**Description**
DateTimeFormattingAttribute	This is the default constructor of the class and is overloaded with different options in the signature to allow a call to specify the date stamp format or date stamp format and time stamp format.

4.3.1.3 SMTP Notification Attribute Class

The Exception Logger File contains multiple classes made up to create a event log and write events to the log. This class defines the SMTP Notification.

File Name: ExceptionLogger.cs	**Class Name: SMTPNotificationAttribute**
Method Name	**Description**
SMTPNotificationAttribute	This is the default constructor of the class and is overloaded with different options in the signature to allow a call to specify the SMTP Distribution Level, SMTP Server and SMTP Distribution List.

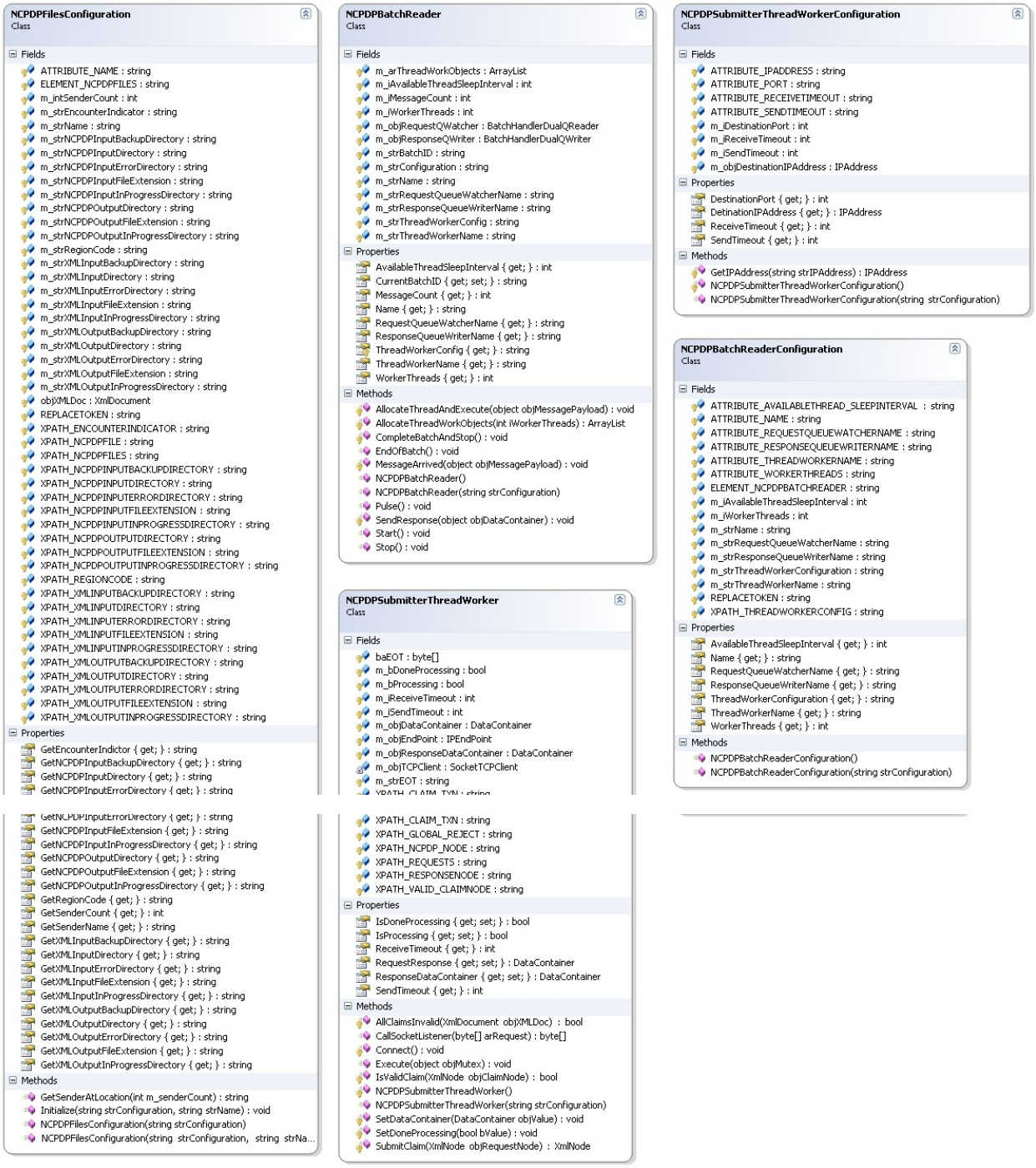
4.3.1.4 Exception Logger Class

The Exception Logger File contains multiple classes made up to create a event log and write events to the log. This class contains the calls to log events.

File Name: ExceptionLogger.cs	**Class Name: ExceptionLogger**
Method Name	**Description**
ExceptionLogger	This is the default constructor of the class.
LogInformation	This method returns the optional fields in the specified segment.
AssociateSourceToLog	This method creates an event log with the appropriate name.

4.3.2 NCPDP Batch Project

The NCPDP Batch Project contains the mechanisms to read and write objects to message queues. It also contains the logic needed to send transactions via a socket to the MMIS.



4.3.2.1 NCPDP Files Configuration Class

This class loads the NCPDP Configuration File into local variables providing an easy way to retrieve values.

File Name: NCPDPFilesConfiguration.cs	**Class Name: NcpdpFilesConfiguration**
Method Name	**Description**
NCPDPFilesConfiguration	This is the default constructor of the class and is overloaded with different options in the signature to allow a call to specify the string configuration or string configuration and submitter.
GetSenderAtLocation	This method allows us to retrieve a senders name in the configuration by sending the submitter location occurrence.
Initialize	This method sets the local variables for a a particular submitter.

4.3.2.2 NCPDP Batch Reader Class

This class is responsible for pulling transactions off of the request queues and calling the appropriate function to execute the transaction.

File Name: NCPDPBatchReader.cs	**Class Name: NcpdpBatchReader**
Method Name	**Description**
NCPDPBatchReader	This is the default constructor.
Start	This method is called to start the service. This method signals to start watching the request queues for transactions.
Stop	This method is called to stop the service. Before the service stops, CompleteBatchAndStop is called. This method signals to stop watching the request queues for transactions.
Pulse	This method iterates through one set of transactions on the request queues.
CompleteBatchAndStop	This method sets the current count to 0.
EndOfBatch	This method sets the appropriate thread condition after completed processing.
MessageArrived	This method sets the value of local variables and calls AllocateThreadAndExecute.
AllocateThreadWorkObjects	This method initializes the number of threads based upon the threads value in the configuration.
AllocateThreadAndExecute	This method spins through the requests with the number of threads allocated calling Execute to send the transaction to be processesd. After processing call SendResponse to

File Name: NCPDPBatchReader.cs	**Class Name: NcpdpBatchReader**
Method Name	**Description**
	create a response entry in the response queue.
SendRespone	This method sends a unique data message to the Response Queue Writer.

4.3.2.3 NCPDP Submitter Thread Worker Class

This class contains the mechanisms to submit individual transactions to the MMIS. It contains logic that verifies that the transaction has not been rejected due to compliance reasons.

File Name: NCPDPSubmitterThreadWorker.cs	**Class Name: NCPDPSubmitterThreadWorker**
Method Name	**Description**
NCPDPDSubmitterThreadWorker	This is the default constructor.
Execute	This method contains the process flow for submitting a single transaction via a socket to the MMIS box.
SubmitClaim	This method submits a single request.
IsValidClaim	This method verifies that the claim does not have a response.
AllClaimsInvalid	This method verifies that all claims have not been rejected due to compliance reasons.
SetDataContainer	This method converts an object into a data container.
SetDoneProcessing	This method indicates that the transaction is done processing.
Connect	This method connects to a listening socket.
CallSocketListener	This method connects to a listening socket, sends the request data and waits until a response is returned with an EOT character at the end. The EOT character indicates the end of both the request and response.

4.3.2.4 NCPDP Batch Reader Configuration Class

This class loads the NCPDP Configuration File into local variables providing an easy way to retrieve values.

File Name: NCPDPBatchReaderConfiguration.cs	**Class Name: NCPDPBatchReaderConfiguration**
Method Name	**Description**
NCPDPBatchReaderConfiguration	This is the default constructor.

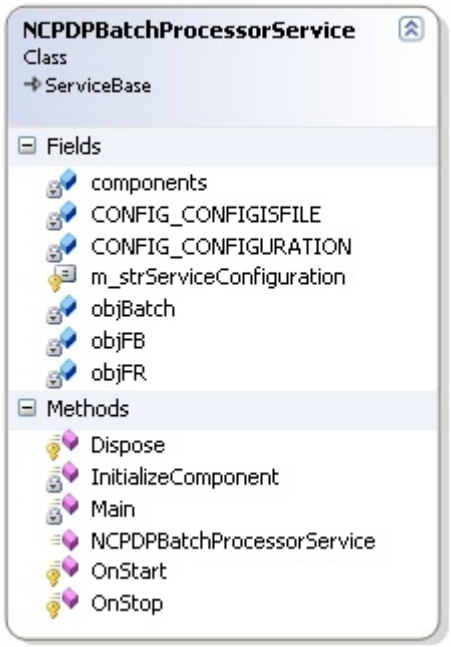
4.3.2.5 NCPDP Submitter Thread Worker Configuration

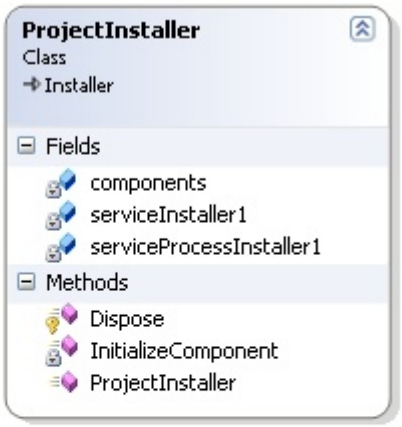
This class loads the NCPDP Configuration File into local variables providing an easy way to retrieve values.

File Name: NCPDPSubmitterThreadWorkerConfiguration.cs	**Class Name: NCPDPSubmitterThreadWorkerConfiguration**
Method Name	**Description**
NCPDPDSubmitterThreadWorkerConfiguration	This is the default constructor.
IPAddress	This method transforms the ip address in a string format to a byte array.

4.3.3 NCPDP Batch Submitter Service Project

The NCPDP Batch Submitter Service Project is responsible for starting the individual programs to process XML transactions. These applications include reading a XML file placing individual transactions on request queues, sending request data on queues to the MMIS and creating individual transactions on response queues, and creating a XML file from transactions on the response queues.





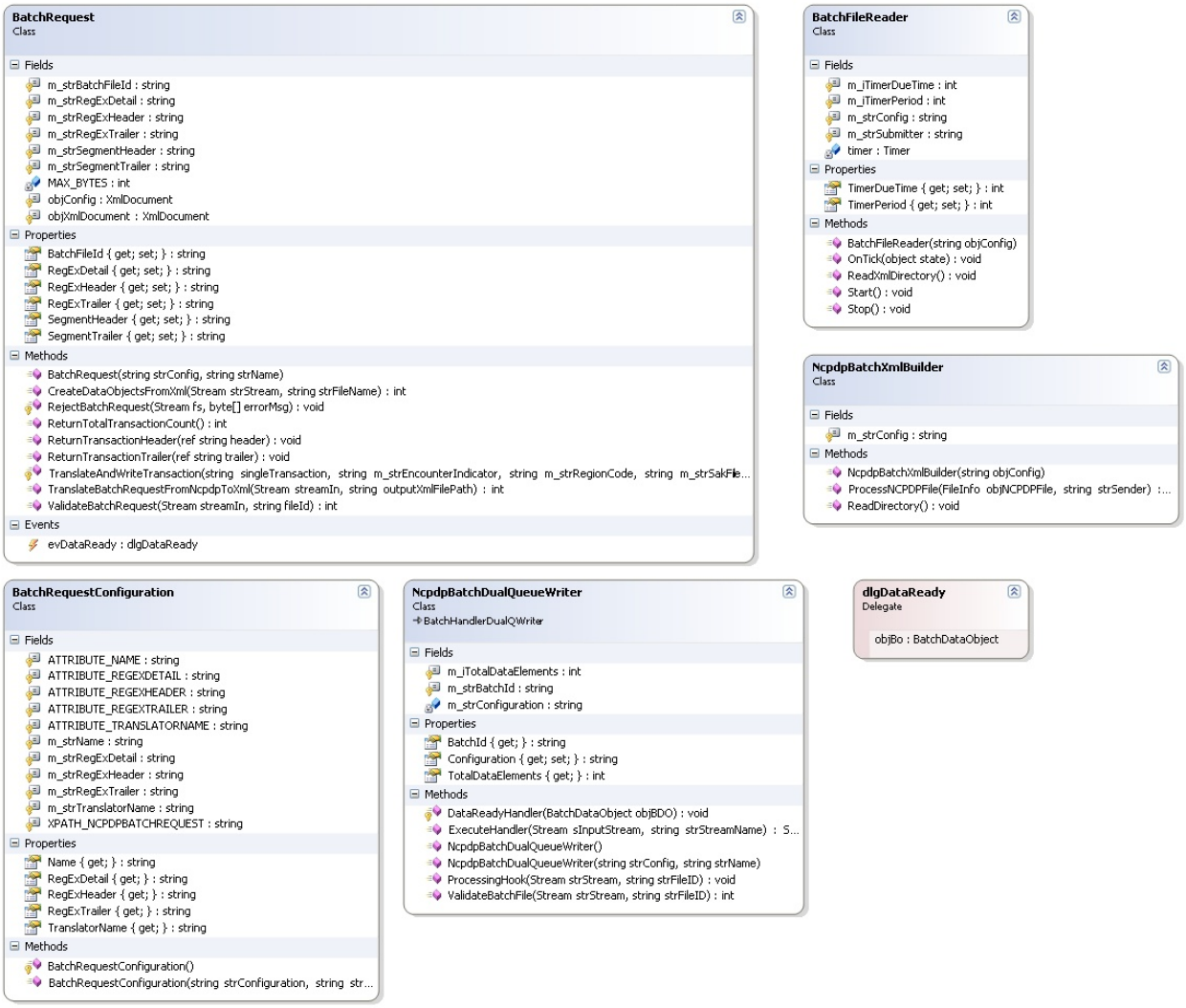
4.3.3.1 NCPDP Batch Processor Service Class

This class is responsible for starting and stopping the BatchFileReader, NCPDPBatchReader and NCPDPFileBuilder.

File Name: NCPDPBatchProcessorService.cs	**Class Name: NCPDPBatchProcessorService**
Method Name	**Description**
NCPDPDBatchProcessorService	This is the default constructor.
Main	This is the main entry point for the process to create a service that starts and stops the processes in the OnStart and OnStop methods.
InitializeComponent	This method defines the services name.
Dispose	This method cleans up any resources being used.
OnStart	This method starts the BatchFileReader by calling Start() NCPDPBatchReader by calling Start() NCPDPFileBuilder by calling Start()
OnStop	This method stops the: BatchFileReader by calling Stop() NCPDPBatchReader by calling Stop() NCPDPFileBuilder by calling Stop()

4.3.4 NCPDP Input Batch Request Project

The NCPDP Input Batch Request Project contains the logic to accept and validate a NCPDP Batch 1.1 file. Once the Batch 1.1 file is validated, each individual payload containing a NCPDP Telecommunication 5.1 transaction is validated and translated to XML. Each individual XML response is appended to a since XML file. This file can be moved directly to the MMIS input directory for processing, or it can also be loaded onto message queues for processing.



4.3.4.1 Batch Request Class

This class provides access to send a batch file and to translate that file payloads into XML.

File Name: BatchRequest.cs	**Class Name: BatchRequest**
Method Name	**Description**
BatchRequest	This is the default constructor.
TranslateBatchRequestFromNcpdpToXml	This method opens up a batch NCPDP file and iterates thru each payload.

File Name: BatchRequest.cs	**Class Name: BatchRequest**
Method Name	**Description**
TranslateAndWriteTransaction	This method validates a batch detail and translates the payload appending each transaction to a xml file.
CreateDataObjectsFromXml	This method stores the translated information about a single payload and other information sending it to a queue.
ReturnTransactionHeader	This method returns the header on a batch file.
ReturnTransactionTrailer	This method returns the trailer on a batch file.
ReturnTotalTransactionCount	This method returns the total count located in the trailer.
ValidateBatchRequest	This method validates the batch header trailer and counts details.
RejectBatchRequest	This method sets the batch transmission indicator to error and add error message to the free text section in the batch trailer.

4.3.4.2 Batch Request Configuration Class

This class loads the NCPDP Configuration File into local variables providing an easy way to retrieve values.

File Name: BatchRequestConfiguration.cs	**Class Name: BatchRequestConfiguration**
Method Name	**Description**
BatchRequestConfiguration	This is the default constructor.

4.3.4.3 NCPDP Batch Dual Queue Writer Class

This class is responsible taking in a XML request file and writing the individual transactions to the data queue.

File Name: NCPDPBatchDualQueueWriter.cs	**Class Name: NCPDPBatchDualQueueWriter**
Method Name	**Description**
NCPDPBatchDualQueueWriter	This is the default constructor.
ExecuteHandler	This method takes in a translated XML file request. It calls ProcessingHook and SendCommandMessage
ValidateBatchFile	Not implemented.
ProcessingHook	This method loops thru all the individual proprietary XML transactions in the file calling DataReadyHandler.
DataReadyHandler	This method sends the data message to the data queue.

4.3.4.4 Batch File Reader Class

This class is responsible for looking for XML translated files in a given directory at intervals. When it sees an XML file, it calls the NCPDPBatchDualQueueWriter class.

File Name: BatchFileReader.cs	**Class Name: BatchFileReader**
Method Name	**Description**
BatchFileReader	This is the default constructor.
Start	This method starts the process for looking for input files.
Stop	This method stops the process for looking for input files
ReadXmlDirectory	This file loops thru all the XML files in a directory invoking the NCPDPBatchDualQueueWriter method ExecuteHandler for each XML file found.
OnTick	This method reads a predefined directory looking for files to process.

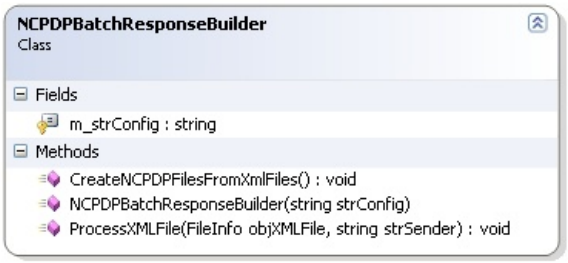
4.3.4.5 Ncpdp Batch Xml Builder Class

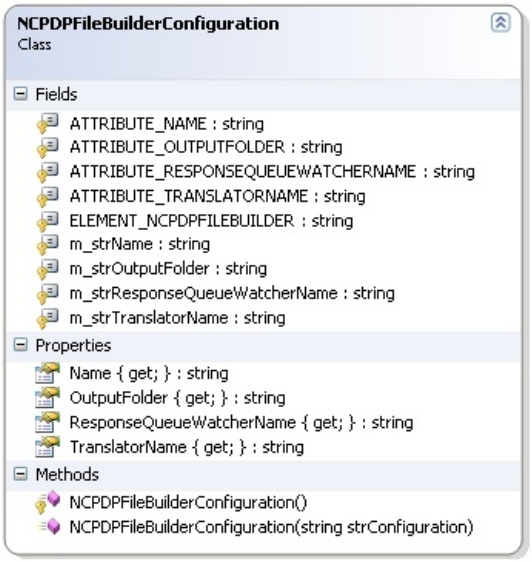
This class is used as the entry point to translate a NCPDP Batch 1.1 file.

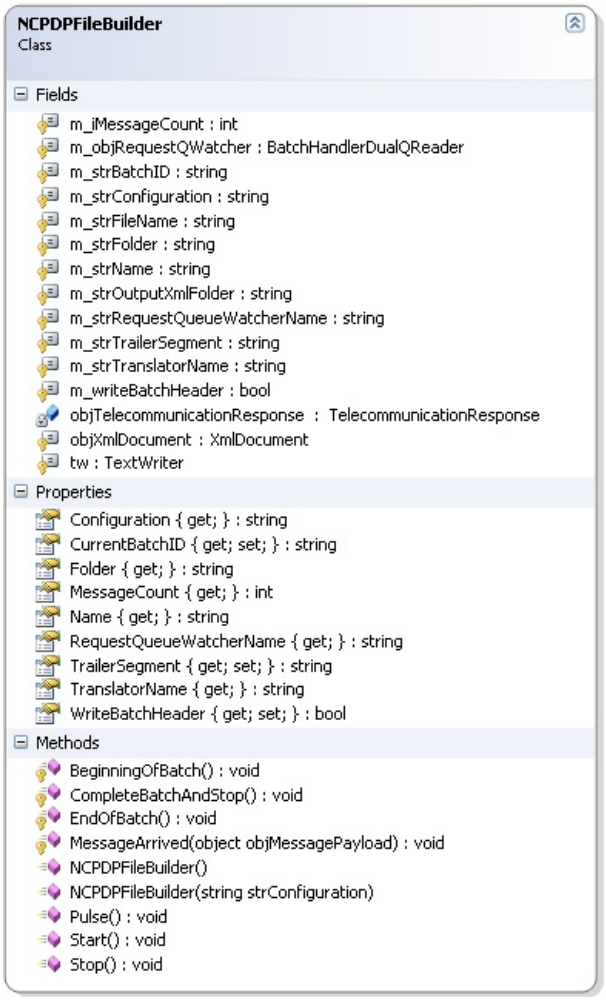
File Name: NcpdpBatchXmlBuilder.cs	**Class Name: NcpdpBatchXmlBuilder**
Method Name	**Description**
NcpdpBatchXmlBuilder	This is the default constructor.
ProcessNCPDPFile	This method is called to process a specified NCPDP Batch 1.1 File. After validating the NCPDP Batch 1.1 File it creates an XML wrapper in and calls the logic to process the file.
ReadDirectory	This method is not needed but kept for future possible use. This method read a particular input directory for NCPDP Batch Files.

4.3.5 NCPDP Output Batch Response Project

The NCPDP Output Batch Response Project contains the logic to create a XML response file based upon individual data objects on the response queue representing single transactions. Once the XML response file is created, it can then translate the XML file into a NCPDP Telecommunication 5.1 response file.







4.3.5.1 NCPDP Batch Response Builder Class

This class is used as the entry point to translate a XML file into a NCPDP Batch 1.1 file.

File Name: NCPDPBatchResponseBuilder.cs	**Class Name: NCPDPBatchResponseBuilder**
Method Name	**Description**
NCPDPBatchResponseBuilder	This is the default constructor.
ProcessXMLFile	Creates a NCPDP response file.
CreateNCPDPFilesFromXmlFiles	This method processes all XML files in a directory by translating them into NCPDP 5.1 responses.

4.3.5.2 NCPDP File Builder Configuration Class

This class loads the NCPDP Configuration File into local variables providing an easy way to retrieve values.

File Name: NCPDPFileBuilderConfiguration.cs	**Class Name: NCPDPFileBuilderConfiguration**
Method Name	**Description**
NCPDPFileBuilderConfiguration	This is the default constructor.

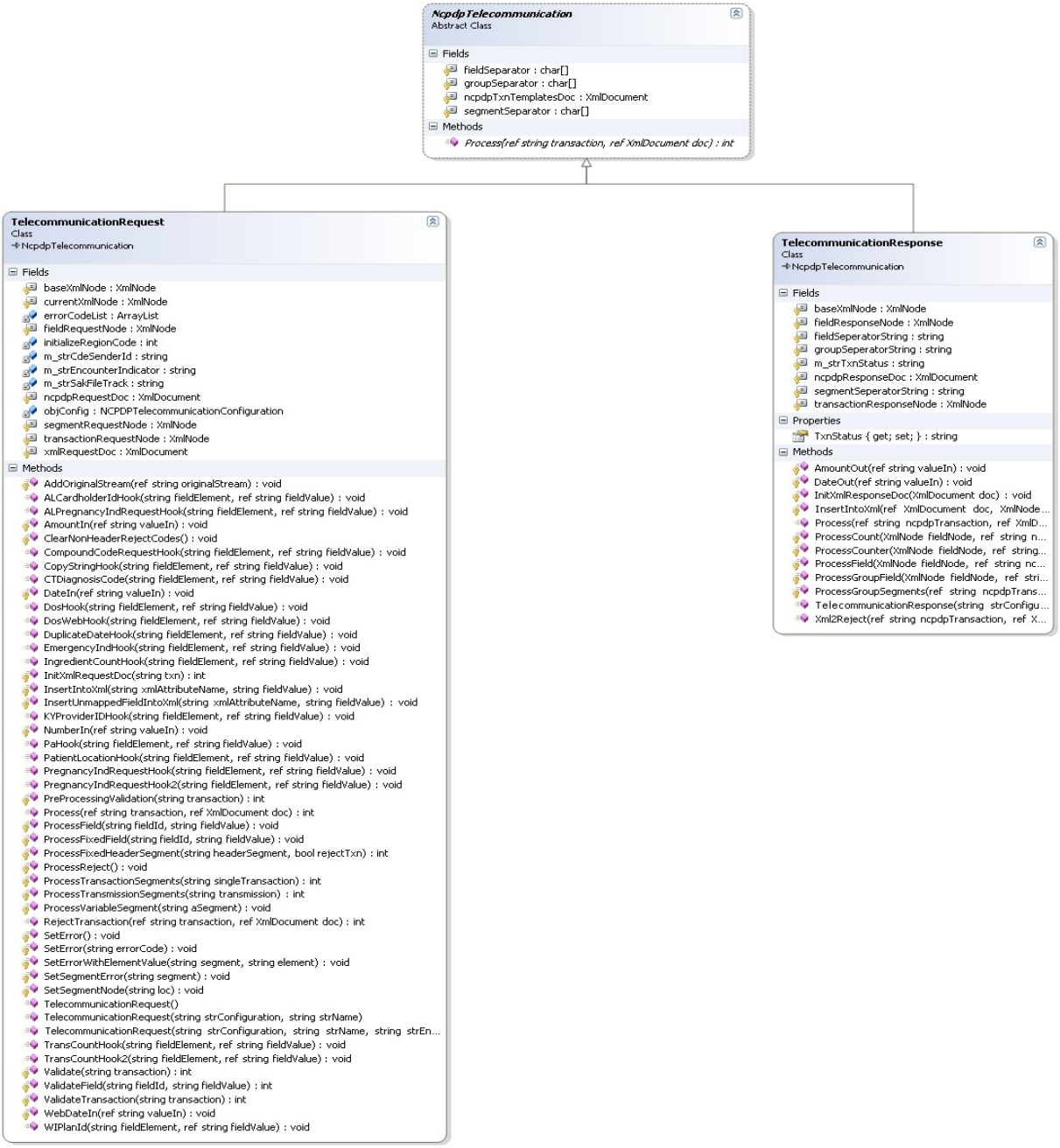
4.3.5.3 NCPDP File Builder Class

This class monitors directories defined in the configuration waiting for XML files. If an XML response file is located, the appropriate logic is called to translate the file into a NCPDP Batch 1.1. file.

File Name: NCPDPFileBuilder.cs	**Class Name: NCPDPFileBuilder**
Method Name	**Description**
NCPDPFileBuilder	This is the default constructor.
Start	This is the method called to start the service looking for response queue messages needing to be processed
Stop	This is the method called to stop the service.
Pulse	This method iterates through one command response message on the queue.
CompleteBatchAndStop	Set the message count indicator to 0
BeginningOfBatch	Creates a XML Batch response file.
EndOfBatch	Saves the XML Batch response file and calls CompleteBatchAndStop.
MessageArrived	Writes the individual data objects to the XML file.

4.3.6 NCPDP Telecommunication Translator Project

The NCPDP Telecommunication Translator project contains the logic to validate and translate individual NCPDP Telecommunication 5.1 requests. The translator leverages multiple configuration files for translating purposes which are listed at the end of this document.



NCPDPTelecommunicationConfiguration
Class

Fields
- ATTRIBUTE_B1ALLOWEDSEGMENTS : string
- ATTRIBUTE_B2ALLOWEDSEGMENTS : string
- ATTRIBUTE_B3ALLOWEDSEGMENTS : string
- ATTRIBUTE_BATCHXMLCONFIGPATH : string
- ATTRIBUTE_E1ALLOWEDSEGMENTS : string
- ATTRIBUTE_NAME : string
- ATTRIBUTE_NCPDPBATCHDUALQWRITERNAME : string
- ATTRIBUTE_NCPDPXMLREQUESTCONFIGPATH : string
- ATTRIBUTE_NCPDPXMLRESPONSECONFIGPATH : string
- ATTRIBUTE_NCPDPXMLTXNTEMPLATESCONFIGPATH : string
- ATTRIBUTE_P2ALLOWEDSEGMENTS : string
- ATTRIBUTE_P3ALLOWEDSEGMENTS : string
- ATTRIBUTE_P4ALLOWEDSEGMENTS : string
- ATTRIBUTE_SUPPORTEDTRANSACTIONS : string
- ATTRIBUTE_TIMERDUETIME : string
- ATTRIBUTE_TIMERPERIOD : string
- m_iTimerDueTime : int
- m_iTimerPeriod : int
- m_strB1AllowedSegments : string
- m_strB2AllowedSegments : string
- m_strB3AllowedSegments : string
- m_strBatchXmlConfigPath : string
- m_strClaimMandatoryFields : string
- m_strClaimOptionalFields : string
- m_strClinicalMandatoryFields : string
- m_strClinicalOptionalFields : string
- m_strCOBMandatoryFields : string
- m_strCOBOptionalFields : string
- m_strCompoundMandatoryFields : string
- m_strCompoundOptionalFields : string
- m_strCouponMandatoryFields : string
- m_strCouponOptionalFields : string
- m_strDURMandatoryFields : string
- m_strDUROptionalFields : string
- m_strE1AllowedSegments : string
- m_strInsuranceMandatoryFields : string
- m_strInsuranceOptionalFields : string
- m_strName : string
- m_strNcpdpBatchDualQWriterName : string
- m_strNcpdpXmlRequestConfigPath : string
- m_strNcpdpXmlResponseConfigPath : string
- m_strNcpdpXmlTxnTemplatesConfigPath : string
- m_strP2AllowedSegments : string
- m_strP3AllowedSegments : string
- m_strP4AllowedSegments : string
- m_strPatientMandatoryFields : string
- m_strPatientOptionalFields : string
- m_strPrescriberMandatoryFields : string
- m_strPrescriberOptionalFields : string
- m_strPricingMandatoryFields : string
- m_strPricingOptionalFields : string
- m_strPriorAuthMandatoryFields : string
- m_strPriorAuthOptionalFields : string
- m_strProviderMandatoryFields : string
- m_strProviderOptionalFields : string
- m_strSupportedTransactions : string
- m_strWorkersCompMandatoryFields : string
- m_strWorkersCompOptionalFields : string
- REPLACETOKEN : string
- XPATH_CLAIMSEGMENT : string
- XPATH_CLINICALSEGMENT : string
- XPATH_COBSEGMENT : string
- XPATH_COMPOUNDSEGMENT : string
- XPATH_COUPONSEGMENT : string
- XPATH_DURSEGMENT : string
- XPATH_INSURANCESEGMENT : string
- XPATH_NCPDPTELECOMMUNICATIONTRANSLATOR : string
- XPATH_PATIENTSEGMENT : string
- XPATH_PRESCRIBERSEGMENT : string
- XPATH_PRICINGSEGMENT : string
- XPATH_PRIORAUTHSEGMENT : string
- XPATH_PROVIDERSEGMENT : string
- XPATH_SEGMENTDEFINITION : string
- XPATH_TRANSACTIONDEFINITION : string
- XPATH_WORKERSCOMPSEGMENT : string

Properties
- BatchXmlConfigPath { get; } : string
- Name { get; } : string
- NcpdpBatchDualQWriterName { get; } : string
- NcpdpXmlRequestConfigPath { get; } : string
- NcpdpXmlResponseConfigPath { get; } : string
- NcpdpXmlTxnTemplatesConfigPath { get; } : string
- SupportedTransactions { get; } : string
- TimerDueTime { get; } : int
- TimerPeriod { get; } : int

Methods
- GetAllowedSegments(string transactionCode) : string
- GetMandatoryFields(string transactionCode) : string
- GetOptionalFields(string transactionCode) : string
- NCPDPTelecommunicationConfiguration()
- NCPDPTelecommunicationConfiguration(string strConfigurati...

BatchDataObject
Class

Fields
- m_bRejectTransaction : bool
- m_strDetail : string
- m_strFileId : string
- m_strHeader : string
- m_strRequest : string
- m_strResponse : string
- m_strSender : string
- m_strTrailer : string

Properties
- Detail { get; set; } : string
- FileId { get; set; } : string
- Header { get; set; } : string
- rejectTransaction { get; set; } : bool
- Request { get; set; } : string
- Response { get; set; } : string
- Sender { get; set; } : string
- Trailer { get; set; } : string

Methods
- BatchDataObject()

NcpdpException
Class
Exception

Methods
- NcpdpException()
- NcpdpException(Exception ex)
- NcpdpException(string strConfiguration)

ExceptionLogger
Class

Fields
- ATTRIBUTE_SMTPDISTRIBUTIONLEVEL : string
- ATTRIBUTE_SMTPDISTRIBUTIONLIST : string
- ATTRIBUTE_SMTPSERVER : string
- DATESTAMPFORMAT : string
- DEFAULT_LOGNAME : string
- DEFAULT_SOURCENAME : string
- m_iSequenceID : int
- m_objSMTPDistributionLevel : EventLogEntryType
- m_strSMTPDistributionList : string
- m_strSMTPServer : string
- TIMESTAMPFORMAT : string

Properties
- Sequence { get; } : int
- SMTPDistributionLevel { get; } : string
- SMTPDistributionList { get; } : string
- SMTPServer { get; } : string

Methods
- AssociateSourceToLog(string strSourceName, string strLogName) : void
- ExceptionLogger()
- FormatLogMessage(string strLogMessage) : string
- FormatLogMessage(string strLogMessage, string strIPAddress, string strPort, string strInstanceID, string strFromEndPoint) : string
- FormatLogMessage(string strLogMessage, string strIPAddress, string strPort, string strInstanceID, string strFromEndPoint, string str...
- LogException(string strExceptionString) : bool
- LogException(string strExceptionString, string LogName) : bool
- LogException(string strExceptionString, string strLogName, string strSourceName) : bool
- LogInformation(string strInformationString) : bool
- LogInformation(string strInformationString, string LogName) : bool
- LogInformation(string strInformationString, string strIPAddress, string strPortID, string strInstanceID, string strFromEndPoint, string...
- LogInformation(string strInformationString, string strIPAddress, string strPortID, string strInstanceID, string strFromEndPoint, string...
- LogInformation(string strInformationString, string strLogName, string strSourceName) : bool
- LogWarning(string strWarningString) : bool
- LogWarning(string strWarningString, string LogName) : bool
- LogWarning(string strWarningString, string strLogName, string strSourceName) : bool
- SendEmail(string strSourcename, string strMessage, EventLogEntryType objEventLogEntryType) : void
- SetLogging(string strSMTPServer, string strSMTPDistributionList, string strSMTPDistributionLevel) : void

4.3.6.1 Ncpdp Telecommunication Class

This class. It contains only the abstract method that all Sub Classes must contain. Process is the main method called to translate between NCPDP and XML.

File Name: NcpdpTelecommunication.cs	**Class Name: NcpdpTelecommunication**
Method Name	**Description**
Process	Defines an abstract method for other sub classes to define

4.3.6.2 Telecommunication Request Class

This class contains the code to Translate from NCPDP 5.1 to XML.

File Name: TelecommunicationRequests.cs	**Class Name: TelecommunicationRequests**
Method Name	**Description**
TelecommunicationRequest	This is the default constructor of the class.
Process	This method defines the calling functionality to create a xml document from a NCPDP 5.1 Telecommunication request.
AmountIn	This method transforms an over punched value into a numeric value.
DateIn	This method transforms a date from ccyymmdd to ccyy-mm-dd
InitXmlRequest	This method initializes the XML based upon the transaction id. It also initializes basic variables based upon the id.
InsertIntoXml	This method creates an attribute with a value in the xml.
InsertUnmappedFieldIntoXml	This method creates an attribute with a value in the xml. This method is used primarly to capture NCPDP fields into a location that were not defined in the mapping.
NumberIn	This method transforms value by inserting a decimal point at a particular location.
ProcessField	This method contains the basic flow for each NCPDP field.
ProcessFixedField	This method contains the basic flow for a fixed length field.
ProcessFixedHeaderSegment	This method process the NCPDP 5.1 fixed header segment.
AddOriginalStream	This method adds the original raw NCPDP 5.1 stream to the xml.
ClearNonHeaderRejectCodes	This method clears the reject codes from the error code list.

File Name: TelecommunicationRequests.cs	Class Name: TelecommunicationRequests
Method Name	**Description**
ProcessReject	This method creates an xml response with appropriate attributes.
RejectTransaction	This method can create an xml response with the appropriate attributes.
PreProcessingValidation	This method contains HIPAA checks that would cause a transaction to fail.
ProcessTransactionSegments	This method loops thru all segments within a group.
ProcessTransmissionSegments	This method loops thru all the transmission segments.
ProcessVariableSegment	This method processes all the elements within a segment.
SetError	This method adds an error code to the xml based upon a reject code in the mapping.
SetError	This method adds an error code to the xml based upon a reject code sent.
SetErrorWithElementValue	This method adds an error code to the xml based upon a segment and reject code sent.
SetSegmentError	This method adds an error code to the xml
SetSegmentNode	This method sets the segment node in the transaction request.
ValidateTransaction	This method contains logic to set HIPAA errors if the NCPDP transaction is found to be at fault.
Validate	This method contains logic to set HIPAA errors if the NCPDP transmission is found to be at fault.
ValidateField	This method contains logic to validate the field matches the regular expression in the maps.
WebDateIn	This method transforms the date value from ccyymmdd to mm/dd/ccyy
CompoundCodeRequestHook	This method sets logic for the compound transaction.
DosHook	This method assigns dates to two locations. It is referenced by the request map.
DosWebHook	This method assigns dates to two locations. It is referenced by the request map.
DuplicateDateHook	This method assigns dates to two locations. It is referenced by the request map.

File Name: TelecommunicationRequests.cs	**Class Name: TelecommunicationRequests**
Method Name	**Description**
EmergencyIndHook	This method sets a attribute in the xml based upon a value of 3 for emergency. It is referenced by the request map.
IngredientCountHook	This method places the ingredient count into to locations. It is referenced by the request map.
PaHook	This method transforms the pa from 11 bytes to not copy the leading 0. It is referenced by the request map.
ALCardholderIdHook	This method transforms the value by placing the first 12 characters into 1 field, and the last character into another field. It is referenced by the request map.
PatientLocationHook	This method transforms the value by placing a Y or N into a field if it matches regular expressions. It is referenced by the request map.
PregnancyIndRequestHook	This method transforms the value by placing a P into an attribute if the field value is 2. Otherwise, it places a space. It is referenced by the request map.
ALPregnancyIndRequestHook	This method transforms the value by placing a Y into an attribute if the field value is 2. Otherwise, it places a N. It is referenced by the request map.
PregnancyIndRequestHook	This method transforms the value by placing a P into an attribute if the field value is 2. Otherwise, it places a N. It is referenced by the request map.
KYProviderIDHook	This method transforms the value by removing the trailing 0's from a 10 digit number. It is referenced by the request map.
TransCountHook	This method transforms the value by placing a 1 if the field value was space and maps to two locations. It is referenced by the request map.
TransCountHook2	This method transforms the value by placing a 1 if the field value was a space. It is referenced by the request map.
WIPlanId	This method transforms the value by adding WISC_ before the value. It is referenced by the request map.
CTDiagnosisCode	This method transforms the value by removing the period. It is referenced by the request map.
CopyStringHook	This method places the value into two locations. It is referenced by the request map.

4.3.6.3 Telecommunication Response Class

This class contains the code to Translate from XML to NCPDP 5.1

File Name: TelecommunicationResponse.cs	**Class Name: TelecommunicationResponse**
Method Name	**Description**
NCPDPTelecommunciationRespons e	This is the default constructor of the class.
Xml2Reject	This method creates an NCPDP rejected response from xml.
Process	This method defines the calling functionality to create a NCPDP 5.1 Telecommunication response from xml.
AmountOut	This method transforms a decimal dollar value into an over punch value.
DateOut	This method transforms a date from mm/dd/ccyy or ccyy-mm-dd to ccyymmdd.
InitXmlResponse	This method initializes the ncpdp response node based upon the transaction id.
InsertIntoXml	This method inserts attributes into the xml and is used in the InitXmlResponse.
ProcessCount	This method loops through a repeating counting group.
ProcessCounter	This method loops through a repeating counter group.
ProcessField	This method maps the xml attribute and value to a ncpdp element.
ProcessGroupField	This method processes an individual field in the ProcessCount or ProcessCounter methods.
ProcessGroupSegments	This method defines the main processing inside a transaction group to create the ncpdp output.

4.3.6.4 NCPDP Telecommunication Configuration Class

This class loads the NCPDP Configuration File into local variables providing an easy way to retrieve values.

File Name: NCPDPTelecommunicationConfiguration.cs	**Class Name: NCDPDPTelecommunicationConfiguration**
Method Name	**Description**
NCPDPTelecommunciationConfigur ation	This is the default constructor of the class.
GetMandatoryFields	This method returns the Mandatory Fields in the specified

File Name: NCPDPTelecommunicationConfiguration.cs	**Class Name: NCDPDPTelecommunicationConfiguration**
Method Name	**Description**
	segment.
GetOptionalFields	This method returns the Optional Fields in the specified segment.
GetAllowedSegments	This method returns the Allowed Segments in the specified transaction.

4.3.6.5 NCPDP Exception Class

This class serves as a sub class from the Exception class.

File Name: NCPDPException.cs	**Class Name: NCDPDPException**
Method Name	**Description**
NCPDPException	The constructor is overloaded to send in no arguments, a configuration or an exception.

4.3.6.6 Exception Logger Class

This class is called to log informational ncpdp exceptions.

File Name: ExceptionLogger.cs	**Class Name: ExceptionLogger**
Method Name	**Description**
ExceptionLogger	This is the default constructor of the class.
SetLogging	This method sets the level of event entry upon a state of error, warning or informational.
LogInformation	This method returns the optional fields in the specified segment.
AssociateSourceToLog	This method creates an event log with the appropriate name.
LogWarning	This method logs a warning.
LogException	This method logs an exception.
FormatLogMessage	This method formats the log entry placing a timestamp, sequence and other information when writing the event.

4.3.6.7 Batch Data Objcect Class

This class provides an easy way to set and retrieve data information.

File Name: BatchDataObject.cs	**Class Name: BatchDataObject**
Method Name	**Description**
BatchDataObject	This is the default constructor of the class.

4.4 NCPDP Batch Processing Configuration Files

The NCPDP Batch Processing requires configuration files. These files indicate where to find NCPDP and XML files. The configurations contain information including the IP and Port to send socket traffic to and validation and mapping rules.

4.4.1 ncpdpBatch.xml Config

This configuration file represents the NCPDP Batch 1.1 fields in the header, trailer and detail.

```
<request>
      <header>
            <field id='880-K4'
                     name='Text Indicator'
               regEx='x02'
         />
         <field id='701'
                     name='Segment Identifier'
               regEx='00'
         />
         <field id='880-K6'
                     name='Transmission Type'
               regEx='T|R|E'
         />
         <field id='880-K1'
                     name='Sender Id'
               regEx='[\x20-\x7E]{24}'
         />
         <field id='806-5C'
                     name='Batch Number'
               regEx='[0-9]{7}'
         />
         <field id='880-K2'
                     name='Creation Date'
               regEx='[0-9]{8}'
         />
         <field id='880-K3'
                     name='Creation Time'
               regEx='[0-9]{4}'
         />
         <field id='702'
                     name='File Type'
               regEx='P|T'
         />
         <field id='102-A2'
                     name='Version/Release Number'
               regEx='[\x20-\x7E]{2}'
         />
         <field id='880-K7'
                     name='Receiver Id'
               regEx='[\x20-\x7E]{24}'
         />
      </header>
      <detail>
            <field id='880-K4'
                     name='Text Indicator'
```

```
                regEx='[\x20-\x7E]{24}'
            />
            <field id='701'
                        name='Segment Identifier'
                regEx='G1'
            />
            <field id='880-K5'
                        name='Transaction Reference Number'
                regEx='[\x20-\x7E]{10}'
            />
        </detail>
        <trailer>
              <field id='880-K4'
                        name='Text Indicator'
                regEx='[\x20-\x7E]{24}'
            />
            <field id='701'
                        name='Segment Identifier'
                regEx='99'
            />
            <field id='806-5C'
                        name='Batch Number'
                regEx='[0-9]{7}'
            />
            <field id='751'
                        name='Record Count'
                regEx='[0-9]{10}'
            />
            <field id='504-F4'
                        name='Message'
                regEx='[\x20-\x7E]{35}'
            />
        </trailer>
</request>
```

File Name: ncpdpBatch.xml	
Attribute Name	**Description**
id	The batch field identification.
name	The batch field name.
regEx	The regular expression to match on what value was sent.

4.4.2 ncpdpRequests.xml config

This configuration file represents the NCPDP Telecommunication 5.1 transactions, segments and fields.

```
<request>
    <transaction id='B1'
                 xmlTemplate='claimList'
                 parentNode='claim'
                 errors='invalid_field'
                 cde_clm_type='P'
                 region='25'
```

```
                element='pPhrm'
    >
        <segment id='00'
                 name='transaction header segment'
                 required='M'
                 level='H'
                 format='F'
                 addSegId='N'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='A1'
                   name='bin number'
                   element='ncpdp'
                   attribute='bin_number'
                   len='6'
                   type='F'
                   regEx='610461'
                   rejectCode='01'
            />
            <field id='A2'
                   name='version'
                   element='ncpdp'
                   attribute='version'
                   len='2'
                   type='F'
                   regEx='51'
                   rejectCode='02'
            />
            <field id='A3'
                   name='transaction code'
                   element='ncpdp'
                   attribute='txn_code'
                   len='2'
                   type='F'
                   regEx='B1'
                   rejectCode='03'
            />
            <field id='A4'
                   name='pcn'
                   element='ncpdp'
                   attribute='pcn'
                   len='10'
                   type='S'
                   regEx='[\x20-\x7E]{10}$'
                   rejectCode='04'
            />
            <field id='A9'
                   name='transaction count'
                   element='ncpdp'
                   attribute='txn_count'
                   len='1'
                   type='H'
                   hook='TransCountHook'
                   element2='pPhrm'
                   attribute2='num_dtl_total'
                   attribute3='num_atomic_txns'
```

```
                regEx='^[1-4]{1}$'
                rejectCode='A9'
        />
        <field id='B2'
                name='provider id qualifier'
                element='ncpdp'
                attribute='prov_id_qual'
                len='2'
                type='F'
                regEx='(^[ ]+$)|(^[0][0-9]$)|(^[1][0-4]$)|99'
                rejectCode='B2'
        />
        <field id='B1'
                name='provider id'
                element='pPhrm/phrmHdrKey'
                attribute='id_provider'
                len='15'
                type='H'
                hook='KYProviderIDHook'
                len1='8'
                regEx='^[0-9 ]+$'
                rejectCode='05'
        />
        <field id='D1'
                name='date of service'
                element='pPhrm'
                attribute='dte_dispense'
                len='8'
                type='H'
                hook='DosHook'
                element2='ncpdp'
                attribute2='date_of_service'
                regEx='^[0-9]{8}$'
                rejectCode='15'
        />
        <field id='AK'
                name='certification id'
                element='ncpdp'
                attribute='cert_id'
                len='10'
                type='S'
                regEx='^[\x20-\x7E]{10}$'
                rejectCode='AK'
        />
    </segment>
    <segment id='01'
            name='patient segment'
            required='O'
            level='H'
            format='V'
            addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='CA'
                name='patient first name'
                element='pPhrm/phrmHdrKey'
```

```
                attribute='clm_recip_fst_nam'
                len='12'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='CA'
        />
        <field id='CB'
                name='patient last name'
                element='pPhrm/phrmHdrKey'
                attribute='clm_lst_nam_recip'
                len='15'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='CB'
        />
        <field id='C7'
                name='patient location'
                element='pPhrm'
                attribute='ind_nurse_home'
                len='2'
                type='H'
                hook='PatientLocationHook'
                element2='pPhrm/phrmHdrKey'
                attribute2='cde_patient_location'
                nursingHomeValues='0|3|4|7'
                regEx='(^[0-9]{1}$)|(^[0][0-9]$)|10|11'
                rejectCode='12'
        />
        <field id='2C'
                name='pregnancy indicator'
                element='pPhrm'
                attribute='ind_pregnancy'
                len='1'
                type='H'
                hook='PregnancyIndRequestHook'
                regEx='^[ 1-2]{1}$'
                rejectCode='2C'
        />
    </segment>
    <segment id='04'
            name='insurance segment'
            required='M'
            level='H'
            format='V'
            addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='C2'
                name='cardholder id'
                element='pPhrm/phrmHdrKey'
                attribute='id_medicaid'
                len='20'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='07'
        />
```

```
        <field id='C1'
               name='group number'
               element='pPhrm/phrmHdrKey'
               attribute='id_prov_rendering'
               len='10'
               type='S'
               regEx='^[\x20-\x7E]+$'
               rejectCode='06'
        />
    </segment>
      <segment id='07'
             name='claim segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='EM'
               name='rx number qualifier'
               element='ncpdp'
               attribute='rx_qual'
               len='1'
               type='F'
               regEx='^[ 12]$'
               rejectCode='EM'
        />
        <field id='D2'
               name='rx number'
               element='pPhrm'
               attribute='num_prscrip'
               len='7'
               dec='0'
               type='S'
               regEx='^[0-9]+$'
               rejectCode='16'
        />
        <field id='E1'
               name='product id qualifier'
               element='ncpdp'
               attribute='ndc_qual'
               len='2'
               type='F'
               regEx='(^[0-9]{1}$)|(^[0][0-9]$)|([ ])|10|11|12|13|99'
               rejectCode='E1'
        />
        <field id='D7'
               name='product id'
               element='pPhrm/detail/phrmDtlKey'
               attribute='cde_ndc'
               len='19'
               type='S'
               regEx='^[\x20-\x7E]+$'
               rejectCode='21'
        />
        <field id='EN'
```

```
            name='prescription service num'
            element='pPhrm/phrmHdrOpt'
            attribute='assoc_num_prscrip'
            len='7'
            dec='0'
            type='N'
            regEx='^[0-9]+$'
            rejectCode='EN'
      />
      <field id='EP'
            name='prescription service date'
            element='pPhrm/phrmHdrOpt'
            attribute='assoc_dte_dispense'
            len='8'
            type='D'
            regEx='^[0-9]{8}$'
            rejectCode='EP'
      />
      <field id='E7'
            name='quantity dispensed'
            element='pPhrm/detail'
            attribute='qty_dispense'
            len='10'
            dec='3'
            type='N'
            regEx='^[0-9]+$'
            rejectCode='E7'
      />
      <field id='D3'
            name='fill number'
            element='pPhrm'
            attribute='qty_refill'
            len='2'
            dec='0'
            type='N'
            regEx='^[0-9]+$'
            rejectCode='17'
      />
      <field id='D5'
            name='days supply'
            element='pPhrm'
            attribute='num_day_supply'
            len='3'
            dec='0'
            type='N'
            regEx='^[0-9]+$'
            rejectCode='19'
      />
      <field id='D6'
            name='compound code'
            element='pPhrm'
            attribute='cde_clm_type'
            len='1'
            type='H'
            hook='CompoundCodeRequestHook'
            parm1='pPhrm/detail/phrmDtlKey'
            parm2='pPhrm/detail'
```

```
            regEx='^[0-2]+$'
            rejectCode='20'
      />
      <field id='D8'
            name='DAW code'
            element='pPhrm'
            attribute='ind_brand_med_nec'
            len='1'
            type='S'
            regEx='^[0-9]+$'
            rejectCode='22'
      />
      <field id='DE'
            name='date prescribed'
            element='pPhrm'
            attribute='dte_prescribe'
            len='8'
            type='D'
            regEx='^[0-9]{8}$'
            rejectCode='28'
      />
      <field id='DI'
            name='level of service'
            element='pPhrm'
            attribute='ind_emergency'
            len='2'
            type='H'
            hook='EmergencyIndHook'
            element2='pPhrm/phrmHdrKey'
            attribute2='level_of_svc'
            regEx='(^[0-6]{1}$)|(^[0][0-6]$)'
            rejectCode='32'
      />
      <field id='DK'
            name='clarification'
            element='pPhrm'
            attribute='cde_clarification'
            len='2'
            type='F'
            regEx='(^[0-9]{1}$)|(^[0][0-9]$)|99'
            rejectCode='34'
      />
      <field id='EU'
            name='PA type code'
            element='ncpdp'
            attribute='pa_type'
            len='2'
            type='F'
            regEx='(^[0-8]{1}$)|(^[0][0-8]$)'
            rejectCode='EU'
      />
      <field id='EV'
            name='PA number'
            element='pPhrm'
            attribute='num_prior_auth'
            len='11'
            type='H'
```

```
                hook='PaHook'
                regEx='^[0-9]+$'
                rejectCode='EV'
        />
        <field id='C8'
                name='other coverage code'
                element='pPhrm/phrmHdrKey'
                attribute='cde_other_coverage'
                len='2'
                type='F'
                regEx='([0][0-8])|(^[0-8]{1}$)'
                rejectCode='13'
        />
        <field id='HD'
                name='dispensing status'
                element='pPhrm'
                attribute='dispense_status'
                len='1'
                type='S'
                regEx='^[PC]{1}$'
                rejectCode='HD'
        />
        <field id='HF'
                name='quantity dispense intended'
                element='pPhrm/phrmHdrOpt'
                attribute='qty_disp_intend'
                len='10'
                dec='3'
                type='N'
                regEx='^[0-9]+$'
                rejectCode='HF'
        />
        <field id='HG'
                name='quantity days intended'
                element='pPhrm/phrmHdrOpt'
                attribute='qty_days_intend'
                len='3'
                dec='0'
                type='N'
                regEx='^[0-9]+$'
                rejectCode='HG'
        />
      </segment>
    <segment id='03'
            name='prescriber segment'
            required='O'
            level='D'
            format='V'
            addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='EZ'
                name='prescriber id qualifier'
                element='ncpdp'
                attribute='prescriber_qual'
                len='2'
```

```
                type='F'
                regEx='(^[ ]+$)|(^[0][0-9]$)|10|11|12|13|14|99'
                rejectCode='EZ'
        />
        <field id='DB'
                name='prescriber id'
                element='pPhrm'
                attribute='id_prov_prescrb'
                len='15'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='56'
        />
    </segment>
      <segment id='05'
              name='COB/other payments segment'
              required='O'
              level='D'
              format='V'
              addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='4C'
                name='COB count'
                element='pPhrm'
                attribute='op_count'
                len='1'
                type='F'
                grouping='C'
                groupElement='phrmCobXref'
                regEx='^[1-9]{1}$'
                rejectCode='4C'
        >
            <field id='5C'
                    name='other payer type'
                    element='phrmCobXref'
                    attribute='cde_op_cov_type'
                    len='2'
                    type='F'
                    groupElement='phrmCobXref'
                    regEx='(^[ ]+$)|(^[0][123])|(^[9][89])'
                    rejectCode='5C'
            />
            <field id='E8'
                    name='other payer date'
                    element='phrmCobXref'
                    attribute='dte_other_payer'
                    len='8'
                    type='D'
                    regEx='^[0-9]{8}$'
                    rejectCode='E8'
            />
            <field id='HB'
                    name='other amount paid count'
                    element='phrmCobXref'
                    attribute='op_amt_paid_cnt'
```

```
                    len='1'
                    type='F'
                    grouping='C'
                    groupElement='phrmCob'
                    regEx='^[0-9]{1}$'
                    rejectCode='HB'
            >
                <field id='HC'
                       name='other amount paid qualifier'
                       element='phrmCob'
                       attribute='cde_op_amt_paid_qual'
                       len='2'
                       type='F'
                       groupElement='phrmCob'
                       regEx='(^[ ]+$)|(^[0][1-8])|(^[9][89])'
                       rejectCode='HC'
                />
                <field id='DV'
                       name='other amount paid'
                       element='phrmCob'
                       attribute='amt_op_paid'
                       len='8'
                       dec='2'
                       type='A'
                       regEx='^[0-9]+[{}A-R]$|[{}A-R]'
                       rejectCode='DV'
                />
            </field>
        </field>
      </segment>
    <segment id='08'
             name='DUR/PPS segment'
             required='O'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='7E'
               name='DUR counter'
               element='pPhrm'
               attribute='dur_cde_counter'
               len='1'
               type='F'
               grouping='R'
               groupElement='phrmDur'
               regEx='^[1-9]{1}$'
               rejectCode='7E'
        >
                    <field id='E4'
               name='reason for service'
               element='phrmDur'
               attribute='reason_for_svc_cde'
               len='2'
               type='H'
               hook='CopyStringHook'
```

```
                    element2='pPhrm'
                    attribute2='cde_ncpdp_conflict'
                    groupElement='phrmDur'

regEx='(^[A][DNRT]$)|(^[C][DHS]$)|(^[D][ACDFILMS]$)|(^[E][DRX]$)|HD|ID|LD|LK|
LR|MC|MN|MS|MX|(^[N][ACDFNPRS]$)|OH|(^[P][ACGHNPRS]$)|RE|RF|(^[S][CDEFRX]$)|(
^[T][DNP]$)'

                    rejectCode='E4'
                         />
                         <field id='E5'
                    name='professional service code'
                    element='phrmDur'
                    attribute='professional_svc_cde'
                    len='2'
                    type='H'
                    hook='CopyStringHook'
                    element2='pPhrm'
                    attribute2='cde_dur_intrvntn'
                    groupElement='phrmDur'

regEx='00|AS|CC|DE|FE|GP|MA|M0|MR|PE|PH|PM|P0|PT|R0|RT|SC|SW|TC|TH'
                    rejectCode='E5'
                         />
                         <field id='E6'
                    name='result of service code'
                    element='phrmDur'
                    attribute='result_of_svc_cde'
                    len='2'
                    type='H'
                    hook='CopyStringHook'
                    element2='pPhrm'
                    attribute2='cde_dur_outcome'
                    groupElement='phrmDur'
                    regEx='([00])|(^[1][A-K]$)|(^[2][AB]$)|(^[3][A-KMN]$)'
                    rejectCode='E6'
                         />
                   </field>
        </segment>
        <segment id='11'
                 name='pricing segment'
                 required='M'
                 level='D'
                 format='V'
                 addSegId='Y'
        >
            <status value='A'/>
            <status value='R'/>
             <field id='E3'
                    name='amount incentive'
                    element='pPhrm/phrmHdrOpt'
                    attribute='amt_incentive'
                    len='8'
                    dec='2'
                    type='A'
                    regEx='^[0-9]+[{}A-R]$'
                    rejectCode='E3'
```

```
        />
        <field id='DQ'
               name='amount billed'
               element='pPhrm'
               attribute='amt_billed'
               len='8'
               dec='2'
               type='A'
               regEx='^[0-9]+[{}A-R]$'
               rejectCode='DQ'
        />
        <field id='DU'
               name='amount billed'
               element='pPhrm/phrmHdrOpt'
               attribute='amt_gross_due'
               len='8'
               dec='2'
               type='A'
               regEx='^[0-9]+[{}A-R]$'
               rejectCode='DU'
        />
    </segment>
    <segment id='10'
             name='compound segment'
             required='O'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='EF'
               name='compound dosage code'
               element='ncpdp'
               attribute='cmpd_dose_code'
               len='2'
               type='F'
               regEx='(^[ ]+$)|(^[0][1-9])|(^[1][0-8])'
               rejectCode='EF'
        />
        <field id='EG'
               name='compound dispensing form'
               element='ncpdp'
               attribute='cmpd_disp_form'
               len='1'
               type='F'
               regEx='^[123]{1}$'
               rejectCode='EG'
        />
        <field id='EH'
               name='compound route of administration'
               element='pPhrm/phrmHdrOpt'
               attribute='cde_cmpd_rte_admin'
               len='2'
               type='F'
               regEx='(^[0-9]$)|(^[0][0-9]$)|(^[1][0-9]$)|(^[2][012]$)'
               rejectCode='EH'
```

```
/>
<field id='EC'
       name='compound ingredient count'
       element='pPhrm'
       attribute='num_dtl_total'
       len='2'
       type='H'
       hook='IngredientCountHook'
       element2='ncpdp'
       attribute2='num_atomic_txns'
       grouping='C'
       groupElement='detail'
       regEx='(^[0-9]$)|(^[0][0-9]$)'
       rejectCode='EC'
>
    <field id='RE'
           name='compound id qualifier'
           element='phrmDtlKey'
           attribute='cmpd_id_qual'
           len='2'
           type='F'
           groupElement='detail'
           regEx='(^[ ]+$)|(^[0][0-5]$)|11|12|13|99'
           rejectCode='RE'
    />
    <field id='TE'
           name='compound id'
           element='phrmDtlKey'
           attribute='cde_ndc'
           len='19'
           type='S'
           regEx='^[\x20-\x7E]+$'
           rejectCode='TE'
    />
    <field id='ED'
           name='compound ingredient quantity'
           element='detail'
           attribute='qty_dispense'
           len='10'
           dec='3'
           type='N'
           regEx='^[0-9]+$'
           rejectCode='ED'
    />
    <field id='EE'
           name='compound ingredient cost'
           element='detail'
           attribute='ingred_cost'
           len='8'
           dec='2'
           type='A'
           regEx='^[0-9]+[{}A-R]$'
           rejectCode='EE'
    />
    <field id='UE'
           name='compound ingredient cost basis'
           element='detail'
```

```
                    attribute='ingred_cost_basis'
                    len='2'
                    type='F'
                    regEx='(^[ ]+$)|(^[0][0-9]$)|(^[0-9]$)'
                    rejectCode='UE'
                />
            </field>
        </segment>
        <segment id='13'
                 name='clinical segment'
                 required='O'
                 level='D'
                 format='V'
                 addSegId='Y'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='VE'
                   name='diagnosis code count'
                   element='pPhrm'
                   attribute='diag_count'
                   len='1'
                   type='F'
                   grouping='C'
                   groupElement='diag_xref'
                   regEx='^[0-9]{1}$'
                   rejectCode='VE'
            >
                <field id='WE'
                       name='diagnosis code qualifier'
                       element='diag_xref'
                       attribute='diag_qual'
                       len='2'
                       type='F'
                       groupElement='diag_xref'
                       regEx='(^[ ]+$)|(^[0][0-7]$)|99'
                       rejectCode='WE'
                />
                <field id='DO'
                       name='diagnosis code'
                       element='diag_xref'
                       attribute='cde_diag'
                       len='15'
                       type='S'
                       regEx='^[ .0-9]+$'
                       rejectCode='39'
                />
                <field id='XE'
                       name='clinical counter'
                       element='diag_xref'
                       attribute='clinical_counter'
                       len='1'
                       dec='0'
                       type='N'
                       regEx='^[0-9]{1}$'
                       rejectCode='XE'
                />
```

```
                <field id='ZE'
                       name='measure date'
                       element='diag_xref'
                       attribute='measure_date'
                       len='8'
                       type='D'
                       regEx='^[0-9]{8}$'
                       rejectCode='ZE'
                />
                <field id='H1'
                       name='measure time'
                       element='diag_xref'
                       attribute='measure_time'
                       len='4'
                       dec='0'
                       type='N'
                       regEx='^[0-9]+$'
                       rejectCode='H1'
                />
                <field id='H2'
                       name='measure dim'
                       element='diag_xref'
                       attribute='measure_dim'
                       len='2'
                       type='F'
                       regEx='(^[ ]+$)|(^[0][1-9])|(^[12][0-
9])|(^[3][01234])|(99)'
                       rejectCode='H2'
                />
                <field id='H3'
                       name='measure unit'
                       element='diag_xref'
                       attribute='measure_unit'
                       len='2'
                       type='F'
                       regEx='(^[ ]+$)|(^[ ][1-9]$)|(^[0][1-9])|(^[1][0-
9])|(^[2][0-7])'
                       rejectCode='H3'
                />
                <field id='H4'
                       name='measure value'
                       element='diag_xref'
                       attribute='measure_value'
                       len='15'
                       type='S'
                       regEx='^[\x20-\x7E]+$'
                       rejectCode='H4'
                />
            </field>
        </segment>
    </transaction>
    <transaction id='B2'
                 xmlTemplate='claimList'
                 parentNode='claim'
                 errors='invalid_field'
                 cde_clm_type='R'
                 region='59'
```

```
                element='pPhrm'
    >
        <segment id='00'
                 name='transaction header segment'
                 required='M'
                 level='H'
                 format='F'
                 addSegId='N'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='A1'
                   name='bin number'
                   element='ncpdp'
                   attribute='bin_number'
                   len='6'
                   type='F'
                   regEx='610461'
                   rejectCode='01'
            />
            <field id='A2'
                   name='version'
                   element='ncpdp'
                   attribute='version'
                   len='2'
                   dec='0'
                   type='F'
                   regEx='51'
                   rejectCode='02'
            />
            <field id='A3'
                   name='transaction code'
                   element='ncpdp'
                   attribute='txn_code'
                   len='2'
                   dec='0'
                   type='F'
                   regEx='B2'
                   rejectCode='03'
            />
            <field id='A4'
                   name='pcn'
                   element='ncpdp'
                   attribute='pcn'
                   len='10'
                   dec='0'
                   type='S'
                   regEx='[\x20-\x7E]{10}$'
                   rejectCode='04'
            />
            <field id='A9'
                   name='transaction count'
                   element='ncpdp'
                   attribute='txn_count'
                   len='1'
                   dec='0'
                   type='H'
```

```
                hook='TransCountHook'
                element2='pPhrm'
                attribute2='num_dtl_total'
                attribute3='num_atomic_txns'
                regEx='^[1-4]{1}$'
                rejectCode='A9'
        />
        <field id='B2'
                name='provider id qualifier'
                element='ncpdp'
                attribute='prov_id_qual'
                len='2'
                dec='0'
                type='F'
                regEx='(^[ ]+$)|(^[0][0-9]$)|(^[1][0-4]$)|99'
                rejectCode='B2'
        />
        <field id='B1'
                name='provider id'
                element='pPhrm/phrmHdrKey'
                attribute='id_provider'
                len='15'
                type='H'
                hook='KYProviderIDHook'
                len1='8'
                regEx='^[0-9 ]+$'
                rejectCode='05'
        />
        <field id='D1'
                name='date of service'
                element='pPhrm'
                attribute='dte_dispense'
                len='8'
                dec='0'
                type='H'
                hook='DosHook'
                element2='ncpdp'
                attribute2='date_of_service'
                regEx='^[0-9]{8}$'
                rejectCode='15'
        />
        <field id='AK'
                name='certification id'
                element='ncpdp'
                attribute='cert_id'
                len='10'
                dec='0'
                type='S'
                regEx='^[\x20-\x7E]{10}$'
                rejectCode='AK'
        />
    </segment>
      <segment id='07'
            name='claim segment'
            required='M'
            level='D'
            format='V'
```

```
                addSegId='Y'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='EM'
                   name='rx number qualifier'
                   element='ncpdp'
                   attribute='rx_qual'
                   len='1'
                   dec='0'
                   type='F'
                   regEx='^[ 12]$'
                   rejectCode='EM'
            />
            <field id='D2'
                   name='rx number'
                   element='pPhrm'
                   attribute='num_prscrip'
                   len='7'
                   dec='0'
                   type='S'
                   regEx='^[0-9]+$'
                   rejectCode='16'
            />
            <field id='E1'
                   name='product id qualifier'
                   element='ncpdp'
                   attribute='ndc_qual'
                   len='2'
                   dec='0'
                   type='F'
                   regEx='(^[0-9]{1}$)|(^[0][0-9]$)|([ ])|10|11|12|13|99'
                   rejectCode='E1'
            />
            <field id='D7'
                   name='product id'
                   element='pPhrm/detail/phrmDtlKey'
                   attribute='cde_ndc'
                   len='19'
                   dec='0'
                   type='S'
                   regEx='^[\x20-\x7E]+$'
                   rejectCode='21'
            />
      </segment>
    </transaction>
     <transaction id='B3'
                xmlTemplate='claimList'
                parentNode='claim'
                errors='invalid_field'
                cde_clm_type='P'
                region='25'
                element='pPhrm'
    >
        <segment id='00'
                name='transaction header segment'
                required='M'
```

```
                level='H'
                format='F'
                addSegId='N'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='A1'
                   name='bin number'
                   element='ncpdp'
                   attribute='bin_number'
                   len='6'
                   type='F'
                   regEx='610461'
                   rejectCode='01'
            />
            <field id='A2'
                   name='version'
                   element='ncpdp'
                   attribute='version'
                   len='2'
                   type='F'
                   regEx='51'
                   rejectCode='02'
            />
            <field id='A3'
                   name='transaction code'
                   element='ncpdp'
                   attribute='txn_code'
                   len='2'
                   type='F'
                   regEx='B3'
                   rejectCode='03'
            />
            <field id='A4'
                   name='pcn'
                   element='ncpdp'
                   attribute='pcn'
                   len='10'
                   type='S'
                   regEx='[\x20-\x7E]{10}$'
                   rejectCode='04'
            />
            <field id='A9'
                   name='transaction count'
                   element='ncpdp'
                   attribute='txn_count'
                   len='1'
                   type='H'
                   hook='TransCountHook'
                   element2='pPhrm'
                   attribute2='num_dtl_total'
                   attribute3='num_atomic_txns'
                   regEx='^[1-4]{1}$'
                   rejectCode='A9'
            />
            <field id='B2'
                   name='provider id qualifier'
```

```
                element='ncpdp'
                attribute='prov_id_qual'
                len='2'
                type='F'
                regEx='(^[ ]+$)|(^[0][0-9]$)|(^[1][0-4]$)|99'
                rejectCode='B2'
        />
        <field id='B1'
                name='provider id'
                element='pPhrm/phrmHdrKey'
                attribute='id_provider'
                len='15'
                type='H'
                hook='KYProviderIDHook'
                len1='8'
                regEx='^[0-9 ]+$'
                rejectCode='05'
        />
        <field id='D1'
                name='date of service'
                element='pPhrm'
                attribute='dte_dispense'
                len='8'
                type='H'
                hook='DosHook'
                element2='ncpdp'
                attribute2='date_of_service'
                regEx='^[0-9]{8}$'
                rejectCode='15'
        />
        <field id='AK'
                name='certification id'
                element='ncpdp'
                attribute='cert_id'
                len='10'
                type='S'
                regEx='^[\x20-\x7E]{10}$'
                rejectCode='AK'
        />
    </segment>
    <segment id='01'
            name='patient segment'
            required='O'
            level='H'
            format='V'
            addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='CA'
                name='patient first name'
                element='pPhrm/phrmHdrKey'
                attribute='clm_recip_fst_nam'
                len='12'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='CA'
```

```
        />
        <field id='CB'
               name='patient last name'
               element='pPhrm/phrmHdrKey'
               attribute='clm_lst_nam_recip'
               len='15'
               type='S'
               regEx='^[\x20-\x7E]+$'
               rejectCode='CB'
        />
        <field id='C7'
               name='patient location'
               element='pPhrm'
               attribute='ind_nurse_home'
               len='2'
               type='H'
               hook='PatientLocationHook'
               element2='pPhrm/phrmHdrKey'
               attribute2='cde_patient_location'
               nursingHomeValues='0|3|4|7'
               regEx='(^[0-9]{1}$)|(^[0][0-9]$)|10|11'
               rejectCode='12'
        />
        <field id='2C'
               name='pregnancy indicator'
               element='pPhrm'
               attribute='ind_pregnancy'
               len='1'
               type='H'
               hook='PregnancyIndRequestHook'
               regEx='^[ 1-2]{1}$'
               rejectCode='2C'
        />
    </segment>
    <segment id='04'
             name='insurance segment'
             required='M'
             level='H'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='C2'
               name='cardholder id'
               element='pPhrm/phrmHdrKey'
               attribute='id_medicaid'
               len='20'
               type='S'
               regEx='^[\x20-\x7E]+$'
               rejectCode='07'
        />
        <field id='C1'
               name='group number'
               element='pPhrm/phrmHdrKey'
               attribute='id_prov_rendering'
               len='10'
```

```
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='06'
        />
    </segment>
      <segment id='07'
            name='claim segment'
            required='M'
            level='D'
            format='V'
            addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='EM'
                name='rx number qualifier'
                element='ncpdp'
                attribute='rx_qual'
                len='1'
                type='F'
                regEx='^[ 12]$'
                rejectCode='EM'
        />
        <field id='D2'
                name='rx number'
                element='pPhrm'
                attribute='num_prscrip'
                len='7'
                dec='0'
                type='S'
                regEx='^[0-9]+$'
                rejectCode='16'
        />
        <field id='E1'
                name='product id qualifier'
                element='ncpdp'
                attribute='ndc_qual'
                len='2'
                type='F'
                regEx='(^[0-9]{1}$)|(^[0][0-9]$)|([ ])|10|11|12|13|99'
                rejectCode='E1'
        />
        <field id='D7'
                name='product id'
                element='pPhrm/detail/phrmDtlKey'
                attribute='cde_ndc'
                len='19'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='21'
        />
        <field id='EN'
                name='prescription service num'
                element='pPhrm/phrmHdrOpt'
                attribute='assoc_num_prscrip'
                len='7'
                dec='0'
```

```
            type='N'
            regEx='^[0-9]+$'
            rejectCode='EN'
      />
      <field id='EP'
            name='prescription service date'
            element='pPhrm/phrmHdrOpt'
            attribute='assoc_dte_dispense'
            len='8'
            type='D'
            regEx='^[0-9]{8}$'
            rejectCode='EP'
      />
      <field id='E7'
            name='quantity dispensed'
            element='pPhrm/detail'
            attribute='qty_dispense'
            len='10'
            dec='3'
            type='N'
            regEx='^[0-9]+$'
            rejectCode='E7'
      />
      <field id='D3'
            name='fill number'
            element='pPhrm'
            attribute='qty_refill'
            len='2'
            dec='0'
            type='N'
            regEx='^[0-9]+$'
            rejectCode='17'
      />
      <field id='D5'
            name='days supply'
            element='pPhrm'
            attribute='num_day_supply'
            len='3'
            dec='0'
            type='N'
            regEx='^[0-9]+$'
            rejectCode='19'
      />
      <field id='D6'
            name='compound code'
            element='pPhrm'
            attribute='cde_clm_type'
            len='1'
            type='H'
            hook='CompoundCodeRequestHook'
            parm1='pPhrm/detail/phrmDtlKey'
            parm2='pPhrm/detail'
            regEx='^[0-2]+$'
            rejectCode='20'
      />
      <field id='D8'
            name='DAW code'
```

```
            element='pPhrm'
            attribute='ind_brand_med_nec'
            len='1'
            type='S'
            regEx='^[0-9]+$'
            rejectCode='22'
      />
      <field id='DE'
            name='date prescribed'
            element='pPhrm'
            attribute='dte_prescribe'
            len='8'
            type='D'
            regEx='^[0-9]{8}$'
            rejectCode='28'
      />
      <field id='DI'
            name='level of service'
            element='pPhrm'
            attribute='ind_emergency'
            len='2'
            type='H'
            hook='EmergencyIndHook'
            element2='pPhrm/phrmHdrKey'
            attribute2='level_of_svc'
            regEx='(^[0-6]{1}$)|(^[0][0-6]$)'
            rejectCode='32'
      />
      <field id='DK'
            name='clarification'
            element='pPhrm'
            attribute='cde_clarification'
            len='2'
            type='F'
            regEx='(^[0-9]{1}$)|(^[0][0-9]$)|99'
            rejectCode='34'
      />
      <field id='EU'
            name='PA type code'
            element='ncpdp'
            attribute='pa_type'
            len='2'
            type='F'
            regEx='(^[0-8]{1}$)|(^[0][0-8]$)'
            rejectCode='EU'
      />
      <field id='EV'
            name='PA number'
            element='pPhrm'
            attribute='num_prior_auth'
            len='11'
            type='H'
            hook='PaHook'
            regEx='^[0-9]+$'
            rejectCode='EV'
      />
      <field id='C8'
```

```
                name='other coverage code'
                element='pPhrm/phrmHdrKey'
                attribute='cde_other_coverage'
                len='2'
                type='F'
                regEx='([0][0-8])|(^[0-8]{1}$)'
                rejectCode='13'
        />
        <field id='HD'
                name='dispensing status'
                element='pPhrm'
                attribute='dispense_status'
                len='1'
                type='S'
                regEx='^[PC]{1}$'
                rejectCode='HD'
        />
        <field id='HF'
                name='quantity dispense intended'
                element='pPhrm/phrmHdrOpt'
                attribute='qty_disp_intend'
                len='10'
                dec='3'
                type='N'
                regEx='^[0-9]+$'
                rejectCode='HF'
        />
        <field id='HG'
                name='quantity days intended'
                element='pPhrm/phrmHdrOpt'
                attribute='qty_days_intend'
                len='3'
                dec='0'
                type='N'
                regEx='^[0-9]+$'
                rejectCode='HG'
        />
      </segment>
    <segment id='03'
            name='prescriber segment'
            required='O'
            level='D'
            format='V'
            addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='EZ'
                name='prescriber id qualifier'
                element='ncpdp'
                attribute='prescriber_qual'
                len='2'
                type='F'
                regEx='(^[ ]+$)|(^[0][0-9]$)|10|11|12|13|14|99'
                rejectCode='EZ'
        />
        <field id='DB'
```

```
                name='prescriber id'
                element='pPhrm'
                attribute='id_prov_prescrb'
                len='15'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='56'
        />
    </segment>
      <segment id='05'
             name='COB/other payments segment'
             required='O'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='4C'
                name='COB count'
                element='pPhrm'
                attribute='op_count'
                len='1'
                type='F'
                grouping='C'
                groupElement='phrmCobXref'
                regEx='^[1-9]{1}$'
                rejectCode='4C'
        >
            <field id='5C'
                   name='other payer type'
                   element='phrmCobXref'
                   attribute='cde_op_cov_type'
                   len='2'
                   type='F'
                   groupElement='phrmCobXref'
                   regEx='(^[ ]+$)|(^[0][123])|(^[9][89])'
                   rejectCode='5C'
            />
            <field id='E8'
                   name='other payer date'
                   element='phrmCobXref'
                   attribute='dte_other_payer'
                   len='8'
                   type='D'
                   regEx='^[0-9]{8}$'
                   rejectCode='E8'
            />
            <field id='HB'
                   name='other amount paid count'
                   element='phrmCobXref'
                   attribute='op_amt_paid_cnt'
                   len='1'
                   type='F'
                   grouping='C'
                   groupElement='phrmCob'
                   regEx='^[0-9]{1}$'
```

```
                        rejectCode='HB'
                >
                    <field id='HC'
                           name='other amount paid qualifier'
                           element='phrmCob'
                           attribute='cde_op_amt_paid_qual'
                           len='2'
                           type='F'
                           groupElement='phrmCob'
                           regEx='(^[ ]+$)|(^[0][1-8])|(^[9][89])'
                           rejectCode='HC'
                    />
                    <field id='DV'
                           name='other amount paid'
                           element='phrmCob'
                           attribute='amt_op_paid'
                           len='8'
                           dec='2'
                           type='A'
                           regEx='^[0-9]+[{}A-R]$|[{}A-R]'
                           rejectCode='DV'
                    />
                </field>
            </field>
          </segment>
        <segment id='08'
                 name='DUR/PPS segment'
                 required='O'
                 level='D'
                 format='V'
                 addSegId='Y'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='7E'
                   name='DUR counter'
                   element='pPhrm'
                   attribute='dur_cde_counter'
                   len='1'
                   type='F'
                   grouping='R'
                   groupElement='phrmDur'
                   regEx='^[1-9]{1}$'
                   rejectCode='7E'
            >
                        <field id='E4'
                   name='reason for service'
                   element='phrmDur'
                   attribute='reason_for_svc_cde'
                   len='2'
                   type='H'
                   hook='CopyStringHook'
                   element2='pPhrm'
                   attribute2='cde_ncpdp_conflict'
                   groupElement='phrmDur'

regEx='(^[A][DNRT]$)|(^[C][DHS]$)|(^[D][ACDFILMS]$)|(^[E][DRX]$)|HD|ID|LD|LK|
```

```
LR|MC|MN|MS|MX|(^[N][ACDFNPRS]$)|OH|(^[P][ACGHNPRS]$)|RE|RF|(^[S][CDEFRX]$)|(
^[T][DNP]$)'

                    rejectCode='E4'
                         />
                         <field id='E5'
                    name='professional service code'
                    element='phrmDur'
                    attribute='professional_svc_cde'
                    len='2'
                    type='H'
                    hook='CopyStringHook'
                    element2='pPhrm'
                    attribute2='cde_dur_intrvntn'
                    groupElement='phrmDur'

regEx='00|AS|CC|DE|FE|GP|MA|M0|MR|PE|PH|PM|P0|PT|R0|RT|SC|SW|TC|TH'
                    rejectCode='E5'
                         />
                         <field id='E6'
                    name='result of service code'
                    element='phrmDur'
                    attribute='result_of_svc_cde'
                    len='2'
                    type='H'
                    hook='CopyStringHook'
                    element2='pPhrm'
                    attribute2='cde_dur_outcome'
                    groupElement='phrmDur'
                    regEx='([00])|(^[1][A-K]$)|(^[2][AB]$)|(^[3][A-KMN]$)'
                    rejectCode='E6'
                         />
                   </field>
        </segment>
        <segment id='11'
                 name='pricing segment'
                 required='M'
                 level='D'
                 format='V'
                 addSegId='Y'
        >
            <status value='A'/>
            <status value='R'/>
             <field id='E3'
                    name='amount incentive'
                    element='pPhrm/phrmHdrOpt'
                    attribute='amt_incentive'
                    len='8'
                    dec='2'
                    type='A'
                    regEx='^[0-9]+[{}A-R]$'
                    rejectCode='E3'
            />
            <field id='DQ'
                    name='amount billed'
                    element='pPhrm'
                    attribute='amt_billed'
```

```
                len='8'
                dec='2'
                type='A'
                regEx='^[0-9]+[{}A-R]$'
                rejectCode='DQ'
        />
        <field id='DU'
                name='amount billed'
                element='pPhrm/phrmHdrOpt'
                attribute='amt_gross_due'
                len='8'
                dec='2'
                type='A'
                regEx='^[0-9]+[{}A-R]$'
                rejectCode='DU'
        />
    </segment>
    <segment id='10'
            name='compound segment'
            required='O'
            level='D'
            format='V'
            addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='EF'
                name='compound dosage code'
                element='ncpdp'
                attribute='cmpd_dose_code'
                len='2'
                type='F'
                regEx='(^[ ]+$)|(^[0][1-9])|(^[1][0-8])'
                rejectCode='EF'
        />
        <field id='EG'
                name='compound dispensing form'
                element='ncpdp'
                attribute='cmpd_disp_form'
                len='1'
                type='F'
                regEx='^[123]{1}$'
                rejectCode='EG'
        />
        <field id='EH'
                name='compound route of administration'
                element='pPhrm/phrmHdrOpt'
                attribute='cde_cmpd_rte_admin'
                len='2'
                type='F'
                regEx='(^[0-9]$)|(^[0][0-9]$)|(^[1][0-9]$)|(^[2][012]$)'
                rejectCode='EH'
        />
        <field id='EC'
                name='compound ingredient count'
                element='pPhrm'
                attribute='num_dtl_total'
```

```
                len='2'
                type='H'
                hook='IngredientCountHook'
                element2='ncpdp'
                attribute2='num_atomic_txns'
                grouping='C'
                groupElement='detail'
                regEx='(^[0-9]$)|(^[0][0-9]$)'
                rejectCode='EC'
        >
            <field id='RE'
                   name='compound id qualifier'
                   element='phrmDtlKey'
                   attribute='cmpd_id_qual'
                   len='2'
                   type='F'
                   groupElement='detail'
                   regEx='(^[ ]+$)|(^[0][0-5]$)|11|12|13|99'
                   rejectCode='RE'
            />
            <field id='TE'
                   name='compound id'
                   element='phrmDtlKey'
                   attribute='cde_ndc'
                   len='19'
                   type='S'
                   regEx='^[\x20-\x7E]+$'
                   rejectCode='TE'
            />
            <field id='ED'
                   name='compound ingredient quantity'
                   element='detail'
                   attribute='qty_dispense'
                   len='10'
                   dec='3'
                   type='N'
                   regEx='^[0-9]+$'
                   rejectCode='ED'
            />
            <field id='EE'
                   name='compound ingredient cost'
                   element='detail'
                   attribute='ingred_cost'
                   len='8'
                   dec='2'
                   type='A'
                   regEx='^[0-9]+[{}A-R]$'
                   rejectCode='EE'
            />
            <field id='UE'
                   name='compound ingredient cost basis'
                   element='detail'
                   attribute='ingred_cost_basis'
                   len='2'
                   type='F'
                   regEx='(^[ ]+$)|(^[0][0-9]$)|(^[0-9]$)'
                   rejectCode='UE'
```

```
            />
        </field>
    </segment>
    <segment id='13'
             name='clinical segment'
             required='O'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='VE'
               name='diagnosis code count'
               element='pPhrm'
               attribute='diag_count'
               len='1'
               type='F'
               grouping='C'
               groupElement='diag_xref'
               regEx='^[0-9]{1}$'
               rejectCode='VE'
        >
            <field id='WE'
                   name='diagnosis code qualifier'
                   element='diag_xref'
                   attribute='diag_qual'
                   len='2'
                   type='F'
                   groupElement='diag_xref'
                   regEx='(^[ ]+$)|(^[0][0-7]$)|99'
                   rejectCode='WE'
            />
            <field id='DO'
                   name='diagnosis code'
                   element='diag_xref'
                   attribute='cde_diag'
                   len='15'
                   type='S'
                   regEx='^[ .0-9]+$'
                   rejectCode='39'
            />
            <field id='XE'
                   name='clinical counter'
                   element='diag_xref'
                   attribute='clinical_counter'
                   len='1'
                   dec='0'
                   type='N'
                   regEx='^[0-9]{1}$'
                   rejectCode='XE'
            />
            <field id='ZE'
                   name='measure date'
                   element='diag_xref'
                   attribute='measure_date'
                   len='8'
```

```
                        type='D'
                        regEx='^[0-9]{8}$'
                        rejectCode='ZE'
                />
                <field id='H1'
                        name='measure time'
                        element='diag_xref'
                        attribute='measure_time'
                        len='4'
                        dec='0'
                        type='N'
                        regEx='^[0-9]+$'
                        rejectCode='H1'
                />
                <field id='H2'
                        name='measure dim'
                        element='diag_xref'
                        attribute='measure_dim'
                        len='2'
                        type='F'
                        regEx='(^[ ]+$)|(^[0][1-9])|(^[12][0-
9])|(^[3][01234])|(99)'
                        rejectCode='H2'
                />
                <field id='H3'
                        name='measure unit'
                        element='diag_xref'
                        attribute='measure_unit'
                        len='2'
                        type='F'
                        regEx='(^[ ]+$)|(^[ ][1-9]$)|(^[0][1-9])|(^[1][0-
9])|(^[2][0-7])'
                        rejectCode='H3'
                />
                <field id='H4'
                        name='measure value'
                        element='diag_xref'
                        attribute='measure_value'
                        len='15'
                        type='S'
                        regEx='^[\x20-\x7E]+$'
                        rejectCode='H4'
                />
            </field>
        </segment>
    </transaction>
    <reject_transaction id='99'
                 version='51'
                 xmlTemplate='invalidTxnList'
                 parentNode='invalidTxn'
                 errors='reject'
                 maxTxns='1'
                 region='0'
    >
        <segment id='00'
                 name='transaction header segment'
                 required='M'
```

```
                level='H'
                format='F'
                addSegId='N'
        >
            <status value='R'/>
            <field id='A1'
                   name='bin number'
                   element='ncpdp'
                   attribute='bin_number'
                   len='6'
                   dec='0'
                   type='S'
                   regEx='^[0-9]{6}$'
                   rejectCode='01'
            />
            <field id='A2'
                   name='version'
                   element='ncpdp'
                   attribute='version'
                   len='2'
                   dec='0'
                   type='S'
                   regEx='51'
                   rejectCode='03'
            />
            <field id='A3'
                   name='transaction code'
                   element='ncpdp'
                   attribute='txn_code'
                   len='2'
                   dec='0'
                   type='S'
                   regEx='B1|B2|B3|E1|P2|P3|P4'
                   rejectCode='03'
            />
            <field id='A4'
                   name='pcn'
                   element='ncpdp'
                   attribute='pcn'
                   len='10'
                   dec='0'
                   type='S'
                   regEx='^[\x20-\x7E]+$'
                   rejectCode='04'
            />
            <field id='A9'
                   name='transaction count'
                   element='ncpdp'
                   attribute='txn_count'
                   len='1'
                   dec='0'
                   type='N'
                   regEx='^[1-4]{1}$'
                   rejectCode='A9'
            />
            <field id='B2'
                   name='provider id qualifier'
```

```
                element='ncpdp'
                attribute='prov_id_qual'
                len='2'
                dec='0'
                type='S'
                regEx='05'
                rejectCode='B2'
        />
        <field id='B1'
                name='provider id'
                element='ncpdp'
                attribute='input_prov_id'
                len='15'
                dec='0'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='05'
        />
        <field id='D1'
                name='date of service'
                element='ncpdp'
                attribute='date_of_service'
                len='8'
                dec='0'
                type='D'
                regEx='^[0-9]{8}$'
                rejectCode='15'
        />
        <field id='AK'
                name='certification id'
                element='ncpdp'
                attribute='cert_id'
                len='10'
                dec='0'
                type='S'
                regEx='^[\x20-\x7E]+$'
                rejectCode='AK'
        />
    </segment>
  </reject_transaction>
</request>
```

File Name: ncpdpRequests.xml	
Attribute Name	**Description**
Id	Represents the transaction, segment or field.
xmlTemplate	Represents which xml template to use from the ncpdpTxnTemplates.xml file.
parentNode	Refers to the parent node in the xml.
errors	Defines the xml node where errors are collected
cde_clm_type	Defines the default code claim type in the xml.

File Name: ncpdpRequests.xml	
Attribute Name	**Description**
region	Defines the default region in the xml
element	Defines the default xml node at the segment level or the xml node to write the field attribute to.
name	Used only for readability, but defines either the segment or field name.
required	Used only for readability, but defines if the segment is mandatory or optional
level	Defines if the segment pertains to the header or detail portion of the transaction.
format	Defines if the segment is fixed or variable.
addSegId	Defines if the segment ID should be added.
value	Defines if the segment is used for a Accepted or Rejected value.
attribute	Defines the attribute name to write the field to.
len	Defines the maximum length for the field.
type	Defines the logic to map the field, F for Fixed, N for Numeric, S for String, H for Hook, D for Date, W for Web Date, etc.
hook	Defines unique processing that must be done for that field during translation.
regEx	Defines the regular expression that is used. This is applied against every field.
rejectCode	Defines the reject code if the value sumitted is in error or does not match the regular expression.
element2 or attribute2	Defines additional locations in the xml that the field is mapped to.

4.4.3 ncpdpResponses.xml config

This configuration file represents the NCPDP Batch 1.1 fields in the header, trailer and detail

```
<response>
    <transaction id='B1'
                 version='51'
                 parentNode='claim'
                 name='51 claim response'
    >
        <segment id='00'
```

```
                name='response header segment'
                required='M'
                level='H'
                format='F'
                addSegId='N'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='A2'
                   name='version'
                   element='ncpdp'
                   attribute='version'
                   len='2'
                   dec='0'
                   type='S'
            />
            <field id='A3'
                   name='transaction code'
                   element='ncpdp'
                   attribute='txn_code'
                   len='2'
                   dec='0'
                   type='S'
            />
            <field id='A9'
                   name='transaction count'
                   element='ncpdp'
                   attribute='txn_count'
                   len='1'
                   dec='0'
                   type='N'
            />
            <field id='F1'
                   name='header response status'
                   element='response/ncpdp'
                   attribute='response_status'
                   len='1'
                   dec='0'
                   type='S'
            />
            <field id='B2'
                   name='provider id qualifier'
                   element='ncpdp'
                   attribute='prov_id_qual'
                   len='2'
                   dec='0'
                   type='S'
            />
            <field id='B1'
                   name='provider id'
                   element='pPhrm/phrmHdrKey'
                   attribute='id_provider'
                   len='15'
                   dec='0'
                   type='S'
            />
            <field id='D1'
```

```
                name='date of service'
                element='*[@dte_dispense]'
                attribute='dte_dispense'
                len='8'
                dec='0'
                type='D'
        />
    </segment>
    <segment id='20'
             name='response message segment'
             required='O'
             level='H'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='D'/>
        <status value='R'/>
        <field id='F4'
               name='message'
               element='response/ncpdp'
               attribute='message'
               len='200'
               dec='0'
               type='S'
        />
    </segment>
    <segment id='21'
             name='response status segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='P'/>
        <status value='D'/>
        <field id='AN'
               name='transaction response status'
               element='response/ncpdp'
               attribute='txn_status'
               len='1'
               dec='0'
               type='S'
        />
        <field id='F3'
               name='authorization number'
               element='*[@num_icn]'
               attribute='num_icn'
               len='20'
               dec='0'
               type='S'
        />
        <field id='FQ'
               name='additional message information'
               element='response/ncpdp'
               attribute='additional_message'
               len='200'
```

```
                dec='0'
                type='S'
        />
    </segment>
    <segment id='21'
             name='response status segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='R'/>
        <field id='AN'
               name='transaction response status'
               element='response/ncpdp'
               attribute='txn_status'
               len='1'
               dec='0'
               type='S'
        />
        <field id='F3'
               name='authorization number'
               element='*[@num_icn]'
               attribute='num_icn'
               len='20'
               dec='0'
               type='S'
        />
        <field id='FA'
               name='reject count'
               element='response/ncpdp'
               attribute='reject_count'
               len='2'
               dec='0'
               type='N'
               grouping='C'
               groupElement='response/ncpdp/reject'
        >
            <field id='FB'
                   name='reject code'
                   element='reject'
                   attribute='code'
                   len='3'
                   dec='0'
                   type='S'
            />
        </field>
        <field id='FQ'
               name='additional message information'
               element='response/ncpdp'
               attribute='additional_message'
               len='200'
               dec='0'
               type='S'
        />
    </segment>
      <segment id='22'
```

```
                name='response claim segment'
                required='M'
                level='D'
                format='V'
                addSegId='Y'
        >
            <status value='P'/>
            <status value='D'/>
            <status value='R'/>
            <field id='EM'
                   name='rx number qualifier'
                   element='ncpdp'
                   attribute='rx_qual'
                   len='1'
                   dec='0'
                   type='S'
            />
            <field id='D2'
                   name='rx number'
                   element='*[@num_prscrip]'
                   attribute='num_prscrip'
                   len='7'
                   dec='0'
                   type='N'
            />
          </segment>
        <segment id='23'
                name='response pricing segment'
                required='M'
                level='D'
                format='V'
                addSegId='Y'
        >
            <status value='P'/>
            <field id='F5'
                   name='amount of copay/coinsurance'
                   element='response/header'
                   attribute='amt_co_pay'
                   len='8'
                   dec='2'
                   type='A'
            />
            <field id='F6'
                   name='ingredient cost paid'
                   element='response/header/detail'
                   attribute='amt_alwd'
                   len='8'
                   dec='2'
                   type='A'
            />
            <field id='F7'
                   name='dispensing fee paid'
                   element='response/header'
                   attribute='amt_ndc_profee'
                   len='8'
                   dec='2'
                   type='A'
```

```
        />
        <field id='FL'
               name='incentive amount paid'
               element='response/header'
               attribute='incentive_fee'
               len='8'
               dec='2'
               type='A'
        />
        <field id='F9'
               name='total amount paid'
               element='response/header'
               attribute='amt_paid'
               len='8'
               dec='2'
               type='A'
        />
        <field id='FI'
               name='amount of copay/coinsurance'
               element='response/header'
               attribute='amt_co_pay'
               len='8'
               dec='2'
               type='A'
        />
    </segment>
    <segment id='23'
             name='response pricing segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='D'/>
        <field id='F5'
             name='amount of copay/coinsurance'
             element='response/ncpdp'
             attribute='orig_amt_co_pay'
             len='8'
             dec='2'
             type='A'
        />
        <field id='F7'
             name='dispensing fee paid'
             element='response/ncpdp'
             attribute='orig_disp_fee'
             len='8'
             dec='2'
             type='A'
        />
        <field id='F9'
               name='total amount paid'
               element='response/ncpdp'
               attribute='orig_amt_paid'
               len='8'
               dec='2'
               type='A'
```

```
        />
        <field id='FI'
               name='amount of copay/coinsurance'
               element='response/ncpdp'
               attribute='orig_amt_co_pay'
               len='8'
               dec='2'
               type='A'
        />
    </segment>
    <segment id='24'
             name='response DUR/PPS segment'
             required='O'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='P'/>
        <status value='R'/>
        <field id='J6'
               name='DUR/PPS response code counter'
               element='response/ncpdp/dur_response'
               attribute='dur_code_counter'
               len='1'
               dec='0'
               type='N'
               grouping='R'
               groupElement='response/ncpdp/dur_response'
        >
            <field id='E4'
                   name='reason for service'
                   element='dur_response'
                   attribute='cde_ncpdp_conflict'
                   len='2'
                   dec='0'
                   type='S'
            />
            <field id='FS'
                   name='clinical significance code'
                   element='dur_response'
                   attribute='severity_level'
                   len='1'
                   dec='0'
                   type='S'
            />
            <field id='FT'
                   name='other pharmacy indicator'
                   element='dur_response'
                   attribute='other_pharm_ind'
                   len='1'
                   dec='0'
                   type='N'
            />
            <field id='FU'
                   name='previous date of fill'
                   element='dur_response'
                   attribute='prev_date_fill'
```

```
                        len='8'
                        dec='0'
                        type='D'
                />
                <field id='FV'
                        name='quantity of previous fill'
                        element='dur_response'
                        attribute='prev_qty_fill'
                        len='10'
                        dec='3'
                        type='N'
                />
                <field id='FW'
                        name='database indicator'
                        element='dur_response'
                        attribute='db_ind'
                        len='1'
                        dec='0'
                        type='S'
                />
                <field id='FX'
                        name='other prescriber indicator'
                        element='dur_response'
                        attribute='other_prescrb_ind'
                        len='1'
                        dec='0'
                        type='N'
                />
                <field id='FY'
                        name='DUR free text message'
                        element='dur_response'
                        attribute='dur_free_text'
                        len='30'
                        dec='0'
                        type='S'
                />
            </field>
        </segment>
    </transaction>
    <transaction id='B2'
                 version='51'
                 parentNode='claim'
                 name='51 reversal response'
    >
        <segment id='00'
                 name='response header segment'
                 required='M'
                 level='H'
                 format='F'
                 addSegId='N'
        >
            <status value='A'/>
            <status value='R'/>
            <field id='A2'
                   name='version'
                   element='ncpdp'
                   attribute='version'
```

```
                len='2'
                dec='0'
                type='S'
        />
        <field id='A3'
                name='transaction code'
                element='ncpdp'
                attribute='txn_code'
                len='2'
                dec='0'
                type='S'
        />
        <field id='A9'
                name='transaction count'
                element='ncpdp'
                attribute='txn_count'
                len='1'
                dec='0'
                type='N'
        />
        <field id='F1'
                name='header response status'
                element='response/ncpdp'
                attribute='response_status'
                len='1'
                dec='0'
                type='S'
        />
        <field id='B2'
                name='provider id qualifier'
                element='ncpdp'
                attribute='prov_id_qual'
                len='2'
                dec='0'
                type='S'
        />
        <field id='B1'
                name='provider id'
                element='pPhrm/phrmHdrKey'
                attribute='id_provider'
                len='15'
                dec='0'
                type='S'
        />
        <field id='D1'
                name='date of service'
                element='*[@dte_dispense]'
                attribute='dte_dispense'
                len='8'
                dec='0'
                type='D'
        />
    </segment>
    <segment id='20'
             name='response message segment'
             required='O'
             level='H'
```

```
                format='V'
                addSegId='Y'
        >
            <status value='A'/>
            <status value='S'/>
            <status value='R'/>
            <field id='F4'
                   name='message'
                   element='response/ncpdp'
                   attribute='message'
                   len='200'
                   dec='0'
                   type='S'
            />
        </segment>
        <segment id='21'
                 name='response status segment'
                 required='M'
                 level='D'
                 format='V'
                 addSegId='Y'
        >
            <status value='A'/>
            <status value='S'/>
            <field id='AN'
                   name='transaction response status'
                   element='response/ncpdp'
                   attribute='txn_status'
                   len='1'
                   dec='0'
                   type='S'
            />
            <field id='F3'
                   name='authorization number'
                   element='*[@num_icn]'
                   attribute='num_icn'
                   len='20'
                   dec='0'
                   type='S'
            />
            <field id='FQ'
                   name='additional message information'
                   element='response/ncpdp'
                   attribute='additional_message'
                   len='200'
                   dec='0'
                   type='S'
            />
        </segment>
        <segment id='21'
                 name='response status segment'
                 required='M'
                 level='D'
                 format='V'
                 addSegId='Y'
        >
            <status value='R'/>
```

```
        <field id='AN'
               name='transaction response status'
               element='response/ncpdp'
               attribute='txn_status'
               len='1'
               dec='0'
               type='S'
        />
        <field id='F3'
               name='authorization number'
               element='*[@num_icn]'
               attribute='num_icn'
               len='20'
               dec='0'
               type='S'
        />
        <field id='FA'
               name='reject count'
               element='response/ncpdp'
               attribute='reject_count'
               len='2'
               dec='0'
               type='N'
               grouping='C'
               groupElement='response/ncpdp/reject'
        >
            <field id='FB'
                   name='reject code'
                   element='reject'
                   attribute='code'
                   len='3'
                   dec='0'
                   type='S'
            />
        </field>
        <field id='FQ'
               name='additional message information'
               element='response/ncpdp'
               attribute='additional_message'
               len='200'
               dec='0'
               type='S'
        />
    </segment>
      <segment id='22'
             name='response claim segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='D'/>
        <status value='R'/>
        <field id='EM'
               name='rx number qualifier'
               element='ncpdp'
```

```
                attribute='rx_qual'
                len='1'
                dec='0'
                type='S'
        />
        <field id='D2'
                name='rx number'
                element='*[@num_prscrip]'
                attribute='num_prscrip'
                len='7'
                dec='0'
                type='N'
        />
    </segment>
</transaction>
<transaction id='B3'
             version='51'
             parentNode='claim'
             name='51 claim response'
>
    <segment id='00'
             name='response header segment'
             required='M'
             level='H'
             format='F'
             addSegId='N'
    >
        <status value='A'/>
        <status value='R'/>
        <field id='A2'
                name='version'
                element='ncpdp'
                attribute='version'
                len='2'
                dec='0'
                type='S'
        />
        <field id='A3'
                name='transaction code'
                element='ncpdp'
                attribute='txn_code'
                len='2'
                dec='0'
                type='S'
        />
        <field id='A9'
                name='transaction count'
                element='ncpdp'
                attribute='txn_count'
                len='1'
                dec='0'
                type='N'
        />
        <field id='F1'
                name='header response status'
                element='response/ncpdp'
                attribute='response_status'
```

```
                len='1'
                dec='0'
                type='S'
        />
        <field id='B2'
                name='provider id qualifier'
                element='ncpdp'
                attribute='prov_id_qual'
                len='2'
                dec='0'
                type='S'
        />
        <field id='B1'
                name='provider id'
                element='pPhrm/phrmHdrKey'
                attribute='id_provider'
                len='15'
                dec='0'
                type='S'
        />
        <field id='D1'
                name='date of service'
                element='*[@dte_dispense]'
                attribute='dte_dispense'
                len='8'
                dec='0'
                type='D'
        />
    </segment>
    <segment id='20'
             name='response message segment'
             required='O'
             level='H'
             format='V'
             addSegId='Y'
    >
        <status value='A'/>
        <status value='D'/>
        <status value='R'/>
        <field id='F4'
                name='message'
                element='response/ncpdp'
                attribute='message'
                len='200'
                dec='0'
                type='S'
        />
    </segment>
    <segment id='21'
             name='response status segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='P'/>
        <status value='D'/>
```

```
        <field id='AN'
               name='transaction response status'
               element='response/ncpdp'
               attribute='txn_status'
               len='1'
               dec='0'
               type='S'
        />
        <field id='F3'
               name='authorization number'
               element='*[@num_icn]'
               attribute='num_icn'
               len='20'
               dec='0'
               type='S'
        />
        <field id='FQ'
               name='additional message information'
               element='response/ncpdp'
               attribute='additional_message'
               len='200'
               dec='0'
               type='S'
        />
    </segment>
    <segment id='21'
             name='response status segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='R'/>
        <field id='AN'
               name='transaction response status'
               element='response/ncpdp'
               attribute='txn_status'
               len='1'
               dec='0'
               type='S'
        />
        <field id='F3'
               name='authorization number'
               element='*[@num_icn]'
               attribute='num_icn'
               len='20'
               dec='0'
               type='S'
        />
        <field id='FA'
               name='reject count'
               element='response/ncpdp'
               attribute='reject_count'
               len='2'
               dec='0'
               type='N'
               grouping='C'
```

```
                groupElement='response/ncpdp/reject'
        >
            <field id='FB'
                   name='reject code'
                   element='reject'
                   attribute='code'
                   len='3'
                   dec='0'
                   type='S'
            />
        </field>
        <field id='FQ'
               name='additional message information'
               element='response/ncpdp'
               attribute='additional_message'
               len='200'
               dec='0'
               type='S'
        />
    </segment>
      <segment id='22'
             name='response claim segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='P'/>
        <status value='D'/>
        <status value='R'/>
        <field id='EM'
               name='rx number qualifier'
               element='ncpdp'
               attribute='rx_qual'
               len='1'
               dec='0'
               type='S'
        />
        <field id='D2'
               name='rx number'
               element='*[@num_prscrip]'
               attribute='num_prscrip'
               len='7'
               dec='0'
               type='N'
        />
      </segment>
    <segment id='23'
             name='response pricing segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='P'/>
        <field id='F5'
               name='amount of copay/coinsurance'
```

```
                element='response/header'
                attribute='amt_co_pay'
                len='8'
                dec='2'
                type='A'
        />
        <field id='F6'
                name='ingredient cost paid'
                element='response/header/detail'
                attribute='amt_alwd'
                len='8'
                dec='2'
                type='A'
        />
        <field id='F7'
                name='dispensing fee paid'
                element='response/header'
                attribute='amt_ndc_profee'
                len='8'
                dec='2'
                type='A'
        />
        <field id='FL'
                name='incentive amount paid'
                element='response/header'
                attribute='incentive_fee'
                len='8'
                dec='2'
                type='A'
        />
        <field id='F9'
                name='total amount paid'
                element='response/header'
                attribute='amt_paid'
                len='8'
                dec='2'
                type='A'
        />
        <field id='FI'
                name='amount of copay/coinsurance'
                element='response/header'
                attribute='amt_co_pay'
                len='8'
                dec='2'
                type='A'
        />
    </segment>
    <segment id='23'
             name='response pricing segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='D'/>
        <field id='F5'
             name='amount of copay/coinsurance'
```

```
                element='response/ncpdp'
                attribute='orig_amt_co_pay'
                len='8'
                dec='2'
                type='A'
            />
            <field id='F7'
                name='dispensing fee paid'
                element='response/ncpdp'
                attribute='orig_disp_fee'
                len='8'
                dec='2'
                type='A'
            />
            <field id='F9'
                   name='total amount paid'
                   element='response/ncpdp'
                   attribute='orig_amt_paid'
                   len='8'
                   dec='2'
                   type='A'
            />
            <field id='FI'
                   name='amount of copay/coinsurance'
                   element='response/ncpdp'
                   attribute='orig_amt_co_pay'
                   len='8'
                   dec='2'
                   type='A'
            />
        </segment>
        <segment id='24'
                name='response DUR/PPS segment'
                required='O'
                level='D'
                format='V'
                addSegId='Y'
        >
            <status value='P'/>
            <status value='R'/>
            <field id='J6'
                   name='DUR/PPS response code counter'
                   element='response/ncpdp/dur_response'
                   attribute='dur_code_counter'
                   len='1'
                   dec='0'
                   type='N'
                   grouping='R'
                   groupElement='response/ncpdp/dur_response'
            >
                <field id='E4'
                       name='reason for service'
                       element='dur_response'
                       attribute='cde_ncpdp_conflict'
                       len='2'
                       dec='0'
                       type='S'
```

```
/>
<field id='FS'
       name='clinical significance code'
       element='dur_response'
       attribute='severity_level'
       len='1'
       dec='0'
       type='S'
/>
<field id='FT'
       name='other pharmacy indicator'
       element='dur_response'
       attribute='other_pharm_ind'
       len='1'
       dec='0'
       type='N'
/>
<field id='FU'
       name='previous date of fill'
       element='dur_response'
       attribute='prev_date_fill'
       len='8'
       dec='0'
       type='D'
/>
<field id='FV'
       name='quantity of previous fill'
       element='dur_response'
       attribute='prev_qty_fill'
       len='10'
       dec='3'
       type='N'
/>
<field id='FW'
       name='database indicator'
       element='dur_response'
       attribute='db_ind'
       len='1'
       dec='0'
       type='S'
/>
<field id='FX'
       name='other prescriber indicator'
       element='dur_response'
       attribute='other_prescrb_ind'
       len='1'
       dec='0'
       type='N'
/>
<field id='FY'
       name='DUR free text message'
       element='dur_response'
       attribute='dur_free_text'
       len='30'
       dec='0'
       type='S'
/>
```

```
            </field>
        </segment>
    </transaction>
    <reject_transaction id='99'
                 version='51'
                 parentNode='invalidTxn'
                 name='reject response'
    >
        <segment id='00'
                 name='response header segment'
                 required='M'
                 level='H'
                 format='F'
                 addSegId='N'
        >
            <status value='R'/>
            <field id='A2'
                   name='version'
                   element='ncpdp'
                   attribute='version'
                   len='2'
                   dec='0'
                   type='S'
            />
            <field id='A3'
                   name='transaction code'
                   element='ncpdp'
                   attribute='txn_code'
                   len='2'
                   dec='0'
                   type='S'
            />
            <field id='A9'
                   name='transaction count'
                   element='ncpdp'
                   attribute='txn_count'
                   len='1'
                   dec='0'
                   type='N'
            />
            <field id='F1'
                   name='header response status'
                   element='response/ncpdp'
                   attribute='response_status'
                   len='1'
                   dec='0'
                   type='S'
            />
            <field id='B2'
                   name='provider id qualifier'
                   element='ncpdp'
                   attribute='prov_id_qual'
                   len='2'
                   dec='0'
                   type='S'
            />
            <field id='B1'
```

```
                name='provider id'
                element='ncpdp'
                attribute='input_prov_id'
                len='15'
                dec='0'
                type='S'
        />
        <field id='D1'
                name='date of service'
                element='ncpdp'
                attribute='date_of_service'
                len='8'
                dec='0'
                type='D'
        />
    </segment>
    <segment id='20'
             name='response message segment'
             required='O'
             level='H'
             format='V'
             addSegId='Y'
    >
        <status value='R'/>
        <field id='F4'
                name='message'
                element='ncpdp'
                attribute='message'
                len='200'
                dec='0'
                type='S'
        />
    </segment>
    <segment id='21'
             name='response status segment'
             required='M'
             level='D'
             format='V'
             addSegId='Y'
    >
        <status value='R'/>
        <field id='AN'
                name='transaction response status'
                element='response/ncpdp'
                attribute='txn_status'
                len='1'
                dec='0'
                type='S'
        />
        <field id='FA'
                name='reject count'
                element='response/ncpdp'
                attribute='reject_count'
                len='2'
                dec='0'
                type='N'
                grouping='C'
```

```
                    groupElement='response/ncpdp/reject'
            >
                <field id='FB'
                       name='reject code'
                       element='reject'
                       attribute='code'
                       len='3'
                       dec='0'
                       type='S'
                />
            </field>
            <field id='FQ'
                   name='additional message information'
                   element='response/ncpdp'
                   attribute='additional_message'
                   len='200'
                   dec='0'
                   type='S'
            />
        </segment>
    </reject_transaction>
</response>
```

File Name: ncpdpResponses.xml	
Attribute Name	**Description**
id	The transaction, segment or field id depending upon location.
version	Text information representing which version of NCPDP this is for.
parentNode	Defines the parent xml node
name	Text information further describing a segment or field
required	Text information describing a segments usage
level	Text information describing the location at the header or detail level.
format	Text information describing if the segment is fixed or variable.
addSegId	Defines if the segment AM value should be added.
value	Defines upon the txn_status value which fields should be sent back.
element	Defines where in the XML to look for an attribute
attribute	Defines the name of the attribute in the XML
len	Defines the length of the field

File Name: ncpdpResponses.xml	
Attribute Name	**Description**
dec	Defines if there is a decimal point
type	Defines if the value is a string, numeric, date, etc.

4.4.4 ncpdpTxnTemplates.xml config

This configuration file represents the NCPDP Telecommunciation template in which to build the XML.

```
<templateList>
    <claimList commit='false' log='false'>
        <claim>
            <pPhrm>
                <detail>
                    <phrmDtlKey/>
                </detail>
                <phrmHdrKey/>
                <phrmHdrOpt/>
            </pPhrm>
            <ncpdp>
            </ncpdp>
        </claim>
    </claimList>
    <priorAuthorizations commit='false' log='false'>
        <priorAuthorization>
            <paLineItem>
            </paLineItem>
            <ncpdp>
                        <unmappedFields/>
            </ncpdp>
        </priorAuthorization>
    </priorAuthorizations>
    <paPharmRevs commit='false' log='false'>
        <paPharmRev>
            <ncpdp>
                        <unmappedFields/>
            </ncpdp>
        </paPharmRev>
    </paPharmRevs>
  <paPharmInqs commit='false' log='false'>
        <paPharmInq>
            <ncpdp>
                        <unmappedFields/>
            </ncpdp>
        </paPharmInq>
    </paPharmInqs>
    <eligVerification commit='true' log='true'>
        <eligRequest>
            <requestLoop>
                <eligInfoReceiver/>
                <eligSubscriber>
                        <eligSubscriberDate/>
                </eligSubscriber>
```

```
            </requestLoop>
            <ncpdp>
                        <unmappedFields/>
            </ncpdp>
        </eligRequest>
    </eligVerification>
    <invalidTxnList commit='false' log='false'>
        <invalidTxn>
            <ncpdp>
                        <unmappedFields/>
            </ncpdp>
        </invalidTxn>
    </invalidTxnList>
</templateList>
```

File Name: ncpdpTxnTemplates.xml	
Attribute Name	**Description**
Commit	Commit the transaction. This servers only as a place holder.
Log	Log the transaction. This servers only as a place holder.

4.4.5 NCPDPBatchConfiguration.xml config

The NCPDP Batch Configuration contains information used by the batch process and translator as well as other applications.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
      <ServiceActivations>
            <ServiceActivation name="NCPDPBatchReader" />
            <ServiceActivation name="NCPDPFileBuilder" />
            <ServiceActivation name="NCPDPTelecommunicationTranslator" />
            <ServiceActivation name="NCPDPFiles" />
      </ServiceActivations>
      <NCPDPFileBuilder name="NCPDPFile"
responseQueueWatcherName="NCPDPResponseBatchHandlerReader"
outputFolder="D:\TestFiles\BatchRequestOutput\"
translatorName="NCPDPTelecommunicationTranslator" />
      <NCPDPBatchReader name="NCPDPBatch" workerThreads="5"
requestQueueWatcherName="NCPDPRequestBatchHandler"
responseQueueWriterName="NCPDPResponseBatchHandler"
threadWorkerName="NCPDPSubmitterThreadWorker"
availableThreadSleepInterval="500" />
      <NCPDPTelecommunicationTranslator
name="NCPDPTelecommunicationTranslator" timerDueTime="10000"
timerPeriod="60000" batchXmlConfigPath="D:\Builds\NCPDP\ncpdpBatch.xml"
ncpdpXmlRequestConfigPath="D:\Builds\NCPDP\ncpdpRequests.xml"
ncpdpXmlResponseConfigPath="D:\Builds\NCPDP\ncpdpResponses.xml"
ncpdpXmlTxnTemplatesConfigPath="D:\Builds\NCPDP\ncpdpTxnTemplates.xml"
ncpdpBatchDualQWriterName="NCPDPBatchRequestHandler" >
            <TransactionDefinition supportedTransactions="B1|B2|B3"
b1Segments="MOOOMOOMOOOMOO" b2Segments="MONNONNMONNONN"
b3Segments="MOOOMOOMOOOMOO" e1Segments="MOONMNNNNNNNNN"
```

```
p2Segments="MOOOONOMOOOOON" p3Segments="MOOOMOOMOOOOMN"
p4Segments="MOOOMOOMOOOOMO" />
            <SegmentDefintion>
                  <Segment name = "01" def="Patient"  mandatoryFields="AM"
optionalFields="CX|CY|C4|C5|CA|CB|CM|CN|CO|CP|CQ|C7|CZ|1C|2C" />
                  <Segment name = "02" def="Provider" mandatoryFields="AM"
optionalFields="EY|E9" />
                  <Segment name = "03" def="Prescriber" mandatoryFields="AM"
optionalFields="EZ|DB|1E|DR|PM|2E|DL|H5|4E" />
                  <Segment name = "04" def="Insurance"
mandatoryFields="AM|C2" optionalFields="CC|CD|CE|FO|C9|8C|C1|C3|C6" />
                  <Segment name = "05" def="COB" mandatoryFields="AM|4C|5C"
optionalFields="6C|7C|E8|HB|HC|DV|5E|6E" />
                  <Segment name = "06" def="WorkersComp"
mandatoryFields="AM|DY" optionalFields="CF|CG|CH|CI|CJ|CK|CL|CR|DZ" />
                  <Segment name = "07" def="Claim"
mandatoryFields="AM|EM|D2|E1|D7"
optionalFields="EN|EP|SE|ER|E7|D3|D5|D6|D8|DE|DF|DJ|DK|ET|C8|DT|EJ|EA|EB|CW|E
K|28|DI|EU|EV|EW|EX|HD|HF|HG" />
                  <Segment name = "08" def="DUR" mandatoryFields="AM"
optionalFields="7E|E4|E5|E6|8E|J9|H6" />
                  <Segment name = "09" def="Coupon"
mandatoryFields="AM|KE|ME" optionalFields="NE" />
                  <Segment name = "10" def="Compound"
mandatoryFields="AM|EF|EG|EH|EC|RE|TE|ED" optionalFields="EE|UE" />
                  <Segment name = "11" def="Pricing" mandatoryFields="AM"
optionalFields="D9|DC|BE|DX|E3|H7|H8|H9|HA|GE|HE|JE|DQ|DU|DN" />
                  <Segment name = "12" def="PriorAuth"
mandatoryFields="AM|PA|PB|PC|PD" optionalFields="PE|PF|PG|PH|PJ|PK|PY|F3|PP"
/>
                  <Segment name = "13" def="Clinical" mandatoryFields="AM"
optionalFields="VE|WE|DO|XE|ZE|H1|H2|H3|H4" />
            </SegmentDefintion>
      </NCPDPTelecommunicationTranslator>
      <NCPDPFiles>
            <NCPDPFile name="Passport">
                  <RegionCode name="71" />
                  <EncounterIndicator name="Y" />
                  <NCPDPInput>
                        <Directory
name="\\10.40.13.12\fts\Passport\NCPDP\Inbound\InBound\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\Inbound\InProgress\" />
                        <BackupDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\Inbound\Backup\" />
                        <ErrorDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\Inbound\Error\" />
                        <FileExtension name="*.*" />
                  </NCPDPInput>
                  <XMLInput>
                        <Directory
name="\\10.40.13.12\fts\Passport\NCPDP\InBoundXml\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\InBoundXml\InProgress\" />
                        <BackupDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\InBoundXml\Backup\" />
```

```
                        <ErrorDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\InBoundXml\Error\" />
                        <FileExtension name="*.xml" />
                  </XMLInput>
                  <XMLOutput>
                        <Directory
name="\\10.40.13.12\fts\Passport\NCPDP\OutBoundXml\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\OutBoundXml\InProgress\" />
                        <BackupDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\OutBoundXml\Backup\" />
                        <ErrorDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\OutBoundXml\Error\" />
                        <FileExtension name="*.xml" />
                  </XMLOutput>
                  <NCPDPOutput>
                        <Directory
name="\\10.40.13.12\fts\Passport\NCPDP\Outbound\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\Passport\NCPDP\Outbound\InProgress\" />
                        <FileExtension name="*.*" />
                  </NCPDPOutput>
            </NCPDPFile>
            <NCPDPFile name="FirstHealth">
                  <RegionCode name="25" />
                  <EncounterIndicator name="H" />
                  <NCPDPInput>
                        <Directory
name="\\10.40.13.12\fts\External\FH_NCPDP\InBound\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\InBound\InProgress\" />
                        <BackupDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\InBound\Backup\" />
                        <ErrorDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\InPound\Error\" />
                        <FileExtension name="*.*" />
                  </NCPDPInput>
                  <XMLInput>
                        <Directory
name="\\10.40.13.12\fts\External\FH_NCPDP\InBoundXml\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\InBoundXml\InProgress\" />
                        <BackupDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\InBoundXml\Backup\" />
                        <ErrorDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\InBoundXml\Error\" />
                        <FileExtension name="*.xml" />
                  </XMLInput>
                  <XMLOutput>
                        <Directory
name="\\10.40.13.12\fts\External\FH_NCPDP\OutBoundXml\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\OutBoundXml\InProgress\" />
                        <BackupDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\OutBoundXml\Backup\" />
                        <ErrorDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\OutBoundXml\Error\" />
```

```
                        <FileExtension name="*.xml" />
                  </XMLOutput>
                  <NCPDPOutput>
                        <Directory
name="\\10.40.13.12\fts\External\FH_NCPDP\OutBound\" />
                        <InProgressDirectory
name="\\10.40.13.12\fts\External\FH_NCPDP\OutBound\InProgress\" />
                        <FileExtension name="*.*" />
                  </NCPDPOutput>
            </NCPDPFile>
      </NCPDPFiles>
      <NCPDPBatchRequest name="NCPDPBatchRequest"
regExHeader="(\u0002)(00)(T|R|E)([ A-Z0-9]{24})([0-9]{7})([0-9]{8})([0-
9]{4})(P|T)([ A-Z0-9]{2})([ A-Z0-9]{24})(\u0003)" regExDetail="(\u0002)(G1)([
A-Z0-9]{10})" regExTrailer="(\u0002)(99)([0-9]{7})([ 0-9]{10})([ A-Z0-
9]{35})(\u0003)" translatorName="NCPDPTelecommunicationTranslator" />
      <NCPDPSubmitterThreadWorker name="NCPDPSubmitterThreadWorker"
iPAddress="10.40.13.140" port="20010" sendTimeout="50000"
receiveTimeout="50000" />
      <BatchHandlers>
            <BatchHandler name="NCPDPBatchRequestHandler">
                  <Assembly
name="D:\Builds\NCPDP\EDS.USGS.MED.EDI.NCPDP.Translator.NcpdpInputBatchReques
tProcessor.dll" />
                  <Class
name="EDS.USGS.MED.EDI.NCPDP.Translator.NcpdpBatchDualQueueWriter" />
                  <Method name="Execute" methodType="MethodBased" />
                  <InitString commandChannel="NCPDPBatchCommandQueueWriter"
dataChannel="NCPDPBatchDataQueueWriter" maxBatchSize="1000"
batchIDType="FileTrackingID" />
            </BatchHandler>
            <BatchHandler name="NCPDPRequestBatchHandler">
                  <Assembly
name="D:\Builds\NCPDP\EDS.USGS.MED.EDI.Common.BatchProcessor.BatchHandlers.dl
l" />
                  <Class
name="EDS.USGS.MED.EDI.Common.BatchProcessor.BatchHandlers.BatchHandlerDualQR
eader" />
                  <Method name="Execute"  methodType="MethodBased" />
                  <InitString commandChannel="NCPDPRequestCommandQueue"
dataChannel="NCPDPRequestDataQueue" maxWorkerThreads="3" throttleInterval="0"
/>
            </BatchHandler>
            <BatchHandler name="NCPDPResponseBatchHandler">
                  <Assembly
name="D:\Builds\NCPDP\EDS.USGS.MED.EDI.Common.BatchProcessor.BatchHandlers.dl
l" />
                  <Class
name="EDS.USGS.MED.EDI.Common.BatchProcessor.BatchHandlers.BatchHandlerDualQW
riter" />
                  <Method name="Execute"  methodType="MethodBased" />
                  <InitString commandChannel="NCPDPResponseCommandQueue"
dataChannel="NCPDPResponseDataQueue" maxWorkerThreads="3"
throttleInterval="0" />
            </BatchHandler>
            <BatchHandler name="NCPDPResponseBatchHandlerReader">
```

```
                  <Assembly
name="D:\Builds\NCPDP\EDS.USGS.MED.EDI.Common.BatchProcessor.BatchHandlers.dl
l" />
                  <Class
name="EDS.USGS.MED.EDI.Common.BatchProcessor.BatchHandlers.BatchHandlerDualQR
eader" />
                  <Method name="Execute"  methodType="MethodBased" />
                  <InitString commandChannel="NCPDPResponseCommandQueue"
dataChannel="NCPDPResponseDataQueue" />
            </BatchHandler>
      </BatchHandlers>
      <Watchers>
            <Watcher name="NCPDPRequestCommandQueue" type="MSMQWatcherType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.MSMQWatcher" />
                  <TriggeringInterval value="3000" />
                  <TriggeringType value = "PollingBased" />
                  <WatcherLocation value="FormatName:PUBLIC=755C36FE-52DC-
417B-8373-A2D0ED_7106F" />
            </Watcher>
            <Watcher name="NCPDPRequestDataQueue" type="MSMQWatcherType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.MSMQWatcher" />
                  <TriggeringInterval value="3000" />
                  <TriggeringType value="None" />
                  <WatcherLocation value="FormatName:PUBLIC=B6476FF0-8881-
481E-A05C-29CEF9C390D9" />
            </Watcher>
            <Watcher name="NCPDPResponseCommandQueue" type="MSMQWatcherType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.MSMQWatcher" />
                  <TriggeringInterval value="3000" />
                  <TriggeringType value="PollingBased" />
                  <WatcherLocation value="FormatName:PUBLIC=711F7E0B-B098-
473B-98F1-8F9236D675F0" />
            </Watcher>
            <Watcher name="NCPDPResponseDataQueue" type="MSMQWatcherType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.MSMQWatcher" />
                  <TriggeringInterval value="3000" />
                  <TriggeringType value="None" />
                  <WatcherLocation value="FormatName:PUBLIC=90219C2B-8C68-
4289-BB96-269716_66C1D" />
            </Watcher>
      </Watchers>
      <Writers>
            <Writer name="NCPDPBatchCommandQueueWriter" type="MSMQWriter">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Writers.MSMQWriter" />
                  <WriteLocation value="FormatName:PUBLIC=755C36FE-52DC-417B-
8373-A2D0ED_7106F" />
                  <WriteAsync value="false" />
            </Writer>
            <Writer name="NCPDPBatchDataQueueWriter" type="MSMQWriter">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Writers.MSMQWriter" />
```

```
                  <WriteLocation value="FormatName:PUBLIC=B6476FF0-8881-481E-
A05C-29CEF9C390D9" />
                  <WriteAsync value="false" />
            </Writer>
            <Writer name="NCPDPResponseCommandQueue"
type="NCPDPResponseCommandQueueWriterType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Writers.MSMQWriter" />
                  <WriteLocation value="FormatName:PUBLIC=711F7E0B-B098-473B-
98F1-8F9236D675F0" />
                  <WriteAsync value="false" />
            </Writer>
            <Writer name="NCPDPResponseDataQueue"
type="NCPDPResponseDataQueueWriterType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Writers.MSMQWriter" />
                  <WriteLocation value="FormatName:PUBLIC=90219C2B-8C68-4289-
BB96-269716_66C1D" />
                  <WriteAsync value="false" />
            </Writer>
      </Writers>
      <WatcherTypes>
            <WatcherType name="MSMQWatcherType">
                  <RegexFilter value="^([A-Za-z0-9.])*$" enabled="true" />
                  <SharedModeDenyReceive value="false" />
                  <ReadTimeout value="30000" />
                  <RecycleInterval value="300000" />
            </WatcherType>
      </WatcherTypes>
      <WriterTypes>
            <WriterType name="NCPDPResponseDataQueueWriterType">
                  <Recoverable value="false" />
            </WriterType>
            <WriterType name="NCPDPResponseCommandQueueWriterType">
                  <Recoverable value="false" />
            </WriterType>
                        <WriterType name="MSMQWriter">
                  <Recoverable value="true" />
            </WriterType>
      </WriterTypes>
</configuration>
```

File Name: NCPDPBatchConfiguration.xml	
Attribute Name	**Description**
ServiceActivation	
name	Defines the services that use this configuration.
NCDPFileBuilder	
name	Refers to the element identifier.
responseQueueWatcherName	Defines the request queue name.

File Name: NCPDPBatchConfiguration.xml	
Attribute Name	**Description**
translatorName	Defines the name of the translator.
NCPDPBatchReader	
name	Refers to the element identifier.
workerThreads	Defines the total number of threads open to process transactions.
requestQueueWatcherName	Defines the queue name to look for requests.
responseQueueWriterName	Defines the queue name to look for responses.
threadWorkerName	Defines which thread name to use.
availableThreadSleepInterval	Defines the sleep interval for each thread.
NCPDPTelecommunicationTranslato r	
name	Refers to the element identifier.
timerDueTime	Defines the timer due time to start.
timerPeriod	Defines the timer duration.
batchXmlConfigPath	Defines the location of the Batch XML Configuration Path which contains the batch mappings.
ncpdpXmlRequestConfigPath	Defines the location of the Batch XML Request Path which contains the request mappings.
ncpdpXmlResponeConfigPath	Defines the location of the Batch XML Response Path which contains the response mappings.
ncpdpXmlTxnTemplatesConfigPath	Defines the location of the Batch XML Transaction Templates file which is used in the mapping.
ncpdpBatchDualQueueWriterName	Defines which Batch Dual Queue Writer to use.
TransactionDefinition	
supportedTransactions	Defines which NCPDP transactions to allow. In this case, allow for the B1 billing, B2 reversal and B3 rebill.
b1Segments	Defines which segments are mandatory or optional for this transaction type.
b2Segments	Defines which segments are mandatory or optional for this transaction type.
b3Segments	Defines which segments are mandatory or optional for this

File Name: NCPDPBatchConfiguration.xml	
Attribute Name	**Description**
	transaction type.
e1Segments	Defines which segments are mandatory or optional for this transaction type.
p2Segments	Defines which segments are mandatory or optional for this transaction type.
p3Segments	Defines which segments are mandatory or optional for this transaction type.
p4Segments	Defines which segments are mandatory or optional for this transaction type.
SegmentDefinition	
name	Refers to the element identifier.
def	Text to describe the segment. This is only used for readability.
mandatoryFields	Defines the mandatory fields in the segment.
optionalFields	Defines the optional fields in the segment.
NCPDPFile name	Refers to the senders name. Since each sender may have specific attributes to map, this allows us to define those changes here.
RegionCode name	Defines the region code value to appear on the xml.
EncounterIndicator name	Defines the encounter indicator value to appear on the xml.
Directory names	Defines the location of where the files appear during processing.
FileExtension name	Defines the file format to look for. This allows us to look for files matching a certain name.
NCPDPBatchRequest	
name	Refers to the element identifier.
regExHeader	Defines the regular expression for the batch header segment.
regExDetail	Defines the regular expression for the batch detail segment.
regExTrailer	Defines the regular expression for the batch trailer

File Name: NCPDPBatchConfiguration.xml	
Attribute Name	**Description**
	segment.
NCPDPSubmitterThreadWorker	
name	Refers to the element identifier.
ipAddress	Defines the MMIS ip address to send socket traffic to.
port	Defines the MMIS port number to send socket traffic to.
sendTimeout	Defines the timeout value for sending the request.
receiveTimeout	Defines the timeout value for receiving the response.
BatchHandler-name	Defines the name to call to use a batch hander.
Assembly-name	Refers to the location of the library.
Class-name	Refers to the class name to call in the library.
Method-name	Refers to the class method name to call in the class.
Method-methodType	Refers to the type of method call.
InitString	
commandChannel	Refers to the name of the command queue. The command queue points to one to many data queues. The command queue refers to a set of data.
dataChannel	Refers to the name of the data queue. One to many data messages make up a command queue.
maxBatchSize	Refers to the maximum batch size.
batchIDType	Refers to what type of ID is held.
maxWorkerThreads	Defines the number of threads connecting to the queues.
throttleInterval	Defines the interval time between messages.
Watcher / Writers	
name	Refers to the element identifier.
type	Defines if the logic is writing to the queue or reading from the queue.
ClassName value	Defines the class to invoke.
TriggeringInterval	Defines the interval time between poling.

File Name: NCPDPBatchConfiguration.xml	
Attribute Name	**Description**
TriggeringType	Defines if an event should call the invoking property.
WatcherLocation	Refers to the queue ID to read messages from.
WriteLocation	Refers to the queue ID to write messages to.
WriteAsync	Defines if messages are written Asynchronously or synchronously.
WatcherType	
name	Refers to the element identifier.
RegexFilter value	Offers the capability of reading Watcher types of a certain type.
RegexFilter enabled	Defines if the RegexFilter should be applied.
SharedModeDenyReceive value	Defines if the watcher deny receive is shared.
ReadTimeout value	Defines timeout values for reading the message from the watcher.
RecycleInterval	Defines time period to recycle the Watcher.
WriterType	
name	Refers to the element identifier.
Recoverable value	Defines if we should continue to process if an error occurred.

5 Appendix C – File Transfer Service Documentation

5.1 Introduction

File Transfer Service (FTS) is a windows service component which is used to transfer files to and from various end points of EDS/ Kentucky Medicaid Management Information System (MMIS) and between KY MMIS and its trading partners. File transfer service can transfer files using different transport types like File Transfer Protocol (FTP), Secure File Transfer Protocol using SSL (FTPS) and Uniform Naming Convention (UNC). The File Transfer Service has several features which are explained below. These features can be configured from FTS's configuration file.

5.1.1 File transfer Service Features

5.1.1.1 Tracking

File transfer service while transferring files, can track file information and file movement events and then logs them in a Oracle or Microsoft SQL Server database. These tracking records can be viewed using File Transfer Service website.

5.1.1.2 Archiving

If configured to archive, File transfer service can archive files to the UNC location given in its archive path configuration.

5.1.1.3 Duplicate Check

While transferring files FTS calculates each file's hash value (crc32) value and stores it in a database. If configured, FTS can check if the file is a duplicate file by checking its hash value against the already persisted hash values of older files.

5.1.1.4 Compress and Uncompress Files

File Transfer service can zip or unzip the file before it transfers it to it's destination depending on the configured values in the configuration file.

5.1.1.5 File Renaming

File Transfer service can rename a file while delivering to some pre-set formats. Renaming can include the Unique System Assigned Key (SAK), transport type, CRC32 value, system date, system time and/ or some constant string value.

5.1.1.6 Multiple File Delivery

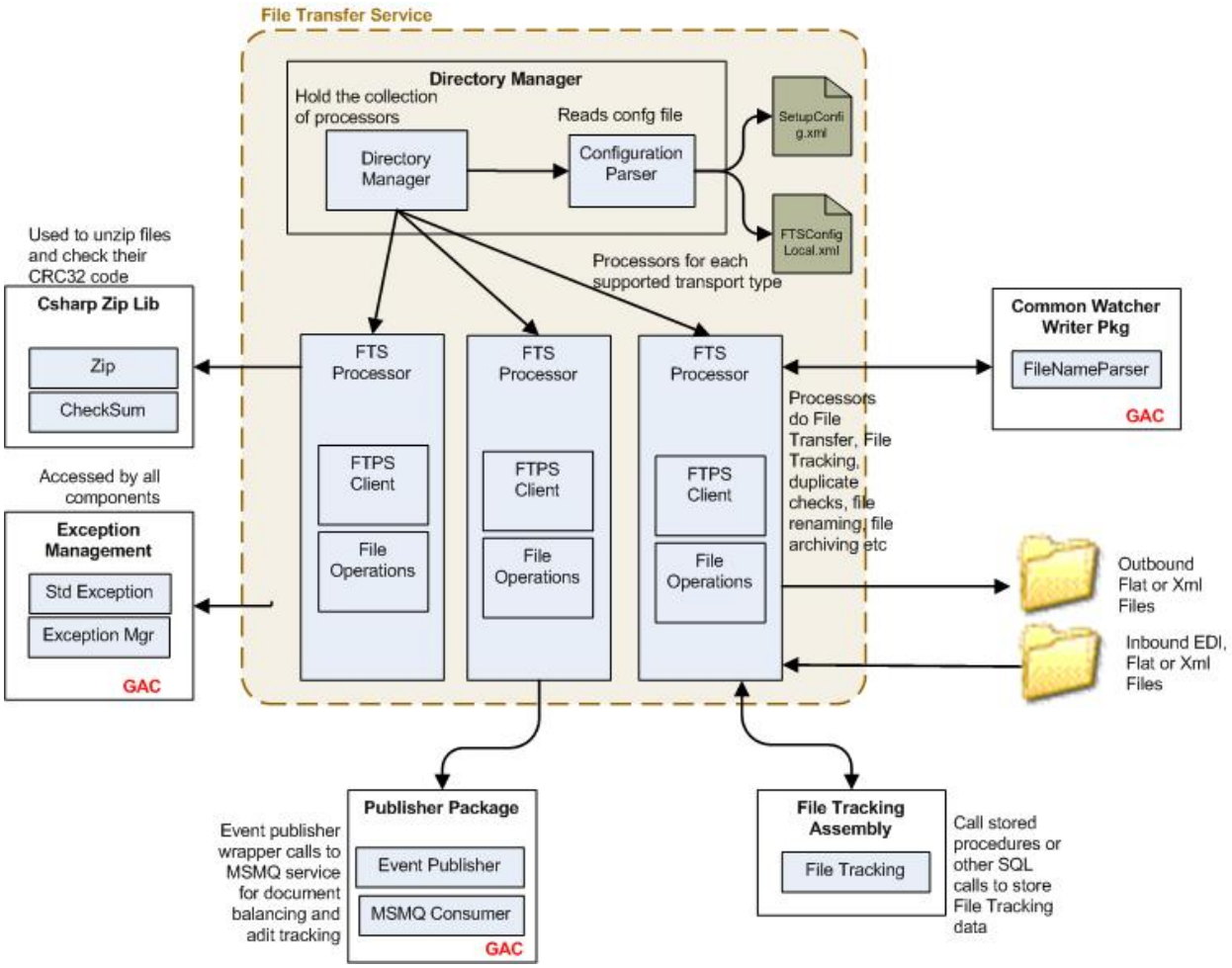
File Transfer service can deliver a copy of the file to up to three different locations. FTS can also be configured to overwrite a file if the file with the same name already exists at the destination location.

5.1.1.7 Exception Handling

If there is any technical or business exception while accessing or transferring file, all the errors are tracked along with other tracking data and an email is generated describing the issue and sent to configured email address.

5.2 File Transfer Service Solution Diagram

Below diagram shows all the components used by File transfer Service and how they interact with each other. In later section this document will go over each project related to File Transfer Service solution in detail.

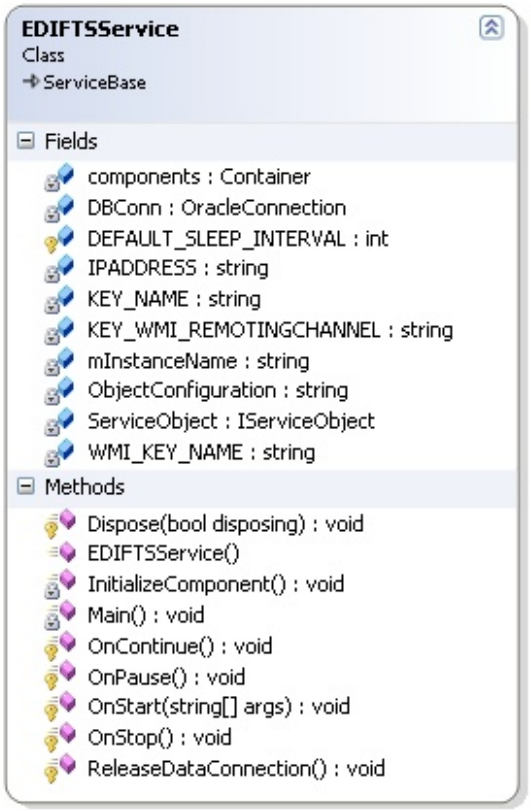


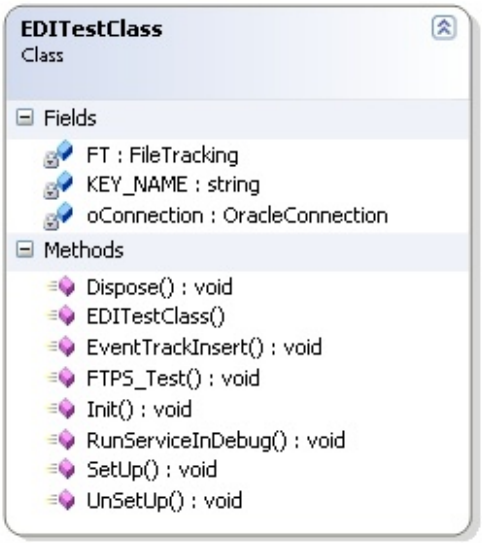
5.3 Projects Description

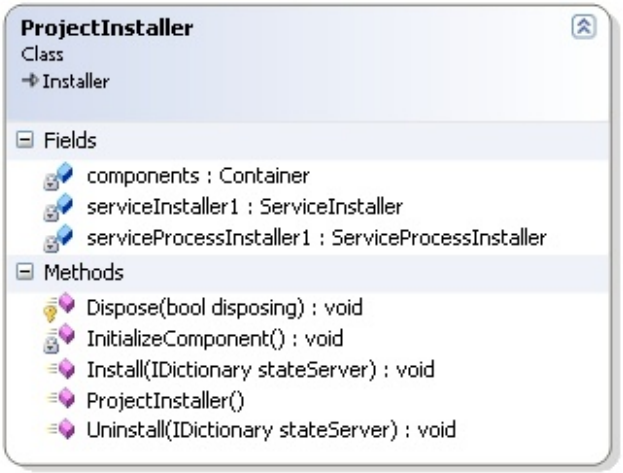
This section will explain all the projects involved in development of the File transfer Service. It will also go in to explaining the individual projects, their classes and the methods involved in supporting the required functionality of the File Transfer Service.

5.3.1 FTS Service Project

FTSService project is the main project which holds the actual windows service component. This windows service component is responsible for calling other assembly projects to perform its designed functions. This project also incorporates the test harness class (EDITestClass) which allows the service to be run in a debug mode using NUnit tool.







5.3.1.1 FTS Service Class

FTS Service is the file which has the main class, “EDIFTSService”, of the FTSService project where the windows service gets the control to Start, Stop, Pause or Restart itself.

File Name: FTSService.cs	**Class Name: EDIFTSService**
Method Name	**Description**
EDIFTSService	This is the default constructor of the class
Main	This is the entry point for the service control. FTS Service is instantiated in this method and is set to run.
InitializeComponent	This method is called by the constructor and is used to initialize the name of the class.
OnStart	This is the method which is called to start the service. This method reads the App.config to get the configuration file and passes it to Directory manager class. It then calls the Start method of the instantiated Directory Manager class.
OnStop	Before stopping, this method calls the ReleaseDataComponent method and then calls the Stop method of the above instantiated Directory Manager object. If successful it stops running the File transfer Service.
ReleaseDataComponent	This method closes the database connection.
Dispose	This method disposes off the service object.

5.3.1.2 Project Installer Class

This class is required to create a proper deployment (msi) package for the FTS service. This msi package is used to deploy the service in run time environment.

File Name: ProjectInstaller.cs	**Class Name: ProjectInstaller**
Method Name	**Description**
ProjectInstaller	This is the default constructor of the class and it calls InitializeComponent.
InitializeComponent	This is the designer generated code and should not be modified by developer using code editor.
Install	This method is called during install process of the windows service deployment package and responsible for installing the service component. This method is also used to setup the install properties of the service like service Name, service description and registry settings.
Uninstall	This method is called during uninstall process of the windows service. This method is used to remove the service related registry settings and finally uninstalls the

File Name: ProjectInstaller.cs	**Class Name: ProjectInstaller**
Method Name	**Description**
	service component.
Dispose	This method disposes off the service object.

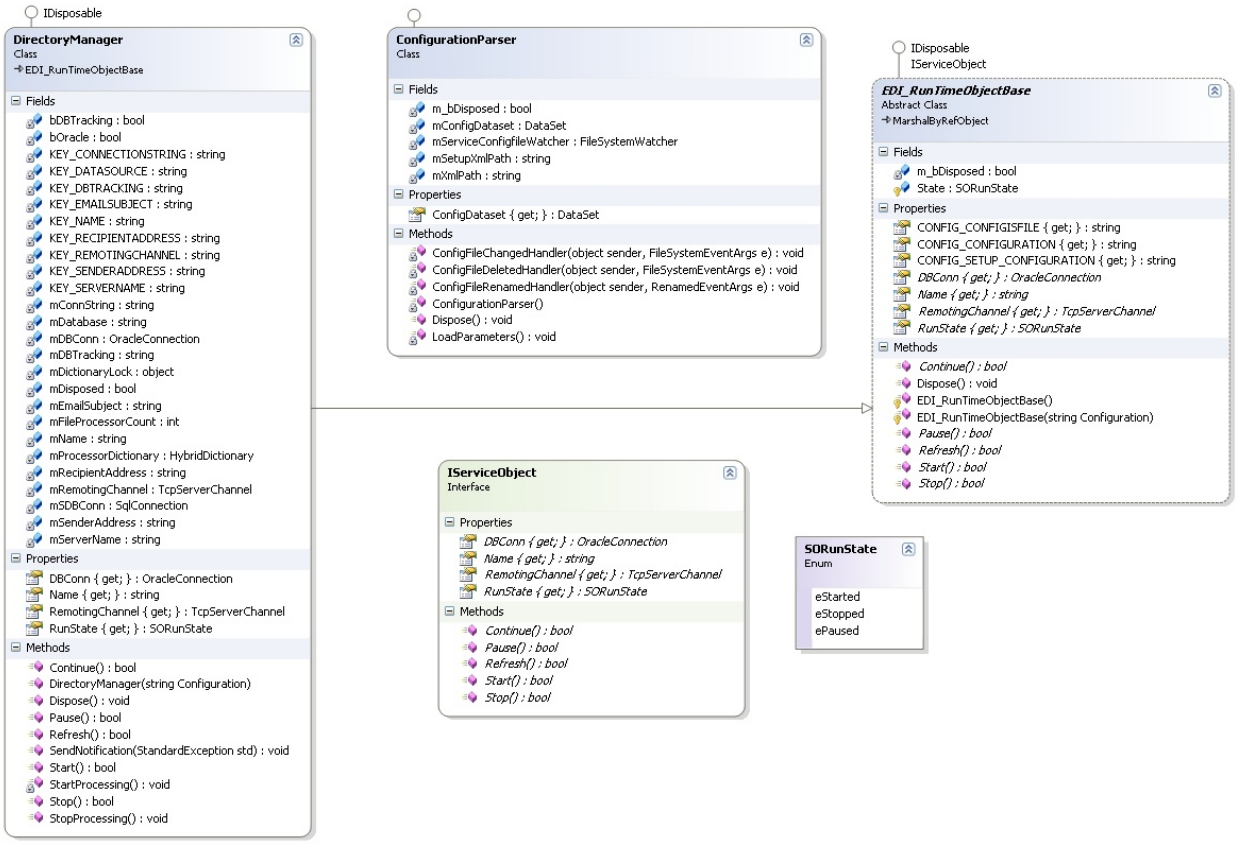
5.3.1.3 EDI Test Class

This class is a Text Fixture class of the NUnit tool and helps the developer to run the windows service in the debug mode.

File Name: EDITestClass.cs	**Class Name: EDITestClass**
Method Name	**Description**
RunFTSInDebug	This is the NUnit test that is run using the NUnit tool to start the File Transfer Service in the Debug mode. This replaces the need to install the service component. It performs the same functionality of getting the configuration file and then it calls the Directory Manager start class.

5.3.2 Directory Manager Project

Directory Manager Project is called by the EDIFTSService class and is responsible for setting up the runtime parameters of the service. In the Directory Manager project File Transfer Service accesses the configuration files and reads the configuration attributes needed to initializes File Transfer Processors for all supported transport types. After reading the configuration values it starts each File transfer Processor in a separate thread. This project is also responsible for cleaning up all the processors before FTS shuts down.



5.3.2.1 Directory Manager Class

Directory Manager Class implements EDI_RunTimeObjectBase class and IDisposable interface. Directory manager class reads the configuration values from App.config (FTSService.exe.config during runtime), SetupConfig.xml and FTSConfigLocal.xml files to get configuration values which are common to the entire service. Based on those values it knows which directories to monitor, at what polling rate and how many threads to create for each transport modes. When the Directory Manager gets a call to stop the File Transfer Service then it is also responsible to kill each file transfer processor thread which will finally stop the service.

File Name: DirectoryManager.cs	**Class Name: DirectoryManager**
Method Name	**Description**
DirectoryManager	This is the default constructor of the class. This method calls Configuration Parser class which reads and parses the SetupConfig.xml and FTSConfigLocal.xml file values. It also reads other common configuration values such as

File Name: DirectoryManager.cs	Class Name: DirectoryManager
Method Name	**Description**
	connection string, error notification email address etc.
Start	This method sets the service state to “Started” and calls StartProcessing method.
StartProcessing	This method goes over the records of SetupConfig.xml file and for each transport type gets the number of allowed threads and polling intervals. For each transport type it then checks the number of configurations set in FTSConfigLocal.xml. Based on the number of configurations set for a transport type it divides the number of directories to poll equally between each thread. Last thread may get more or less directories to monitor depending on the arithmetic mod value between processor thread count and the total number of directories to monitor. Then each thread is given the entire configuration details in a data table format and it is started. After each processor is made and started each processor instance is added in the Hybrid Dictionary and kept in memory.
Stop	This method sets the service state to “Stopped” and calls StopProcessing method.
StopProcessing	This method goes over each processor object in the Processor Directory Hybrid Dictionary and stops each processor and sets its state to ”Stopped”.
SendNotification	This method sends the notification along with publishing the custom Standard Exception object using Exception Manager’s Publish method.
Dispose	This method disposes off the service object.

5.3.2.2 Configuration Parser Class

This is a static class which is specifically made to read the configuration file and pass it to the Directory Manager class.

File Name: ConfigurationParser.cs	Class Name: ConfigurationParser
Method Name	**Description**
ConfigurationParser	This is the default constructor of the class. This is the methods which actually reads the App.config file and reads the path to the SetupConfig.xml and FTSConfigLocal.xml files. It then calls LoadParameters method.
LoadParameters	This method loads the Xml configuration files in a dataset. It then copies the values of the data table to a local dataset which is then made available to other classes though a

File Name: ConfigurationParser.cs	Class Name: ConfigurationParser
Method Name	**Description**
	public property.
Dispose	This method disposes off the service object.

5.3.2.3 IService Object Interface

This is the interface which is implemented by Directory Manager and EDI Runtime Object Base classes. This class is present in other EDI services also so the implementation of all EDI services is consistent.

File Name: ConfigurationParser.cs	Class Name: ConfigurationParser
Method Name	**Description**
Start	Interface method.
Stop	Interface method.
Continue	Interface method.
Refresh	Interface method.
Pause	Interface method.

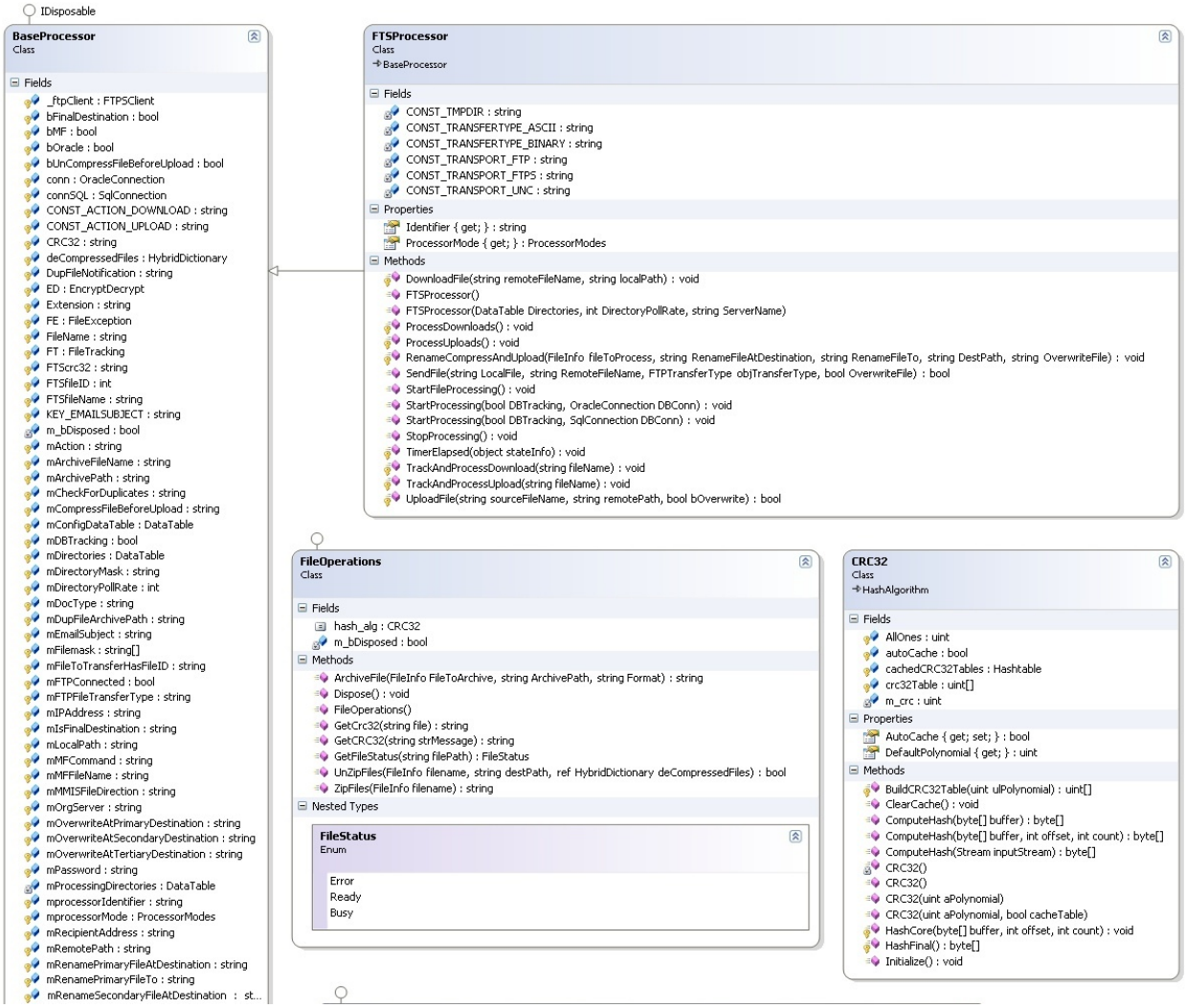
5.3.2.4 EDI Runtime Object Base Class

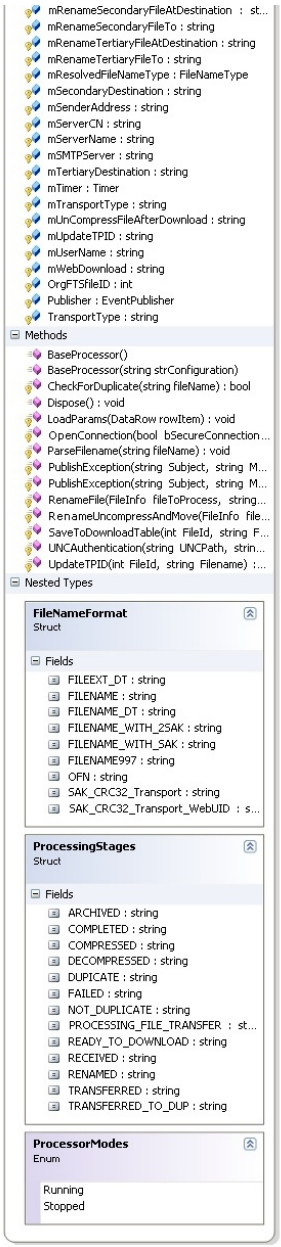
This is an abstract class which implements MarshalByRefObject class, IDisposable interface and IServiceObject interface. The purpose of this class is to give a layer to add EDI subsystem specific code.

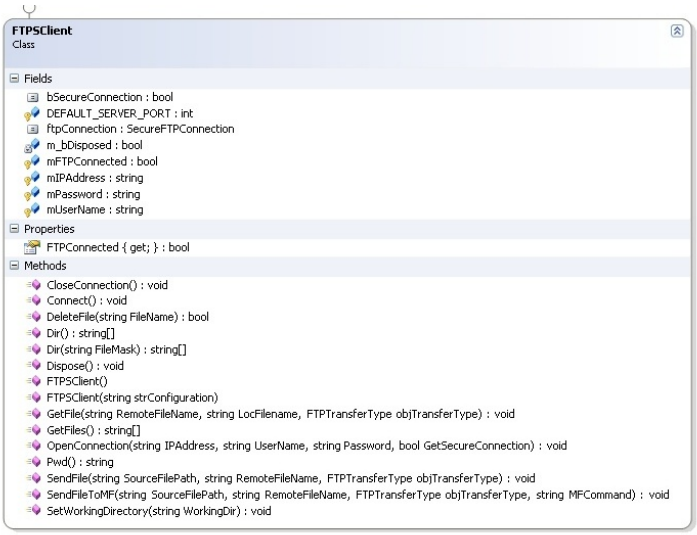
File Name: EDI_RunTimeObjectBase.cs	Class Name: EDI_RunTimeObjectBase
Method Name	**Description**
Start	Abstract method.
Stop	Abstract method.
Continue	Abstract method.
Refresh	Abstract method.
Pause	Abstract method.

5.3.3 FTS Processor Project

This is the project which is actually responsible for allowing File Transfer Service to transfer and track files. File Transfer service starts multiple instances of these assemblies in separate threads and each processor is given one or more directory folders to monitor at a given polling interval.







5.3.3.1 Base Processor Class

As the name says, this is the base class for FTS Processor class. It has all the common functions needed for file transfer and tracking.

File Name: BaseProcessor.cs	**Class Name: BaseProcessor**
Method Name	**Description**
BaseProcessor	This is the default constructor of the class.
LoadParams	This method reads the configuration record of the FTSConfigLocal.xml configuration file and sets the values to local variables.

File Name: BaseProcessor.cs	**Class Name: BaseProcessor**
Method Name	**Description**
OpenConnection	This instantiates the FTPSClient class and calls its OpenConnection method. This is called if the transfer mode is FTP or FTPS and this method opens a connection to the FTP server.
CheckForDuplicate	If the check for duplicate is turned on in the configuration this method uses the CRC32 value of the file to check if the file being transferred is a duplicate file. File name is not considered in the duplicate check calculation. Actual check is done by calling FileTracking class's ValidateIfFileExistsSelect method. If it returns a list of records then the data is read and an email is sent to the preconfigured email address giving details of the present file and the previous record of the similar files.
ParseFileName	This method calls the FileNameParser assemblie's EDIFileName class to parse values like CRC32, System assign Key (SAK), transport type, date, time, original file name and file extension from the file name.
PublishException	This is a common method to persist custom standard exception information and notification information in the database. If the File ID (SAK) is present then it also correlates the exception records with the file tracking records.
RenameFile	This method uses the "rename filename to" configuration value to change the file name of the file being delivered to the destination.
RenameUncompressAndMove	This is the actual method which calls the move or gets method to transfer the file in UNC or FTP mode. Depending upon the configuration settings it may rename the file, uncompress the file by calling UnZipFiles method of File Operations class and finally move the file. Also as per the configuration file settings if the file has WebDownload configuration value set to true this method inserts a record in T_Download table by calling SaveToDownloadTable method.
SaveToDownloadTable	This method is called to save File Tracking record to T_Download table. If a record is present in the T_Download table for a particular trading partner then that trading partner can download the file using web portal panels.
UpdateTPID	This method is called after a successful call to SaveToDownloadTable from RenameUncompressAndMove or from

File Name: BaseProcessor.cs	Class Name: BaseProcessor
Method Name	**Description**
	RenameCompressAndUpload method. This method adds the trading partner ID in the T_FILE_TRACK table against the file tracking record of the file.
Dispose	This method disposes off this object.

5.3.3.2 FTS Processor Class

This is the main class which is responsible for transfer of a file. It implements Base Processor class and with its help does the actual file movement depending upon the configuration values.

File Name: FTSProcessor.cs	Class Name: FTSProcessor
Method Name	**Description**
FTSProcessor	This is the default constructor of the class and is used to set the common values of the configuration being processed.
ProcessDownloads	This method checks the source directory to get the list of all the files that match the configured file mask. For each valid file in the source directory it then calls the DownloadFile method.
DownloadFile	This process actually moves the file if the process is set to download. If the transport type is FTP or FTPS then it gets the file from remote server to local temporary location. If the transport type is UNC then it moves the file from source location to local temporary location.
TrackAndProcessDownload	In this method for every file which does not have File ID (SAK) new file tracking record is created and file tracking is started. After that file is copied to a temporary “working” folder and then RenameUnCompressAndMove method is called. If more than one file copy is needed and if proper configuration exists then the RenameUnCompressAndMove method is called for second and third time also. After all file transfers are completed the copy of the file in the working folder is deleted.
ProcessUploads	This method checks the source directories from where file needs to be uploaded to the destination folder. For all the files which FTS can lock and for the files not having the file extension as “.TMP” this method calls TrackAndProcessUpload method.
TrackAndProcessUpload	In this method for every file which does not have File ID (SAK) new file tracking record is created and file tracking is started. After that file is copied to a temporary “working”

File Name: FTSProcessor.cs	Class Name: FTSProcessor
Method Name	**Description**
	folder and then RenameCompressAndUpload method is called. If more than one file copy is needed and if proper configuration exists then the RenameCompressAndUpload method is called for second and third time also. After all file transfers are completed the copy of the file in the working folder is deleted.
RenameCompressAndUpload	This method compresses or uncompresses files before the files are uploaded by calling UploadFile method. Before compressing or uncompressing It also checks the file extension to check if the file is already compressed or uncompressed and if is then it skips this process and directly goes to UploadFile.
UploadFile	This file is responsible for setting the FTP or FTPS connection settings before the file is uploaded using third party software called EDT FTP Net Pro. Call to that software is made by calling SendFile.
SendFile	This method decides if the file needs to be uploaded to mainframe server or any other FTP or FTPS server. It also checks if the file with the same file name already exists on the remote server. If it exists and overwrite at destination' attribute is set to true then the remote file is deleted otherwise a Standard exception is created and thrown. Final transfer happens by calling SendFile or SendFileToMF of the FTPSClient class.
StartProcessing	This is the method which is called by DirectoryManager when it creates a new thread of the File Transfer processor. This method calls the TimerElapsed delegate and then sends the control back to the Directory Manager so the service component can finish its start service command.
TimerElapsed	This is a delegate which is called once by StartProcessing method when the thread is started. This method has an endless while loop which keeps the thread alive all the time. This method calls StartFileProcessing which does the file transfer for all the directories this thread is assigned to monitor and then returns the call to TimeElapsed method. After that this method puts the thread to sleep for the period of configured polling interval and then again calls the StartFileProcessing method. This way the loop goes on until File transfer Service Stop method is called.
StartFileProcessing	This method goes over all the configurations given to a particular thread to manage. Then for each configuration if required it opens the FTP or FTPS connection and then

File Name: FTSProcessor.cs	Class Name: FTSProcessor
Method Name	**Description**
	depending upon either upload or download action calls ProcessUploads or ProcessDownloads method. This method is also responsible for doing the final exception processing so an exception is never passed back to the TimerElapsed method. This helps in keeping the thread alive.
StopProcessing	This puts the file transfer processor in a stopped state and makes the wait interval variable to infinite.

5.3.3.3 File Operations Class

This class implements IDisposable interface and is responsible to hold all the methods required for file processing during its transfer. Methods implemented in this class are mentioned below.

File Name: FileOperations.cs	Class Name: FileOperations
Method Name	**Description**
FileOperations	This is the default constructor of the class.
ArchiveFile	This method renames the file as per the Archive File Name configuration value and moves it to the Archive location set in the Archive Path folder.
GetCrc32	This method calculates the Crc32 of the file using the ICSharpCode.SharpZipLib assembly's Crc32 class. After getting the Crc32 value it converts the value from number format to hex format which is an eight digit alpha numeric value. This value matches with the Crc32 value calculated by WinZip product.
GetFileStatus	This method tries to get a lock on the file to check if this file is being used by any other process. If it cannot create a lock on the file then it returns a file status of Busy otherwise it returns a status of Ready.
UnZipFiles	This method uses ICSharpCode.SharpZipLib assembly's ZipOutputStream to compress a file. It also calculates the file's Crc32 value and puts it in the zipped file header.
ZipFiles	This method uses ICSharpCode.SharpZipLib assembly's ZipInputStream to uncompress a file. This method goes over all the entries of the zip file and uncompresses them.
Dispose	This method disposes off this object.

5.3.3.4 FTPS Client Class

This class implements IDisposable interface and has functions which are responsible for moving a file. It also acts as a wrapper for API calls to a third party FTP tool EDT FTP Net Pro software. File Transfer Service uses EDT FTP Net Pro to do actual FTP or FTPS mode file transfers.

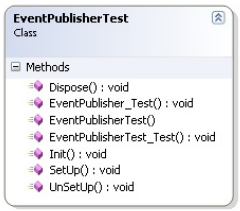
File Name: FTPSClient.cs	**Class Name: FTPSClient**
Method Name	**Description**
FTPSClient	This is the default constructor of the class.
GetFile	This method downloads a specific file from FTP servers to the local location.
GetFiles	This method gets the list of all files from the FTP location.
OpenConnection	This method opens a connection with FTP or FTPS remote server.
Pwd	This method is used to get the FTP control back to the home directory of the login used to connect.
SendFile	This method is used to upload a specific file from local location to the remote FTP or FTPS server.
SendFileToMF	This method is used to upload a specific file from local location to the remote FTP server hosted on Mainframe. This method also executes specific Mainframe commands to set the record set length and destination record set name.
SetWorkingDirectory	This method sets the working directory of the current connection.
Dir	This method gets all the files in the FTP server directory. If a file mask is give as an input parameter then it returns only those file names which satisfy file mask condition.
Dispose	This method disposes off this object.

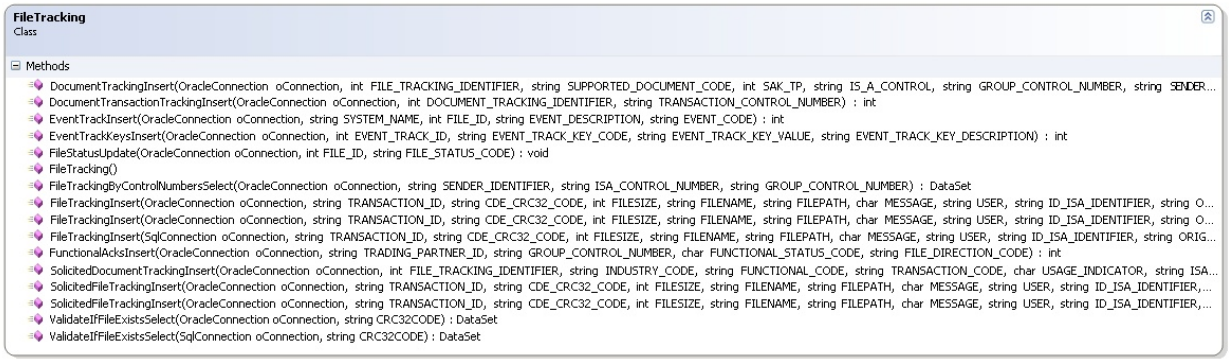
5.3.3.5 CRC32 Class

This class is obsolete. Instead of this class, CRC32 values are calculated using ICSharpCode.SharpZipLib assembly.

5.3.4 File Tracking Library Project

This project is used to create classes which help EDI service to start and continue file tracking through the n-layer solution spanning over multiple servers. This library is accessed from File Transfer Service, pre-processor service, post-processor service and Event Publisher.





5.3.4.1 File Tracking Class

This class has all the functions which provide the ability to add, modify or continue File tracking.

File Name: FileTracking.cs	**Class Name: FileTracking**
Method Name	**Description**
FileTracking	This is the default constructor of the class.
DocumentTrackingInsert	This method is not used by File Transfer Service.
DocumentTransactionTrackingInsert	This method is not used by File Transfer Service.
EventTrackInsert	This method is used to save events (audit steps) of the file processing to the database. This method uses SP_EVENT_TRACK_INS stored procedure of the PKG_TRACKING Oracle package.
FileStatusUpdate	This method is used to update file status in the database. This method uses SP_FILE_TRACK_UPD_STATUS stored procedure of the PKG_TRACKING Oracle package.
FileTrackingByControlNumbersSelect	This method is not used by File Transfer Service.
FileTrackingInsert	This method is used to create a new File Tracking record. This method is called to start the file tracking of any file. This method uses SP_FILE_TRACK_INS stored

File Name: FileTracking.cs	**Class Name: FileTracking**
Method Name	**Description**
	procedure of the PKG_TRACKING Oracle package
FunctionalAckInsert	This method is not used by File Transfer Service.
SolicitedDocumentTrackingInsert	This method is not used by File Transfer Service.
SolicitedFileTrackingInsert	This method is not used by File Transfer Service.
ValidateFileExistsSelect	This method checks the existing T_FILE_TRACK table records to match the Crc32 value of the file. This is used during the duplicate check of the file.

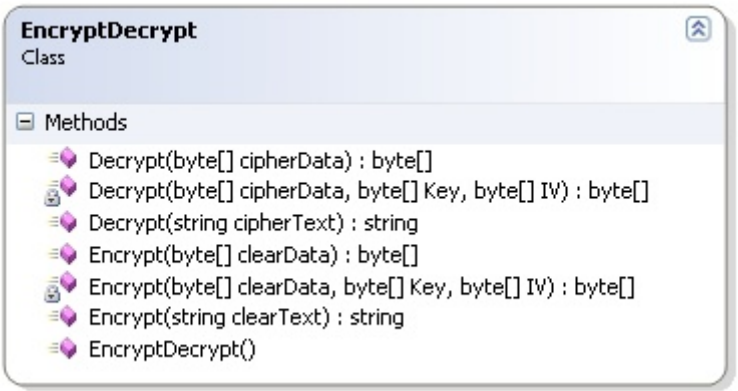
5.3.4.2 File Exception Class

This class provides the ability to correlate exception data persisted in database using Exception Manager Publish method with the File Tracking record.

File Name: FileException.cs	**Class Name: FileException**
Method Name	**Description**
FileException	This is the default constructor of the class.
FileTrackingExceptionsInsert	This method downloads a specific file from FTP servers to the local location.

5.3.5 Encrypt Decrypt Project

This project is used to encrypt or decrypt any text or byte array. This is useful to hide sensitive data like passwords, database connection strings etc from the regular users which can log on the runtime servers.



5.3.5.1 Encrypt Decrypt Class

This class is the main class which provides the methods to either encrypt or decrypt text or byte array. The class uses “Rijndael” algorithm to encrypt or decrypt the data. The reason to choose “Rijndael” algorithm is because it is strong and is available on most of the platforms.

File Name: EncryptDecrypt.cs	**Class Name: EncryptDecrypt**
Method Name	**Description**
EncryptDecrypt	This is the default constructor of the class.
Encrypt	This method is used to encrypt the input data.
Decrypt	This method is used to decrypt the input data.

5.4 File Transfer Service Configuration File

Configuration files are responsible for providing configuration data to File Transfer Service during run time. These files constitute data which needs to be changed often and is needed only during run time.

5.4.1 App.Config

This is the default configuration file that can be built along with the service component. This file access is supported by default .Net components. The purpose of this file is to host values which are common to all service components.

Sample App.config file:

```
<configuration>
      <appSettings>
            <add key="RemotingChannel" value="9085" />
            <add key="InstanceName" value="EDIObjectInstance" />
            <add key="DBTracking" value="ON" />
            <add key="DataSource" value="ORACLE" /> <!-- SQLServer -->
            <add key="DBConnection" value="Data Source=kymism1;User ID=kyedi;
Password=xxxxx; Integrated Security=no;" />
            <add key="Configuration"
value="C:\EDS\EDS.USGS.MED\EDI\EDS.USGS.MED.EDI\Common
Components\FileTransferService\DirectoryManager\FTSConfigLocal.xml" />
            <add key="Setup"
value="C:\EDS\EDS.USGS.MED\EDI\EDS.USGS.MED.EDI\Common
Components\FileTransferService\DirectoryManager\SetUpConfig.xml" />
            <add key="WMIIPAddress" value="SBG" />
            <add key="WMIPortNo" value="9090" />
            <add key="WMIRuntimeObjectName" value="EDS.MSMQ.TCP" />
            <add key="RecipientAddress" value="manish.sethi-eds@eds.com" />
            <add key="SenderAddress" value="manish.sethi-eds@eds.com" />
            <add key="EmailSubject" value="FTS Testing MANISH " />
            <add key="ServerName" value="BBS" />
            <!-- production licence -->
            <!-- expires 30 Dec 2099 -->
            <add key="license.owner" value="EDSKentuckyMedicaid"/>
            <add key="license.product" value="edtftpnetpro"/>
            <add key="license.key" value="xxxxxxxxxxx"/>
      </appSettings>
</configuration>
```

File Name: FTSService.exe.config	
Attribute Name	**Description**
RemotingChannel	This attribute holds the value through which we can remote into FTS. This is not yet implemented.
InstanceName	This is the channel name under which it will be registered. This is not yet implemented.
DBTracking	This attribute decides whether FTS should do database tracking during file transfer. Valid values are:

File Name: FTSService.exe.config	
Attribute Name	**Description**
	ON or OFF
DataSource	This attribute's value tells FTS if it needs to log data to Oracle or SQL Server Database. Valid values are: ORACLE or SQLServer.
DBConnection	This attribute value holds the database connection string.
Configuration	This attribute value holds the location of the FTSConfigLocal.xml configuration file.
Setup	This attribute value holds the location of the SetupConfig.xml configuration file.
WMIIPAddress	Not used.
WMIPortNo	Not used.
WMIRuntimeObjectName	Not used.
RecipientAddress	This is the global recipient email address where error notification emails are sent.
SenderAddress	This is the global sender email address used to send error notification emails.
EmailSubject	This is the common text that can be attached to the notification email.
ServerName	This is the name of the server where FTS is deployed.
license.owner	This is the license information for EDT FTP Net Pro software.
license.product	This is the license information for EDT FTP Net Pro software.
license.key	This is the license information for EDT FTP Net Pro software.

5.4.2 FTSConfigLocal.xml

This is the main configuration file which holds the configuration details about which directory to monitor, where to deliver, what file names to pick, how to rename at destination, where to archive etc. There is supposed to be at least one configuration record for each file transfer. If the source and destination locations are both accessed through FTP or FTPS transport type then such transfers are accomplished using two configuration records, one to download the file from the resource server and the other to upload the file to the destination server.

Sample FTSConfigLocal.xml file:

```
<configurationSettings>
<configuration
TransportType="UNC"
Action="upload"
MMISFileDirection="INBOUND"
Source="C:\Edifecs_testing\FTS\In"
SourceFileMasks="*.zip"
Destination="C:\Edifecs_testing\FTS\Out"
SecondaryDestination="C:\Edifecs_testing\FTS\Out2"
TertiaryDestination=""
CheckForDuplicates="False"
DupFileArchivePath="C:\FTS\DUPLICATE"
FileToTransferHasFileID="false"
RenamePrimaryFileAtDestination="true"
RenamePrimaryFileTo="&lt;SAK&gt;_&lt;CRC32&gt;_&lt;TransportType&gt;_*.*"
RenameSecondaryFileAtDestination="false"
RenameSecondaryFileTo=""
RenameTertiaryFileAtDestination="false"
RenameTertiaryFileTo=""
ArchivePath="C:\FTS_Archive\Member\binary"
ArchiveFileName=""
CompressFileBeforeUpload="false"
UnCompressFileAfterDownload="false"
UnCompressFileBeforeUpload="true"
FTPServerIPAddress="205.204.195.16"
FTPServerUserName="FTPS_TEST"
FTPServerPassword="LXRcZwPTfn+lVmae5MqZRLHEMjVevflvjM8nKzOrZnQ="
FTPFileTransferType="BINARY"
NotificationRecipientAddress="John.Doe@eds.com"
NotificationSenderAddress=" John.Doe@eds.com"
NotificationSMTPServer=""
OverwriteAtPrimaryDestination="false"
OverwriteAtSecondaryDestination="false"
OverwriteAtTertiaryDestination="false"
IsFinalDestination="false"
OriginatingServer="Commonwealth">
</configuration>
</configurationSettings>
```


File Name: FTSConfigLocal.xml	
Attribute Name	**Description**
TransportType	This is the transport type which will be used to move the file. Possible values are: UNC, FTP or FTPS.
Action	This value determines if we are uploading a file to the destination or downloading it from there.
MMISFileDirection	This attribute determines the direction of file with respect to MMIS.
Source	This is the directory or location where source file will be

File Name: FTSConfigLocal.xml	
Attribute Name	**Description**
	picked from.
SourceFileMasks	This value determines the file mask FTS is supposed to use to pick up source file. Files not conforming to the source file mask value are not transferred.
Destination	This is the destination location where the primary file will be delivered.
SecondaryDestination	This is the destination location where the second copy of the file will be delivered.
TertiaryDestination	This is the destination location where the third copy of the file will be delivered.
RenamePrimaryFileAtDestination	This attribute takes true or false value to determine if the primary file should be renamed before being delivered to destination.
RenamePrimaryFileTo	This value describes the format primary file should be renamed to. Possible values are: <SAK>_<CRC32>_<TransportType>_* <SAK>_<CRC32>_<TransportType>_<WebUID>_* <FILENAME> <997> <FILENAMEWSAK> <FILENAMEW2SAK> <FILENAME>_<DT> <EXT>_<DT> <OFN>
RenameSecondaryFileAtDestination	This attribute takes true or false value to determine if the second file should be renamed before being delivered to destination.
RenameSecondaryFileTo	This value describes the format second file should be renamed to. Possible values are: <SAK>_<CRC32>_<TransportType>_* <SAK>_<CRC32>_<TransportType>_<WebUID>_* <FILENAME> <997> <FILENAMEWSAK> <FILENAMEW2SAK> <FILENAME>_<DT> <EXT>_<DT> <OFN>
RenameTertiaryFileAtDestination	This attribute takes true or false value to determine if the third file should be renamed before being delivered to

File Name: FTSConfigLocal.xml	
Attribute Name	**Description**
	destination.
RenameTertiaryFileTo	This value describes the format third file should be renamed to. Possible values are: <SAK>_<CRC32>_<TransportType>_* <SAK>_<CRC32>_<TransportType>_<WebUID>_* <FILENAME> <997> <FILENAMEWSAK> <FILENAMEW2SAK> <FILENAME>_<DT> <EXT>_<DT> <OFN>
CheckForDuplicates	This attribute takes true or false value. If the value is set to true then FTS checks if the file is a duplicate otherwise duplicate check is not performed.
DupFileArchivePath	If the file is a duplicate file then this attribute value determines where the failed file should be moved to.
ArchivePath	This attributes gives the location where the file should be archived. If this value is blank then the file is not archived.
ArchiveFileName	This field determines how file should be renamed before it is archived.
CompressFileBeforeUpload	This attribute takes a true or false value where true value leads the file to be zipped before it is uploaded to the destination.
UnCompressFileAfterDownload	This attribute takes a true or false value where true value leads the file to be unzipped before it is downloaded from the source.
UnCompressFileBeforeUpload	This attribute takes a true or false value where true value leads the file to be unzipped before it is uploaded to the destination.
FTPServerIPAddress	This field holds the host name or the IP address of the FTP server.
FTPServerUserName	This field has the login value for the FTP server.
FTPServerPassword	This field has the password to access the FTP server. This value is encrypted and FTS decrypts this value before using.
FTPFileTransferType	This determines the FTP transfer type. Possible values

File Name: FTSConfigLocal.xml	
Attribute Name	**Description**
	are: BINARY ASCII
NotificationRecipientAddress	This is the recipient email address where error notification emails are sent if there is an error while transferring files related to this configuration.
NotificationSenderAddress	This is the sender email address which is used to send notification emails if there is an error while transferring files related to this configuration.
NotificationSMTPServer	This is the SMTP server used to send notification emails.
OverwriteAtPrimaryDestination	This attribute takes a true or false value. If set to true and if a file with a similar name to primary file already exists at the destination folder then the destination file is overwritten.
OverwriteAtSecondaryDestination	This attribute takes a true or false value. If set to true and if a file with a similar name to secondary file already exists at the destination folder then the destination file is overwritten.
OverwriteAtTertiaryDestination	This attribute takes a true or false value. If set to true and if a file with a similar name to tertiary file already exists at the destination folder then the destination file is overwritten.
IsFinalDestination	This attribute takes a true or false value. If this value is set to true then after file transfer, File Status is updated to “COMPLETED” otherwise it is updated to “TRANSFERRED”.
OriginatingServer	This is information only field used while creating a file tracking record and is supposed to hold the value of the server name where the source file exists.

5.4.3 SetupConfig.xml

This configuration file is responsible for setting up the thread count values of each supported transport type and their polling interval.

```
<Configuration>
      <SetUp TransportType="FTP" ThreadCount="2" PollRate="1000" />
      <SetUp TransportType="UNC" ThreadCount="5" PollRate="1000" />
      <SetUp TransportType="FTPS" ThreadCount="2" PollRate="1000" />
</Configuration>
```

File Name: SetupConfig.xml	
Attribute Name	**Description**
TransportType	This attribute value defines the transport type whose thread count and polling rate will be configured.
ThreadCount	This attribute value gives the number of threads to be created for a specific transport type.
PollingInterval	This value determines the polling interval of each directory.

6 Appendix D – FTS/ TPM UI User Manual

6.1 Introduction

The FTS / TPM website serves two primary purposes. The first purpose is to monitor and track files that have been passed through the File Transfer Service system. These files include Electronic Data Interchange (EDI) data and other file format transactions. The secondary purpose is to manage and maintain the EDI Trading Partner agreement for Providers, Billing agents, clinics, Managed Care Organizations (MCO) and similar entities.

6.1.1 Overview

This user manual provides a detailed analysis of the File Tracking and Trading Partner Management areas of functionality.

6.1.2 File Tracking

EDI services work together to create an n-layer solution which can monitor and track all files. These files can be X12 files associated with a specific document type or other generic files going to or from MMIS system. For X12 files EDI services do more detailed tracking in which it gets the key values from the file contents while for any other file type File Transfer Service just logs generic file details and its audit events.

6.1.3 Trading Partner Management

This area of the web application is used by EDI Help desk to setup or maintain trading partner contracts and configure their contract for different types of EDI transactions. EDI documents sent by trading partners are processed by EDI services only if they pass validation against trading partner contracts.

6.1.4 Maintenance

Maintenance is an area of the application that provides support utilities to the management of Trading Partners; however, the FTS XML configuration application is also included in the Maintenance area of the application and is detailed in this document.

6.2 System Description

This section of the user manual provides a description of each functional area and a detailed account of the fields that exist on each page of the application.

6.2.1 File Tracking

File Tracking is the general term used to track files that fall into many different levels of criteria.

6.2.1.1 Page: File Tracking – Search

6.2.1.1.1 Description:

The basic file tracking search page can search for every kind (including EDI) of file that is processed by the File Transfer Process.

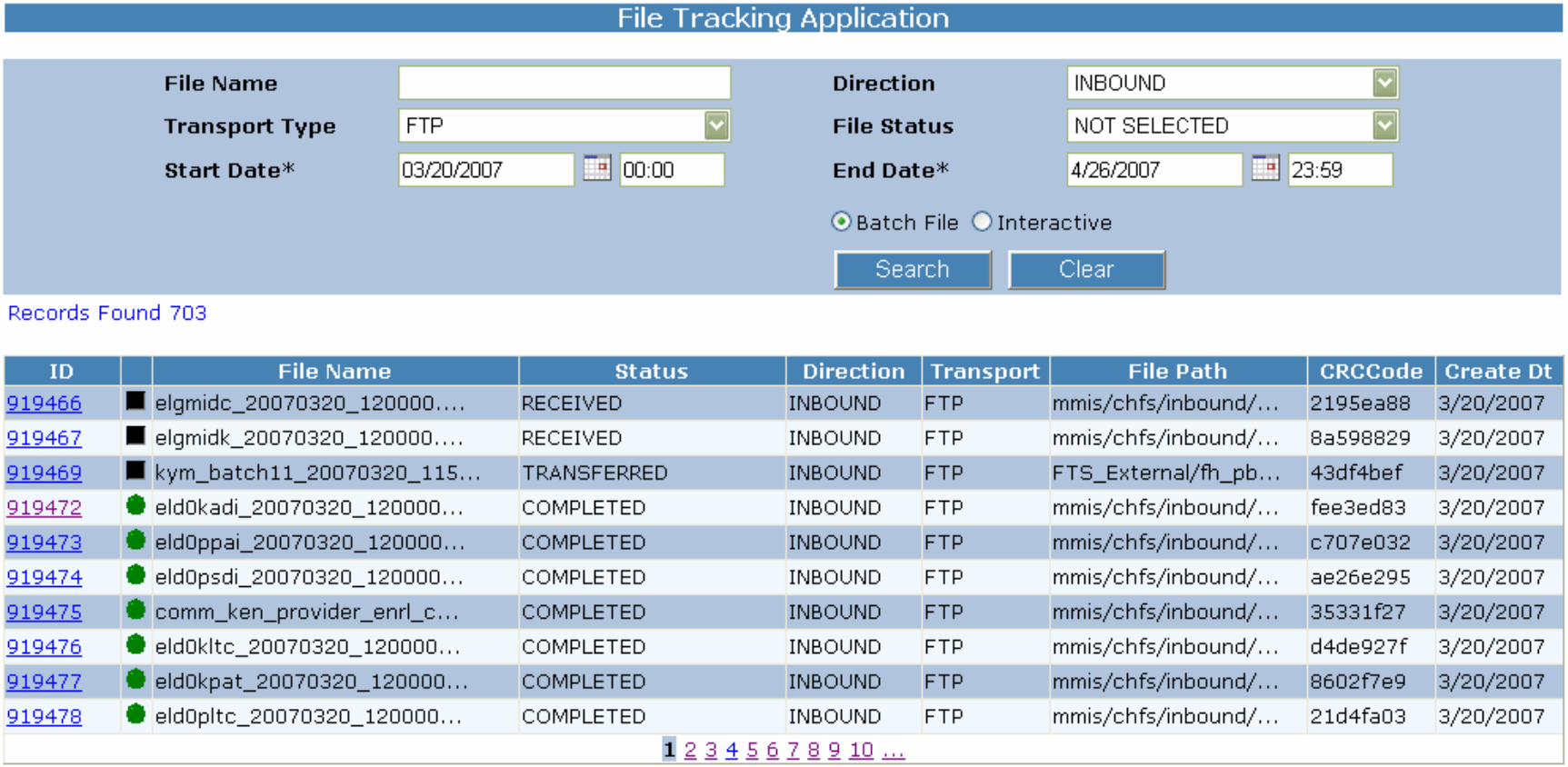
6.2.1.1.2 Technical Name:

EDIUtilities.FileTracking

6.2.1.1.3 Web Page Name:

File Tracking

6.2.1.1.4 File Tracking - Layout



6.2.1.1.5 File Tracking - Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File Name	The complete name of the file including the extension	Text Box	VARCHAR 2	255	T_FILE_TRAC K	NAM_File
Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR	255	T_CDE_FILE_	DSC_FILE _

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
			2		DIRECTION	DIRECTIO N
Transport Type	The method used to transport the file through the system	Drop Down Box	VARCHAR 2	50	T_CDE_ TRANSPORT	CDE_ TRANSPO RT
File Status	The current status of the file	Text Box	VARCHAR 2	255	T_CDE_FILE_ STATUS	DSC_FILE _ STATUS
Start Date	The date to begin the search	Text Box	DATE		T_FILE_ TRACK	DTE_ CREATED
End Date	The date to end the search	Text Box	DATE		T_FILE_ TRACK	DTE_ CREATED
Batch / Interactive	These radio controls allow the search results to be limited to batch or interactive files.	Radio Control				Calculated
ID	The ID of the file	Label (Selectable Link)	NUMBER	9	T_FILE_ TRACK	SAK_FILE_ TRACK
Status	The current status of the file	Label	VARCHAR 2	255	T_CDE_FILE_ STATUS	DSC_FILE _ STATUS
Direction	The direction of the file with respect to MMIS	Label	VARCHAR 2	255	T_CDE_FILE_ DIRECTION	DSC_FILE _ DIRECTIO N

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Transport	The method used to transport the file through the system	Label	VARCHAR 2	50	T_CDE_ TRANSPORT	CDE_ TRANSPO RT
File Path	The complete path to the location of the file	Label	VARCHAR 2	255	T_FILE_TRAC K	NAM_FILE PATH
CRC Code	The Cyclic Redundancy Check (CRC) used to ensure no intermediate changes were made to the file	Label	VARCHAR 2	8	T_FILE_ TRACK	CDE_CRC 32
Create Date	The date the file was initially created	Label	DATE		T_FILE_ TRACK	DTE_CRE ATED

6.2.1.2 Page: EDI File Tracking – Search

6.2.1.2.1 Description:

EDI File Tracking provides a method to search for specific EDI files that are associated with a certain document type. Files can also be further limited to a specific trading partner.

6.2.1.2.2 Technical Name:

EDIUtilities.EDIFileTracking

6.2.1.2.3 Web Page Name:

EDI File Tracking

6.2.1.2.4 EDI File Tracking – Layout

EDI File Tracking

File Name		Document Type	NOT SELECTED
Trading Partner Name		Trading Partner ID	
Direction	INBOUND	File Status	NOT SELECTED
Start Date*	03/20/2007 00:00	End Date*	4/04/2007 23:59
Transport Type	FTP		Batch File / Interactive

Search Clear

Records Found 43

ID		File Name	Doc Type	TP ID	Status	Direction	File Path	Create Dt
919779	▲	KYW837PR_0_20070228_1200...	837_P_...	BLOCKVISION	FAILED	INBOUND	FTS_External/passp...	3/20/2007
919787	◆	nsf_20070320.dat	NSF_BA...		PARTIALLY_ACCEPTED	INBOUND	mmis/kytc/inbound	3/20/2007
919824	▲	KYW837PR_0_20070321_1234...	837_P_...	KYMEDICAID	FAILED	INBOUND	FTS_External/passp...	3/21/2007
919854	●	KYW837IE_0_20070305adj.z...	837_I_...	KYMEDICAID	COMPLETED	INBOUND	FTS_External/passp...	3/21/2007
919855	◆	KYW837IN_0_20061227_0318...	837_I_...	KYMEDICAID	PARTIALLY_ACCEPTED	INBOUND	FTS_External/passp...	3/21/2007
919856	●	KYW837IN_ENCXADJ.zip	837_I_...	KYMEDICAID	COMPLETED	INBOUND	FTS_External/passp...	3/21/2007
919857	●	KYW837IN_ENCXVOID.zip	837_I_...	KYMEDICAID	COMPLETED	INBOUND	FTS_External/passp...	3/21/2007
919858	●	KYW837PE_0_ENCPADJ.zip	837_P_...	KYMEDICAID	COMPLETED	INBOUND	FTS_External/passp...	3/21/2007
919859	●	KYW837PR_0_20070303_0318...	837_P_...	KYMEDICAID	COMPLETED	INBOUND	FTS_External/passp...	3/21/2007
919861	●	KYW837PR_XOVOID.zip	837_P_...	KYMEDICAID	COMPLETED	INBOUND	FTS_External/passp...	3/21/2007

1 2 3 4 5

6.2.1.2.5 EDI File Tracking – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File Name	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_FILE

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Document Type	The document type of the file	Drop Down Box	VARCHAR2	50	T_CDE_ SUPPORTED_ DOC	CDE_ IDENTIFICATION
Trading Partner Name	The name of the Trading Partner associated with this file	Text Box	VARCHAR2	50	T_TP	NAM_TP
Trading Partner ID	The ID of the Trading Partner associated with this file	Text Box	NUMBER	9	T_TP	SAK_TP
Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS
File Status	The current status of the file	Text Box	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
Start Date	The date to begin the search	Text Box	DATE		T_FILE_ TRACK	DTE_ CREATED
End Date	The date to end the search	Text Box	DATE		T_FILE_ TRACK	DTE_ CREATED
Transport Type	The method used to transport the file through the system	Drop Down Box	VARCHAR2	50	T_CDE_ TRANSPORT	CDE_ TRANSPORT
Batch / Interactive	These radio controls allow the search results to be limited to batch or interactive files.	Radio Control				Calculated
ID	The ID of the file	Label (Selectable Link)	NUMBER	9	T_FILE_ TRACK	SAK_FILE_ TRACK
Doc Type	The document type of the file	Label	VARCHAR2	50	T_CDE_ SUPPORTED_ DOC	CDE_ IDENTIFICATION

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
TP ID	The ID of the Trading Partner associated with this file	Label	NUMBER	9	T_TP	SAK_TP
Status	The current status of the file	Label	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
Direction	The direction of the file with respect to MMIS	Label	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS
File Path	The complete path to the location of the file	Label	VARCHAR2	255	T_FILE_TRACK	NAM_FILEPATH
Create Date	The date the file was initially created	Label	DATE		T_FILE_ TRACK	DTE_CREATED

6.2.1.3 Page: File Tracking Details

6.2.1.3.1 Description:

Both the basic and EDI file tracking search pages will return a list of records for all of the files that matched the criteria chosen in the search panel. A single record can be chosen to display a full analysis for the specified file. The File Tracking Details page displays a detailed view of the file, along with any correlation files, events, exceptions, or notifications that are associated with the file. The panels below give a detailed breakdown of each field included for every panel.

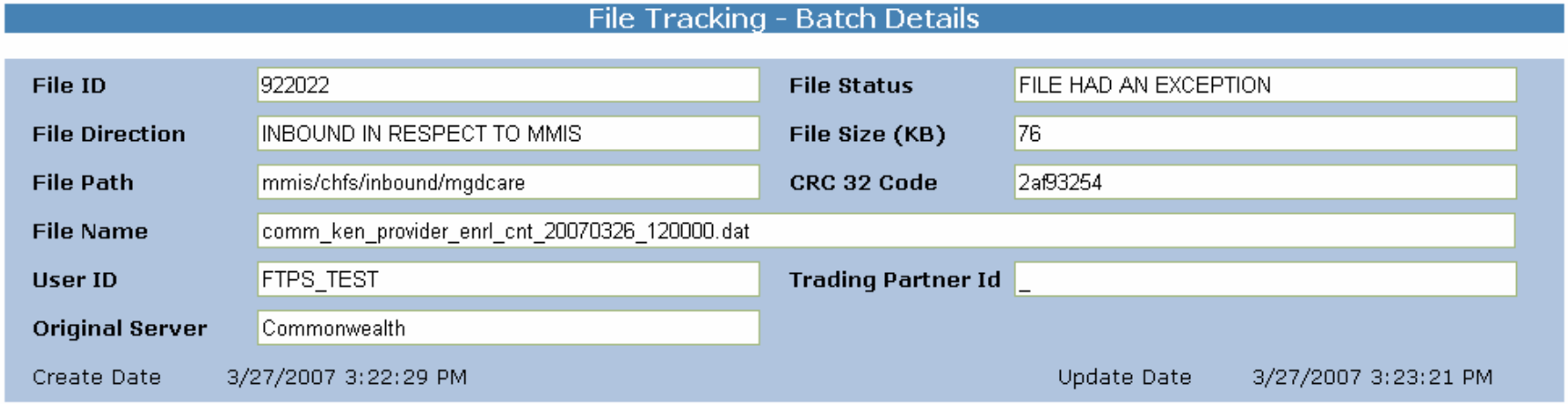
6.2.1.3.2 Technical Name:

EDIUtilities.FileTrackingDetails

6.2.1.3.3 Web Page Name:

File Tracking Details

6.2.1.3.4 File Tracking Batch Details – Layout



6.2.1.3.5 File Tracking Batch Details – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File ID	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_File
File Status	The current status of the file	Text Box	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
File Size (KB)	The size of the file	Text Box	NUMBER	11	T_FILE_TRACK	NUM_FILESIZE
File Path	The complete path to the location of the file	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_FILEPATH
CRC 32 Code	The Cyclic Redundancy Check (CRC) used to ensure no intermediate changes were made to the file	Text Box	VARCHAR2	8	T_FILE_ TRACK	CDE_CRC32
File Name	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_FILE
User ID	The User ID for FTP and FTPS transfers	Text Box	VARCHAR2	50	T_FILE_TRACK	ID_USER
Trading Partner ID	The ID of the Trading Partner associated with this file	Text Box	NUMBER	9	T_TP	SAK_TP
Original Server	The name of the server the file originated from	Text Box	VARCHAR2	50	T_FILE_TRACK	NAM_ORGN_ SERVER
Create Date	The date the file was initially created	Label	DATE		T_FILE_ TRACK	DTE_CREATED
Update Date	The date the file was last updated	Label	DATE		T_FILE_ TRACK	DTE_LAST_ UPDATED

6.2.1.3.6 File Tracking Solicited Details – Layout

File Tracking - Solicited Details

File ID	903290	**File Status**	PROCESSING COMPLETED
File Direction	INTERACTIVE	**File Size (B)**	525
Port Instance	0	**IP Address**	0
User Id	akyxix_NRM	**Trading Partner Id**	000000001
Original Server	USOLWKYVM078		

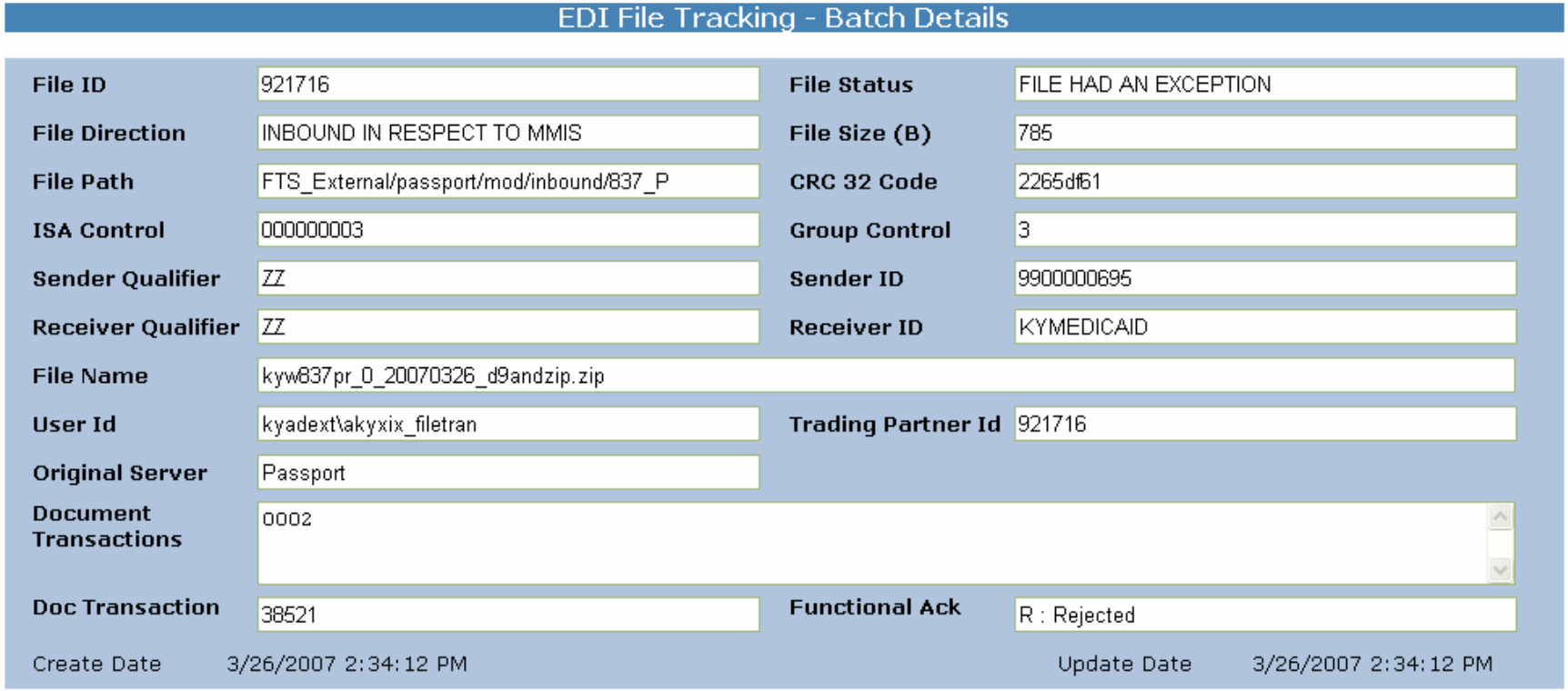
Create Date 3/1/2007 10:31:45 AM Update Date 3/1/2007 10:31:52 AM

6.2.1.3.7 File Tracking Solicited Details – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File ID	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_File
File Status	The current status of the file	Text Box	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS
File Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
File Size (KB)	The size of the file	Text Box	NUMBER	11	T_FILE_TRACK	NUM_FILESIZE
Port Instance	The single instance of the available defined port	Text Box	NUMBER	11	T_FILE_ SOLICITED_ TRACK	NUM_ INBOUND_ PORT_ INSTANCE
IP Address	The destination IP Address for the transfer of the file	Text Box	VARCHAR2	20	T_FILE_ SOLICITED_ TRACK	ADR_IP
User ID	The User ID for FTP and FTPS transfers	Text Box	VARCHAR2	50	T_FILE_TRACK	ID_USER

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Trading Partner ID	The ID of the Trading Partner associated with this file	Text Box	NUMBER	9	T_TP	SAK_TP
Original Server	The name of the server the file originated from	Text Box	VARCHAR2	50	T_FILE_TRACK	NAM_ORGN_ SERVER
Create Date	The date the file was initially created	Label	DATE		T_FILE_ TRACK	DTE_CREATED
Update Date	The date the file was last updated	Label	DATE		T_FILE_ TRACK	DTE_LAST_ UPDATED

6.2.1.3.8 EDI File Tracking Batch Details – Layout



6.2.1.3.9 EDI File Tracking Batch Details – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File ID	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_File
File Status	The current status of the file	Text Box	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS
File Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
File Size (KB)	The size of the file	Text Box	NUMBER	11	T_FILE_TRACK	NUM_FILESIZE
File Path	The complete path to the location of the file	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_FILEPATH
CRC 32 Code	The Cyclic Redundancy Check (CRC) used to ensure no intermediate changes were made to the file	Text Box	VARCHAR2	8	T_FILE_ TRACK	CDE_CRC32
ISA Control	Unique ISA Segment number	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRACK	NUM_ISA_ CNTRL
Group Control	Unique Group Segment number	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRACK	NUM_GROUP_ CNTRL
Sender Qualifier	Qualifier describing the Sender ID	Text Box	VARCHAR2	2	T_ DOCUMENT_ TRACK	ID_SND_ QUALIFY
Sender ID	The ID of the sending entity	Text Box	VARCHAR2	15	T_ DOCUMENT_ TRACK	ID_SND
Receiver Qualifier	Qualifier describing the Receiver ID	Text Box	VARCHAR2	2	T_ DOCUMENT_ TRACK	ID_RECV_ QUALIFY

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Receiver ID	The ID of the receiving entity	Text Box	VARCHAR2	15	T_ DOCUMENT_ TRACK	ID_RECV
File Name	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_FILE
User ID	The User ID for FTP and FTPS transfers	Text Box	VARCHAR2	50	T_FILE_TRACK	ID_USER
Trading Partner ID	The ID of the Trading Partner associated with this file	Text Box	NUMBER	9	T_TP	SAK_TP
Original Server	The name of the server the file originated from	Text Box	VARCHAR2	50	T_FILE_TRACK	NAM_ORGN_ SERVER
Document Transactions	Control Numbers for multiple transactions	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRANS-ACTIONS	NUM_ TRANSACTION_ CNTRL
Doc Transaction	The identification number of the document transaction	Text Box	NUMBER	9	T_ DOCUMENT_ TRACK	SAK_ DOCUMENT_ TRACK
Functional Ack	Acknowledgement Status	Text Box	CHAR	1	T_ FUNCTIONAL_ ACKS	CDE_ FUNCTIONAL_ STATUS
Create Date	The date the file was initially created	Label	DATE		T_FILE_ TRACK	DTE_CREATED
Update Date	The date the file was last updated	Label	DATE		T_FILE_ TRACK	DTE_LAST_ UPDATED

6.2.1.3.10 EDI File Tracking Solicited Details – Layout

EDI File Tracking - Solicited Details

File ID	920718	**File Status**	PROCESSING COMPLETED
File Direction	INTERACTIVE	**File Size (B)**	435
ISA Control	900000007	**Group Control**	900000007
Sender Qualifier	ZZ	**Sender ID**	64112949
Receiver Qualifier	ZZ	**Receiver ID**	KYMEDICAID
User Id	akyxix_NRM	**Trading Partner Id**	900000007
Original Server	USOLWKYVM078		
Port Instance	1	**IP Address**	0
Document Transactions	900000007		
Doc Transaction		**Functional Ack**	

Create Date 3/25/2007 9:24:02 AM Update Date 3/25/2007 9:24:02 AM

6.2.1.3.11 EDI File Tracking Solicited Details – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File ID	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_File
File Status	The current status of the file	Text Box	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS
File Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
File Size (KB)	The size of the file	Text Box	NUMBER	11	T_FILE_TRACK	NUM_FILESIZE

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
ISA Control	Unique ISA Segment number	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRACK	NUM_ISA_ CNTRL
Group Control	Unique Group Segment number	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRACK	NUM_GROUP_ CNTRL
Sender Qualifier	Qualifier describing the Sender ID	Text Box	VARCHAR2	2	T_ DOCUMENT_ TRACK	ID_SND_ QUALIFY
Sender ID	The ID of the sending entity	Text Box	VARCHAR2	15	T_ DOCUMENT_ TRACK	ID_SND
Receiver Qualifier	Qualifier describing the Receiver ID	Text Box	VARCHAR2	2	T_ DOCUMENT_ TRACK	ID_RECV_ QUALIFY
Receiver ID	The ID of the receiving entity	Text Box	VARCHAR2	15	T_ DOCUMENT_ TRACK	ID_RECV
User ID	The User ID for FTP and FTPS transfers	Text Box	VARCHAR2	50	T_FILE_TRACK	ID_USER
Trading Partner ID	The ID of the Trading Partner associated with this file	Text Box	NUMBER	9	T_TP	SAK_TP
Original Server	The name of the server the file originated from	Text Box	VARCHAR2	50	T_FILE_TRACK	NAM_ORGN_ SERVER
Port Instance	The single instance of the available defined port	Text Box	NUMBER	11	T_FILE_ SOLICITED_ TRACK	NUM_ INBOUND_ PORT_ INSTANCE

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
IP Address	The destination IP Address for the transfer of the file	Text Box	VARCHAR2	20	T_FILE_ SOLICITED_ TRACK	ADR_IP
Document Transactions	Control Numbers for multiple transactions	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRANS-ACTIONS	NUM_ TRANSACTION_ CNTRL
Doc Transaction	The identification number of the document transaction	Text Box	NUMBER	9	T_ DOCUMENT_ TRACK	SAK_ DOCUMENT_ TRACK
Functional Ack	Acknowledgement Status	Text Box	CHAR	1	T_ FUNCTIONAL_ ACKS	CDE_ FUNCTIONAL_ STATUS
Create Date	The date the file was initially created	Label	DATE		T_FILE_ TRACK	DTE_CREATED
Update Date	The date the file was last updated	Label	DATE		T_FILE_ TRACK	DTE_LAST_ UPDATED

6.2.1.3.12 File Tracking Details Correlation Batch File - Layout

Correlation File

EDI File Tracking - Batch Details

File ID	921717	**File Status**	PROCESSING COMPLETED
File Direction	OUTBOUND IN RESPECT TO MMIS	**File Size (B)**	285
File Path	\\10.40.13.12\FTS\Passport\ACKs\Outbound\X12	**CRC 32 Code**	e2284c97
ISA Control	000921716	**Group Control**	1
Sender Qualifier	ZZ	**Sender ID**	KYMEDICAID
Receiver Qualifier	ZZ	**Receiver ID**	9900000695
File Name	921716_837PEX12BATCH_497816_1of1_KYW837PR_0_20070326_D9ANDZIP.txt.997.dat		
Document Transactions	0001		
Doc Transaction		**Functional Ack**	

Create Date 3/26/2007 2:34:14 PM Update Date 3/26/2007 2:34:14 PM

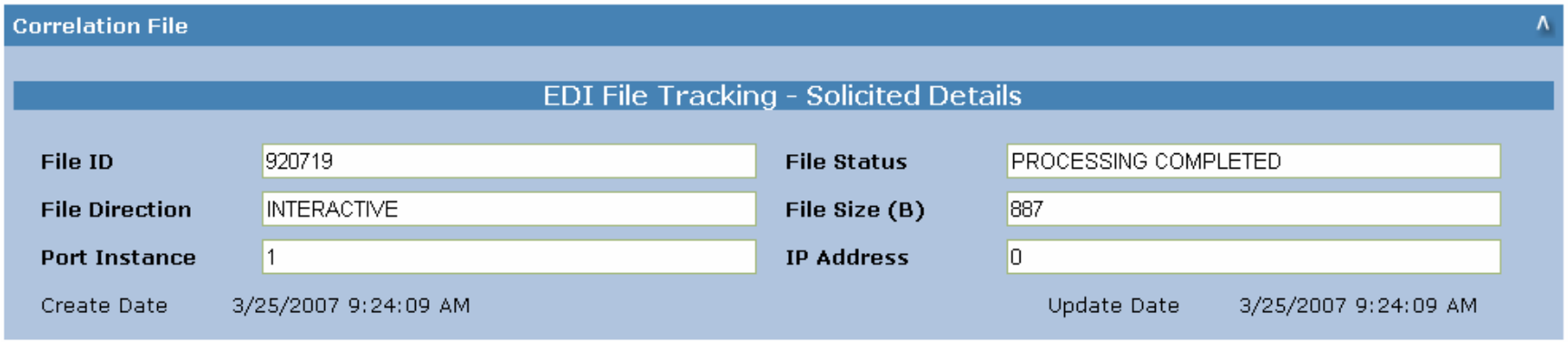
6.2.1.3.13 File Tracking Details Correlation Batch File – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File ID	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_File
File Status	The current status of the file	Text Box	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS
File Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
File Size (B)	The size of the file	Text Box	NUMBER	11	T_FILE_TRACK	NUM_FILESIZE

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File Path	The complete path to the location of the file	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_FILEPATH
CRC 32 Code	The Cyclic Redundancy Check (CRC) used to ensure no intermediate changes were made to the file	Text Box	VARCHAR2	8	T_FILE_ TRACK	CDE_CRC32
ISA Control	Unique ISA Segment number	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRACK	NUM_ISA_ CNTRL
Group Control	Unique Group Segment number	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRACK	NUM_GROUP_ CNTRL
Sender Qualifier	Qualifier describing the Sender ID	Text Box	VARCHAR2	2	T_ DOCUMENT_ TRACK	ID_SND_ QUALIFY
Sender ID	The ID of the sending entity	Text Box	VARCHAR2	15	T_ DOCUMENT_ TRACK	ID_SND
Receiver Qualifier	Qualifier describing the Receiver ID	Text Box	VARCHAR2	2	T_ DOCUMENT_ TRACK	ID_RECV_ QUALIFY
Receiver ID	The ID of the receiving entity	Text Box	VARCHAR2	15	T_ DOCUMENT_ TRACK	ID_RECV
Document Transactions	Control Numbers for multiple transactions	Text Box	VARCHAR2	9	T_ DOCUMENT_ TRANS-ACTIONS	NUM_ TRANSACTION_ CNTRL

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Doc Transaction	The identification number of the document transaction	Text Box	NUMBER	9	T_ DOCUMENT_ TRACK	SAK_ DOCUMENT_ TRACK
Functional Ack	Acknowledgement Status	Text Box	CHAR	1	T_ FUNCTIONAL_ ACKS	CDE_ FUNCTIONAL_ STATUS
Create Date	The date the file was initially created	Label	DATE		T_FILE_ TRACK	DTE_CREATED
Update Date	The date the file was last updated	Label	DATE		T_FILE_ TRACK	DTE_LAST_ UPDATED

6.2.1.3.14 File Tracking Details Correlation Solicited File – Layout



6.2.1.3.15 File Tracking Details Correlation Solicited File – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File ID	The complete name of the file including the extension	Text Box	VARCHAR2	255	T_FILE_TRACK	NAM_File
File Status	The current status of the file	Text Box	VARCHAR2	255	T_CDE_FILE_ STATUS	DSC_FILE_ STATUS

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
File Direction	The direction of the file with respect to MMIS	Text Box	VARCHAR2	255	T_CDE_FILE_ DIRECTION	DSC_FILE_ DIRECTION
File Size (KB)	The size of the file	Text Box	NUMBER	11	T_FILE_TRACK	NUM_FILESIZE
Port Instance	The single instance of the available defined port	Text Box	NUMBER	11	T_FILE_ SOLICITED_ TRACK	NUM_ INBOUND_ PORT_ INSTANCE
IP Address	The destination IP Address for the transfer of the file	Text Box	VARCHAR2	20	T_FILE_ SOLICITED_ TRACK	ADR_IP

6.2.1.3.16 File Tracking Details Event - Layout

Events

Event	Event Comments	Create Dte
RECEIVED_PREPROCESSOR	\\10.40.13.12\FTS\Passport\X12\Inbound	3/26/2007 2:31:59 PM
BALANCING_COUNT	1	3/26/2007 2:31:59 PM

Event Key	Event Value	Event Key Comments	Create Dt
COUNT_BEFORE_VALIDATION	1		3/26/2007 2:31:59 PM
COUNT_AFTER_VALIDATION	0		3/26/2007 2:32:00 PM

RENAMED	921716_837PEX12BATCH_497816_1of1_KYW837PR_0_20070326_D9ANDZIP.txt	3/26/2007 2:31:59 PM
HIPAA_VALIDATION_ERROR		3/26/2007 2:32:00 PM
KPI	0	3/26/2007 2:32:00 PM
PASSED_TO_FAILEDFOLDER	\\10.40.13.12\FTS\FAILED\	3/26/2007 2:32:00 PM
RENAMED_ACK	921717_921716_837PEX12BATCH_KYW837PR_0_20070326_D9ANDZIP.dat.997.dat	3/26/2007 2:32:02 PM
CORRELATED_FUNCTIONAL_ACK	Response File ID - 921717	3/26/2007 2:32:02 PM

6.2.1.3.17 File Tracking Details Events – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Event	The event code	Label	VARCHAR2	50	T_CDE_EVENT	CDE_EVENT
Event Comments	Comments about the event	Label	VARCHAR2	250	T_EVENT_ TRACK	DSC_EVENT_ TRACK

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Create Dte	The date this event was created	Label	TIMESTAMP	6	T_EVENT_ TRACK	DTE_CREATED
Event Key	The value of the event key	Label	VARCHAR2	50	T_CDE_ EVENT_KEY	CDE_EVENT_ KEY
Event Value	The value of the event this key is associated.	Label	VARCHAR2	255	T_EVENT_ TRACK_KEYS	TXT_EVENT_ KEY_VALUE
Event Key Comments	Comments about the event key	Label	VARCHAR2	255	T_EVENT_ TRACK_KEYS	DSC_EVENT_ TRACK_KEYS
Create Dt	The date this event key was created	Label	TIMESTAMP	6	T_EVENT_ TRACK_KEYS	DTE_CREATED

6.2.1.3.18 File Tracking Details Exceptions – Layout

Exceptions

Assembly	Method	Exception Dt	Message	Class	Server
FTSProcessor	RenameUncompressA...	3/27/2007 6:55:04 PM	File exists exception has occurred.	BaseProcessor	FILE-01-MO

6.2.1.3.19 File Tracking Details Exceptions – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Assembly	The name of the assembly that the exception originated from	Label	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	NAM_ASSEMBLY
Method	The name of the specific method that threw the exception	Label	VARCHAR2	500	T_TRACK_ EXCEPTION_ HANDLE	NAM_METHOD
Exception Date	The date the exception was created	Label	DATE		T_TRACK_ EXCEPTION_ HANDLE	DTE_ EXCEPTION

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Message	The inner most message associated with the exception	Label (Selectable Link)	VARCHAR2	1000	T_TRACK_ EXCEPTION_ HANDLE	TXT_MESSAGE
Class	The name of the class that the exception originated	Label	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	NAM_CLASS
Server	The name of the machine which the exception occurred	Label	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	NAM_MACHINE

6.2.1.3.20 File Tracking Details Exception Details – Layout

File Tracking - Exception Details

Previous Record | Next Record

Exception Message	File exists exception has occurred.		
Exception Type	EDS.USGS.MED.EDI.ExceptionManagement.StandardException		
Server Name	FILE-01-MO	Assembly Name	FTSProcessor
Class Name	BaseProcessor	Method Name	RenameUncompressAndMove
Process ID	46124	Thread ID	46004
Exception Date	3/27/2007 6:55:04 PM	Batch ID	79471bc6-8849-47c8-acd8-dbf3e65ec0f4
Stack Trace			

6.2.1.3.21 File Tracking Details Exception Details – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Previous Record	If more than one exception exists, this control will display the previous exception	Button				
Next Record	If more than one exception exists, this control will display the next exception	Button				
Exception Message	The inner most message associated with the exception	Text Box	VARCHAR2	1000	T_TRACK_ EXCEPTION_ HANDLE	TXT_MESSAGE
Exception Type	The full name of the exception type	Text Box	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	TXT_ EXCEPTION_ TYPE
Server Name	The name of the machine on which the exception occurred	Text Box	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	NAM_MACHINE
Assembly Name	The name of the assembly that the exception originated from	Text Box	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	NAM_ASSEMBLY
Class Name	The name of the class that the exception originated from.	Text Box	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	NAM_CLASS
Method Name	The name of the specific method that threw the exception	Text Box	VARCHAR2	500	T_TRACK_ EXCEPTION_ HANDLE	NAM_METHOD
Process ID	The process ID that the exception originated from	Text Box	VARCHAR2	10	T_TRACK_ EXCEPTION_ HANDLE	ID_PROCESS

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Thread ID	The ID of the executing thread that found the exception	Text Box	VARCHAR2	100	T_TRACK_ EXCEPTION_ HANDLE	NAM_THREAD
Exception Date	The date the exception was created	Text Box	DATE		T_TRACK_ EXCEPTION_ HANDLE	DTE_ EXCEPTION
Batch ID	The ID of a group of exceptions	Text Box	VARCHAR2	50	T_TRACK_ EXCEPTION_ HANDLE	ID_BATCH
Stack Trace	A printout of the stack trace at the time the exception occurred	Text Box	VARCHAR2	2000	T_TRACK_ EXCEPTION_ HANDLE	TXT_STACK_ TRACE

6.2.1.3.22 File Tracking Details Notifications – Layout

Notifications

To	From	Subject	Date
manish.sethi-eds@eds.com	manish.sethi-eds@eds.com	File already exists.	3/27/2007 6:55:04 PM

6.2.1.3.23 File Tracking Details Notifications – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
To	The originating address of the notification	Label	VARCHAR2	200	T_TRACK_ NOTIFICATION	ADR_ NOTIFICATION_ TO
From	The recipient address of the notification	Label	VARCHAR2	200	T_TRACK_ NOTIFICATION	ADR_ NOTIFICATION_ FROM
Subject	The subject of the notification	Label	VARCHAR2	255	T_TRACK_ NOTIFICATION	TXT_ NOTIFICATION_

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
						SUBJECT
Date	The date the notification was created	Label	DATE		T_TRACK_ NOTIFICATION	DTE_ NOTIFICATION

6.2.2 Trading Partner Management

Trading Partner management is used to Add, Delete, and modify trading partners. The specific attributes of a trading partner can be modified, along with the types of allowable documents for each trading partner.

6.2.2.1 Page: Trading Partner – Search

6.2.2.2 Description:

The trading partner search page can be used to search for any trading partner within the system.

6.2.2.2.1 Technical Name:

EDIUtilities.TradingPartner

6.2.2.2.2 Web Page Name:

Trading Partners

6.2.2.2.3 Trading Partner Search – Layout

Trading Partner Search Page

Name		TP ID	
Provider ID		NPI ID	
Type	PROVIDER	Status	NOT SELECTED

Search Clear

Records Found 36741

Name	TP ID	Status	Provider ID	NPI	Contact	Phone
	9900009032	ACTIVE	9900009032	9900009032		
CHRISTIAN CARE COMMUNITIE...	33900382	ACTIVE	33900382			(339)003-820...
COMMUNITY PHARMACY CARE ...	90230194	ACTIVE	90230194			9023019400
DAWSON POINTE NURSING HOME	9900000991	ACTIVE	9900000991	9900000991	JAY FRANCES	(270)797-202...
DISCOUNT PHARM ...	54005434	ACTIVE	54005434			(540)054-340...
DYERSBURG DIALYSIS ...	39000104	ACTIVE	39000104			(390)001-040...
IPS ...	37000577	ACTIVE	37000577			(370)005-770...
MALLORY S ROBERTS ...	45002565	ACTIVE	45002565			(450)025-650...
SILVERTON HOME TRAIDIALYSIS ...	39000385	ACTIVE	39000385			(390)003-850...
THERAPY CONCEPTS INC ...	90272220	ACTIVE	90272220			9027222000

1 2 3 4 5 6 7 8 9 10 ...

6.2.2.2.4 Trading Partner Search – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Name	The name of the Trading Partner	Text Box	VARCHAR2	50	T_TP	NAM_TP
TP ID	The ID of the Trading Partner	Text Box	NUMBER	9	T_TP	SAK_TP
Provider ID	The ID of the Trading Partner in the MMIS system	Text Box	CHAR	15	T_TP	ID_PROVIDER
NPI ID	The National Provider Identification of the Trading Partner	Text Box	VARCHAR2	50	T_TP	CDE_NPI
Type	The type of the Trading Partner	Drop Down Box	VARCHAR2	50	T_CDE_TP_ TYPE	CDE_TP_ TYPE
Status	The current status of the Trading Partner	Drop Down Box	VARCHAR2	50	T_CDE_TP_ STATUS	CDE_TP_ STATUS
Contact	The name of an individual or group associated with this Trading Partner	Label	VARCHAR2	60	T_TP	NAM_ CONTACT
Phone	The contact phone number for this Trading Partner	Label	VARCHAR2	15	T_TP	NUM_PHONE

6.2.2.3 Page: Trading Partner Management

6.2.2.3.1 Description:

The trading partner management page is used to update both existing and recently created trading partners.

6.2.2.3.2 Technical Name:

EDIUtilities.TradingPartnerMgmt

6.2.2.3.3 Web Page Name:

Trading Partner Management

6.2.2.3.4 Trading Partner Management – Layout

Trading Partner Management

* required information

Name*	THERAPY CONCEPTS INC	**NPI***	
Provider ID*	90272220	**ISA Identifier***	90272220
GSApplication code*		**Sequence Number***	1
TP Type*	PROVIDER	**TP Status Type***	ACTIVE
Status Effective Date*	1/1/1900	**Authorization Qualifier***	00
Address*	1093B REED HATRMAN HWY	**Authorization Code**	
City*	CINCINNATI	**Authorization Date**	1/1/1900
State*	OH	**SecurityQualifier***	00
ZipCode*	45242	**Security code**	
Country		**VAN Indicator***	N
Contact		**ISA ID Qualifier***	ZZ Mutually Defined
Phone	(902)722-2000	**Agreement Mailed Date**	1/1/1900
Phone Extn	0000	**Agreement Received Date**	1/1/1900
Fax	9027222000	**Component Separator***	: Colon
Email		**Segment Terminator***	~ Tilde

Allowed Transport

Available Transport: STORAGE MEDIA, SFTP, UNC

Add / Remove

Selected Transport: WEB

Notes

Create Date :11/28/2006 12:00:00 AM

Update Date :11/28/2006 12:00:00 AM

Save Configure Delete

6.2.2.3.5 Trading Partner Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Name	The name of the Trading Partner	Text Box	VARCHAR2	50	T_TP	NAM_TP
NPI	The National Provider Identification of the Trading Partner	Text Box	VARCHAR2	50	T_TP	CDE_NPI
Provider ID	The ID of the Trading Partner in the MMIS system	Text Box	CHAR	15	T_TP	ID_PROVIDER
ISA Identifier	The Trading Partner ID	Text Box	VARCHAR2	15	T_TP	ID_ISA_ IDENTIFIER
GSApplication Code	The Group Application Segment	Text Box	VARCHAR2	15	T_TP	CDE_GS_ APPLICATION
Sequence Number	Internal identification number for the Trading Partner	Text Box	NUMBER	11	T_TP	NUM_SEQ
TP Type	The type of the Trading Partner	Drop Down Box	VARCHAR2	50	T_CDE_TP_ TYPE	CDE_TP_ TYPE
TP Status Type	The current status of the Trading Partner	Drop Down Box	VARCHAR2	50	T_CDE_TP_ STATUS	CDE_TP_ STATUS
Status Effective Date	The date the Trading Partner becomes active	Text Box	DATE		T_TP	DTE_STATUS_ EFFECTIVE_ DATE
Address	The street address of the Trading Partner	Text Box	VARCHAR2	55	T_TP	ADR_STREET_ 1, ADR_STREET_ 2
City	The city of the Trading Partner	Text Box	VARCHAR2	30	T_TP	ADR_CITY
State	The state of the Trading Partner	Drop Down Box	CHAR	2	T_TP	ADR_STATE
Zip Code	The zip code of the Trading Partner	Text Box	VARCHAR2	15	T_TP	ADR_ZIP

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Country	The country of the Trading Partner	Text Box	CHAR	3	T_TP	CDE_ COUNTRY
Contact	The name of an individual or group associated with this Trading Partner	Text Box	VARCHAR2	60	T_TP	NAM_ CONTACT
Phone	The contact phone number for this Trading Partner	Text Box	VARCHAR2	15	T_TP	NUM_PHONE
Phone Ext	The phone number extension for this Trading Partner	Text Box	VARCHAR2	10	T_TP	NUM_PHONE_ EXT
Fax	The fax number for this Trading Partner	Text Box	VARCHAR2	15	T_TP	NUM_FAX
Email	The contact email address of this Trading Partner	Text Box	VARCHAR2	100	T_TP	ADR_EMAIL
Authorization Qualifier	Used to qualify the Authorization Code	Drop Down Box	VARCHAR2	2	T_TP	CDE_ AUTHORIZATION_ QUALIFIER
Authorization Code	The Trading Partner's Authorization Code	Text Box	VARCHAR2	10	T_TP	CDE_ AUTHORIZATION
Authorization Date	The date this Trading Partner was authorized	Text Box	DATE		T_TP	DTE_ AUTHORIZATION
Security Qualifier	Used to qualify the Security Code	Drop Down Box	VARCHAR2	2	T_TP	CDE_SECURITY_ QUALIFIER
Security Code	The Trading Partner's Security Code	Text Box	VARCHAR2	10	T_TP	CDE_SECURITY
VAN Indicator	The VAN Indicator of the Trading Partner	Drop Down Box	CHAR	1	T_TP	IND_VAN
ISA ID Qualifier	Used to qualify the Trading Partner's ID	Drop Down Box	VARCHAR2	2	T_TP	ID_ISA_ IDENTIFIER_ QUALIFIER

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Agreement Mailed Date	The date agreement notification was mailed	Text Box	DATE		T_TP	DTE_ AGREEMENT_ MAILED
Agreement Received Date	The date agreement notification was received	Text Box	DATE		T_TP	DTE_ AGREEMENT_ RECV
Component Separator	Document Separator	Drop Down Box	VARCHAR2	10	T_TP	TXT_ COMPONENT_ SEPARATOR
Segment Terminator	Document Terminator	Drop Down Box	VARCHAR2	10	T_TP	TXT_SEGMENT_ SEPARATOR
Allowed Transport	The transportation protocols this trading partner can use to send and receive files	Selection Control				
Notes	Comments regarding this Trading Partner	Text Box	VARCHAR2	CLOB	T_TP	TXT_NOTES
Create Date	The date this Trading Partner was created	Text Box	DATE		T_TP	DTE_CREATED
Update Date	The date this Trading Partner was last updated	Text Box	DATE		T_TP	DTE_LAST_ UPDATED

6.2.2.4 Page: Trading Partner Configuration

6.2.2.4.1 Description:

The Trading Partner Configuration page is used to specify the criteria in which certain document types can be processed for each trading partner. Trading Partner configurations can be added, deleted, and updated for a specific trading partner.

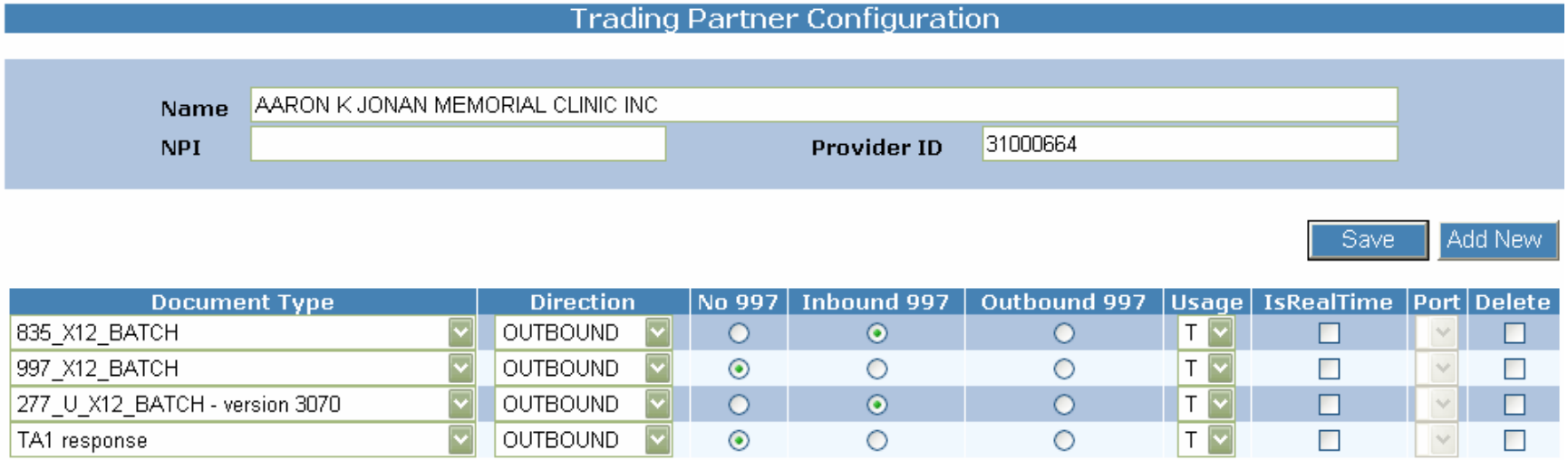
6.2.2.4.2 Technical Name:

EDIUtilities.TradingPartnerConfiguration

6.2.2.4.3 Web Page Name:

Trading Partner Configuration

6.2.2.4.4 Trading Partner Configuration – Layout



6.2.2.4.5 Trading Partner Configuration – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Name	The name of the Trading Partner	Text Box	VARCHAR2	50	T_TP	NAM_TP
NPI	The National Provider Identification of the Trading Partner	Text Box	VARCHAR2	50	T_TP	CDE_NPI

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Provider ID	The ID of the Trading Partner in the MMIS system	Text Box	CHAR	15	T_TP	ID_PROVIDER
Document Type	The file type for this configuration	Drop Down Box	VARCHAR2	15	T_CDE_ SUPPORTED_ DOC	CDE_ IDENTIFICATION
Direction	The direction of the selected document type	Drop Down Box	VARCHAR2	50	T_CDE_ FILE_ DIRECTION	CDE_FILE_ DIRECTION
No 997	The configuration does not send or receive the 997 document type	Radio Control				
Inbound 997	The configuration can receive the 997 document type	Radio Control	CHAR	1	T_TP_CONFIG	IND_EXPECT_ INBOUND_997
Outbound 997	The configuration can send the 997 document type	Radio Control	CHAR	1	T_TP_CONFIG	IND_SENT_ OUTBOUND_ 997
Usage	The current usage code for the configuration. Possible values are “T” or “P”	Drop Down Box	CHAR	1	T_TP_CONFIG	CDE_USAGE
IsRealTime		Check Box				
Port	The port chosen for the configuration	Drop Down Box	CHAR	1	T_PORT	CDE_PORT
Delete	Checking this control will delete the configuration	Check Box				

6.2.2.5 Page: Trading Partner Add

6.2.2.5.1 Description:

The Trading Partner Add page is used to create new trading partners. After the trading partner is initially created, it can be updated normally through the Trading Partner Modification page.

6.2.2.5.2 Technical Name:

EDIUtilities.TPPreAdd

6.2.2.5.3 Web Page Name:

Trading Partner Add

6.2.2.5.4 Trading Partner Add – Layout

6.2.2.5.5 Trading Partner Add – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Name	The name of the Trading Partner	Text Box	VARCHAR2	50	T_TP	NAM_TP
ISA Identifier	The Trading Partner ID	Text Box	VARCHAR2	15	T_TP	ID_ISA_ IDENTIFIER
GSApplication Code	The Group Application Segment	Text Box	VARCHAR2	15	T_TP	CDE_GS_ APPLICATION
Provider ID	The ID of the Trading Partner in the MMIS system	Text Box	CHAR	15	T_TP	ID_PROVIDER
NPI ID	The National Provider Identification of the Trading Partner	Text Box	VARCHAR2	50	T_TP	CDE_NPI

6.2.3 Maintenance

The maintenance area of the application includes miscellaneous utilities that assist in Trading Partner management. The maintenance section also includes the FTS configuration application.

6.2.3.1 Page: Event Codes

6.2.3.1.1 Description:

This Event Codes page displays the current event codes in the system along with their descriptions.

6.2.3.1.2 Technical Name:

EDIUtilities.Event

6.2.3.1.3 Web Page Name:

Events

6.2.3.1.4 Event Codes – Layout

Event Codes

Create a New Event Code

Event Code	Event Code Description
VALID_EDI_FILE	VALID EDI FILE
FA_FILE_DELETED	FA FILE DELETED
DIFFERENT_CRC32CODE	DIFFERENT CRC32CODE
PASSED_TO_EDIFECS	PASSED TO EDIFECS
PASSED_TO_BIZTALK	PASSED TO BIZTALK
RECEIVED_PREPROCESSOR	RECEIVED PREPROCESSOR
PROCESSING_PREPROCESSOR	PROCESSING PREPROCESSOR
PASSED_TO_PREPROCESSOR	PASSED TO PREPROCESSOR

1 2 3 4 5 6 7

Create a New Event Code

6.2.3.1.5 Event Codes – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Event Code	The identification code for the event	Label (Selectable Link)	VARCHAR2	50	T_CDE_EVENT	CDE_EVENT

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Event Code Description	The description of the event	Label	VARCHAR2	255	T_CDE_EVENT	DSC_EVENT

6.2.3.2 Page: Event Code Management

6.2.3.2.1 Description:

The Event Code Management page allows event codes to be updated and created.

6.2.3.2.2 Technical Name:

EDIUtilities.EventMgmt

6.2.3.2.3 Web Page Name:

Event Code Management

6.2.3.2.4 Event Code Management – Layout

6.2.3.3 Event Code Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Event Code	The identification code for the event	Text Box	VARCHAR2	50	T_CDE_EVENT	CDE_EVENT
Event Code Description	The description of the event	Text Box	VARCHAR2	255	T_CDE_EVENT	DSC_EVENT

6.2.3.4 Page: Event Keys

6.2.3.4.1 Description:

This Event Keys page displays the current event keys in the system along with their descriptions.

6.2.3.4.2 Technical Name:

EDIUtilities.EventKey

6.2.3.4.3 Web Page Name:

Event Keys

6.2.3.4.4 Event Keys – Layout

Event Keys

Create a New Event Key

Event Key	Event Key Description
BILLED_AMOUNT	BILLED_AMOUNT
COUNT_AFTER_VALIDATION	COUNT_AFTER_VALIDATION
COUNT_BEFORE_VALIDATION	COUNT_BEFORE_VALIDATION
PROVIDER_PAYMENT_AMOUNT	PROVIDER_PAYMENT_AMOUNT
COUNT_BEFORE_SENDING_TO_IC	COUNT_BEFORE_SENDING_TO_IC
CERTIFIED_COUNT	CERTIFIED_COUNT
PREMIUM_PAYMENT_AMOUNT	PREMIUM_PAYMENT_AMOUNT
INSURED_MEMBER_COUNT	INSURED_MEMBER_COUNT

1 2

Create a New Event Key

6.2.3.4.5 Event Keys – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Event Key	The identification code of the event key	Label (Selectable Link)	VARCHAR2	50	T_CDE_EVENT_KEY	CDE_EVENT_KEY
Event Key Description	The description of the event key	Label	VARCHAR2	255	T_CDE_EVENT_KEY	DSC_EVENT_KEY

6.2.3.5 Page: Event Key Management

6.2.3.5.1 Description:

The Event Key Management page allows event keys to be updated and created.

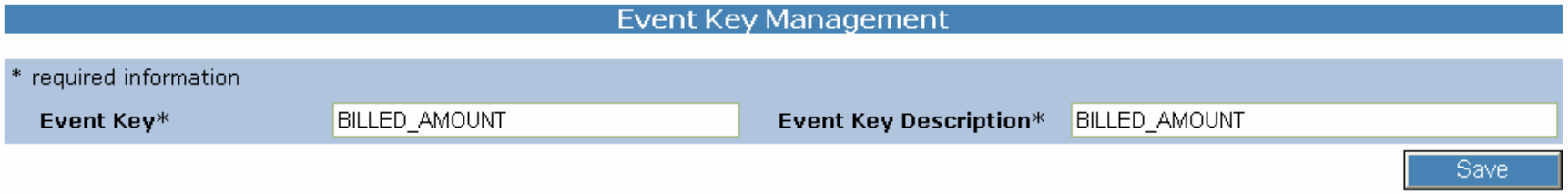
6.2.3.5.2 Technical Name:

EDIUtilities.EventKeyMgmt

6.2.3.5.3 Web Page Name:

Event Key Management

6.2.3.5.4 Event Key Management – Layout



6.2.3.5.5 Event Key Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Event Key	The identification code of the event key	Label (Selectable Link)	VARCHAR2	50	T_CDE_EVENT_KEY	CDE_EVENT_KEY
Event Key Description	The description of the event key	Label	VARCHAR2	255	T_CDE_EVENT_KEY	DSC_EVENT_KEY

6.2.3.6 Page: Ports

6.2.3.6.1 Description:

The Ports page displays the current ports in the system along with their descriptions.

6.2.3.6.2 Technical Name:

EDIUtilities.Port

6.2.3.6.3 Web Page Name:

Port

6.2.3.6.4 Ports – Layout

Ports

Create a New Port

Port Number	Type	TP Name	ISA Identifier
8088	UNASSIGNED	Suzanne Test ID	9900002003
9099	UNASSIGNED	Leonard Test ID	9900002001
8088	TEST	Donna Test	9900000003

Create a New Port

6.2.3.6.5 Ports – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Port Number	The actual port number	Label (Selectable Link)	NUMBER	9	T_PORT	NUM_PORT
Type	The current type of the port	Label	VARCHAR2	255	T_CDE_PORT	DSC_PORT
TP Name	The name of the Trading Partner	Label	VARCHAR2	50	T_TP	NAM_TP
ISA Identifier	The Trading Partner ID	Label	VARCHAR2	15	T_TP	ID_ISA_ IDENTIFIER

6.2.3.7 Page: Port Management

6.2.3.7.1 Description:

The Event Key Management page allows event keys to be updated and created.

6.2.3.7.2 Technical Name:

EDIUtilities.PortMgmt

6.2.3.7.3 Web Page Name:

Port Management

6.2.3.7.4 Port Management – Layout

6.2.3.7.5 Port Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Port Number	The actual port number	Text Box	NUMBER	9	T_PORT	NUM_PORT
Port Type	The current type of the port	Text Box	VARCHAR2	255	T_CDE_PORT	DSC_PORT

6.2.3.8 Page: Supported Documents

6.2.3.8.1 Description:
This page provides for the management of all current supported document types within the system.

6.2.3.8.2 Technical Name:
EDIUtilities.SupportedDoc

6.2.3.8.3 Web Page Name:
Supported Documents

6.2.3.8.4 Supported Documents – Layout

Supported Documents

Create a New Supported Document

Document Type	Short Name	Industry Code	Func Code	Trans Code	Is Encounter
270_X12_BATCH	270	004010X092A1	HS	270	N
271_X12_INTERACTIVE	271	004010X092A1	HB	271	N
270_X12_INTERACTIVE	270	004010X092A1	HS	270	N
271_X12_BATCH	271	004010X092A1	HB	271	N
276_X12_BATCH	276	004010X093A1	HR	276	N
276_X12_INTERACTIVE	276	004010X093A1	HR	276	N
277_X12_BATCH	277	004010X093A1	HN	277	N
277_X12_INTERACTIVE	277	004010X093A1	HN	277	N
837_D_X12_BATCH	837	004010X097A1	HC	837	N
837_I_X12_BATCH	837I	004010X096A1	HC	837	N
837_P_X12_BATCH	837P	004010X098A1	HC	837	N
837_P_E_X12_BATCH	837P	004010X098A1	HC	837	Y
837_I_E_X12_BATCH	837I	004010X096A1	HC	837	Y
837_D_E_X12_BATCH	837D	004010X097A1	HC	837	Y
277_U_X12_BATCH	277	003070X070	HN	277	N
278_X12_BATCH	278	004010X094A1	HI	278	N
278_R_X12_BATCH	278	004010X094A1	HI	278	N
834_X12_BATCH	834	004010X095A1	BE	834	N
835_X12_BATCH	835	004010X091A1	HP	835	N
820_X12_BATCH	820	004010X061A1	RA	820	N
997_X12_BATCH	997	004010X097	FA	997	N
TA1_X12_BATCH	TA1	*	*	*	N
NEMT_EXTRACT	NEMT	NA	NA	NET	N
KENPAC_EXTRACT	KENPAC	NA	NA	KPC	N
NCPDP_GENERIC	NCPDP_GENERIC	NA	NA	NCP	N
NSF_BATCH	NSF	NA	NA	NSF	N

Create a New Supported Document

6.2.3.8.5 Supported Documents – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Document Type	The complete identification code for the	Field	VARCHAR2	50	T_	CDE_

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
	supported document type	(Selectable Link)			SUPPORTED_ DOC	IDENTIFICATION
Short Name	A short description code for the supported document type	Field	VARCHAR2	25	T_ SUPPORTED_ DOC	DSC_SHORT_ NAME
Industry Code	The Industry Code associated with the document type	Field	VARCHAR2	12	T_ SUPPORTED_ DOC	CDE_ INDUSTRY
Func Code	The Functional Code associated with the document type	Field	CHAR	2	T_ SUPPORTED_ DOC	CDE_ FUNCTIONAL
Trans Code	An abbreviated code for the document type	Field	CHAR	3	T_ SUPPORTED_ DOC	CDE_ TRANSACTION
Is Encounter	This value indicates if the document type is an encounter	Field	CHAR	1	T_ SUPPORTED_ DOC	IND_ ENCOUNTER

6.2.3.9 Page: Supported Document Management

6.2.3.9.1 Description:

The Supported Document Management page allows for the creation and modification of supported documents.

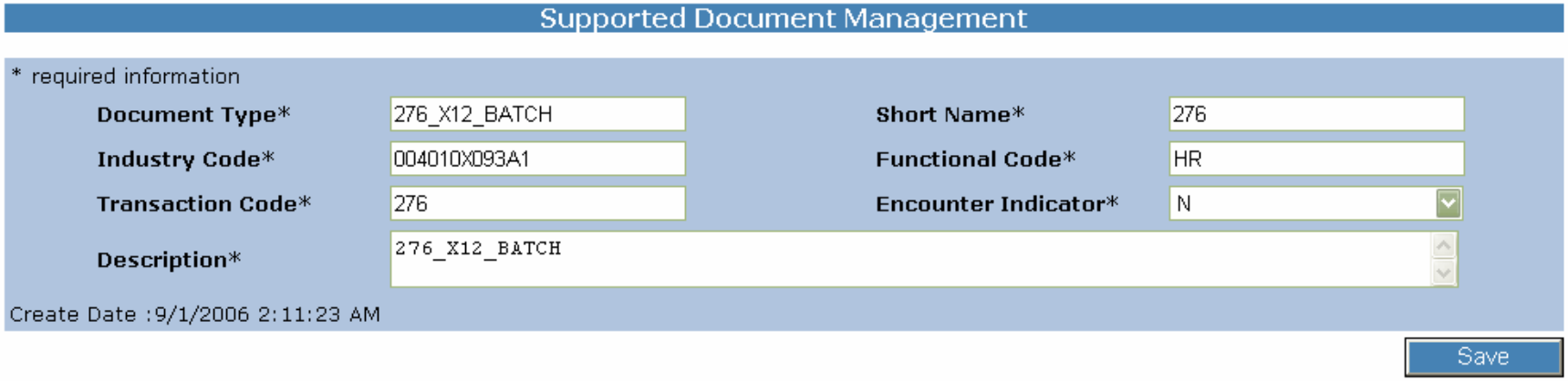
6.2.3.9.2 Technical Name:

EDIUtilities.SupportedDocMgmt

6.2.3.9.3 Web Page Name:

Supported Document Management

6.2.3.9.4 Supported Document Management – Layout



6.2.3.9.5 Supported Document Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Document Type	The complete identification code for the supported document type	Text Box	VARCHAR2	50	T_ SUPPORTED_ DOC	CDE_ IDENTIFICATION
Short Name	A short description code for the supported document type	Text Box	VARCHAR2	25	T_ SUPPORTED_ DOC	DSC_SHORT_ NAME
Industry Code	The Industry Code associated with the	Text Box	VARCHAR2	12	T_ SUPPORTED_	CDE_

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
	document type				DOC	INDUSTRY
Func Code	The Functional Code associated with the document type	Text Box	CHAR	2	T_ SUPPORTED_ DOC	CDE_ FUNCTIONAL
Trans Code	An abbreviated code for the document type	Text Box	CHAR	3	T_ SUPPORTED_ DOC	CDE_ TRANSACTION
Encounter Indicator	This value indicates if the document type is an encounter	Drop Down Box	CHAR	1	T_ SUPPORTED_ DOC	IND_ ENCOUNTER
Description	The description of the document type	Text Box	VARCHAR2	255	T_ SUPPORTED_ DOC	DSC_ SUPPORTED_ DOC
Create Date	The date the document type was created	Label	DATE		T_ SUPPORTED_ DOC	DTE_CREATED

6.2.3.10 Page: Trading Partner Types

6.2.3.10.1 Description:
This page lists the current trading partner types available.

6.2.3.10.2 Technical Name:
EDIUtilities.TPType

6.2.3.10.3 Web Page Name:
Trading Partner Types

6.2.3.10.4 Trading Partner Types – Layout

Trading Partner Types

Create a New Trading Partner Type

Code	Description
CORPORATION	CORPORATION
VENDOR	VENDOR
PROVIDER	PROVIDER
CLINIC	CLINIC
AGENT	AGENT
MCO	Managed Care Organization

Create a New Trading Partner Type

6.2.3.10.5 Trading Partner Types – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Code	The identification code for the Trading Partner type	Label (Selectable Link)	VARCHAR2	50	T_CDE_TP_TYPE	CDE_TP_TYPE
Description	The description of the Trading Partner type	Label	VARCHAR2	255	T_CDE_TP_TYPE	DSC_TP_TYPE

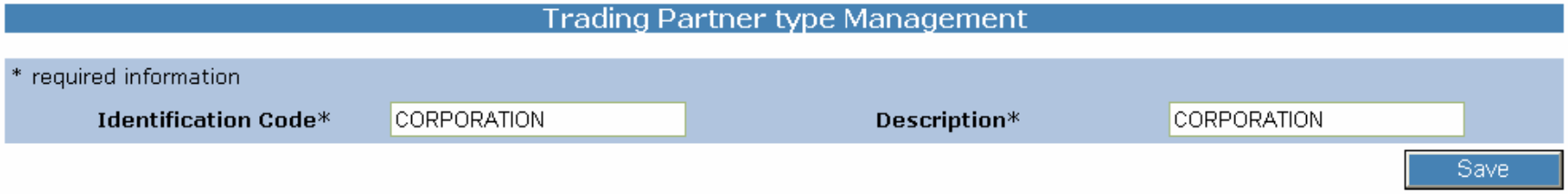
6.2.3.11 Page: Trading Partner Type Management

6.2.3.11.1 Description:
This page allows for the creation and modification of trading partner types.

6.2.3.11.2 Technical Name:
EDIUtilities.TPTypeMgmt

6.2.3.11.3 Web Page Name:
Trading Partner Type Management

6.2.3.11.4 Trading Partner Type Management – Layout



6.2.3.11.5 Trading Partner Type Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Identification Code	The identification code for the Trading Partner type	Text Box	VARCHAR2	50	T_CDE_TP_TYPE	CDE_TP_TYPE
Description	The description of the Trading Partner type	Text Box	VARCHAR2	255	T_CDE_TP_TYPE	DSC_TP_TYPE

6.2.3.12 Page: FTS Configuration

This section is not yet fully implemented and in the meantime FTS configurations changes can be done by modifying the configuration XML files through a text editor.

6.2.3.12.1 Description:

The FTS Configuration page is a distinct application that allows for the creation and modification of FTS XML configuration files. Each environment contains multiple configurations that can be updated individually. Each configuration within an environment is identified through a unique **identifier** attribute. Configurations can be updated and created.

Each environment can also be generated and saved.

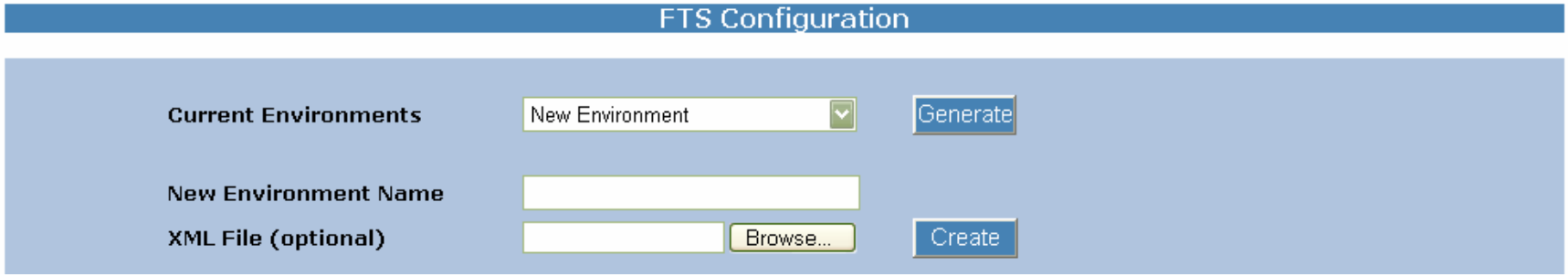
6.2.3.12.2 Technical Name:

EDIUtilities.FTS Configuration

6.2.3.12.3 Web Page Name:

File Transfer Service Configuration

6.2.3.12.4 FTS Configuration Environment Management – Layout

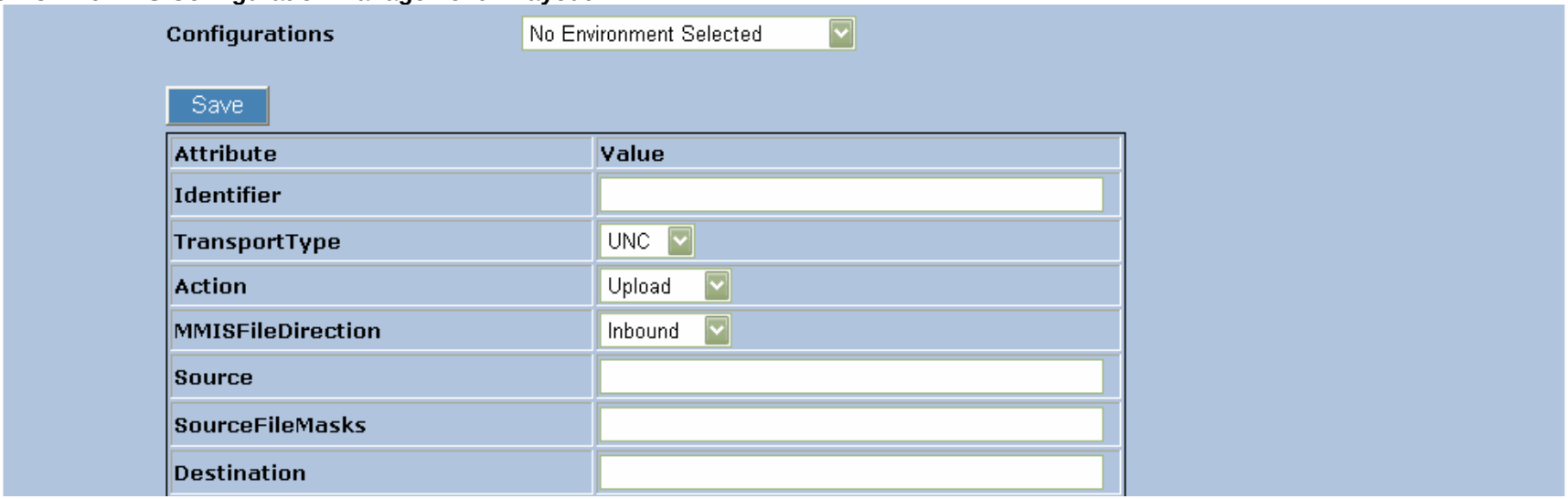


6.2.3.12.5 FTS Configuration Environment Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Current Environments	The environment which is currently selected (Can also be used to create a new environment)	Drop Down Box	Text	50	Environment	Environment-Name
Generate	Used to generate the selected environment	Button				
New Environment	The name of the new environment	Text Box	Text	50	Environment	Environment-

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Name						Name
XML File (Optional)	An existing configuration file can be submitted to the application	Label				
Browse	This control is used to create an “Open File Dialog” on the users system to locate an existing XML configuration file	Browse Control				
Create	Used to transform an existing XML configuration file into a dynamic file that is used by the application	Button				

6.2.3.12.6 FTS Configuration Management – Layout



6.2.3.12.7 FTS Configuration Management – Field Descriptions

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
Configurations	The configuration identifier for the currently	Drop Down	Text	200	Configuration	Identifier

Field	Description	Field Type	Data Type	Length	DB Table	DB Attributes
	selected environment.	Box				
Save	Used to save the current configuration	Button				
Attribute	This is an attribute of the currently selected configuration	Label				
Value	This is the value of the attribute for the currently selected configuration. The editable control can vary based on the type of attribute.	*Varies*	Text	200	Configuration	

7 Appendix E – Documentation for Eligibility Inquiry (270/ 271)

7.1 Introduction

The eligibility verification function is invoked by the SoapInvokeMethod() which is initiated by the xml270276Handler that runs in unison with the xml270276srvrd.

Or it's directly called by the batch program edip270d.

The eligibility verification function gets the input inquiry from the Caller method.

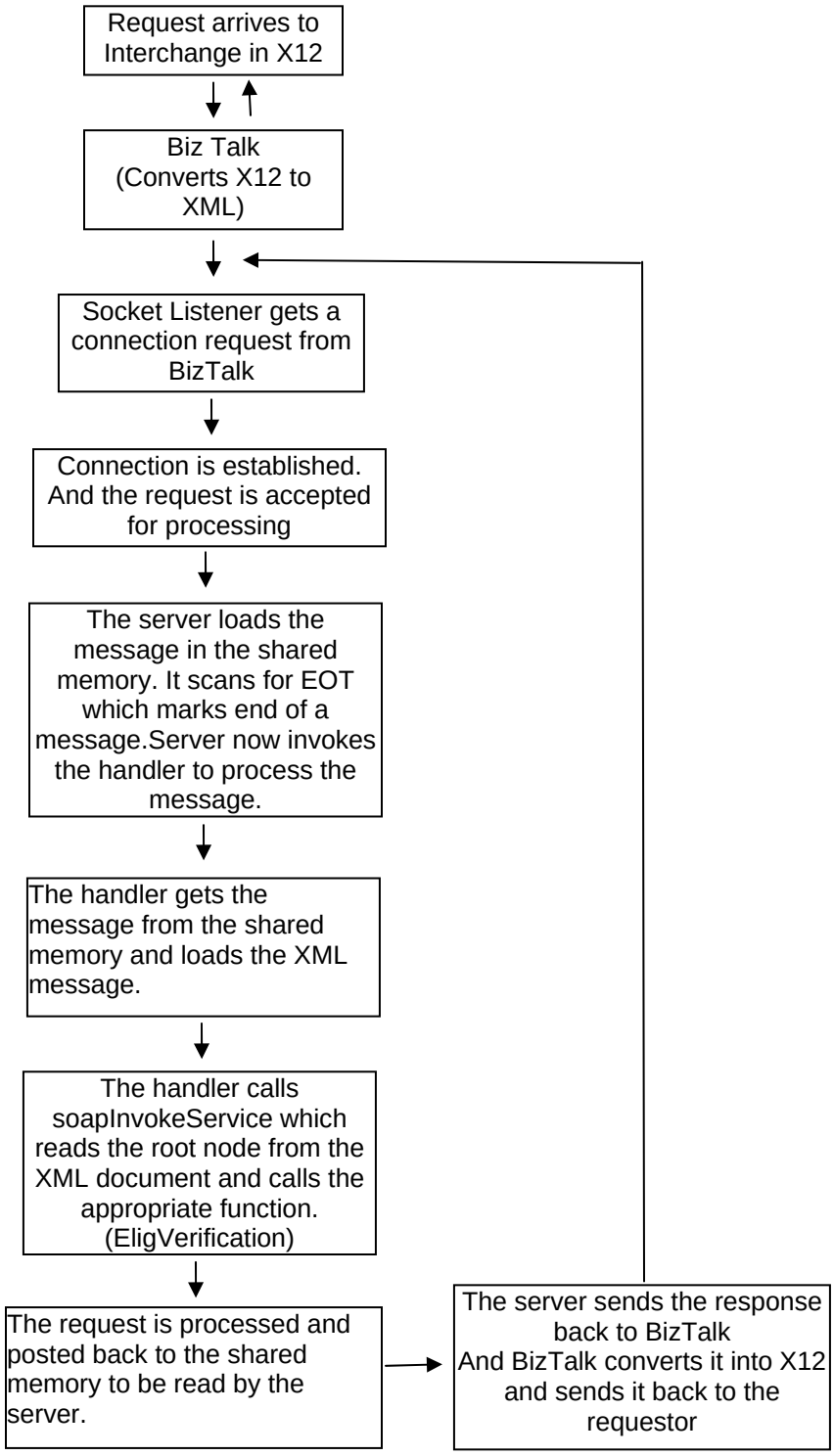
The main purpose of this module is eligibility verification. The response is sent back to the caller function. The handler takes the response and sends it back to the client in the appropriate format. (XML)

7.2 Process Flow

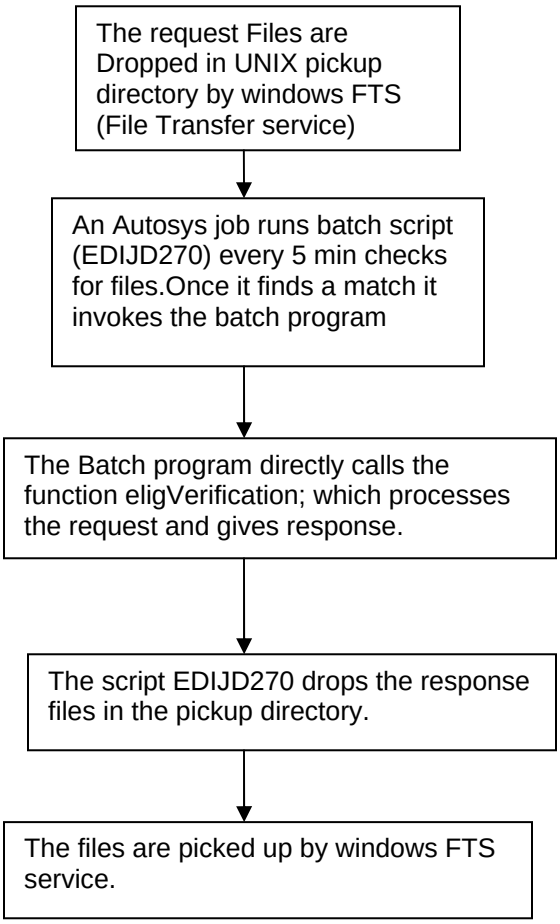
UNIX can process eligibility verification by 2 methods:

- Interactive Processing (Socket Listener)
- Batch Processing

7.2.1 Interactive (socket listener)



7.2.2 Batch Processing:



7.2.3 Detailed design description:

Program source: evselig.sc

Make file: libevs.so.mak

Shared library: libevs.so

Input: pointer to an XMLElement that contains a request.

Output: pointer to an XMLElement that contains an Eligibility request and response elements

7.3 Entry points/functions

- eligVerification() : This function is the entry point to the eligibility verification
- setEligError():This function creates the error node in the response.
- verifyProvider(): This function validates the provider.

- verifyProviderSpeciality: This function determines the providers specialization.
- getNumEvsVer() :This function returns a unique EVS number.
- updateEVSTable():Adds the inquiry to the inquiry table
- verifyDates():This function validates the to/from dates on the request.
- verifyProvDates(): Performs edits on the provider dates.
- verifyRecipient(): Performs edits on the recipient, checks if the recipient is genuine and has active membership.
- MemberInfo():fetches the members card issuance date if available.
- AddMedicaidInfo: Prepares response with Medicaid information.
- AddQMBInfo: Prepares response with QMB information.
- AddMedPartAInfo: Prepares response with MedPartA information.
- AddMedPartBInfo: Prepares response with MedPartB information.
- lockinInfo: Prepares response for lock in Information
- AddUltraSoundInfo: Prepares response with UltraSound information.
- AddKCHIPInfo: Prepares response with KCHIP information.
- AddHospiceInfo: Prepares response with Hospice information.
- eligMgdCareInfo_kenpac: Prepares response for Managed care information
- eligTPLInfo :Prepares response for the TPL information
- AddHearingInfo: Prepares response with Hearing information.
- AddVisionInfo: Prepares response with Vision information.
- AddDentalInfo: Prepares response with Dental information.
- AddEPSDTInfo: Prepares response with EPSDT information.
- AddBenefitPlanInfo: Prepares response with Benefit Plan information.
- AddCoPayInfo: Prepares response with Co Pay information.
- AddOutOfPocketMax: Prepares response with Out of Pocket max information.
- AddOutOfPocketUsed: Prepares response with Out of Pocket Used information.
- nameLookup:This function performs name lookup
- caseLookup: This function performs case lookup
- locInfoWeb: This function gets level of care for the web.

7.3.1 Flow Diagram:

7.3.2 High level process flow

The Eligibility Verification program takes a 270 inquiry transaction and validates if the transaction is eligible or not. This takes the XML of the input inquiry (270) and edits the file by adding the eligibility nodes if the transaction valid, other wise error nodes are added to the input XML and sent back. Hence the response (271) is the modified XML of the input transaction. The complete process involves checking /validating the fields in the input XML and editing the input file by adding the necessary attributes to prepare the response (271).

The first step of the function is to check the connectivity with the database. After getting connected to the database the process loops through the “eligRequest” nodes.

There are mainly 4 types of transactions WEB,VRS,XWB,EDI(default).The field clientSecurityKey is checked to see the type of transaction. If the transaction is not XWEB the provider information in the input is validated to check if the provider has an active contract using the provider number and provider location code .

Now a unique EVS transaction number is calculated , using this value the version number of the transaction is set in the XML .Next step is to verify the From and to dates of service; there are some conditions to check the validity ,

The from and to dates of service must be in a 13 month window, The to date of service must be on or after the from date of service, Future eligibility over a calendar month away cannot be requested, Next months eligibility can only be requested if the monthly roll for that month has been processed and if the transaction is VRS then the from and to dates of service must be within the same month.

For all transactions other then XWEB, provider dates are also validates to check if the service dates are within the provider plan enrollment.

For web only requests a name and case lookup is performed to complete the eligibility verification process.

The name look up is performed using the first and last name of the recipient, after the process a node nameBase is added with the following attributes: subscriberID, first name, last name, subscriberDateOfBirth and subscriberCaseNumber.

The case look up is performed based on the case number and the a xml node Case Base is added to the response with the following attributes: subscriberID, lastName, firstName, middleName, caseDateOfBirth, headCaseName .

For non-web transactions the subscriber details are verified to check if the recipient is active. The SubscriberId, subscriber SSN, subscriberDateOfBirth , sex, last name and first name and Dates of service used for the validation .Once validation is completed the “eligSubcriber” node is edited with the attributes , subscrierDateOfBirth, subscriberId, firstName, lastName etc.

There are 3 types or inquires

- Medicaid ID
- SSN, last name, first name
- last name,first name,sex,DOB,Dates of Service.

If newBornIndicator =’Y’ in request

- Get sak_recip for mother's medicaid id from t_re_base
- For this sak_recip, get sak_case from t_re_case_xref
- For the above sak_case, find all rows in t_re_case_xref and order by dte_cert descending. Now select sak_recip for the first one in the list.
- Check if the DOB associated with the above sak_recip is older than 365 days. If older than 365 days, return eligibility of mother. Otherwise, return eligibility of the new born.

At this point the eligibility nodes for the transaction are created with various information like Medicaiid,dental, vision,benefit plan details etc. The information is retrieved based on the subscriber’s Medicaid ID.

Additionally in the input transaction has a “loc info” field the level of care information is also obtained from the database based on the subscriber’s Medicaid ID and the necessary fields are the added to the response.

If any error occurs during any phase of the entire process, an error node is appended to the response and the process is exited.

T_EVS_INQUIRY table which keeps track of the transactions is updated regardless of edit failure.

At the end of the whole process the XML response is sent back to the calling function and the xml270276 handler sends back the response to the client which initiated the request.

7.3.3 SQL Queries and Context:

Int verifyProviderSpeciality(XMLElement *eligReq,WORK_ST *wks_ptr)

```
EXEC SQL
 SELECT NVL(cde_prov_type,' ')
 INTO  :wks.szProvSpec
 FROM  t_pr_type
 WHERE sak_prov_loc =  :wks.lzSakProvLoc;
```

int verifyRecipient (XMLElement *eligSub, WORK_ST *wks_ptr)

1. Inquiry based on Medicaid Id

```
EXEC SQL SELECT   count(*),
                  sak_recip,
                   id_medicaid,
                   nam_first,
                   nam_last,
                   nam_mid_init,
                   dte_birth,
                   num_ssn,
                   ind_active,
                   cde_sex,
```

```
                    cde_count
INTO
                   :wks.lzRowCount,
                   :wks.lzSakRecip ,
                     :wks.szIdMedicaid,
                     :wks.szNamFirst,
                     :wks.szNamLast,
                     :wks.szNamMid,
                     :wks.szRecipDob,
                     :wks.szRecipSSN,
                     :wks.szIndActive,
                     :wks.szCdeSex,
                   :wks.szCounty

FROM   t_re_base
WHERE  id_medicaid = :wks.szReqIdMedicaid
GROUP BY  sak_recip, id_medicaid, nam_first, nam_last, nam_mid_init, dte_birth,   num_ssn, ind_active,
cde_sex, cde_county ;
```

2. Inquiry based on SSN, firstname and last name

```
EXEC SQL
SELECT sak_recip,
id_medicaid,
nam_first,
nam_last,
nam_mid_init,
dte_birth,
num_ssn,
ind_active,
cde_sex,
cde_county
```

```
FROM t_re_base
WHERE  num_ssn = '%s' AND  nam_first like upper(substr('%s',1,7)) || '%'
AND  nam_last  like upper(substr('%s',1,5))  || '%'"
```

3 . Inquire based on Name ,DOB,sex,Dates of Service

```
EXEC SQL  SELECT
        a.sak_recip,
        a.id_medicaid,
        a.nam_first,
        a.nam_last ,
        a.nam_mid_init,
        a.dte_birth,
        a.num_ssn,
       a.ind_active,
       a.cde_sex,
      a.cde_county
   FROM   t_re_base a ,  t_re_elig b
   WHERE a.sak_recip = b.sak_recip
   AND a.nam_first like upper(substr('%s',1,7)) || '%c'
 AND  a.nam_last  like upper(substr('%s',1,5)) ||'%c' AND
  a.cde_sex ='%s' AND  a.dte_birth = '%s' AND  b.dte_end >= to_number('%s')
 AND b.dte_effective <= to_number('%s' )" ,wks.szNamFirst,'%',wks.szNamLast,'%'
wks.szCdeSex,wks.szRecipDob,wks.szfromDateOfService,wks.sztoDateOfService
```

int MemberInfo (XMLElement *eligSub, WORK_ST *wks_ptr)

Retrieve extra member data

```
EXEC SQL
DECLARE member_curs CURSOR FOR
SELECT
 NVL(e.cde_aid_category, ' ')
FROM
  t_re_aid_elig b,
  t_cde_aid e
WHERE b.sak_recip = :wks.lzSakRecip
AND b.sak_cde_aid = e.sak_cde_aid
AND b.cde_status1 <> 'H'
AND e.cde_aid_category IN ('Z','ZK','ZJ','ZQ','ZL')
AND b.dte_effective <= to_number(:wks.sztoDateOfService)
AND b.dte_end >= to_number(:wks.szfromDateOfService) ;
```

Obtain card issuance Date

```
SELECT NVL(cast(dte_issue as char(8)), '00000000')
     from t_re_id_card_track
     WHERE sak_recip = :wks.lzSakRecip
     ORDER BY dte_issue DESC;
```

int AddMedicaidInfo(XMLElement *eligSub, WORK_ST *wks_ptr)

```
SELECT a.dte_effective, a.dte_end
  FROM   t_re_elig a, t_pub_hlth_pgm b
  WHERE  a.sak_recip = :wks.lzSakRecip
  AND   a.sak_pub_hlth = b.sak_pub_hlth
  AND   a.cde_status1 <> 'H'
  AND   b.cde_pgm_health = 'NONE'
  AND a.dte_effective <= to_number(:wks.sztoDateOfService)
  AND a.dte_end >= to_number(:wks.szfromDateOfService);
```

int AddQMBInfo(XMLElement *eligSub, WORK_ST *wks_ptr)

```
EXEC SQL
  DECLARE qmb_curs CURSOR FOR
  SELECT a.dte_effective, a.dte_end
  FROM   t_re_elig a, t_pub_hlth_pgm b
  WHERE  a.sak_recip = :wks.lzSakRecip
  AND   a.sak_pub_hlth = b.sak_pub_hlth
  AND   b.cde_pgm_health = 'QMB'
  AND   a.dte_effective <= to_number(:wks.szfromDateOfService)
  AND   a.dte_end >= to_number(:wks.sztoDateOfService);
```

int AddMedPartAInfo (XMLElement *eligSub, WORK_ST *wks_ptr)

```
SELECT to_char( dte_effective),
       to_char(dte_end)
  FROM t_re_medicare_a
  WHERE sak_recip =  :wks.lzSakRecip
  AND dte_effective <= to_number(:wks.sztoDateOfService)
  AND dte_end >= to_number(:wks.szfromDateOfService)
  ORDER BY dte_end DESC, dte_effective DESC ;
```

int AddMedPartBInfo (XMLElement *eligSub, WORK_ST *wks_ptr)

```
SELECT to_char( dte_effective),
       to_char(dte_end)
  FROM t_re_medicare_b
  WHERE sak_recip =  :wks.lzSakRecip
  AND dte_effective <= to_number(:wks.sztoDateOfService)
  AND dte_end >= to_number(:wks.szfromDateOfService)
```

```
  ORDER BY dte_end DESC, dte_effective DESC ;
```

int lockinInfo(XMLElement *eligSub,WORK_ST *wks_ptr)

```
SELECT a.dte_effective,
        a.dte_end,
        decode(p.ind_name_type, 'P', '1', 'B', '2', ' '),
        p.name,
        decode(p.ind_name_type, 'P', nvl(RTRIM(SUBSTR(p.name, 1, 25)), ' '), ' '),
        decode(p.ind_name_type, 'P', nvl(RTRIM(SUBSTR(p.name, 26, 13)), ' '), ' '),
        n.num_phone,
        b.cde_prov_type
    FROM t_re_assign_plan a,
        t_pr_type b,
        t_pr_nam p,
        t_pr_loc_nm_adr l,
        t_pr_adr n,
        t_benefit_plan_group g,
        t_benefit_plan_type h
    WHERE a.sak_recip = :wks.lzSakRecip
      AND a.sak_prov_loc = b.sak_prov_loc
      AND b.sak_prov_loc = l.sak_prov_loc
      AND l.ind_addr_type = 'H'
      AND l.sak_prov = n.sak_prov
      AND l.sak_short_address = n.sak_short_address
      AND b.sak_prov = p.sak_prov
      AND p.sak_short_name = l.sak_short_name
      AND a.sak_pub_hlth = g.sak_pub_hlth
      AND g.sak_benefit_plan_type = h.sak_benefit_plan_type
      AND h.dsc_50 = 'LOCKIN'
      AND a.dte_effective <= to_number(:wks.sztoDateOfService)
```

```
    AND a.dte_end >= to_number(:wks.szfromDateOfService)
  ORDER BY a.dte_end DESC;
```

int AddUltraSoundInfo (XMLElement *eligSub, WORK_ST *wks_ptr)

```
Select a.dte_first_svc, a.dte_last_svc, b.cde_proc
  From t_mphx_phys_dtl a, t_proc b, t_proc_group c
  Where a.sak_recip = :wks.lzSakRecip
  And a.dte_first_svc >= to_number(:wks.szfromDateOfService)
  And a.dte_last_svc <= to_number(:wks.sztoDateOfService)
  And a.ind <> 'R'
  And a.ind <> 'A'
  And a.sak_procedure = c.sak_procedure_from
  And c.sak_proc_type = 1122
  And c.dte_effective <= to_number(:wks.sztoDateOfService)
  And c.dte_end >= to_number(:wks.szfromDateOfService)
  And c.sak_procedure_from = b.sak_procedure
  order by a.dte_first_svc DESC;
```

int AddKCHIPInfo(XMLElement *eligSub, WORK_ST *wks_ptr)

```
SELECT a.dte_effective, a.dte_end
  FROM   t_re_elig a, t_pub_hlth_pgm b
  WHERE  a.sak_recip = :wks.lzSakRecip
  AND   a.sak_pub_hlth = b.sak_pub_hlth
  AND   b.cde_pgm_health = 'KCHIP'
  AND   a.cde_status1 <> 'H'
  AND  a.dte_effective <= to_number(:wks.sztoDateOfService)
  AND  a.dte_end >= to_number(:wks.szfromDateOfService);
```

int AddHospiceInfo(XMLElement *eligSub, WORK_ST *wks_ptr)

```
SELECT a.dte_effective, a.dte_end
  FROM   t_re_elig a, t_pub_hlth_pgm b
  WHERE  a.sak_recip = :wks.lzSakRecip
  AND   a.sak_pub_hlth = b.sak_pub_hlth
  AND   b.cde_pgm_health IN('HOSP','L-M7','L-M9')
  AND   a.cde_status1 <> 'H'
  AND   a.dte_effective <= to_number(:wks.sztoDateOfService)
  AND   a.dte_end >= to_number(:wks.szfromDateOfService);
```

int eligMgdCareInfo_kenpac (XMLElement *eligSub, WORK_ST *wks_ptr)

```
SELECT decode(trim(b.cde_pgm_health),'PART', 'P', 'KENP', 'K', 'LOCKM', 'L'),
       b.dsc_pgm_health,
       a.dte_effective,
       a.dte_end
  FROM   t_pub_hlth_pgm b,t_re_pmp_assign a
  WHERE  a.sak_recip = :wks.lzSakRecip
  AND    a.cde_status1 <>'H'
  AND    a.sak_pub_hlth = b.sak_pub_hlth
  AND    b.cde_pgm_health IN('PART', 'KENP', 'LOCKM')
  AND    a.dte_effective <= to_number(:wks.sztoDateOfService)
  AND    a.dte_end >= to_number(:wks.szfromDateOfService)
  ORDER BY a.dte_end DESC, a.dte_effective DESC ;
```

For Kenpac:

```
SELECT
         NVL(nam.ind_name_type,' '),
         NVL(nam.name,' '),
```

```
        decode(nam.ind_name_type,'P', NVL(rtrim(substr(nam.name,1,25)),' '), ' '),
decode(nam.ind_name_type,'P', NVL(rtrim(substr(nam.name, 26, 13)),' '),' '),
        NVL( adr.num_phone ,' '),
        ' ',
        ' ',
        pmp_assign.cde_mc_region
FROM    t_re_pmp_assign pmp_assign,
        t_pub_hlth_pgm pgm,
        t_pmp_svc_loc pmp_loc,
        t_pr_nam nam,
        t_pr_loc_nm_adr loc,
        t_pr_adr adr
WHERE   pmp_assign.sak_recip = :wks.lzSakRecip
AND     pmp_assign.cde_status1 <>'H'
AND     pmp_assign.dte_effective <= to_number(:wks.sztoDateOfService)
AND     pmp_assign.dte_end >= to_number(:wks.szfromDateOfService)
AND     pmp_assign.sak_pmp_ser_loc = pmp_loc.sak_pmp_ser_loc
AND     pmp_loc.sak_prov_loc = loc.sak_prov_loc
AND    loc.sak_prov = nam.sak_prov
AND    loc.sak_short_name = nam.sak_short_name
AND    loc.ind_addr_type = 'S'
AND    loc.sak_prov = adr.sak_prov
AND    loc.sak_short_address = adr.sak_short_address
AND     pmp_assign.sak_pub_hlth = pgm.sak_pub_hlth
AND     pgm.cde_pgm_health IN('KENP')
AND     pmp_assign.dte_effective <= pgm.dte_end
AND     pmp_assign.dte_end >= pgm.dte_effective
ORDER BY pmp_assign.dte_end DESC,
            pmp_assign.dte_effective DESC ;
```

For LockM or Partnership:

```
SELECT     NVL(nam.ind_name_type,' '),
           NVL(nam.name,' '),
           decode(nam.ind_name_type,'P', NVL(rtrim(substr(nam.name,1,25)),' '), ' ')  ,
           decode(nam.ind_name_type,'P', NVL(rtrim(substr(nam.name, 26, 13)),' '),' '),
           NVL( adr.num_phone ,' '),
           NVL(pmp_assign.cde_mc_region,' '),     /*region code added CO 2086*/
           NVL(nam.name,' '),
           NVL( adr.num_phone ,' ')
       FROM    t_re_pmp_assign pmp_assign,
           t_pub_hlth_pgm pgm,
           t_pmp_svc_loc pmp_loc,
           t_pr_nam nam,
           t_pr_loc_nm_adr loc,
           t_pr_adr adr
       WHERE  pmp_assign.sak_recip = :wks.lzSakRecip
       AND    pmp_assign.cde_status1 <>'H'
       AND    pmp_assign.dte_effective <= to_number(:wks.sztoDateOfService)
       AND    pmp_assign.dte_end >= to_number(:wks.szfromDateOfService)
       AND    pmp_assign.sak_pmp_ser_loc = pmp_loc.sak_pmp_ser_loc
       AND    pmp_loc.sak_prov_loc = loc.sak_prov_loc
       AND    loc.sak_prov = nam.sak_prov
       AND    loc.sak_short_name = nam.sak_short_name
       AND    loc.ind_addr_type = 'S'
       AND    loc.sak_prov = adr.sak_prov
       AND    loc.sak_short_address = adr.sak_short_address
       AND    pmp_assign.sak_pub_hlth = pgm.sak_pub_hlth
       AND    pgm.cde_pgm_health IN('PART', 'LOCKM')
       AND    pmp_assign.dte_effective <= pgm.dte_end
       AND    pmp_assign.dte_end >= pgm.dte_effective
       ORDER BY pmp_assign.dte_end DESC,
```

```
                pmp_assign.dte_effective DESC ;
```

int eligTPLInfo(XMLElement *eligSub, WORK_ST *wks_ptr)

```
  SELECT
    b.dte_effective,
    b.dte_end,
    DECODE(a.num_tpl_policy, '              ', ' ', a.num_tpl_policy),
    NVL(RTRIM(c.nam_first),' '),
    NVL(RTRIM(c.nam_last),' '),
    NVL(RTRIM(f.nam_first),' '),
    NVL(RTRIM(f.nam_last),' '),
    SUBSTR(d.dsc_cover_type, 1, 50),
    /* JMM
    SUBSTR(e.nam_bus,1,35), */
    SUBSTR(e.nam_bus,1,45),
    e.cde_carrier,
    /* JMM */
    a.sak_policy_ownr,
    a.cde_policy_ownr
  FROM
    t_tpl_resource a,
    t_coverage_xref b,
    t_policy_holder c,
    t_coverage_type d,
    t_tpl_carrier e,
    t_re_base f
  WHERE a.sak_recip = :wks.lzSakRecip
   AND a.sak_tpl_resource = b.sak_tpl_resource
   AND a.sak_policy_ownr = c.sak_pol_hold(+)
   AND b.cde_coverage = d.cde_coverage
```

```
AND a.sak_carrier = e.sak_carrier
AND a.sak_recip = f.sak_recip(+)
AND b.dte_effective <= to_number(:wks.sztoDateOfService)
AND b.dte_end >= to_number(:wks.szfromDateOfService)
ORDER BY b.dte_end DESC;
```

int AddHearingInfo (XMLElement *eligSub, WORK_ST *wks_ptr)

```
Select a.dte_first_svc, a.dte_last_svc, b.cde_proc
  From t_mphx_phys_dtl a, t_proc b, t_proc_group c
  Where a.sak_recip = :wks.lzSakRecip
  And a.dte_first_svc >= to_number(:wks.szfromDateOfService)
  And a.dte_last_svc <= to_number(:wks.sztoDateOfService)
  And a.ind <> 'R'
  And a.ind <> 'A'
  And a.sak_procedure = c.sak_procedure_from
  And c.sak_proc_type = 1121
  And c.dte_effective <= to_number(:wks.sztoDateOfService)
  And c.dte_end >= to_number(:wks.szfromDateOfService)
  And c.sak_procedure_from = b.sak_procedure
  order by a.dte_first_svc DESC;
```

int AddVisionInfo (XMLElement *eligSub,WORK_ST *wks_ptr)

```
Select a.dte_first_svc, a.dte_last_svc, b.cde_proc
  From t_mphx_phys_dtl a, t_proc b, t_proc_group c
  Where a.sak_recip = :wks.lzSakRecip
  And a.dte_first_svc >= to_number(:wks.szfromDateOfService)
  And a.dte_last_svc <= to_number(:wks.sztoDateOfService)
  And a.ind <> 'R'
  And a.ind <> 'A'
```

```
And a.sak_procedure = c.sak_procedure_from
And c.sak_proc_type = 1123
And c.dte_effective <= to_number(:wks.sztoDateOfService)
And c.dte_end >= to_number(:wks.szfromDateOfService)
And c.sak_procedure_from = b.sak_procedure
order by a.dte_first_svc DESC;
```

int AddDentalInfo (XMLElement *eligSub, WORK_ST *wks_ptr)

```
Select distinct a.dte_first_svc, b.cde_proc
From t_mphx_dental_dtl a, t_proc b, t_proc_group c
Where a.sak_recip = :wks.lzSakRecip
And a.dte_first_svc >= to_number(:wks.szfromDateOfService)
And a.ind <> 'R'
And a.ind <> 'A'
And a.sak_procedure = c.sak_procedure_from
And c.sak_proc_type IN(142,1001,1120,3000)
And c.dte_effective <= to_number(:wks.sztoDateOfService)
And c.dte_end >= to_number(:wks.szfromDateOfService)
And c.sak_procedure_from = b.sak_procedure
order by a.dte_first_svc DESC;
```

int AddEPSDTInfo (XMLElement *eligSub, WORK_ST *wks_ptr)

```
SELECT a.dte_first_svc,b.cde_proc
From t_re_eps_hist_ext a, t_proc b, t_proc_group c
Where a.sak_recip = :wks.lzSakRecip
And a.dte_first_svc >= to_number(:wks.szfromDateOfService)
And a.sak_procedure = c.sak_procedure_from
And c.sak_proc_type IN(44, 60 ,61, 64, 80, 116, 1008,
            1009 ,1010,1040)
```

```
And c.dte_effective <= to_number(:wks.sztoDateOfService)
And c.dte_end >= to_number(:wks.szfromDateOfService)
And c.sak_procedure_from = b.sak_procedure
order by a.dte_first_svc DESC;
```

int AddBenefitPlanInfo(XMLElement *eligSub, WORK_ST *wks_ptr)

For inquiries on or before july 1st 2006

```
SELECT a.dte_effective, a.dte_end,b.dsc_pgm_health
  FROM   t_re_elig a, t_pub_hlth_pgm b
  WHERE  a.sak_recip = :wks.lzSakRecip
  AND    a.sak_pub_hlth = b.sak_pub_hlth
  AND    a.cde_status1 <> 'H'
  AND    upper(b.cde_type_plan) = 'BNFT'
  AND    upper(b.cde_pgm_health)NOT IN('QMB','NONE','KCHIP')
  AND    a.dte_effective <= to_number(:wks.sztoDateOfService1)
  AND    a.dte_end >= to_number(:wks.szfromDateOfService1);

EXEC SQL SELECT  ind_copay
       INTO  :indCopay
       FROM   t_re_old_copay
       WHERE  sak_recip = :wks.lzSakRecip
       AND    dte_effective <= to_number(:wks.sztoDateOfService1)
       AND    dte_end >= to_number(:wks.szfromDateOfService1);
```

For Inquiries after july 1st 2006

```
SELECT a.dte_effective, a.dte_end,b.ind_copay,b.dsc_pgm_health
  FROM   t_re_elig a, t_pub_hlth_pgm b
  WHERE  a.sak_recip = :wks.lzSakRecip
```

```
  AND    a.sak_pub_hlth = b.sak_pub_hlth
  AND   a.cde_status1 <> 'H'
  AND    upper(b.cde_type_plan) = 'BNFT'
  AND    upper(b.cde_pgm_health)NOT IN('QMB','NONE','KCHIP')
  AND    a.dte_effective <= to_number(:wks.sztoDateOfService)
  AND    a.dte_end >= to_number(:wks.szfromDateOfService);
```

int AddOutOfPocketUsed(XMLElement *eligSub, WORK_ST *wks_ptr)

```
EXEC SQL
  SELECT NVL(sum(amt_copay),0)
  INTO :copay_amt
  FROM t_clm_copay_x
  WHERE sak_recip =:wks.lzSakRecip
  AND cde_status1 = 'A'
  AND dte_year_mth <= to_number(substr(:wks.sztoDateOfService,0,6))
  AND dte_year_mth >= to_number(substr(:szyearBeginDate,0,6));

EXEC SQL
  SELECT NVL(sum(amt_coinsurance),0)
  INTO :coins_amt
  FROM t_clm_coins_x
  WHERE sak_recip =:wks.lzSakRecip
  AND cde_status1 = 'A'
  AND dte_year_mth <= to_number(substr(:wks.sztoDateOfService,0,6))
  AND dte_year_mth >= to_number(substr(:szyearBeginDate,0,6));
```

int caseLookup(XMLElement *eligSub,XMLElement *eligNode, WORK_ST *wks_ptr)

```
        SELECT * FROM (
                SELECT a.*, rownum rnum
```

```
            FROM (
                SELECT
                a.nam_last,
                a.nam_first,
                a.nam_mid_init,
                id_medicaid,
                c.nam_last caseLName,
                c.nam_first caseFName,
                c.nam_mid_init caseMName,
                dte_birth"
                FROM    t_re_case a, t_re_case_xref b, t_re_base c
                WHERE  a.sak_case = b.sak_case
                AND    b.sak_recip = c.sak_recip
                AND    a.num_case = upper(' wks.szCaseNumber')
                AND    c.ind_active = 'Y'
                ORDER BY b.sak_recip ) a
            WHERE rownum <= szMaxRows )
WHERE rnum >= szMinRows
```

int nameLookup(XMLElement *eligSub, XMLElement *eligNode, WORK_ST *wks_ptr)

```
SELECT * FROM (
    SELECT a.*, rownum rnum
    FROM (
        SELECT
            id_medicaid,
            a.nam_first,
            a.nam_last,
            a.dte_birth,
            c.num_case
        FROM   t_re_base a, t_re_case_xref b, t_re_case c
```

```
            WHERE   a.sak_recip = b.sak_recip"
      AND   b.sak_case = c.sak_case"
      AND   a.nam_last like upper(substr('%s',1,5)) || '%c'"
      AND   a.nam_first like upper(substr('%s',1,7)) || '%c'"
      AND   c.rowid = (
                        SELECT g.rowid"
                  FROM t_re_case g, t_re_case_xref x"
                  WHERE g.sak_case = x.sak_case"
                  AND x.sak_recip = a.sak_recip"
                  AND x.dte_cert = (
                              SELECT max(dte_cert)
                        FROM t_re_case_xref
                              WHERE sak_recip = a.sak_recip)
                        AND rownum = 1)
                        /* dte_birth included if sDateDb non blank */
                        AND     dte_birth = sDateDb
                        AND     ind_active = 'Y'
                        ORDER BY a.nam_last, a.nam_first ) a
            WHERE rownum <= szMaxRows )
      WHERE rnum >= szMinRows
```

XMLElement *locInfoWeb(XMLElement *eligNode,WORK_ST *wks_ptr)

```
      SELECT  NVL(b.id_provider,'Invalid'),a.cde_service_loc,
            a.cde_loc, d.dsc_loc,a.dte_effective, a.dte_end
      FROM    T_RE_LOC a, T_PR_PROV b, T_RE_BASE c, T_RE_LOC_CODE d
      WHERE   a.sak_prov = b.sak_prov
            AND     c.id_medicaid = :id
            AND     c.sak_recip = a.sak_recip
            AND     a.cde_loc = d.cde_loc
            AND NOT(a.dte_effective < :search_eff
```

```
        AND     a.dte_end       < :search_eff)
        AND NOT(a.dte_effective > :search_end
        AND     a.dte_end       > :search_end)
ORDER BY a.dte_effective desc, a.dte_end desc;

SELECT  a.dte_admit, a.dte_term, b.id_provider, a.cde_service_loc
FROM    t_re_loc_dates a,t_pr_prov b, t_re_base c
WHERE   a.sak_prov = b.sak_prov
        AND     a.SAK_RECIP = c.sak_recip
        AND     c.id_medicaid = :id
        AND NOT(a.dte_admit     < :search_eff
        AND     a.dte_term      < :search_eff)
        AND NOT(a.dte_admit     > :search_end
        AND     a.dte_term      > :search_end)
ORDER BY a.dte_admit DESC, a.dte_term DESC;

SELECT
        b.id_medicaid,
        a.cde_type,
        a.amt_patnt_liab,
        a.dte_effective,
        a.dte_end
FROM    t_re_pat_liab a, t_re_base b
WHERE   a.sak_recip = b.sak_recip
        AND     b.id_medicaid = :id
        AND NOT(a.dte_effective < :search_eff
        AND     a.dte_end       < :search_eff)
        AND NOT(a.dte_effective > :search_end
        AND     a.dte_end       > :search_end)
ORDER BY a.dte_effective DESC, a.dte_end DESC;
```

7.4 Sample Request and Response

7.4.1 Request


```
<eligVerification txnElapsedTime="txnElapsedTime_0"
responseStatus="responseStatus_1" loc_info="true">

<eligRequest clientSecurityKey="EDI" transactionId="270"
ISAControlNumber="ISAControlNumber_2"

GSControlNumber="GSControlNumber_3" receiverAppName="receiverAppName_4"
transactionControlNumber="transacti"

tradingPartnerId="123456" referenceId="referenceIdreferenceIdreferenc">

<eligInfoSource entityCode="ent" entityTypeQualifier="e"
orgName="orgNameorgNameorgNameorgNameorgName"

firstName="firstNamefirstNamefirstNa" middleName="middleNamemiddleNamemiddl"
nameSuffix="nameSuffix"

sourceIdCodeQualifier="so"
sourceIdCode="sourceIdCodesourceIdCodesourceIdCodesourceIdCodesourceIdCodesou
rceIdCodesourceId" />

<eligInfoReceiver entityCode="ent" entityTypeQualifier="e"
orgName="orgNameorgNameorgNameorgNameorgName"

firstName="firstNamefirstNamefirstNa" middleName="middleNamemiddleNamemiddl"
nameSuffix="nameSuffix"

providerNumber="1000000323" receiverIdCodeQualifier="SV" />

<eligSubscriber entityCode="entityCode_0"
entityTypeQualifier="entityTypeQualifier_1" lastName="lastName_2"

firstName="firstName_3" middleName="middleName_4" nameSuffix="nameSuffix_5"
subscriberIdCodeQualifier="subscriberIdCodeQualifier_6"

subscriberId="000031719" subscriberSSN="subscriberSSN_8"
subscriberPatientAcctNumber="subscriberPatientAcctNumber_9"

referenceIdQualifier="referenceIdQualifier_10" referenceId="referenceId_11"
subscriberAddress1="subscriberAddress1_12"

subscriberAddress2="subscriberAddress2_13" subscriberCity="subscriberCity_14"
subscriberState="subscriberState_15"

subscriberZip="subscriberZip_16" subscriberCountry="subscriberCountry_17"
subscriberLocationQualifier="subscriberLocationQualifier_18"

subscriberLocationIdCode="subscriberLocationIdCode_19"
planSponsorName="planSponsorName_20"

subscriberDateOfBirth="subscriberDateOfBirth_21"
subscriberGender="subscriberGender_22"
newBornIndicator="newBornIndicator_24">

<eligSubscriberTrn traceTypeCode="t"
traceReferenceId="traceReferenceIdtraceReference"
traceAssigningEntityId="traceAssig"
```

```
traceAssigningEntityAddId="traceAssigningEntityAddIdtrace"
verificationNumber="100" />

<eligSubscriberDate fromDateOfService="20060101" toDateOfService="20060101"
/>

</eligSubscriber>

</eligRequest>

</eligVerification>
```


7.4.2 Response


```
<eligVerification txnElapsedTime="txnElapsedTime_0" responseStatus="A"
loc_info="true">

<eligRequest clientSecurityKey="EDI" transactionId="270"
ISAControlNumber="ISAControlNumber_2"

GSControlNumber="GSControlNumber_3" receiverAppName="receiverAppName_4"
transactionControlNumber="transacti"

tradingPartnerId="123456" referenceId="referenceIdreferenceIdreferenc">

<eligInfoSource entityCode="ent" entityTypeQualifier="e"
orgName="orgNameorgNameorgNameorgNameorgName"

firstName="firstNamefirstNamefirstNa" middleName="middleNamemiddleNamemiddl"
nameSuffix="nameSuffix"

sourceIdCodeQualifier="so"
sourceIdCode="sourceIdCodesourceIdCodesourceIdCodesourceIdCodesourceIdCodesou
rceIdCodesourceId" />

<eligInfoReceiver entityCode="ent" entityTypeQualifier="e"
orgName="orgNameorgNameorgNameorgNameorgName"

firstName="firstNamefirstNamefirstNa" middleName="middleNamemiddleNamemiddl"
nameSuffix="nameSuffix"

providerNumber="1000000323" receiverIdCodeQualifier="SV" />

<eligSubscriber lastName="PARRENT" middleName="middleName_4"
subscriberId="000031719" subscriberAddress1="subscriberAddress1_12"

subscriberCity="subscriberCity_14" planSponsorName="planSponsorName_20"
entityTypeQualifier="entityTypeQualifier_1"

subscriberIdCodeQualifier="MI" referenceId="referenceId_11"
subscriberState="subscriberState_15"

subscriberCountry="subscriberCountry_17" subscriberDateOfBirth="04/15/1975"
firstName="MARGIE"

subscriberAddress2="subscriberAddress2_13" subscriberLocationQualifier="CY"
subscriberLocationIdCode="055"

entityCode="entityCode_0" nameSuffix="nameSuffix_5"
subscriberSSN="subscriberSSN_8"
subscriberPatientAcctNumber="subscriberPatientAcctNumber_9"
```

```
referenceIdQualifier="referenceIdQualifier_10"
subscriberZip="subscriberZip_16" subscriberGender="F"

newBornIndicator="newBornIndicator_24">

<eligSubscriberTrn traceTypeCode="t"
traceReferenceId="traceReferenceIdtraceReference"
traceAssigningEntityId="traceAssig"

traceAssigningEntityAddId="traceAssigningEntityAddIdtrace"
verificationNumber="0706800001" />

<eligSubscriberDate fromDateOfService="01/01/2006"
toDateOfService="01/01/2006" />

<eligResponse eligType="Medicaid">

<eligInformation eligIndicator="1" coverageLevelCode="IND"
serviceTypeCode="30" insuranceTypeCode="MC" />

<eligDates dateQualifier="307" fromEligDate="01/01/2006"
toEligDate="01/01/2006" />

</eligResponse>

</eligSubscriber>

</eligRequest>

</eligVerification>
```

8 Appendix F – Documentation for Claim Status Request (270/ 271)

8.1 Introduction

The Claim Status Request function is invoked by the SoapInvokeMethod() which is initiated by the xml270276Handler that runs in unison with the xml270276srvrd.

Or it's directly called by the batch program edip276d.

The Claim Status Request function gets the input inquiry from the Caller method.

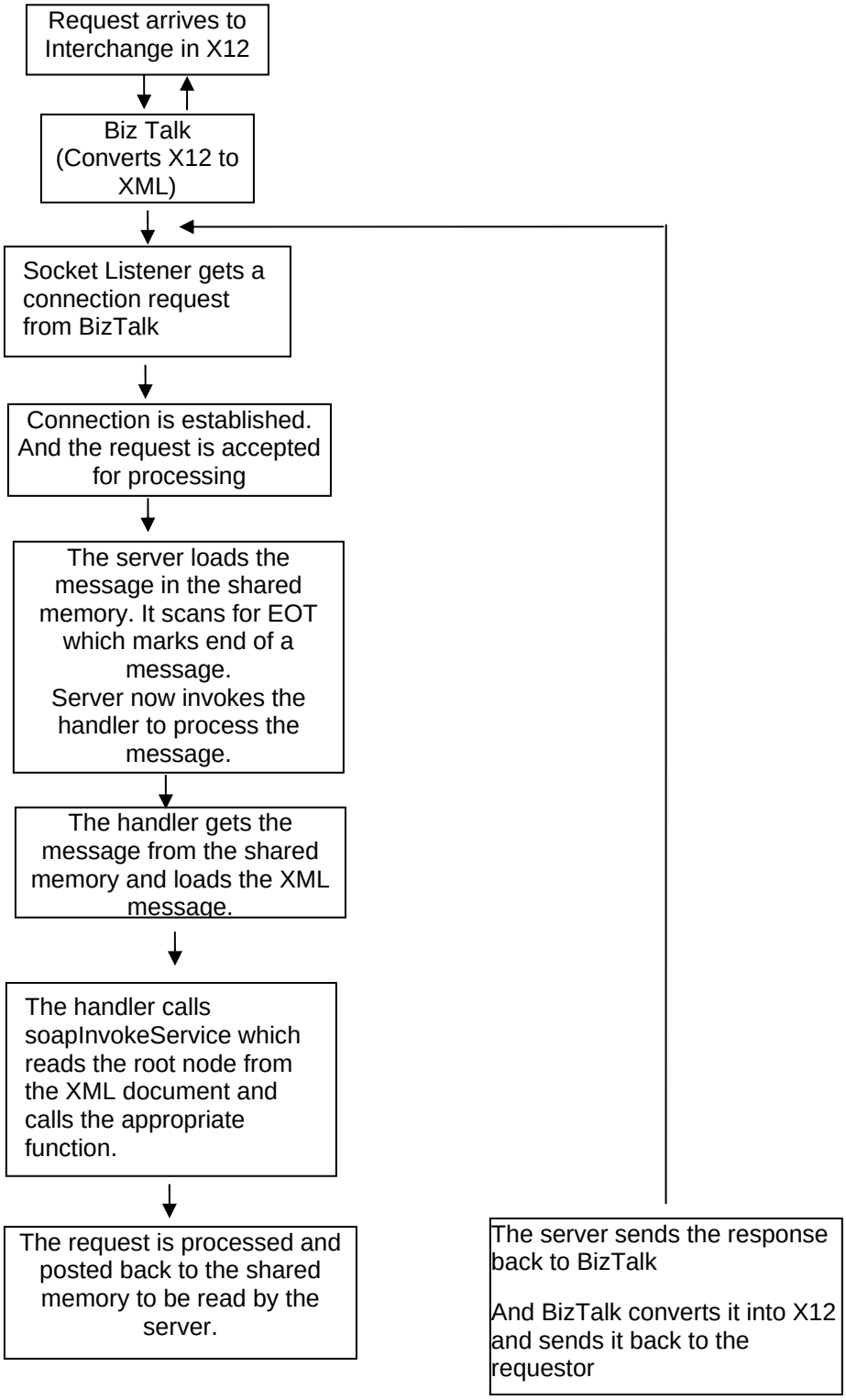
The main purpose of this module is claim inquiry. The claim inquiry transaction contains information on recipient and provider to uniquely identify the claims from the database. The response is sent back to the caller function. The handler takes the response and sends it back to the client in the appropriate format. (XML)

8.2 Process Flow

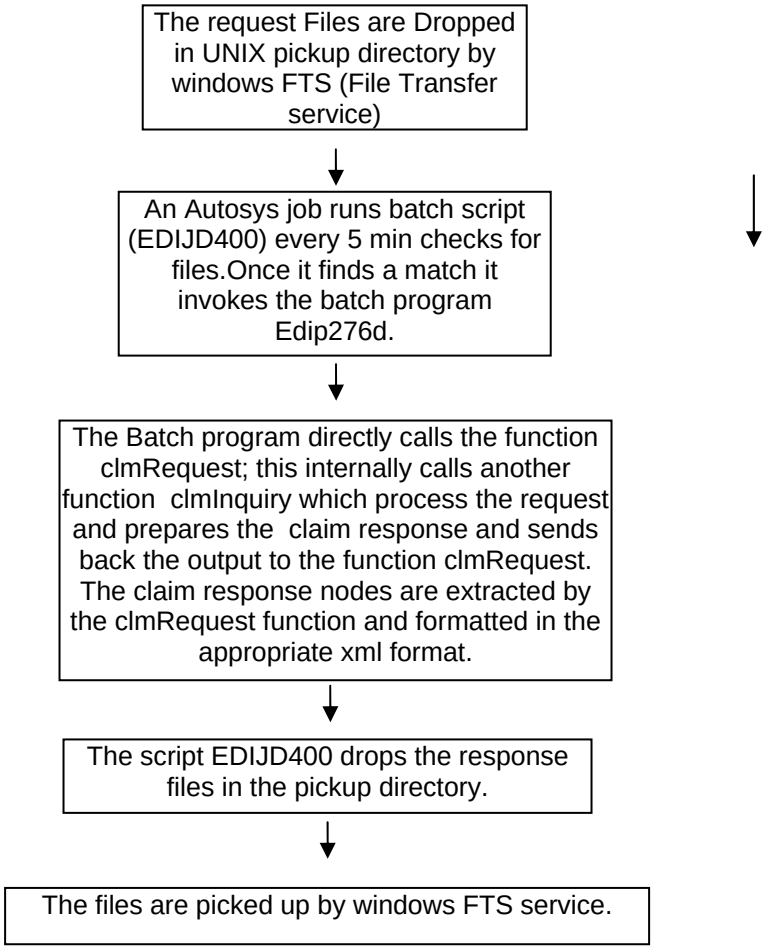
UNIX can process Claim Status Request by 2 methods:

- Interactive Processing (Socket Listener)
- Batch Processing.

8.2.1 2.1 Interactive (socket listener):



8.2.2 Batch Processing:



8.2.3 Detailed design description:

Program source: edip277d.c**,** clminq.sc

Makefile: libevs.so.mak

Shared library: libevs.so

Input Pointer to an XMLElement that contains a Claim inquiry Transaction

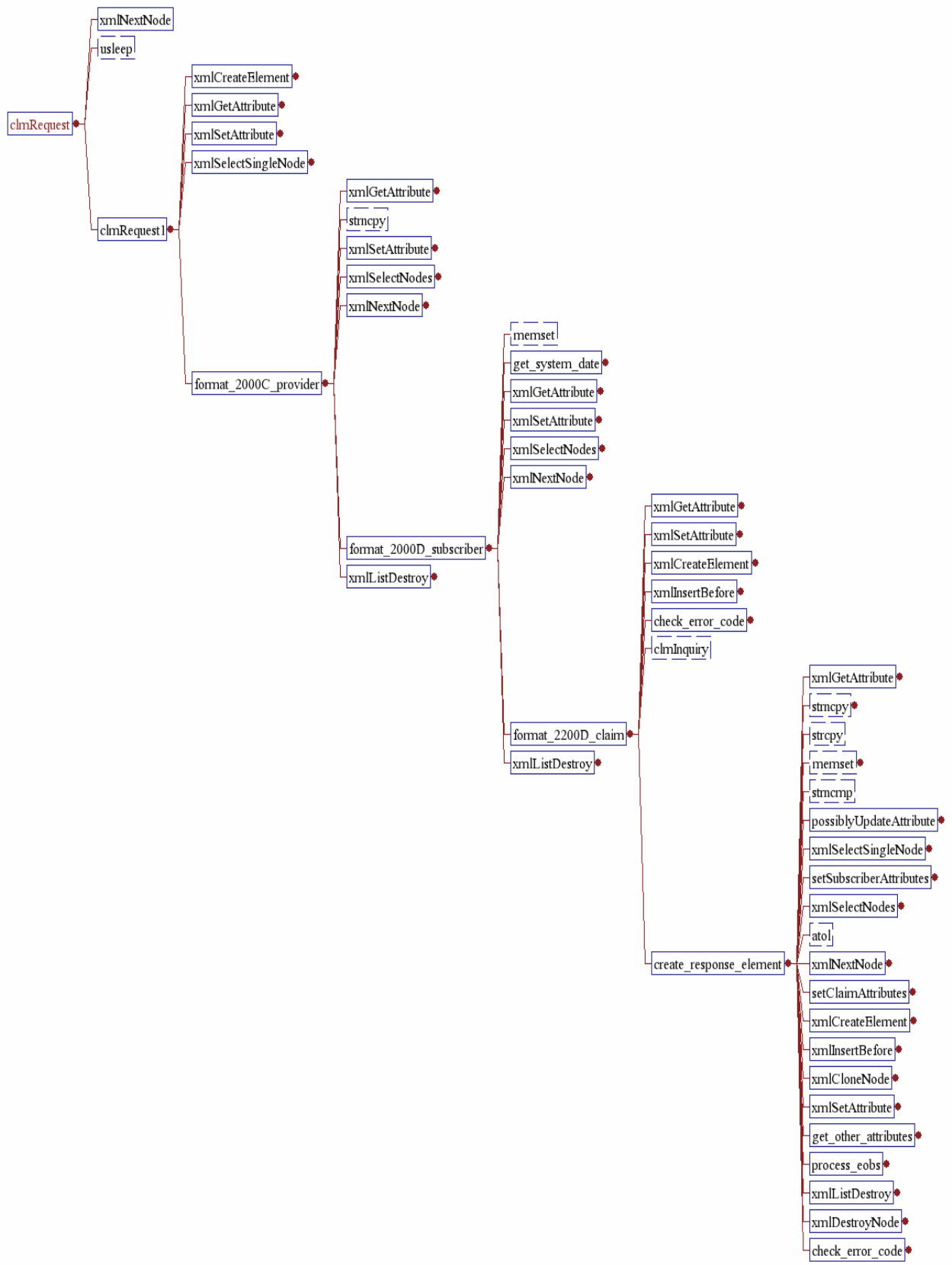
Output Pointer to an XMLElement that contains a

Claim response element.

8.2.4 Entry points/functions (edip277d.c)

- clmRequest() : Entry point or driver function for claim inquiry.
- clmRequest1() : Stores values in claim inquiry node from the 276 Claim status request node and goes through the hierarchy of the XML in order to get the data required for the clmInquiry.
- format_2000C_provider (): Retrieves necessary provider information from the provider Node of the 276 XML.
- format_2000D_subscriber (): Retrieves subscriber information from the subscriber node of the 276 xml necessary to run clmInquiry.
- format_2200D_claim(): Retrieves claim information from the claim node of
- The 276 file necessary to run clmInquiry. It uses this data to create the claimStatusResponse on 277 xml.
- create_response_element():Retrieves claim information from the claim node of
- the 276 file necessary to run clmInquiry. It uses this data to create the claimStatusResponse on 277 file.
- setSubscriberAttributes(): sets the Subscriber attributes in 277 xml based on the response recieved from function clmInquiry().
- process_eobs():Processes EOBs for each claim response received from clmInquiry().
- get_other_attributes(): sets the claim status and category codes in the 277 xml.
- check_error_code(): Checks error code returned from claim inquiry and populates appropriate fields on output file.
- setClaimAttributes(): set DOS and ICN attributes in ClaimStatusClaim node

8.2.5 Flow Diagram:



8.2.6 High level process flow:

The claim status request program takes a 276 inquiry transaction and returns the claim information for the subscriber. This takes the XML of the input inquiry (276) and edits the file by adding the claim response nodes if the transaction valid, other wise error nodes are added to the input XML and sent back. Hence the response (277) is the modified XML of the input transaction. The complete process involves checking /validating the fields in the input XML and editing the input file by adding the necessary attributes to prepare the response (277).

There is another important step to this logic, the function clmRequest() create another intermediate xml to be inputted to the function clmInquiry() which does the actual processing of the request.

The response that comes from clmInquiry() function has the claim and eob nodes which the driver function clmRequest() extracts and formats it into the 277 , i.e. the original request received from the caller.

The function clmRequest() is the starting point for the 276/277 processing, this function examines if the request came through interactive or batch process . There is flag which indicates the type of transaction.

If the transaction is batch we give a sleep time of 100 milliseconds, in order to give more priority to the interactive transactions over batch.

The main difference between interactive and batch is that batch transactions can have more that one request in a single 276 request where as interactive usually has 1 request.

The next step is to call the clmRequest1() function which checks if the necessary criteria is present to process the request , transaction id, client security key and it prepares the intermediate xml for clmInquiry() function with these attributes.

format_2000C_provider() function gets the provider information(provider id and provider qualifier) from the 276 request and adds them to the intermediate xml.

format_2000D_subscriber() gets the subscriber information like last, first names and subscriber id and adds it to intermediate xml. It loops through the claimStatusClaim nodes and calls the function format_2200D_claim() in order to add the claim related attributes to the intermediate XML.

format_2200D_claim() puts several fields like claimStatusEffectiveDate, billedamount,paidAmount.service dates, amountBilled etc. for each claimStatusClaim node in the 276 .

Now that the intermediate input is ready, the main function clmInquiry() is called for processing. Once the response is obtained from this function , the 277 is now prepared with the intermediate response we obtained from the clmInquiry function.

The response is formatted by the function **create_response_element().**

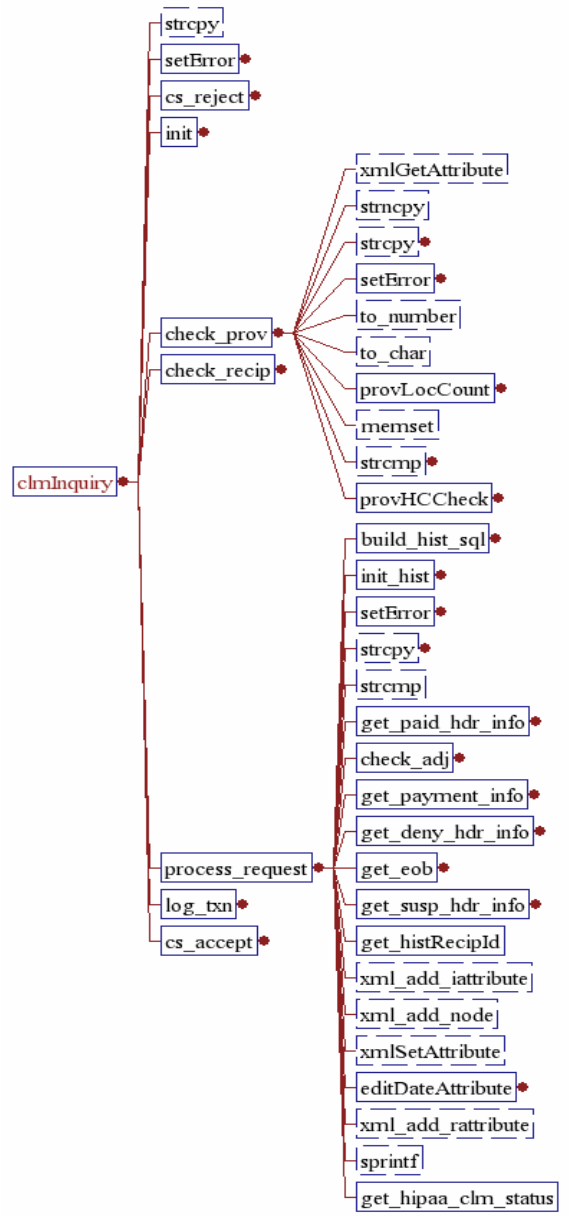
This function calls several functions like **process_eob()**, to put eob information in 277, **addSubscriberAttributes()** to add subscriber related information to 277, **setClaimAttributes()** to add ICN and dates of service to 277,get_other_attributes to add claim status codes in 277.

If there was any error in process the request by clmInquiry , the respective error information is added by calling the function **check_error_code().**

8.2.7 Entry points/functions (clmInq.sc)

- clmInquiry : This is the main entry point for this library.
- Init : Establish Database session, Validate EVS version, clientSecurityKey and transactionID and senderID.
- cs_reject : This will compose a error/reject response xml.
- cs_accept : This will compose a success / accept response xml with valid response status, version number, number of recipients and request date.
- check_prov : This function will validate the provider information and validate in database.
- provLocCount :gets the provider saks .
- provHCCheck :This function checks to see any of the locations listed for the provider id are tagged as a health care provider.
- check_recip: This function will validate the recipient information and validate in the database.
- process_request: This function will process the actual xml claim request.
- build_hist_sql: This will create the history on all the claims based on the particular request.
- init_hist : Initialize all the claim history structures so that the next database fetch can load the claim records.
- get_paid_hdr_info : This will retrieve the amount to be paid information.
- get_deny_hdr_info :This will retrieve the claim denial information.
- check_adj: This function will verify whether any adjustments are required.
- get_histrecipid :This function will retrieve the history of the recipient id's.
- get_hippa_clm_status :This function will retrieve the hippa claim status.
- get_eob : This particular function will retrieve the eligibility of benefits.
- get_payment_info: This particular function will get the payment type and information from the database.
- get_susp_hdr_info: This will retrieve the claim suspension information.
- log_txn: This function will log the transaction into the database.

8.2.8 Flow Diagram:



8.2.9 High level Process Flow:

Algorithm

- Read and extract the information from the claim request.
- Validate the mandatory fields.
- If validation failed, create a claim reject response xml.
- Check whether the provider information is valid and is present in database.
- If provider is not valid, generate an error response xml.

- Build an SQL dynamically to verify the claims in the database.
- If the data is invalid generate a reject claim response xml.
- Fire the Dynamic SQL and fetch the record set.
- If the Claim Status field from the data base is ‘P’
    - If the Claim Type field is ‘M’ or ‘B’ get the total reimbursement in the case of physician claim
    - If the Claim Type field is ‘D’ get the total reimbursement in the case of Dental claim.
    - If the Claim Type field is ‘P’ or ‘Q’ get the total reimbursement in the case of Pharmacy claim.
    - If the Claim Type field is ‘I’ or ‘L’ or ‘H’ or ‘A’ or ‘C’ get the total reimbursement in the case of UB claim.
    - Check whether any adjustment in the payment info is required.
    - Get the payment information.
- If the Claim Status field from the database is ‘D’ get the claim denied information.
    - If the Claim Type field is ‘M’ or ‘B’ get the denied information in the case of physician claim
    - If the Claim Type field is ‘D’ get the denied information in the case of Dental claim.
    - If the Claim Type field is ‘P’ or ‘Q’ get the denied information in the case of Pharmacy claim.
    - If the Claim Type field is ‘I’ or ‘L’ or ‘H’ or ‘A’ or ‘C’ get the denied information in the case of UB claim.
    - Check whether any adjustment in the claim are required.
    - Retrieve the Explanation of benefit, why the claim was denied.
- If the Claim Status field from the data base is ‘S’ get the claim Suspended Information.
    - If the Claim Type field is ‘M’ or ‘B’ get the claim suspended information in the case of physician claim
    - If the Claim Type field is ‘D’ get the claim suspended in the case of Dental claim.
    - If the Claim Type field is ‘P’ or ‘Q’ get the claim suspended in the case of Pharmacy claim.
    - If the Claim Type field is ‘I’ or ‘L’ or ‘H’ or ‘A’ or ‘C’ get the claim suspended in the case of UB claim.
    - Check whether any adjustment in the claim are required.
    - Retrieve the Explanation of benefit, why the claim was suspended.

- Compose the Claim Response xml.
- Add recipient information.
- Add Claim Response Information for the recipient.
- Add Payment Information along with payment types as ACH, Check or None.
- Add the Eligibility of Benefits.
- Add all the claim responses for all the claims and add a num claim response element.

8.2.10 SQL Queries and Context:

	SQL QUERY	PURPOSE
1	SELECT DATE_PARM_1 INTO :sqlDate FROM t_system_parms WHERE nam_program = 'HCNPIDTE' and to_number(to_char(sysdate, 'YYYYMMDD')) between dte_effective and dte_end; SELECT count(*), to_number(to_char(sysdate, 'YYYYMMDD')) INTO :count, :todaysDate FROM t_pr_svc_loc WHERE sak_prov_loc in (SELECT sak_prov_loc FROM t_pr_identifier WHERE id_provider = :sqlProvID AND sak_prov_loc > 0) AND ind_healthcare = 'Y';	Validating the Provider Information Check_prov()
2	SELECT sak_recip, nam_last, nam_first INTO : sakRecip, : namLast, : namFirst FROM t_re_base WHERE id_medicaid = : recipId;	Validating the Recipient information Check_recip()
3	SELECT sak_claim, sak_recip, num_icn_fl, to_char(dte_first_svc), to_char(dte_to_date), amt_billed, cde_clm_type, cde_clm_status, cde_partition_id FROM aim.t_hist_directory	Building the claim History Build_hist_sql()

	WHERE sak_prov = : sakProv AND sak_recip = : sakRecip AND cde_service_loc = ': svcLoc	
4	SELECT a. fld_name, to_char(a.dte_final) FROM aim histParId. tbl_name a WHERE a.sak_claim = : histSakClaim (table names and field names are as follows: tbl_name, "t_pd_phys_hdr"(M,B) fld_name, "amt_tot_reimb" tbl_name, "t_pd_dntl_hdr" (D) fld_name, "amt_paid" tbl_name, "t_pd_pharm_hdr" fld_name, "amt_tot_reimb"(P,Q) tbl_name, "t_pd_ub92_hdr(I,L,H,O,A,C) fld_name, "amt_paid")	Getting the Paid header information Get_paid_hdr_info()
5	SELECT sak_claim_adj FROM aim tbl_name WHERE sak_claim = : histSakClaim OR sak_claim_adj = : histSakClaim tbl_name : t_adj_phys_xref(M,B) t_adj_dent_xref(D) t_adj_phrm_xref(P,Q) t_adj_ub92_xref(I,L,H,O,A,C)	Getting the Adjustment information Check_adj()
6	SELECT a.sak_eob FROM aim histParId,.t_clm_eob_xref a, aim histParId.t_claim_error b WHERE a.sak_claim = : histSakClaim AND a.sak_claim = b.sak_claim	Getting the EOB Get_eob()

	AND a.sak_eob = b.sak_eob AND a.num_dtl = b.num_dtl AND a.ind_ra_print = 'Y' AND a.cde_stat_error = 'C' ORDER BY b.dte_generic, b.tme_stamp	
7	SELECT count(*) FROM aimhistParId. tbl_name a WHERE a.sak_claim = :histSakClaim Tbl_name: t_susp_phys_hdr(M,B) t_susp_dental_hdr(D) t_susp_phrm_hdr(P,Q) t_susp_ub92_hdr(I,L,H,O,A,C)	Getting the count of the suspended headers Get_susp_hdr_info()
8	SELECT id_medicaid, to_char(dte_birth), cde_sex INTO : histRecipId, : histDteBirth, : histCdeSex FROM t_re_base WHERE sak_recip = : histSakRecip;	Getting the history of the recipient id's Get_histRecipId()
9	SELECT cde_clm_cat_status INTO : histHcClmStatCat FROM t_clm_status_xref WHERE cde_clm_status = histClmStatus;	Getting the HIPAA claim status Get_hipaa_clm_status()
10	SELECT to_char(a.dte_final) FROM aim histParId. tbl_name a WHERE a.sak_claim = : histSakClaim Tbl_name: t_deny_phys_hdr(M,B) t_deny_dntl_hdr(D) t_deny_phrm_hdr(P,Q) t_deny_ub92_hdr(I,L,H,O,A,C)	Getting the deny header information Get_deny_hdr_info()

8.3 Sample Request and Response

8.3.1 Request


```
<claimStatus276>

<claimStatusRequest clientSecurityKey="KYMED" transacti²nId="276"
ISAControlNumber="906311" GSControlNumber="1"

tradingPartnerId="9900002003     ">

<claimStatusPayer entityCode="PR" entityTypeQualifier="2"
payerName="KYMEDICAID" payerIdCodeQualifier="PI"

payerId="KYMEDICAID"></claimStatusPayer>

<claimStatusReceiver entityCode="41" entityTypeQualifier="2" orgName="NORTON
HEALTHCARE" firstName=""

middleName="" nameSuffix="" receiverIdCodeQualifier="46"
receiverId="446566717"></claimStatusReceiver>

<claimStatusProvider entityCode="1P" entityTypeQualifier="2" orgName="LINCARE
INC" firstName="" middleName=""

namePrefix="" nameSuffix="" providerIdCodeQualifier="SV"
providerId="90234170"></claimStatusPrÔvider>

<claimStatusSubscriber birthDateQualifier="D8"
subscriberBirthDate="04/29/1926" subscriberGender="M" entityCode="QC"

entityTypeQualifier="1" orgName="LOPEZ" firstName="AUGUSTO" middleName=""
namePrefix="" nameSuffix=""

subscriberIdCÔdeQualifier="MI" subscriberId="7370075324">

<claimStatusClaim traceNumber="2007006001458"
submittedTotalChargeAmount="559.61" serviceFromDate="06/08/2006"

serviceToDate="06/08/2006" ICNNum="2007023002057"></claimStatusClaim>

</claimStatusSubscriber>

</claimStatusRequest>

</claimStatus276>
```


8.3.2 Response


```
<claimStatus276 txnElapsedTime="8.20"><claimStatusRequest
clientSecurityKey="KYMED" transacti u110 ?Id="277" ISAControlNumber="906311"
GSControlNumber="1"

tradingPartnerId="9900002003     "><claimStatusPayer entityCode="PR"
entityTypeQualifier="2" payerName="KYMEDICAID" payerIdCodeQualifier="PI"

payerId="KYMEDICAID" />

<claimStatusReceiver entityCode="41" entityTypeQualifier="2" orgName="NORTON
HEALTHCARE" firstName=""
```

```
middleName="" nameSuffix="" receiverIdCodeQualifier="46"
receiverId="446566717" />

<claimStatusPrÔvider entityCode="1P" entityTypeQualifier="2" orgName="LINCARE
INC" firstName="" middleName=""

namePrefix="" nameSuffix="" providerIdCodeQualifier="SV"
providerId="90234170" />

<claimStatusSubscriber birthDateQualifier="D8"
subscriberBirthDate="04/29/1926" entityTypeQualifier="1"

orgName="LOPEZ" middleName="" namePrefix="" subscriberIdCÔdeQualifier="MI"
subscriberId="7370075324"


subscriberGender="M" entityCode="QC" firstName="AUGUSTO" nameSuffix="">

<claimStatusClaim traceNumber="2007006001458"
submittedTotalChargeAmount="559.61" serviceFr_mDate="06/08/2006"

serviceToDate="06/08/2006" ICNNum="2007023002057">

<claimStatusResponse claimStatusCode1="3" statusCategoryCode1="F1"
claimStatusEffectiveDate="03/06/2007"

billedAmount="559.61" paidAmount="46.20" paidDate="07/21/2006"
payTypeCode="ACH" />

</claimStatusClaim> </claimStatusSubscriber></claimStatusRequest>

</claimStatus276>
```

9 Appendix G – Cold Harbor (DMS) Environment Provider Enrollment Detailed Design

9.1 Executive Summary

This document provides detailed information on the physical design of the Provider Enrollment BizTalk application that will be deployed into the Cold Harbor (DMS) environment. Current functionality includes support for initial provider enrollment, provider maintenance and an EFT response as a separate process. The First Health FIQM system will submit SOAP requests for either an initial enrollment or a maintenance update. This request will be forwarded by BizTalk to the interChange Provider Enrollment subsystem. InterChange will return either a provider number or an appropriate error message. This response will then be forwarded back to FIQM by means of a separate web service. The BizTalk Provider Enrollment application fully leverages the EAI framework for tracking, error handling, configuration management and logging support.

An optional short-term copy of all inbound and outbound messages can be archived on a central message server to allow for troubleshooting and manual restarts if necessary. In addition, as part of the standardized tracking from the EAI Framework record counts, message types, sender and receiver information and processing duration data will be logged to the database. The tracking instance data can be viewed at the Business Activity Services (BAS) web site using the query service.

9.2 Introduction

The following capabilities are provided in a standardized manner to Provider Enrollment from the EAI framework which is deployed as a prerequisite to Provider Enrollment and other BizTalk applications such as Prior Auth and Provider Enrollment. The EAI framework is discussed in detail in a separate document but a brief summary of available features is provided below:

Centralized configuration management for pipeline components, orchestrations and BAM settings.

Exception handling

General purpose logging

Data access for .NET components that need database access via ADO.NET

Short-term archiving of all inbound and outbound messages

Message tracking and balancing of inbound and outbound record counts.

There are three main technologies which enable the shared architecture:

A customized version of Enterprise Library 1.1 to provide configuration management, exception handling, data access and logging capabilities. Enhancements to the Enterprise Library include a custom exception policy class that exposes the error handling instance identifier (a GUID) to improve error tracking, an XML formatter for logging messages to the database, a BAM event sink to simplify BAM integration and improved packaging and installation so the library can be installed and used immediately. See the Enterprise Library for DDI EDI document for more information.

BizTalk 2004 BAM to provide tracking, visibility into application processing and measurement of key performance indicators (KPIs). Using a combination of the BAM API (via the Enterprise Library BAM Sink) and instrumented orchestrations, BAM activity data will be captured and stored in relational tables. In addition, automated SQL Server DTS packages allow this data to be aggregated into an OLAP cube and periodically archived off the live system.

Custom .NET pipeline components to archive inbound and outbound messages and log message tracking data as BAM related activities. Configuration and error handling for the pipeline components is also provided by Enterprise Library.

9.3 Purpose

The purpose of this document is to describe the physical design characteristics of the BizTalk applications Provider Enrollment that will be hosted in the Cold Harbor DMS environment.

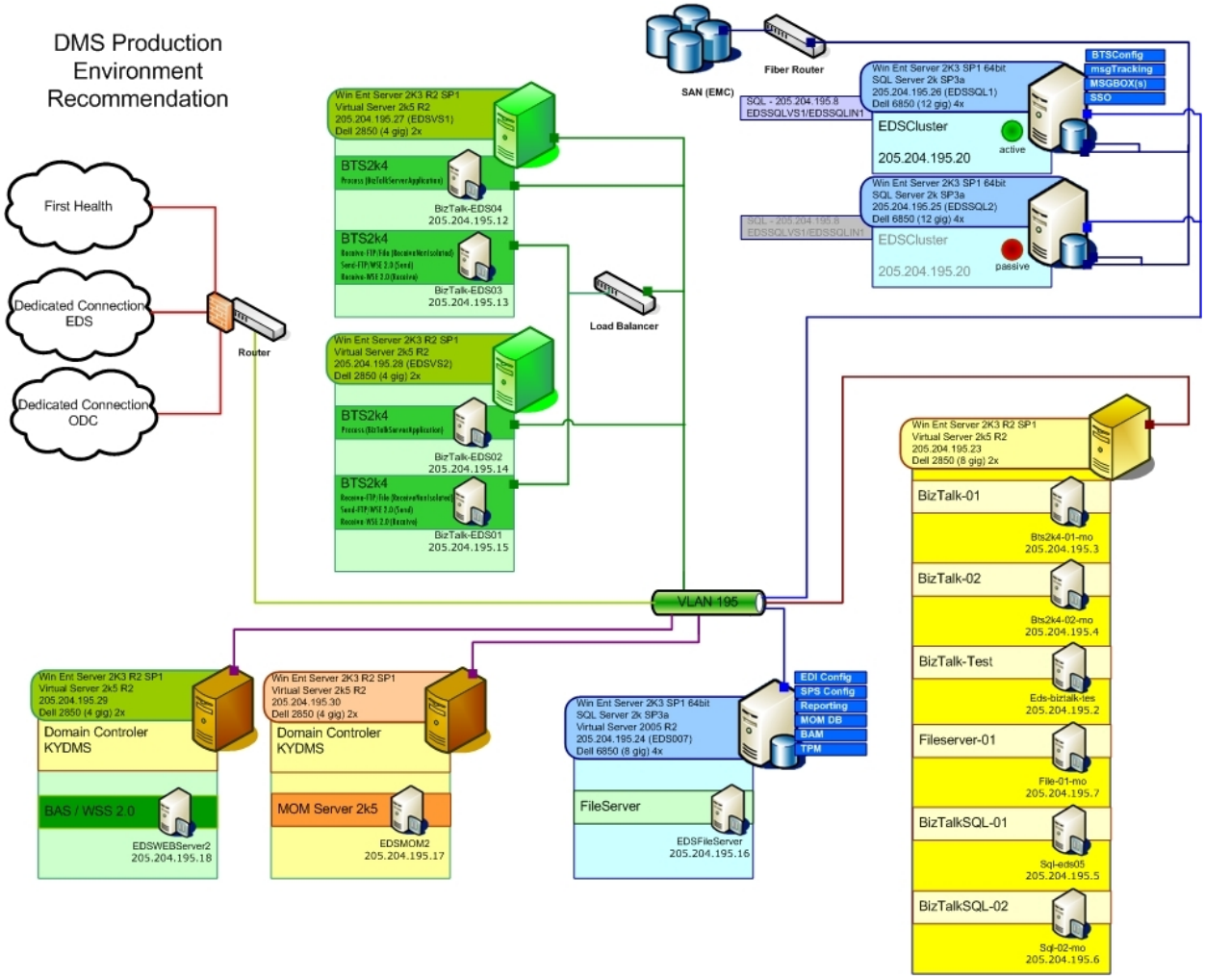
This application is second application to be deployed into the environment that will leverage the common application architecture for configuration management, logging, error handling, tracking and message archiving.

9.4 Scope

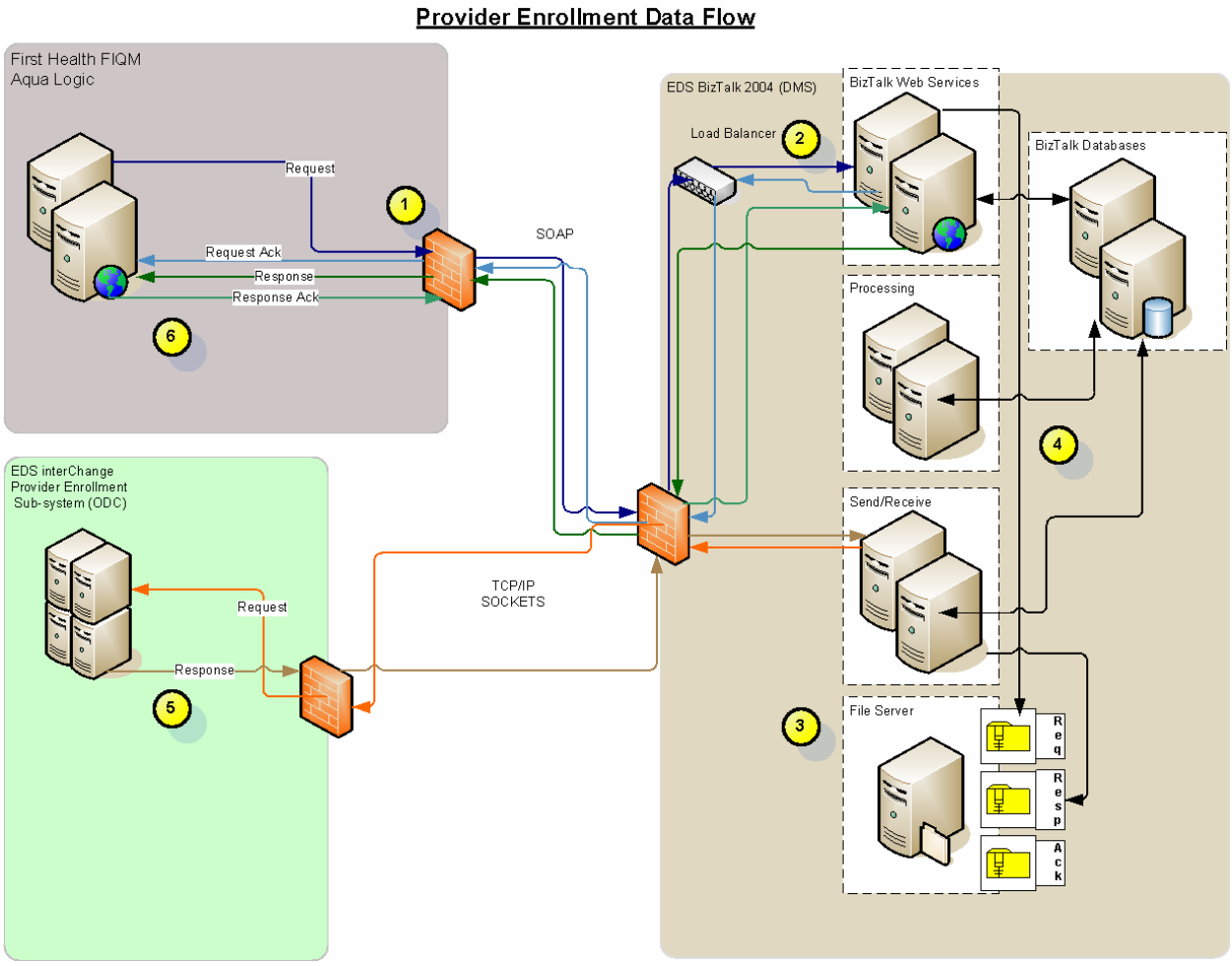
The scope of this document is the detailed physical design of the Provider Enrollment BizTalk application that will be deployed in the Cold Harbor DMS environment. Information on data flows, application structure and configuration is also provided.

9.5 Environment Setup

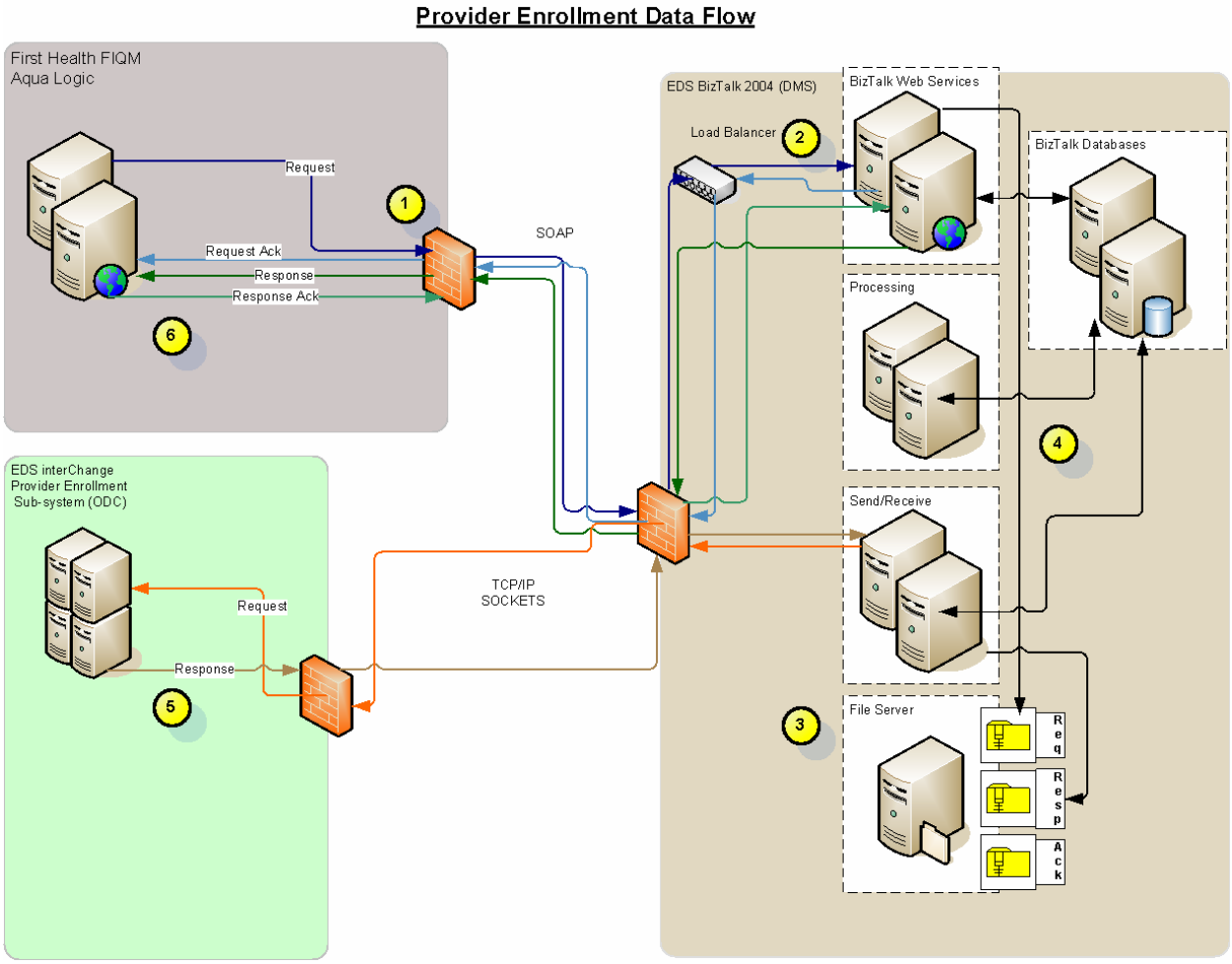
The following diagram describes the CH BizTalk environment setup associated with Provider Enrollment and other BizTalk EAI Transactions. CH BizTalk Environment has total 4 BizTalk 2003 Servers configured as part of BizTalk Group with two Clustered Messaging SQl servers , one non clustered Tracking SQL Server and File Server for short term Request, Response xml file achieving.



9.6 Data Flow Description - Enrollment



The following diagram describes the data flow associated with Provider Enrollment. In simple terms, SOAP web service requests are made from the FIQM Aqua Logic web service to a BizTalk web service. This request is transformed to an interChange compatible message that is sent via TCP/IP sockets or via Large Batch file using file Adaptor. Once interChange has processed the request, a provider number or a collection of errors will be returned. This response is transformed back into an FIQM compatible format and submitted by calling an Aqua Logic web service. All web service calls will be secured by implementing features of the OASIS web services security standard. Subsequent to the enrollment request or change, interChange may generate a separate EFT response as part of a weekly batch process. The EFT response is described in the next section of this document. Tracking and optional archiving of all BizTalk inbound and outbound messages is handled by the EAI framework.



FIQM via an Aqua Logic web service invokes the BizTalk Provider Enrollment application via a BizTalk published web service. Message level encryption, signing and authentication will be implemented based on the OASIS WS Security standard. X.509 certificates will be used to formally authenticate FIQM and BizTalk. Web Service policies (yet to be defined) will authorize approved services such as submitting a request or returning a response. Firewalls between FIQM and BizTalk have been configured to allow HTTP port 80 traffic to pass between both systems.

The BizTalk web services are load balanced so as to spread request handling between the two instances that are configured for SOAP send/receive. In addition, load balancing provides short-term high availability in case one of the instances becomes temporarily disabled. As a further security measure, IIS will be configured to only accept HTTP requests from a limited number of IP addresses. The profile of the web service on the network will also be reduced to prevent unauthorized persons from generating the web service definition language (WSDL) that describes the web service.

Once a message is received by BizTalk a pipeline component in the receive pipeline can optional make a copy of the message into an archive folder on the file server. In addition, the same component makes a call into BAM to start a new tracking activity. The activity is used to track orchestration processing milestones and information about each inbound and outbound message. See the EAI Framework General Design for detailed information on how messages are tracked using BAM.

Maps inside the BizTalk orchestration are used to transform the request into two outbound messages. The first map is used to generate a provider enrollment request that is compatible with interChange. The second map will generate a simple acknowledgement that will be returned to FIQM if no errors occur in the initial processing. If either map fails, a SOAP fault message is created and returned to FIQM. This fault indicates that a system level error occurred and that the request was not successfully forwarded to interChange. If Received Requests to interChange are more than 250 total record counts are sent using Batch route via the file adaptor using synchronous solicit-response pattern. If Received Requests to interChange are less than 250 Total record counts are sent via the TCP/IP Sockets adapter using a synchronous solicit-response pattern. The send pipeline using the sockets adapter has been configured to keep trying the request in case the listener on the interChange side is not immediately available. The orchestration has also been profiled using the Tracking Profile Editor (TPE) so that processing milestones can be monitored. Note that all inbound and outbound messages that enter and leave the orchestration are tracked. The initial received message for a running process will be converted to a multi-part message. This is a message that consists of the message body and a separate part, the BAMPart which is used to coordinate tracking activities between the receive pipeline, the orchestration and the send pipeline.

InterChange is expected to return the real time enrollment response within 2-3 seconds and it should be sent on the same port connection that was established to submit the request. The response will be mapped back to an FIQM compatible format and submitted to FIQM by invoking a separate Aqua Logic web service. This approach keeps FIQM disconnected and loosely coupled with BizTalk and makes the request-response pattern scalable and fault tolerant. Again, BizTalk will keep trying the Aqua Logic web service in case it is not immediately available.

InterChange is expected to return the Batch enrollment response after processing to a folder where BizTalk watches for response using File Adaptor. The response will be mapped back to an FIQM compatible format and submitted to FIQM by invoking a separate Aqua Logic web service. PE BizTalk Orchestration uses Correlation concept for handling the large batch PE files. This approach keeps FIQM disconnected and loosely coupled with BizTalk and makes the request-response pattern scalable and fault tolerant. Again, BizTalk will keep trying the Aqua Logic web service in case it is not immediately available.

The Aqua Logic web service that receives the enrollment response will return a simple acknowledgement that the response was safely received. BizTalk will track this acknowledgement and optionally archive it.

PE BizTalk Process has separate Reprocess orchestration for reprocessing failed Provider Enrollment. This Reprocess PE orchestration will be initiated by PE Request xml files which are pulled from the PE Request archive folders. This process will avoid resubmission of Error PE Request incase of errors in PE Orchestration process.Data Flow Description – Enrollment Sequence Diagram

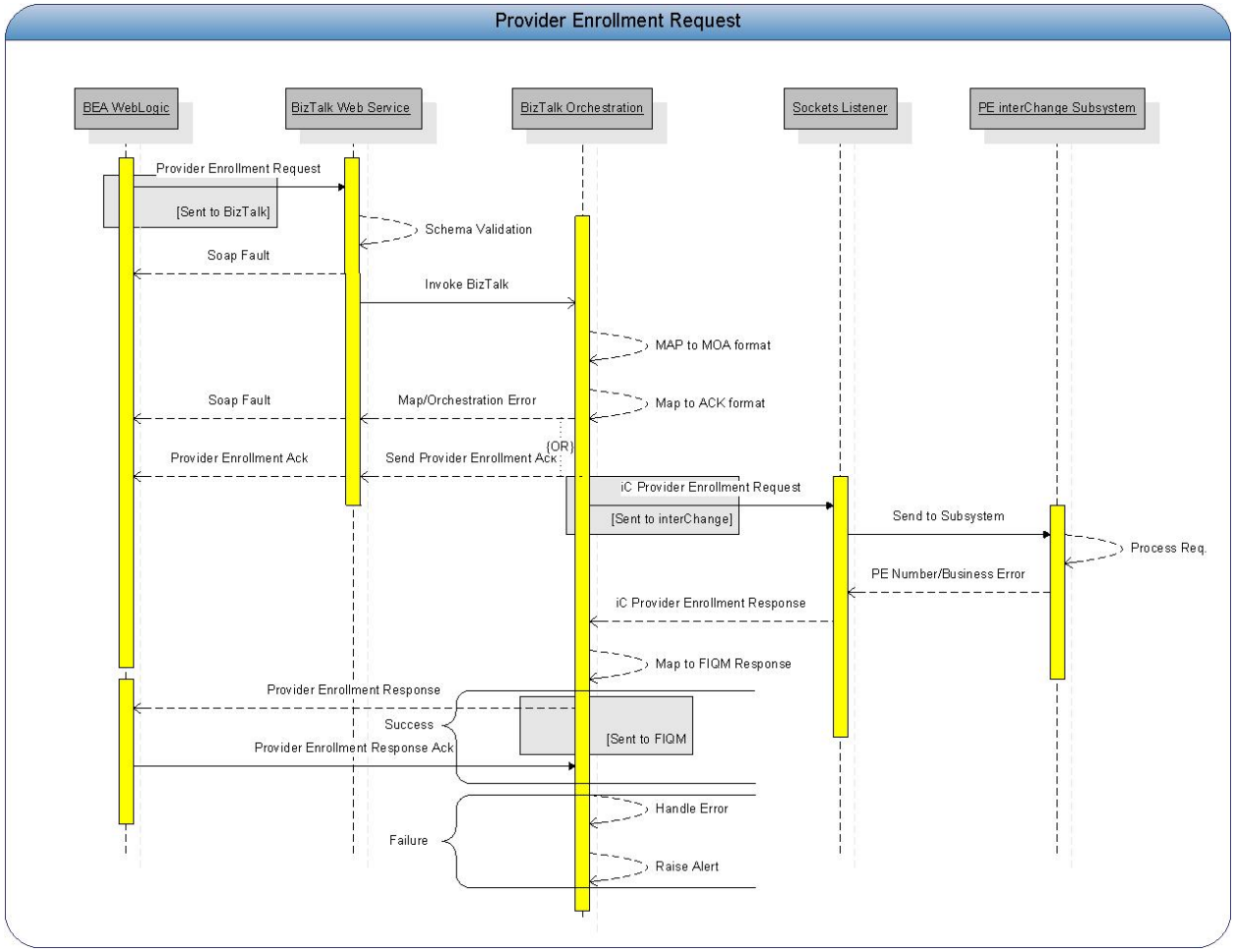
The following sequence diagram shows the key points of Provider Enrollment processing. Key points of the pattern are as follows.

Interactions between FIQM and BizTalk are disconnected and asynchronous to allow for flexibility, scalability and fault tolerance in the case of system failure.

SOAP web services are used as the transport between BizTalk and FIQM as they are well understand, easy to setup and they provide support for message/transport level encryption and security.

Interactions between BizTalk and interChange is based on TCP/IP sockets as it is the currently the recommended way for BizTalk applications in Cold Harbor to communicate with a subsystem.

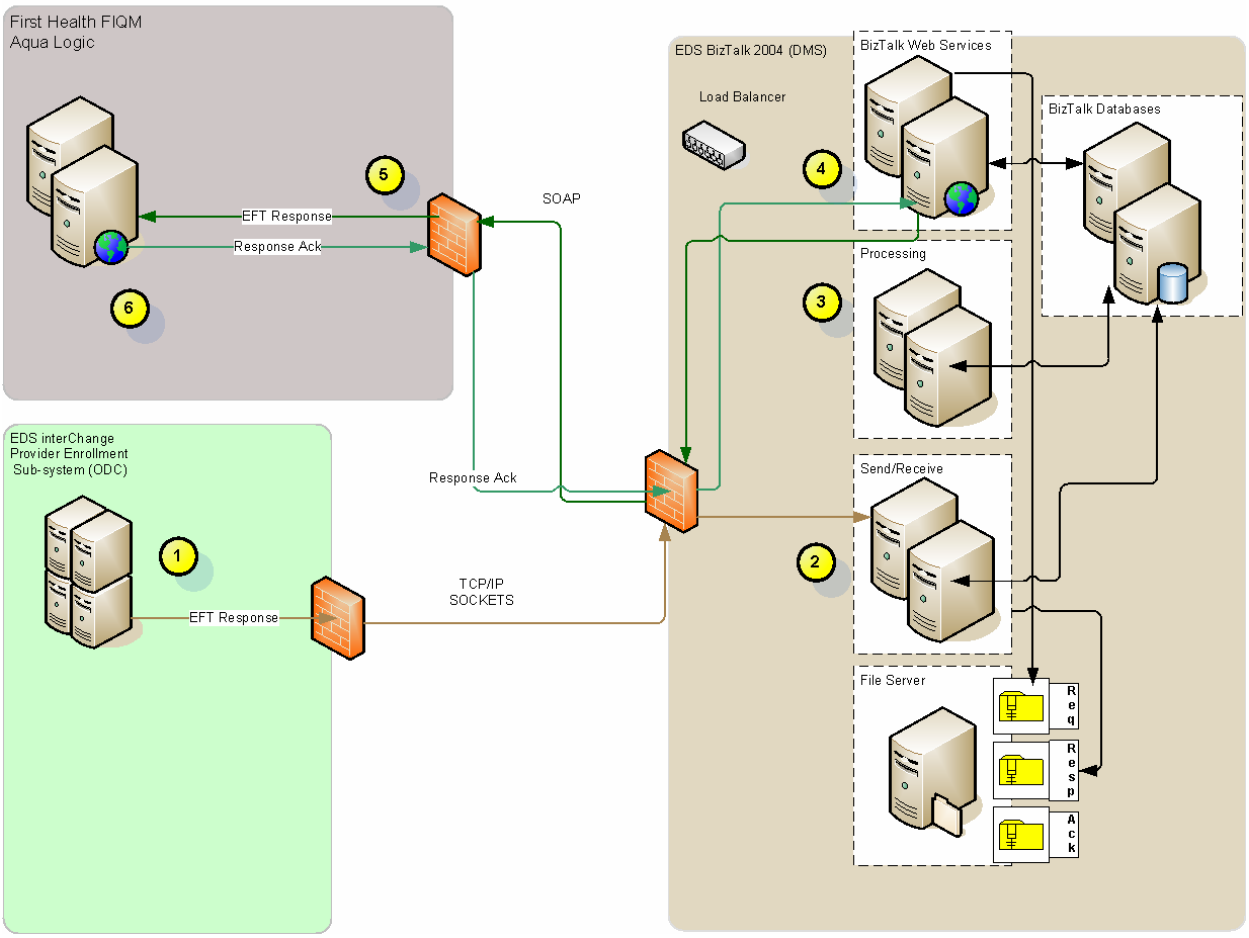
Systems that are sending data between architectural tiers are responsible for making multiple retries and making a best effort to deliver the request in the event that a receiver is currently not available.



9.7 Data Flow Description – EFT Response

The following diagram describes the data flow associated with the EFT response. This process is executed as part of a batch process that executes weekly under interChange. It is expected that interChange will send individual EFT responses to BizTalk so they can be forwarded to FIQM. As in the Provider Enrollment scenario, communication between interChange and BizTalk is via TCP/IP sockets and SOAP between BizTalk and FIQM. All web service calls will be secured by implementing features of the OASIS web services security standard. The Aqua Logic web service is also expected to return a simple acknowledgement that it has received the request. Tracking and optional archiving of all BizTalk inbound and outbound messages is handled by the EAI framework.

Provider Enrollment EFT Response



As part of a weekly batch cycle, inter FIQM will send individual EFT responses back to BizTalk so they can be forwarded back to FIQM. Again, the interChange sub-system will communicate via TCP/IP sockets to submit the EFT response. A custom Sockets adapter for BizTalk is listening for interChange to send a message based on an agreed IP address and port number. Once the adapter successfully receives an EFT response it will be submitted into BizTalk. Note that this receive port has a secondary receive location that can receive an EFT response from a dropped file.

Using a custom receive pipeline component, BizTalk will start a new BAM root level tracking activity and a separate file/message level activity that is related to the root. Again, the EAI

framework is responsible for adding a separate BAMPart to the message so the orchestration will receive a multi-part message that coordinates tracking activities and enables TPE instrumentation of the orchestration.

A map inside the orchestration is executed to convert the EFT response to an FIQM compatible format. If an error occurs during this mapping operation, processing will be terminated and error messages will be written to the event log and DDIEDI database.

If no orchestration errors occur, BizTalk will invoke the Aqua Logic web service for EFT responses. This is a separate web service to the one used for Provider Enrollment responses. Based on the pattern described below, BizTalk will continue to try the web service in case it is not immediately available. The outbound message is tracked by BAM and can be optionally archived onto the file server.

Once FIQM successfully receives the EFT response, it will echo back a simple acknowledgement which BizTalk will again track and optionally archive.

9.7.1 Data Flow Description – EFT Response Sequence Diagram

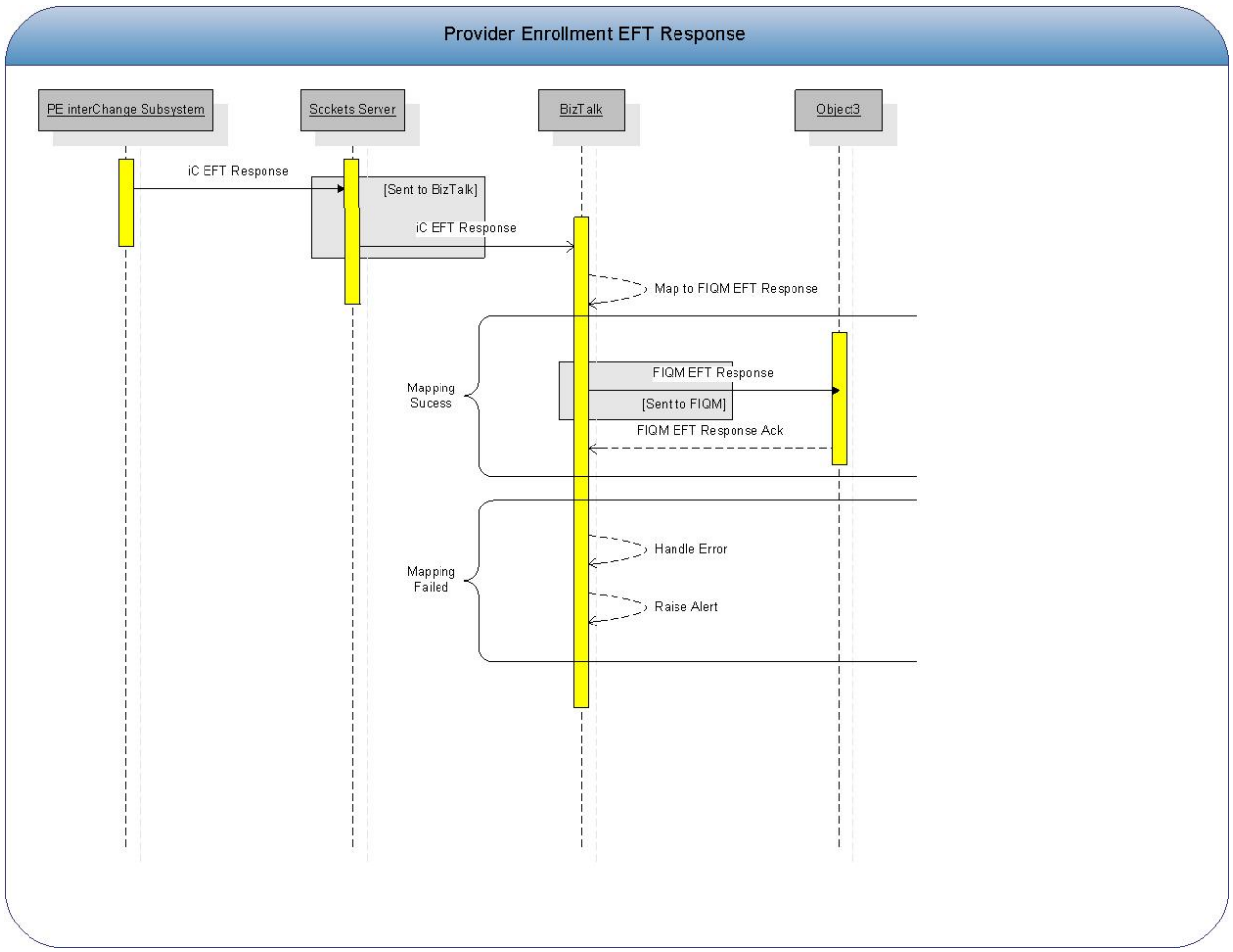
The following sequence diagram shows some key points of Provider Enrollment EFT processing. Key points of the pattern are as follows.

Interactions between FIQM and BizTalk are disconnected and asynchronous to allow for flexibility, scalability and fault tolerance in the case of system failure.

SOAP web services are used as the transport between BizTalk and FIQM as they are well understand, easy to setup and they provide support for message/transport level encryption and security.

Interactions between BizTalk and interChange is based on TCP/IP sockets as it is the currently the recommended way for BizTalk applications in Cold Harbor to communicate with a subsystem.

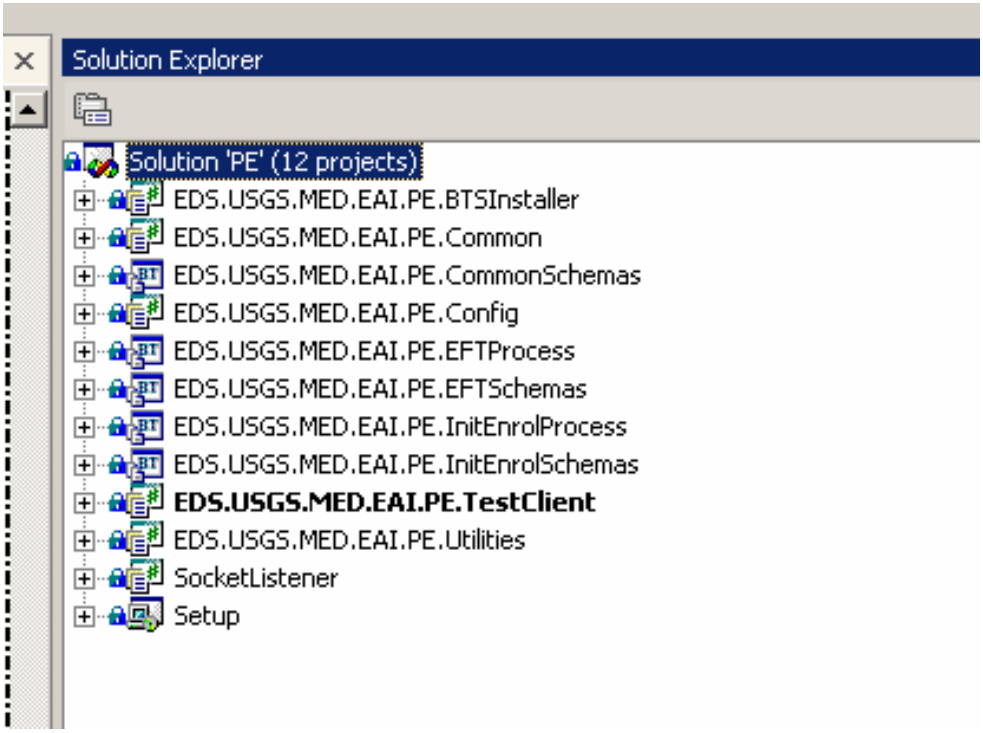
Systems that are sending data between architectural tiers are responsible for making multiple retries and making a best effort to deliver the request in the event that a receiver is currently not available.


Provider Enrollment EFT Response
PE interChange Subsystem
Sockets Server
BizTalk
Object3
iC EFT Response
[Sent to BizTalk]
iC EFT Response
Map to FIQM EFT Response
Mapping Sucess
FIQM EFT Response
[Sent to FIQM]
FIQM EFT Response Ack
Mapping Failed
Handle Error
Raise Alert

9.8 Application Overview

The aim of the following section is to provide a brief overview of the Provider Enrollment solution which contains the .NET and BizTalk artifacts for both Provider Enrollment and EFT Response although the functionality is separated into separate projects.

9.8.1 BizTalk projects



- EDS.USGS.MED.EAI.PE.BTSInstaller – the solution installer that requires on a base installer found in EDS.USGS.MED.Utilities.Install. The main function of this project is to fix the BizTalk bindings and invoke the base installer to deploy, bind and start the solution;
- EDS.USGS.MED.EAI.PE.Common – Common base classes for the solution including the Product class that controls versioning of .NET assemblies;
- EDS.USGS.MED.EAI.PE.CommonSchemas – Common schemas shared between EFT and Provider Enrollment and any subsequent PE related applications;
- EDS.USGS.MED.EAI.PE.Config – Enterprise Library generated project to provide design-time and run-time configuration settings;
- EDS.USGS.MED.EAI.PE.EFTProcess – BizTalk assembly containing the EFT orchestration;
- EDS.USGS.MED.EAI.PE.EFTSchemas – BizTalk assembly containing the EFT schemas, maps and pipelines;
- EDS.USGS.MED.EAI.PE.InitEnrollProcess – BizTalk assembly containing the enrollment orchestration;

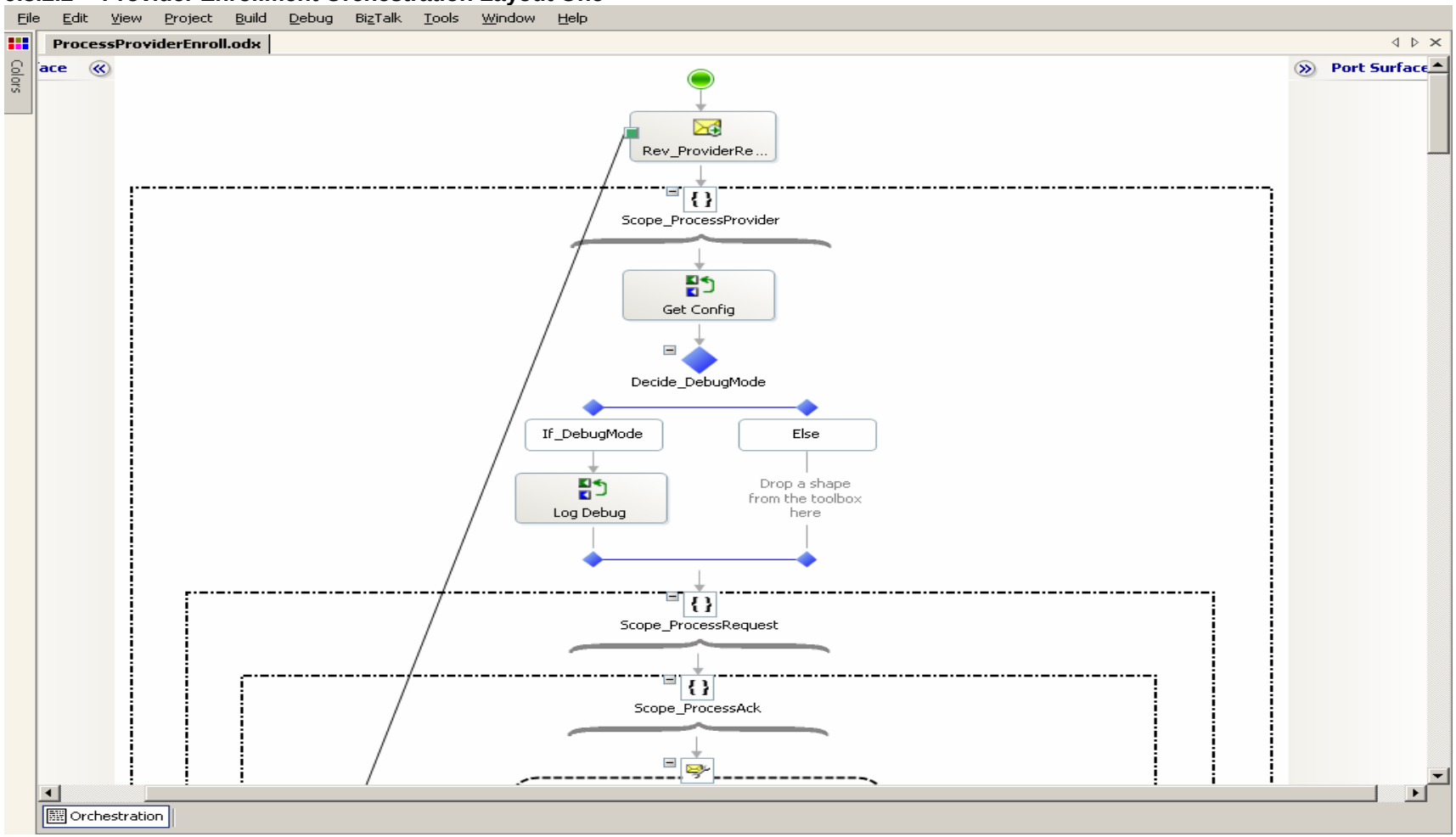
- EDS.USGS.MED.EAI.PE.InitEnrollSchemas – BizTalk assembly containing the enrollment schemas, maps and pipeline; and,
- EDS.USGS.MED.EAI.PE.Utilities – .NET utility classes such as orchestration helpers and code for scripting functoids.

9.8.2 Provider Enrollment Orchestration

9.8.2.1 Description

The PE request is received by a SOAP web service and configuration information is retrieved. The initial request has been converted by a pipeline component into a multi-part message with the original message and a BAMPart with the tracking activity IDs. If debug mode is enabled, the incoming message is written to the debug log (a flat-file). The request is mapped to the MOA format and a simple ACK format.

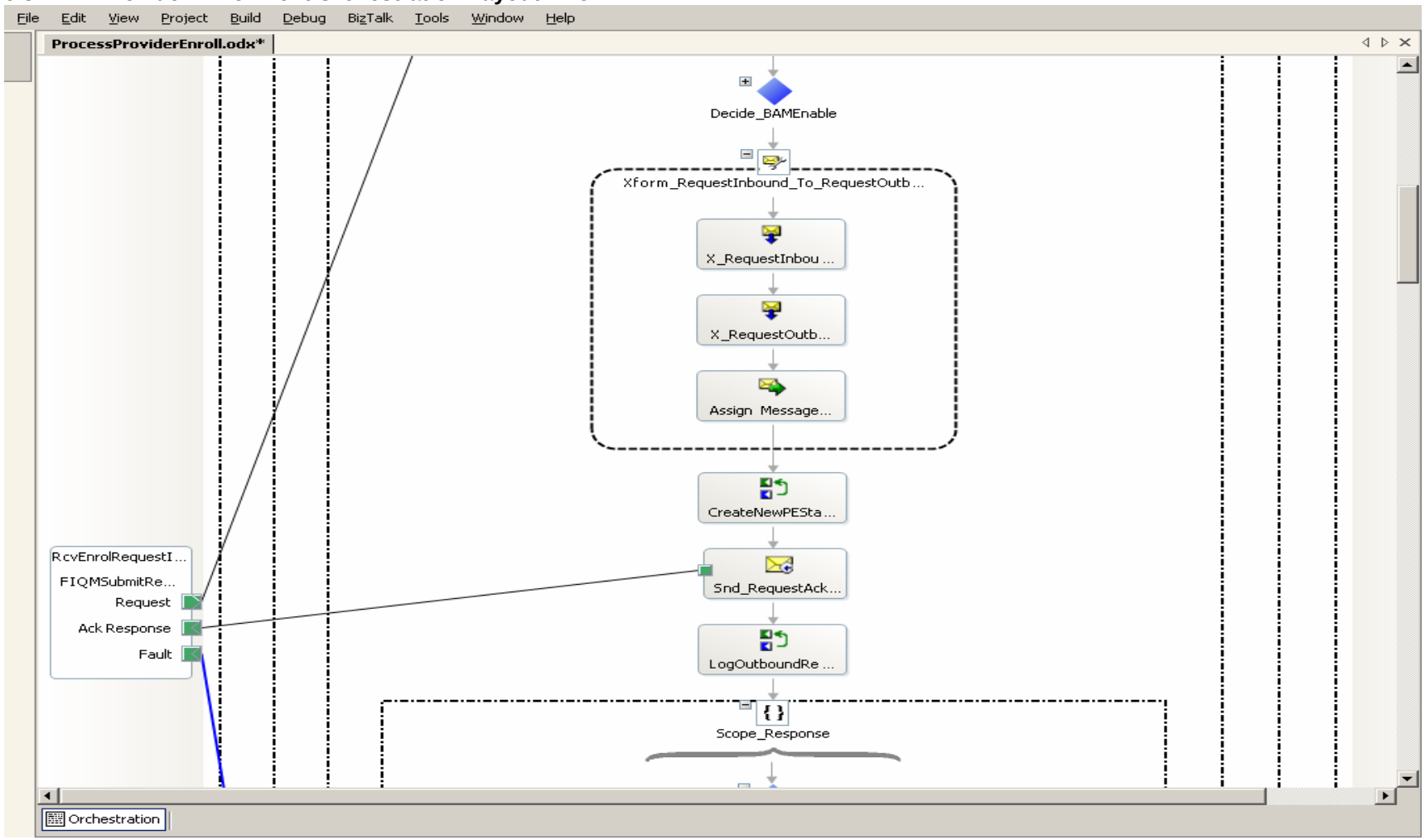
9.8.2.2 Provider Enrollment Orchestration Layout One



9.8.2.3 Description

If the mapping operation in the prior step is successful, the ACK is returned as a response message. Prior to sending acknowledgement PE Orchestration calls Custom Tracking library which creates a new record in PE Tracking Table which is used in Reprocessing failed Provider Enrollments. If an exception is thrown, a custom SOAP fault is constructed and returned instead. The SOAP fault indicates to FIQM that there was an internal failure and the request could not be forwarded to interChange.

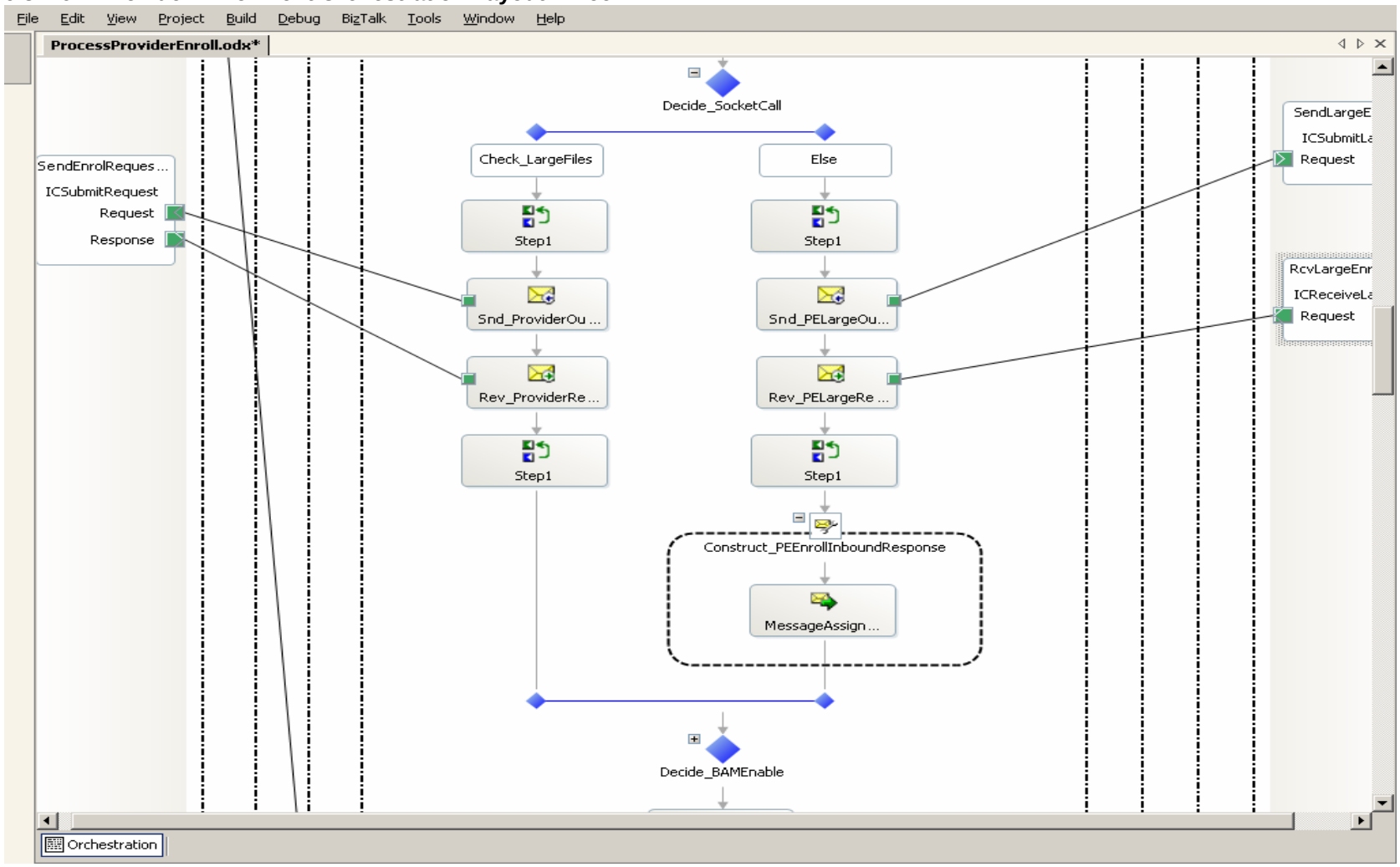
9.8.2.4 Provider Enrollment Orchestration Layout Two



9.8.2.5 Description

The PE request is sent to interChange using a synchronous solicit-response pattern and file send , Receive ports The PE request now in MOA format is checked for total record counts and if count is less than 250 record, is sent to interChange using synchronous solicit pattern via custom socket adaptor call. The PE request now in MOA format is checked for total record counts and if count is more than 250 record, is sent to interChange using file send and receive ports using correlation Batch Process.

9.8.2.6 Provider Enrollment Orchestration Layout Three

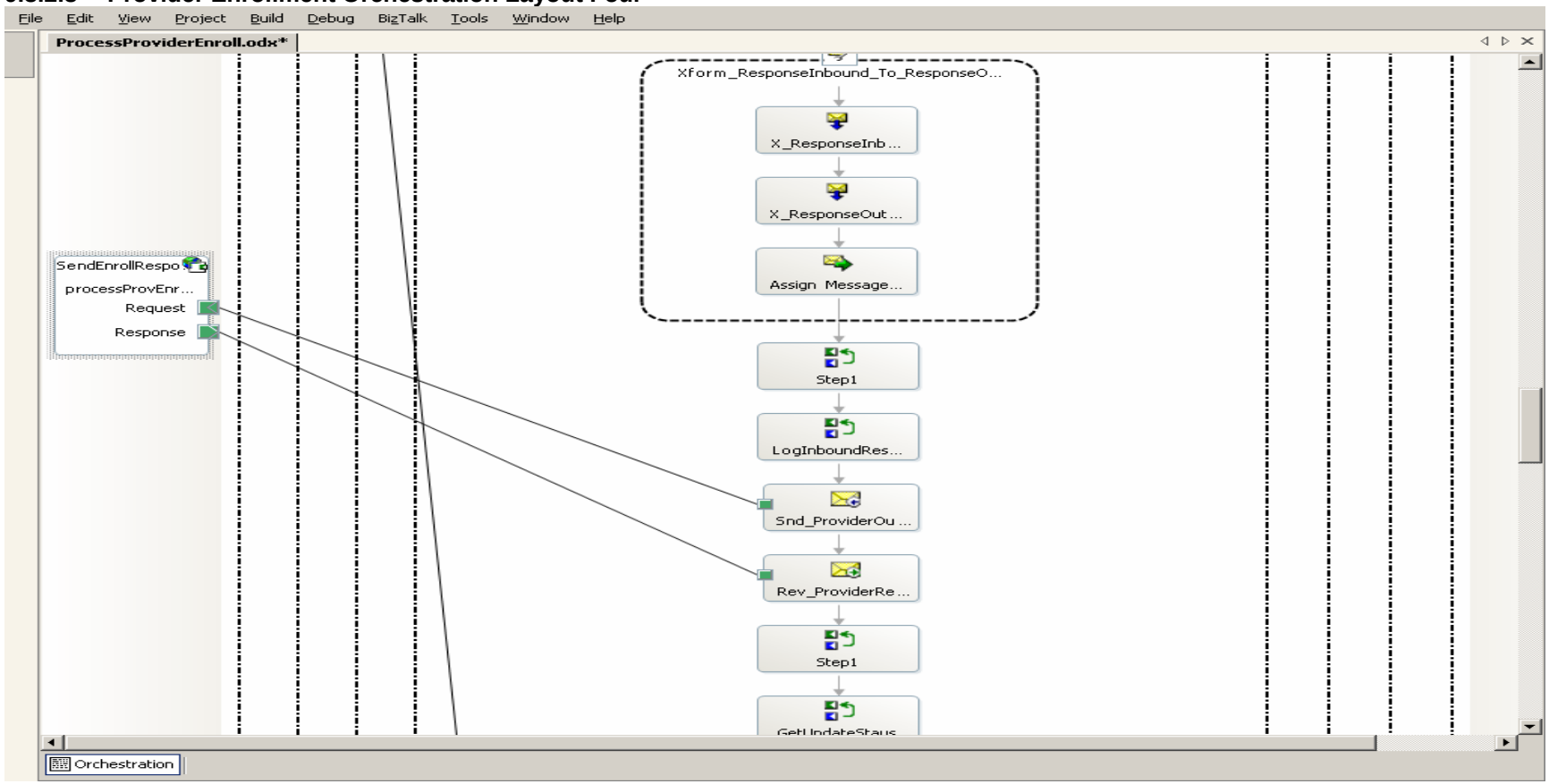


9.8.2.7 Description

The response from interChange is mapped back into FIQM format. The response message is tracked automatically by a pipeline component but must be manually added as a related activity to the root. This is because subsequent/intermediate receives are unaware of the root activity Id at the time the message is received. The response message which is in FIQM format is send to FIQM by using synchronous solicit-response pattern – calling FIQM Web service call and receives the acknowledgement from FIQM Web Service. The response acknowledgement is tracked automatically by a pipeline component and archived optionally. Once the response acknowledgment is received PE Orchestration calls the Custom Tracking Library for updating the completion status to Tracking record created in Custom tracking table.Provider Enrollment Orchestration Layout Four

Note: The response acknowledgement is tracked automatically by a pipeline component and archived optionally. Once the response acknowledgment is received PE Orchestration calls the Custom Tracking Library for updating the completion status to Tracking record created in Custom tracking table.

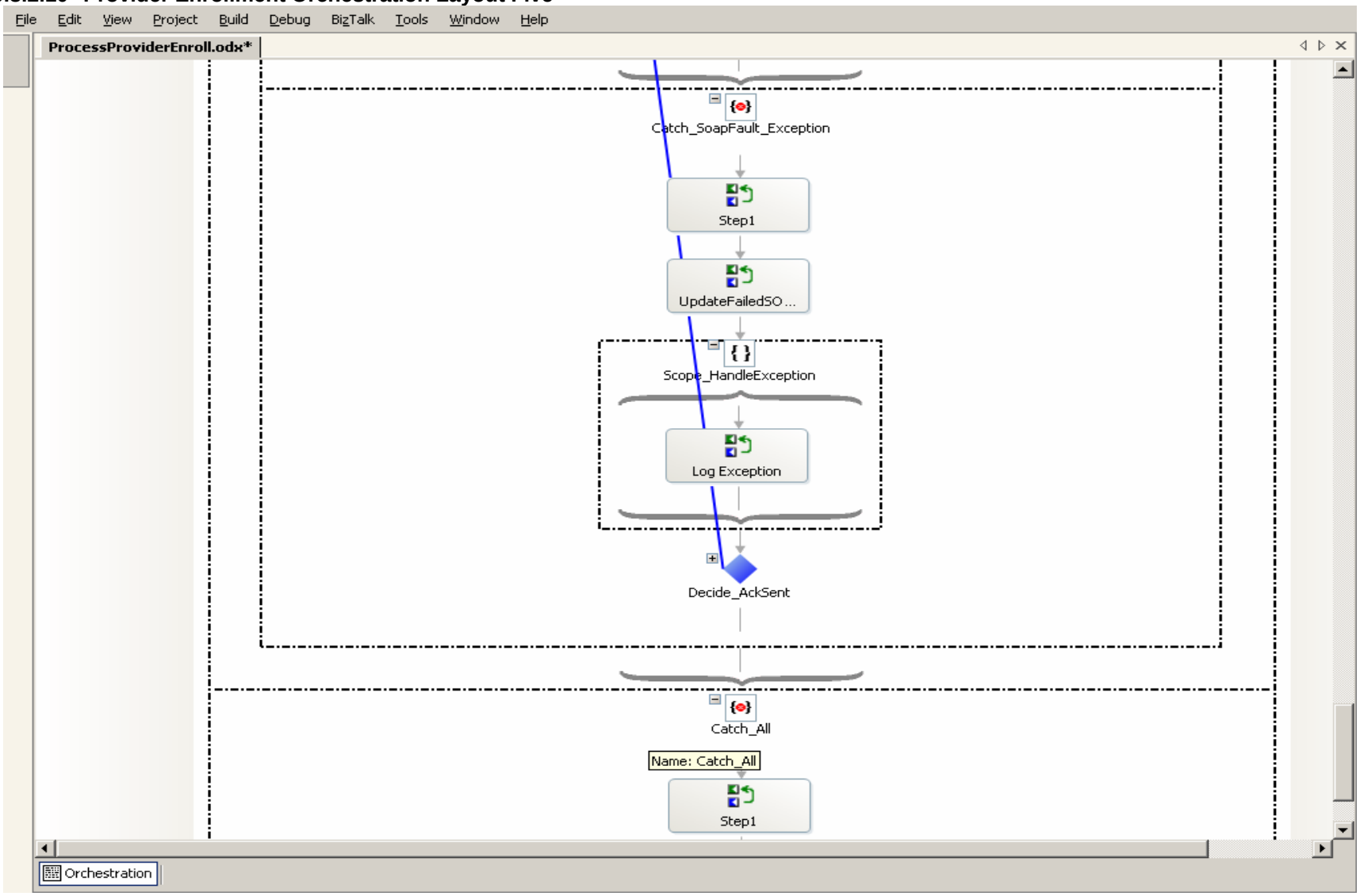
9.8.2.8 Provider Enrollment Orchestration Layout Four



9.8.2.9 Description

Multiple exception handlers are provided to catch either inbound or outbound message processing errors.

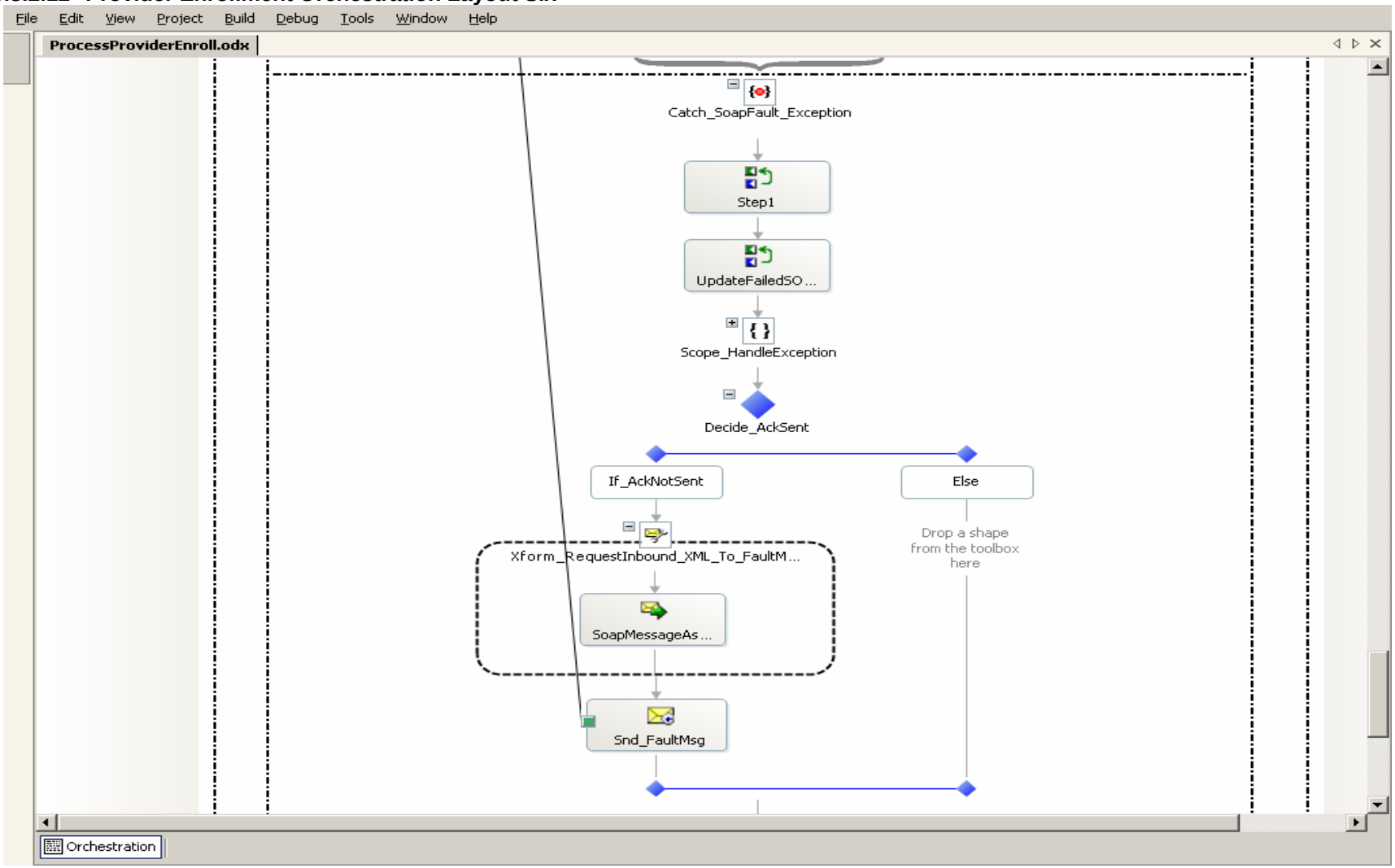
9.8.2.10 Provider Enrollment Orchestration Layout Five



9.8.2.11 Description

Note that the exception handler associated with processing the original request must contain conditional logic to ensure that only the ACK or a fault message is sent as part of the initial-request response. The BizTalk "succeeded" function is used to test if the process request transaction has failed.

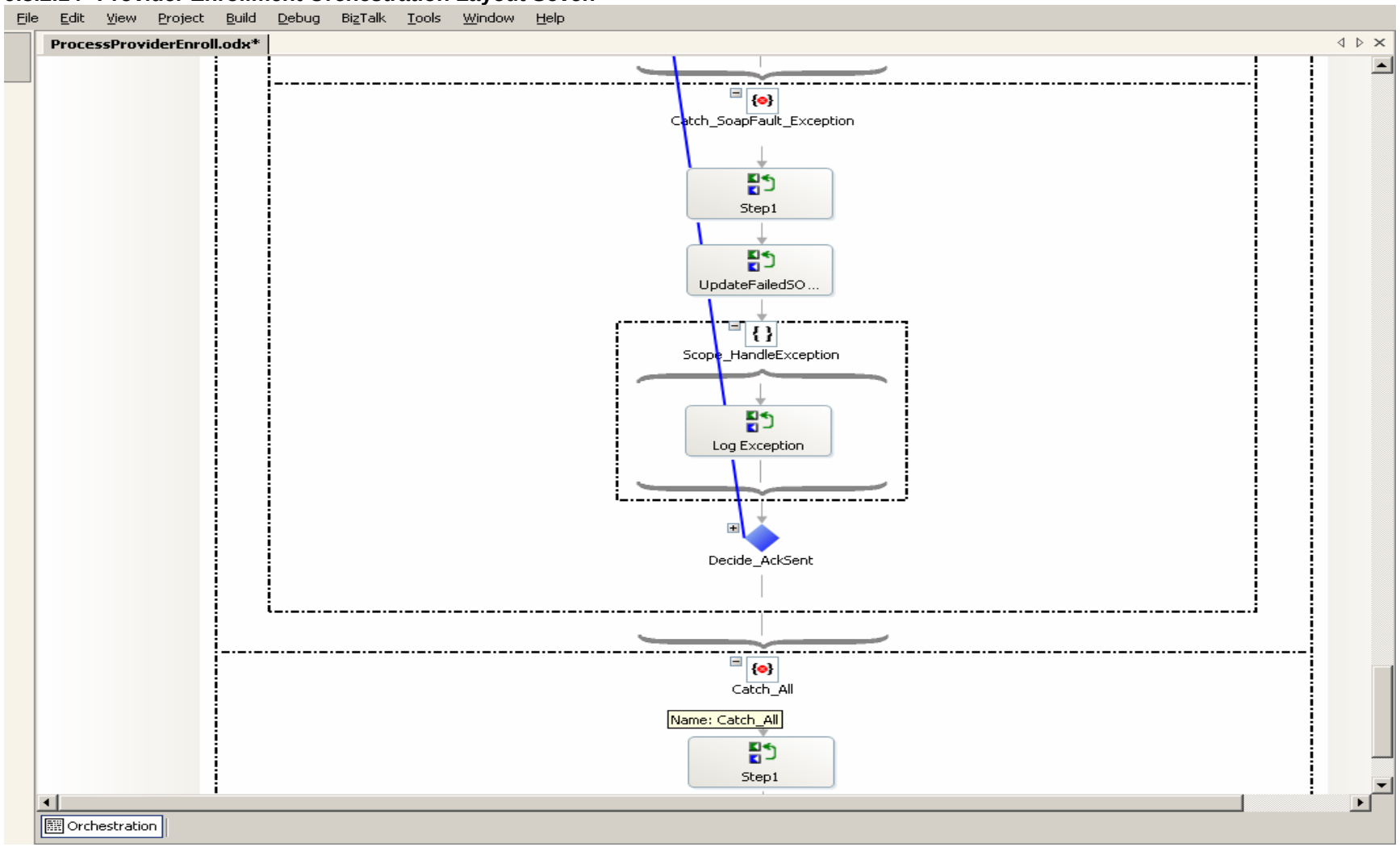
9.8.2.12 Provider Enrollment Orchestration Layout Six



9.8.2.13 Description

The exception handlers will call a custom tracking library which will update the Failure status to above created PE record which will be used for reprocessing failed Provider Enrollments. The exception handlers will catch and handle any processing errors and set the “failed” milestone on the root tracking activity.

9.8.2.14 Provider Enrollment Orchestration Layout Seven

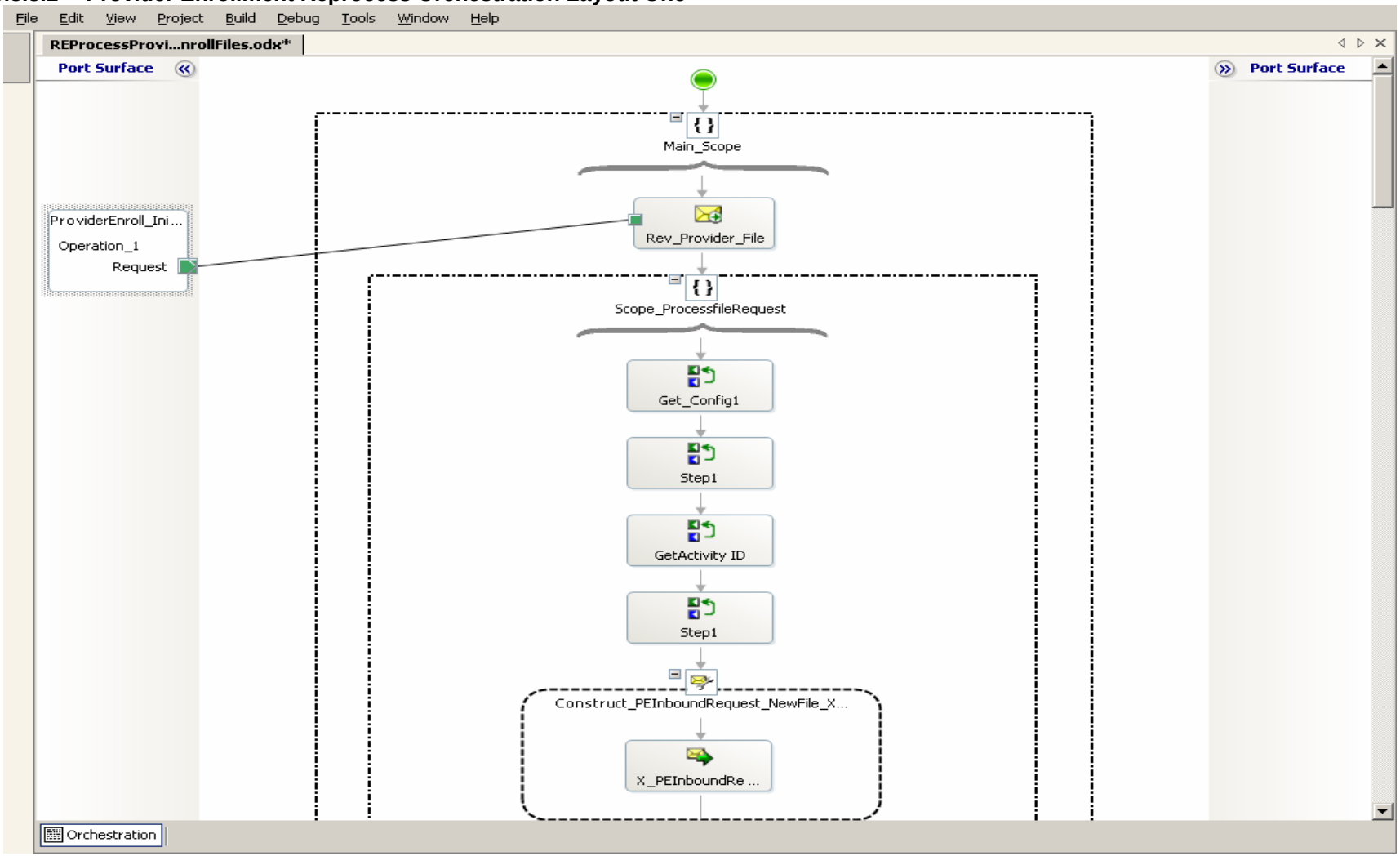


9.8.3 Provider Enrollment Reprocess Orchestration

9.8.3.1 Description

The PE failed request is received by a File Receive Port and configuration information is retrieved. If debug mode is enabled, the incoming message is written to the debug log (a flat-file).

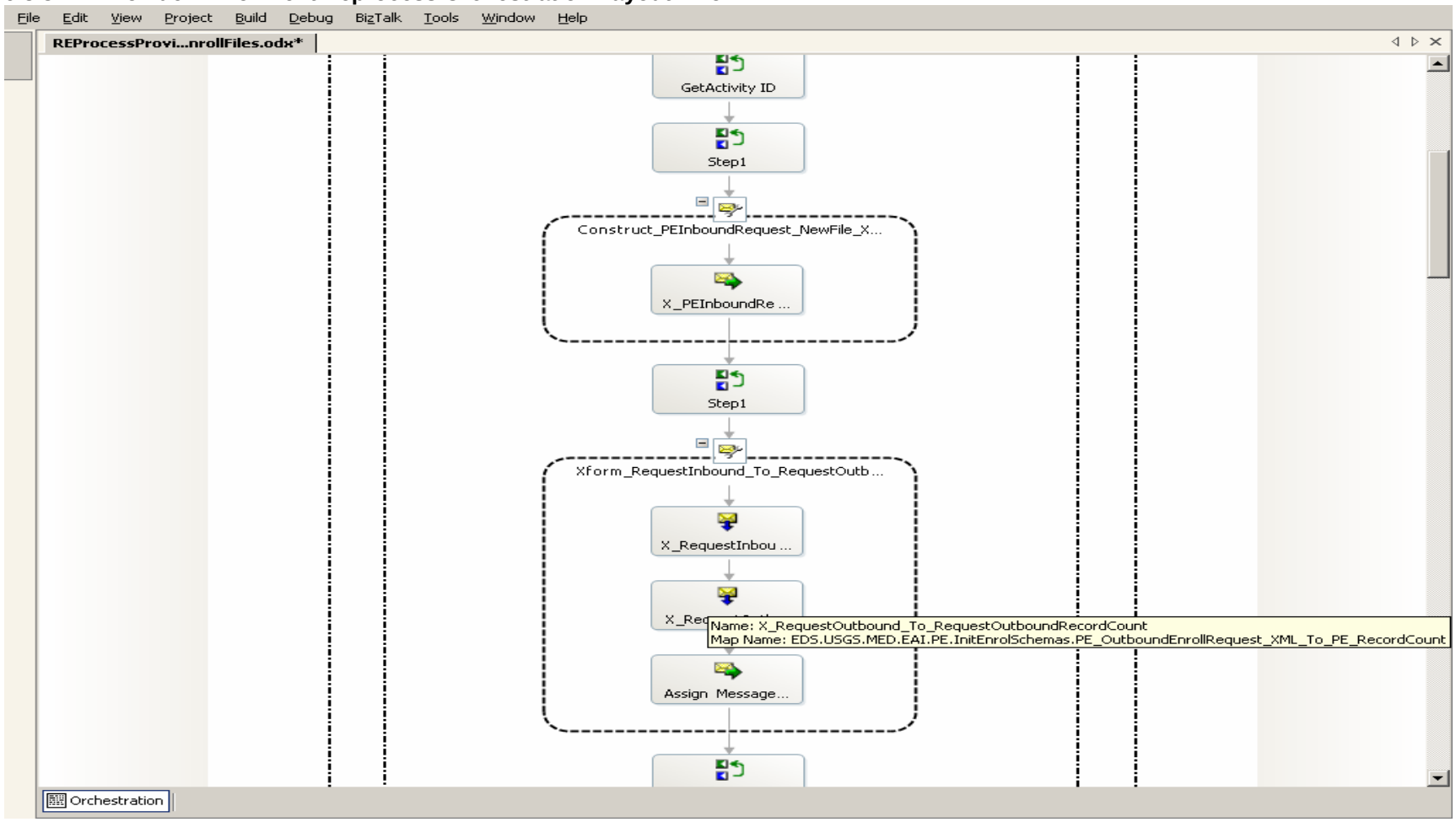
9.8.3.2 Provider Enrollment Reprocess Orchestration Layout One



9.8.3.3 Description

Based upon received PE failed Request file name orchestration calls the custom tracking library to retrieve tracking information from PE custom retry tracking table. Tracking Library returns BAM ActivityID and RelativeActivityID. Once BAM related information is retrieved by Custom tracking Library, Orchestration calls the Tracking library method to increment the Retry Count, which can be used to find the no of retries attempted before PE completion.

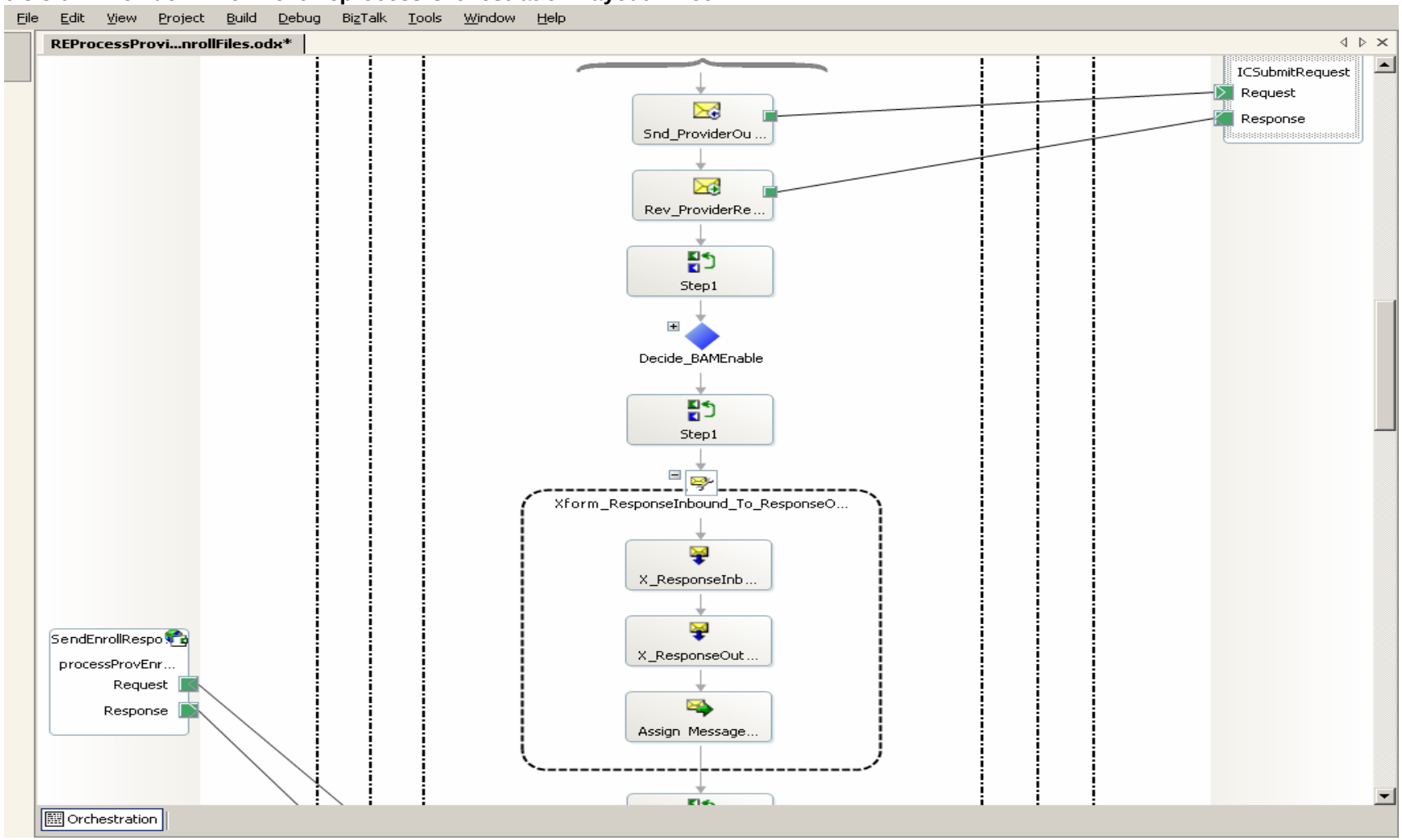
9.8.3.4 Provider Enrollment Reprocess Orchestration Layout Two



9.8.3.5 Description

The PE request is sent to interChange using a synchronous solicit-response pattern. The response from interChange is mapped back into FIQM format. The response message is tracked automatically by a pipeline component but must be manually added as a related activity to the root. This is because subsequent/intermediate receives are unaware of the root activity Id at the time the message is received.

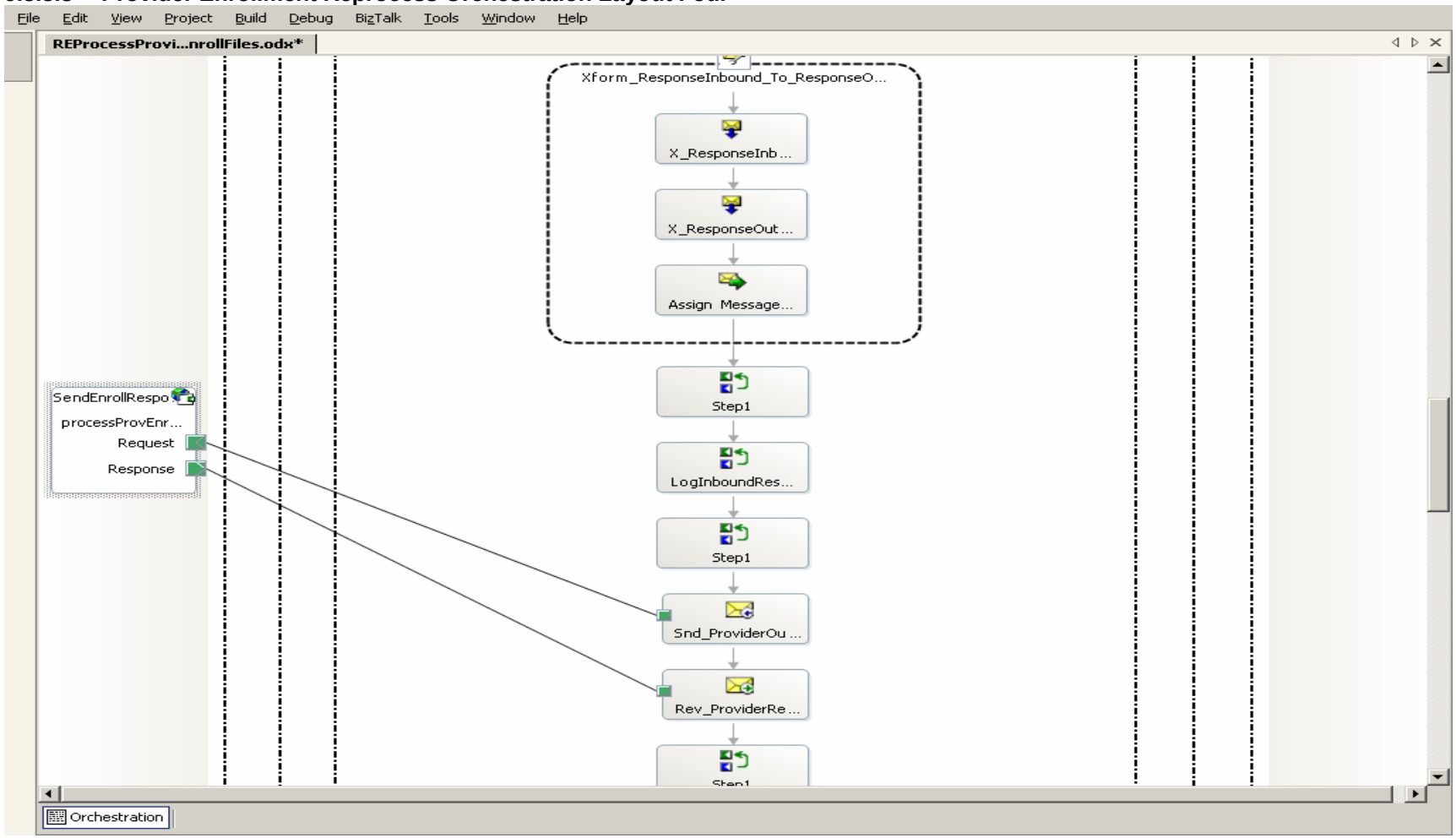
9.8.3.6 Provider Enrollment Reprocess Orchestration Layout Three



9.8.3.7 Description

The response message which is in FIQM format is send to FIQM by using synchronous solicit-response pattern – calling FIQM Web service call and receives the acknowledgement from FIQM Web Service. The response acknowledgement is tracked automatically by a pipeline component and archived optionally. Once the response acknowledgment is received PE Orchestration calls the Custom Tracking Library for updating the completion status to Tracking record created in the Custom tracking table. Multiple exception handlers are provided to catch either inbound or outbound message processing errors.

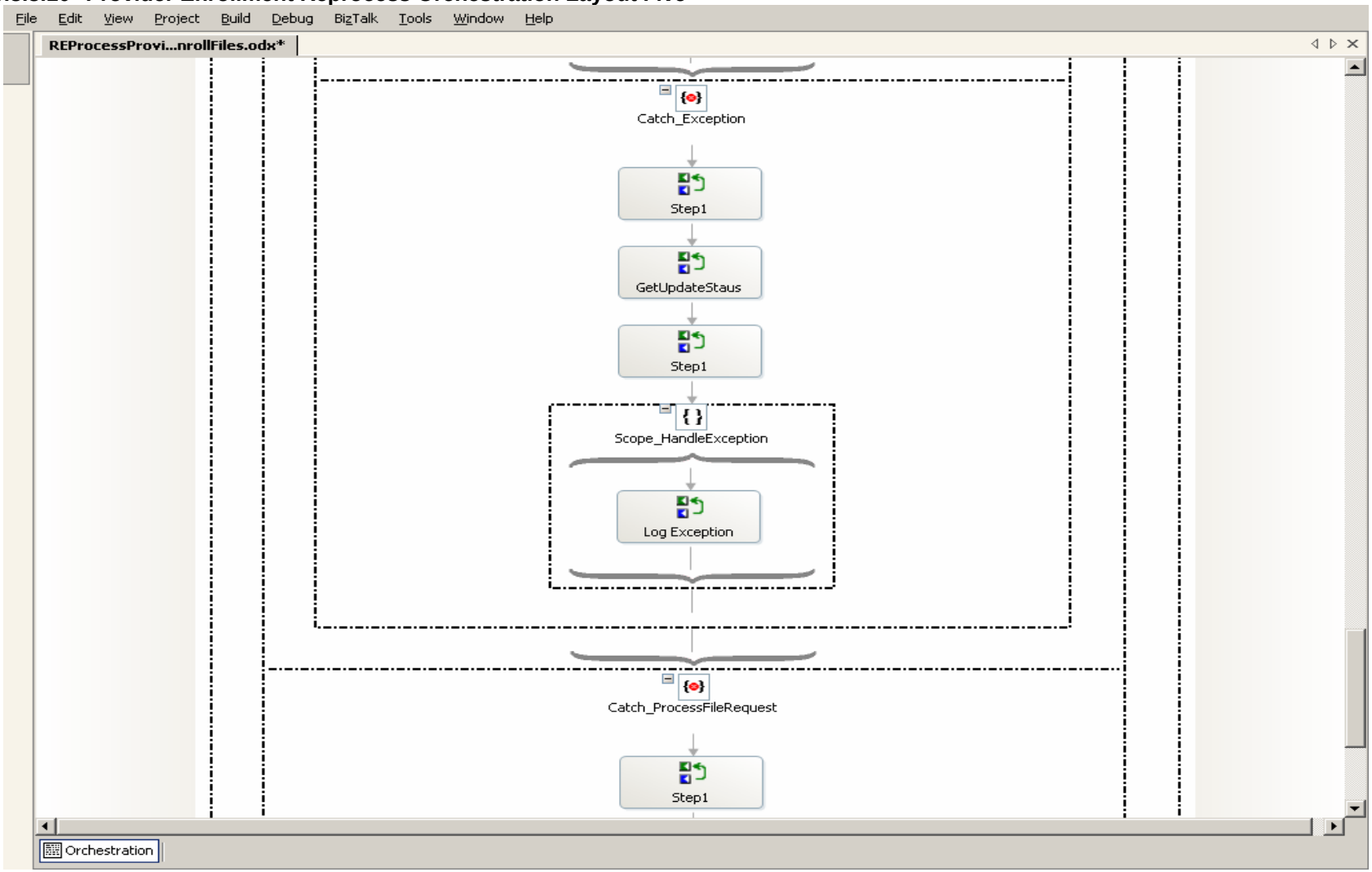
9.8.3.8 Provider Enrollment Reprocess Orchestration Layout Four



9.8.3.9 Description

Multiple exception handlers are provided to catch either inbound or outbound message processing errors.

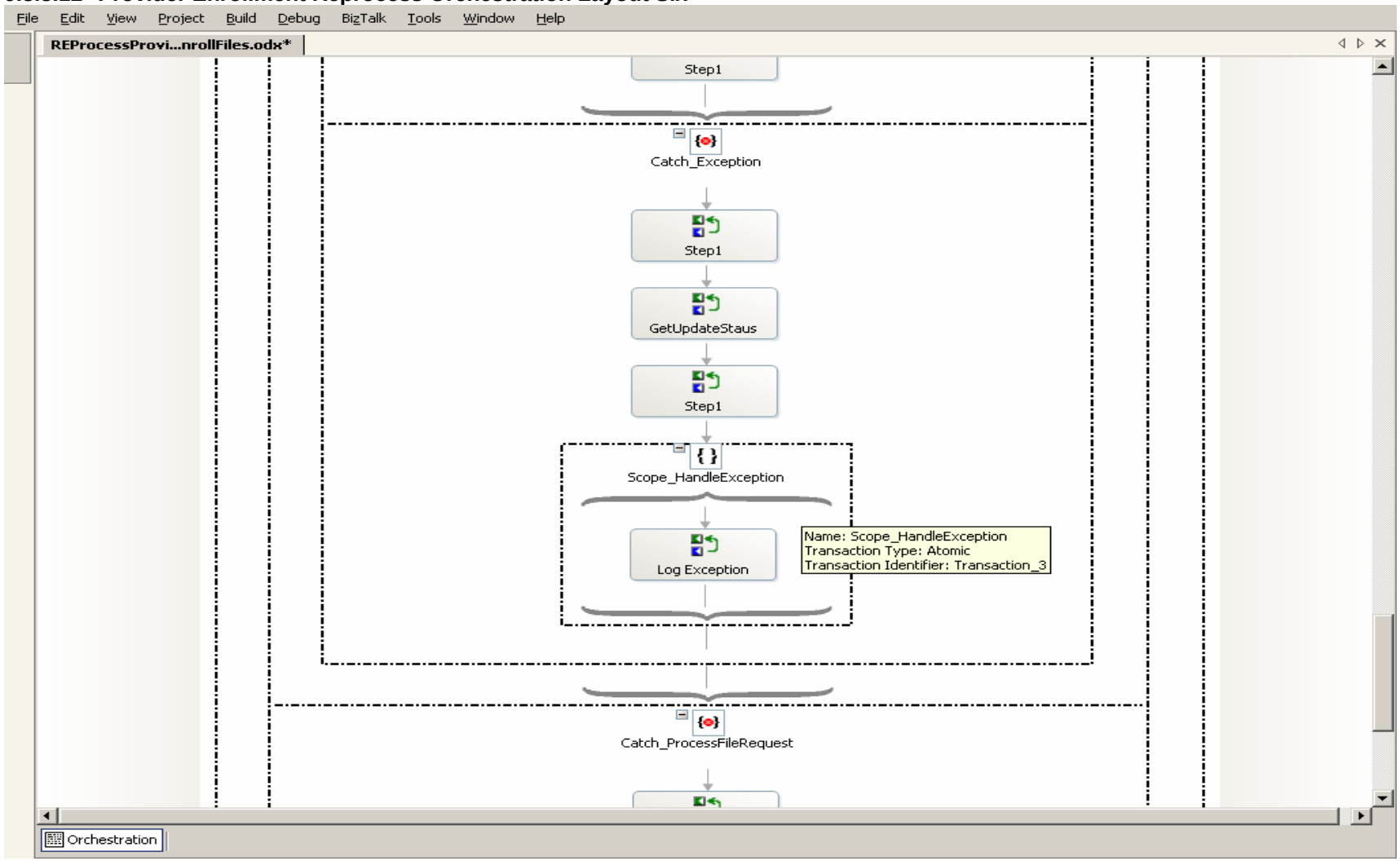
9.8.3.10 Provider Enrollment Reprocess Orchestration Layout Five



9.8.3.11 Description

The exception handlers will call a custom tracking library which will update the Failure status to above retrieved PE record which will be used for PE retry process. The exception handlers will catch and handle any processing errors and set the “failed” milestone on the root tracking activity.

9.8.3.12 Provider Enrollment Reprocess Orchestration Layout Six

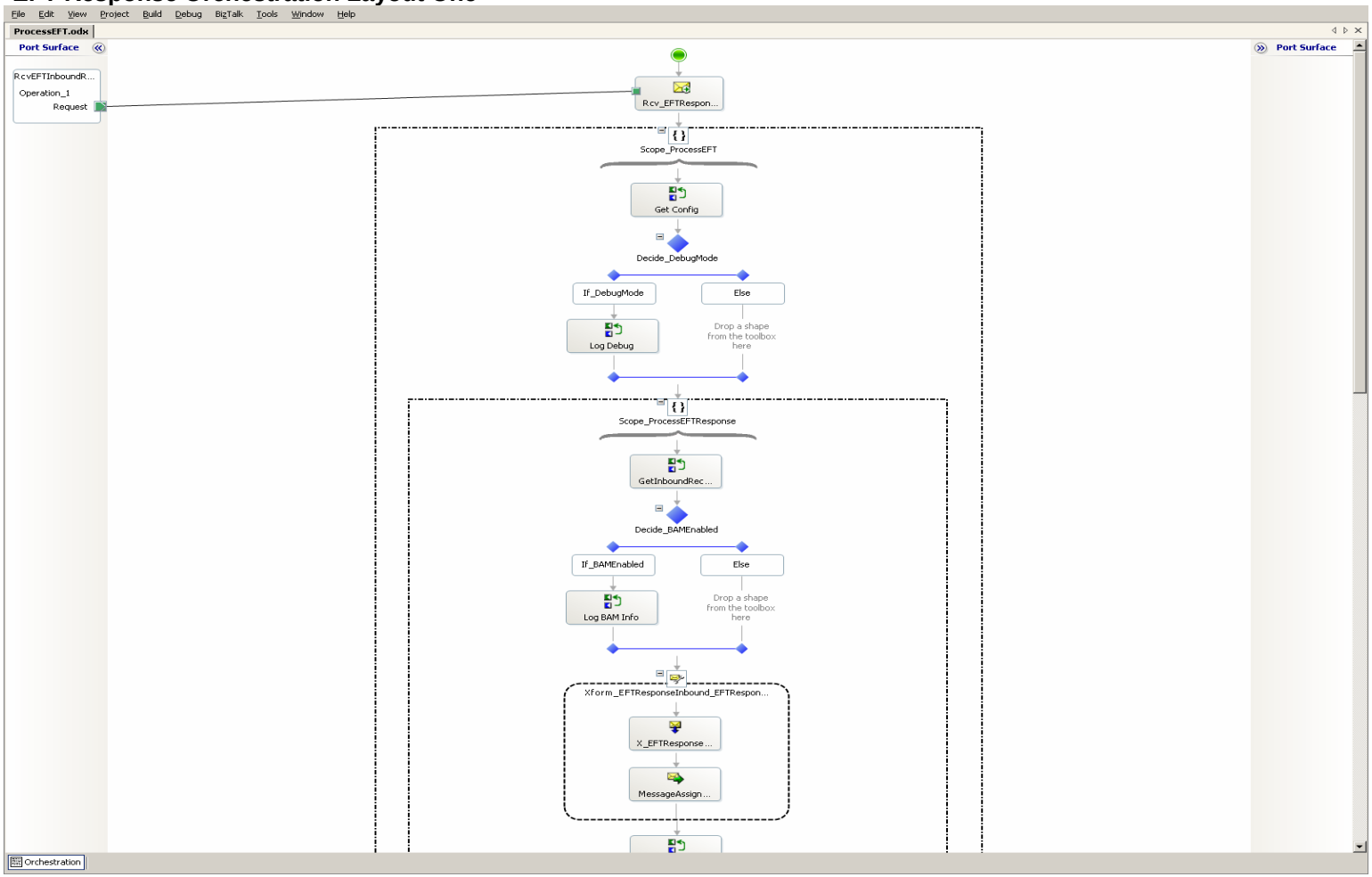


9.8.4 EFT Response Orchestration

9.8.4.1 Description

An EFT response is received via the SOCKETS adapter and configuration information is retrieved from Enterprise Library. This initial received message is a multi-part message that includes a BAMPart with tracking IDs for the root and current file/message activity. If debug mode is enabled, the incoming message is written to the debug log (a flat-file). The response is mapped to the FIQM format ready to be sent via web services.

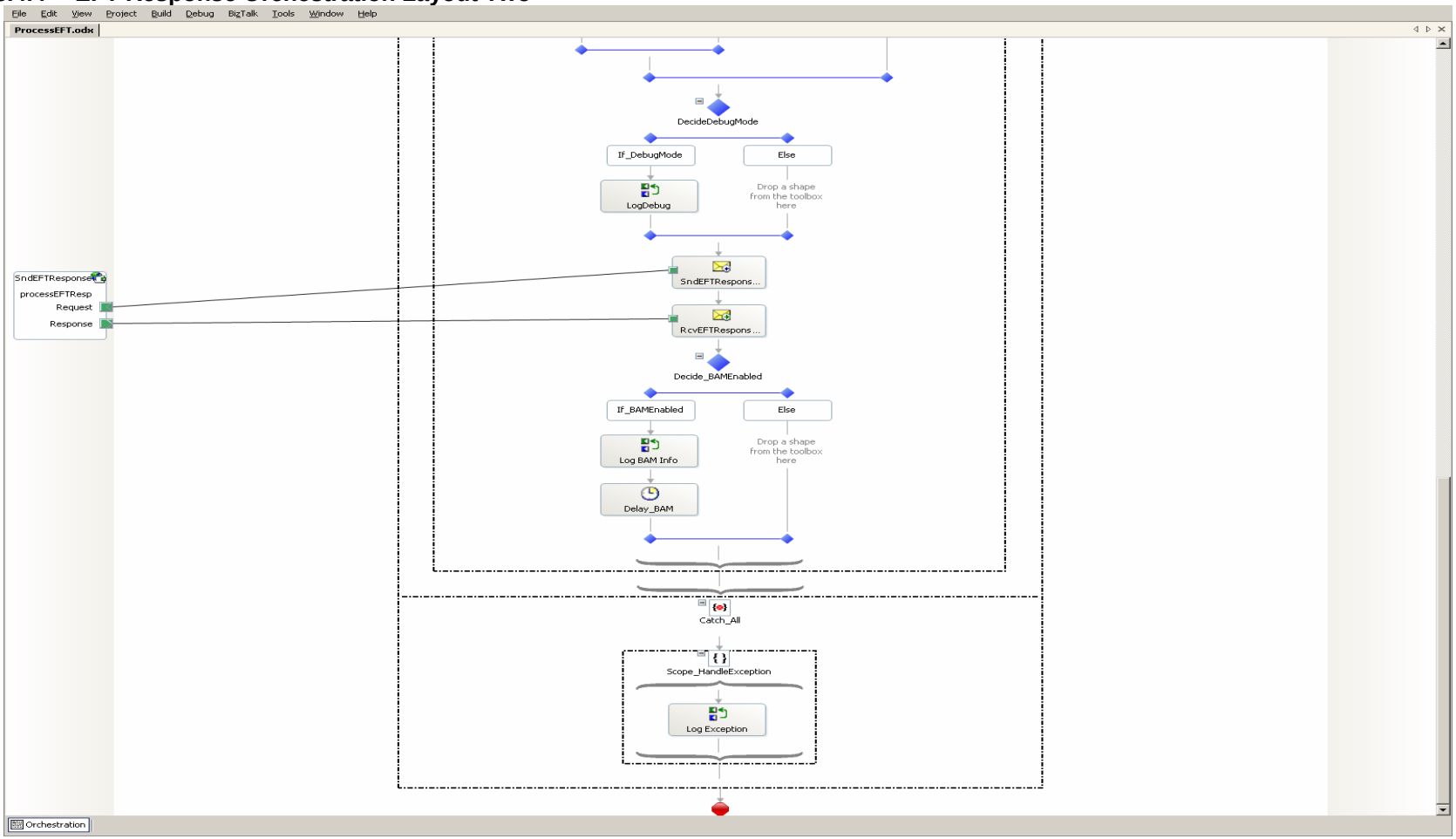
9.8.4.2 EFT Response Orchestration Layout One



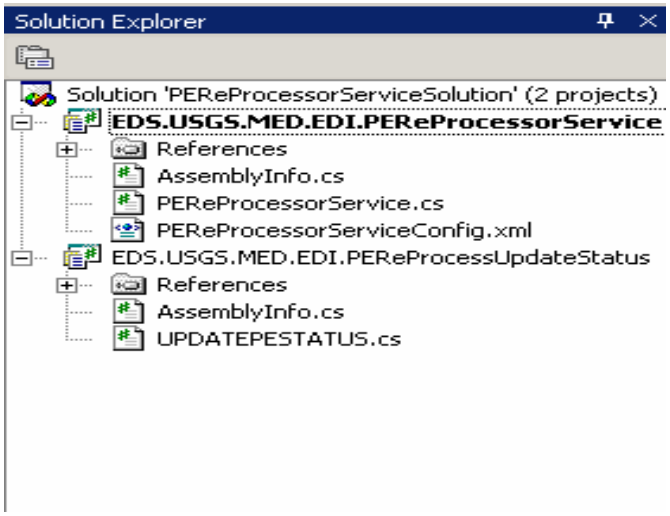
9.8.4.3 Description

The FIQM EFT Response web service is invoked and a simple ACK should be received. This message is tracked as a related file/message tracking activity. An exception handler will catch and handle any processing errors and set the “failed” milestone on the root tracking activity.

9.8.4.4 EFT Response Orchestration Layout Two



9.8.5 PE ReProcessor Service Solution Dot Net projects



- EDS.USGS.MED.EDI.PEReProcessorService – Custom Dot Net Console Application containing methods to retrieve the failed Provider Enrollments for Re processing
- EDS.USGS.MED.EDI.PEReProcessUpdateStatus – Custom Dot Net utility class Library containing methods to retrieve archive file information and update PE Orchestration Failure/Completion Tracking Status

9.8.5.1 PE ReProcessor Service Console Application

The PE ReProcessor Console Application has main method for retrieving the failed Provider Enrollments information and submitting the PE xml files for Re Processing using Provider Enrollment Reprocess Orchestration. The Console Application is Scheduled to run for every 4minutes which runs query against the Custom PE Processing Tracking Table and Fetches the Failed PE files, submits files to a folder where BizTalk Orchestration is configured to pick up for reprocessing. Once the file is fetched and submitted for reprocessing Console application updates the 'PROCESSING' Status to the Record in Tracking Table.

9.8.5.2 Custom PE Tracking Table

Table Name: tblPR_Status

Database Name: BAMPrimaryImport

Column Name	Type	Length	Nullable	Column Description
ActivityID	varchar	256	No	Field used to store BAM activity ID information of PE Request
Status	varchar	256	No	Field used to store PE Processing Status like PROCESSING, FAILED, COMPLETED
Retry Count	int	4	Yes	Field used to store the PE Processing Retry count

Column Name	Type	Length	Nullable	Column Description
CreateDate	datetime	8	No	Field used to store PE Processing create Date time
UpdateDate	datetime	8	Yes	Field used to store PE Retry Processing Date time
RelativeActivityID	varchar	256	No	Field used to store BAM Relative Activity ID information of PE Request

9.8.5.3 EDS.USGS.MED.EDI.PEReProcessUpdateStatus Class Library

This is a Custom Dot Net Utility Class library which is called from the both PE Orchestration and PE Reprocess Orchestration for creating the record in Custom PE Tracking Table and updating the Orchestrations Failed/Completed Status.

File Name: UPDATEPESTATUS.cs	Class Name: UPDATEPESTATUS
Method Name	**Description**
UPDATEPESTATUS	This is the default constructor of the class and is used to initialize the class variable and reading the xml configuration to read Database connection string.
GetActivityID	This method is called by the PE Reprocess orchestration to get the BAM Activity ID information of Processing PE Request xml file. GetActivityID method takes the Archived file Name as input parameter and runs a query against the bam_EAITracking_CompletedRelationships and bam_EAIFileTracking_Completed tables in BAMPrimaryImport Database and returns Activity ID.
GetRelativeActivityID	This method is called by the PE Reprocess orchestration to get the BAM Relative Activity ID information of Processing PE Request xml file. GetRelativeActivityID method takes the ActivityID as input parameter and runs a query against the tblPR_Status PE Custom Tracking table in BAMPrimaryImport Database and returns Relative Activity ID.
CreateNewPEstatusRecord	This method is called by PE orchestration to create new record in tblPR_Status custom tracking table in BAMPrimaryImport Database. CreateNewPEstatusRecord method accepts ActivityID, RelativeActivityID as input parameters.
UpdateRetryCount	This method is called by PE Reprocess orchestration to update Retry count to above created tblPR_status custom tracking table record. UpdateRetryCount method accepts ActivityID has input parameter.

File Name: UPDATEPESTATUS.cs	Class Name: UPDATEPESTATUS
Method Name	**Description**
UpdatePEStatus	This method is called by both PE orchestrations to update the orchestrations processing completed or failed status. UpdatePEStatus accepts ActivityID and Status as input string parameters.

9.8.6 BAM Instrumentation

9.8.6.1 BAM Definition

The BAM activity for the EAI framework must be deployed as prerequisite to deploying the Provider Enrollment tracking profile. The BAM definition consists of the following components.

The BAM_EAITracking.xls spreadsheet

Defines the EAITracking and EAIFileTracking activites which are made up of individual business data elements and milestones. The EAITracking view defines the OLAP dimensions and processing durations so that average processing times, and transaction counts can be reported by day, month, quarter and year.

The BAM_EAITrackingLiveData.xls spreadsheet

The spreadsheet that is used by an end user to view aggregated tracking data from the OLAP cube.

The BAM_EAITracking.xml exported BAM definition that is used to instrument orchestrations using the EAI framework.

The BAM definition that was exported from the BAM_EAITracking.xls spreadsheet that when deployed creates all the SQL Server artifacts such as tables, views, and DTS packages to automate data maintenance and updating the OLAP cube with the latest information.

Batch messages have been created so that the BAM definition can be quickly deployed or removed as required. Once the BAM definition has been deployed an orchestration can be instrumented.

9.8.6.2 Provider Enrollment Tracking Profile

In order to track processing milestones the orchestration must be instrumented with the BAM Tracking Profile Editor (TPE). The tracking profile consists of the following items.

EDS.USGS.MED.EAI.PE.InitEnrolEmpty.btt – An empty tracking profile that when deployed removes the associated tracking profile from the provider enrollment orchestration.

EDS.USGS.MED.EAI.PE.InitEnrol.btt – The tracking profile that when deployed instruments the provider enrollment orchestration for BAM.

EDS.USGS.MED.EAI.PE.EFTProcessEmpty.btt – An empty tracking profile that when deployed removes the associated tracking profile from the EFT process orchestration.

EDS.USGS.MED.EAI.PE.EFTProcess.btt – The tracking profile that when deployed instruments the EFT process orchestration for BAM.

Batch messages have been created so that the Provider Enrollment tracking profile can be quickly deployed or removed as required.

The following table shows the mapping between orchestration shapes and the BAM tracking definition. Note that some of the values are set by the Receive Archiver or Send Archiver component and therefore are not mapped by the orchestration.

EAITracking Activity Data	Orchestration Shape or Pipeline component
AppName	Set by the receive pipeline on the initial receive
Failed	Log Exception shapes in the error handlers or by pipeline components
HandlingInstanceId	Set by calling the BAM API in the error handler
ProcessBegin	The initial receive message shape
ProcessEnd	The last receive message shape that receives an ACK from FIQM.
ProcessType	Set by the receive pipeline component on the initial receive
RequestReceived	Set by the receive pipeline component on the initial receive
ResponseSent	Set by the send pipeline component on the final send
ReceiverPartyId	Set by the receive pipeline component on the initial receive
SenderPartyId	Set by the send pipeline component on the final send
ActivityID	Maps to the inbound BAMPart message part ActivityID attribute.

9.8.7 Publishing the BizTalk Web Service

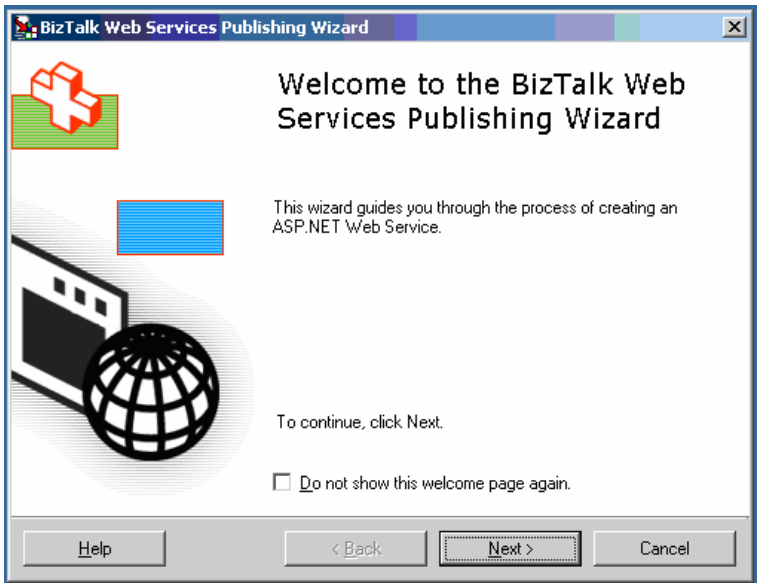
The Provider Enrollment orchestration uses a BizTalk published web service to receive enrollment requests. The web service should be created by using the “Publish schemas as web services” option rather than using the “Publish BizTalk orchestrations as web services” option. The reason for this that the initial received message is defined as being a multi-part message with a message body of the request and an additional message part known as the BAMPart. If the orchestration was published as a web service the BAMPart would show up in the signature of the operation that is described in the WSDL. As the BAMPart is a hidden message part, the explicit schemas for the request and response should be used.

The following steps describe how to publish the PE web service using the standard web services wizard. When the web service is upgraded to include support for the WSE Adapter, the equivalent WSE wizard should be also used to publish the web service also using the schemas.

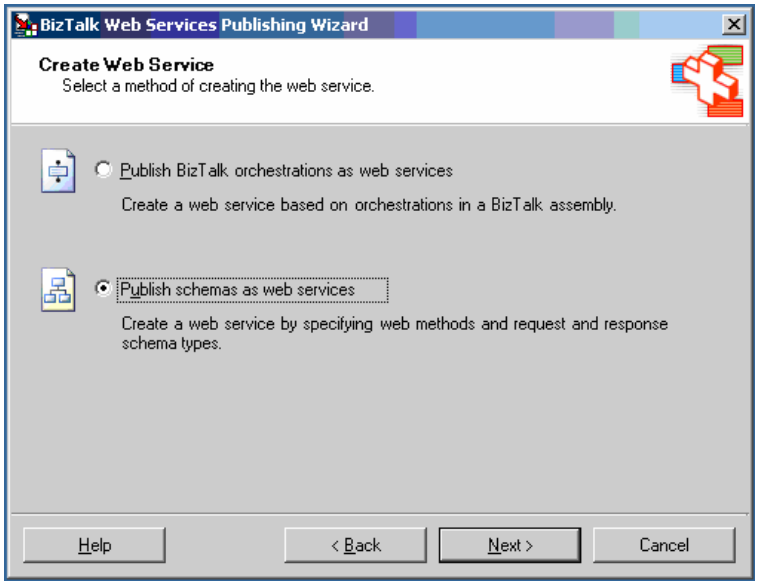
Follow these steps to publish the web service:

Ensure that the ProvEnroll has been configured as an excluded path using the SharePoint administrative site. This will ensure that Windows SharePoint Services does not try to manage the virtual directory.

Start the BizTalk Web Services Publishing Wizard and click next to continue.



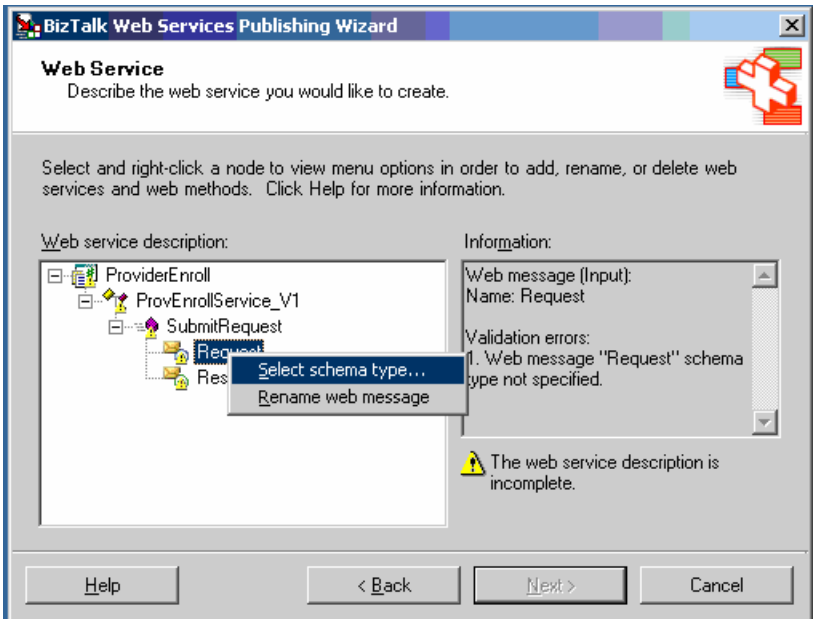
Select the “Publish schemas as web service” option and click next to continue.

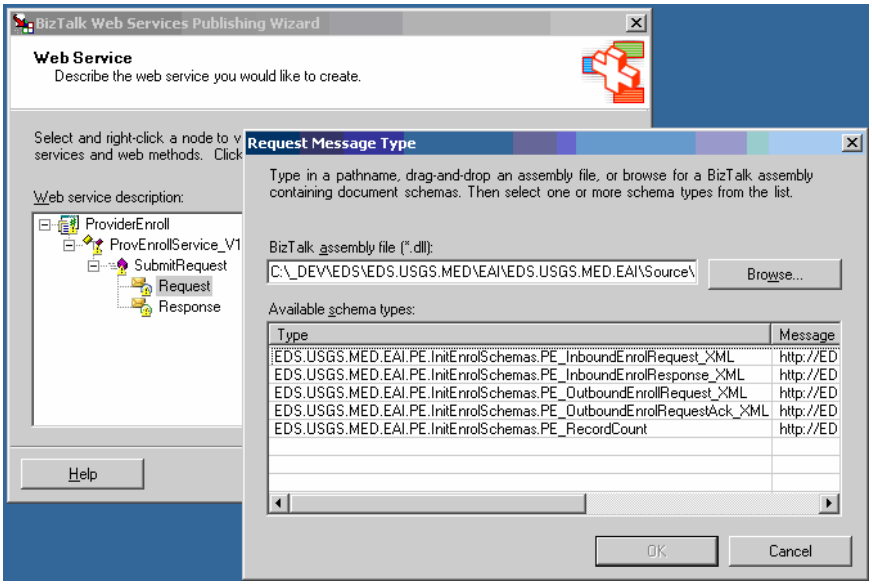


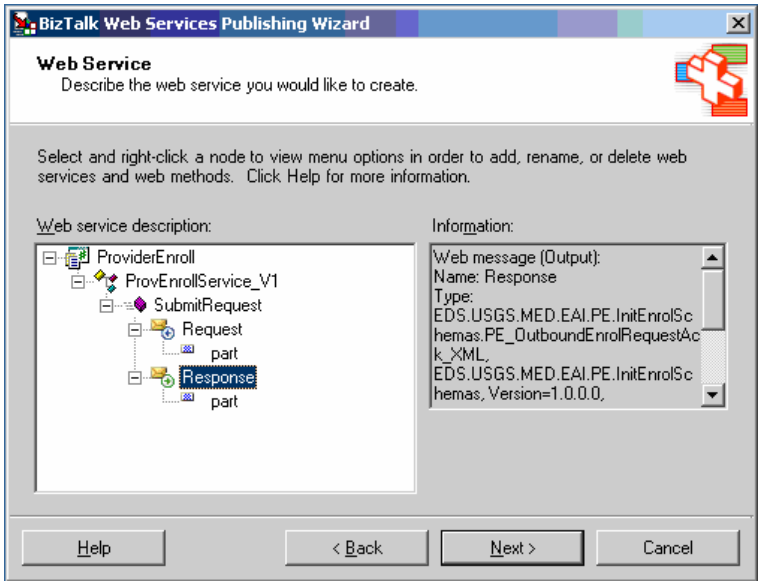
Rename the web service description, web service operation and web method as shown in the screen shot below. Note that the operation has a versioned name “_v1” to allow for side-by-side deployments of an upgraded web service.

Select the PE request schema for the request

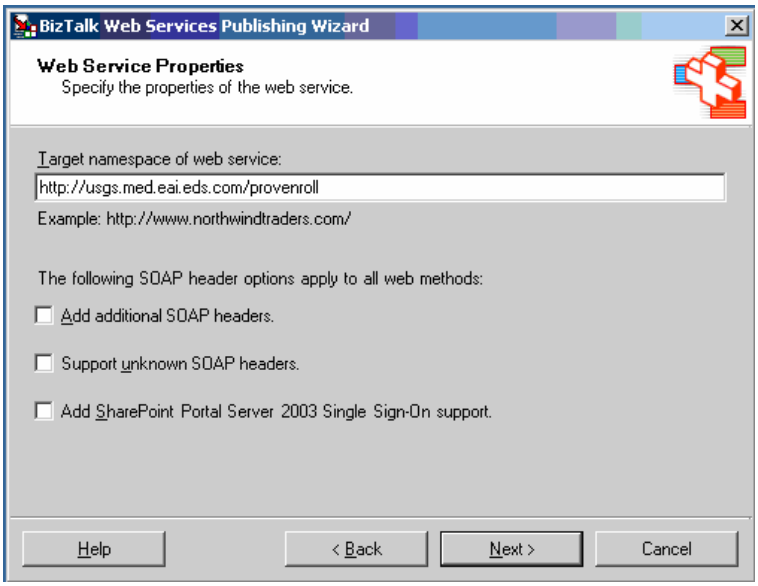
Select the PE request acknowledgement schema for the acknowledgement


BizTalk Web Services Publishing Wizard
Web Service
Describe the web service you would like to create.
Select and right-click a node to view menu options in order to add, rename, or delete web services and web methods. Click Help for more information.
Web service description:
Information:
ProviderEnroll
ProvEnrollService_V1
SubmitRequest
Request
Res
Select schema type...
Rename web message
Web message (Input):
Name: Request
Validation errors:
1. Web message "Request" schema type not specified.
The web service description is incomplete.
Help
< Back
Next >
Cancel

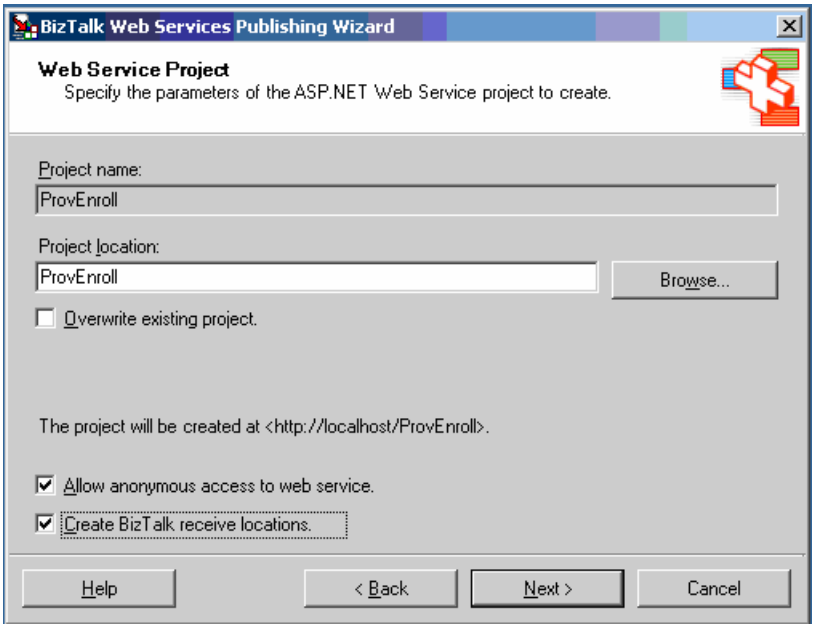

BizTalk Web Services Publishing Wizard
Web Service
Describe the web service you would like to create.
Web service description:
ProviderEnroll
ProvEnrollService_V1
SubmitRequest
Request
Response
Help
Request Message Type
Type in a pathname, drag-and-drop an assembly file, or browse for a BizTalk assembly containing document schemas. Then select one or more schema types from the list.
BizTalk assembly file (*.dll):
C:_DEV\EDS\EDS.USGS.MED\EAI\EDS.USGS.MED.EAI\Source\
Browse...
Available schema types:
Type
Message
EDS.USGS.MED.EAI.PE.InitEnrolSchemas.PE_InboundEnrolRequest_XML
http://ED
EDS.USGS.MED.EAI.PE.InitEnrolSchemas.PE_InboundEnrolResponse_XML
http://ED
EDS.USGS.MED.EAI.PE.InitEnrolSchemas.PE_OutboundEnrollRequest_XML
http://ED
EDS.USGS.MED.EAI.PE.InitEnrolSchemas.PE_OutboundEnrolRequestAck_XML
http://ED
EDS.USGS.MED.EAI.PE.InitEnrolSchemas.PE_RecordCount
http://ED
OK
Cancel



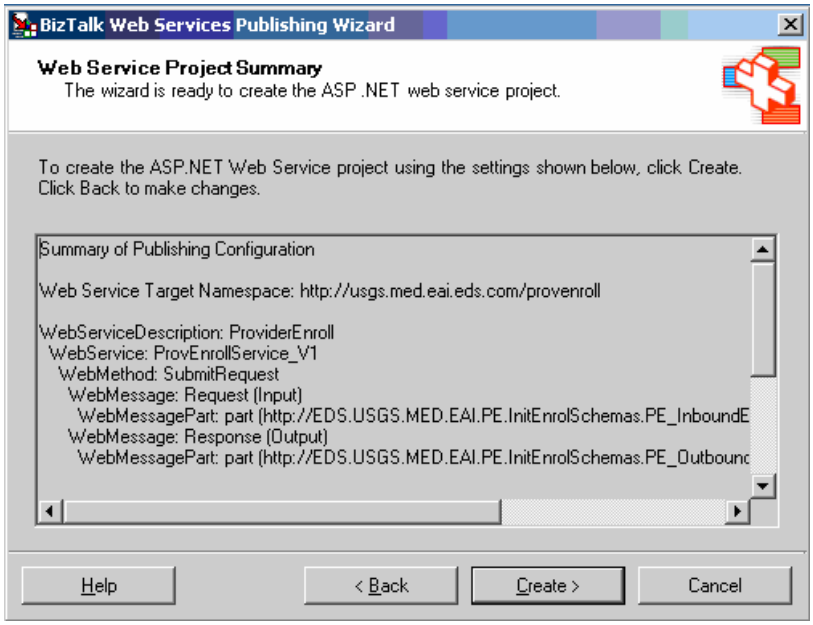
Set the target namespace to http://usgs.meds.eai.eds.com/provenroll.



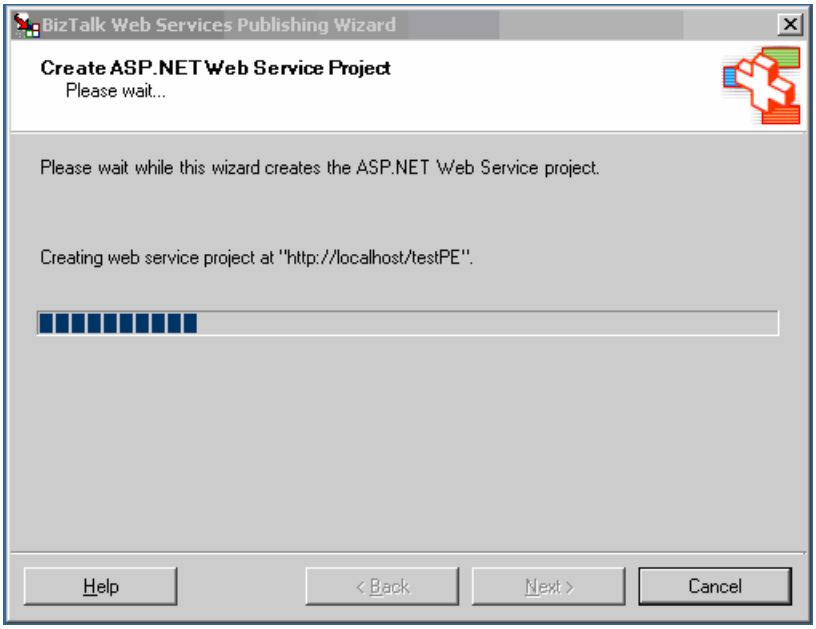
Set the project location (virtual directory) to ProvEnroll. Select “Allow anonymous” and “Create BizTalk receive locations” if this is the first time the web service has been generated.



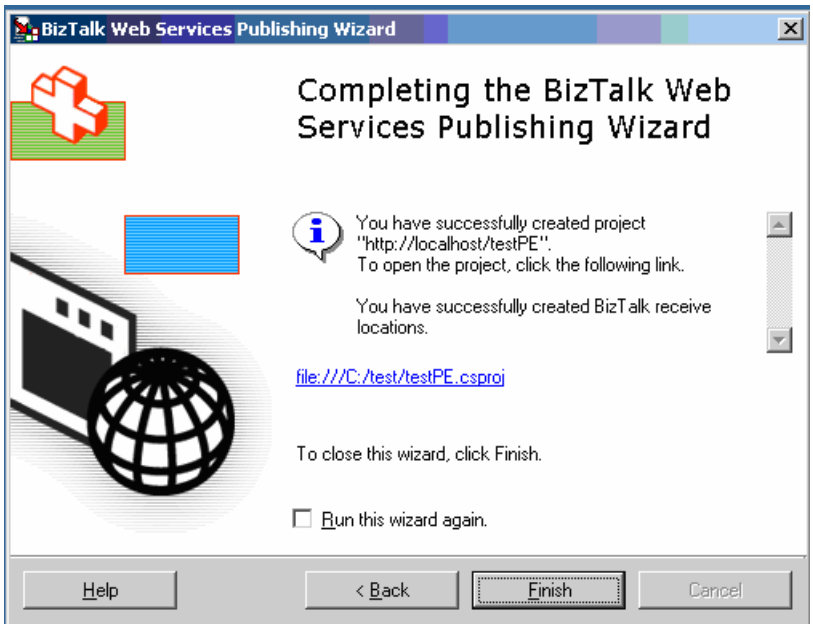
View the summary page and click “Next” when ready.



A dialog similar to the one shown below will be displayed.



If the web service was published correctly, a dialog similar to the one shown below will be displayed.



Open the IIS manager and verify that the web service is enabled for anonymous access (in development) and that it is running under any of the BizTalk application pools such as the TPM

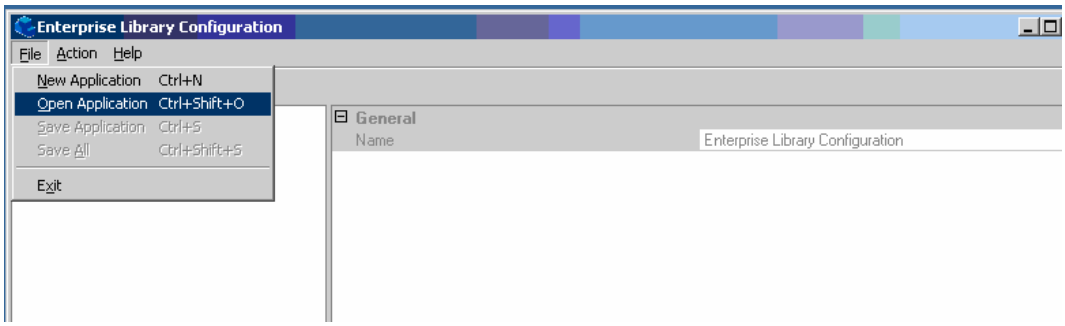
Perform an IISRESET in order to ensure that web service is ready to be invoked.

9.9 Configuration Management

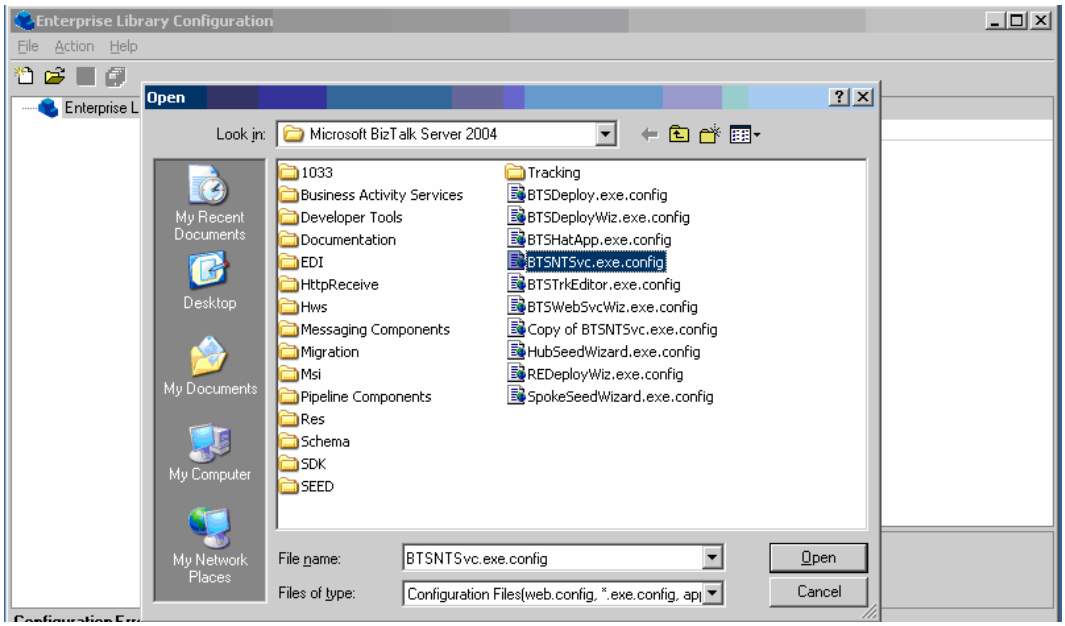
9.9.1 Overview

As the EAI framework is installed as a prerequisite to Provider Enrollment, many of the settings required for send, receive and BAM configuration; error handling and general purpose logging

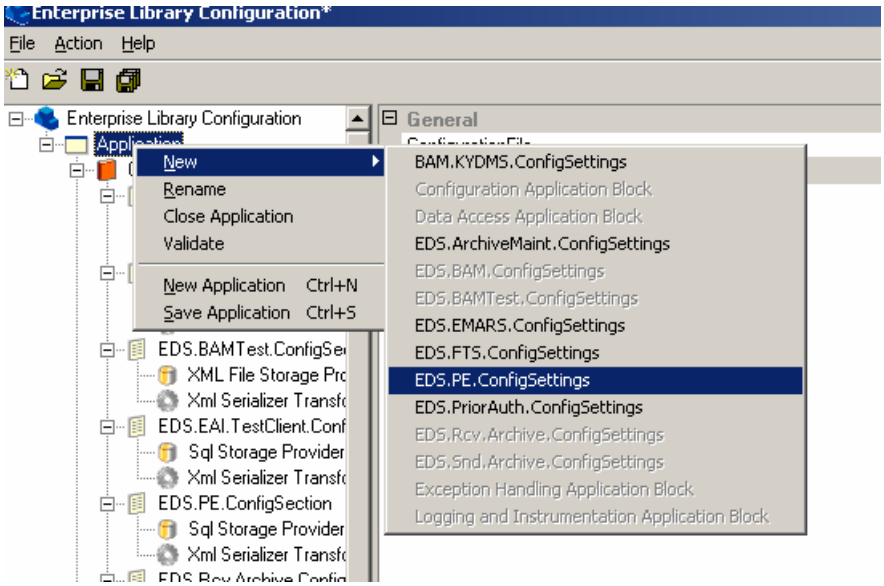
are already established. This section of the document describes only the Provider Enrollment specific settings that are needed after the application is installed into the target environment.



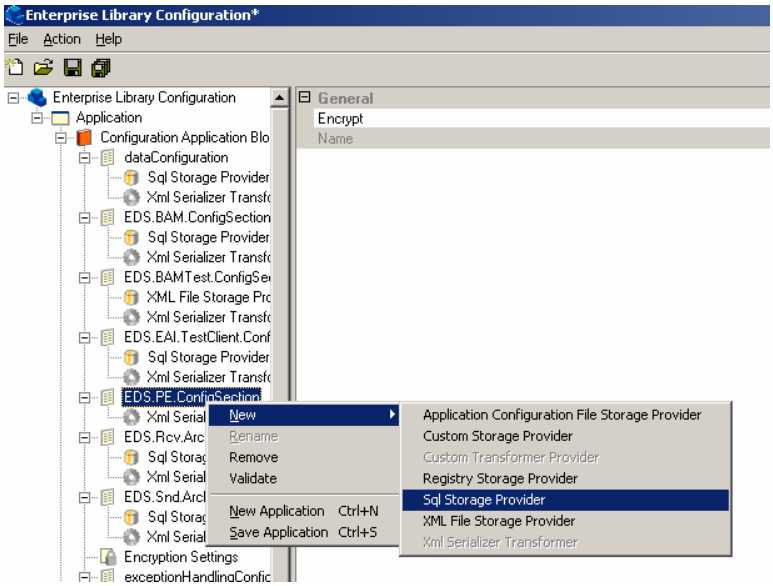
Using the Enterprise Library admin console, the PROVIDER ENROLLMENT configuration section can be added to the BizTalk BTSNTSvc.exe.config message. From the message menu select “Open “Application.”



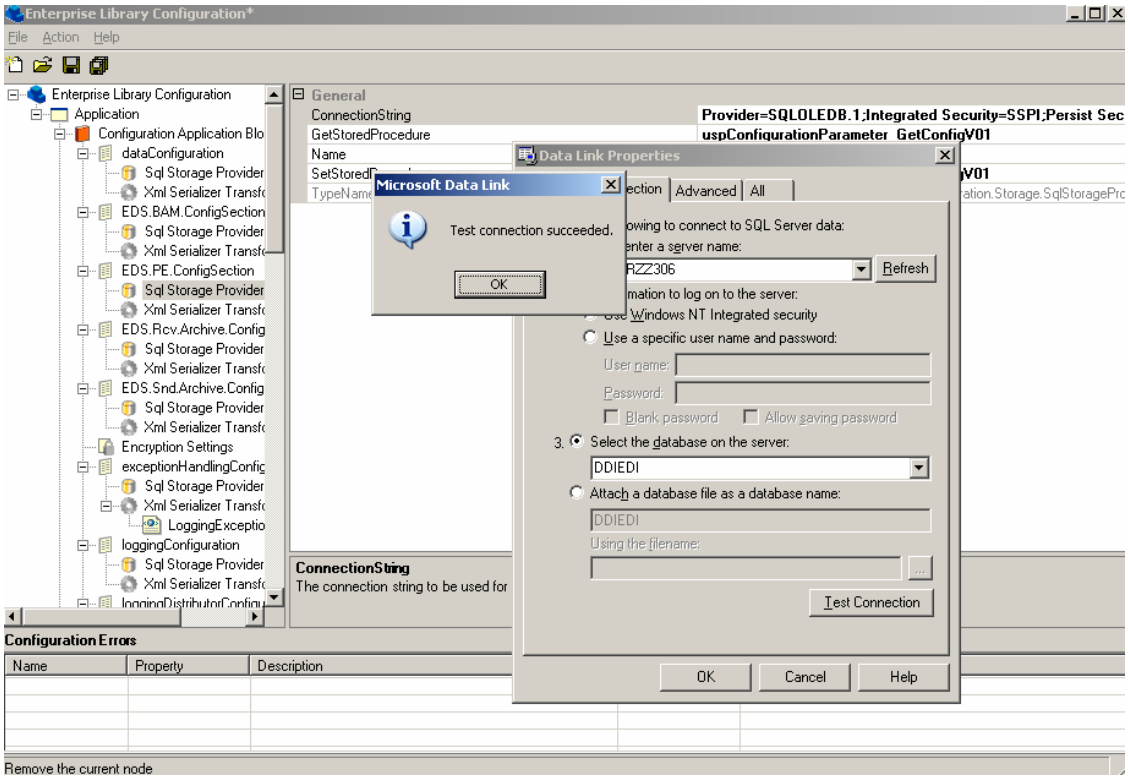
Locate the BTSNTSvc.exe.config message so that the Provider Enrollment settings can be added.



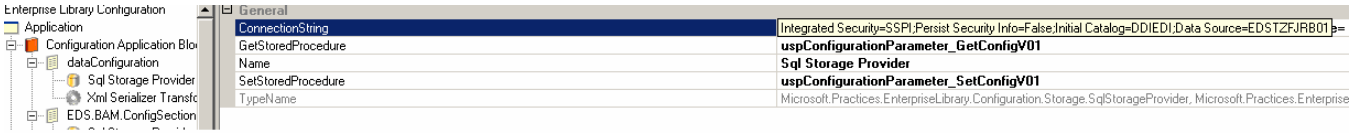
By right-clicking on the Application node it is possible to add an EDS.PE.ConfigSettings configuration section to the application settings.



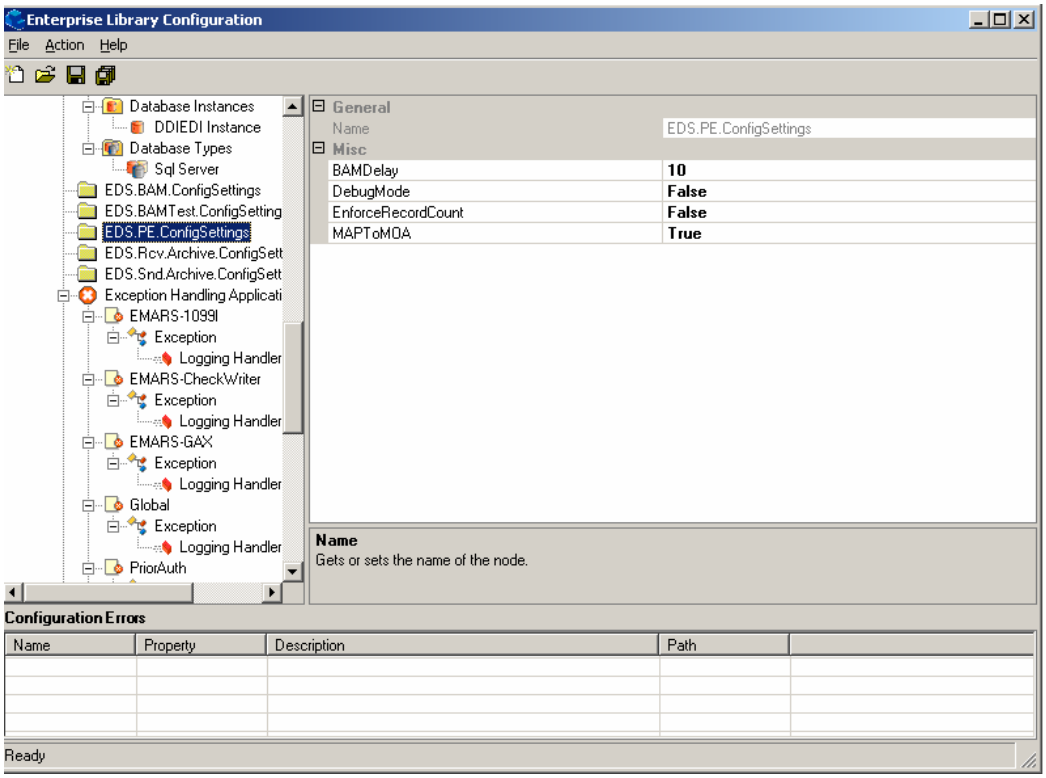
The default storage for configuration settings is for an XML message but this must be changed to SQL Server so that the settings can be shared across multiple BizTalk servers but administered from a single console session. Remove the XML Storage Provider and select the SQL Storage Provider as a replacement.



1. Select the ConnectionString property and enter the server name and DDIEDI database name. Click on Test Connection .A small dialog with Test connection succeeded dialog box is displayed



2. The other parameters should not need to be changed as they relate to the names of the stored procedures that will be used to save and retrieve the Provider Enrollment settings.



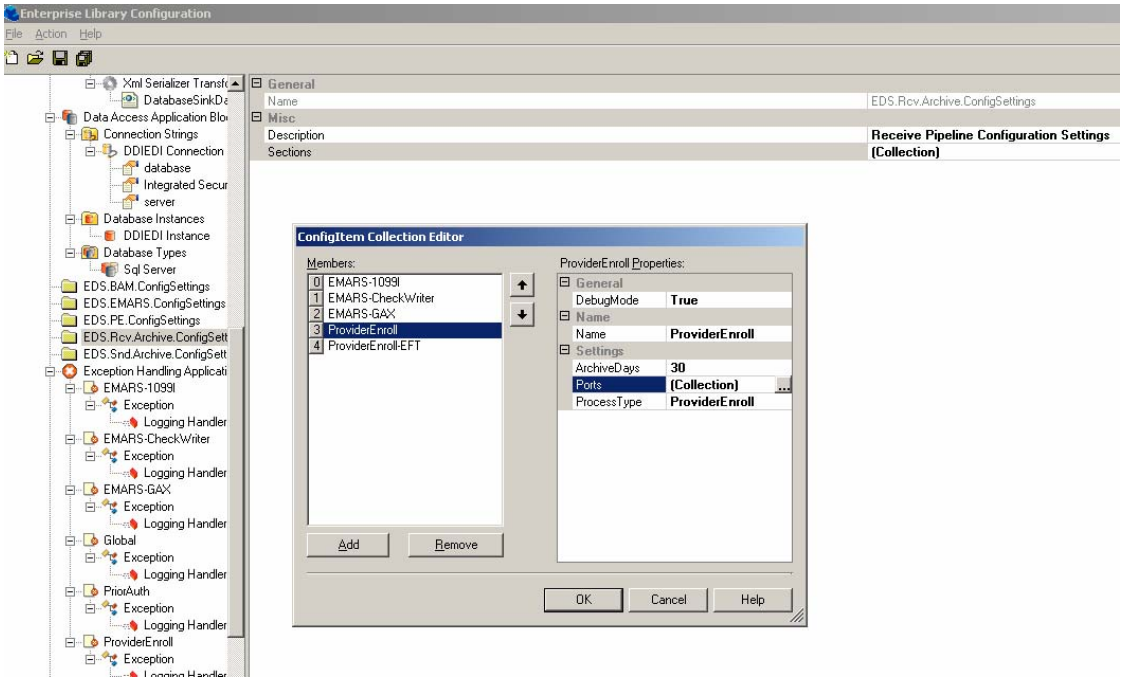
3. Once the storage provider has been corrected, the actual settings that are used by Provider Enrollment and EFT can be added into the configuration. The table shown below provides typical settings that will be needed.

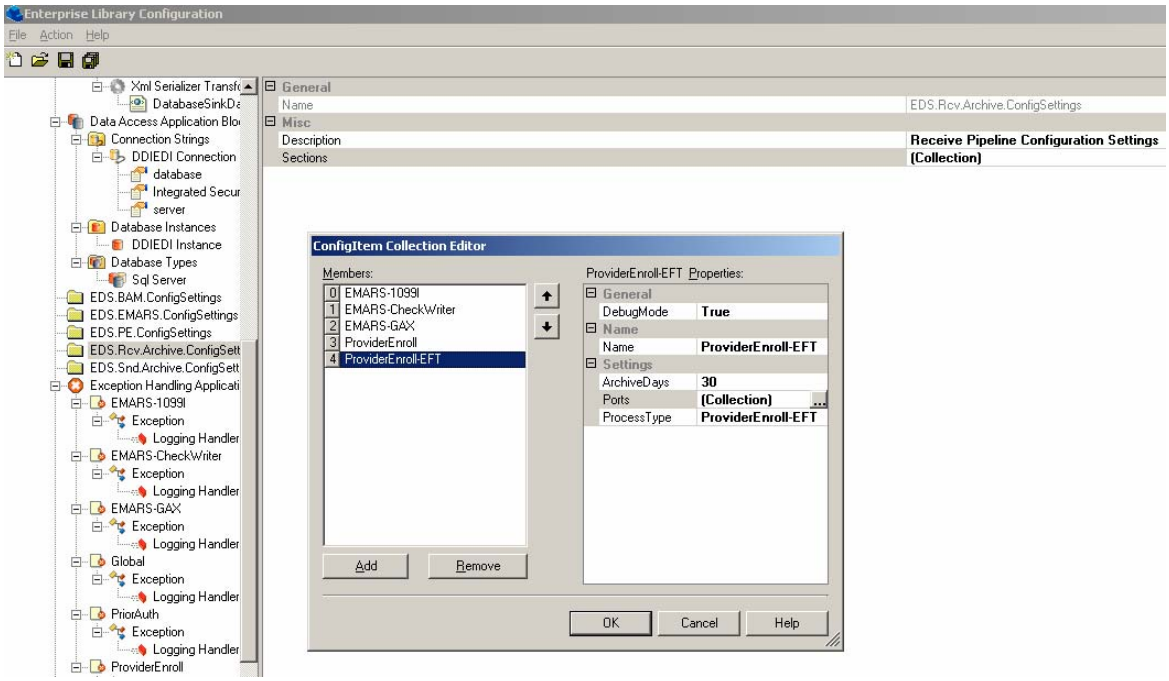
FTP	Typical Value	Description
BAMDelay	10 (seconds)	Used by a delay shape at the end of the orchestrations. This setting allows all buffered BAM API calls to complete prior to the orchestration terminating
DebugMode	False for Production, True for development or test	Enables additional logging
EnforceRecordCount	False	Record counts are gathered but not compared on inbound and outbound messages as the structures differ greatly.
MapToMOA	True	Only used in a development environment

9.9.2 Receive Side Configuration

Once the Provider Enrollment configuration settings have been entered settings for the receive and send archiver components can also be entered into Enterprise Library. These settings are used to specify the archive location, message type and sender and receiver parameters.

The following screen shots show how to locate the receive-side settings for Provider Enrollment and the EFT when editing the main BTSNTSvc.exe.config file.





Receive Pipeline Configuration Settings	Typical Value	Description
Name	ProviderEnroll	The application name
DebugMode	False	Enables/disables additional logging
Archive Days	30	A value in days for how long a received message will be retained in the archive
Ports	Collection	A collection of receive ports (locations) for this application
Process Type	ProviderEnroll	The process type used in tracking data

Receive Port Settings		
Port configuration	**Typical Value**	**Description**
Port Name	ProviderEnroll_InitEnroll_XML	The name of the receive port as defined in the binding message
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\FileProcessing\PE\Enroll\FIQM\In\Request\Archive	The archive message folder for this port
Message Type	PE_FIQM_REQ	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	BTS	The recipient of the message
SenderId	FIQM	The sender of the message

Port configuration	Typical Value	Description
Port Name	ProviderEnroll_EnrollRequest_SOCKET	The name of the receive port defined in the bindings
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\FileProcessing\PE\Enroll\iC\In\Archive	The archive message folder for this port
Message Type	PE_IC_RESP	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	BTS	Message Recipient
SenderId	IC	Message Sender

Port configuration	Typical Value	Description
Port Name	ProviderEnroll_EnrollResponse_SOAP	The name of the receive port as defined in the bindings
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\FileProcessing\PE\Enroll\FIQM\In\Ack\Archive	The archive message folder for this port
Message Type	PE_FIQM_RESP_ACK	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE

Port configuration	Typical Value	Description
ReceiverId	BTS	Message Recipient
SenderId	FIQM	Message Sender

Receive Pipeline Configuration Settings	Typical Value	Description
Name	ProviderEnroll-EFT	The application name
DebugMode	False	Enables/disables additional logging
Archive Days	30	A value in days for how long a received message will be retained in the archive
Ports	Collection	A collection of receive ports (locations) for this application
Process Type	ProviderEnroll-EFT	The process type used in tracking data

Receive Port Settings		
Port configuration	**Typical Value**	**Description**
Port Name	ProviderEnroll_EFTResponseIC_XML	The name of the receive port as defined in the binding message
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\fileprocessing\PE\EFT\iC\In\Archive	The archive message folder for

Receive Port Settings		
		this port
Message Type	PE_EFT_IC_RESP	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	BTS	The recipient of the message
SenderId	IC	The sender of the message

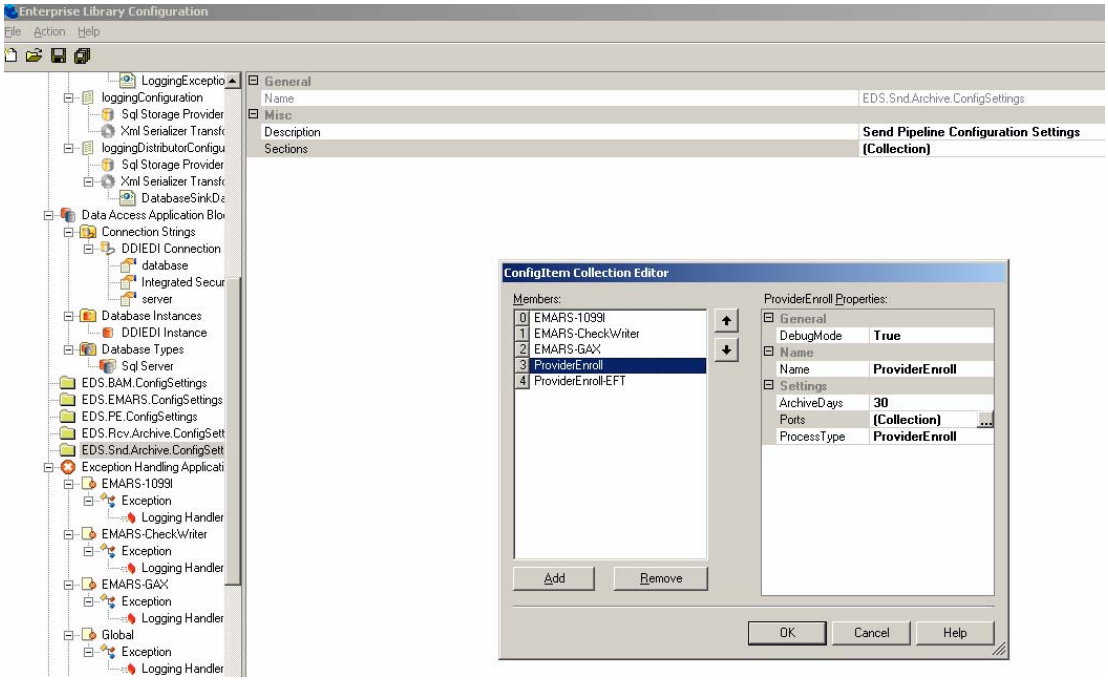
Port configuration	**Typical Value**	**Description**
Port Name	ProviderEnroll_EFTResponseIC_XML	The name of the receive port as defined in the binding message
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\fileprocessing\PE\EFT\iC\In\Archive	The archive message folder for this port
Message Type	PE_EFT_IC_RESP	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	BTS	The recipient of the message
SenderId	IC	The sender of the message

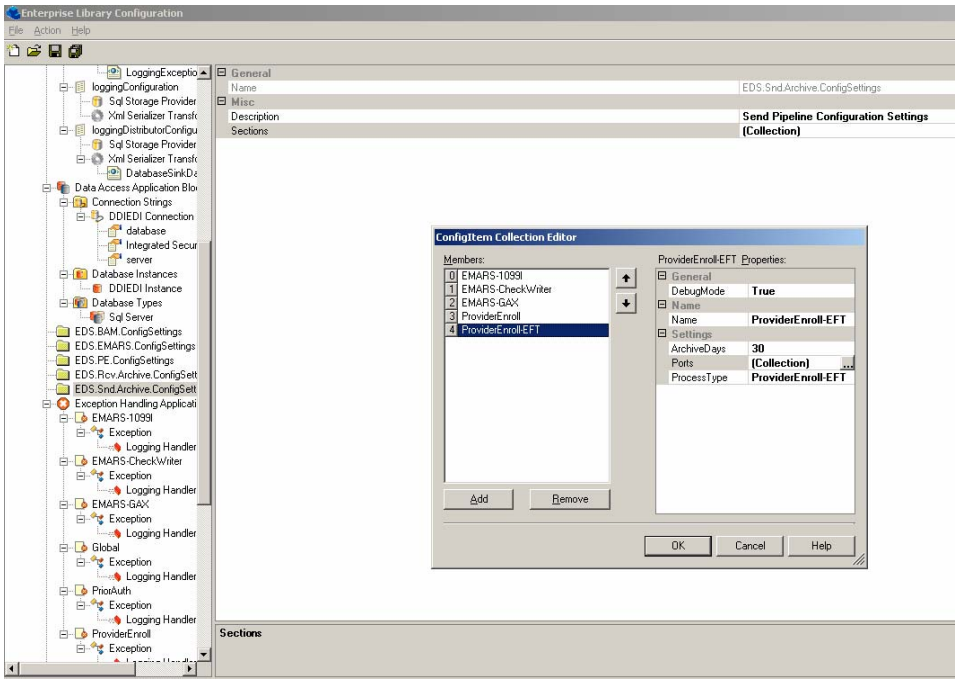
Port configuration	Typical Value	Description
Port Name	ProviderEnroll_EFTResponseFIQM_XML	The name of the receive port defined in the bindings
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\fileprocessing\PE\EFT\FIQM\In\Archive	The archive message folder for this port
Message Type	PE_FIQM_EFT_ACK	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	BTS	Message Recipient
SenderId	FIQM	Message Sender

9.9.3 Send Side Configuration

The following screen shots show how to locate the send-side settings for Provider Enrollment and the EFT when editing the main BTSNTSvc.exe.config file.

These settings are used to specify the archive location, message type and sender and receiver parameters for the send archiver.





Send Pipeline Configuration Settings	Typical Value	Description
Name	ProviderEnroll	The application name
DebugMode	False	Enables/disables additional logging
Archive Days	30	A value in days for how long a

Send Pipeline Configuration Settings	Typical Value	Description
		received message will be retained in the archive
Ports	Collection	A collection of receive ports (locations) for this application
Process Type	ProviderEnroll	The process type used in tracking data

Send Port Settings		
Port configuration	**Typical Value**	**Description**
Port Name	ProviderEnroll_InitEnroll_XML	The name of the receive port as defined in the binding message
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\FileProcessing\PE\Enroll\FIQM\Out\Ack\Archive	The archive message folder for this port
Message Type	PE_FIQM_REQ_ACK	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	FIQM	The recipient of the message
SenderId	BTS	The sender of the message

Port configuration	Typical Value	Description
Port Name	ProviderEnroll_EnrollRequest_SOCKET	The name of the receive port defined in the bindings

Port configuration	Typical Value	Description
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\FileProcessing\PE\Enroll\iC\Out\Archive	The archive message folder for this port
Message Type	PE_IC_REQ	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	IC	Message Recipient
SenderId	BTS	Message Sender

Port configuration	Typical Value	Description
Port Name	ProviderEnroll_EnrollResponse_SOAP	The name of the receive port as defined in the bindings
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\FileProcessing\PE\Enroll\FIQM\Out\Response\Archive	The archive message folder for this port
Message Type	PE_FIQM_RESP	The request message type.
Message Type Length	0	Not used for PE
Message Type Offset	0	Not used for PE
ReceiverId	FIQM	Message Recipient
SenderId	BTS	Message Sender

Send Pipeline Configuration Settings	Typical Value	Description
Name	ProviderEnroll-EFT	The application name
DebugMode	False	Enables/disables additional logging
Archive Days	30	A value in days for how long a received message will be retained in the archive
Ports	Collection	A collection of receive ports (locations) for this application
Process Type	ProviderEnroll-EFT	The process type used in tracking data

Send Port Settings		
Port configuration	**Typical Value**	**Description**
Port Name	ProviderEnroll_EFTResponseFIQM_XML	The name of the receive port as defined in the binding message
Archive Enabled	True	Enables/Disables archiving to the file server
Archive Extension	XML	The expected message extension
Archive Folder	\\SERVERNAME\fileprocessing\PE\EFT\FIQM\Out\Archive	The archive message folder for this port
Message Type	PE_EFT_FIQM_RESP	The request message type.
Message Type Length	0	Not used for PE

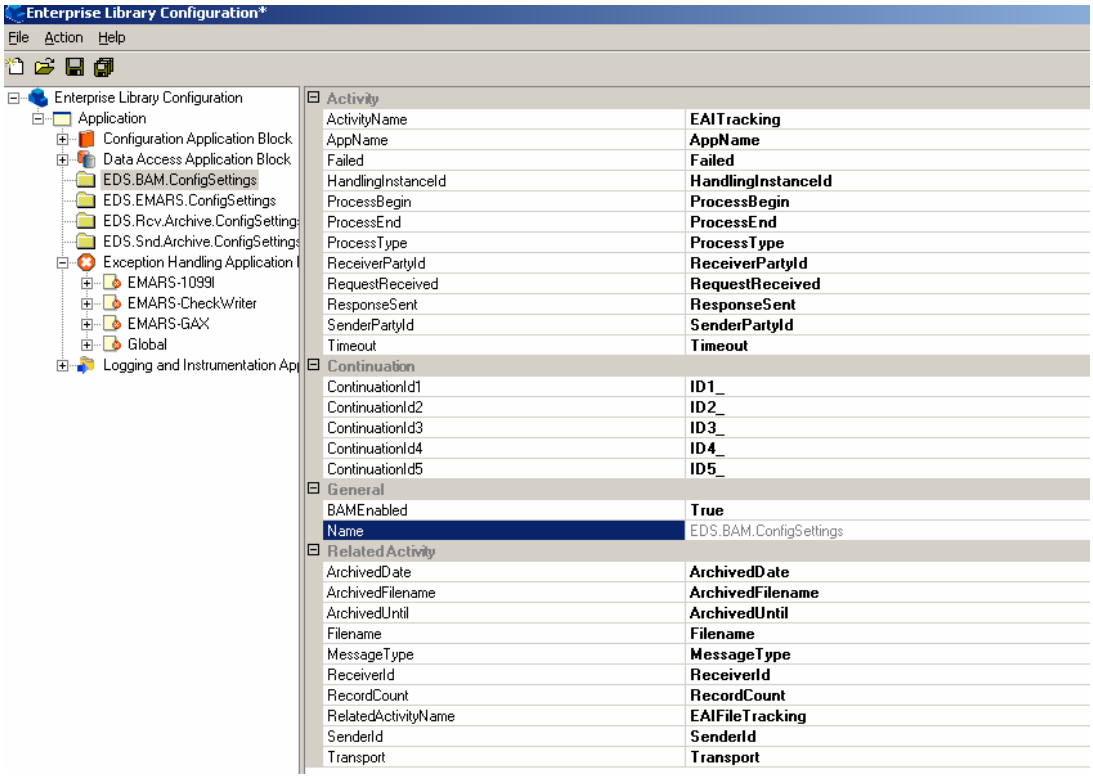
Send Port Settings		
Message Type Offset	0	Not used for PE
ReceiverId	FIQM	The recipient of the message
SenderId	BTS	The sender of the message

9.9.4 BAM Configuration

The BAM configuration setting has a single value that is used to enable/disable BAM Sink logging by the receive and send archive components and the orchestration. The BAM Enabled setting should normally be set to true for model office and production environments.

Name	**Value**
BAMEnabled	True

1. The BAM configuration section contains values that must exactly match the Excel spreadsheet BAM definition. These values are made available in Enterprise Library in case of a name change to any of the data items.



10 Appendix H – BizTalk Documentation – 270-271 Realtime

10.1 Host Configuration

10.1.1 BizTalkServerIsolatedHost

Group Name: KYADMHG\BizTalk Isolated Host Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: Isolated

10.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

10.1.1.2 Receive locations running in this host

RLRealTimeEligVerification

RLRealTimeEligVerification1

10.1.2 RealTimeProcessHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

10.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes.VerifyEligibilityProcess

10.1.2.2 Receive locations running in this host

There are no receive locations running in this host

10.2 Send Port Configuration

10.2.1 SendICRealTimeEligVerification

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: True

10.2.1.1 Primary Transport

Property	Value
Address	Socket://kysun1ap/20270
Type	SOCKET
Ordered Delivery	False
Retry Count	0
Retry Interval	5

10.2.1.2 Service Window

Enabled: False

Start Date: 6/3/2007 1:00:00 AM

End Date: 6/3/2007 12:59:00 AM

10.2.1.3 Secondary Transport

Not yet specified

10.2.1.4 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes.VerifyEligibilityProcess

10.3 Receive Port Configuration

10.3.1 ReceiveRTEligVerification

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Two-Way: True

10.3.1.1 Receive Locations

2. RLRealTimeEligVerification

Property	Value
Address	Socket://10.40.14.87/20270
Transport Protocol	SOCKET
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerIsolatedHost

3. RLRealTimeEligVerification1

Property	Value
Address	Socket://10.40.14.89/20270
Transport Protocol	SOCKET
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerIsolatedHost

10.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes.VerifyEligibilityProcess

10.4 Protocol Configuration

10.4.1 SOCKET

Delete protected: False

Uses adapter framework UI for receive handler configuration: True

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: True

Uses adapter framework UI for send port configuration: True

Starts when the service starts: False

Receive handler of adapter is hosted in-process: False

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: True

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: True

10.4.1.1 Send ports using this protocol

SendICRealTimeClaimStatus

SendICRealTimeEligVerification

10.4.1.2 Receive ports using this protocol

ReceiveRTClaimStatus

ReceiveRTEligVerification

10.5 Assembly Configuration

10.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

10.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

10.5.2 EDS.USGS.MED.EDI.Edifecs270_271Schema

Name: EDS.USGS.MED.EDI.Edifecs270_271Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

10.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs270_271Schema._271_A1_IG

EDS.USGS.MED.EDI.Edifecs270_271Schema._270_A1_IG

10.5.3 EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging

Name: EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

10.5.3.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.Edifecs270Xml_To_Edifecs271Xml

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationResponseXml_To_Edifecs271Xml

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.Edifecs270Xml_To_EligibilityVerificationRequestXml

10.5.3.2 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationRequestResponseType

10.5.4 EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes

Name: EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

10.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes.VerifyEligibilityProcess

10.6 Orchestration Configuration

10.6.1 EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes.VerifyEligibilityProcess

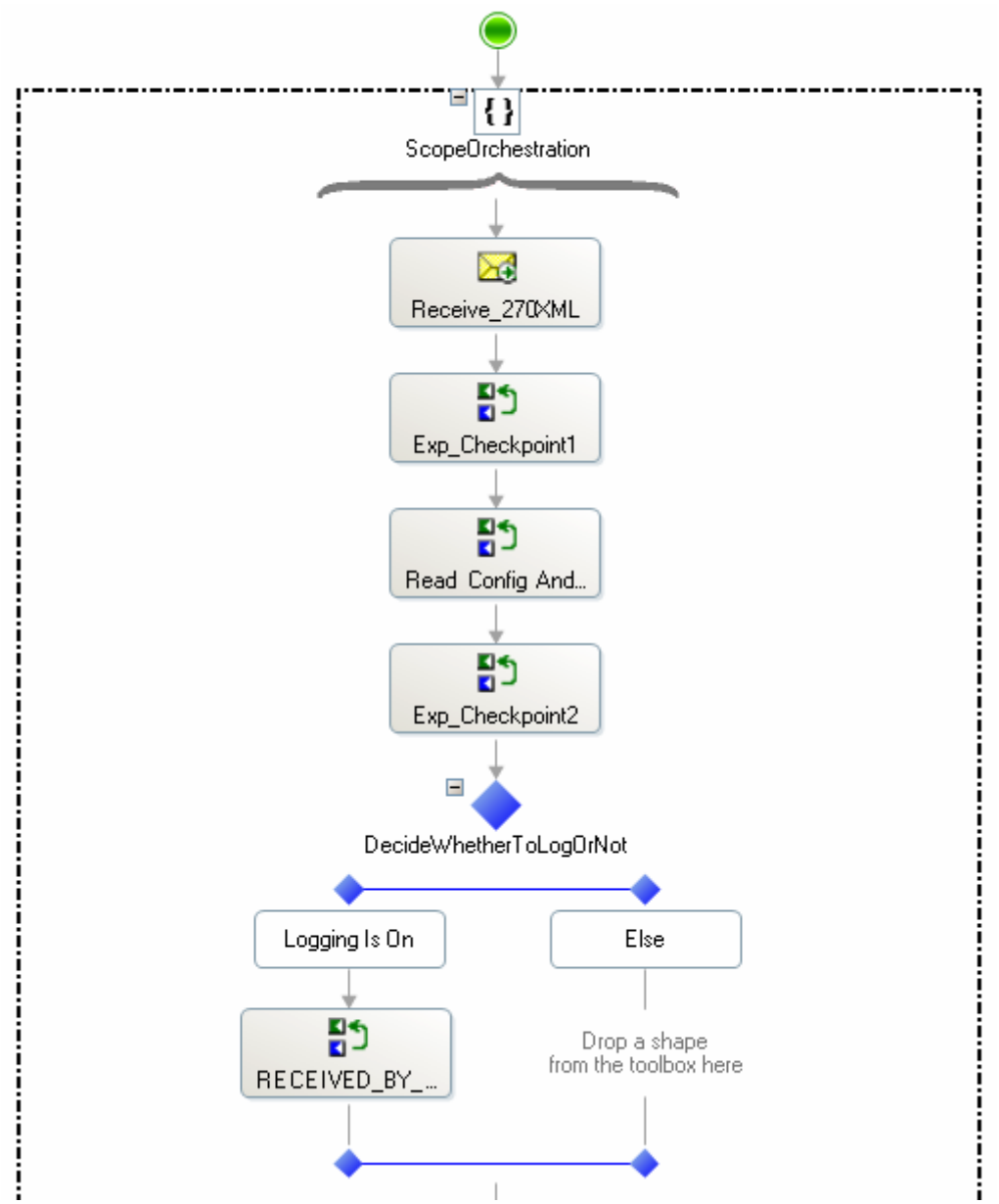
Parent Assembly:
EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

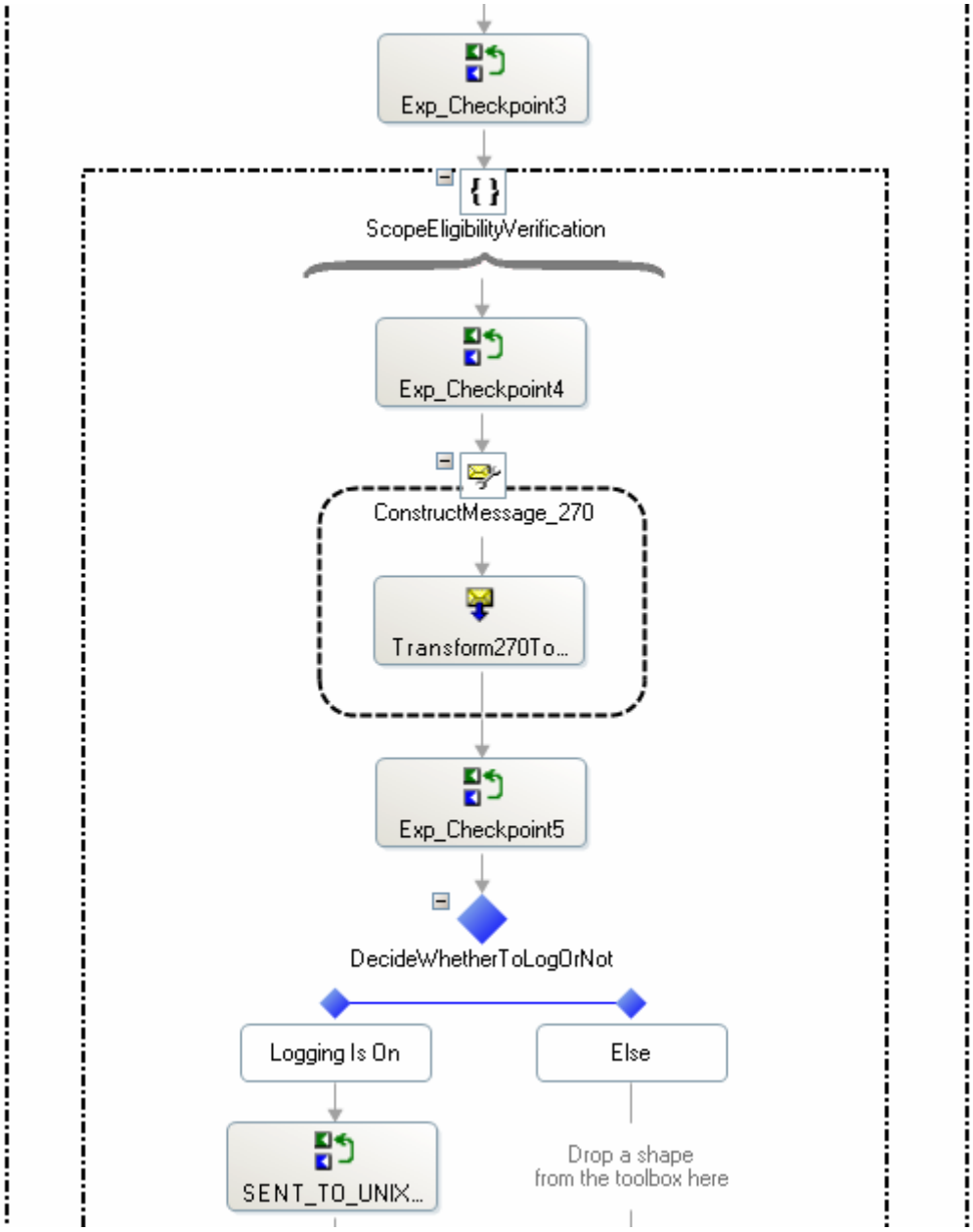
Host: BizTalkServerProcessHost

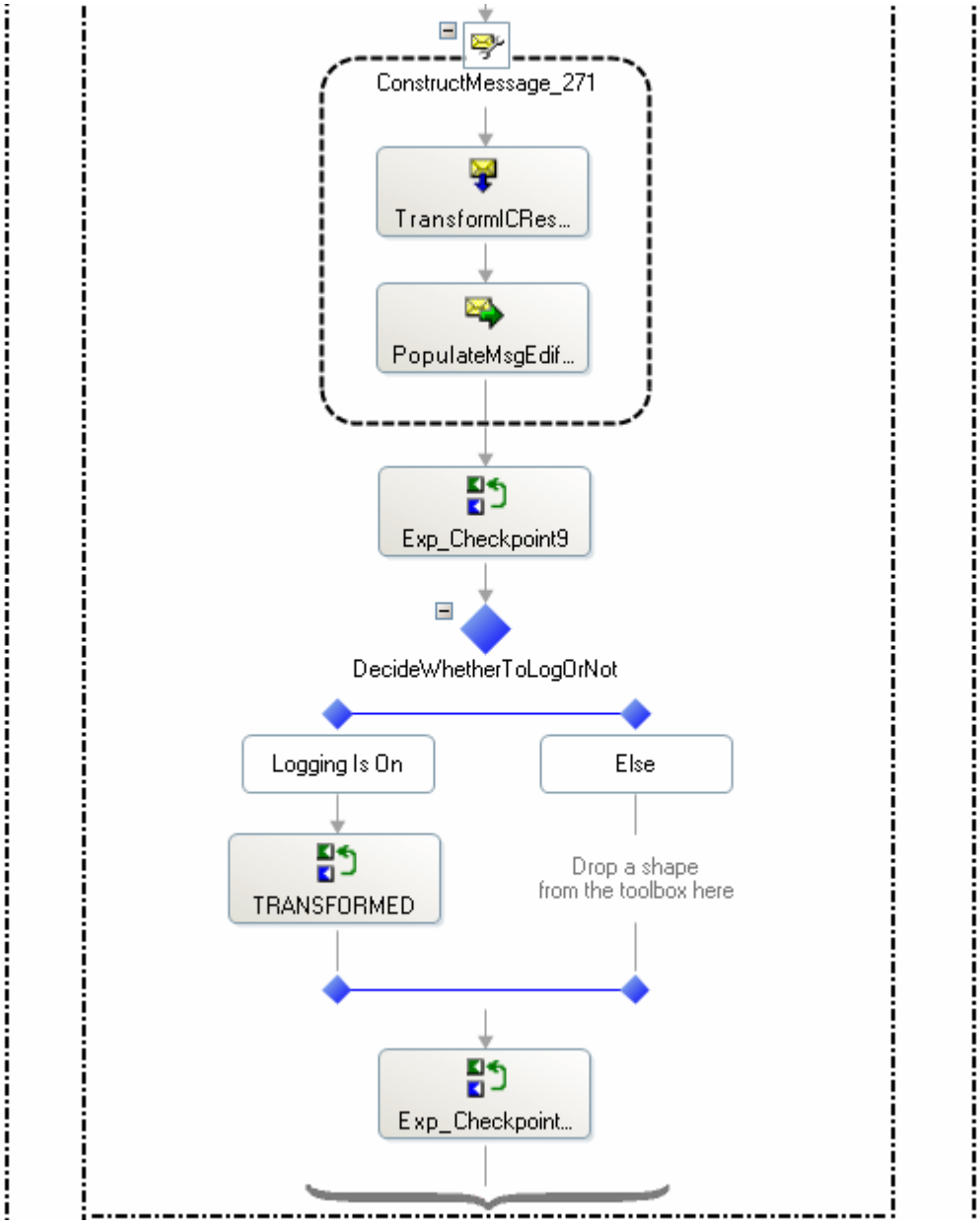
10.6.1.1 Ports contained within this orchestration

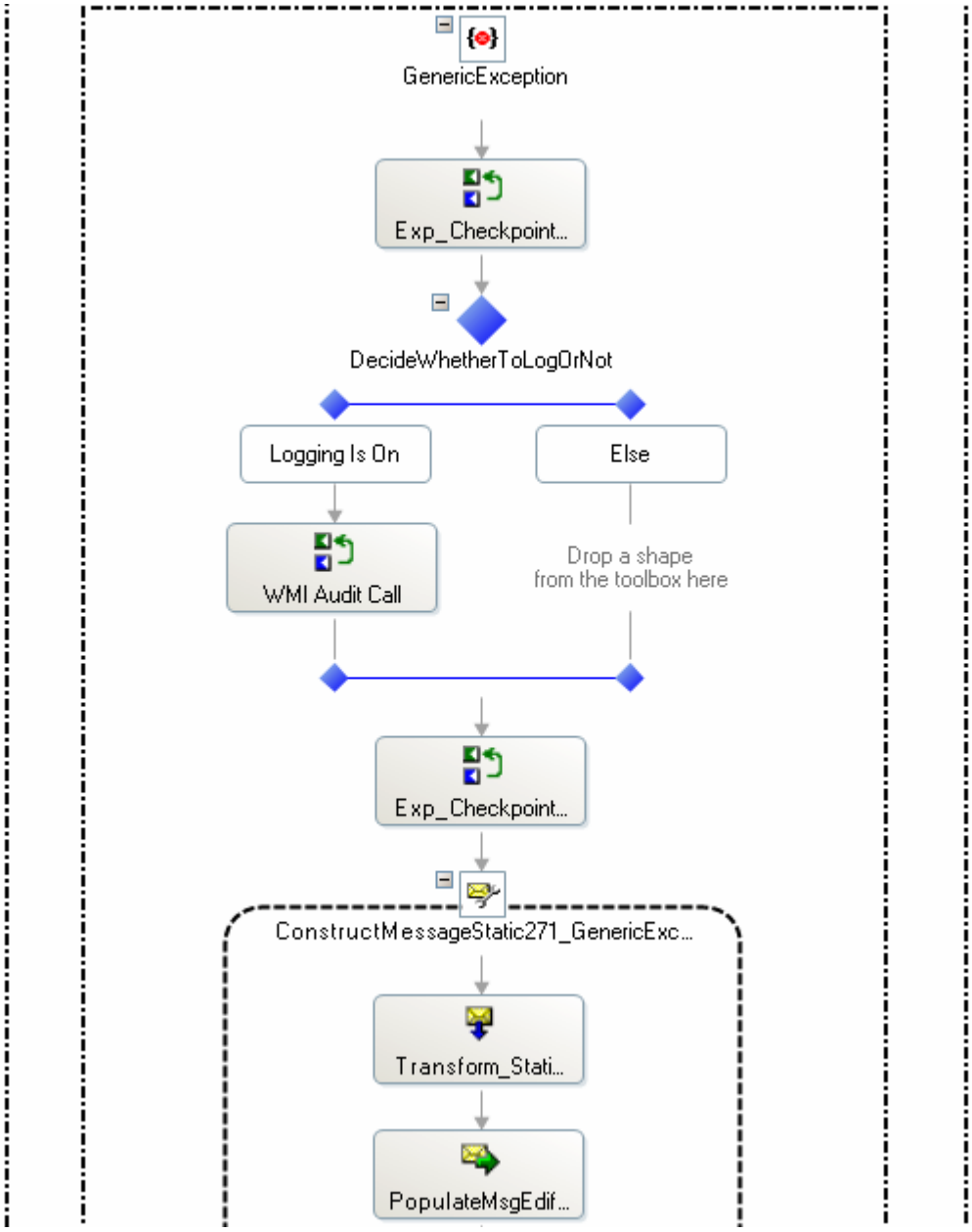
prtReceive270_Send271 - Bound to: ReceiveRTEligVerification

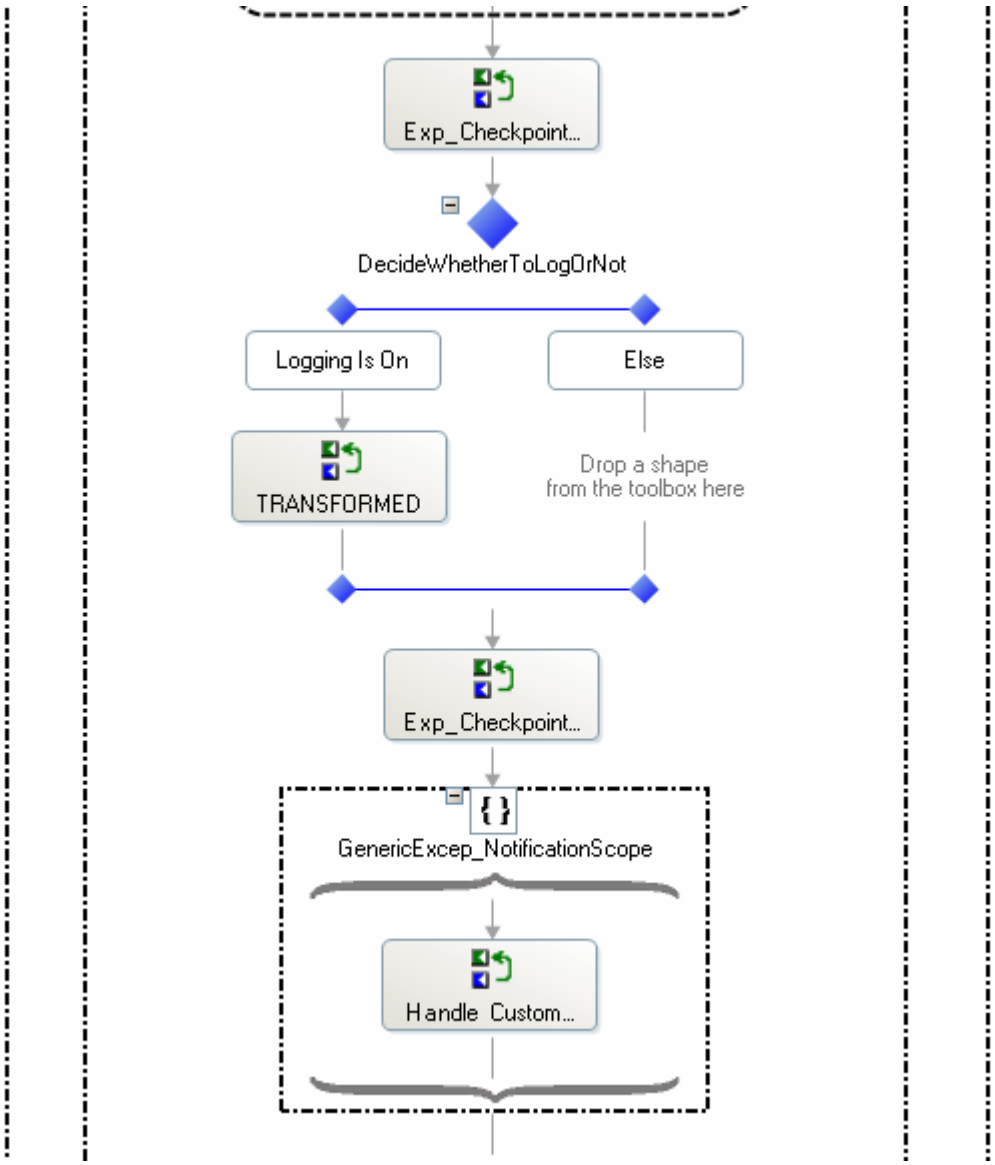
prtSendEligibilityRequest_ReceiveEligibilityResponse - Bound to: SendICRealTimeEligVerification

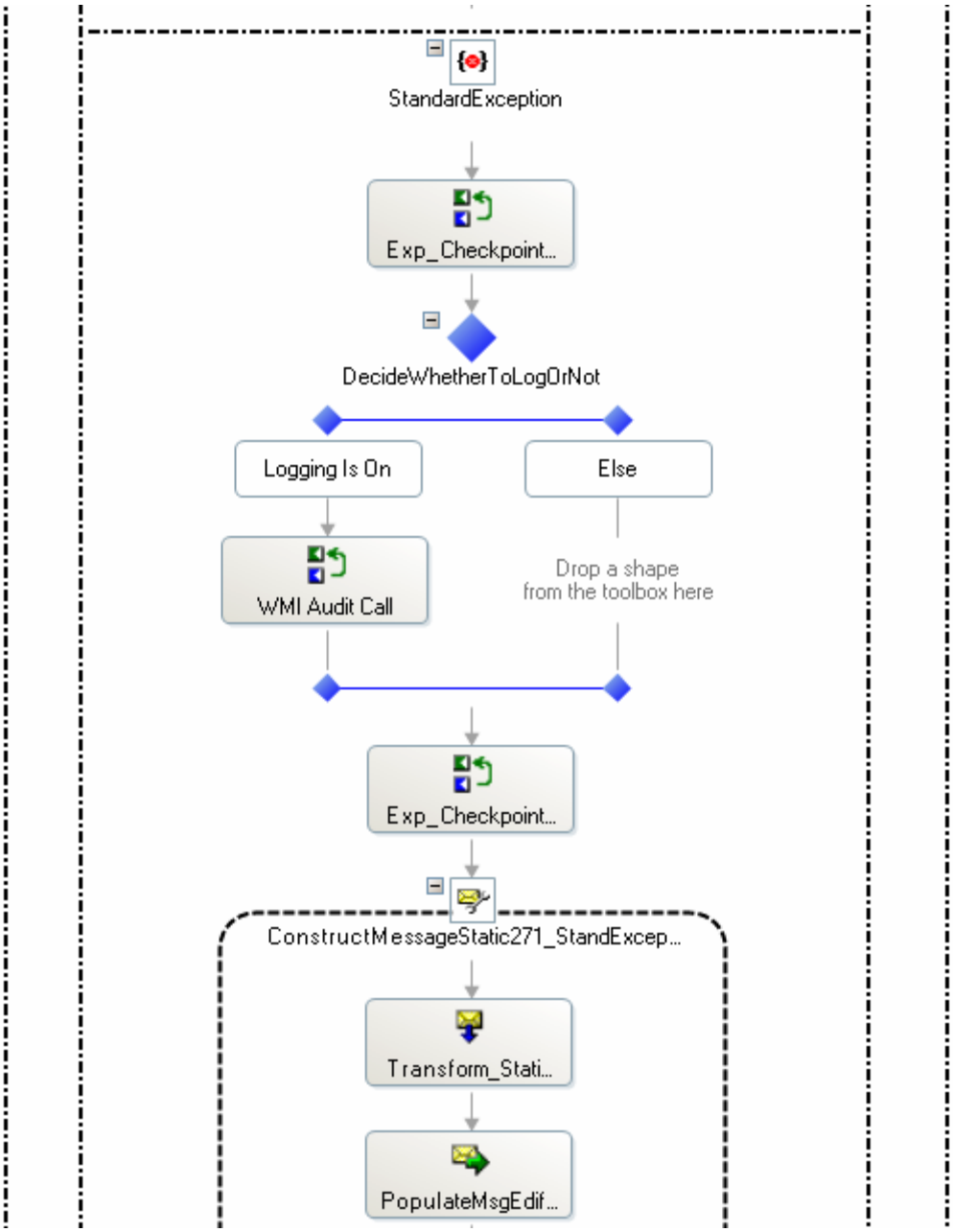


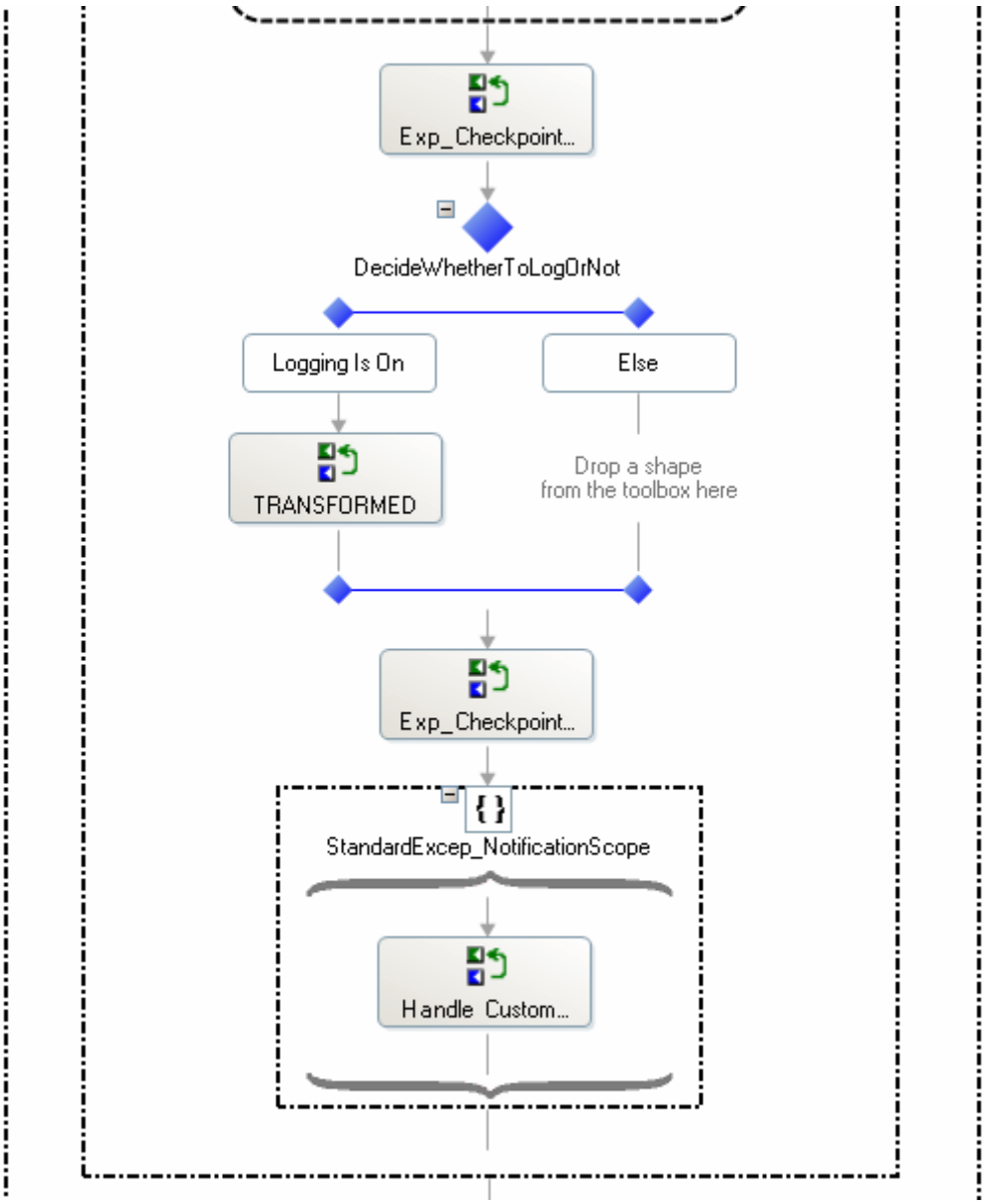

Exp_Checkpoint3
ScopeEligibilityVerification
Exp_Checkpoint4
ConstructMessage_270
Transform270To...
Exp_Checkpoint5
DecideWhetherToLogOrNot
Logging Is On
Else
SENT_TO_UNIX...
Drop a shape
from the toolbox here

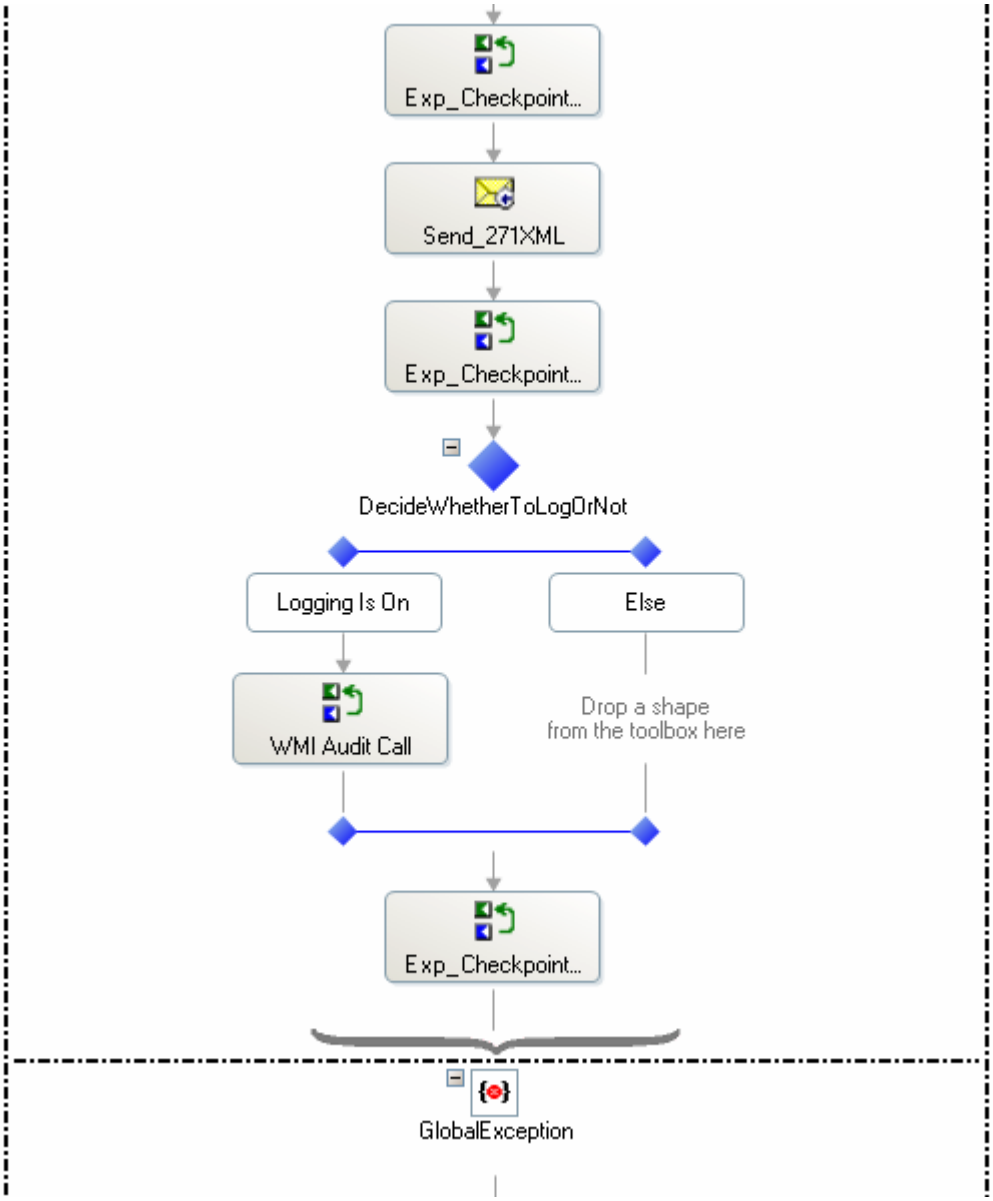

ConstructMessage_271
TransformICRes...
PopulateMsgEdif...
Exp_Checkpoint9
DecideWhetherToLogOrNot
Logging Is On
Else
TRANSFORMED
Drop a shape
from the toolbox here
Exp_Checkpoint...

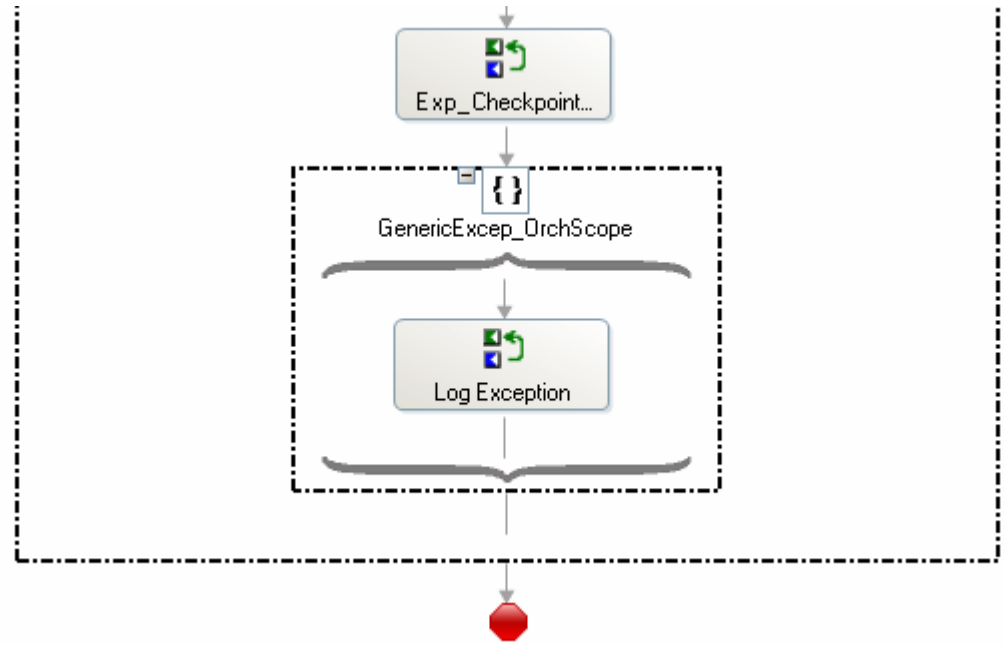

GenericException
Exp_Checkpoint...
DecideWhetherToLogOrNot
Logging Is On
Else
WMI Audit Call
Drop a shape
from the toolbox here
Exp_Checkpoint...
ConstructMessageStatic271_GenericExc...
Transform_Stati...
PopulateMsgEdif...


Exp_Checkpoint...
DecideWhetherToLogOrNot
Logging Is On
Else
TRANSFORMED
Drop a shape
from the toolbox here
Exp_Checkpoint...
GenericExcep_NotificationScope
Handle Custom...


StandardException
Exp_Checkpoint...
DecideWhetherToLogOrNot
Logging Is On
Else
WMI Audit Call
Drop a shape
from the toolbox here
Exp_Checkpoint...
ConstructMessageStatic271_StandExcep...
Transform_Stati...
PopulateMsgEdif...


Exp_Checkpoint...
DecideWhetherToLogOrNot
Logging Is On
Else
TRANSFORMED
Drop a shape
from the toolbox here
Exp_Checkpoint...
StandardExcep_NotificationScope
Handle Custom...


Exp_Checkpoint...
Send_271XML
Exp_Checkpoint...
DecideWhetherToLogOrNot
Logging Is On
Else
WMI Audit Call
Drop a shape
from the toolbox here
Exp_Checkpoint...
GlobalException


Exp_Checkpoint...
GenericExcep_OrchScope
Log Exception

10.7 Pipeline Configuration

10.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

10.7.1.1 Pipeline Process Flow

1. Decode
2. Disassemble
   - XML Disassembler
   - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
3. Validate
4. ResolveParty
   - Party Resolution
   - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

10.7.1.2 Send ports using this pipeline

SendICRealTimeClaimStatus

SendICRealTimeEligVerification

10.7.1.3 Receive locations using this pipeline

RLRealTimeClaimStatus

RLRealTimeClaimStatus1

RLRealTimeEligVerification

RLRealTimeEligVerification1

10.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

Pipeline Process Flow

5. Pre-Assemble
6. Assemble

- XML Assembler
- Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

7. Encode

10.7.2.1 Send ports using this pipeline

SendICRealTimeClaimStatus

SendICRealTimeEligVerification

10.7.2.2 Receive locations using this pipeline

There are no receive locations using this pipeline

11 Appendix I – Biz Talk Documentation – 276-277 Realtime

11.1 Host Configuration

11.1.1 BizTalkServerIsolatedHost

Group Name: KYADMHG\BizTalk Isolated Host Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: Isolated

11.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

11.1.1.2 Receive locations running in this host

RLRealTimeClaimStatus

RLRealTimeClaimStatus1

RLRealTimeEligVerification

RLRealTimeEligVerification1

11.1.2 RealTimeProcessHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

11.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes.VerifyClaimStatusProcess

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Processes.VerifyEligibilityProcess

11.1.2.2 Receive locations running in this host

There are no receive locations running in this host

11.2 Send Port Configuration

11.2.1 SendICRealTimeClaimStatus

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: True

11.2.1.1 Primary Transport

Property	Value
Address	Socket://kysun1ap/20276
Type	SOCKET
Ordered Delivery	False
Retry Count	0
Retry Interval	5

11.2.1.2 Service Window

Enabled: False

Start Date: 6/3/2007 1:00:00 AM

End Date: 6/3/2007 12:59:00 AM

11.2.1.3 Secondary Transport

Property	Value
Address	Socket://10.40.13.230/20276
Type	SOCKET
Ordered Delivery	False
Retry Count	0
Retry Interval	5

11.2.1.4 Service Window

Enabled: False

Start Date: 6/3/2007 1:00:00 AM

End Date: 6/3/2007 12:59:00 AM

11.2.1.5 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes.VerifyClaimStatusProcess

11.3 Receive Port Configuration

11.3.1 ReceiveRTClaimStatus

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Two-Way: True

11.3.1.1 Receive Locations

1. RLRealTimeClaimStatus

Property	Value
Address	Socket://10.40.14.87/20276
Transport Protocol	SOCKET
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerIsolatedHost

2. RLRealTimeClaimStatus1

Property	Value
Address	Socket://10.40.14.89/20276
Transport Protocol	SOCKET
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerIsolatedHost

11.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes.VerifyClaimStatusProcess

11.4 Protocol Configuration

11.4.1 SOCKET

Delete protected: False

Uses adapter framework UI for receive handler configuration: True

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: True

Uses adapter framework UI for send port configuration: True

Starts when the service starts: False

Receive handler of adapter is hosted in-process: False

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: True

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: True

11.4.1.1 Send ports using this protocol

SendICRealTimeClaimStatus

SendICRealTimeEligVerification

11.4.1.2 Receive ports using this protocol

ReceiveRTClaimStatus

ReceiveRTClaimStatus

ReceiveRTEligVerification

ReceiveRTEligVerification

11.5 Assembly Configuration

11.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

11.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

11.5.2 EDS.USGS.MED.EDI.Edifecs276_277Schema

Name: EDS.USGS.MED.EDI.Edifecs276_277Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

11.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs276_277Schema._277_A1_IG

EDS.USGS.MED.EDI.Edifecs276_277Schema._276_A1_IG

11.5.3 EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Messaging

Name: EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

11.5.3.1 Maps contained in this assembly

GenericClaimStatusVerification276_277Messaging.ClaimStatusResponseXml_To_Edifecs277Xml

GenericClaimStatusVerification276_277Messaging.Edifecs276Xml_To_ClaimStatusVerificationRequestXml

GenericClaimStatusVerification276_277Messaging.Edifecs276Xml_To_Edifecs277Xml

11.5.3.2 Schema contained in this assembly

GenericClaimStatusVerification276_277Messaging.ClaimStatusVerificationRequestResponse

11.5.4 EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes

Name: EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

11.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes.VerifyClaimStatusProcess

11.6 Orchestration Configuration

11.6.1 EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes.VerifyClaimStatusProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

11.6.1.1 Ports contained within this orchestration

prtReceive276_Send277 - Bound to: ReceiveRTClaimStatus

prtSendClaimRequest_ReceiveClaimResponse - Bound to: SendICRealTimeClaimStatus

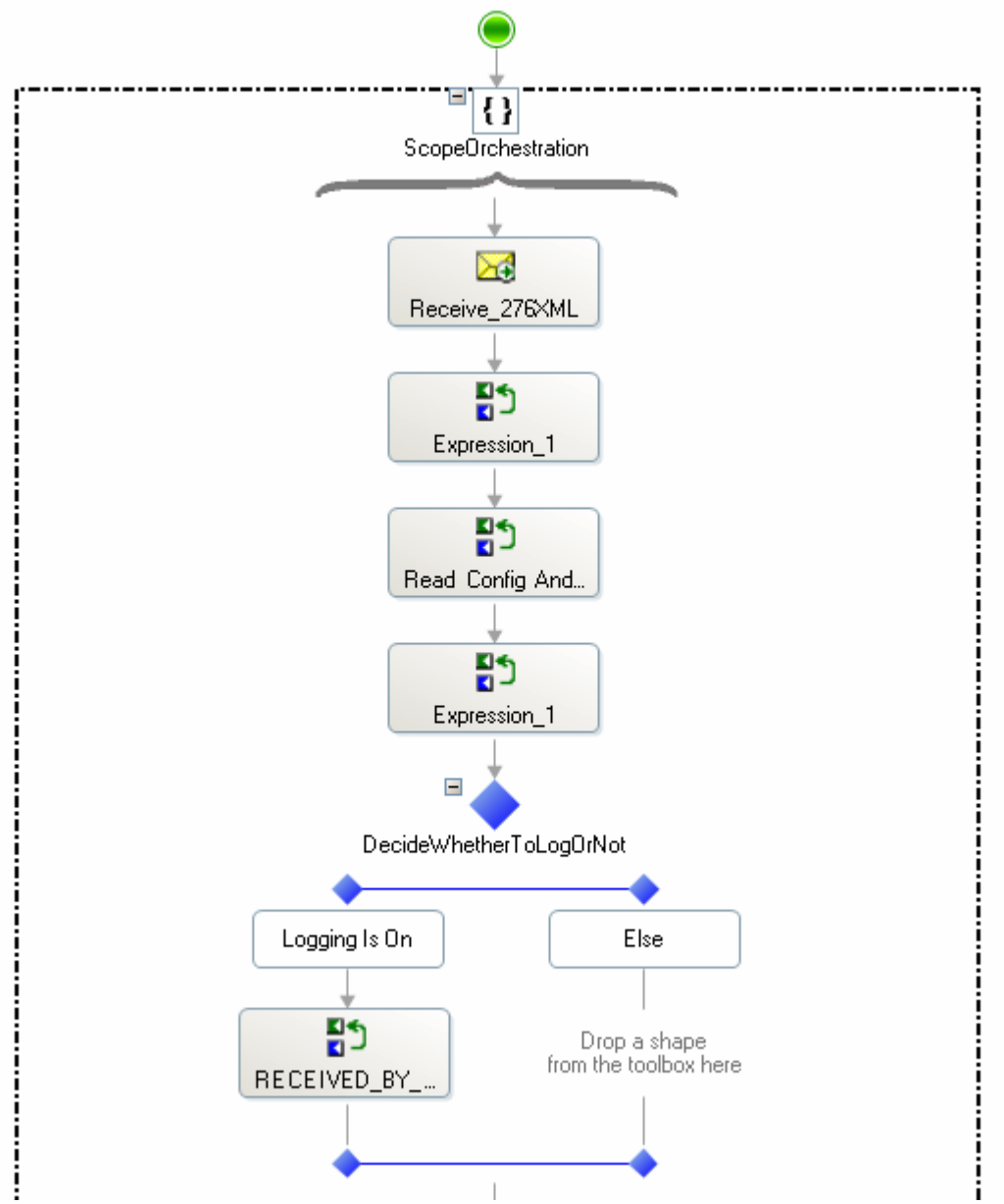
11.6.1.2 Messages defined within this orchestration

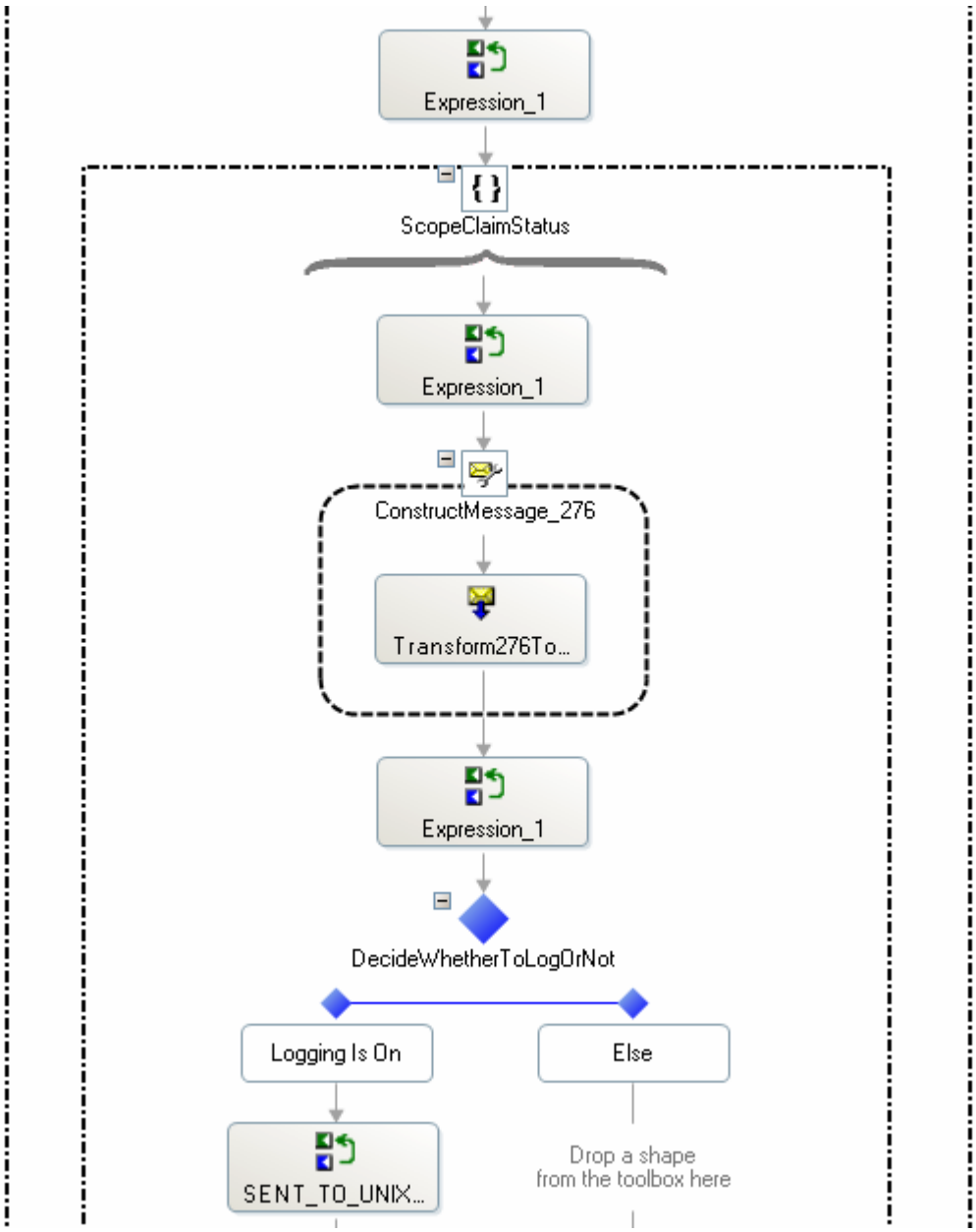
MsgInterchangeResponse-
GenericClaimStatusVerification276_277Messaging.ClaimStatusVerificationRequestResponse

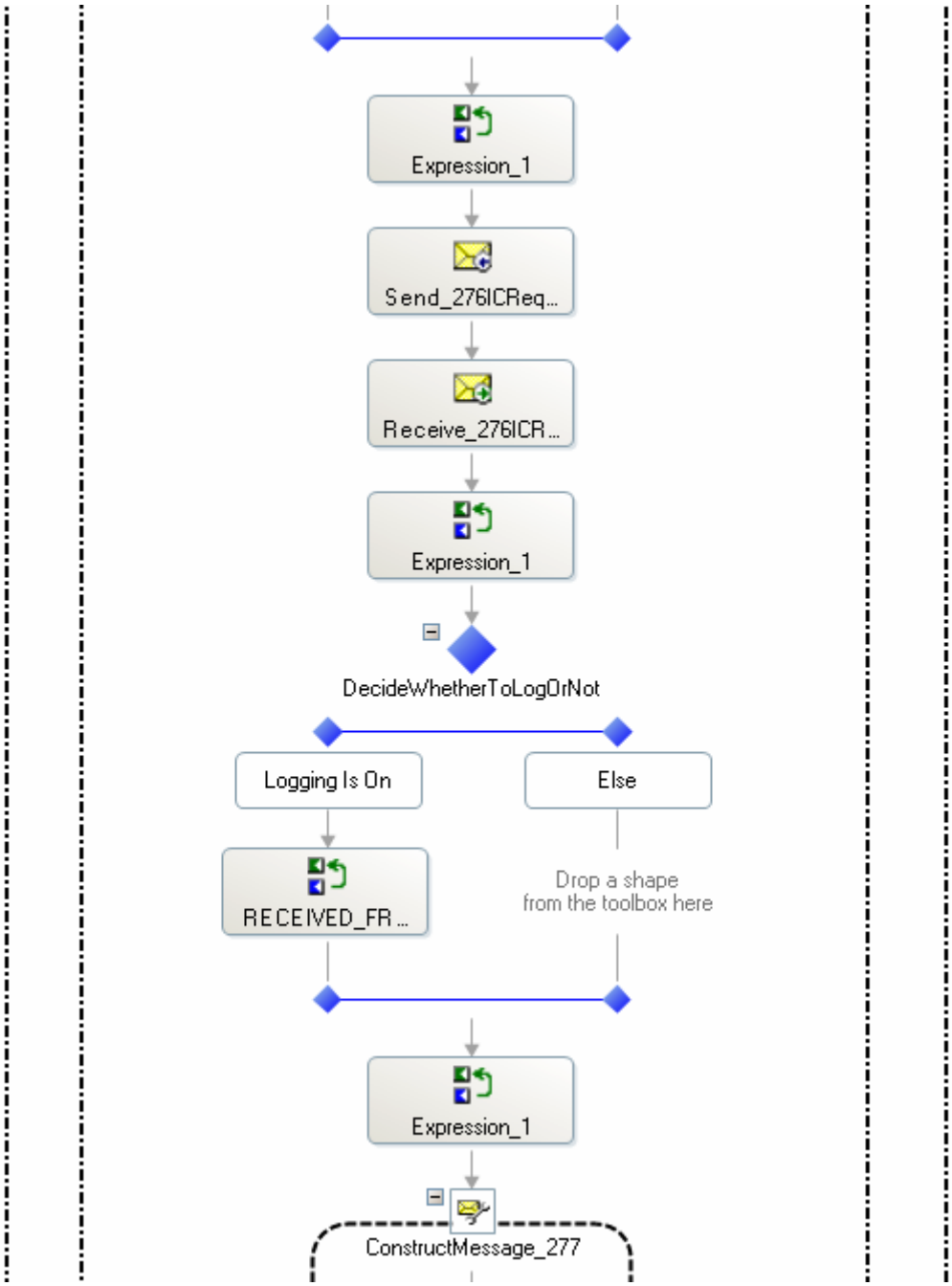
MsgInterchangeRequest-
GenericClaimStatusVerification276_277Messaging.ClaimStatusVerificationRequestResponse

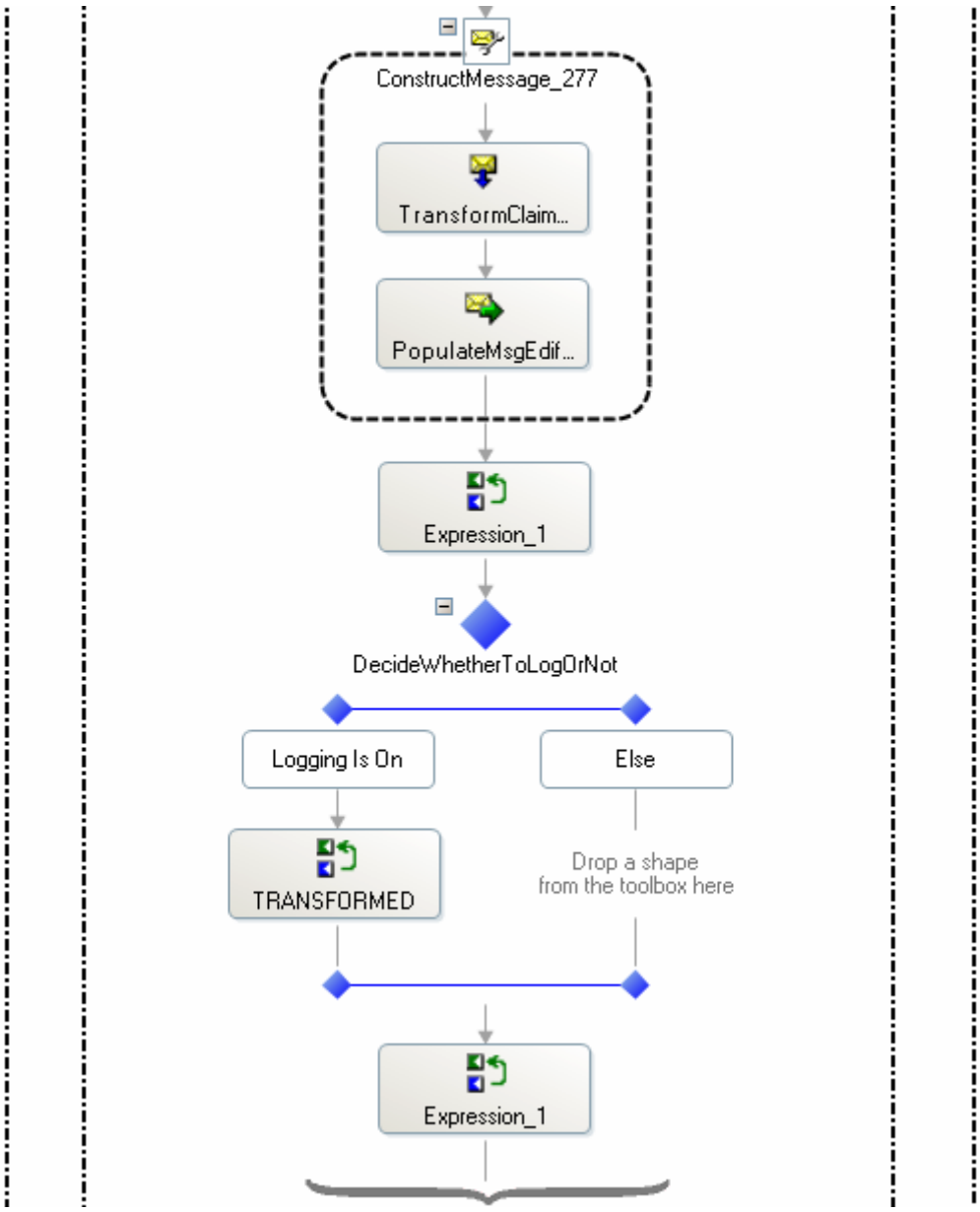
MsgEdifecs277- EDS.USGS.MED.EDI.Edifecs276_277Schema._277_A1_IG

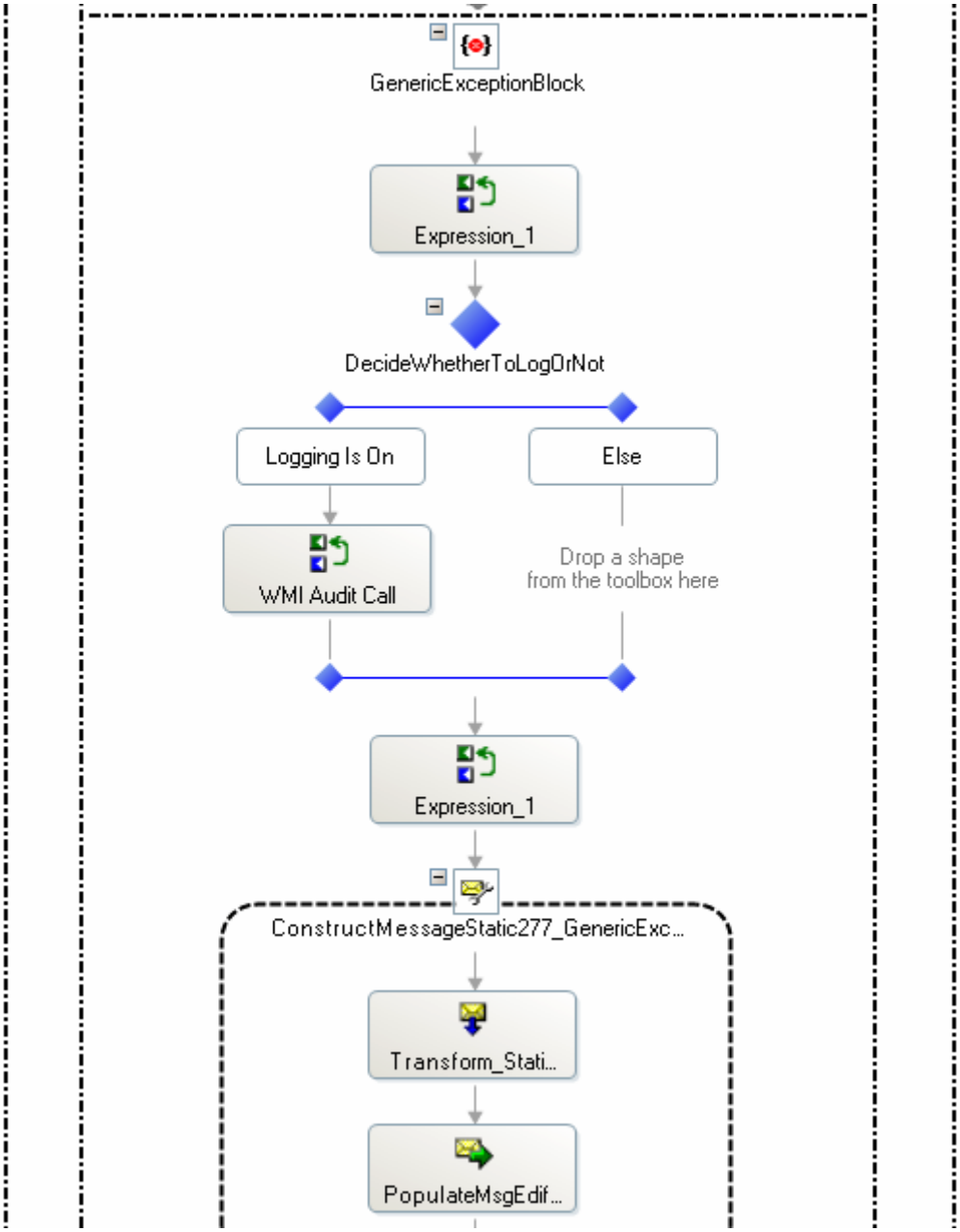
MsgEdifecs276- EDS.USGS.MED.EDI.Edifecs276_277Schema._276_A1_IG

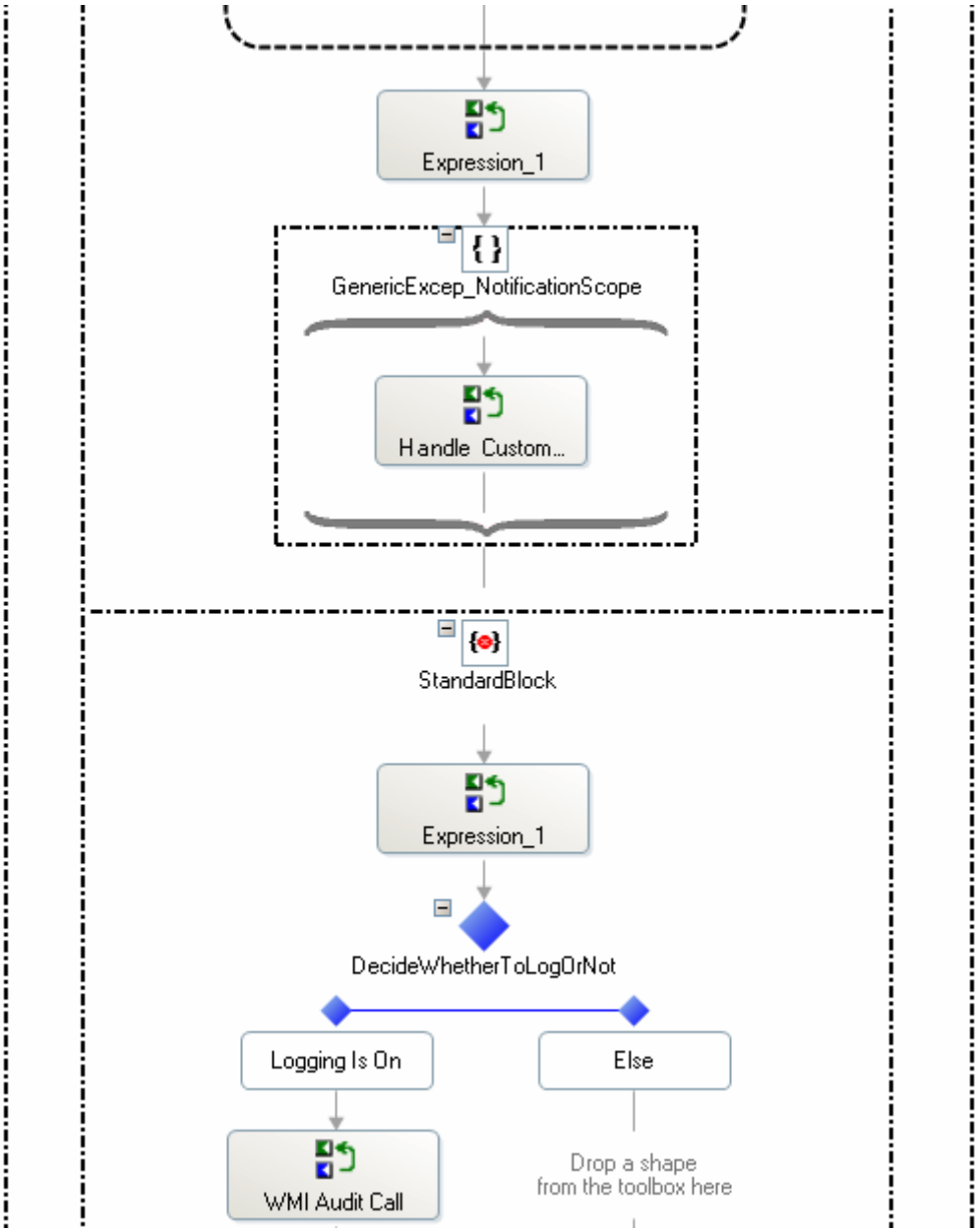

ScopeOrchestration
Receive_276XML
Expression_1
Read Config And...
Expression_1
DecideWhetherToLogOrNot
Logging Is On
Else
RECEIVED_BY_...
Drop a shape
from the toolbox here

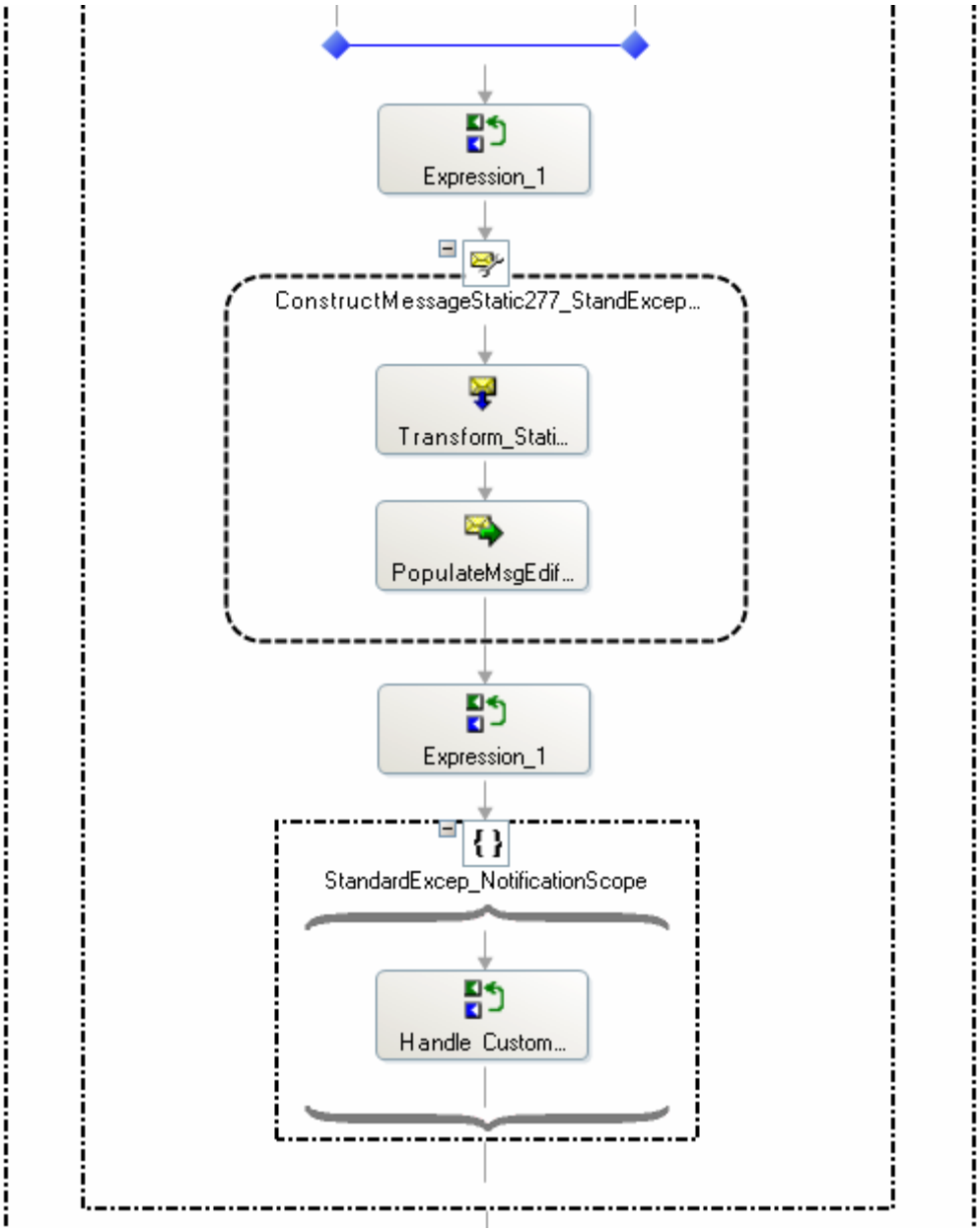

Expression_1
ScopeClaimStatus
Expression_1
ConstructMessage_276
Transform276To...
Expression_1
DecideWhetherToLogOrNot
Logging Is On
Else
SENT_TO_UNIX...
Drop a shape
from the toolbox here

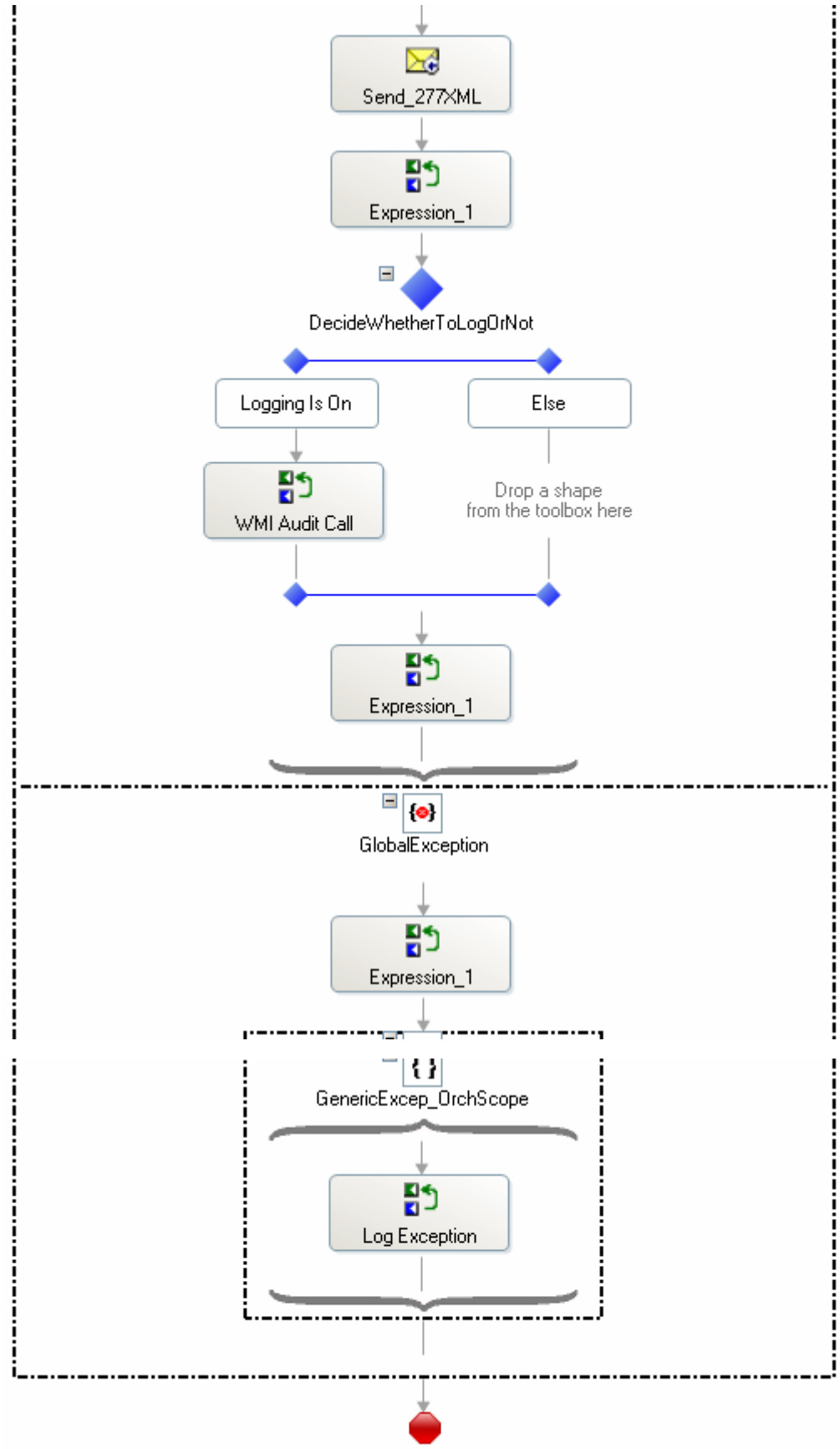

Expression_1
Send_276ICReq...
Receive_276ICR...
Expression_1
DecideWhetherToLogOrNot
Logging Is On
Else
RECEIVED_FR ...
Drop a shape
from the toolbox here
Expression_1
ConstructMessage_277


ConstructMessage_277
TransformClaim...
PopulateMsgEdif...
Expression_1
DecideWhetherToLogOrNot
Logging Is On
Else
TRANSFORMED
Drop a shape
from the toolbox here
Expression_1


GenericExceptionBlock
Expression_1
DecideWhetherToLogOrNot
Logging Is On
Else
WMI Audit Call
Drop a shape
from the toolbox here
Expression_1
ConstructMessageStatic277_GenericExc...
Transform_Stati...
PopulateMsgEdif...


Expression_1
GenericExcep_NotificationScope
Handle Custom...
StandardBlock
Expression_1
DecideWhetherToLogOrNot
Logging Is On
Else
WMI Audit Call
Drop a shape
from the toolbox here


Expression_1
ConstructMessageStatic277_StandExcep...
Transform_Stati...
PopulateMsgEdif...
Expression_1
StandardExcep_NotificationScope
Handle Custom...


Send_277XML
Expression_1
DecideWhetherToLogOrNot
Logging Is On
Else
WMI Audit Call
Drop a shape
from the toolbox here
Expression_1
GlobalException
Expression_1
GenericExcep_OrchScope
Log Exception

11.7 Pipeline Configuration

11.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

11.7.1.1 Pipeline Process Flow

3. Decode
4. Disassemble
    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
5. Validate
6. ResolveParty
    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

11.7.1.2 Send ports using this pipeline

SendICRealTimeClaimStatus

SendICRealTimeEligVerification

11.7.1.3 Receive locations using this pipeline

RLRealTimeClaimStatus

RLRealTimeClaimStatus1

RLRealTimeEligVerification

RLRealTimeEligVerification1

11.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

11.7.2.1 Pipeline Process Flow

1. Pre-Assemble
2. Assemble
    - XML Assembler

- Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

3. Encode

11.7.2.2 Send ports using this pipeline

SendICRealTimeClaimStatus

SendICRealTimeEligVerification

11.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

12 Appendix J – Biz Talk Documentation – 270 Batch

12.1 Host Configuration

12.1.1 BizTalkServerProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

12.1.1.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType

12.1.1.2 Receive locations running in this host

There are no receive locations running in this host

12.1.2 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

12.1.2.1 Orchestrations running in this host

There are no orchestrations running in this host

12.1.2.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

12.1.3 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

12.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

12.1.3.2 Receive locations running in this host

There are no receive locations running in this host

12.2 Send Port Configuration

12.2.1 SendICEligVerificationReq

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: False

12.2.1.1 Primary Transport

Property	Value
Address	\\10.40.14.111\fts\Generic\Eligibility270_271\Inbound270\InterChange\%SourceFileName%.xml
Type	FILE
Ordered Delivery	False
Retry Count	3
Retry Interval	5

12.2.1.2 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

12.2.1.3 Secondary Transport

Property	Value
Address	\\10.40.14.111\fts\Generic\Eligibility270_271\Inbound270\InterChange\%SourceFileName%.xml
Type	FILE
Ordered Delivery	False
Retry Count	3
Retry Interval	5

12.2.1.4 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

12.2.1.5 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

12.3 Receive Port Configuration

12.3.1 ReceiveEligVerificationReq

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

12.3.1.1 Receive Locations

1. RLEligVerificationReqGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\Eligibility270_271\Inbound270\XData*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

12.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

12.4 Protocol Configuration

12.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

12.4.1.1 Send ports using this protocol

SendICEligVerificationReq

12.4.1.2 Receive ports using this protocol

ReceiveEligVerificationReq

12.5 Assembly Configuration

12.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

12.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

12.5.2 EDS.USGS.MED.EDI.Edifecs270_271Schema

Name: EDS.USGS.MED.EDI.Edifecs270_271Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

12.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs270_271Schema._271_A1_IG

EDS.USGS.MED.EDI.Edifecs270_271Schema._270_A1_IG

12.5.3 EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging

Name: EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

12.5.3.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.Edifecs270Xml_To_Edifecs271Xml

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationResponseXml_To_Edifecs271Xml

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.Edifecs270Xml_To_EligibilityVerificationRequestXml

12.5.3.2 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationRequestResponseType

12.5.4 EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes

Name: EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

12.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

12.6 Orchestration Configuration

12.6.1 EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerApplication

12.6.1.1 Ports contained within this orchestration

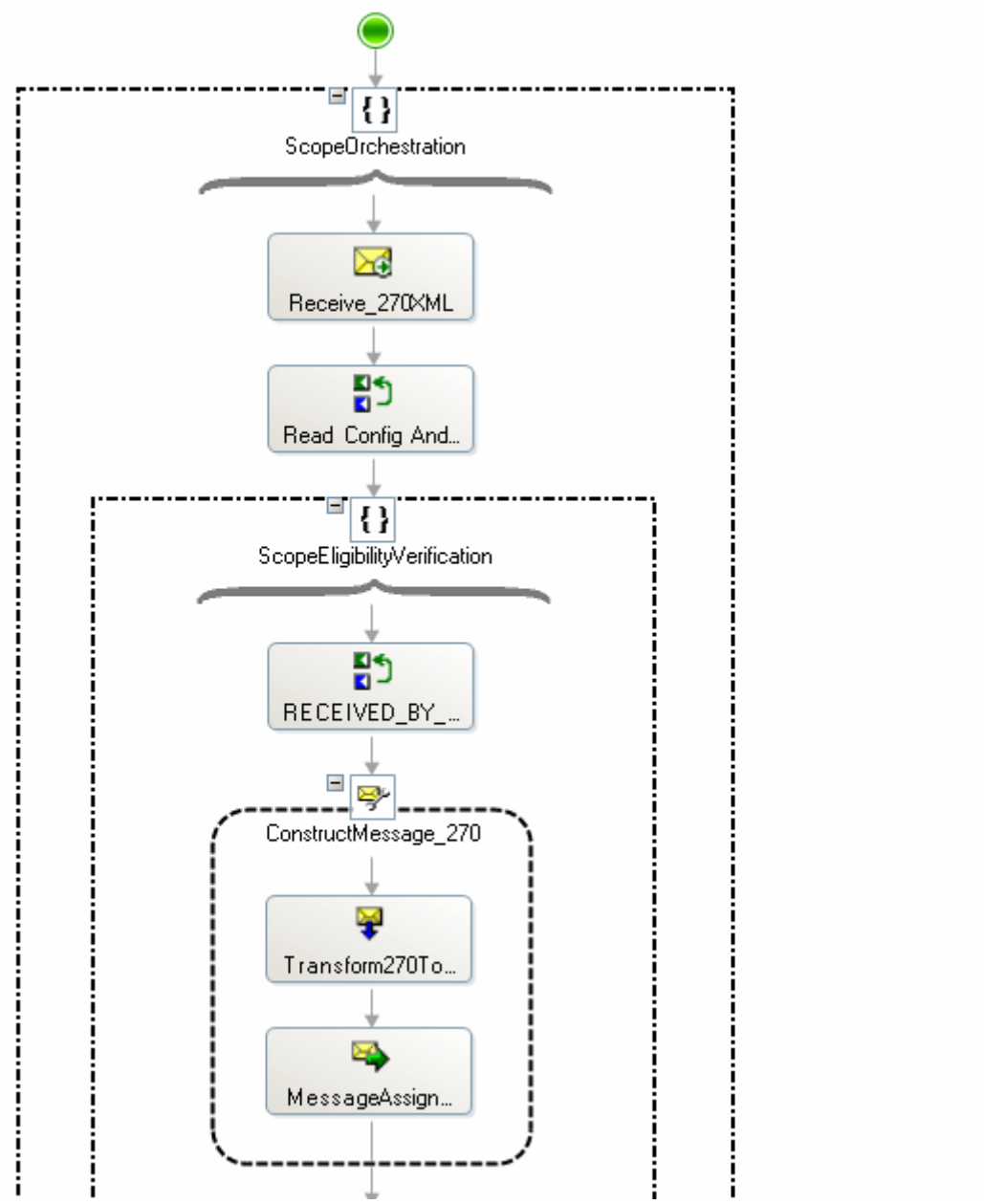
prtReceive270 - Bound to: ReceiveEligVerificationReq

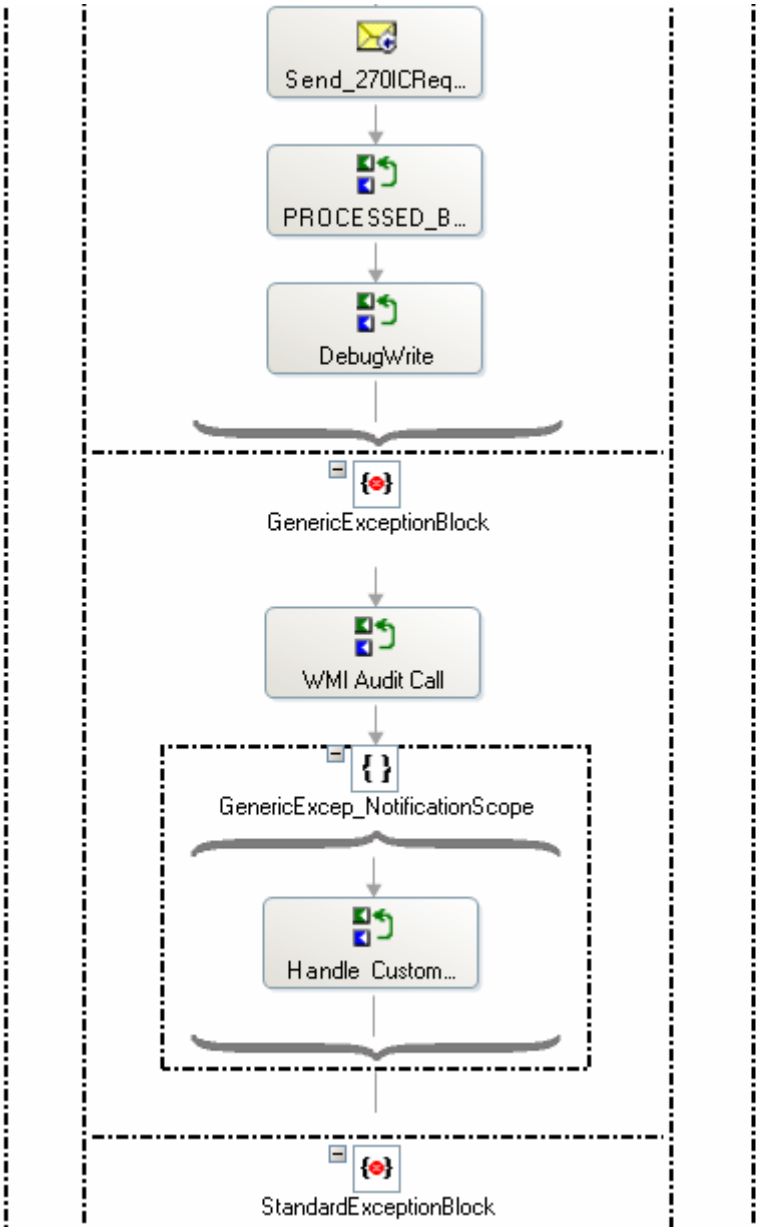
prtSendEligibilityVerificationReq - Bound to: SendICEligVerificationReq

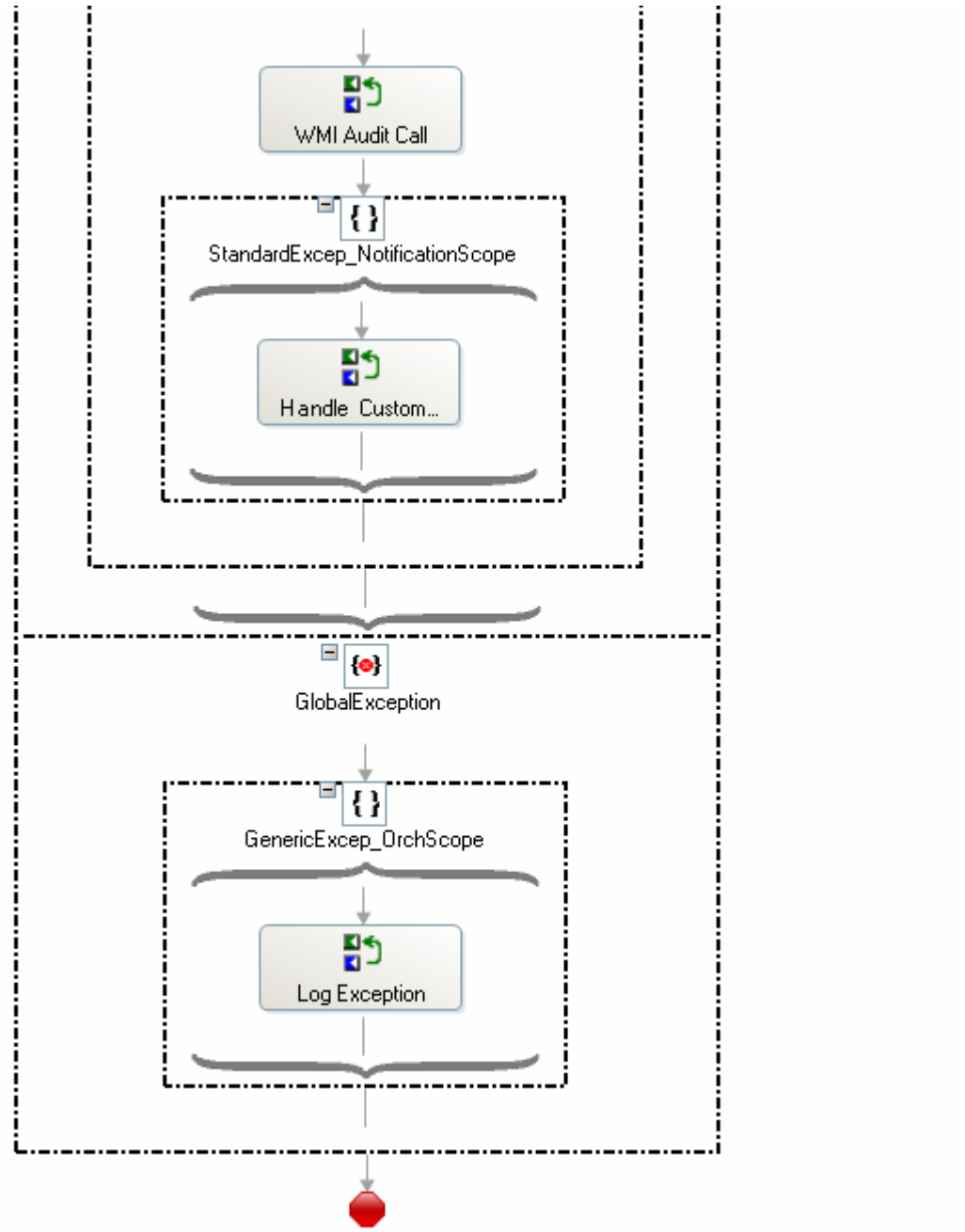
12.6.1.2 Messages defined within this orchestration

MsgEDIFECS270 - EDS.USGS.MED.EDI.Edifecs270_271Schema._270_A1_IG

MsgInterchangeRequest - EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationRequestResponseType


ScopeOrchestration
Receive_270XML
Read Config And...
ScopeEligibilityVerification
RECEIVED_BY_...
ConstructMessage_270
Transform270To...
MessageAssign...


Send_270ICReq...
PROCESSED_B...
DebugWrite
GenericExceptionBlock
WMI Audit Call
GenericExcep_NotificationScope
Handle Custom...
StandardExceptionBlock


WMI Audit Call
StandardExcep_NotificationScope
Handle Custom...
GlobalException
GenericExcep_OrchScope
Log Exception

12.7 Pipeline Configuration

12.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

12.7.1.1 Pipeline Process Flow

7. Decode
8. Disassemble
    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
9. Validate
10. ResolveParty
    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

12.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

12.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

12.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

12.7.2.1 Pipeline Process Flow

11. Pre-Assemble
12. Assemble
    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
13. Encode

12.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

12.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

13 Appendix K – Biz Talk Documentation – 271 Batch

13.1 Host Configuration

13.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

13.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

13.1.1.2 Receive locations running in this host

There are no receive locations running in this host

13.1.2 BizTalkServerProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

13.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType

13.1.2.2 Receive locations running in this host
There are no receive locations running in this host

13.1.3 BizTalkServerReceiveHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

13.1.3.1 Orchestrations running in this host
There are no orchestrations running in this host

13.1.3.2 Receive locations running in this host
RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

13.1.4 BizTalkServerSendHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

13.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

13.1.4.2 Receive locations running in this host

There are no receive locations running in this host

13.2 Send Port Configuration

13.2.1 SendXDATA271

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: False

13.2.1.1 Primary Transport

Property	Value
Address	\\10.40.14.111\fts\Generic\Eligibility270_271\Outbound271\XData\%SourceFileName%.xml
Type	FILE
Ordered Delivery	False
Retry Count	3
Retry Interval	5

13.2.1.2 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

13.2.1.3 Secondary Transport

Property	Value
Address	\\10.40.14.111\fts\Generic\Eligibility270_271\Outbound271\XData\%SourceFileName%.xml
Type	FILE
Ordered Delivery	False
Retry Count	3
Retry Interval	5

13.2.1.4 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

13.2.1.5 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

13.3 Receive Port Configuration

13.3.1 ReceiveICEligVerificationResp

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

13.3.1.1 Receive Locations

14. RLICEligVerifRespGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\Eligibility270_271\Outbound271\Preprocessor*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

13.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

13.4 Protocol Configuration

13.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

13.4.1.1 Send ports using this protocol

SendXDATA271

13.4.1.2 Receive ports using this protocol

ReceiveICEligVerificationResp

13.5 Assembly Configuration

13.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

13.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

13.5.2 EDS.USGS.MED.EDI.Edifecs270_271Schema

Name: EDS.USGS.MED.EDI.Edifecs270_271Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

13.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs270_271Schema._271_A1_IG

EDS.USGS.MED.EDI.Edifecs270_271Schema._270_A1_IG

13.5.3 EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging

Name: EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

13.5.3.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.Edifecs270Xml_To_Edifecs271Xml

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationResponseXml_To_Edifecs271Xml

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.Edifecs270Xml_To_EligibilityVerificationRequestXml

13.5.3.2 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationRequestResponseType

13.5.4 EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes

Name: EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

13.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

13.6 Orchestration Configuration

13.6.1 EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

13.6.1.1 Ports contained within this orchestration

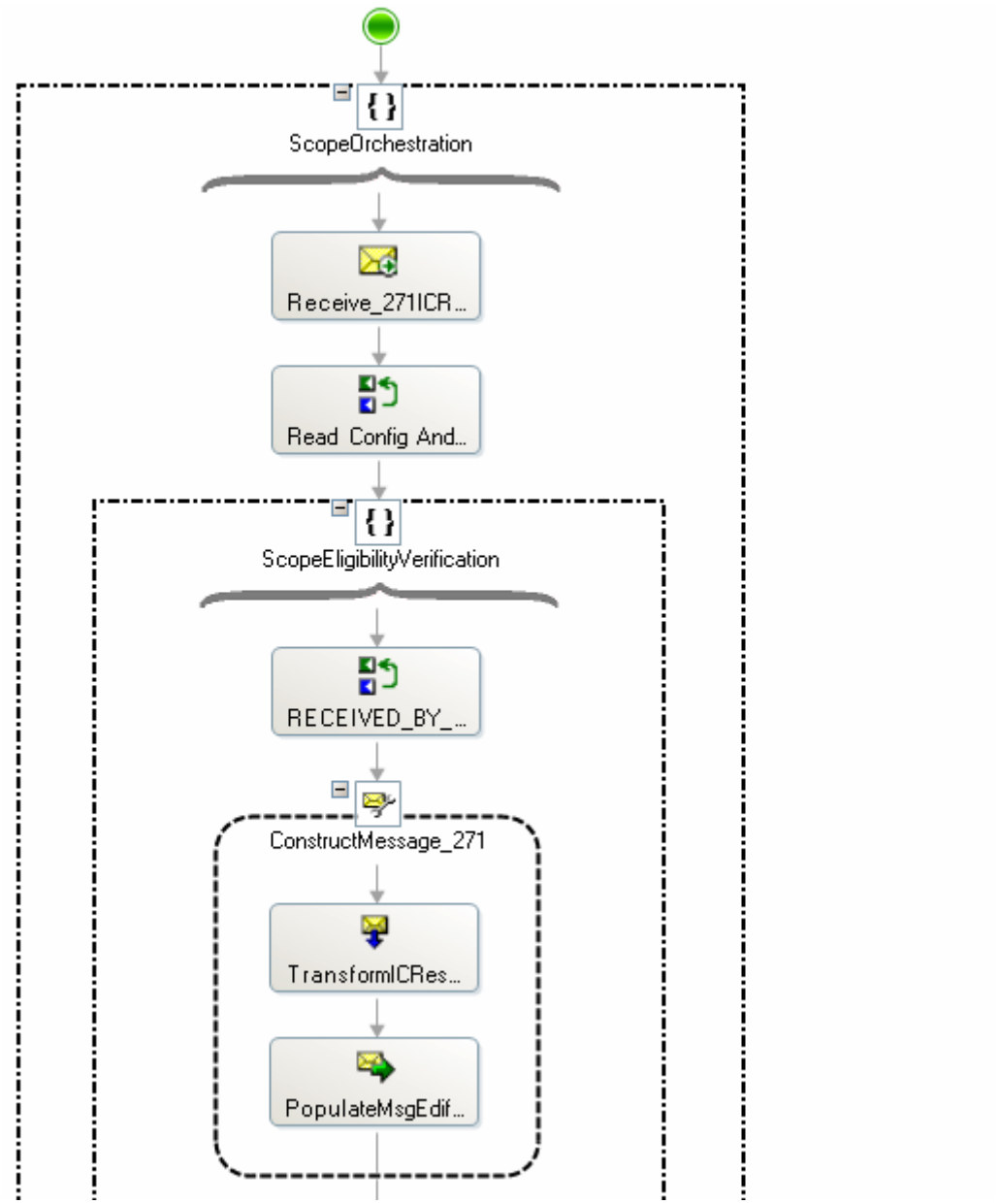
prtReceiveEligibilityVerificationResponse - Bound to: ReceiveICEligVerificationResp

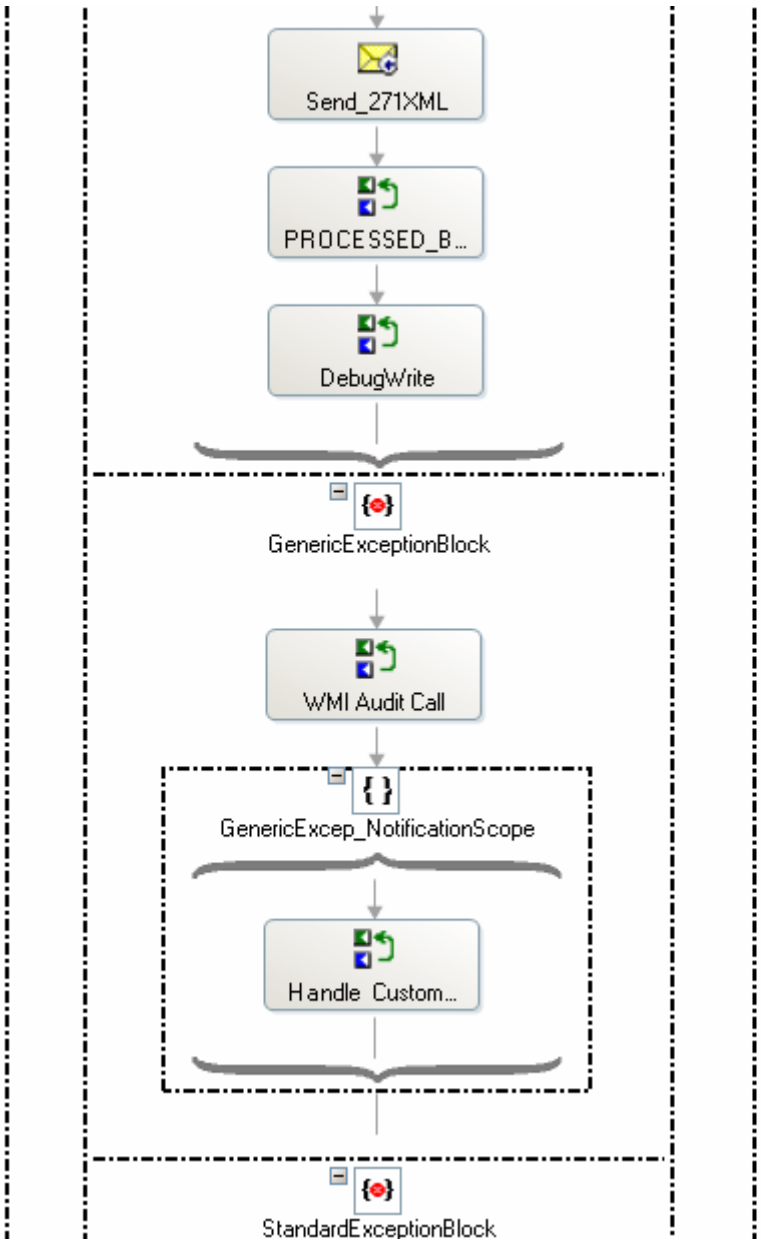
prtSend271 - Bound to: SendXDATA271

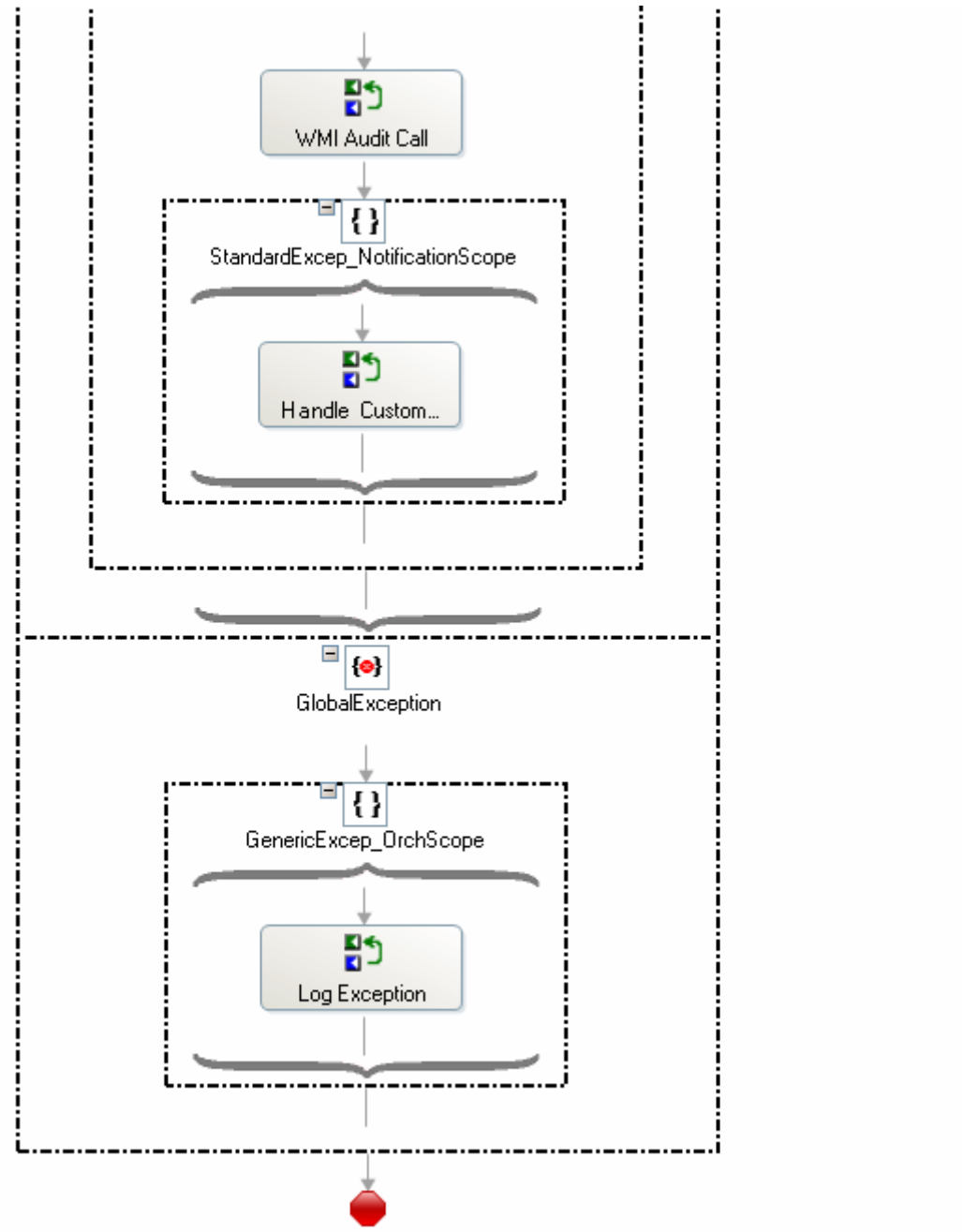
13.6.1.2 Messages defined within this orchestration

MsgEDIFECS271 - EDS.USGS.MED.EDI.Edifecs270_271Schema._271_A1_IG

MsgInterchangeResponset - EDS.USGS.MED.EDI.GenericEligibilityVerification270_271Messaging.EligibilityVerificationRequestResponseType


ScopeOrchestration
Receive_271ICR...
Read_Config And...
ScopeEligibilityVerification
RECEIVED_BY_...
ConstructMessage_271
TransformICRes...
PopulateMsgEdif...


Send_271XML
PROCESSED_B...
DebugWrite
GenericExceptionBlock
WMI Audit Call
GenericExcep_NotificationScope
Handle Custom...
StandardExceptionBlock


WMI Audit Call
StandardExcep_NotificationScope
Handle Custom...
GlobalException
GenericExcep_OrchScope
Log Exception

13.7 Pipeline Configuration

13.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

13.7.1.1 Pipeline Process Flow

15. Decode

16. Disassemble

    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

17. Validate

18. ResolveParty

    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

13.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

13.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

13.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

13.7.2.1 Pipeline Process Flow

19. Pre-Assemble

20. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

21. Encode

13.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

13.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

14 Appendix L – Biz Talk Documentation – 276 Batch

14.1 Host Configuration

14.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

14.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

14.1.1.2 Receive locations running in this host

There are no receive locations running in this host

14.1.2 BizTalkServerProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

14.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType

14.1.2.2 Receive locations running in this host
There are no receive locations running in this host

14.1.3 BizTalkServerReceiveHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

14.1.3.1 Orchestrations running in this host
There are no orchestrations running in this host

14.1.3.2 Receive locations running in this host
RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

14.1.4 BizTalkServerSendHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

14.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

14.1.4.2 Receive locations running in this host

There are no receive locations running in this host

14.2 Send Port Configuration

14.2.1 SendICClaimStatusReq

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: False

14.2.1.1 Primary Transport

- **Property**
- **Value**
- Address
- \\10.40.14.111\fts\Generic\ClaimStatus276_277\Inbound276\InterChange\%SourceFileName%.xml
- Type
- FILE
- Ordered Delivery
- False
- Retry Count
- 3
- Retry Interval
- 5

14.2.1.2 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

14.2.1.3 Secondary Transport

- **Property**
- **Value**
- Address
- \\10.40.14.111\fts\Generic\ClaimStatus276_277\Inbound276\InterChange\%SourceFileName%.xml
- Type
- FILE
- Ordered Delivery
- False
- Retry Count
- 3
- Retry Interval
- 5

14.2.1.4 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

14.2.1.5 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

14.3 Receive Port Configuration

14.3.1 ReceiveClaimStatusReq

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

14.3.1.1 Receive Locations

1. RLClaimStatusReqGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\ClaimStatus276_277\Inbound276\XData*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

14.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

14.4 Protocol Configuration

14.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

14.4.1.1 Send ports using this protocol

SendICClaimStatusReq

14.4.1.2 Receive ports using this protocol

ReceiveClaimStatusReq

14.5 Assembly Configuration

14.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

14.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

14.5.2 EDS.USGS.MED.EDI.Edifecs276_277Schema

Name: EDS.USGS.MED.EDI.Edifecs276_277Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

14.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs276_277Schema._277_A1_IG

EDS.USGS.MED.EDI.Edifecs276_277Schema._276_A1_IG

14.5.3 EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Messaging

Name: EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

14.5.3.1 Maps contained in this assembly

GenericClaimStatusVerification276_277Messaging.ClaimStatusResponseXml_To_Edifecs277Xml

GenericClaimStatusVerification276_277Messaging.Edifecs276Xml_To_ClaimStatusVerificationRequestXml

GenericClaimStatusVerification276_277Messaging.Edifecs276Xml_To_Edifecs277Xml

14.5.3.2 Schema contained in this assembly

GenericClaimStatusVerification276_277Messaging.ClaimStatusVerificationRequestResponse

14.5.4 EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes

Name: EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

14.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

14.6 Orchestration Configuration

14.6.1 EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

14.6.1.1 Ports contained within this orchestration

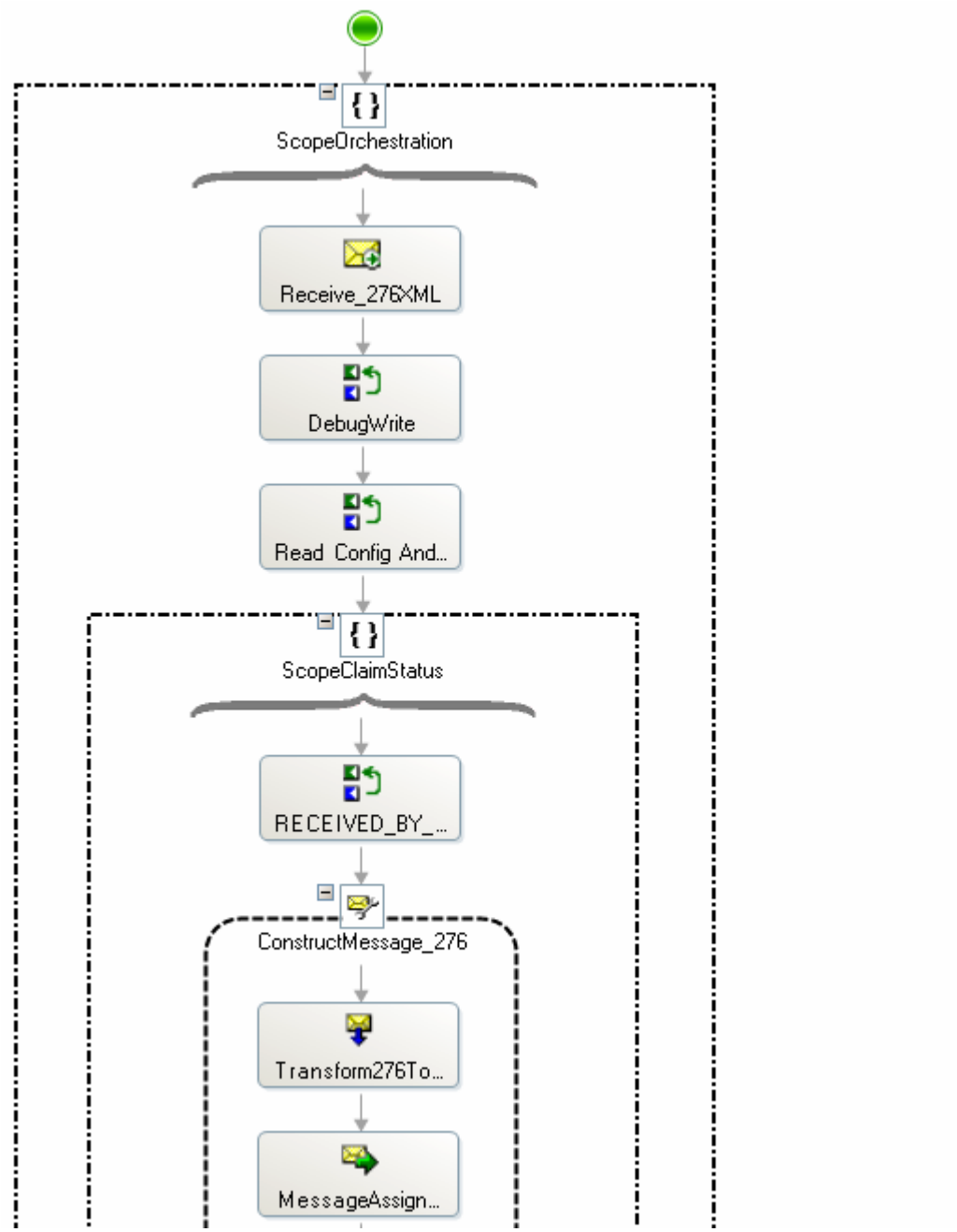
prtReceive276 - Bound to: ReceiveClaimStatusReq

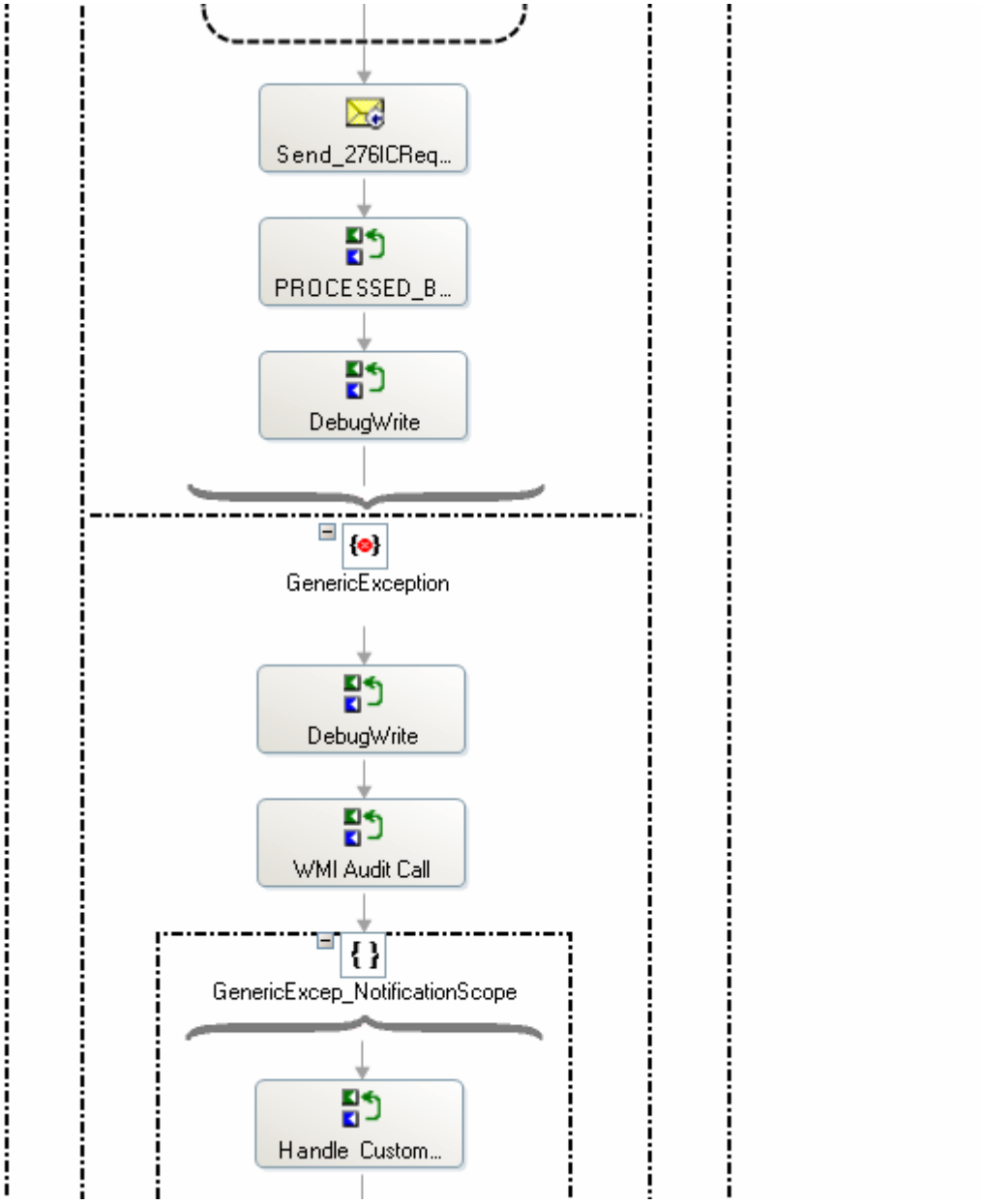
prtSendClaimStatusRequest - Bound to: SendICClaimStatusReq

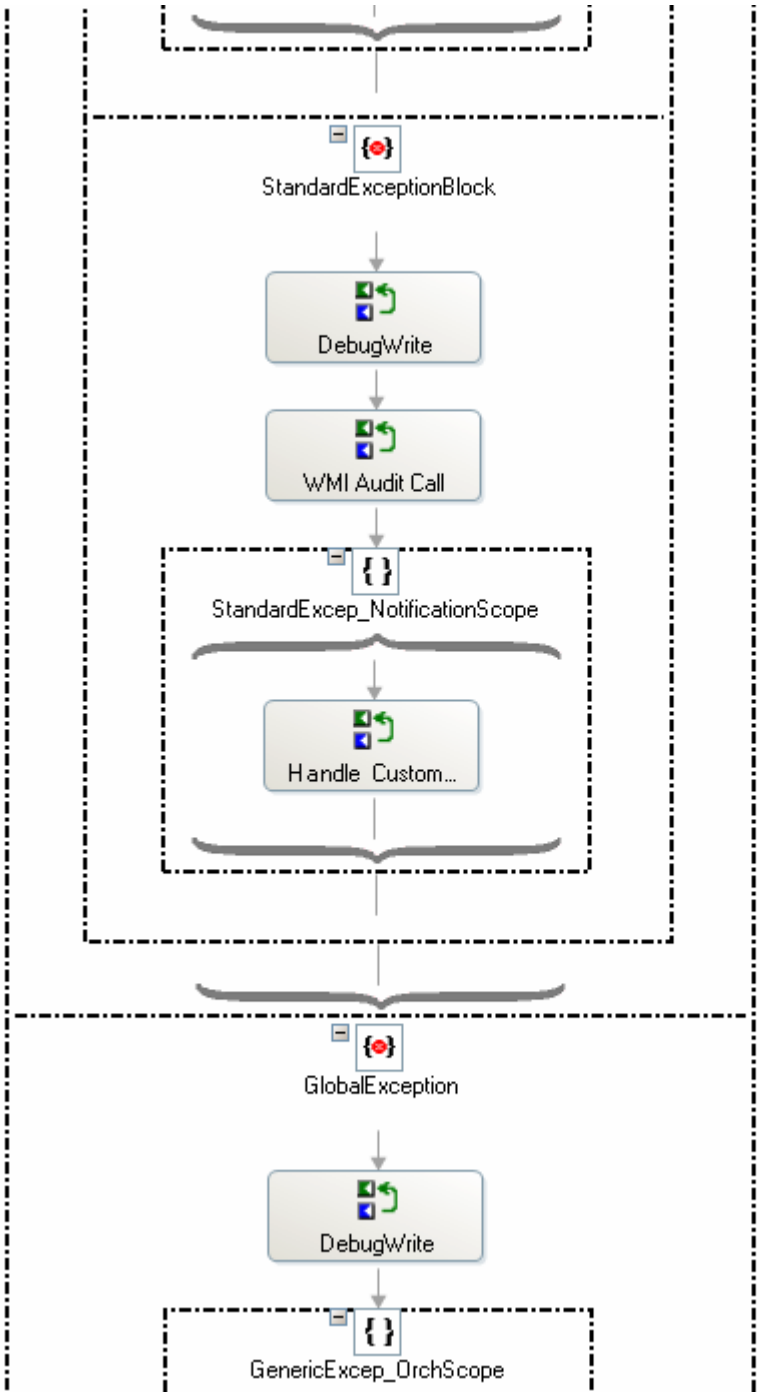
14.6.1.2 Messages defined within this orchestration

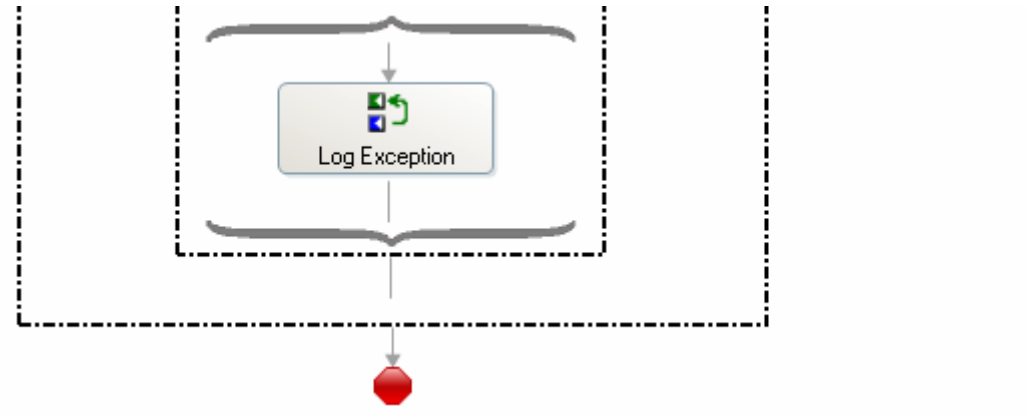
MsgInterchangeRequest - GenericClaimStatusVerification276_277Messaging.ClaimStatusVerificationRequestResponse

MsgEdifecs276 - EDS.USGS.MED.EDI.Edifecs276_277Schema._276_A1_IG


ScopeOrchestration
Receive_276XML
DebugWrite
Read Config And...
ScopeClaimStatus
RECEIVED_BY_...
ConstructMessage_276
Transform276To...
MessageAssign...


Send_276ICReq...
PROCESSED_B...
DebugWrite
GenericException
DebugWrite
WMI Audit Call
GenericExcep_NotificationScope
Handle_Custom...


StandardExceptionBlock
DebugWrite
WMI Audit Call
StandardExcep_NotificationScope
Handle Custom...
GlobalException
DebugWrite
GenericExcep_OrchScope



14.7 Pipeline Configuration

14.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

14.7.1.1 Pipeline Process Flow

22. Decode

23. Disassemble

    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

24. Validate

25. ResolveParty

    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

14.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

14.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

14.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

14.7.2.1 Pipeline Process Flow

26. Pre-Assemble
27. Assemble
    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
28. Encode

14.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

14.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

15 Appendix M – Biz Talk Documentation – 277 Batch

15.1 Host Configuration

15.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

15.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

15.1.1.2 Receive locations running in this host

There are no receive locations running in this host

15.1.2 BizTalkServerProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

15.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType

15.1.2.2 Receive locations running in this host
There are no receive locations running in this host

15.1.3 BizTalkServerReceiveHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

15.1.3.1 Orchestrations running in this host
There are no orchestrations running in this host

15.1.3.2 Receive locations running in this host
RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

15.1.4 BizTalkServerSendHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

15.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

15.1.4.2 Receive locations running in this host

There are no receive locations running in this host

15.2 Send Port Configuration

15.2.1 SendXDATA277

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: False

15.2.1.1 Primary Transport

Property	Value
Address	\\10.40.14.111\fts\Generic\ClaimStatus276_277\Outbound277\XData\%SourceFileName%.xml
Type	FILE
Ordered Delivery	False
Retry Count	3
Retry Interval	5

15.2.1.2 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

15.2.1.3 Secondary Transport

Property	Value
Address	\\10.40.14.111\fts\Generic\ClaimStatus276_277\Outbound277\XData\%SourceFileName%.xml
Type	FILE
Ordered Delivery	False
Retry Count	3
Retry Interval	5

15.2.1.4 Service Window

Enabled: False

Start Date: 1/2/2007 12:00:00 AM

End Date: 1/1/2007 11:59:59 PM

15.2.1.5 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

15.3 Receive Port Configuration

15.3.1 ReceiveICClaimStatusResp

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

15.3.1.1 Receive Locations

29. RLICClaimStatusRespGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\ClaimStatus276_277\Outbound277\Preprocessor*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

15.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

15.4 Protocol Configuration

15.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

15.4.1.1 Send ports using this protocol

SendXDATA277

15.4.1.2 Receive ports using this protocol

ReceiveICClaimStatusResp

15.5 Assembly Configuration

15.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

15.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

15.5.2 EDS.USGS.MED.EDI.Edifecs276_277Schema

Name: EDS.USGS.MED.EDI.Edifecs276_277Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

15.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs276_277Schema._277_A1_IG

EDS.USGS.MED.EDI.Edifecs276_277Schema._276_A1_IG

15.5.3 EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Messaging

Name: EDS.USGS.MED.EDI.GenericClaimStatusVerification276_277Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

15.5.3.1 Maps contained in this assembly

GenericClaimStatusVerification276_277Messaging.ClaimStatusResponseXml_To_Edifecs277Xml

GenericClaimStatusVerification276_277Messaging.Edifecs276Xml_To_ClaimStatusVerificationRequestXml

GenericClaimStatusVerification276_277Messaging.Edifecs276Xml_To_Edifecs277Xml

15.5.3.2 Schema contained in this assembly

GenericClaimStatusVerification276_277Messaging.ClaimStatusVerificationRequestResponse

15.5.4 EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes

Name: EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

15.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponse Process

15.6 Orchestration Configuration

15.6.1 EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

15.6.1.1 Ports contained within this orchestration

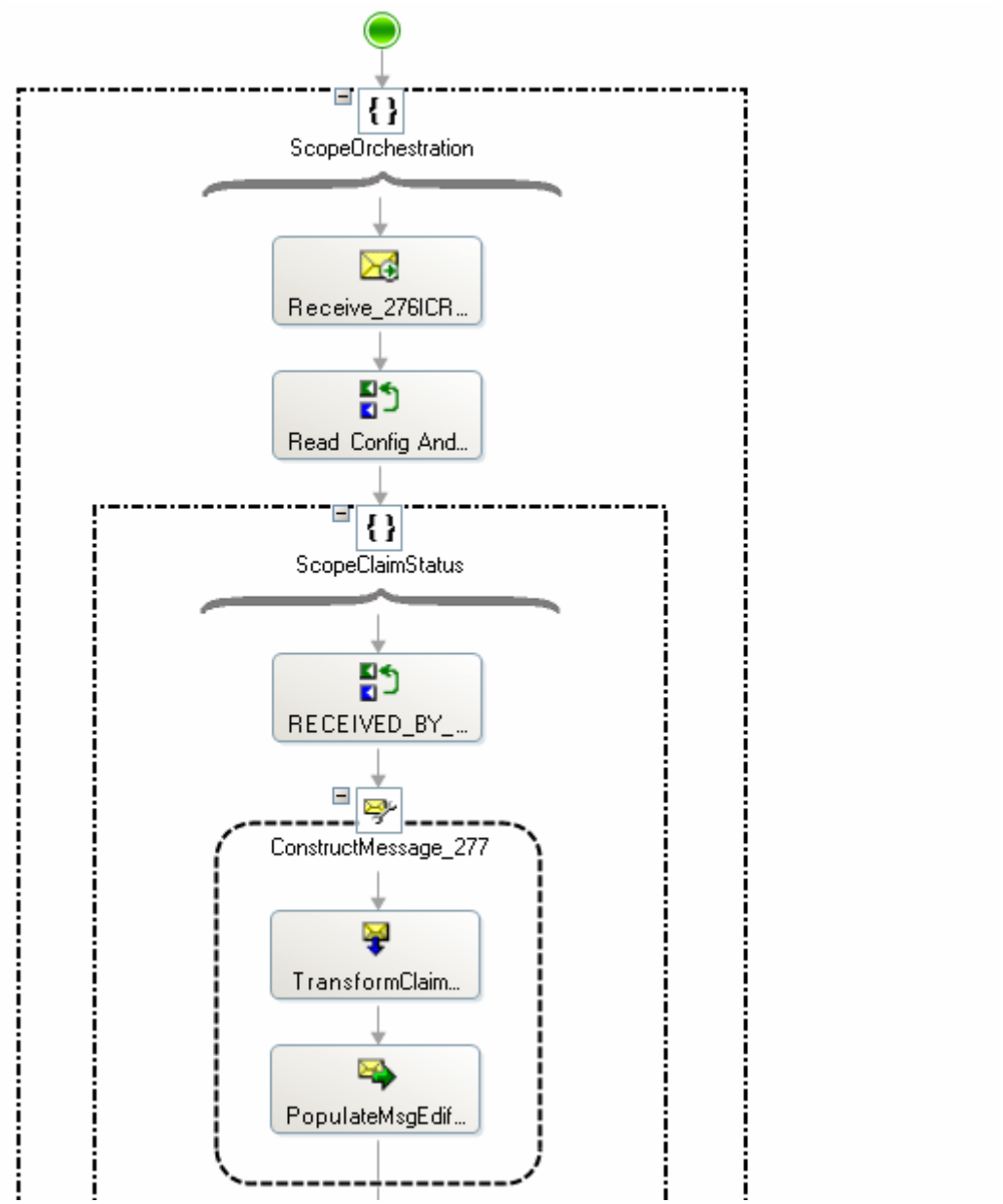
prtReceiveClaimStatusResponse - Bound to: ReceiveICClaimStatusResp

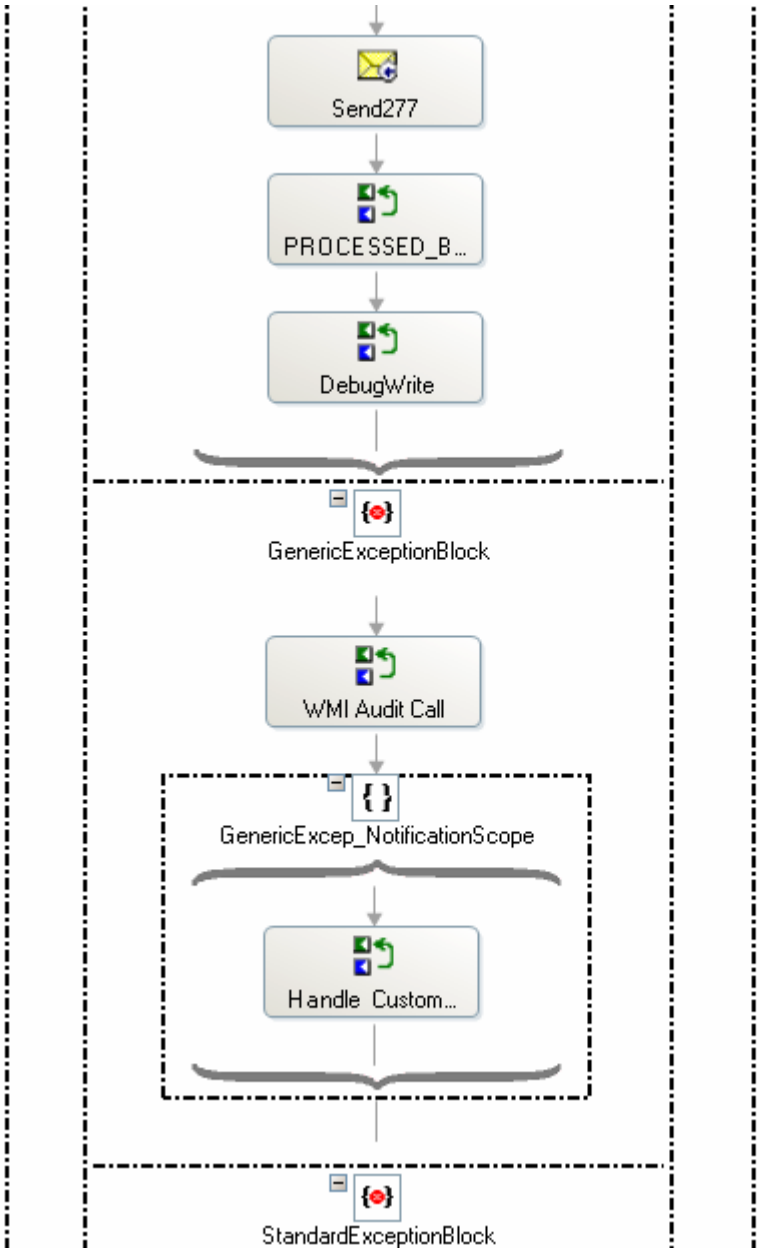
prtSend277 - Bound to: SendXDATA277

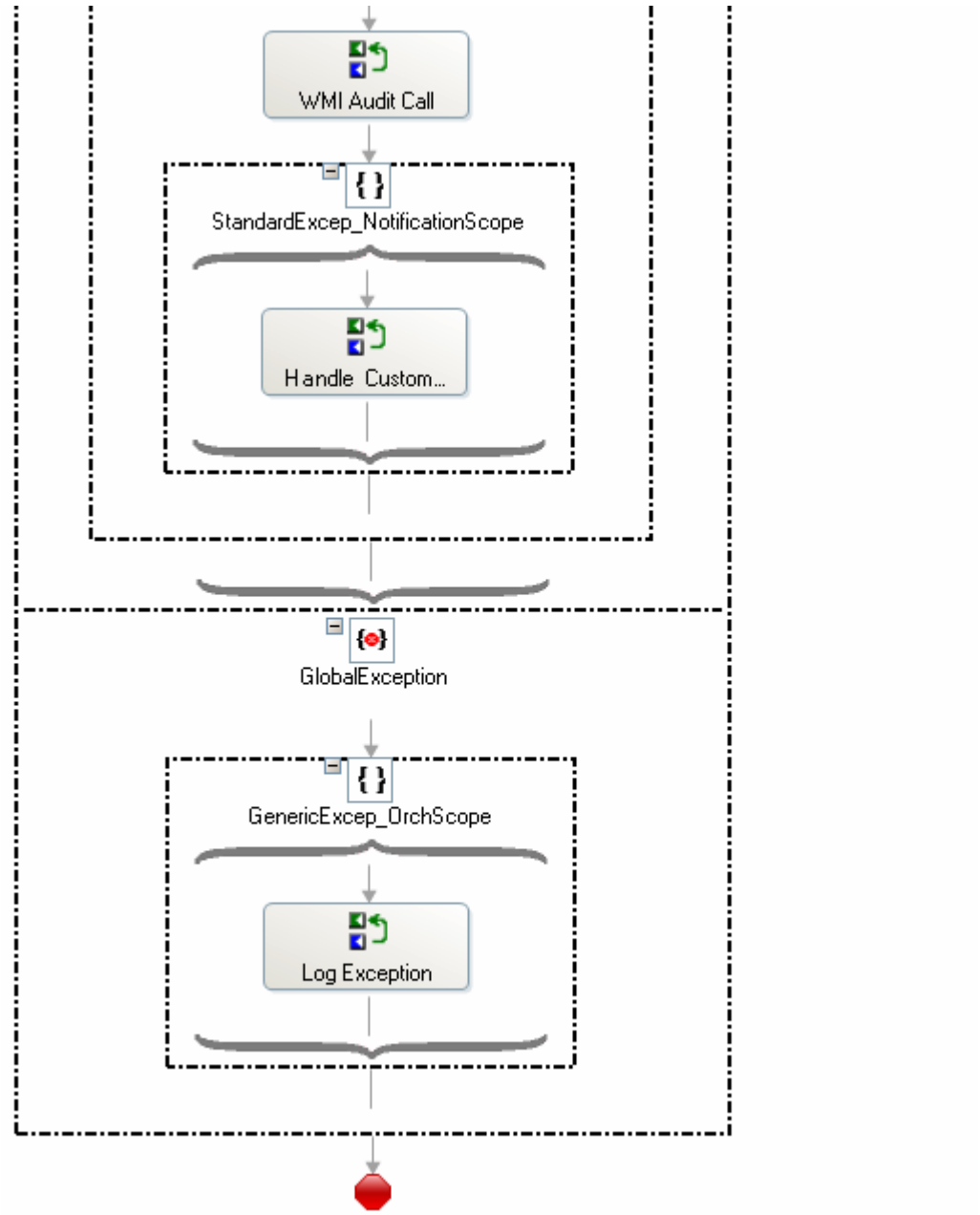
15.6.1.2 Messages defined within this orchestration

MsgInterchangeResponse - GenericClaimStatusVerification276_277Messaging.ClaimStatusVerificationRequestResponse

MsgEdifecs277 - EDS.USGS.MED.EDI.Edifecs276_277Schema._277_A1_IG


ScopeOrchestration
Receive_276ICR...
Read Config And...
ScopeClaimStatus
RECEIVED_BY_...
ConstructMessage_277
TransformClaim...
PopulateMsgEdif...


Send277
PROCESSED_B...
DebugWrite
GenericExceptionBlock
WMI Audit Call
GenericExcep_NotificationScope
Handle Custom...
StandardExceptionBlock


WMI Audit Call
StandardExcep_NotificationScope
Handle Custom...
GlobalException
GenericExcep_OrchScope
Log Exception

15.7 Pipeline Configuration

15.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

15.7.1.1 Pipeline Process Flow

30. Decode

31. Disassemble

    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

32. Validate

    - ResolveParty
    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

15.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

15.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

15.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

15.7.2.1 Pipeline Process Flow

33. Pre-Assemble

34. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

35. Encode

15.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

15.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

16 Appendix N – Biz Talk Documentation – 277U Outbound

16.1 Host Configuration

16.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

16.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

16.1.1.2 Receive locations running in this host

There are no receive locations running in this host

16.1.2 BizTalkServer835277UOutboundProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

16.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess

16.1.2.2 Receive locations running in this host

There are no receive locations running in this host

16.1.3 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

16.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

16.1.3.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

16.1.4 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

16.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

16.1.4.2 Receive locations running in this host

There are no receive locations running in this host

16.2 Send Port Configuration

16.2.1 SendXDATA277U

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

16.2.1.1 Primary Transport

Not yet specified

16.2.1.2 Secondary Transport

Not yet specified

16.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess

16.3 Receive Port Configuration

16.3.1 ReceiveRemittaceAdvice277U

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

16.3.1.1 Receive Locations

1. RLRemittaceAdvice277UGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\RemittanceAdvice277U\Outbound277U\Preprocessor*.*
Transport Protocol	FILE
Receive Pipeline	GenericOutboundRemittanceAdvice277UMessaging.RcvRemittanceAdvice277UPipeLine
Receive Handler	BizTalkServerReceiveHost

2. RLRemittaceAdvice277UPassPort

Property	Value
Address	\\10.40.13.12\fts_prod\fts\Passport\RemittanceAdvice277U\Outbound277U\Preprocessor*.*
Transport Protocol	FILE
Receive Pipeline	GenericOutboundRemittanceAdvice277UMessaging.RcvRemittanceAdvice277UPipeLine
Receive Handler	BizTalkServerReceiveHost

16.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess

16.4 Protocol Configuration

16.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

16.4.1.1 Send ports using this protocol

There are no send ports using this protocol

16.4.1.2 Receive ports using this protocol

ReceiveRemittaceAdvice277U

16.5 Assembly Configuration

16.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

16.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

16.5.2 EDS.USGS.MED.EDI.Edifecs277USchema

Name: EDS.USGS.MED.EDI.Edifecs277USchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

16.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs277USchema._277_v3070_IG

16.5.3 EDS.USGS.MED.EDI.GenericOutboundRemittanceAdvice277UMessaging

Name: EDS.USGS.MED.EDI.GenericOutboundRemittanceAdvice277UMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

16.5.3.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundRemittanceAdvice277UMessaging.OutboundRemittaceAdvice277U_To_Edifecs277UXml

16.5.3.2 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundRemittanceAdvice277UMessaging.Generic277URemittanceAdvice

16.5.4 EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses

Name: EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

16.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess

16.6 Orchestration Configuration

16.6.1 EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServer835277UOutboundProcessHost

16.6.1.1 Ports contained within this orchestration

prtOutboundRemittanceAdvice277U - Bound to: ReceiveRemittaceAdvice277U

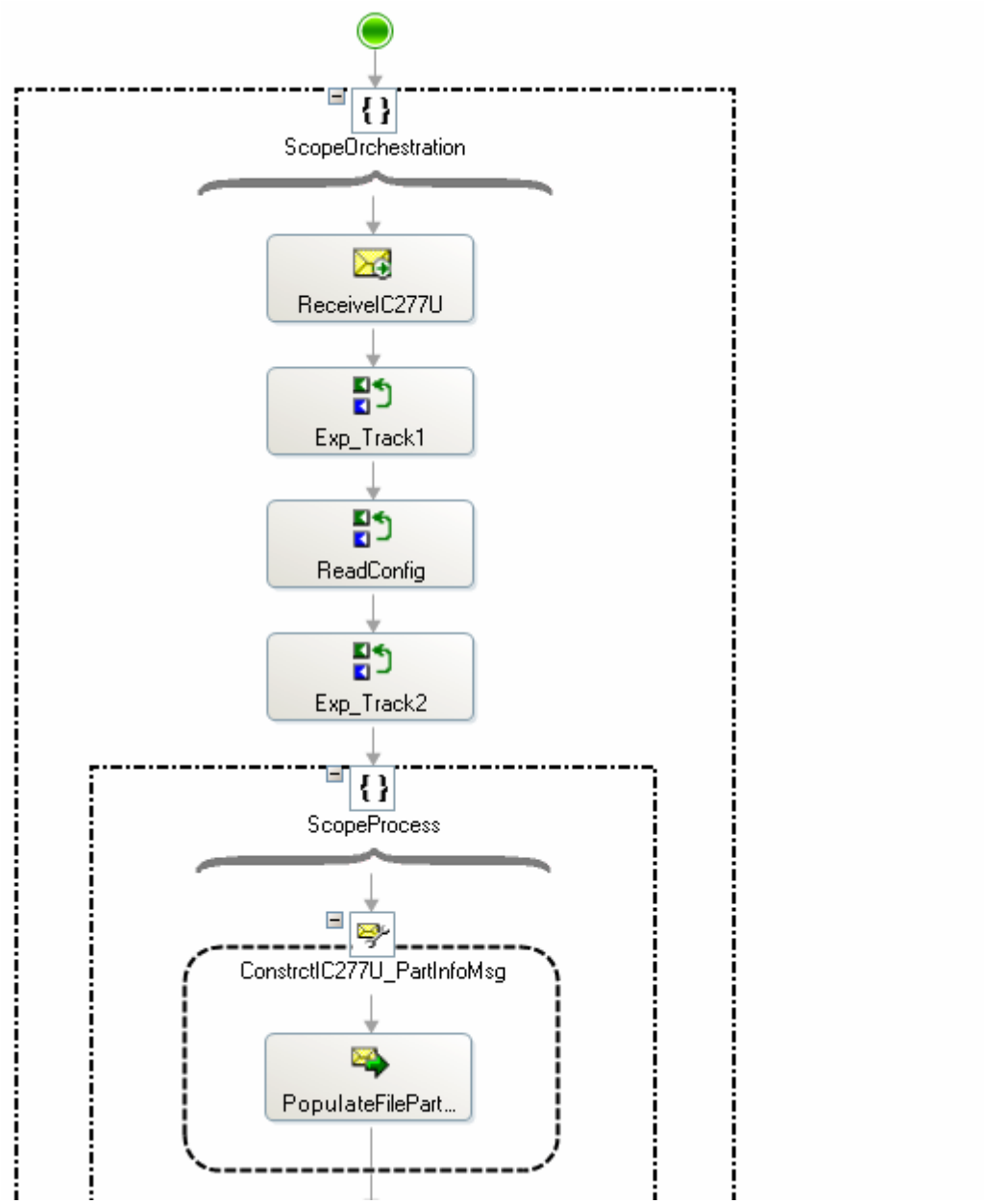
prtOutboundRemittanceAdvice - Bound to: SendXDATA277U

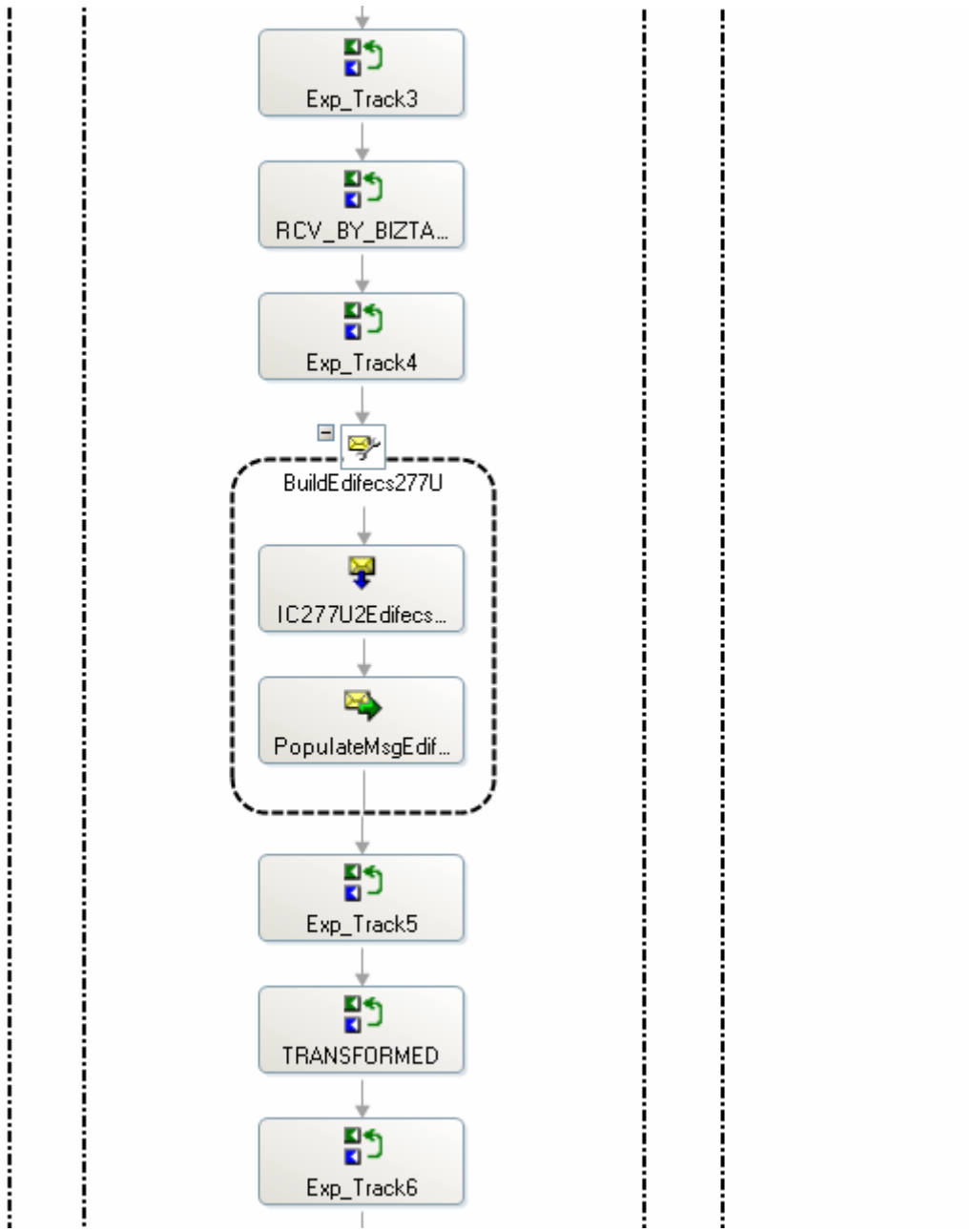
16.6.1.2 Messages defined within this orchestration

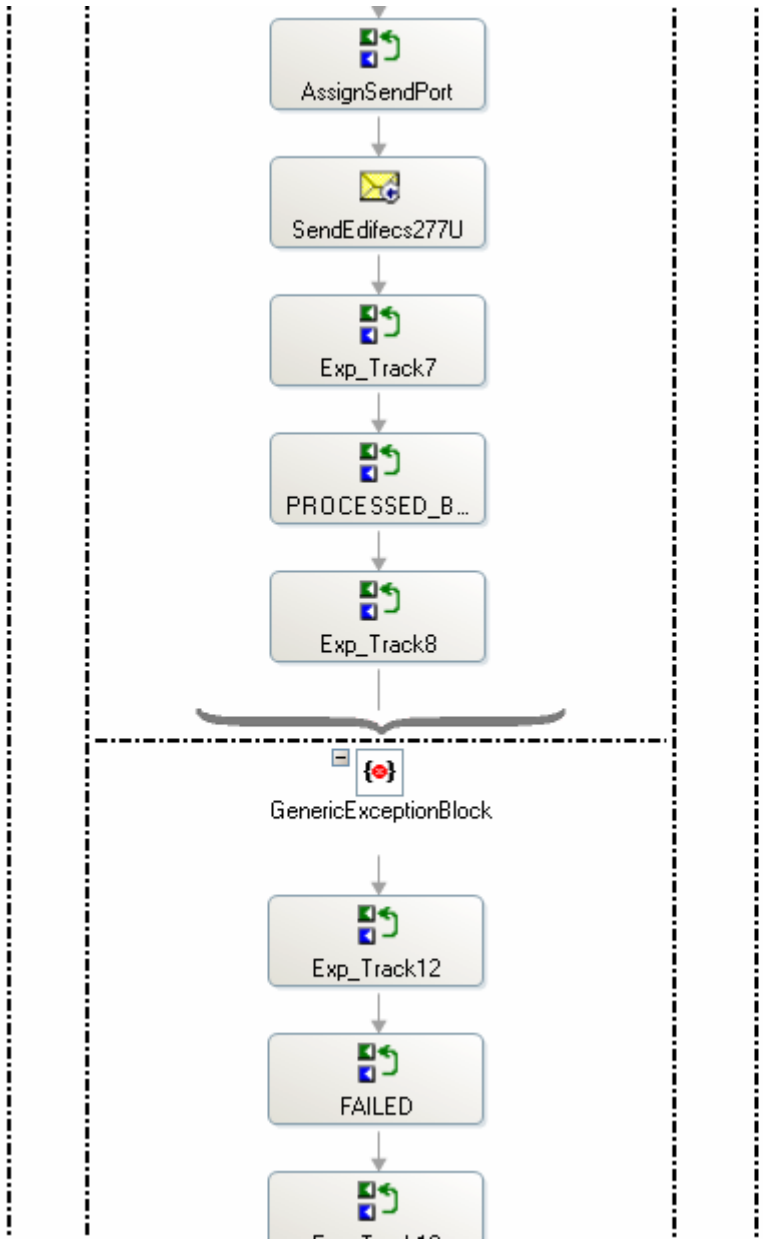
MessageIC277U_PartInfo - EDS.USGS.MED.EDI.GenericOutboundRemittanceAdvice277UMessaging.Generic277URemittanceAdvice

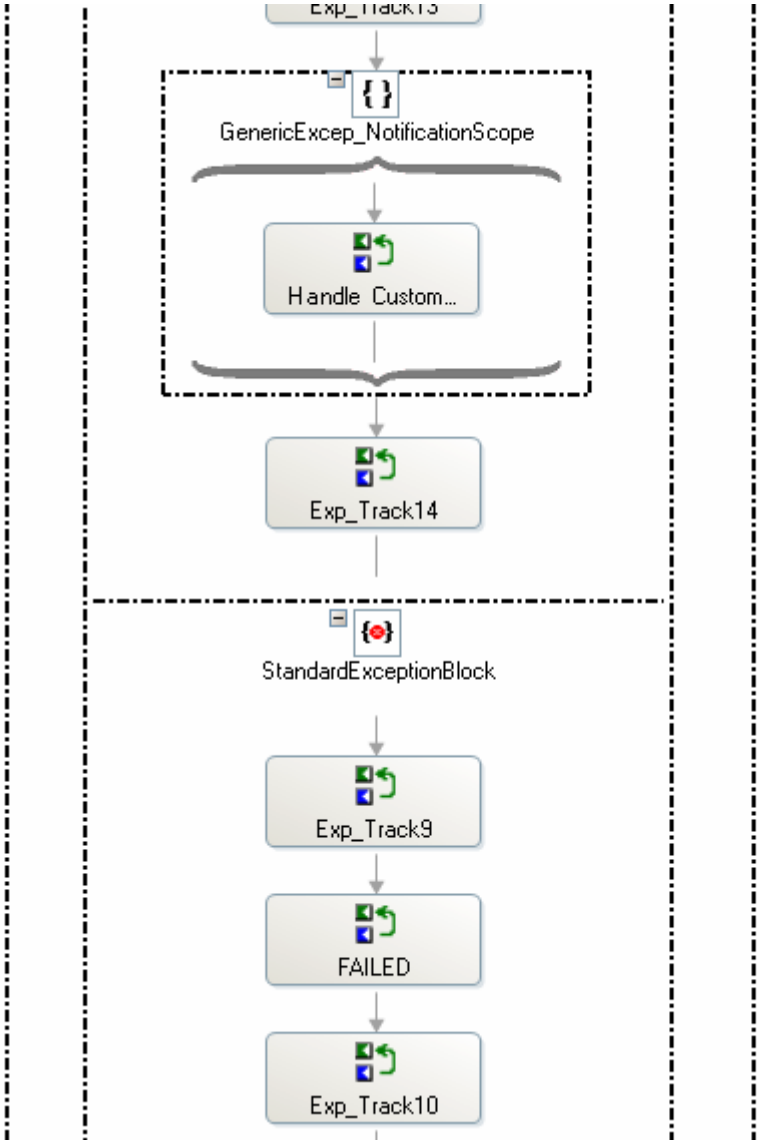
MsgIC277UFFOutbound- EDS.USGS.MED.EDI.GenericOutboundRemittanceAdvice277UMessaging.Generic277URemittanceAdvice

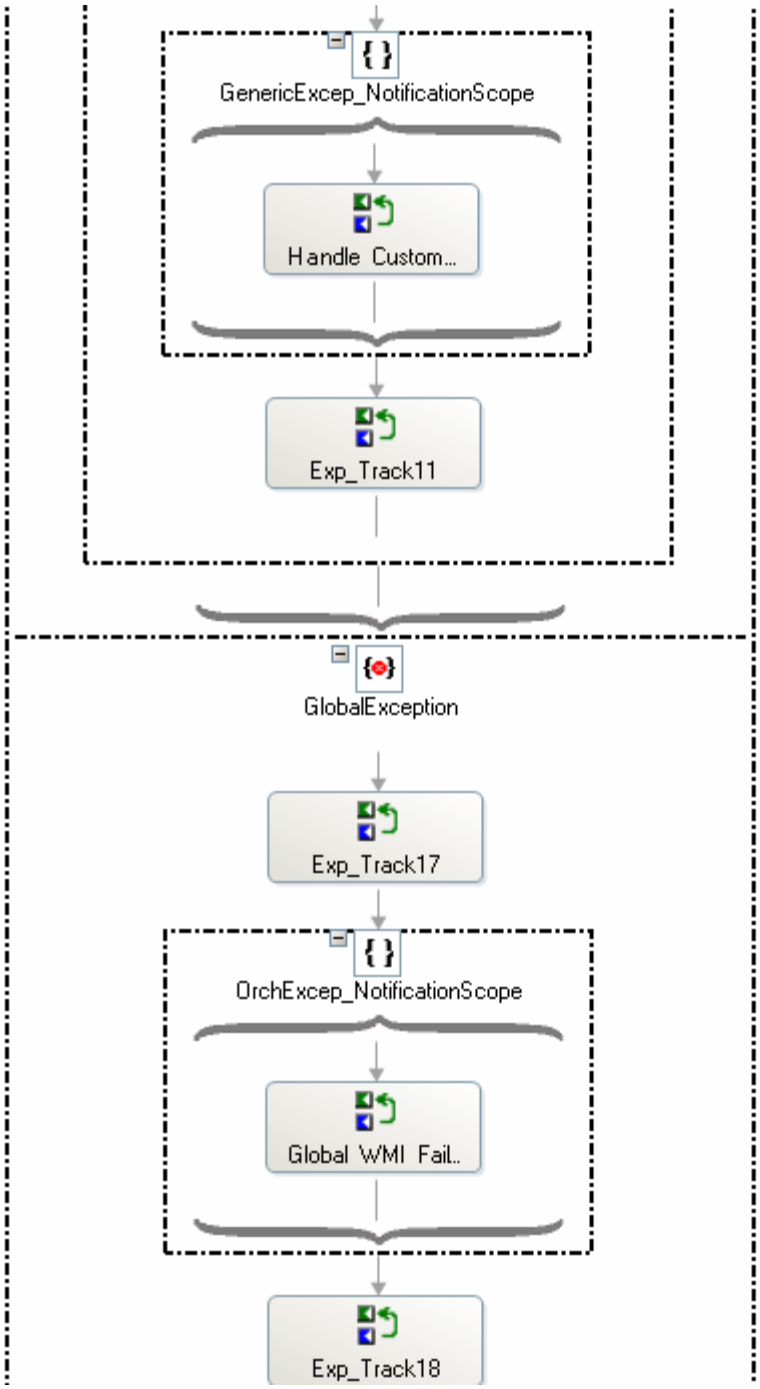
MsgEDIFECS277UXMLOutbound- EDS.USGS.MED.EDI.Edifecs277USchema._277_v3070_IG

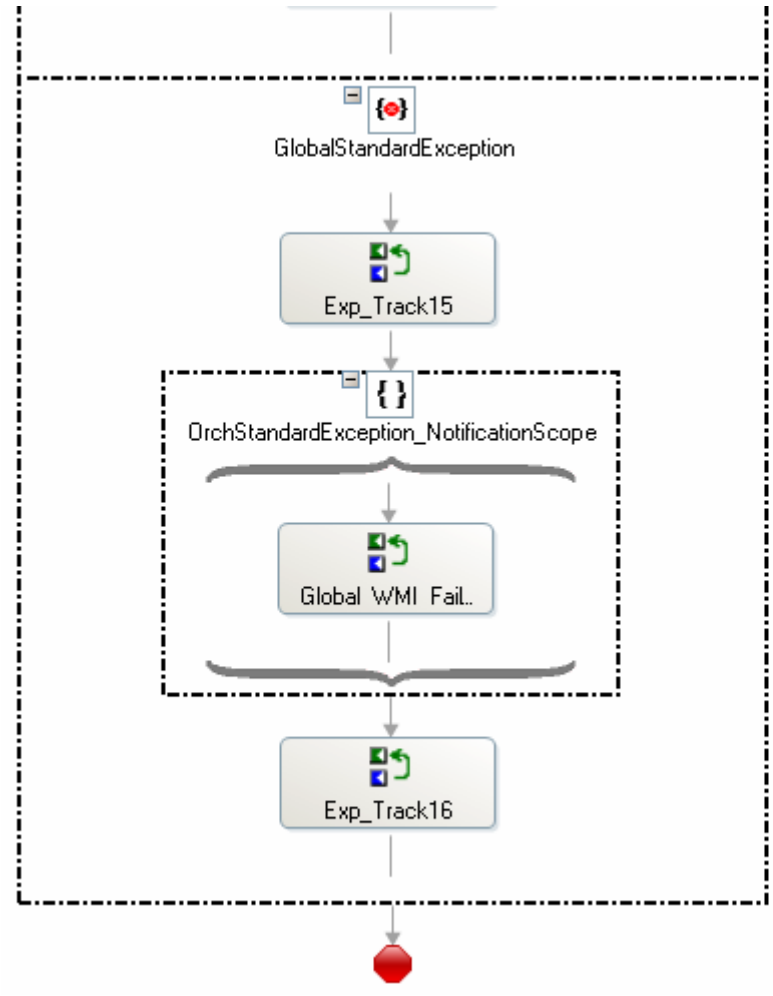

ScopeOrchestration
ReceiveIC277U
Exp_Track1
ReadConfig
Exp_Track2
ScopeProcess
ConstrctIC277U_PartInfoMsg
PopulateFilePart...


Exp_Track3
RCV_BY_BIZTA...
Exp_Track4
BuildEdifecs277U
IC277U2Edifecs...
PopulateMsgEdif...
Exp_Track5
TRANSFORMED
Exp_Track6


AssignSendPort
SendEdifecs277U
Exp_Track7
PROCESSED_B...
Exp_Track8
GenericExceptionBlock
Exp_Track12
FAILED


GenericExcep_NotificationScope
Handle Custom...
Exp_Track14
StandardExceptionBlock
Exp_Track9
FAILED
Exp_Track10


GenericExcep_NotificationScope
Handle Custom...
Exp_Track11
GlobalException
Exp_Track17
OrchExcep_NotificationScope
Global WMI Fail..
Exp_Track18


GlobalStandardException
Exp_Track15
OrchStandardException_NotificationScope
Global WMI Fail..
Exp_Track16

16.7 Pipeline Configuration

16.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

16.7.1.1 Pipeline Process Flow

36. Decode

37. Disassemble

    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

38. Validate

39. ResolveParty

    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

16.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

16.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

16.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

16.7.2.1 Pipeline Process Flow

40. Pre-Assemble

41. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

42. Encode

16.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

16.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

16.7.3 GenericOutboundRemittanceAdvice277UMessaging.RcvRemittanceAdvice277UPipeLine

Pipeline Type: Receive

Assembly Name: GenericOutboundRemittanceAdvice277UMessaging.RcvRemittanceAdvice277UPipeLine, EDS.USGS.MED.EDI.GenericOutboundRemittanceAdvice277UMessaging, Version=1.0.0.0, Culture=neutral, PublicKeyToken=a2fc6bba136a8ae3

16.7.3.1 Pipeline Process Flow

43. Decode

44. Disassemble

    - Flat file disassembler
    - Microsoft.BizTalk.Component.FFDasmComp,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

45. Validate

46. ResolveParty

16.7.3.2 Send ports using this pipeline

There are no send ports using this pipeline

16.7.3.3 Receive locations using this pipeline

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

17 Appendix O – Biz Talk Documentation – 820 Outbound

17.1 Host Configuration

17.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

17.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

17.1.1.2 Receive locations running in this host

There are no receive locations running in this host

17.1.2 BizTalkServer820ProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

17.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess

17.1.2.2 Receive locations running in this host

There are no receive locations running in this host

17.1.3 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

17.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

17.1.3.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

17.1.4 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

17.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

17.1.4.2 Receive locations running in this host

There are no receive locations running in this host

17.2 Send Port Configuration

17.2.1 SendXDATA820

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

17.2.1.1 Primary Transport

Not yet specified

17.2.1.2 Secondary Transport

Not yet specified

17.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess

17.3 Receive Port Configuration

17.3.1 ReceivePremiumPayment

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

17.3.1.1 Receive Locations

1. RLPremiumPayGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\PremiumPayment820\Outbound\Preprocessor*.*
Transport Protocol	FILE
Receive Pipeline	GenericOutboundPremiumPayment820Messaging.RcPremiumPaymentPipeline
Receive Handler	BizTalkServerReceiveHost

17.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess

17.4 Protocol Configuration

17.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

17.4.1.1 Send ports using this protocol

There are no send ports using this protocol

17.4.1.2 Receive ports using this protocol

ReceivePremiumPayment

17.5 Assembly Configuration

17.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

17.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

17.5.2 EDS.USGS.MED.EDI.Edifecs820Schema

Name: EDS.USGS.MED.EDI.Edifecs820Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

17.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs820Schema._820

17.5.3 EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Messaging

Name: EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

17.5.3.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Messaging.OutboundPremiumPayment_To_Edifecs820Xml

17.5.3.2 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Messaging.PremiumPayment

17.5.4 EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes

Name: EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

17.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess

17.6 Orchestration Configuration

17.6.1 EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServer820ProcessHost

17.6.1.1 Ports contained within this orchestration

prtReceivePremiumPayment - Bound to: ReceivePremiumPayment

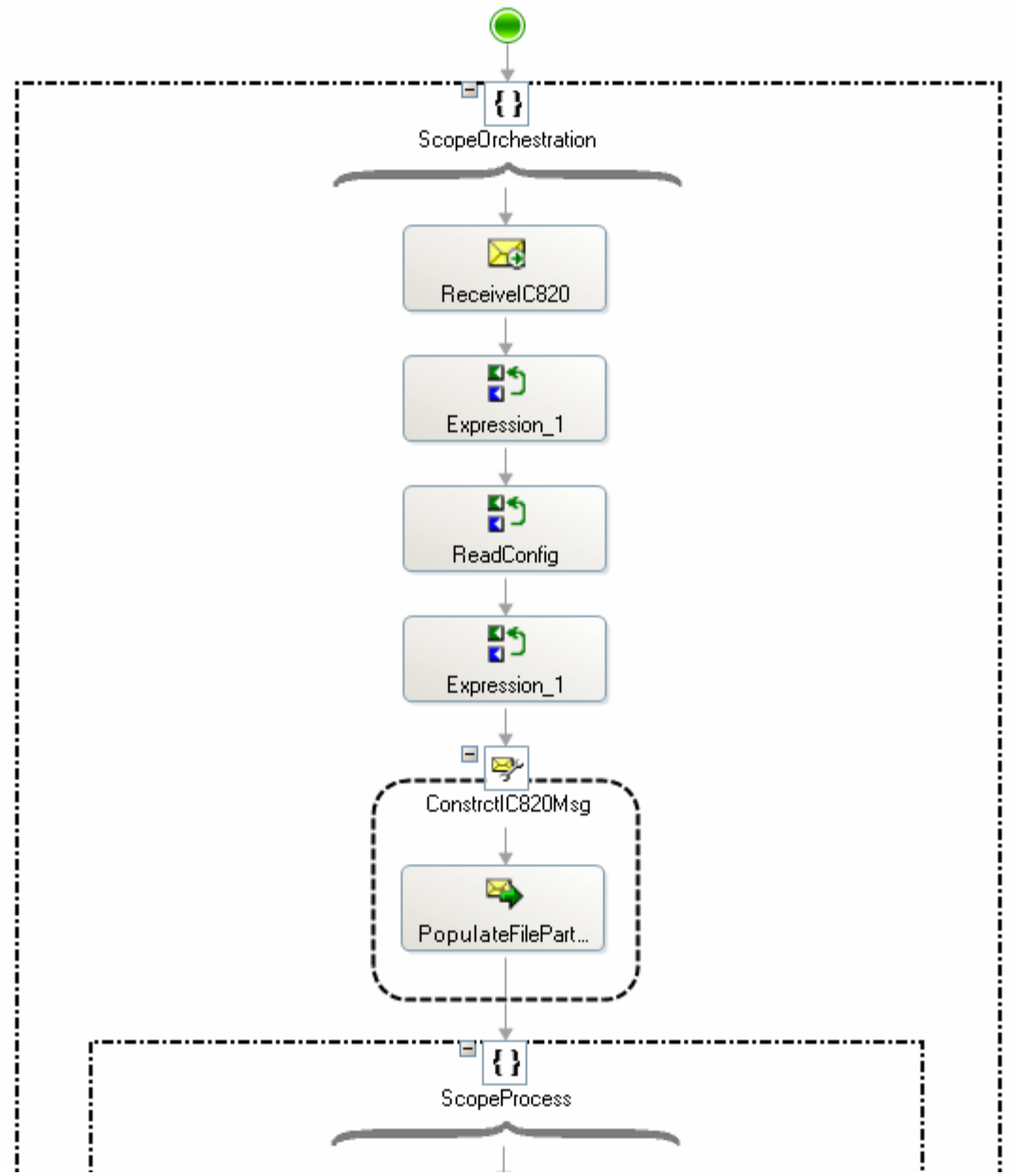
prtSendXML820 - Bound to: SendXDATA820

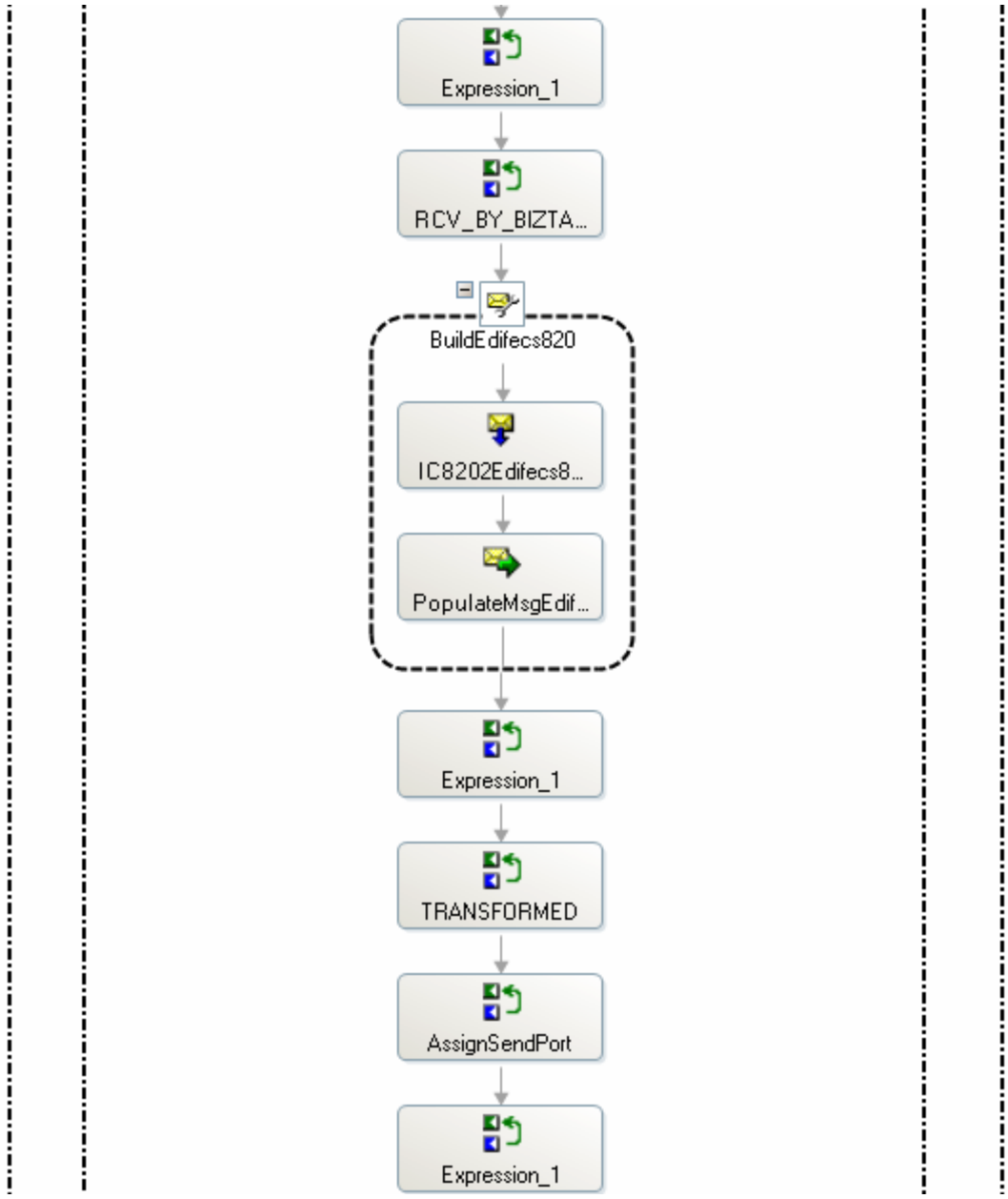
17.6.1.2 Messages defined within this orchestration

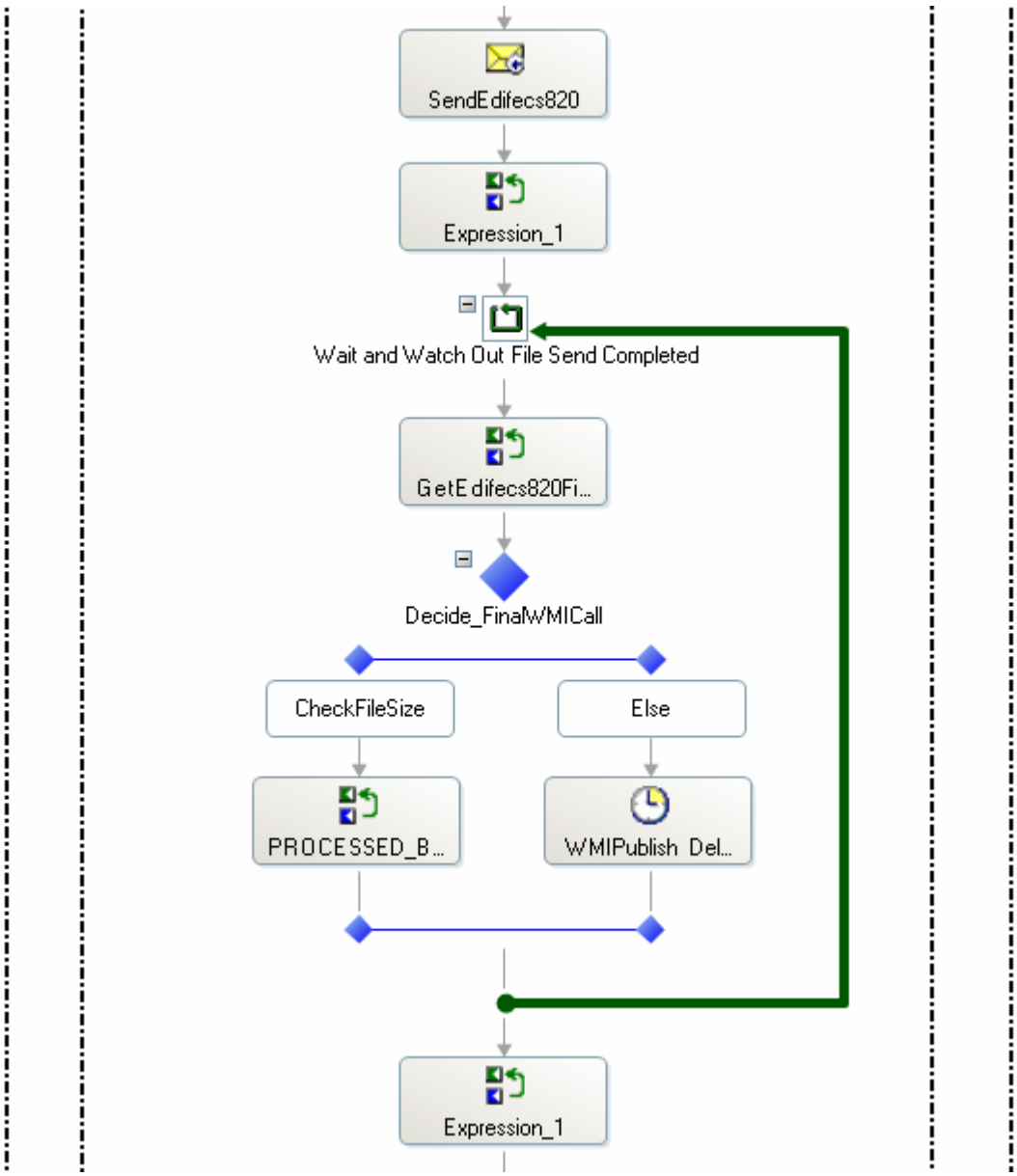
MessageIC820_PartInfo-
EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Messaging.PremiumPayment

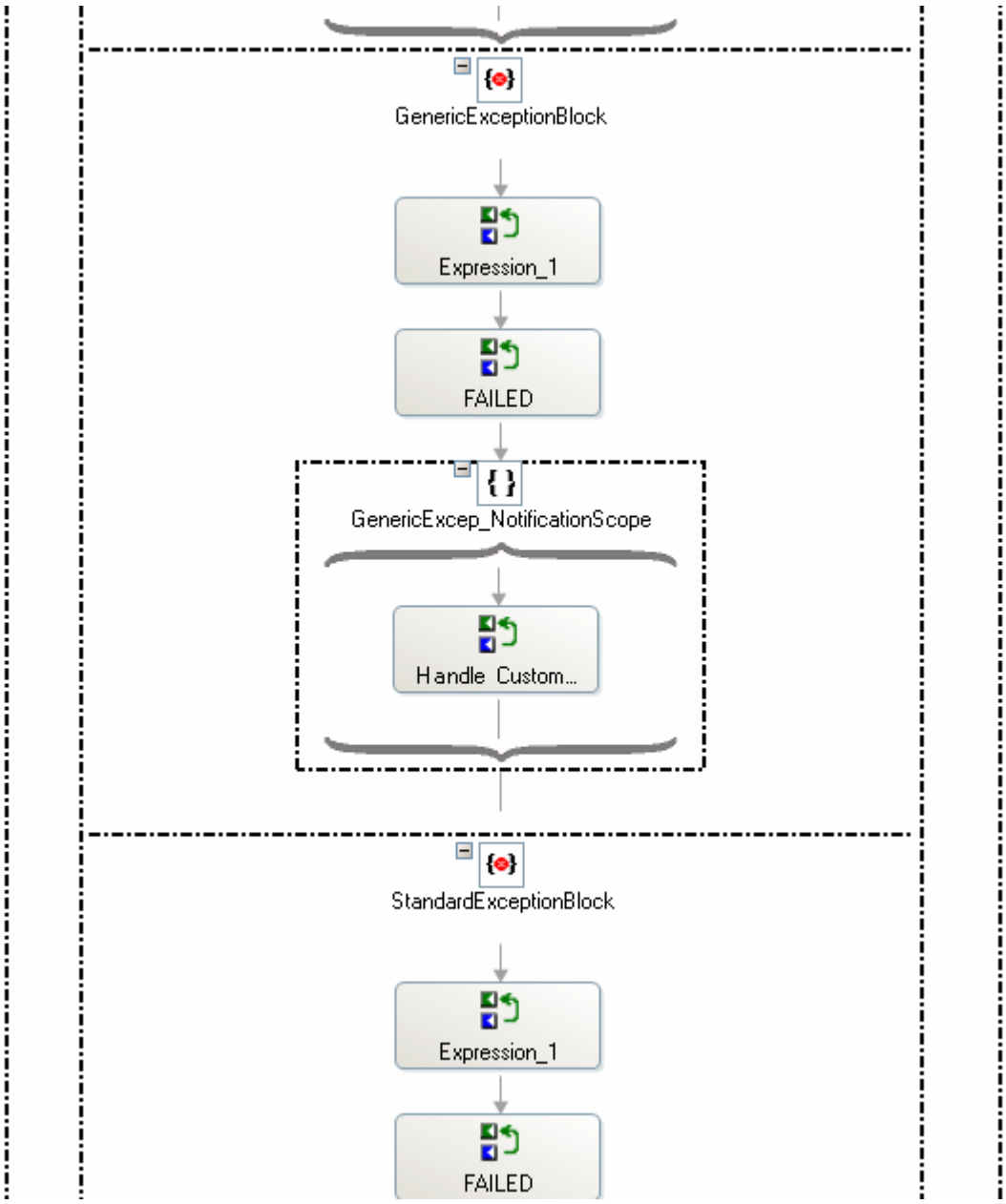
MessageEdifecs820- EDS.USGS.MED.EDI.Edifecs820Schema._820

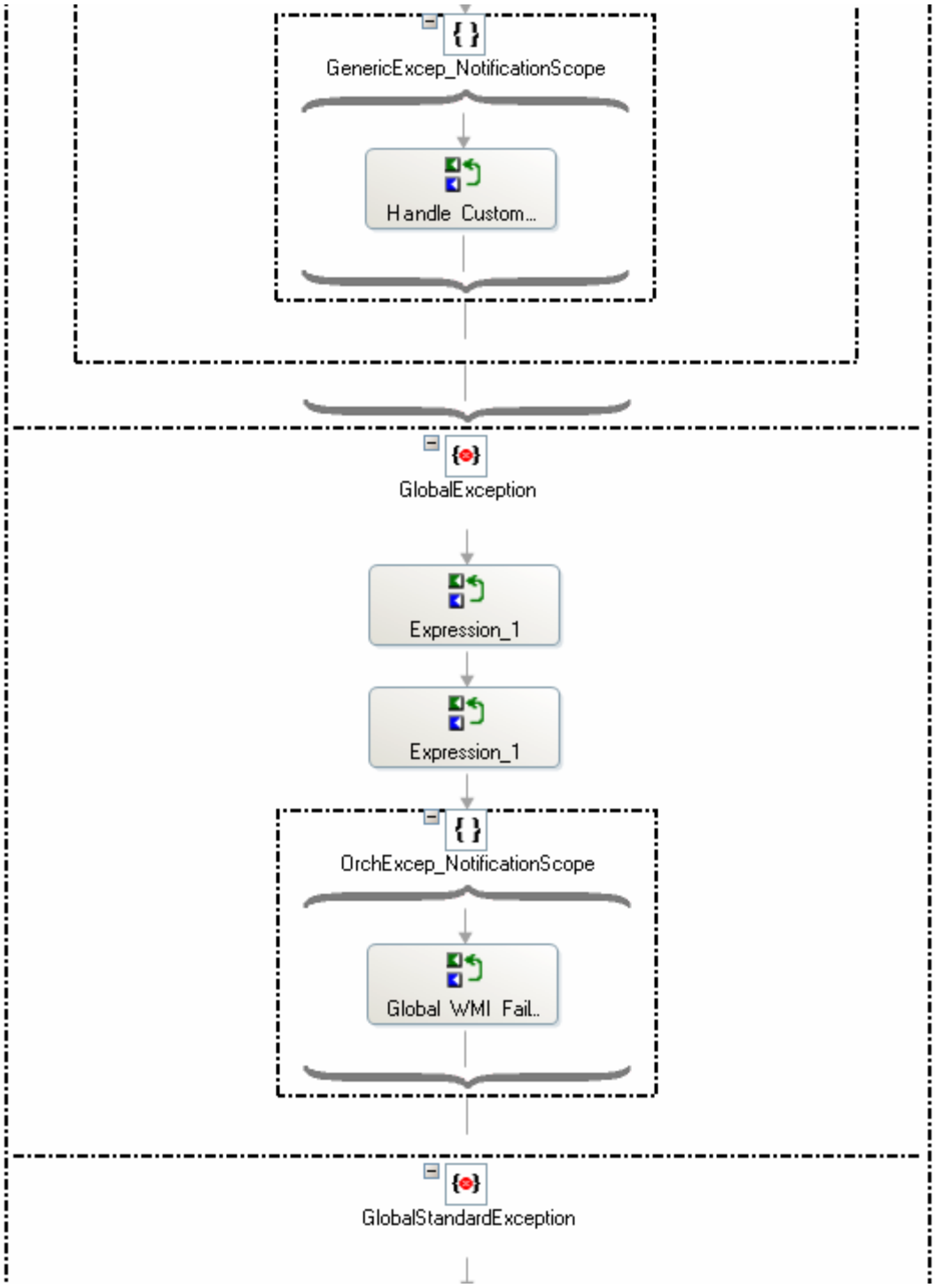
MessageIC820-
EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Messaging.PremiumPayment

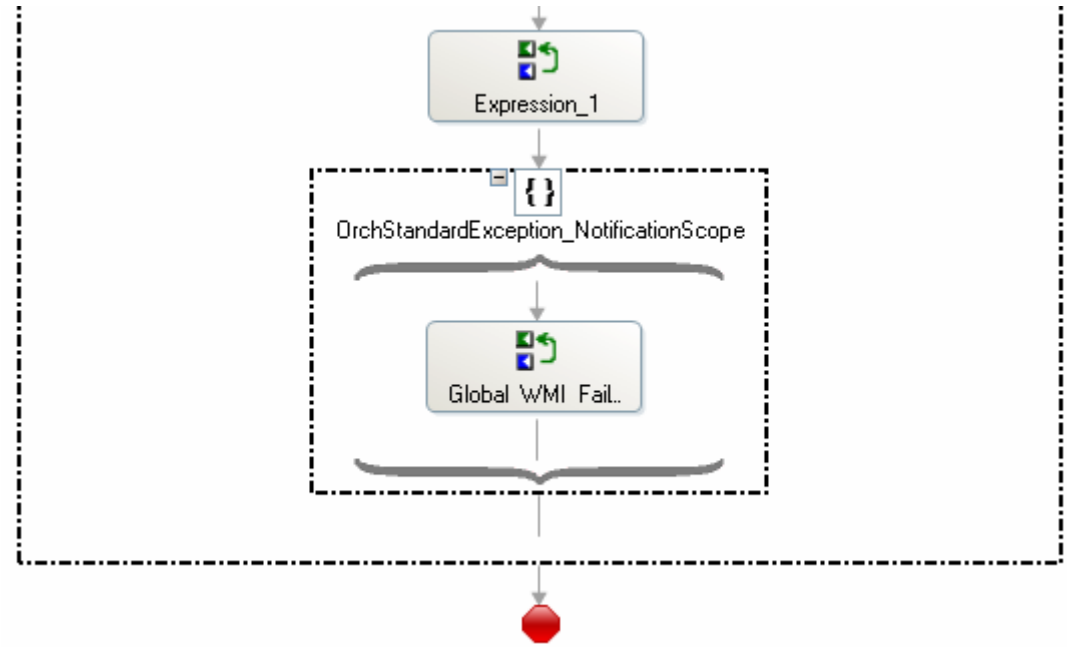

ScopeOrchestration
ReceiveIC820
Expression_1
ReadConfig
Expression_1
ConstrctIC820Msg
PopulateFilePart...
ScopeProcess


Expression_1
RCV_BY_BIZTA...
BuildEdifecs820
IC8202Edifecs8...
PopulateMsgEdif...
Expression_1
TRANSFORMED
AssignSendPort
Expression_1


SendEdifecs820
Expression_1
Wait and Watch Out File Send Completed
GetEdifecs820Fi...
Decide_FinalWMICall
CheckFileSize
Else
PROCESSED_B...
WMIPublish Del...
Expression_1


GenericExceptionBlock
Expression_1
FAILED
GenericExcep_NotificationScope
Handle Custom...
StandardExceptionBlock
Expression_1
FAILED


GenericExcep_NotificationScope
Handle Custom...
GlobalException
Expression_1
Expression_1
OrchExcep_NotificationScope
Global WMI Fail..
GlobalStandardException


Expression_1
OrchStandardException_NotificationScope
Global WMI Fail..

17.7 Pipeline Configuration

17.7.1 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

17.7.1.1 Pipeline Process Flow

47. Pre-Assemble

48. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

49. Encode

17.7.1.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

17.7.1.3 Receive locations using this pipeline

There are no receive locations using this pipeline

17.7.2 GenericOutboundPremiumPayment820Messaging.RcPremiumPaymentPipeline

Pipeline Type: Receive

Assembly Name:
GenericOutboundPremiumPayment820Messaging.RcPremiumPaymentPipeline, EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Messaging, Version=1.0.0.0, Culture=neutral, PublicKeyToken=a2fc6bba136a8ae3

17.7.2.1 Pipeline Process Flow

50. Decode

51. Disassemble

    - Flat file disassembler
    - Microsoft.BizTalk.Component.FFDasmComp,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

52. Validate

    - ResolveParty

17.7.2.2 Send ports using this pipeline

There are no send ports using this pipeline

17.7.2.3 Receive locations using this pipeline

RLPremiumPayGeneric

18 Appendix P – Biz Talk Documentation – 834 Outbound

18.1 Host Configuration

18.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

18.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

18.1.1.2 Receive locations running in this host

There are no receive locations running in this host

18.1.2 BizTalkServer834ProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

18.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess

18.1.2.2 Receive locations running in this host

There are no receive locations running in this host

18.1.3 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

18.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

18.1.3.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

18.1.4 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

18.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

18.1.4.2 Receive locations running in this host

There are no receive locations running in this host

18.2 Send Port Configuration

18.2.1 SendXDATA834

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

18.2.1.1 Primary Transport

Not yet specified

18.2.1.2 Secondary Transport

Not yet specified

18.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess

18.3 Receive Port Configuration

18.3.1 ReceiveBenefitEnrollment

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

18.3.1.1 Receive Locations

53. RLBenefitEnrollGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\BenefitEnrollment834\Outbound834\Preprocessor*.*
Transport Protocol	FILE
Receive Pipeline	GenericOutboundBenefitEnrollment834Messaging.RecvMemberEnrollmentPipeline
Receive Handler	BizTalkServerReceiveHost

18.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess

18.4 Protocol Configuration

18.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

18.4.1.1 Send ports using this protocol

There are no send ports using this protocol

18.4.1.2 Receive ports using this protocol

ReceiveBenefitEnrollment

18.5 Assembly Configuration

18.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

18.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

18.5.2 EDS.USGS.MED.EDI.Edifecs834Schema

Name: EDS.USGS.MED.EDI.Edifecs834Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

18.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs834Schema._834_A1_IG

18.5.3 EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Messaging

Name: EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

18.5.3.1 Maps contained in this assembly

GenericOutboundBenefitEnrollment834Messaging.OutboundBenefitEnrollment_To_Edifecs834Xml

18.5.3.2 Schema contained in this assembly

GenericOutboundBenefitEnrollment834Messaging.MemberEnrollments

18.5.4 EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes

Name: EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

18.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess

18.6 Orchestration Configuration

18.6.1 EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServer834ProcessHost

18.6.1.1 Ports contained within this orchestration

prtReceiveBenefitEnrollment - Bound to: ReceiveBenefitEnrollment

prtSendXML834 - Bound to: SendXDATA834

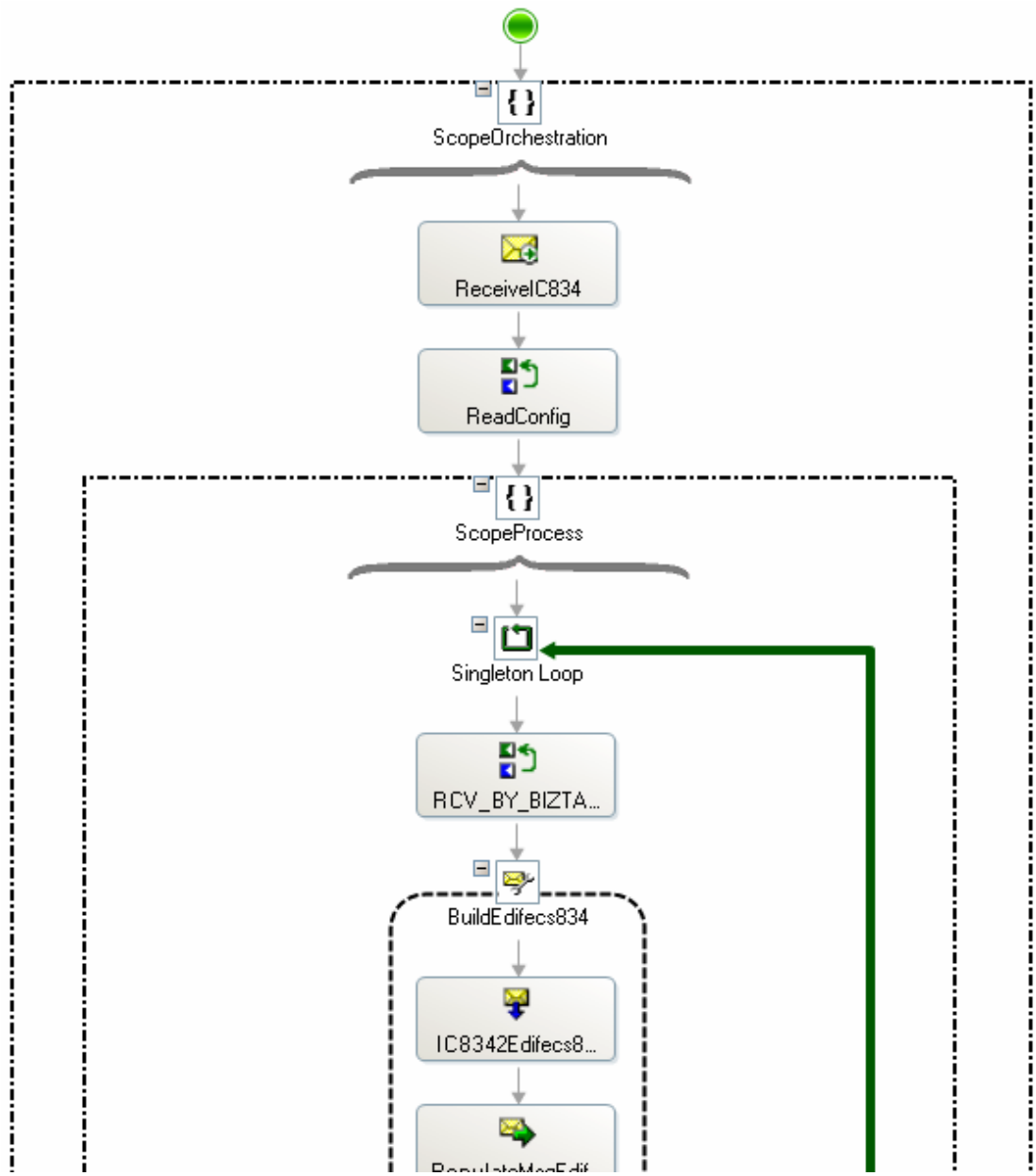
18.6.1.2 Messages defined within this orchestration

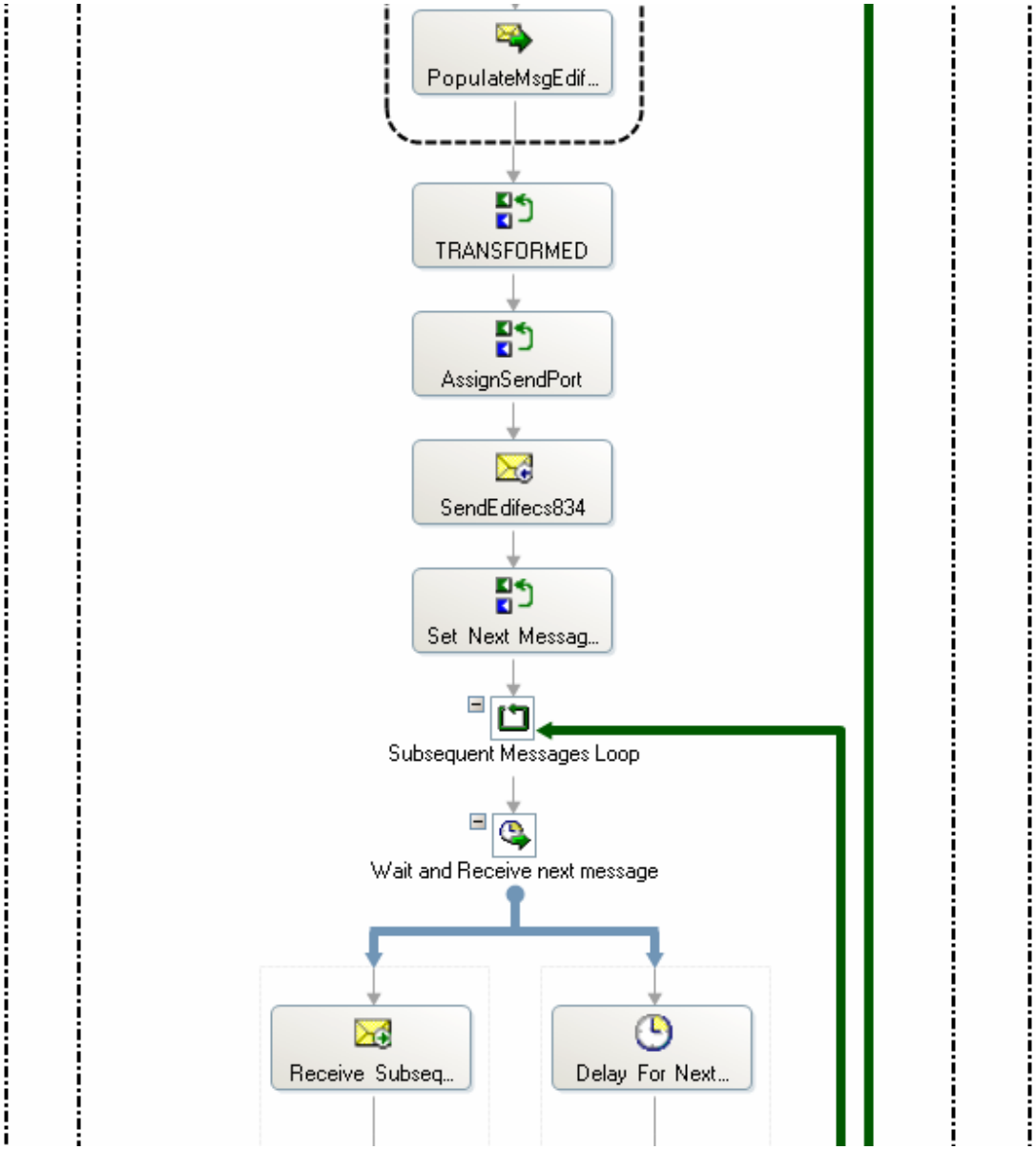
MessageIC834- GenericOutboundBenefitEnrollment834Messaging.MemberEnrollments

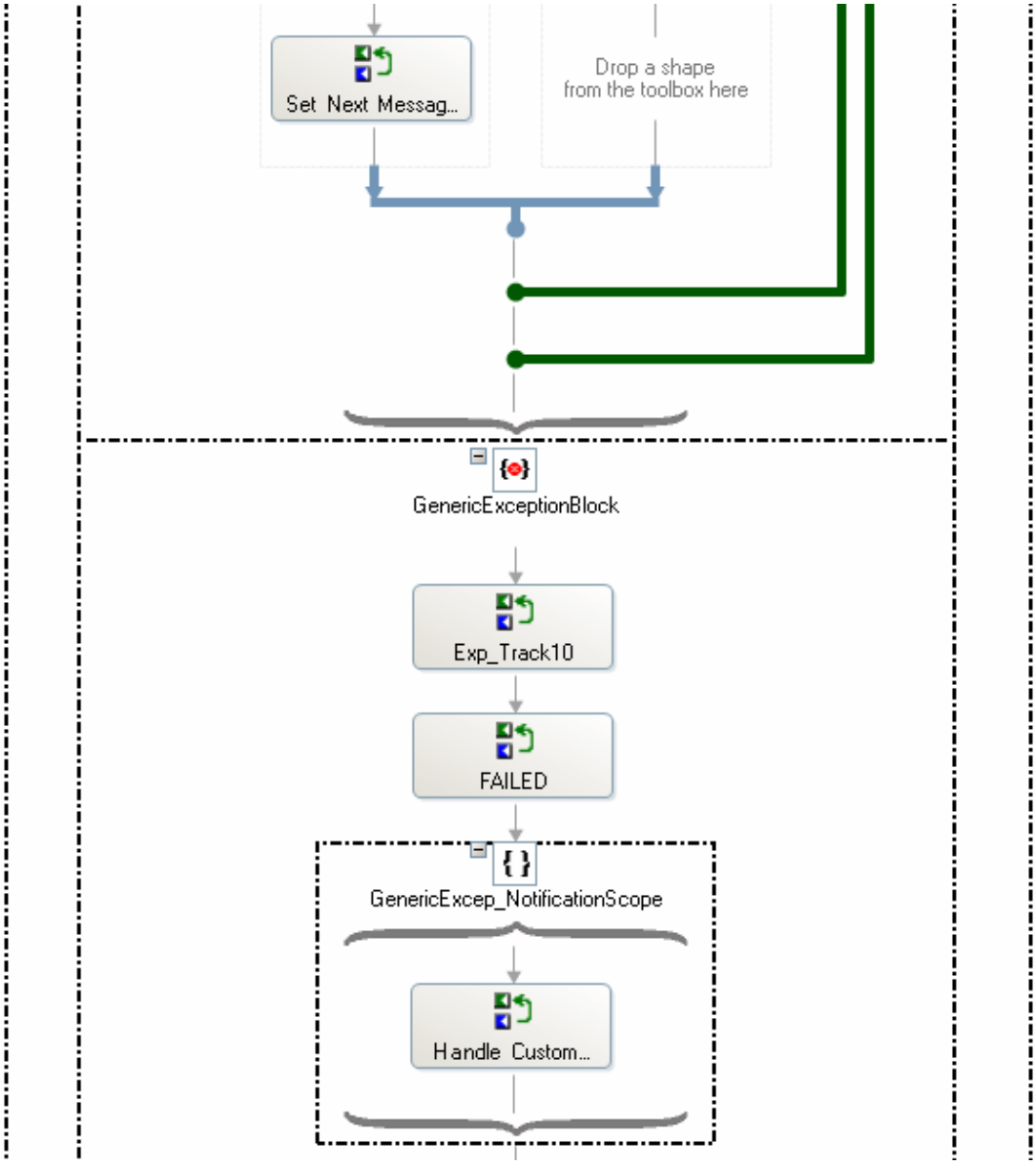
MessageEdifecs834- EDS.USGS.MED.EDI.Edifecs834Schema._834_A1_IG

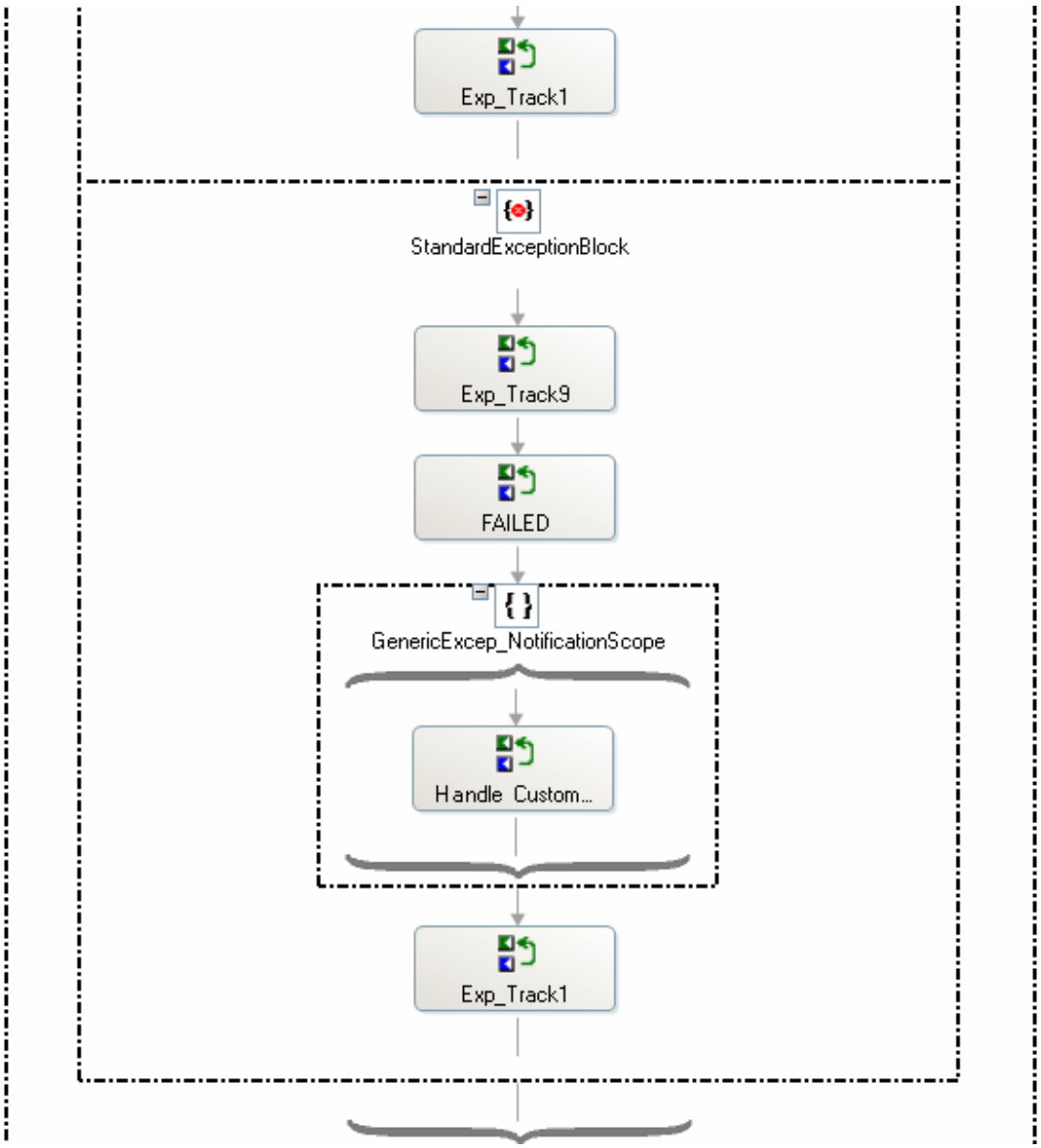
18.6.1.3 Correlation sets defined within this orchestration

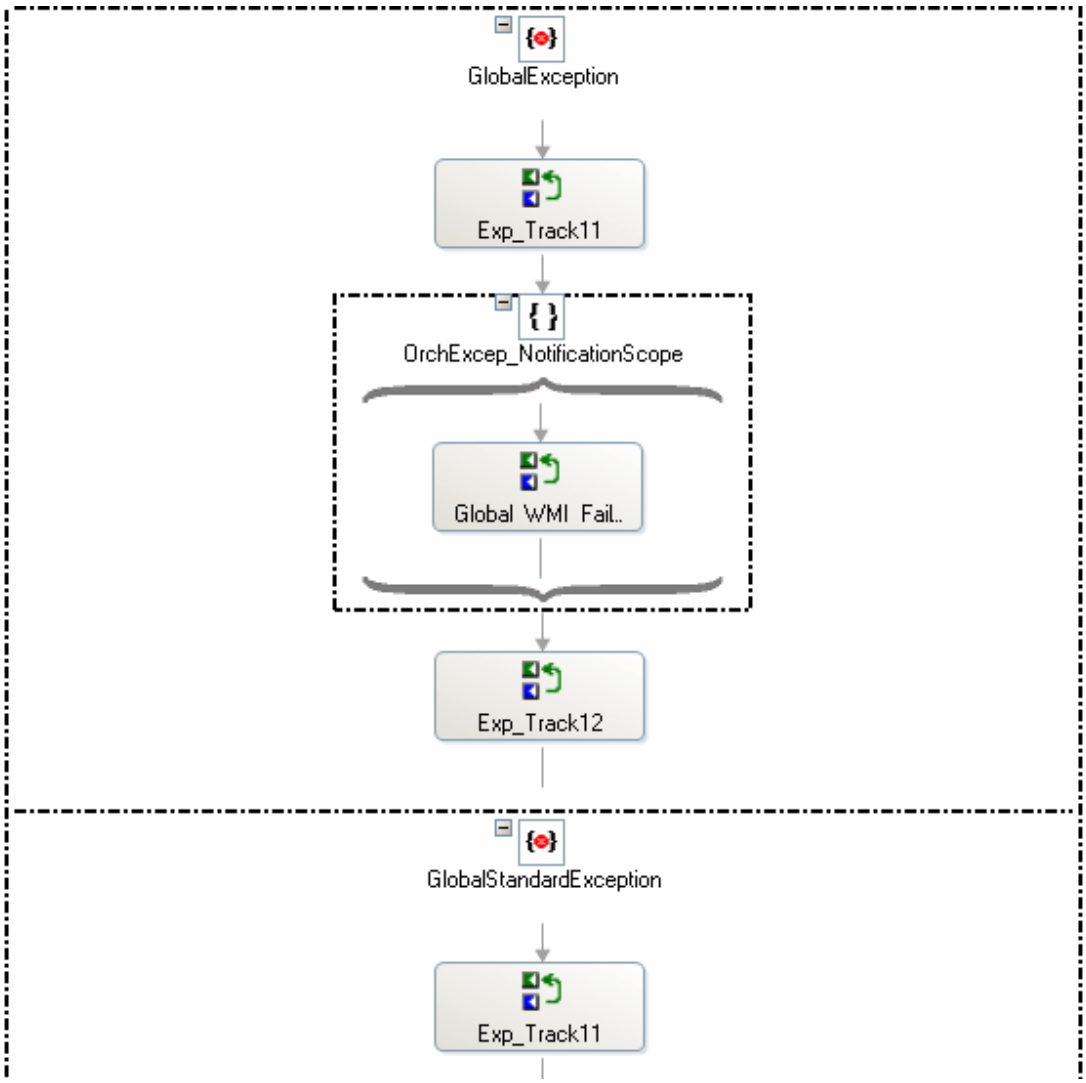
CorrelationForSingletonRecei-
EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.CorrelationTypeForSingletonReceive

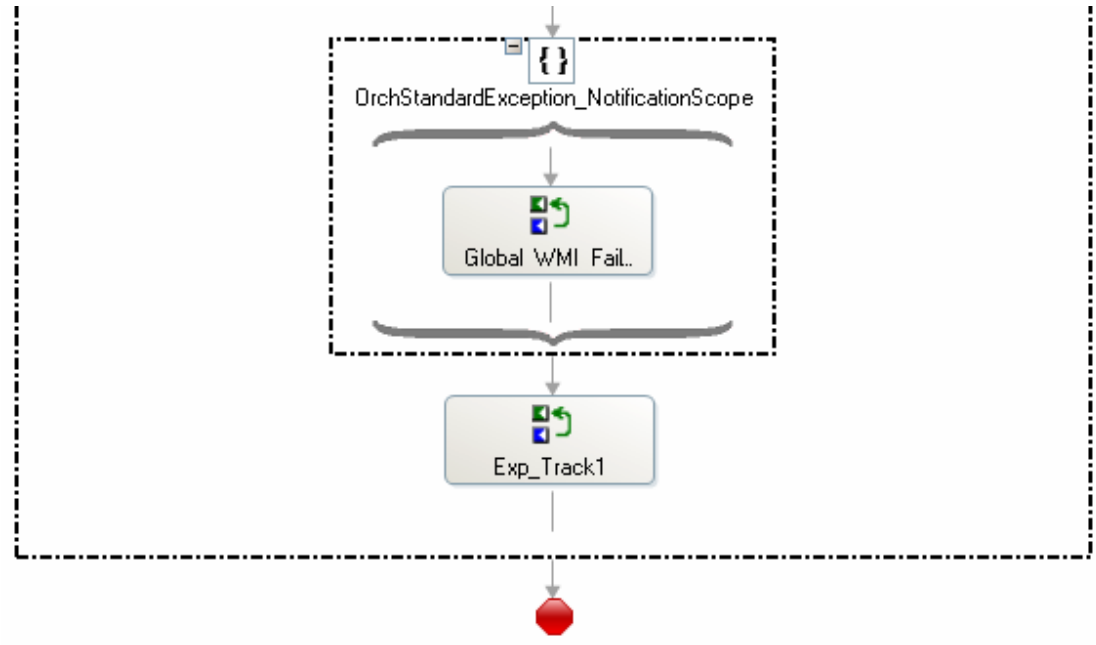

ScopeOrchestration
ReceiveIC834
ReadConfig
ScopeProcess
Singleton Loop
RCV_BY_BIZTA...
BuildEdifecs834
IC8342Edifecs8...


PopulateMsgEdif...
TRANSFORMED
AssignSendPort
SendEdifecs834
Set Next Messag...
Subsequent Messages Loop
Wait and Receive next message
Receive Subseq...
Delay For Next...


Set Next Messag...
Drop a shape
from the toolbox here
GenericExceptionBlock
Exp_Track10
FAILED
GenericExcep_NotificationScope
Handle Custom...


Exp_Track1
StandardExceptionBlock
Exp_Track9
FAILED
GenericExcep_NotificationScope
Handle Custom...
Exp_Track1


GlobalException
Exp_Track11
OrchExcep_NotificationScope
Global WMI Fail..
Exp_Track12
GlobalStandardException
Exp_Track11



18.7 Pipeline Configuration

18.7.1 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

18.7.1.1 Pipeline Process Flow

54. Pre-Assemble

55. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

56. Encode

18.7.1.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

18.7.1.3 Receive locations using this pipeline

There are no receive locations using this pipeline

18.7.2 GenericOutboundBenefitEnrollment834Messaging.RecvMemberEnrollmentPipeline

Pipeline Type: Receive

Assembly Name: GenericOutboundBenefitEnrollment834Messaging.RecvMemberEnrollmentPipeline, EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Messaging, Version=1.0.0.0, Culture=neutral, PublicKeyToken=a2fc6bba136a8ae3

18.7.2.1 Pipeline Process Flow

57. Decode

58. Disassemble

    - Flat file disassembler
    - Microsoft.BizTalk.Component.FFDasmComp,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

59. Validate

60. ResolveParty

18.7.2.2 Send ports using this pipeline

There are no send ports using this pipeline

18.7.2.3 Receive locations using this pipeline

RLBenefitEnrollGeneric

19 Appendix Q – Biz Talk Documentation – 837 Dental Inbound

19.1 Host Configuration

19.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

19.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

19.1.1.2 Receive locations running in this host

There are no receive locations running in this host

19.1.2 BizTalkServerProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

19.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType

19.1.2.2 Receive locations running in this host
There are no receive locations running in this host

19.1.3 BizTalkServerReceiveHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

19.1.3.1 Orchestrations running in this host
There are no orchestrations running in this host

19.1.3.2 Receive locations running in this host
RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

19.1.4 BizTalkServerSendHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

19.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

19.1.4.2 Receive locations running in this host

There are no receive locations running in this host

19.2 Send Port Configuration

19.2.1 SendClaimList837D

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

19.2.1.1 Primary Transport

Not yet specified

19.2.1.2 Secondary Transport

Not yet specified

19.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

19.3 Receive Port Configuration

19.3.1 ReceiveClaim837DXData

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

19.3.1.1 Receive Locations

61. RLClaim837DXDataGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\Claim837D\Inbound837D\XData*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

19.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

19.4 Protocol Configuration

19.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

19.4.1.1 Send ports using this protocol

There are no send ports using this protocol

19.4.1.2 Receive ports using this protocol

ReceiveClaim837DXData

19.5 Assembly Configuration

19.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

19.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

19.5.2 EDS.USGS.MED.EDI.Edifecs837DSchema

Name: EDS.USGS.MED.EDI.Edifecs837DSchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

19.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs837DSchema._837_Q2_IG

19.5.3 EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Name: EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

19.5.3.1 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

19.5.4 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DMessaging

Name: EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

19.5.4.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DMessaging.Edifecs837D_Dependant_Xml_To_ClaimList

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DMessaging.Edifecs837D_Subscriber_Xml_To_ClaimList

19.5.5 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses

Name: EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

19.5.5.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

19.6 Orchestration Configuration

19.6.1 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

19.6.1.1 Ports contained within this orchestration

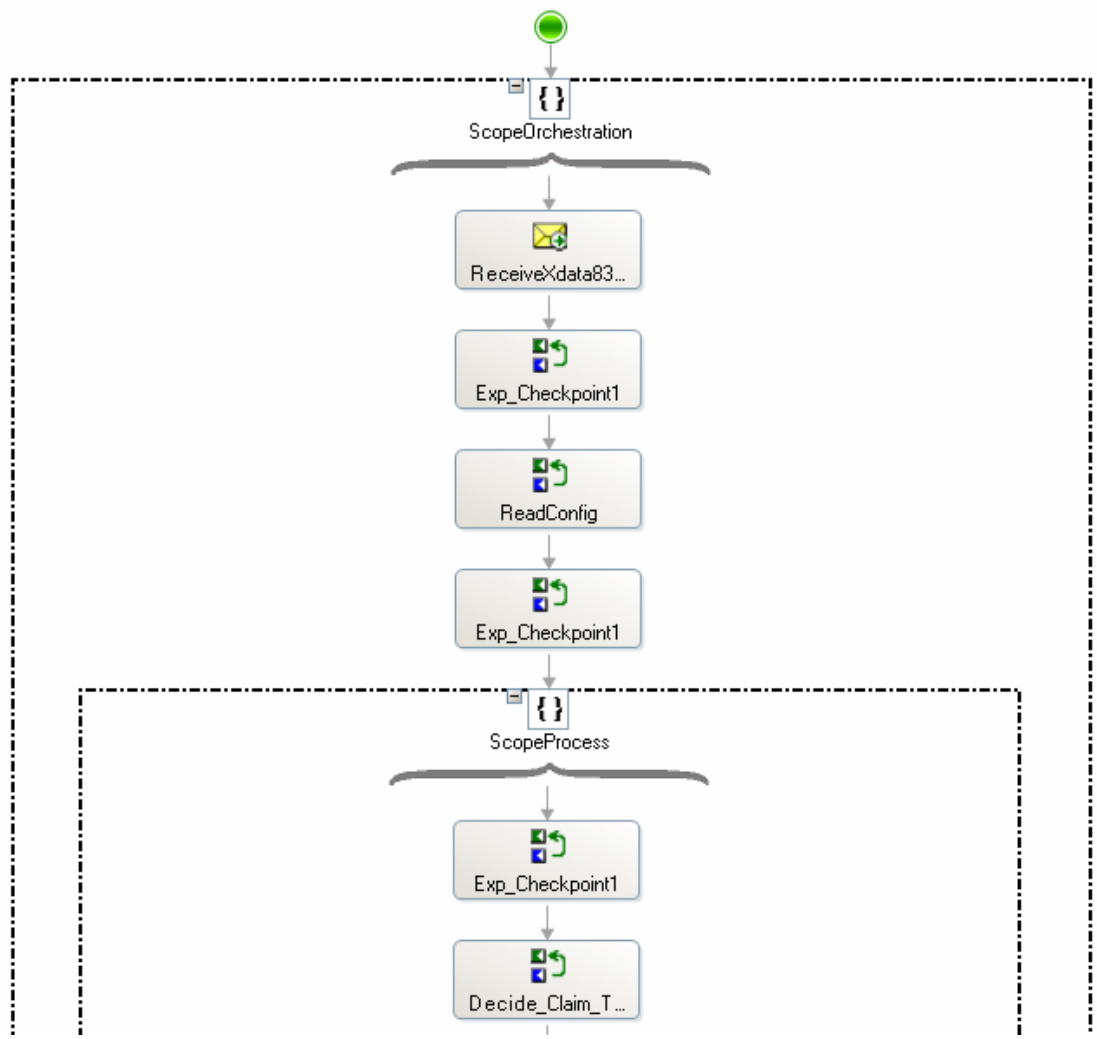
prtReceiveXData837D - Bound to: ReceiveClaim837DXData

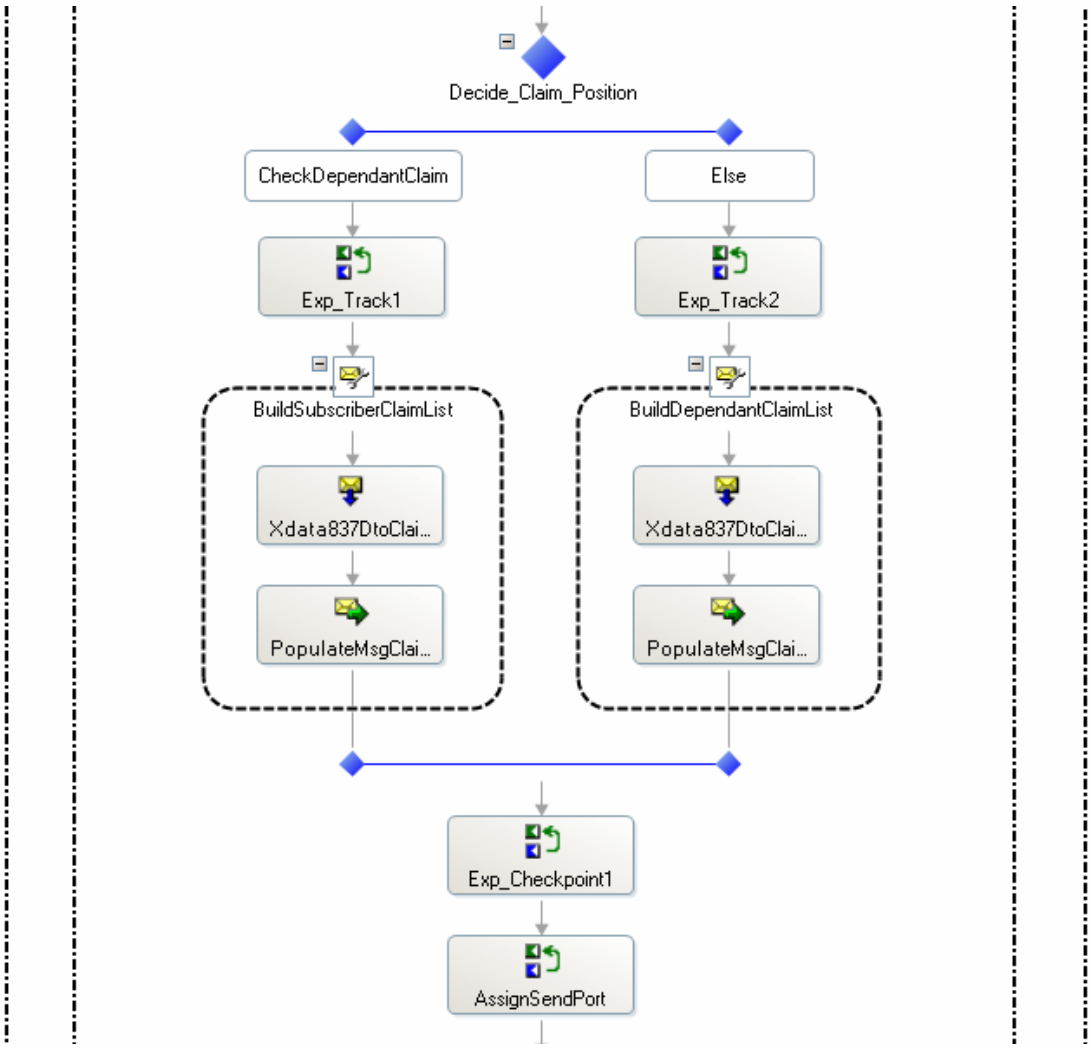
prtSendClaimList - Bound to: SendClaimList837D

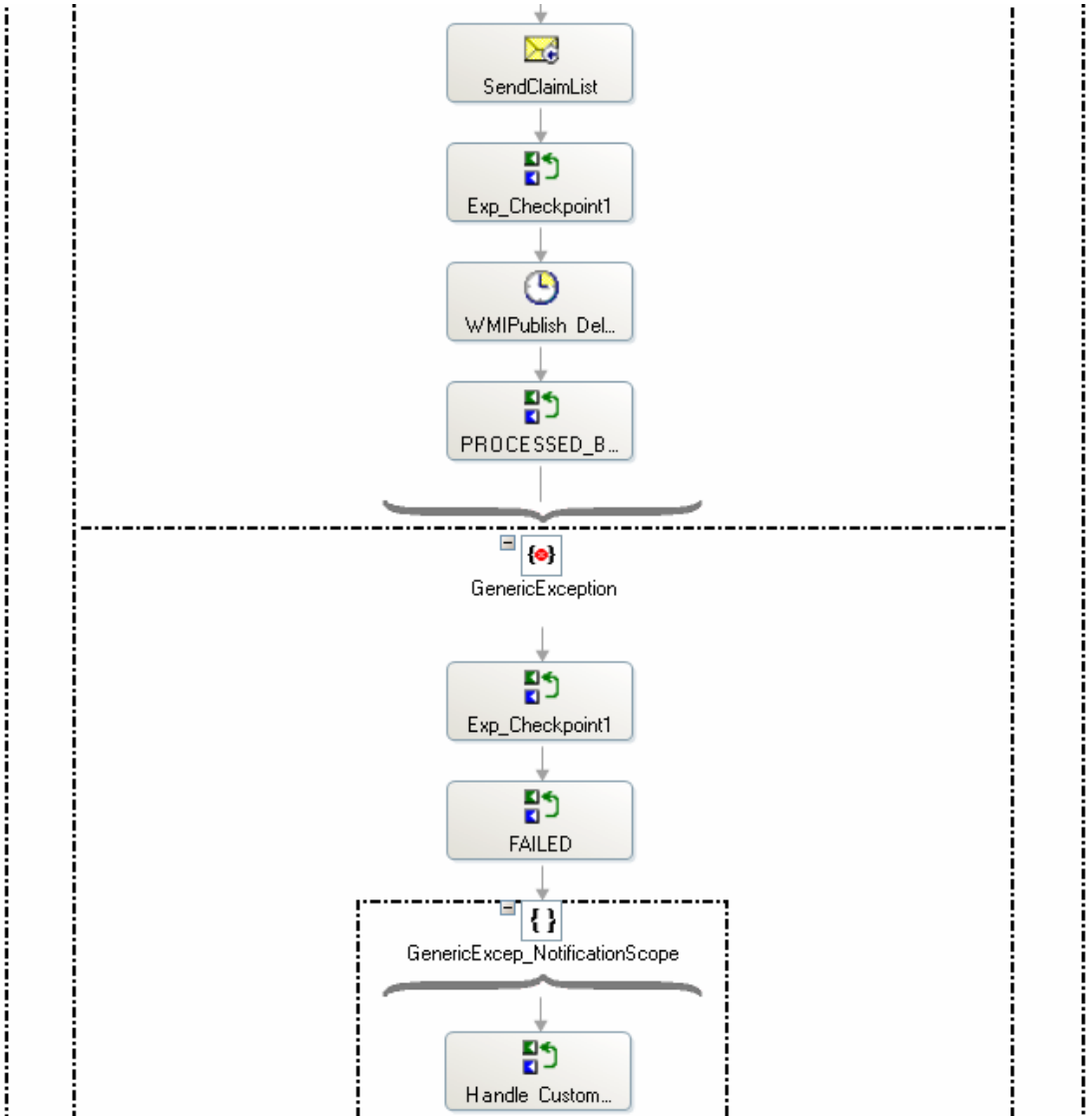
19.6.1.2 Messages defined within this orchestration

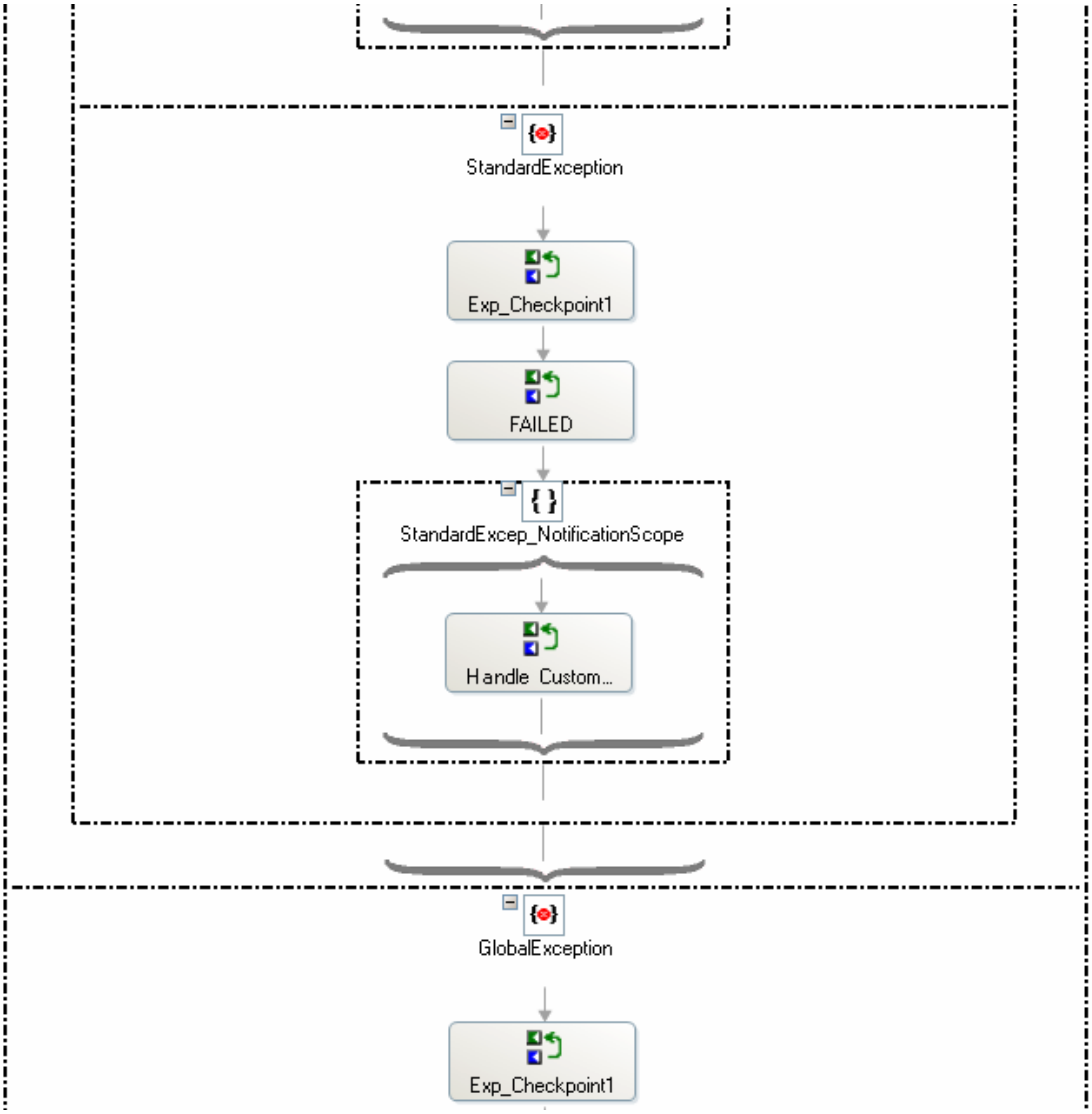
MessageClaimList- EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

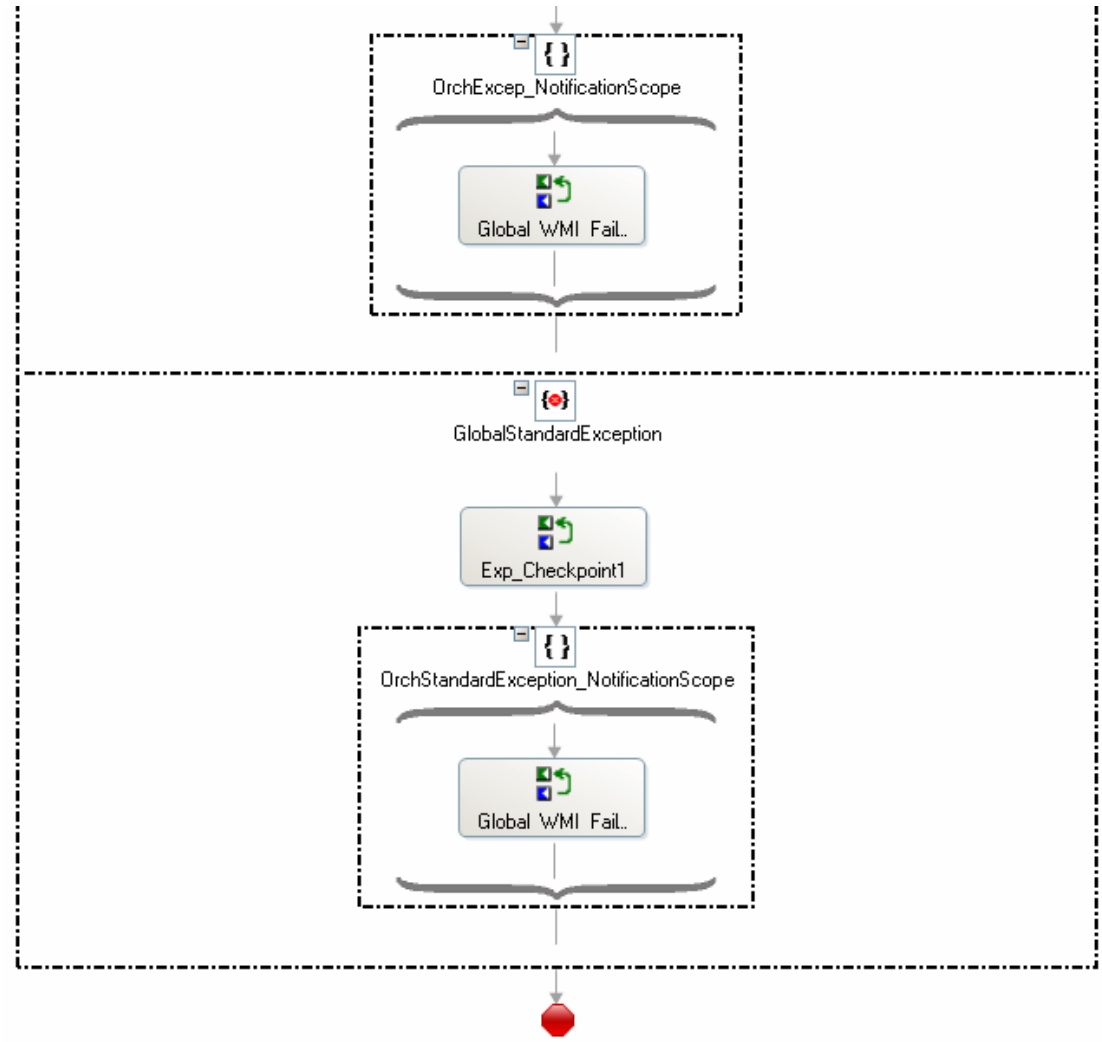
MessageXdata837D- EDS.USGS.MED.EDI.Edifecs837DSchema._837_Q2_IG




Decide_Claim_Position
CheckDependantClaim
Else
Exp_Track1
Exp_Track2
BuildSubscriberClaimList
BuildDependantClaimList
Xdata837DtoClai...
Xdata837DtoClai...
PopulateMsgClai...
PopulateMsgClai...
Exp_Checkpoint1
AssignSendPort


SendClaimList
Exp_Checkpoint1
WMIPublish_Del...
PROCESSED_B...
GenericException
Exp_Checkpoint1
FAILED
GenericExcep_NotificationScope
Handle_Custom...


StandardException
Exp_Checkpoint1
FAILED
StandardExcep_NotificationScope
Handle_Custom...
GlobalException
Exp_Checkpoint1


OrchExcep_NotificationScope
Global WMI Fail..
GlobalStandardException
Exp_Checkpoint1
OrchStandardException_NotificationScope
Global WMI Fail..

19.7 Pipeline Configuration

19.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

19.7.1.1 Pipeline Process Flow

62. Decode

63. Disassemble

    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

64. Validate

65. ResolveParty

    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

19.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

19.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

19.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

19.7.2.1 Pipeline Process Flow

66. Pre-Assemble

67. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

68. Encode

19.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

19.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

20 Appendix R – Biz Talk Documentation – 837 Institutional Inbound

20.1 Host Configuration

20.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

20.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

20.1.1.2 Receive locations running in this host

There are no receive locations running in this host

20.1.2 BizTalkServer837IProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

20.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess

20.1.2.2 Receive locations running in this host

There are no receive locations running in this host

20.1.3 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

20.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

20.1.3.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

20.1.4 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

20.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

20.1.4.2 Receive locations running in this host

There are no receive locations running in this host

20.2 Send Port Configuration

20.2.1 SendClaimList837I

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

20.2.1.1 Primary Transport

Not yet specified

20.2.1.2 Secondary Transport

Not yet specified

20.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess

20.3 Receive Port Configuration

20.3.1 ReceiveInbound837IXData

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

20.3.1.1 Receive Locations

69. RLInbound837IClaimGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\Claim837I\Inbound837I\XData*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

20.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess

20.4 Protocol Configuration

20.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

20.4.1.1 Send ports using this protocol

There are no send ports using this protocol

20.4.1.2 Receive ports using this protocol

ReceiveInbound837IXData

20.5 Assembly Configuration

20.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

20.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

20.5.2 EDS.USGS.MED.EDI.Edifecs837ISchema

Name: EDS.USGS.MED.EDI.Edifecs837ISchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

20.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs837ISchema._837_Q3_IG

20.5.3 EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Name: EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

20.5.3.1 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

20.5.4 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IMessaging

Name: EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

20.5.4.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IMessaging.Edifecs837I_Dependant_Xml_To_ClaimList

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IMessaging.Edifecs837I_Subscriber_Xml_To_ClaimList

20.5.5 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses

Name: EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

20.5.5.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess

20.6 Orchestration Configuration

20.6.1 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

20.6.1.1 Ports contained within this orchestration

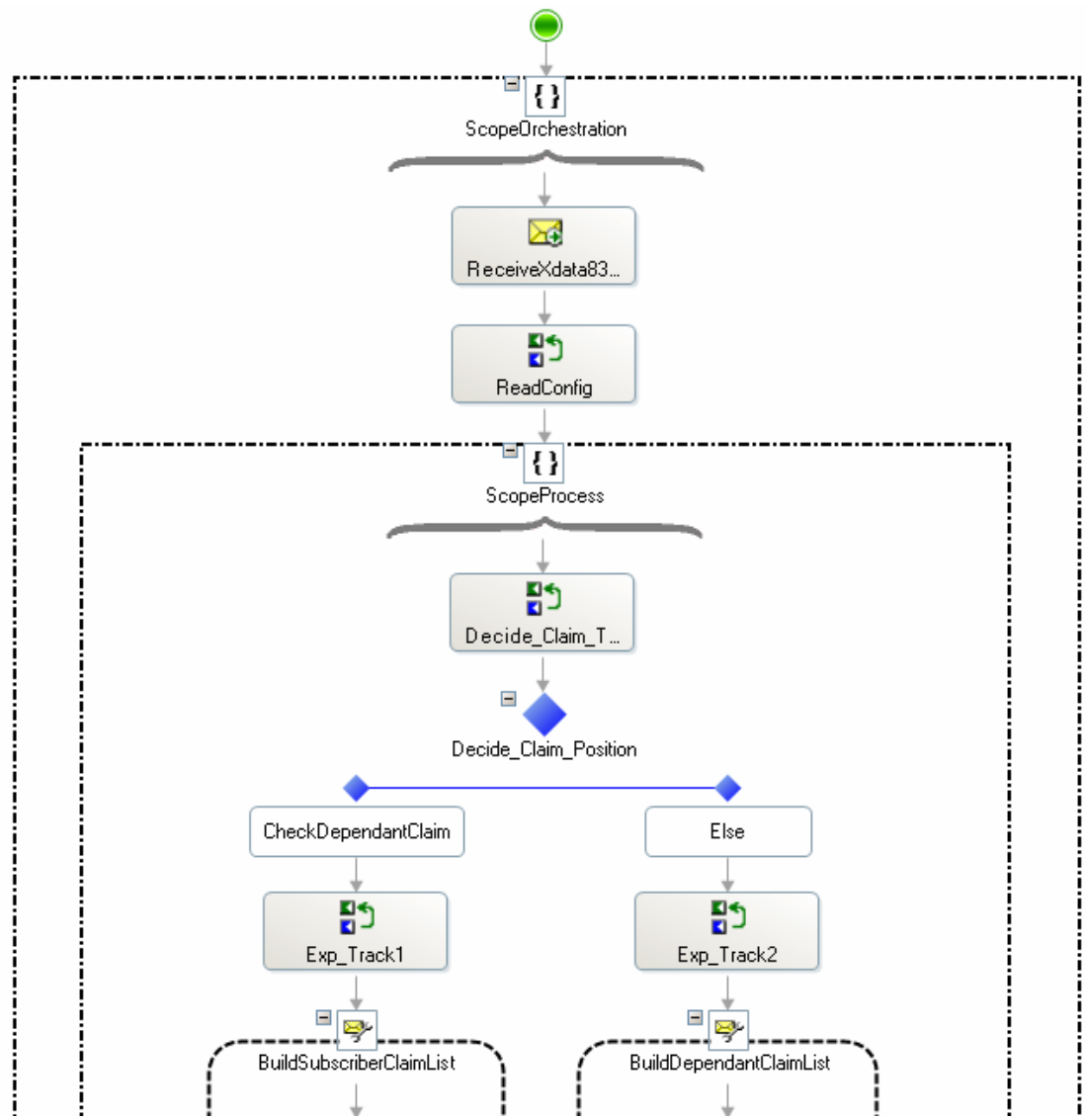
prtReceiveXData837I - Bound to: ReceiveInbound837IXData

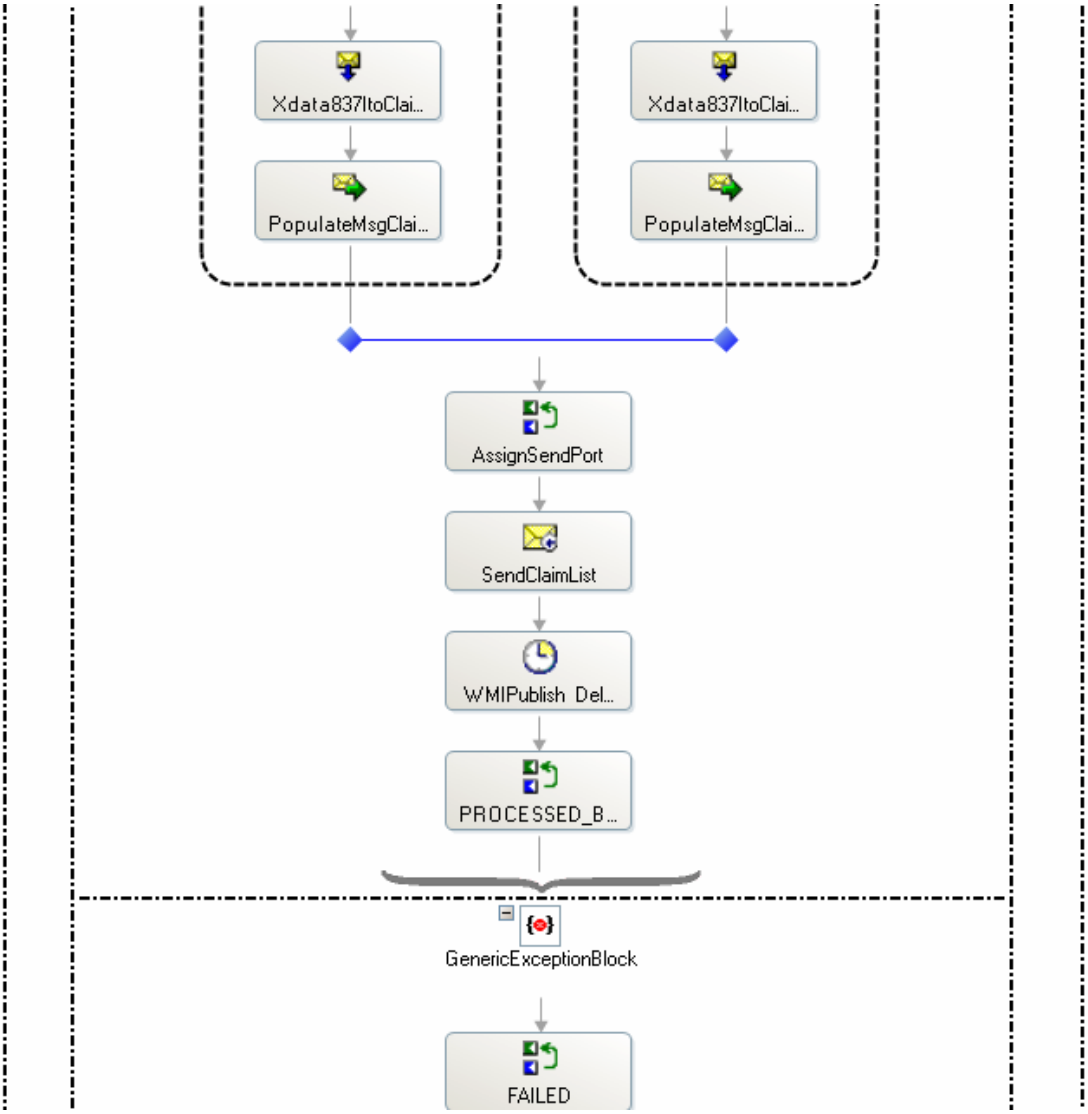
prtSendClaimList - Bound to: SendClaimList837I

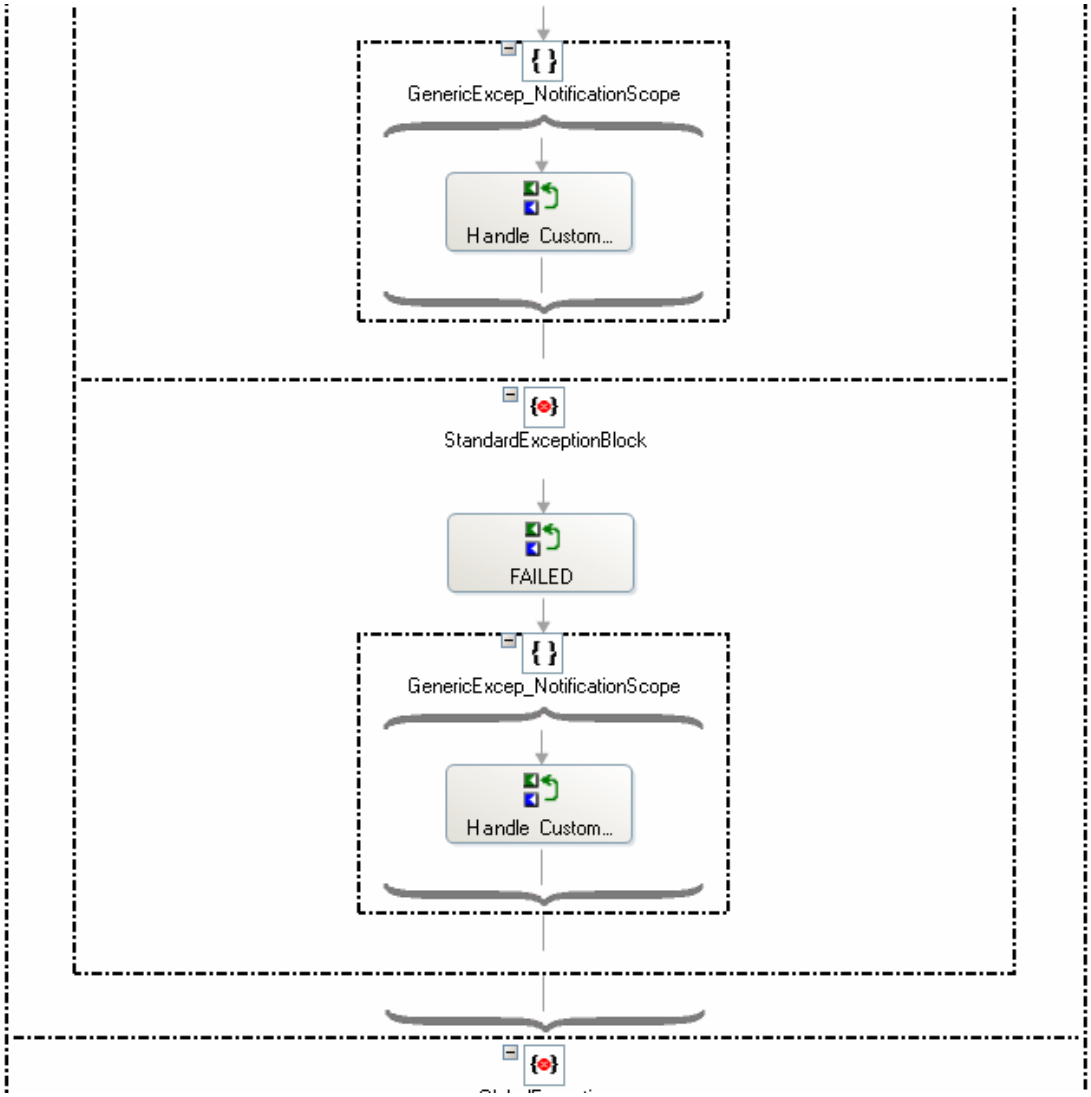
20.6.1.2 Messages defined within this orchestration

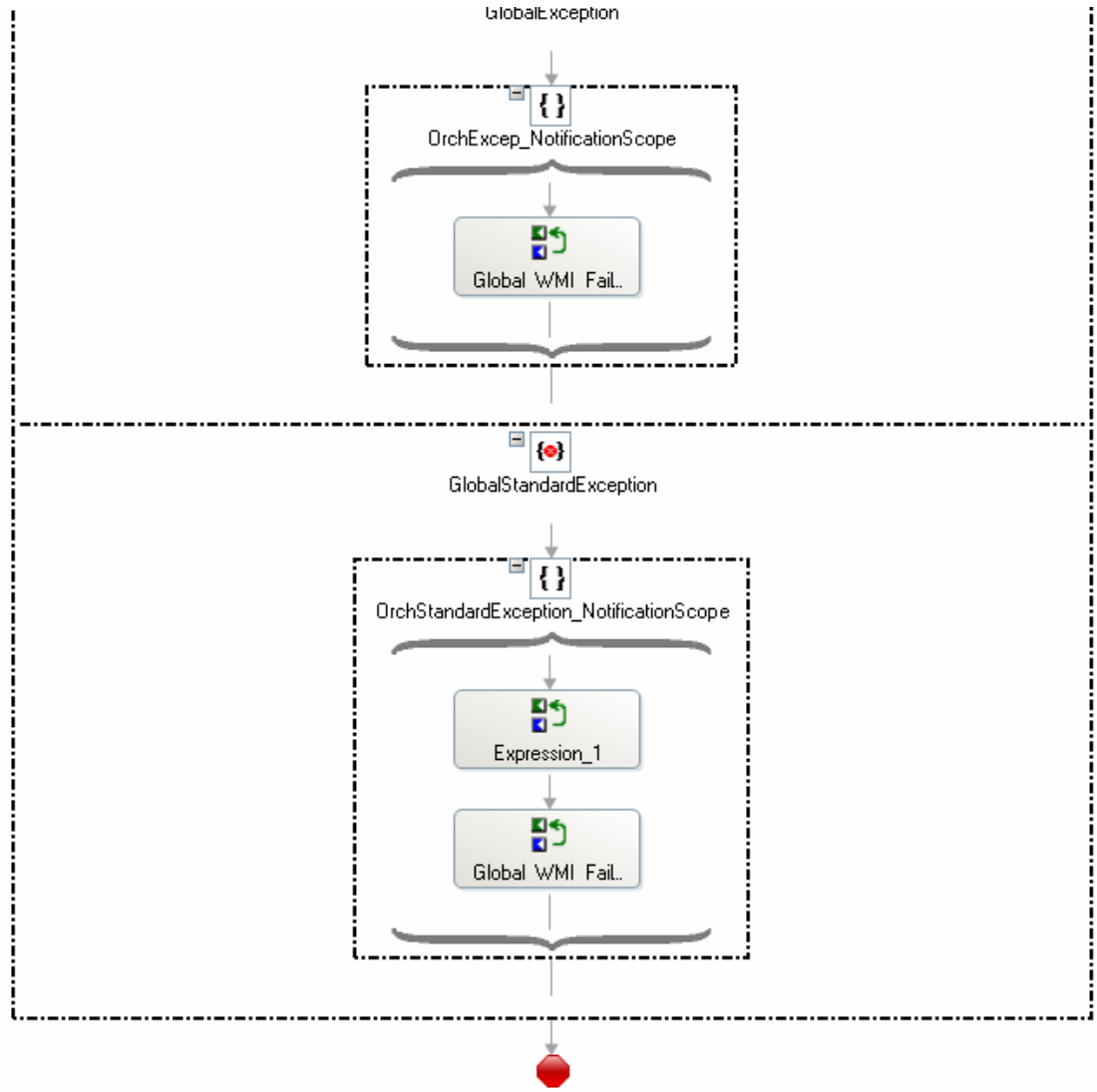
MessageClaimList- EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

MessageXdata837I- EDS.USGS.MED.EDI.Edifecs837ISchema._837_Q3_IG


ScopeOrchestration
ReceiveXdata83...
ReadConfig
ScopeProcess
Decide_Claim_T...
Decide_Claim_Position
CheckDependantClaim
Else
Exp_Track1
Exp_Track2
BuildSubscriberClaimList
BuildDependantClaimList


Xdata837ItoClai...
PopulateMsgClai...
Xdata837ItoClai...
PopulateMsgClai...
AssignSendPort
SendClaimList
WMIPublish_Del...
PROCESSED_B...
GenericExceptionBlock
FAILED


GenericExcep_NotificationScope
Handle Custom...
StandardExceptionBlock
FAILED
GenericExcep_NotificationScope
Handle Custom...


GlobalException
OrchExcep_NotificationScope
Global WMI Fail..
GlobalStandardException
OrchStandardException_NotificationScope
Expression_1
Global WMI Fail..

20.7 Pipeline Configuration

20.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

20.7.1.1 Pipeline Process Flow

70. Decode

71. Disassemble

    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

72. Validate

73. ResolveParty

    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

20.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

20.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

20.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

20.7.2.1 Pipeline Process Flow

74. Pre-Assemble

75. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

76. Encode

20.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

20.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

21 Appendix S – Biz Talk Documentation – 837 Professional Inbound

21.1 Host Configuration

21.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

21.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

21.1.1.2 Receive locations running in this host

There are no receive locations running in this host

21.1.2 BizTalkServer837PProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

21.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess

21.1.2.2 Receive locations running in this host

There are no receive locations running in this host

21.1.3 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

21.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

21.1.3.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

21.1.4 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

21.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

21.1.4.2 Receive locations running in this host

There are no receive locations running in this host

21.2 Send Port Configuration

21.2.1 SendClaimList837P

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

21.2.1.1 Primary Transport

Not yet specified

21.2.1.2 Secondary Transport

Not yet specified

21.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess

21.3 Receive Port Configuration

21.3.1 ReceiveClaim837PXData

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

21.3.1.1 Receive Locations

77. RLClaim837PXDataGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\Claim837P\Inbound837P\XData*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

21.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess

21.4 Protocol Configuration

21.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

21.4.1.1 Send ports using this protocol

There are no send ports using this protocol

21.4.1.2 Receive ports using this protocol

ReceiveClaim837PXData

21.5 Assembly Configuration

21.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

21.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

21.5.2 EDS.USGS.MED.EDI.Edifecs837PSchema

Name: EDS.USGS.MED.EDI.Edifecs837PSchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

21.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs837PSchema._837_Q1_IG

21.5.3 EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Name: EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

21.5.3.1 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

21.5.4 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PMessaging

Name: EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

21.5.4.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PMessaging.Edifecs837P_Subscriber_Xml_To_ClaimList

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PMessaging.Edifecs837P_Dependant_Xml_To_ClaimList

21.5.5 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses

Name: EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

21.5.5.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess

21.6 Orchestration Configuration

21.6.1 EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

21.6.1.1 Ports contained within this orchestration

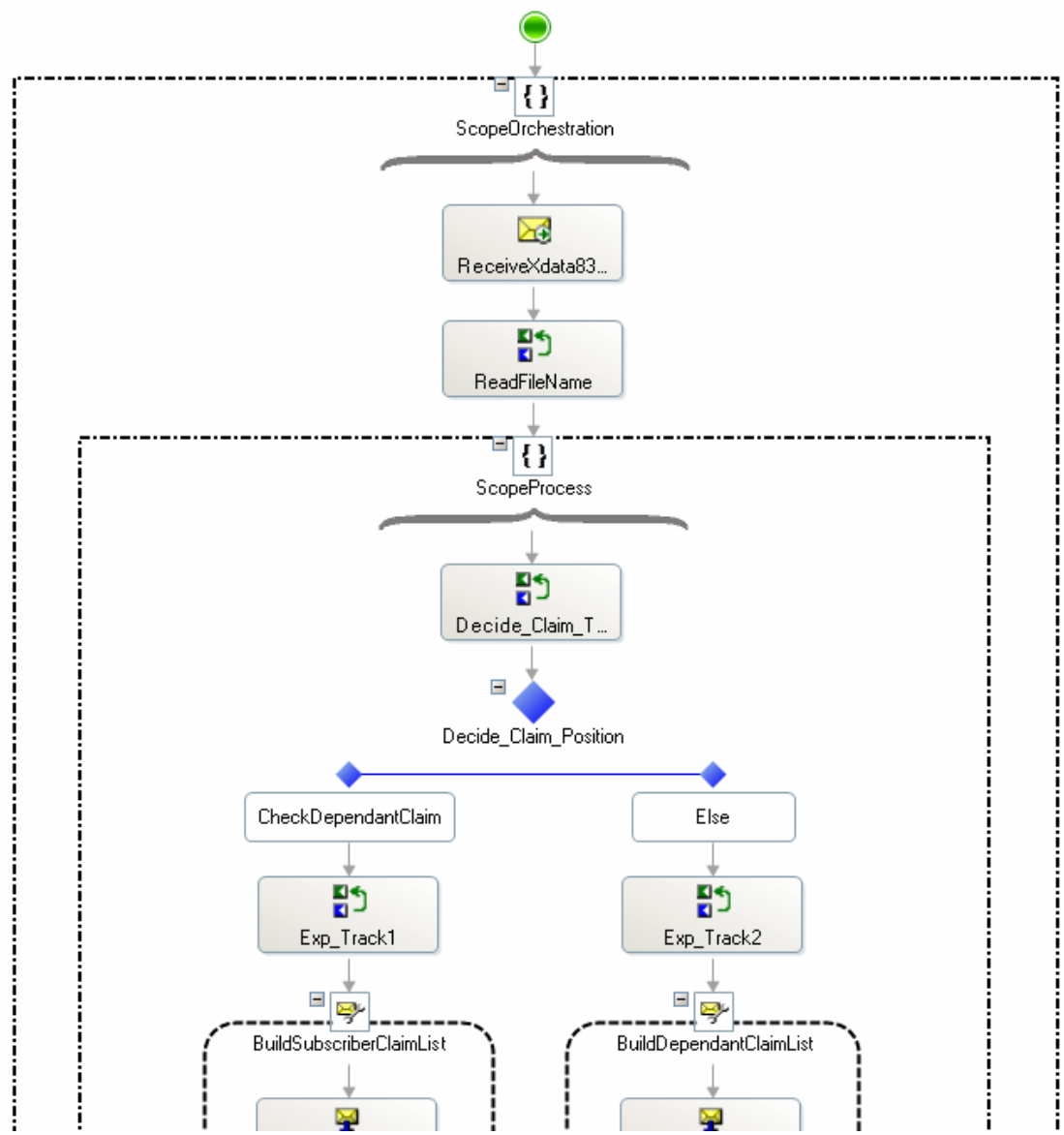
prtReceiveXData837P - Bound to: ReceiveClaim837PXData

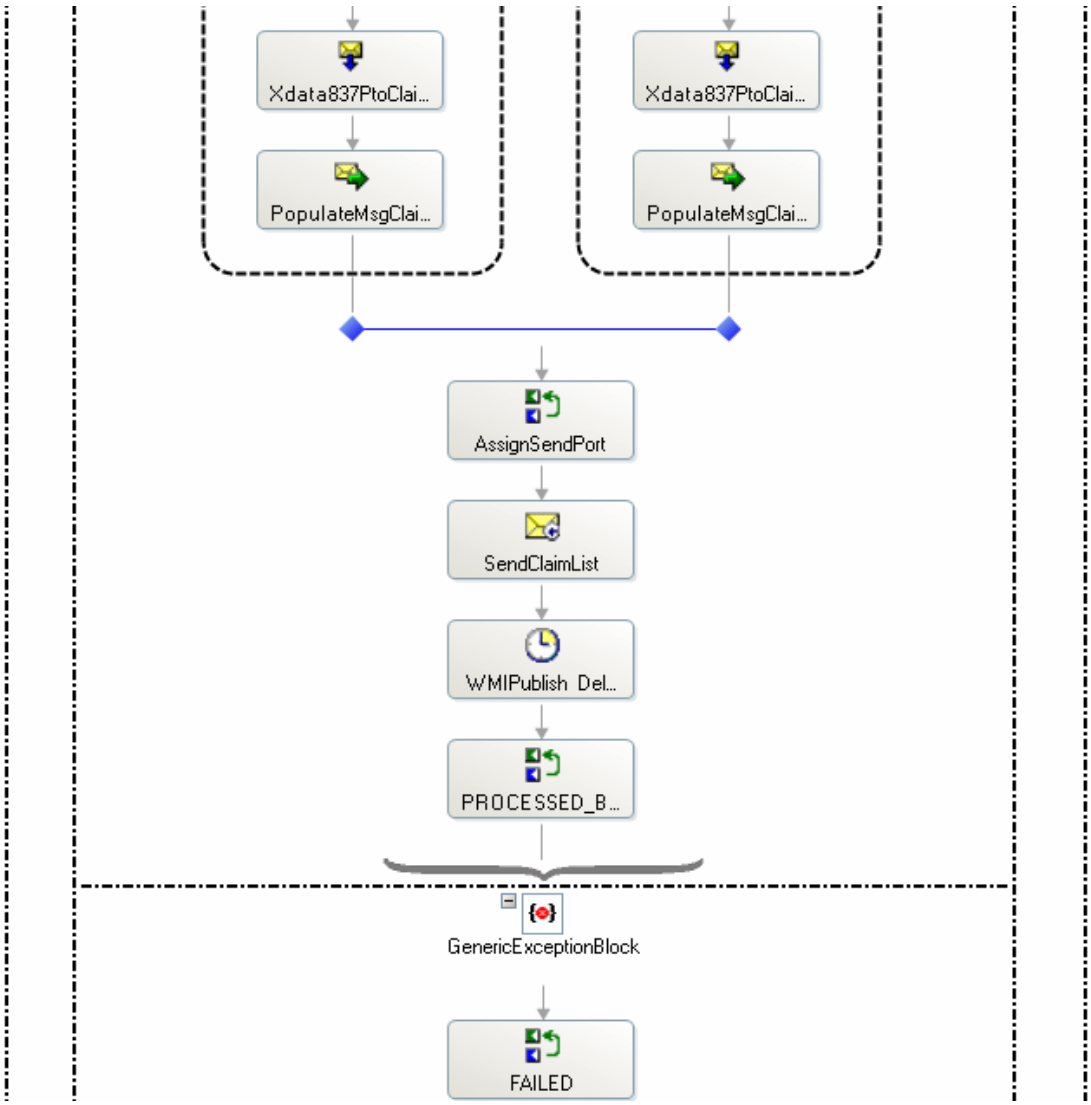
prtSendClaimList - Bound to: SendClaimList837P

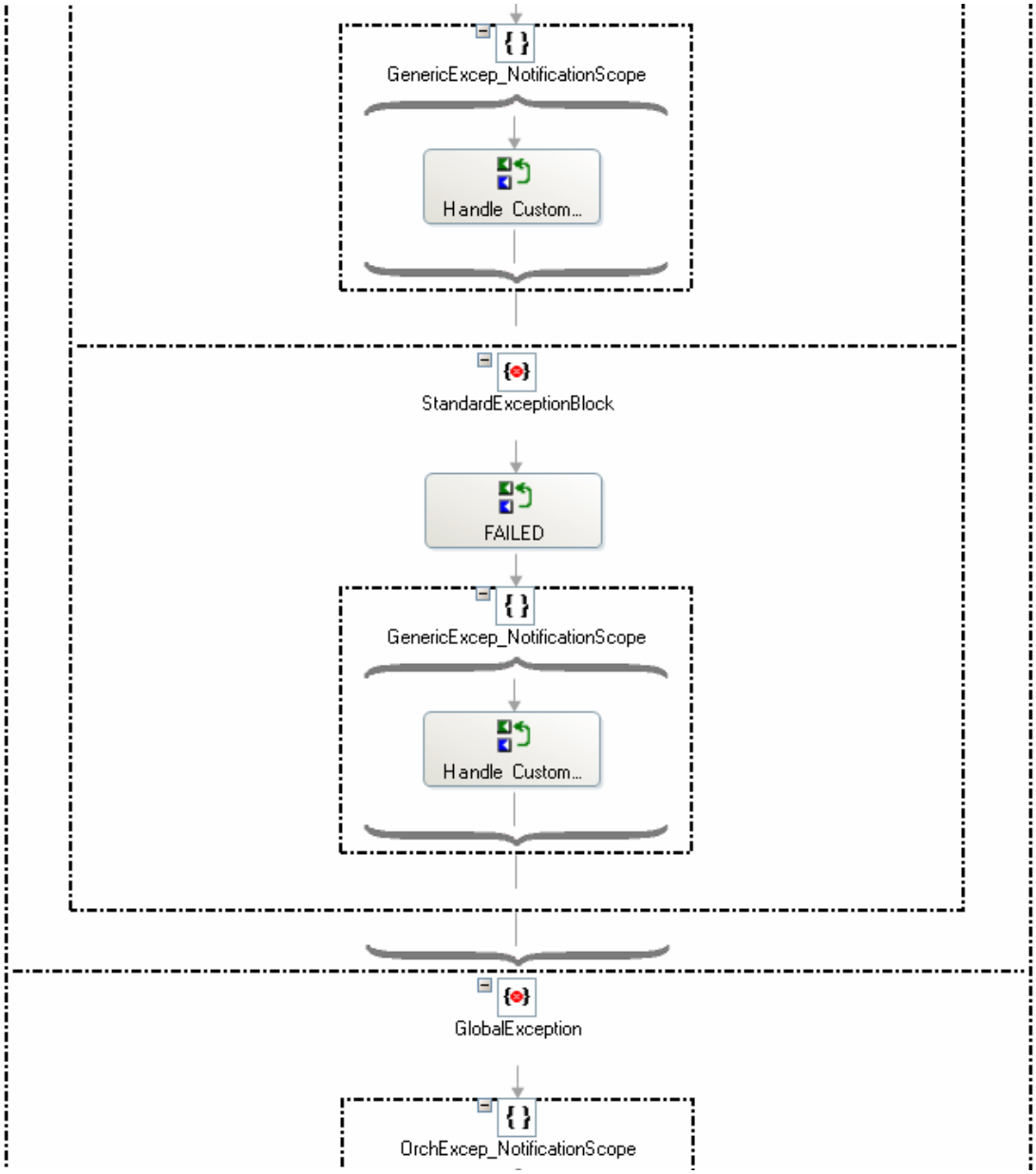
21.6.1.2 Messages defined within this orchestration

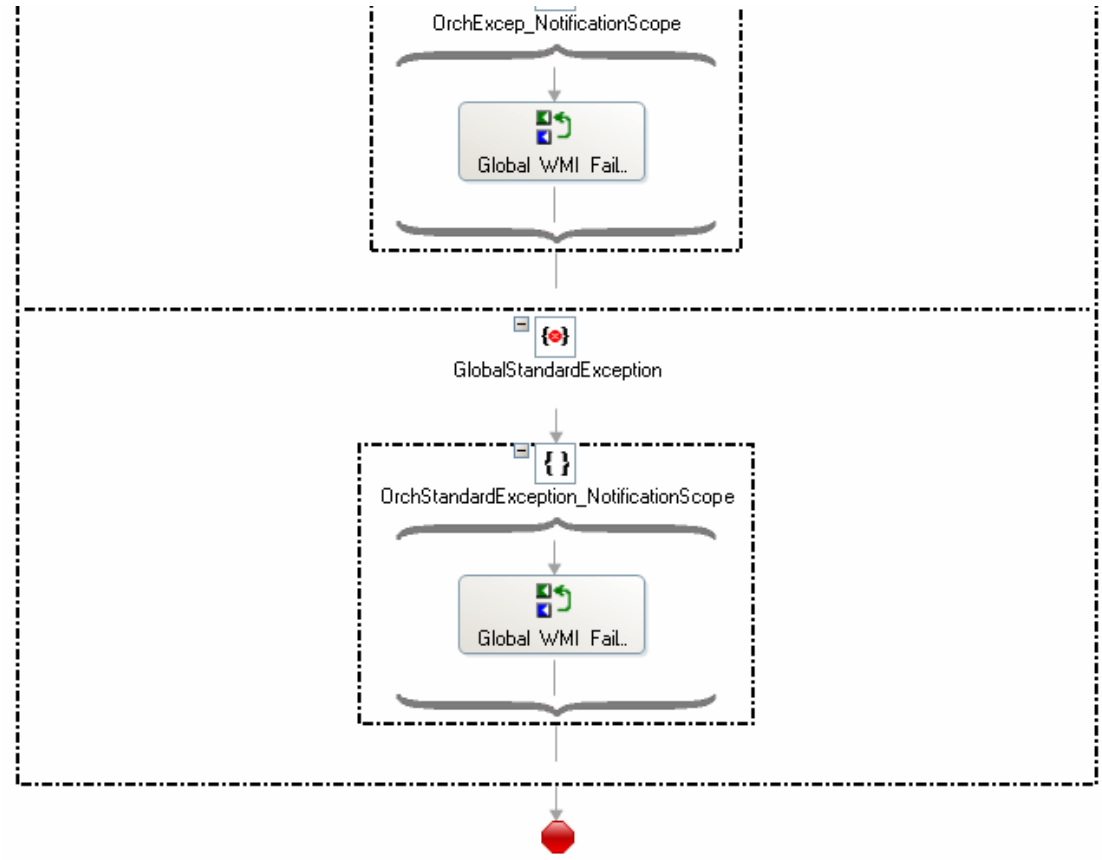
MessageClaimList – EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

MessageXdata837P- EDS.USGS.MED.EDI.Edifecs837PSchema._837_Q1_IG


ScopeOrchestration
ReceiveXdata83...
ReadFileName
ScopeProcess
Decide_Claim_T...
Decide_Claim_Position
CheckDependantClaim
Else
Exp_Track1
Exp_Track2
BuildSubscriberClaimList
BuildDependantClaimList


Xdata837PtoClai...
PopulateMsgClai...
Xdata837PtoClai...
PopulateMsgClai...
AssignSendPort
SendClaimList
WMIPublish_Del...
PROCESSED_B...
GenericExceptionBlock
FAILED


GenericExcep_NotificationScope
Handle Custom...
StandardExceptionBlock
FAILED
GenericExcep_NotificationScope
Handle Custom...
GlobalException
OrchExcep_NotificationScope


OrchExcep_NotificationScope
Global WMI Fail..
GlobalStandardException
OrchStandardException_NotificationScope
Global WMI Fail..

21.7 Pipeline Configuration

21.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

21.7.1.1 Pipeline Process Flow

78. Decode

79. Disassemble

    - XML Disassembler
    - Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

80. Validate

81. ResolveParty

    - Party Resolution
    - Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

21.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

21.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

21.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

21.7.2.1 Pipeline Process Flow

82. Pre-Assemble

83. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

84. Encode

21.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

21.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

22 Appendix T – Biz Talk Documentation – NSF Transportation

22.1 Host Configuration

22.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

22.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

22.1.1.2 Receive locations running in this host

There are no receive locations running in this host

22.1.2 BizTalkServerProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

22.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType

22.1.2.2 Receive locations running in this host
There are no receive locations running in this host

22.1.3 BizTalkServerReceiveHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

22.1.3.1 Orchestrations running in this host
There are no orchestrations running in this host

22.1.3.2 Receive locations running in this host
RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

22.1.4 BizTalkServerSendHost
Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

22.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

22.1.4.2 Receive locations running in this host

There are no receive locations running in this host

22.2 Send Port Configuration

22.2.1 SendNSFTransportationClaimList

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

22.2.1.1 Primary Transport

Not yet specified

22.2.1.2 Secondary Transport

Not yet specified

22.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

22.3 Receive Port Configuration

22.3.1 ReceiveNSFTransportEnctr

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

22.3.1.1 Receive Locations

85. RLNSFTransportEnctrGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\KYTCNSF\Inbound\Preprocessor*.*
Transport Protocol	FILE
Receive Pipeline	EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterMessaging.RecvNSFTranspotEncounterPipeline
Receive Handler	BizTalkServerReceiveHost

22.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

22.4 Protocol Configuration

22.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

22.4.1.1 Send ports using this protocol

There are no send ports using this protocol

22.4.1.2 Receive ports using this protocol

ReceiveNSFTransportEnctr

22.5 Assembly Configuration

22.5.1 EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Name: EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

22.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

22.5.2 EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterMessaging

Name: EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

22.5.2.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterMessaging.NSFTransportationEncounter_To_ClaimList

22.5.3 EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses

Name: EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

22.5.3.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

22.6 Orchestration Configuration

22.6.1 EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses,Version=1.0.0.0, Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

22.6.1.1 Ports contained within this orchestration

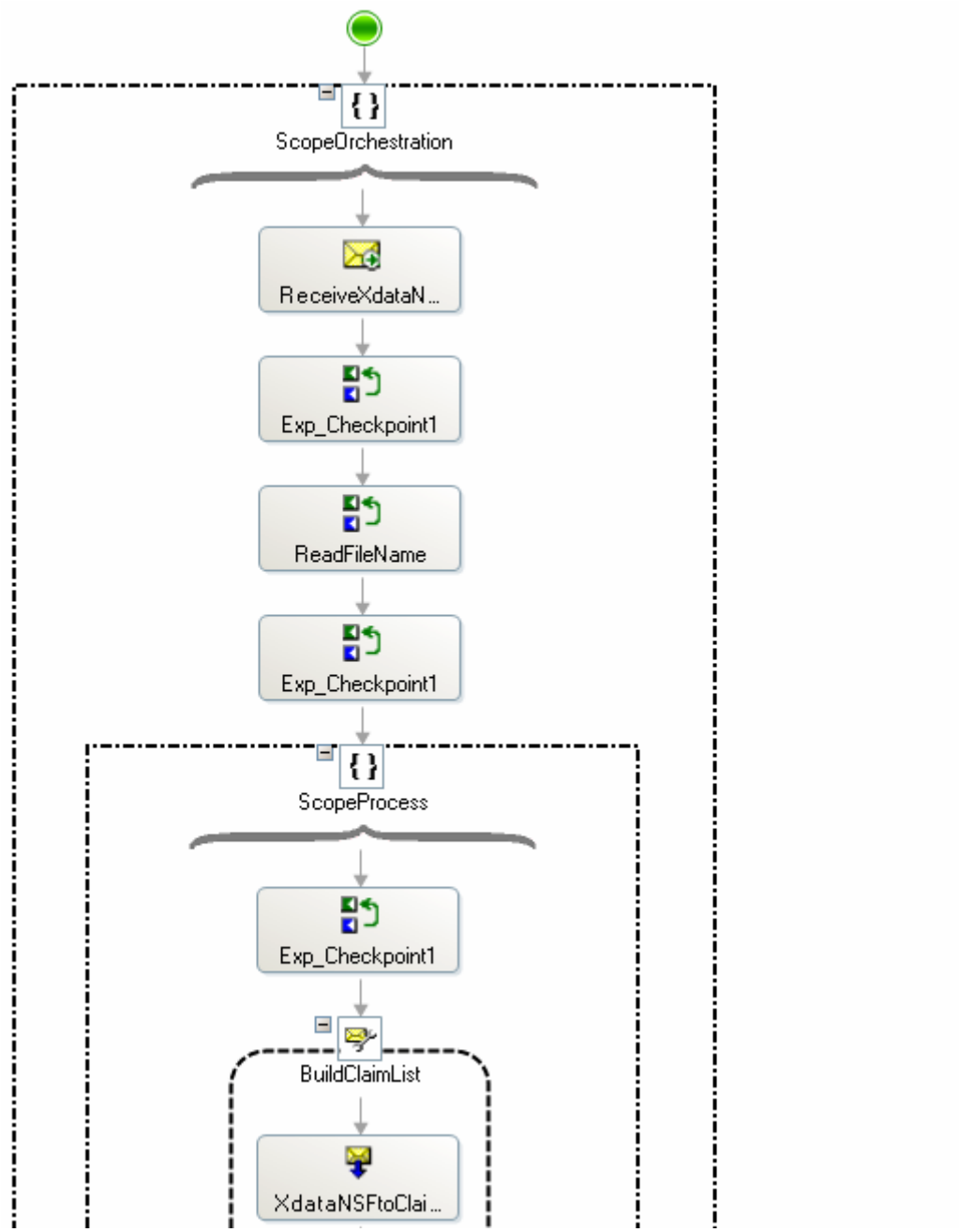
prtReceiveXDataNSFTransport - Bound to: ReceiveNSFTransportEnctr

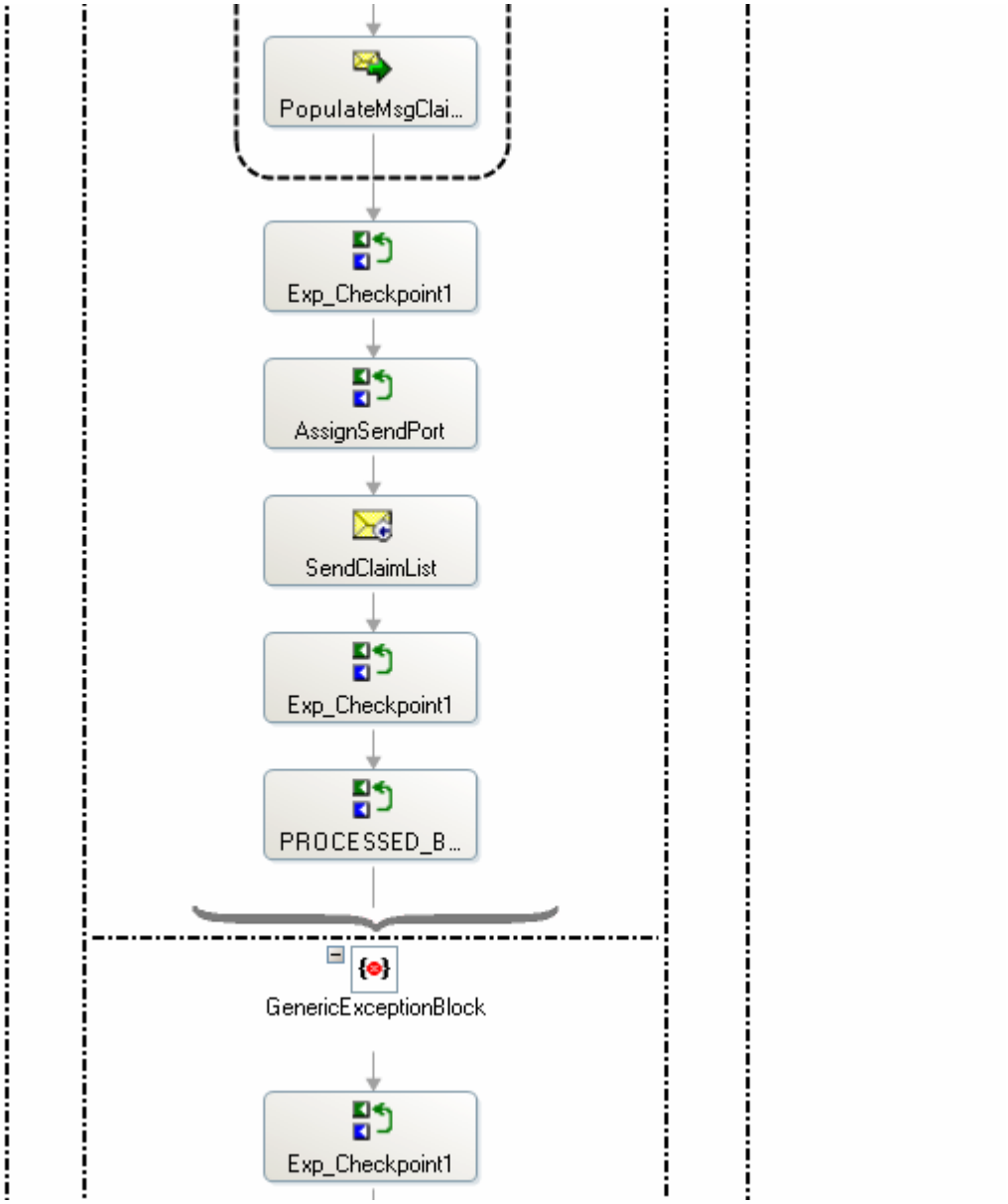
prtSendClaimList - Bound to: SendNSFTransportationClaimList

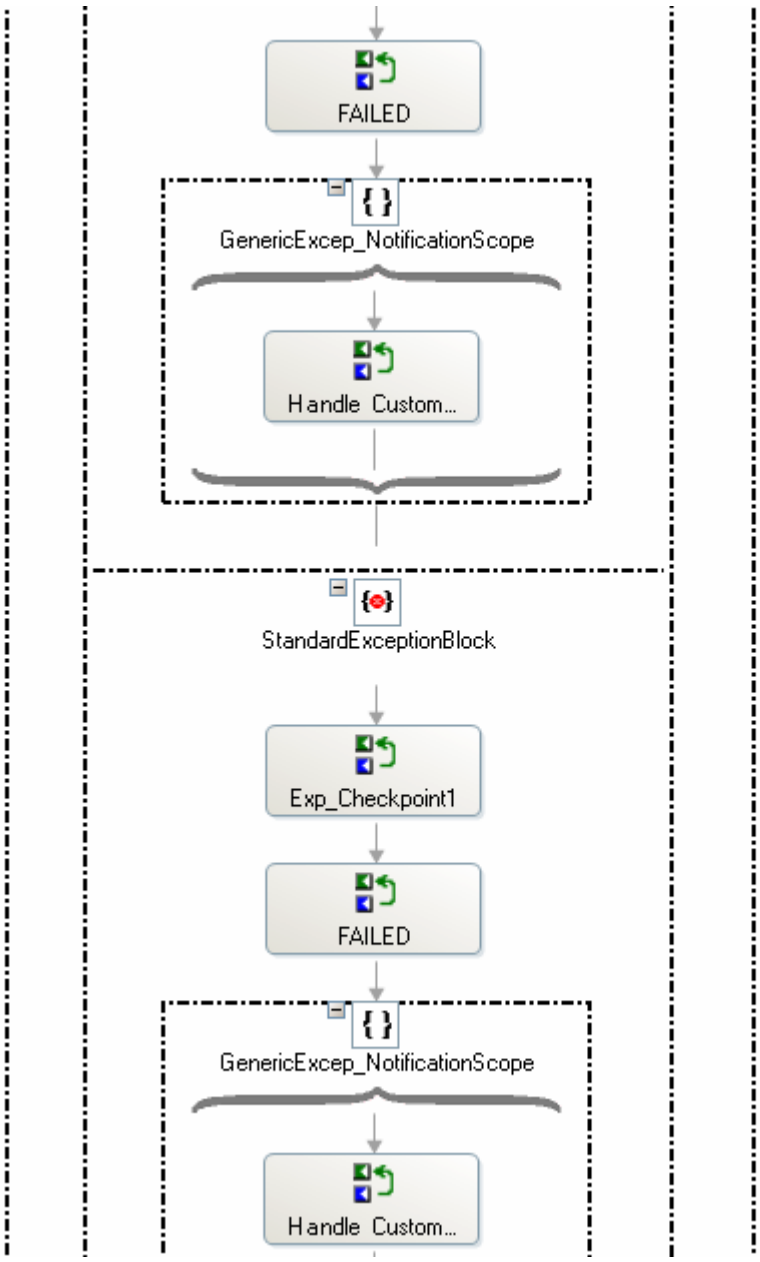
22.6.1.2 Messages defined within this orchestration

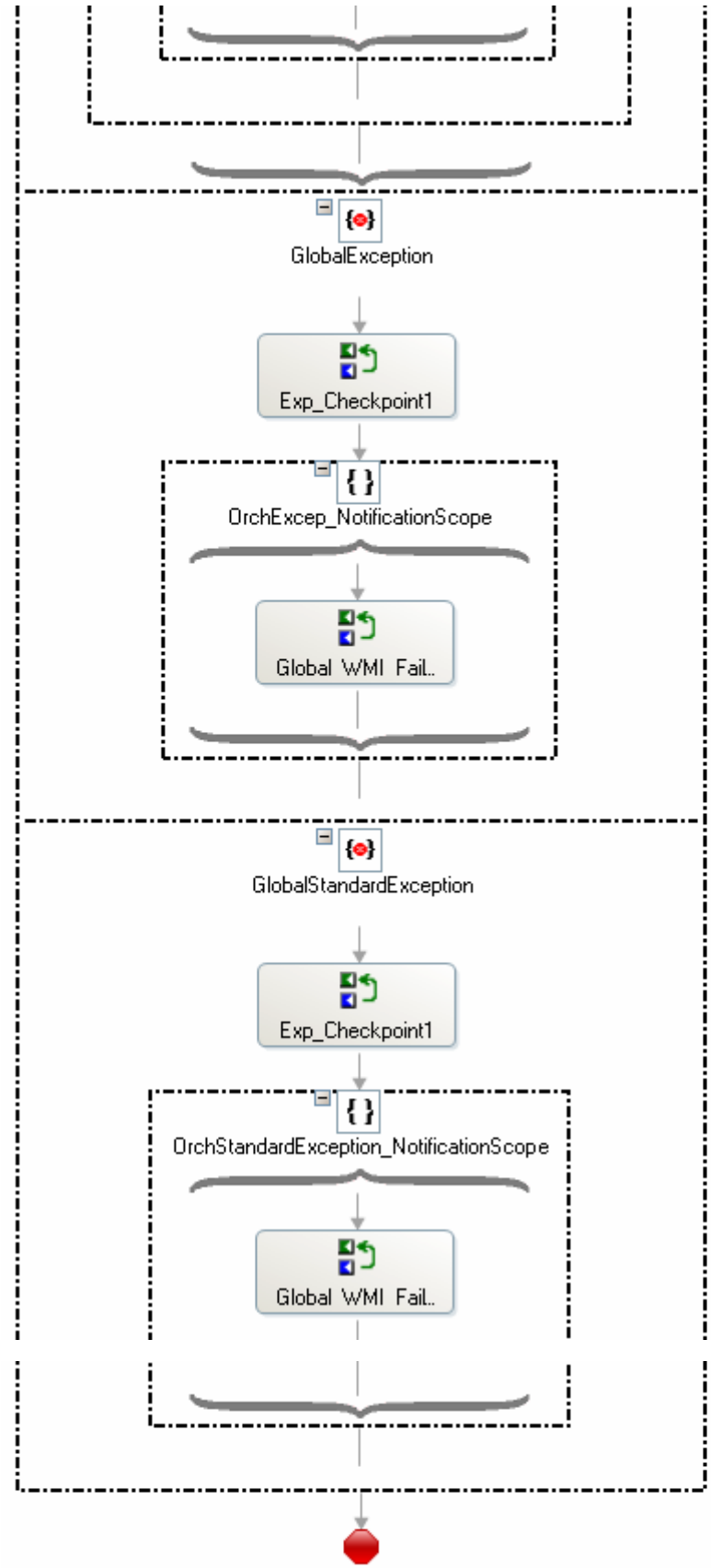
MessageClaimList- EDS.USGS.MED.EDI.GenericHealthCareClaimListMessaging.ClaimList

MessageXdataNSFTransport-
EDS.USGS.MED.EDI.GenericNSFTransportationEncounter.TransportationEncounter


ScopeOrchestration
ReceiveXdataN...
Exp_Checkpoint1
ReadFileName
Exp_Checkpoint1
ScopeProcess
Exp_Checkpoint1
BuildClaimList
XdataNSFtoClai...


PopulateMsgClai...
Exp_Checkpoint1
AssignSendPort
SendClaimList
Exp_Checkpoint1
PROCESSED_B...
GenericExceptionBlock
Exp_Checkpoint1


FAILED
GenericExcep_NotificationScope
Handle Custom...
StandardExceptionBlock
Exp_Checkpoint1
FAILED
GenericExcep_NotificationScope
Handle Custom...


GlobalException
Exp_Checkpoint1
OrchExcep_NotificationScope
Global WMI Fail..
GlobalStandardException
Exp_Checkpoint1
OrchStandardException_NotificationScope
Global WMI Fail..

22.7 Pipeline Configuration

22.7.1 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

22.7.1.1 Pipeline Process Flow

86. Pre-Assemble

87. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

88. Encode

22.7.1.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

22.7.1.3 Receive locations using this pipeline

There are no receive locations using this pipeline

22.7.2 EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterMessaging.RecvNSFTranspotEncounterPipeline

Pipeline Type: Receive

Assembly Name:
EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterMessaging.RecvNSFTranspotEncounterPipeline,
EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterMessaging,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=a2fc6bba136a8ae3

22.7.2.1 Pipeline Process Flow

89. Decode

90. Disassemble

    - Flat file disassembler

    - Microsoft.BizTalk.Component.FFDasmComp,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

91. Validate

92. ResolveParty

22.7.2.2 Send ports using this pipeline

There are no send ports using this pipeline

22.7.2.3 Receive locations using this pipeline

RLNSFTransportEnctrGeneric

23 Appendix U – Biz Talk Documentation – 835 Dental Outbound

23.1 Host Configuration

23.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

23.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

23.1.1.2 Receive locations running in this host

There are no receive locations running in this host

23.1.2 BizTalkServer835277UOutboundProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

23.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess

23.1.2.2 Receive locations running in this host

There are no receive locations running in this host

23.1.3 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

23.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

23.1.3.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

23.1.4 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

23.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

23.1.4.2 Receive locations running in this host

There are no receive locations running in this host

23.2 Send Port Configuration

23.2.1 SendXDATA835

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: True

Two-Way: False

23.2.1.1 Primary Transport

Not yet specified

23.2.1.2 Secondary Transport

Not yet specified

23.2.1.3 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess

23.3 Receive Port Configuration

23.3.1 ReceiveRemittaceAdvice

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

23.3.1.1 Receive Locations

93. RLRemittaceAdviceGeneric

Property	Value
Address	\\10.40.14.111\fts\Generic\RemittaceAdvice835\Outbound\Preprocessor*.*
Transport Protocol	FILE
Receive Pipeline	GenericOutboundRemittaceAdvice835Messaging.RcvRemittanceAdvicePipeLine
Receive Handler	BizTalkServerReceiveHost

23.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess

23.4 Protocol Configuration

23.4.1 FILE

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

23.4.1.1 Send ports using this protocol

There are no send ports using this protocol

23.4.1.2 Receive ports using this protocol

ReceiveRemittaceAdvice

23.5 Assembly Configuration

23.5.1 EDS.USGS.MED.EDI.EdifecsPropertySchema

Name: EDS.USGS.MED.EDI.EdifecsPropertySchema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

23.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.EdifecsPropertySchemaSolution.PromotedPropertySchema

23.5.2 EDS.USGS.MED.EDI.Edifecs835Schema

Name: EDS.USGS.MED.EDI.Edifecs835Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

23.5.2.1 Schema contained in this assembly

EDS.USGS.MED.EDI.Edifecs835Schema._835_W1_IG

23.5.3 EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Messaging

Name: EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Messaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

23.5.3.1 Maps contained in this assembly

GenericOutboundRemittaceAdvice835Messaging.OutboundRemittaceAdvice_To_Edifecs835Xml

23.5.3.2 Schema contained in this assembly

GenericOutboundRemittaceAdvice835Messaging.Generic835RemittanceAdvice

23.5.4 EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes

Name: EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

23.5.4.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess

23.6 Orchestration Configuration

23.6.1 EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServer835277UOutboundProcessHost

23.6.1.1 Ports contained within this orchestration

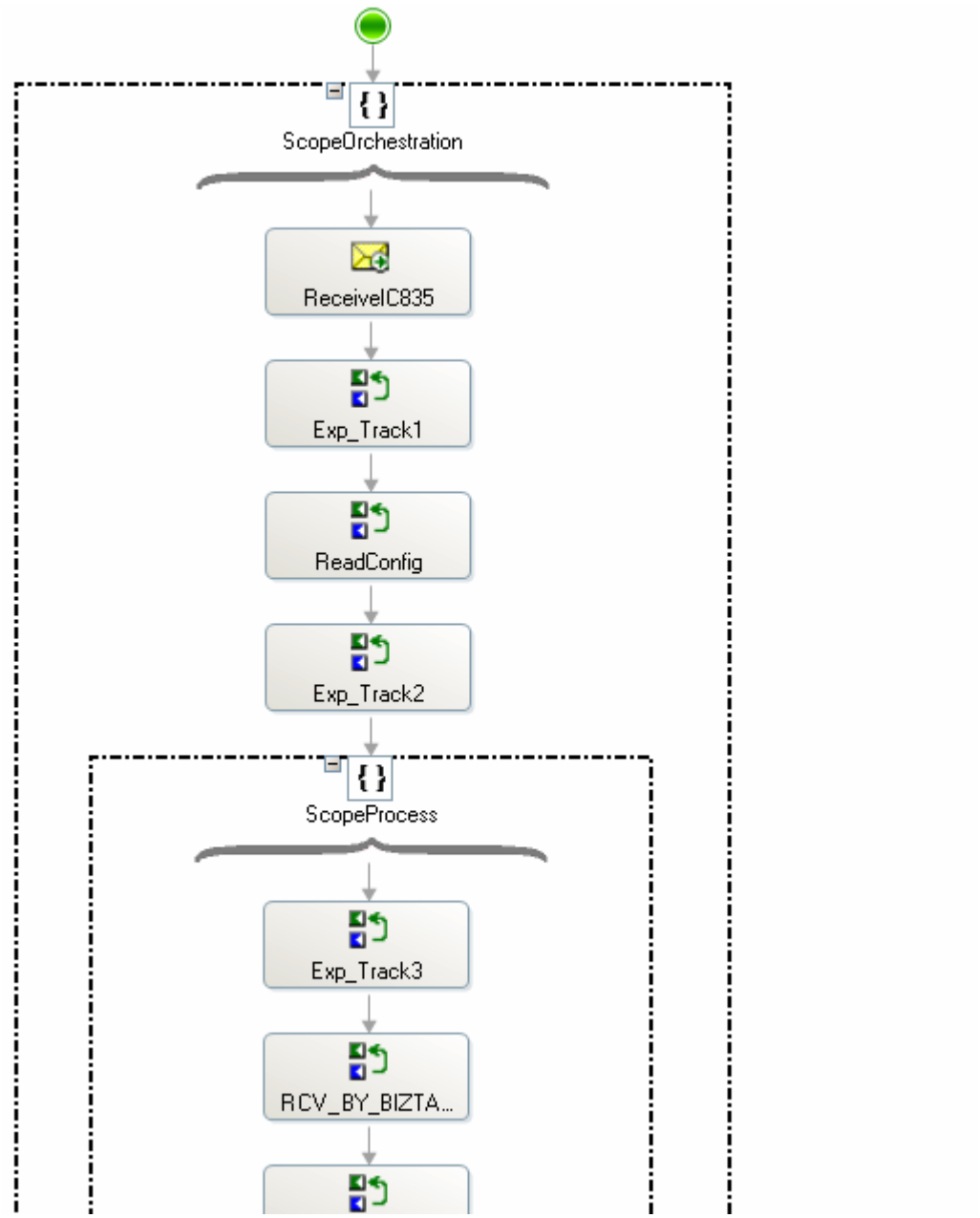
prtOutboundRemittanceAdvice835 - Bound to: ReceiveRemittaceAdvice

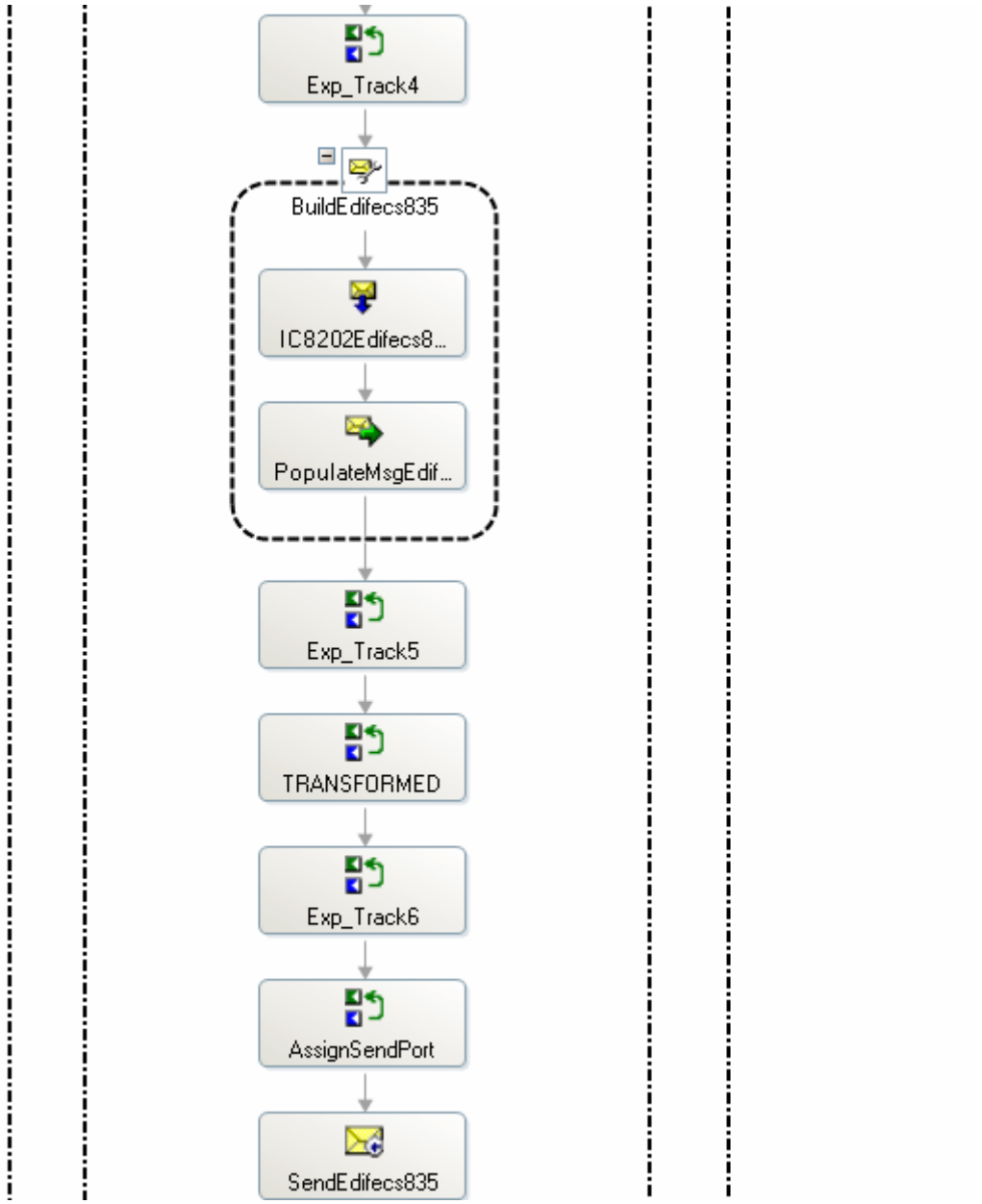
prtOutboundRemittanceAdvice - Bound to: SendXDATA835

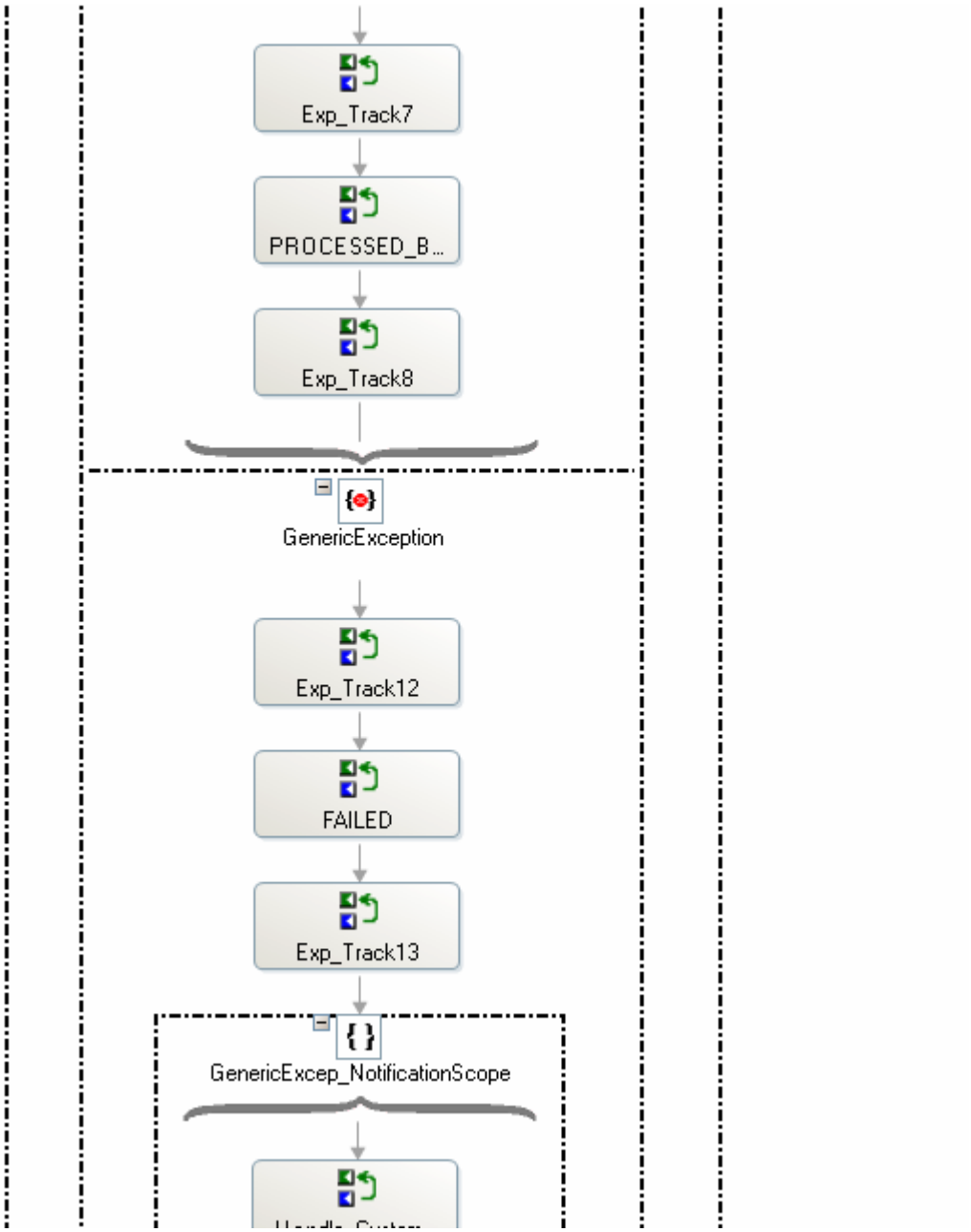
23.6.1.2 Messages defined within this orchestration

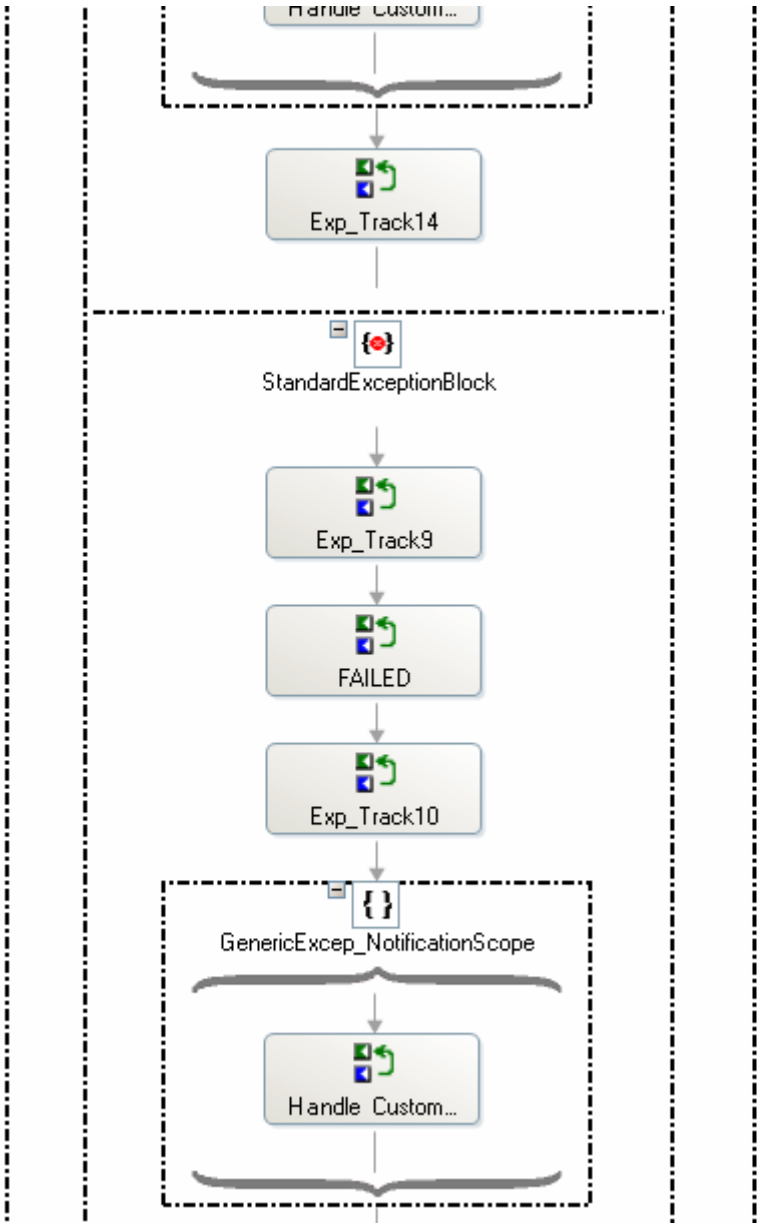
MsgIC835FFOutbound-
GenericOutboundRemittaceAdvice835Messaging.Generic835RemittanceAdvice

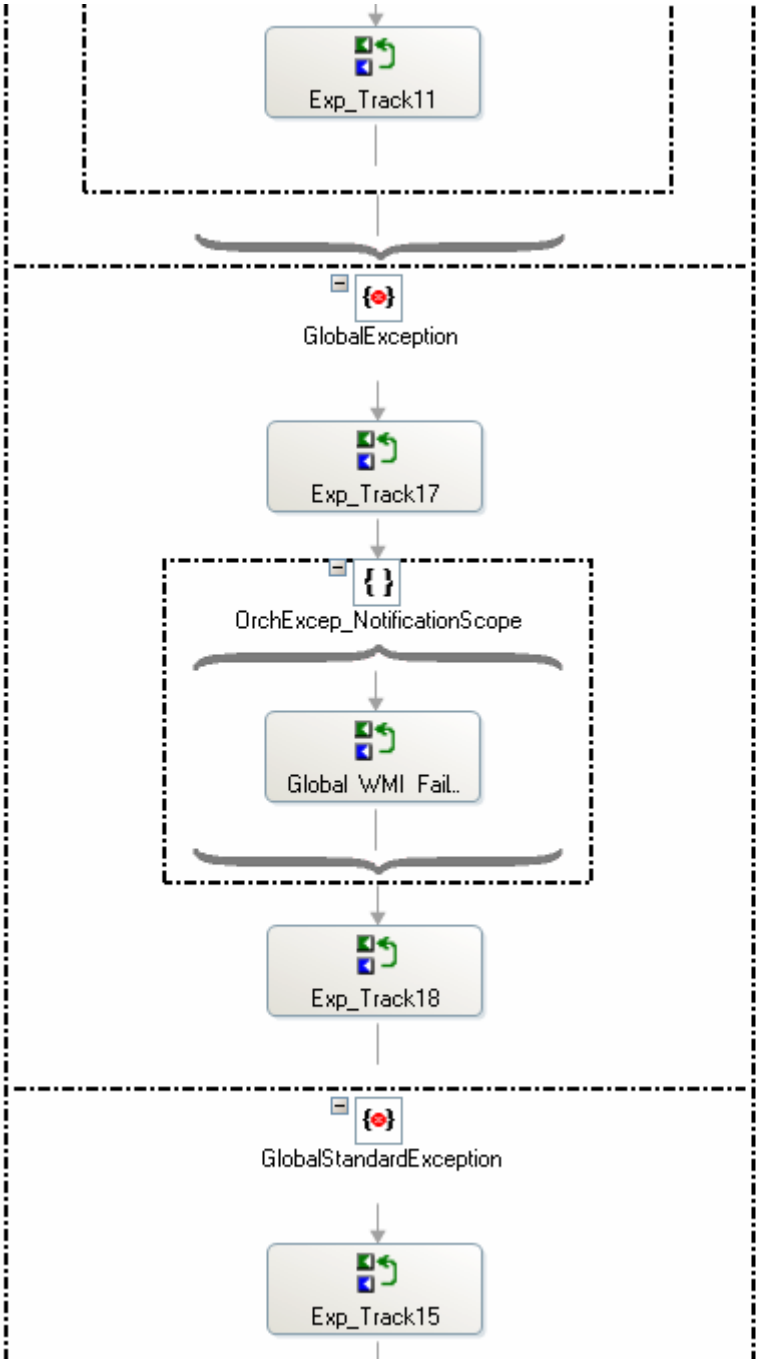
MsgEDIFECS835XMLOutbound- EDS.USGS.MED.EDI.Edifecs835Schema._835_W1_IG

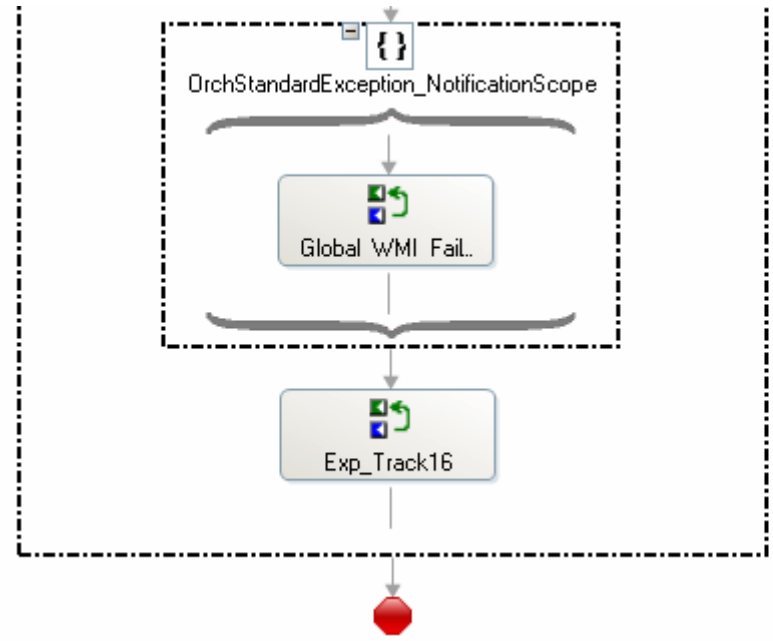

ScopeOrchestration
ReceiveIC835
Exp_Track1
ReadConfig
Exp_Track2
ScopeProcess
Exp_Track3
RCV_BY_BIZTA...


Exp_Track4
BuildEdifecs835
IC8202Edifecs8...
PopulateMsgEdif...
Exp_Track5
TRANSFORMED
Exp_Track6
AssignSendPort
SendEdifecs835


Exp_Track7
PROCESSED_B...
Exp_Track8
GenericException
Exp_Track12
FAILED
Exp_Track13
GenericExcep_NotificationScope


Exp_Track14
StandardExceptionBlock
Exp_Track9
FAILED
Exp_Track10
GenericExcep_NotificationScope
Handle Custom...


Exp_Track11
GlobalException
Exp_Track17
OrchExcep_NotificationScope
Global WMI Fail..
Exp_Track18
GlobalStandardException
Exp_Track15


OrchStandardException_NotificationScope
Global WMI Fail..
Exp_Track16

23.7 Pipeline Configuration

23.7.1 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

23.7.1.1 Pipeline Process Flow

94. Pre-Assemble

95. Assemble

    - XML Assembler
    - Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

96. Encode

23.7.1.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

23.7.1.3 Receive locations using this pipeline

There are no receive locations using this pipeline

23.7.2 GenericOutboundRemittaceAdvice835Messaging.RcvRemittanceAdvicePipeLine

Pipeline Type: Receive

Assembly Name:
GenericOutboundRemittaceAdvice835Messaging.RcvRemittanceAdvicePipeLine, EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Messaging, Version=1.0.0.0, Culture=neutral, PublicKeyToken=a2fc6bba136a8ae3

23.7.2.1 Pipeline Process Flow

97. Decode

98. Disassemble

    - Flat file disassembler
    - Microsoft.BizTalk.Component.FFDasmComp,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

99. Validate

100. ResolveParty

23.7.2.2 Send ports using this pipeline

There are no send ports using this pipeline

23.7.2.3 Receive locations using this pipeline

RLRemittaceAdviceGeneric

24 Appendix V – Biz Talk Documentation – Prior Auth 278 SHPS Inbound

24.1 Host Configuration

24.1.1 BizTalkServerApplication

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: True

Host Type: In-Process

24.1.1.1 Orchestrations running in this host

There are no orchestrations running in this host

24.1.1.2 Receive locations running in this host

There are no receive locations running in this host

24.1.2 BizTalkServerProcessHost

Group Name: KYADMHG\BizTalk Host Users

Host Tracking Enabled: True

Authentication Trusted: True

Default Host: False

Host Type: In-Process

24.1.2.1 Orchestrations running in this host

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess

EDS.USGS.MED.EDI.GenericInboundClaimStatus276Processes.VerifyClaimStatusRequestProcess

EDS.USGS.MED.EDI.GenericOutboundClaimStatus277Processes.VerifyClaimStatusResponseProcess

EDS.USGS.MED.EDI.GenericInboundEligibilityVerification270Processes.VerifyEligibilityRequestProcess

EDS.USGS.MED.EDI.GenericOutboundEligibilityVerification271Processes.VerifyEligibilityResponseProcess

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType

24.1.2.2 Receive locations running in this host

There are no receive locations running in this host

24.1.3 BizTalkServerReceiveHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

24.1.3.1 Orchestrations running in this host

There are no orchestrations running in this host

24.1.3.2 Receive locations running in this host

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLNSFTransportEnctrGeneric

RLPremiumPayGeneric

RLRemittaceAdviceGeneric

RLRemittaceAdvice277UGeneric

RLRemittaceAdvice277UPassPort

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

RLBenefitEnrollGeneric

24.1.4 BizTalkServerSendHost

Group Name: KYADMHG\BizTalk Application Users

Host Tracking Enabled: False

Authentication Trusted: True

Default Host: False

Host Type: In-Process

24.1.4.1 Orchestrations running in this host

There are no orchestrations running in this host

24.1.4.2 Receive locations running in this host

There are no receive locations running in this host

24.2 Send Port Configuration

24.2.1 SendICRealTimeSHPS

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: True

24.2.1.1 Primary Transport

Property	Value
Address	Socket://kysun1ap/31600
Type	SOCKET
Ordered Delivery	False
Retry Count	0
Retry Interval	5

24.2.1.2 Service Window

Enabled: False

Start Date: 5/20/2007 1:00:00 AM

End Date: 5/20/2007 12:59:00 AM

24.2.1.3 Secondary Transport

Not yet specified

24.2.1.4 Orchestrations bound to this send port

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

24.2.2 SHPSWSRequestResponse

Priority: 5

Tracking Type: Not specified

Send Pipeline: Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Dynamic: False

Two-Way: True

24.2.2.1 Primary Transport

Property	Value
Address	http://10.40.14.110/PAInterchangeToSHPSProcessStatusDBUpdate/PAInterchangeToSHPSProcessStatusDBUpdate.asmx

Type	SOAP
Ordered Delivery	False
Retry Count	0
Retry Interval	5

24.2.2.2 Service Window
Enabled: False

Start Date: 5/19/2007 1:00:00 AM

End Date: 5/19/2007 12:59:59 AM

24.2.2.3 Secondary Transport
Not yet specified

24.2.2.4 Orchestrations bound to this send port
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

24.2.3 EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess_prtEMAILSHPSTheFileViaSMTP_a2fc6bba136a8ae3

Priority: 5

Tracking Type: Not specified

Send Pipeline:
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PipelineSendEmailNotificationForBatchFailure

Dynamic: True

Two-Way: False

24.2.3.1 Primary Transport
Not yet specified

24.2.3.2 Secondary Transport
Not yet specified

24.2.3.3 Orchestrations bound to this send port
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

24.3 Receive Port Configuration

24.3.1 ReceiveSHPS

Authentication Type: NotRequired

Tracking Type: Not specified

Send Pipeline: Not specified

Two-Way: False

24.3.1.1 Receive Locations

101. RLSHPS

Property	Value
Address	\\10.40.14.111\fts\Generic\SHPSPA\Data*.xml
Transport Protocol	FILE
Receive Pipeline	Microsoft.BizTalk.DefaultPipelines.XMLReceive
Receive Handler	BizTalkServerReceiveHost

24.3.1.2 Orchestrations bound to this receive port

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

24.4 Protocol Configuration

24.4.1 SOCKET

Delete protected: False

Uses adapter framework UI for receive handler configuration: True

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: True

Uses adapter framework UI for send port configuration: True

Starts when the service starts: False

Receive handler of adapter is hosted in-process: False

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: True

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: True

24.4.1.1 Send ports using this protocol

SendICRealTimeSHPS

24.4.1.2 Receive ports using this protocol

There are no receive ports using this protocol

24.4.2 SOAP

Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: False

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: True

Supports send operations: True

Supports the SOAP protocol: True

Supports solicit-response operations: True

24.4.2.1 Send ports using this protocol
SHPSWSRequestResponse

24.4.2.2 Receive ports using this protocol
There are no receive ports using this protocol

24.4.3 FILE
Delete protected: False

Uses adapter framework UI for receive handler configuration: False

Uses adapter framework UI for send handler configuration: False

Uses adapter framework UI for receive location configuration: False

Uses adapter framework UI for send port configuration: False

Starts when the service starts: False

Receive handler of adapter is hosted in-process: True

Requires a single instance per server: False

Supports static handlers: False

Supports ordered delivery: False

Supports receive operations: True

Supports request-response operations: False

Supports send operations: True

Supports the SOAP protocol: False

Supports solicit-response operations: False

24.4.3.1 Send ports using this protocol
There are no send ports using this protocol

24.4.3.2 Receive ports using this protocol
ReceiveSHPS

24.5 Assembly Configuration

24.5.1 EDS.USGS.MED.EDI.GenericPriorAuth278Schema

Name: EDS.USGS.MED.EDI.GenericPriorAuth278Schema

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

24.5.1.1 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericPriorAuth278Schema.PriorAuth278

24.5.2 EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging

Name: EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

24.5.2.1 Maps contained in this assembly

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.SHPSSingleRequestXML_To_PriorAuthXMLType

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.SHPSDBUpdateSchema_To_WSClassSchema

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.ICResponse_To_SHPSDBUpdateType

24.5.2.2 Schema contained in this assembly

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS_SingleType

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.SHPSDBUpdate.Reference

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS_DBUpdate_SingleType

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS_DBUpdateType

24.5.3 EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess

Name: EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess

Version: 1.0.0.0

Culture: neutral

Public Key Token: a2fc6bba136a8ae3

24.5.3.1 Orchestrations contained in this assembly

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

24.6 Orchestration Configuration

24.6.1 EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess

Parent Assembly:
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess,Version=1.0.0.0,Culture=neutral,PublicKeyToken=a2fc6bba136a8ae3

Host: BizTalkServerProcessHost

24.6.1.1 Ports contained within this orchestration

prtReceive278SHPS_Send278 - Bound to: ReceiveSHPS

prtSendPARequest_ReceivePAResponse - Bound to: SendICRealTimeSHPS

prtSHPDDBUpdate - Bound to: SHPSWSRequestResponse

prtEMAILSHPSTheFileViaSMTP - Bound to:
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess_prtEMAILSHPSTheFileViaSMTP_a2fc6bba136a8ae3

24.6.1.2 Messages defined within this orchestration

MsgEmailNotificationToSHPS1-
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS

WSResponse –
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.SHPSDBUpdate.PAInterchangeToSHPSProcessStatusDBUpdate

WSRequest –
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.SHPSDBUpdate.PAInterchangeToSHPSProcessStatusDBUpdate

MsgInterchangeResponse- EDS.USGS.MED.EDI.GenericPriorAuth278Schema.PriorAuth278

MsgInterchangeRequest- EDS.USGS.MED.EDI.GenericPriorAuth278Schema.PriorAuth278

MsgSHPSIN- EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS

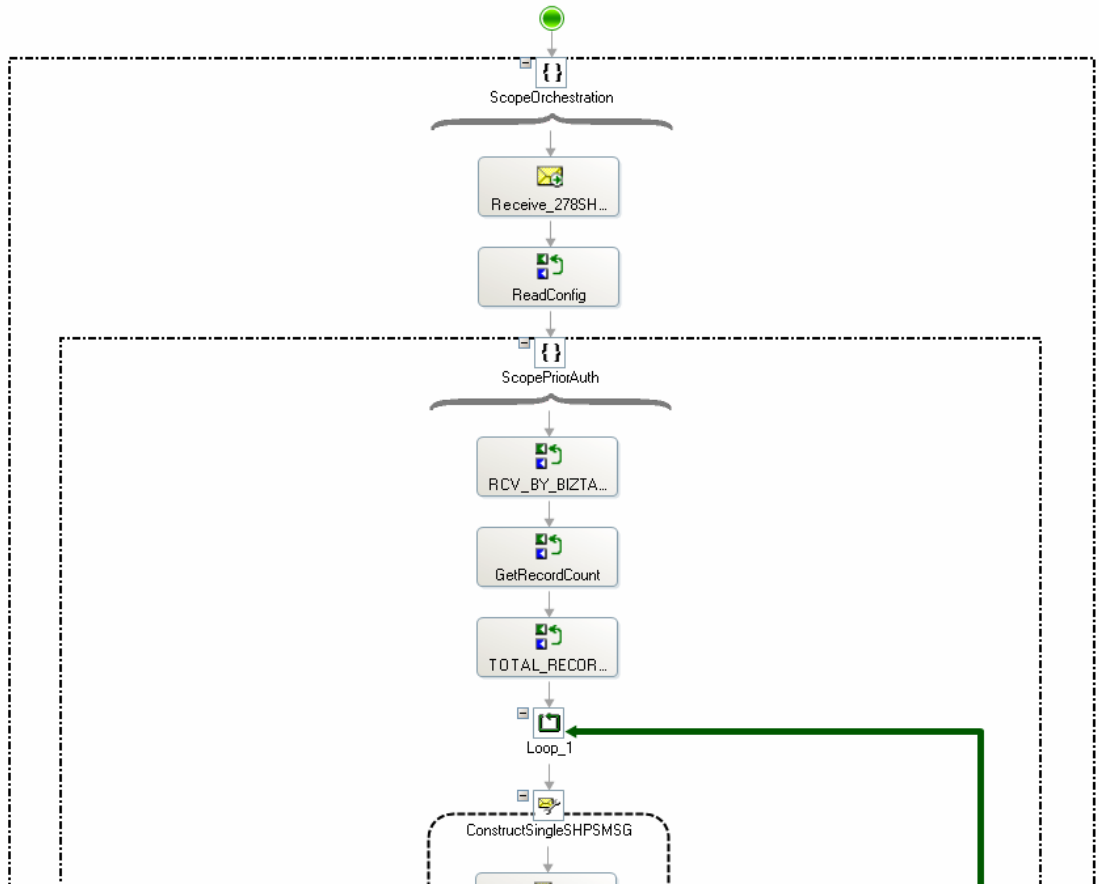
MsgSHPSSingleIn-
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS_SingleType

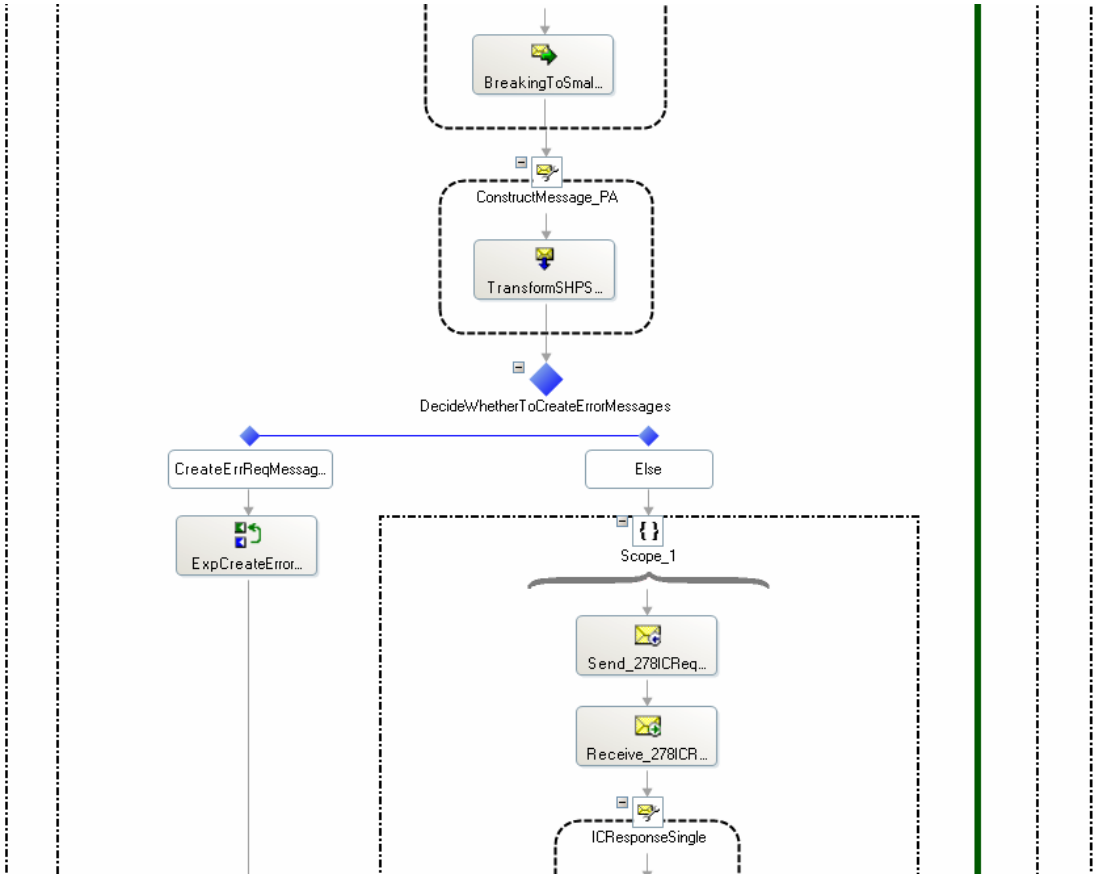
MsgSHPSRequestDB-
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS_DBUpdateType

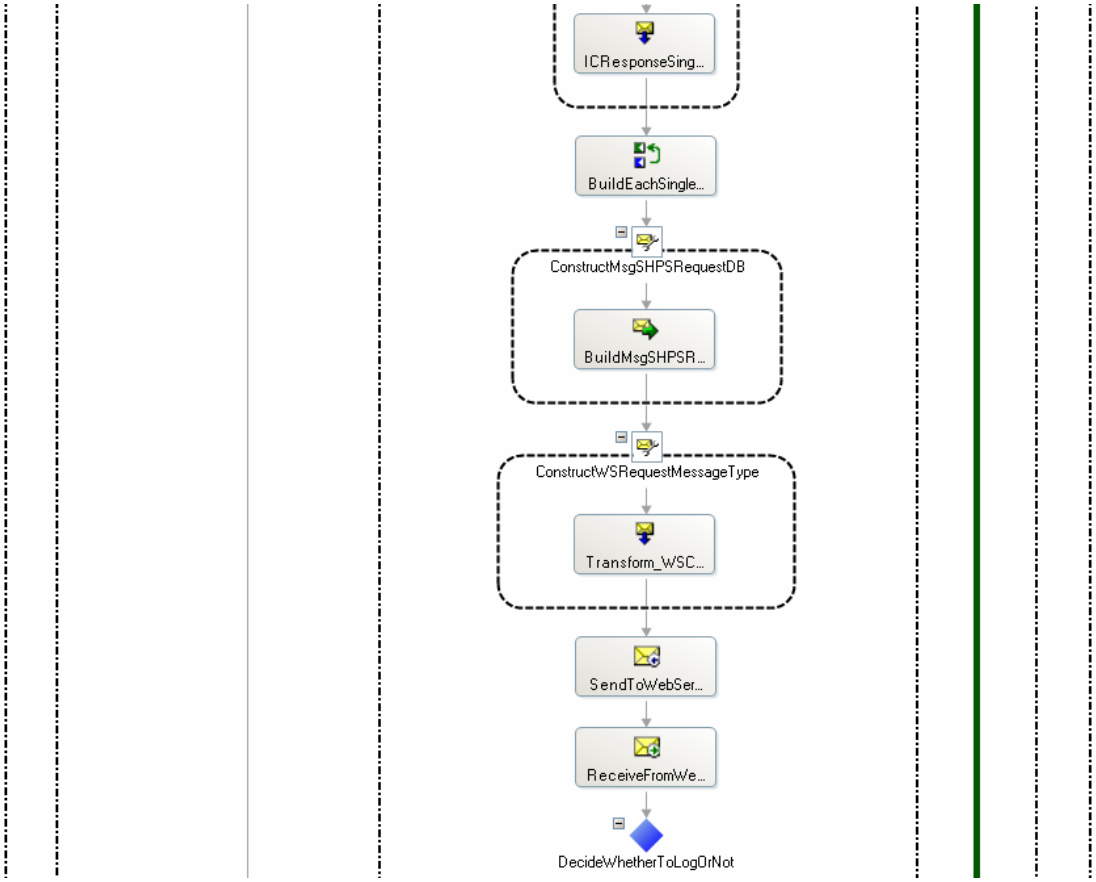
MsgICResponseSingle-
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS_DBUpdate_SingleType

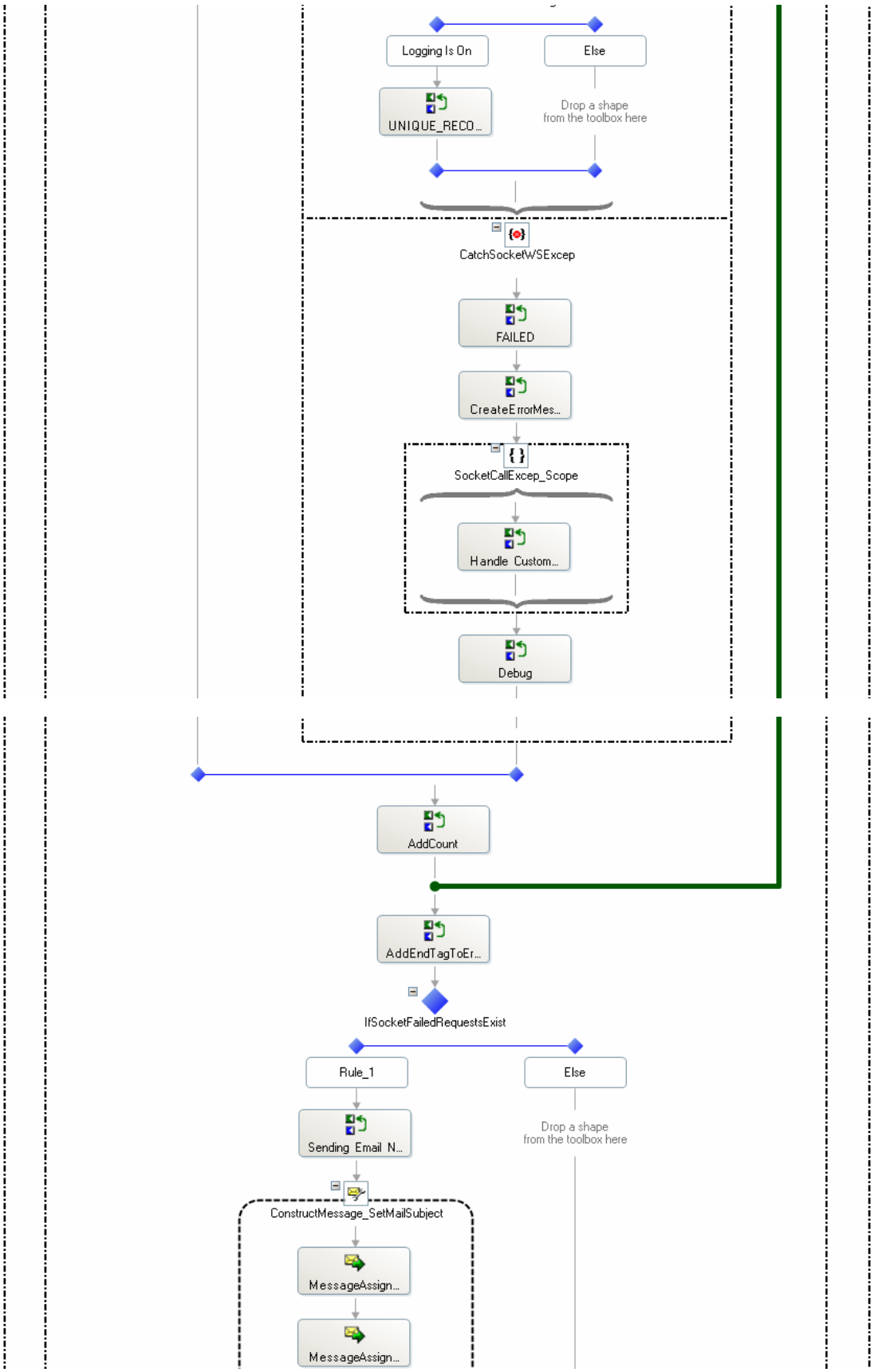
MsgEmailNotificationToSHPS-
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS

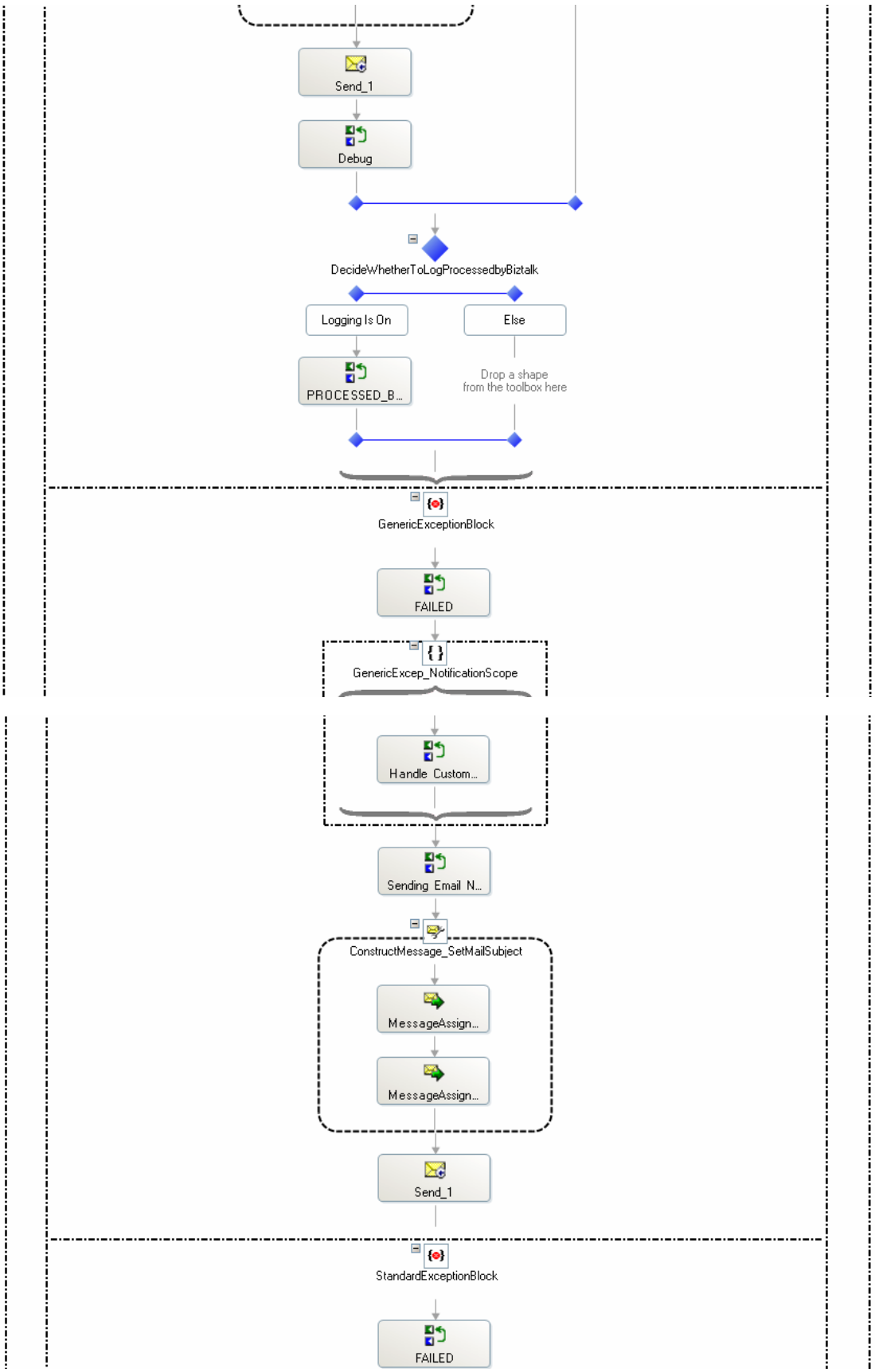
MsgEmailNotificationToSHPS2-
EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PA_SHPS

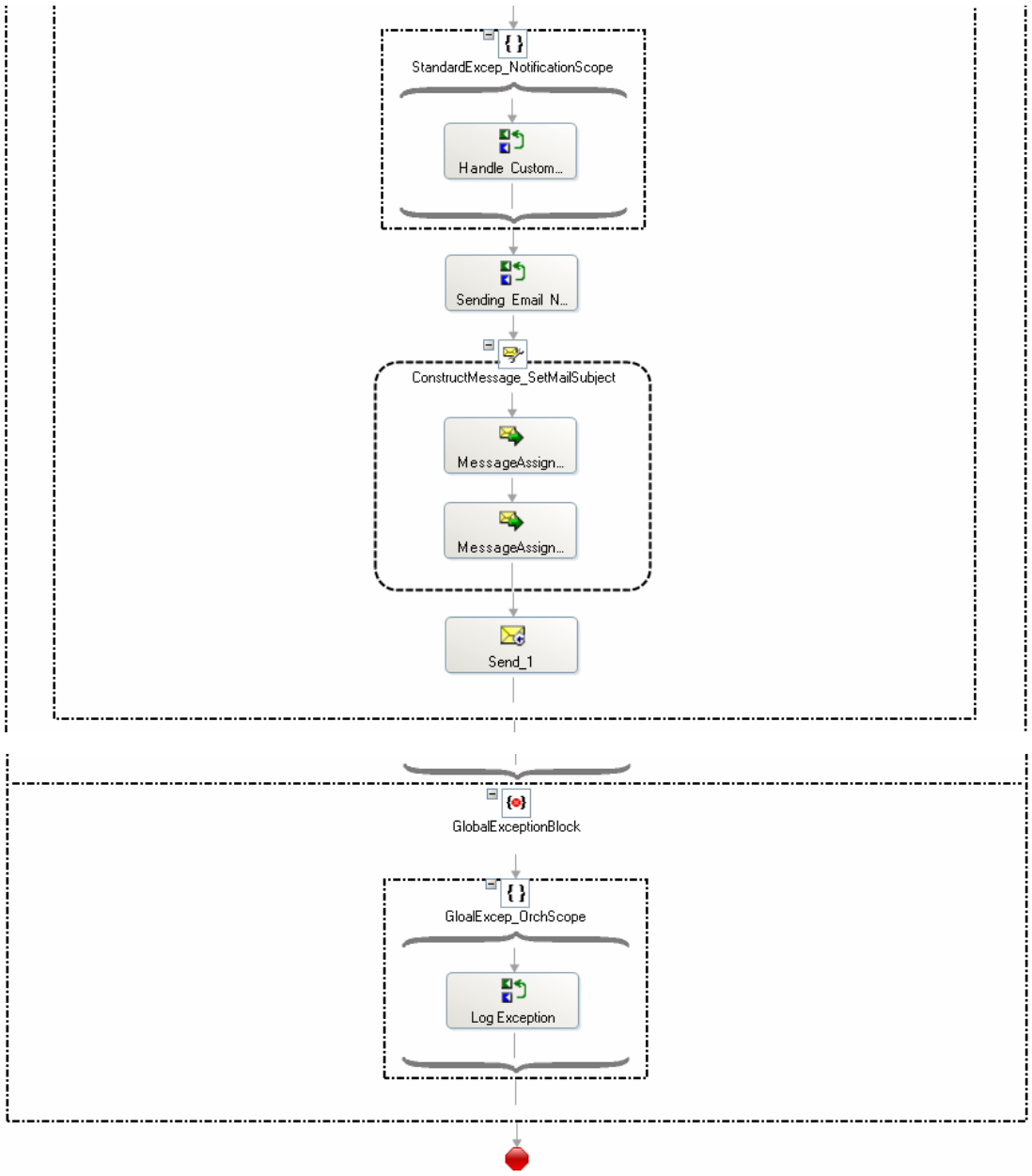



BreakingToSmal...
ConstructMessage_PA
TransformSHPS...
DecideWhetherToCreateErrorMessages
CreateErrReqMessag...
Else
ExpCreateError...
Scope_1
Send_278ICReq...
Receive_278ICR...
ICResponseSingle


ICResponseSing...
BuildEachSingle...
ConstructMsgSHPSRequestDB
BuildMsgSHPSR...
ConstructWSRequestMessageType
Transform_WSC...
SendToWebSer...
ReceiveFromWe...
DecideWhetherToLogOrNot


Logging Is On
Else
UNIQUE_RECO...
Drop a shape
from the toolbox here
CatchSocketWSExcep
FAILED
CreateErrorMes...
SocketCallExcep_Scope
Handle Custom...
Debug
AddCount
AddEndTagToEr...
IfSocketFailedRequestsExist
Rule_1
Else
Sending Email N...
Drop a shape
from the toolbox here
ConstructMessage_SetMailSubject
MessageAssign...
MessageAssign...


Send_1
Debug
DecideWhetherToLogProcessedbyBiztalk
Logging Is On
Else
PROCESSED_B...
Drop a shape
from the toolbox here
GenericExceptionBlock
FAILED
GenericExcep_NotificationScope
Handle Custom...
Sending Email N...
ConstructMessage_SetMailSubject
MessageAssign...
MessageAssign...
Send_1
StandardExceptionBlock
FAILED


StandardExcep_NotificationScope
Handle Custom...
Sending Email N...
ConstructMessage_SetMailSubject
MessageAssign...
MessageAssign...
Send_1
GlobalExceptionBlock
GloalExcep_OrchScope
Log Exception

24.7 Pipeline Configuration

24.7.1 Microsoft.BizTalk.DefaultPipelines.XMLReceive

Pipeline Type: Receive

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

24.7.1.1 Pipeline Process Flow

102. Decode

103. Disassemble

- XML Disassembler
- Microsoft.BizTalk.Component.XmlDasmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

104. Validate

105. ResolveParty

- Party Resolution
- Microsoft.BizTalk.Component.PartyRes, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

24.7.1.2 Send ports using this pipeline

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

24.7.1.3 Receive locations using this pipeline

RLClaim837DXDataGeneric

RLInbound837IClaimGeneric

RLClaim837PXDataGeneric

RLClaimStatusReqGeneric

RLICClaimStatusRespGeneric

RLEligVerificationReqGeneric

RLICEligVerifRespGeneric

RLSHPS

RL278PAFromInternet

24.7.2 Microsoft.BizTalk.DefaultPipelines.XMLTransmit

Pipeline Type: Send

Assembly Name: Microsoft.BizTalk.DefaultPipelines.XMLTransmit, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

24.7.2.1 Pipeline Process Flow

106. Pre-Assemble

107. Assemble

- XML Assembler
- Microsoft.BizTalk.Component.XmlAsmComp, Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

108. Encode

24.7.2.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837DProcesses.InboundHealthCareClaimDentalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837D

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837IProcesses.InboundHealthCareClaimInstitutionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837I

EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundHealthCareClaim837PProcesses.InboundHealthCareClaimProfessionalProcess_prtSendClaimList_a2fc6bba136a8ae3

SendClaimList837P

EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundNSFTransportationEncounterProcesses.InboundNSFTransportProcess_prtSendClaimList_a2fc6bba136a8ae3

SendNSFTransportationClaimList

EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundBenefitEnrollment834Processes.BenefitEnrollmentProcess_prtSendXML834_a2fc6bba136a8ae3

EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundPremiumPayment820Processes.PremiumPaymentProcess_prtSendXML820_a2fc6bba136a8ae3

SendXDATA820

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice835Processes.OutboundRemittaceAdviceProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA835

EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses_1.0.0.0_EDS.USGS.MED.EDI.GenericOutboundRemittaceAdvice277UProcesses.OutboundRemittaceAdvice277UProcess_prtOutboundRemittanceAdvice_a2fc6bba136a8ae3

SendXDATA277U

SendICClaimStatusReq

SendXDATA277

SendICEligVerificationReq

SendXDATA271

SendICRealTimeSHPS

SHPSWSRequestResponse

Send278InternetToWSSHPSDBInsert

SendXDATA834

24.7.2.3 Receive locations using this pipeline

There are no receive locations using this pipeline

24.7.3 EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PipelineSendEmailNotificationForBatchFailure

Pipeline Type: Send

Assembly Name: EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging.PipelineSendEmailNotificationForBatchFailure, EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSMessaging, Version=1.0.0.0, Culture=neutral, PublicKeyToken=a2fc6bba136a8ae3

24.7.3.1 Pipeline Process Flow

109. Pre-Assemble

110. Assemble

- XML assembler
- Microsoft.BizTalk.Component.XmlAsmComp,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

111. Encode

- MIME/SMIME encoder
- Microsoft.BizTalk.Component.MIME_SMIME_Encoder,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

24.7.3.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess_1.0.0.0_EDS.USGS.MED.EDI.GenericInboundPriorAuth278SHPSProcess.PriorAuth278SHPSProcess_prtEMAILSHPSTheFileViaSMTP_a2fc6bba136a8ae3

24.7.3.3 Receive locations using this pipeline

There are no receive locations using this pipeline

24.7.4 EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertMessaging.SendPipelineInternetToShpsSendEmail

Pipeline Type: Send

Assembly Name:
EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertMessaging.SendPipelineInternetToShpsSendEmail,
EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertMessaging, Version=1.0.0.0, Culture=neutral, PublicKeyToken=a2fc6bba136a8ae3

24.7.4.1 Pipeline Process Flow

112. Pre-Assemble

113. Assemble

- XML assembler
- Microsoft.BizTalk.Component.XmlAsmComp,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

114. Encode

- MIME/SMIME encoder
- Microsoft.BizTalk.Component.MIME_SMIME_Encoder,Microsoft.BizTalk.Pipeline.Components, Version=3.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

24.7.4.2 Send ports using this pipeline

EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess_1.0.0.0_EDS.USGS.MED.EDI.GenericPriorAuthInternetToSHPSDBInsertProcess.PriorAuthInternetToSHPSDBInsertProcessType_prtInternetPAToSHPSDBInsert_a2fc6bba136a8ae3

24.7.4.3 Receive locations using this pipeline

There are no receive locations using this pipeline

25 Appendix W – Pre-Processor Service Documentation_V1

25.1 Introduction

Pre-Processor Service is a windows service component which is used to process and prepare files before passing to EDS ODC Batch BizTalk Servers Some of the main functions performed by Pre-Processor Service before it passes the file to BizTalk Servers are document balancing, trading partner management event tracking and Key Performance Indicator (KPI) tracking For inbound (in respect to MMIS) X12 files, it also performs HIPAA validations by making calls to Edifecs XEngine while for outbound files (in rest respect to MMIS) it splits up large sized files as per their document type so they are easily processed by BizTalk.

25.1.1 Pre-Processor Service Features

25.1.1.1 Trading Partner Management

Pre-Processor does four trading partner validations for all X12 documents against the trading partner configurations set by EDI Help Desk Below are the Trading Partner Management (TPM) validations performed by Pre-Processor Service:

- Check if the document sender (for inbound files) or receiver (for outbound files) is a valid trading partner with active status on the date of transmission.
- Check if the transport type through which file was received is configured for the trading partner.
- Check if the document type is supported in the specific direction (for example: Inbound 837 or Outbound 835)
- Check if the Usage Indicator on the ISA segment matches the Usage Indicator set in trading partner configurations.

25.1.1.2 Document Balancing and Tracking

When the file is received by Pre-processor service it checks if the file integrity is intact by checking its Crc32 code value After that it continues tracking the audit events against the existing File ID of the file Pre-Processor Service also counts the number of transactions in each file received and adds these values to event tracking For inbound X12 files which go through HIPAA validation it takes another count after HIPAA validation and collects the key performance indicator and then adds these values to event tracking.

25.1.1.3 HIPAA Validation and Transformation

All inbound X12 documents are passed to Edifecs XEngine for HIPAA validations This is done by calling Edifecs Wrapper which in turn calls XEngine XEngine is also responsible for transforming native X12 to XDATA (proprietary Edifecs Xml) format Another step that happens in XEngine is that X12 documents are broken into smaller files depending upon the Split Rules set in XEngine configurations Currently for KY MMIS all claims are split at claim level leading to one XDATA file for each claim while all other document types are split at ST-SE transaction level leading to one XDATA file for each ST-SE transaction.

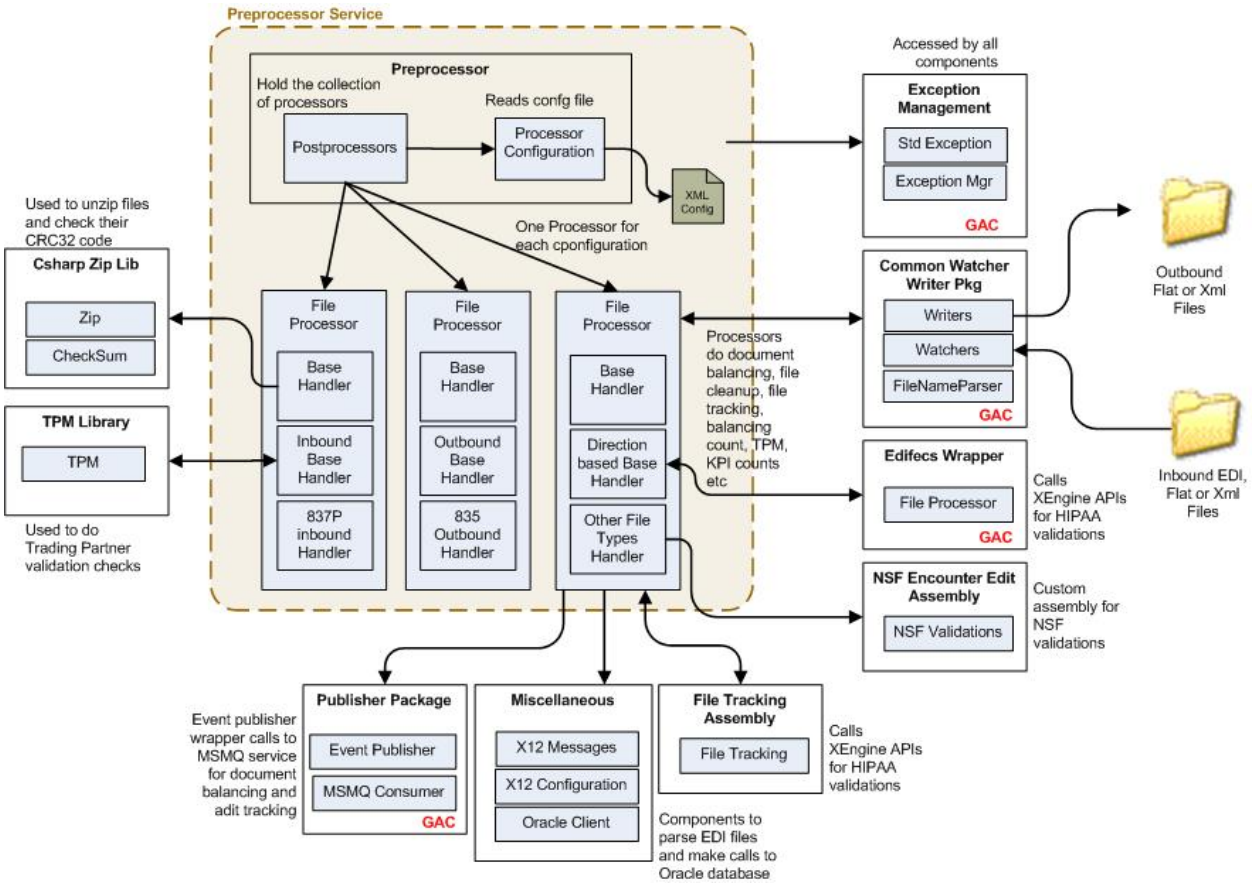
25.1.1.4 File Size Optimization

Pre-Processor Service is designed to optimize the size of large outbound files before these are passed to BizTalk Server If the file size is bigger than the pre-configured size for that document type then the file is split into smaller sized files and renamed with the file part information (for

example: 1 of 10 which would mean part 1 of the file which is split into 10 parts) This leads to faster processing of files in BizTalk server.

25.2 Pre-Processor Service Solution Diagram

Below diagram shows all the components used by Pre-Processor Service and how they interact with each other In later section this document will go over each project related to Pre-Processor Service solution in detail.

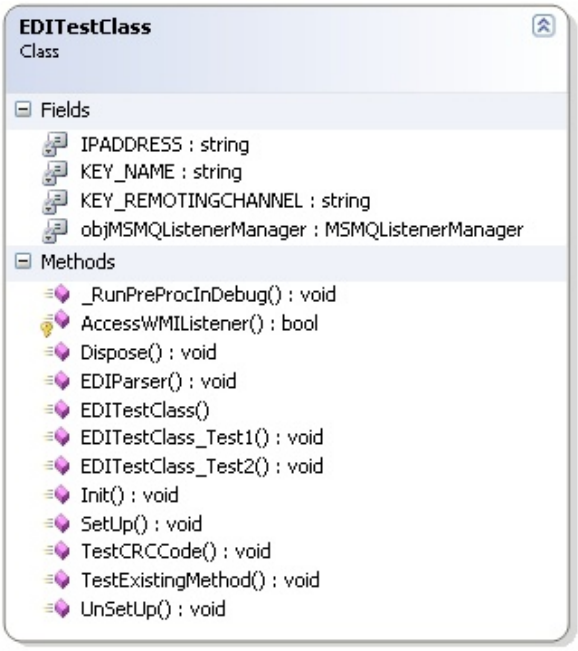


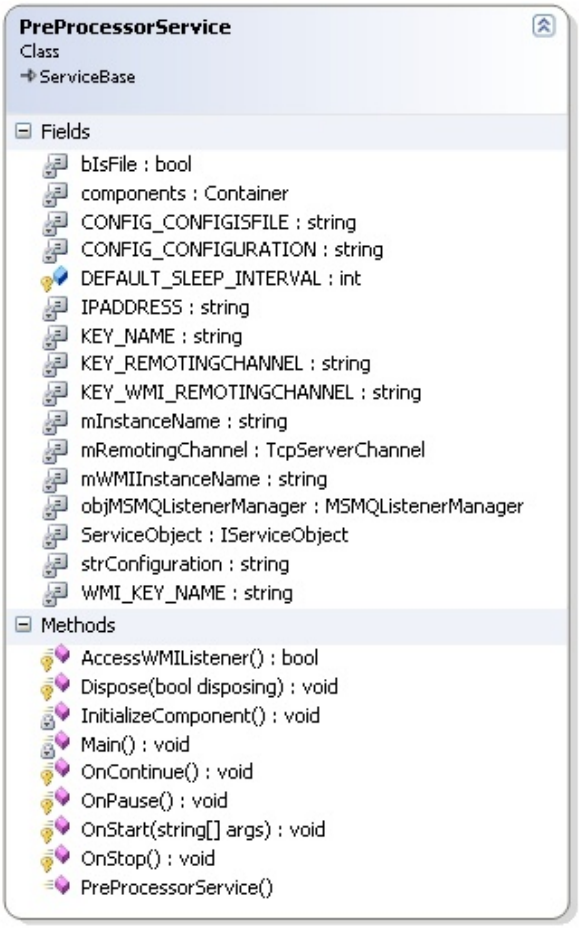
25.3 Project Descriptions

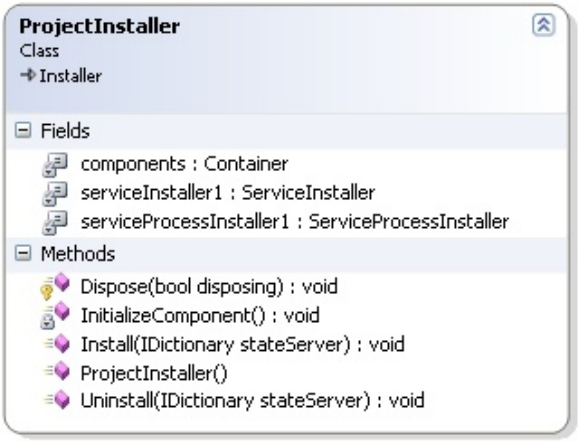
This section will explain all the projects involved in development of the Pre-Processor Service It will also go in to explaining the individual projects, their classes and the methods involved in supporting the required functionality of the Pre-Processor Service.

25.3.1 Pre-Processor Service Project

Pre-Processor Service project is the main project which holds the actual windows service component This windows service component is responsible for calling other assembly projects to perform its designed functions This project also incorporates the test harness class (EDITestClass) which allows the service to be run in a debug mode using NUnit tool.







25.3.1.1 Pre-Processor Service Class

Pre-Processor Service is the file which has the main class, “PreProcessorService”, of the “PreProcessorService” project where the windows service gets the control to Start, Stop, Pause or Restart itself.

File Name: PreProcessorService.cs	**Class Name: PreProcessorService**
Method Name	**Description**
PreProcessorService	This is the default constructor of the class

File Name: PreProcessorService.cs	Class Name: PreProcessorService
Method Name	**Description**
Main	This is the entry point for the service control FTS Service is instantiated in this method and is set to run.
InitializeComponent	This method is called by the constructor and is used to initialize the name of the class.
OnStart	This is the method which is called to start the service This method reads the App.config to get the configuration file and passes it to PreProcessors class of PreProcessor project It also calls AccessWMIListenerService to check if the MSMQ Listener Service is up and running Finally it then calls the Start method of the instantiated PreProcessors class.
OnStop	Before stopping, this method calls the above instantiated PreProcessors object Stop method If successful it stops running the Pre-Processor Service.
AccessWMIListenerService	This method reads the configurations required to remote into MSMQ Listener Service and then remotes into it to check if the service is running and is accepting remote calls.
Dispose	This method disposes off the service object.

25.3.1.2 Project Installer Class

This class is required to create a proper deployment (msi) package for the Pre-Processor service This msi package is used to deploy the service in run time environment.

File Name: ProjectInstaller.cs	Class Name: ProjectInstaller
Method Name	**Description**
ProjectInstaller	This is the default constructor of the class and it calls InitializeComponent.
InitializeComponent	This is the designer generated code and should not be modified by developer using code editor.
Install	This method is called during install process of the windows service deployment package and is responsible for installing the service component This method is also used to setup the install properties of the service like service name, service description and registry settings.
Uninstall	This method is called during uninstall process of the windows service This method is used to remove the service related registry settings and it finally uninstalls the service component.

File Name: ProjectInstaller.cs	**Class Name: ProjectInstaller**
Method Name	**Description**
Dispose	This method disposes off the service object.

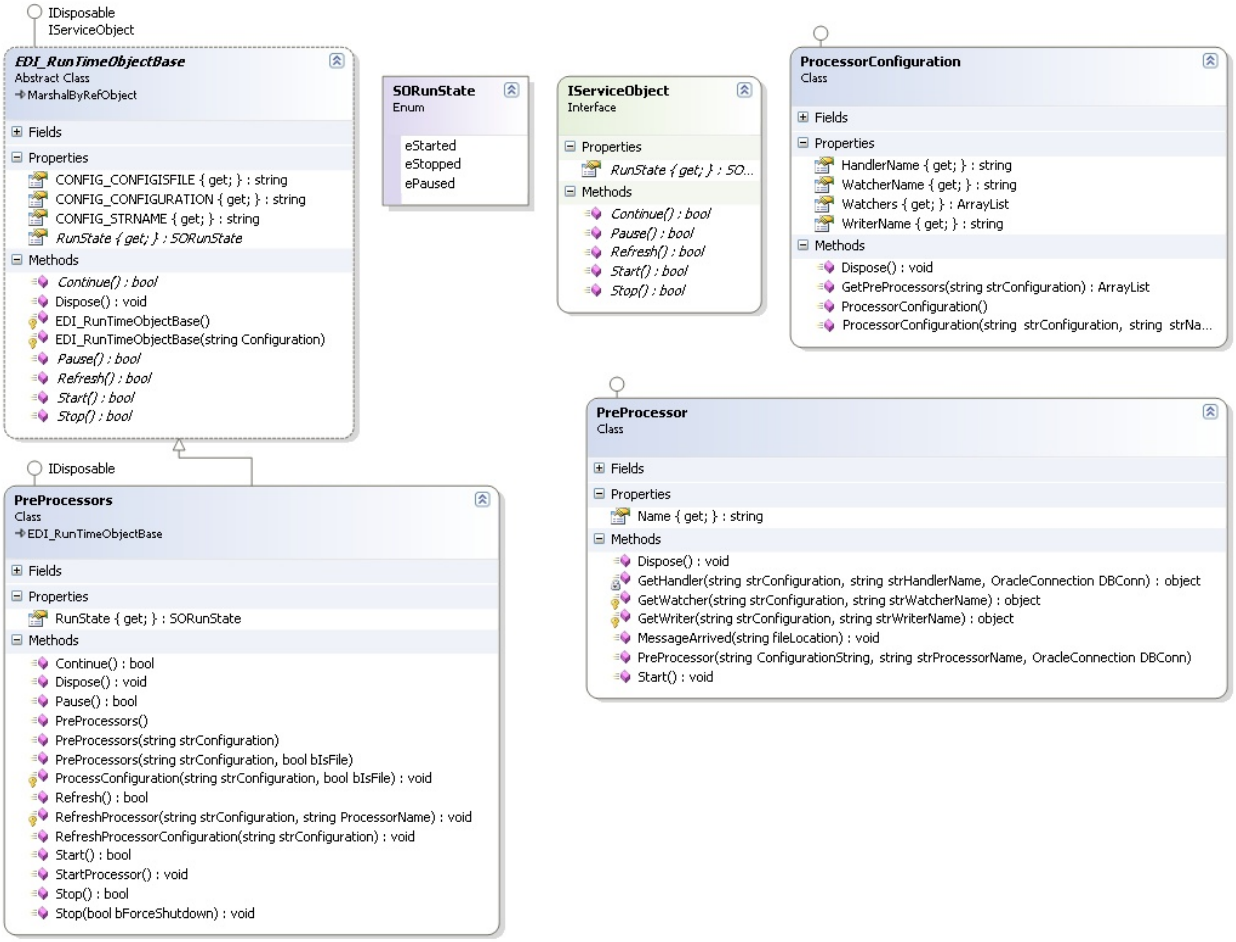
25.3.1.3 EDI Test Class

This class is a Text Fixture class of the NUnit tool and helps the developer to run the windows service in the debug mode.

File Name: EDITestClass.cs	**Class Name: EDITestClass**
Method Name	**Description**
EDITestClass_Test	This is the NUnit test that is run using the NUnit tool to start the Pre-Processor Service in the Debug mode This replaces the need to install the service component It performs the same functionality of getting the configuration file and then it calls the PreProcessors class Start method.

25.3.2 PreProcessor Project

PreProcessor Project is called by the PreProcessorService class and is responsible for setting up the runtime threads and handlers of the service In the PreProcessors project Pre-processor Service accesses the configuration files and reads the configuration attributes needed to initializes PreProcessor handlers for all supported document types After reading the configuration values it starts each Pre-Processor Handler in a separate thread This project is also responsible for cleaning up all the handlers before Pre-Processor Service shuts down.



25.3.2.1 PreProcessors Class

PreProcessors Class implements EDI_RunTimeObjectBase class and IDisposable interface PreProcessors class reads the configuration values from App.config (file name is PreProcessorService.exe.config during runtime), ServiceConfigl.xml files to get configuration values which are common to the entire service Based on those values it creates new Pre-Processor thread for each configuration and starts these threads While starting new thread it passes the value of the directory to monitor and which process handler to use When the PreProcessors class gets a call to stop the Pre-Processor Service then it is also responsible for killing each Pre-Processor handler thread which will finally stop the service.

File Name: PreProcessors.cs	**Class Name: PreProcessors**
Method Name	**Description**
PreProcessors	This is the default constructor of the class This method calls ProcessConfiguration method
Start	This method sets the service state to “Started” and calls StartProcessor method.
StartProcessor	This method goes over each PreProcessor instance in the PreProcessor sorted list and calls its Start method which starts the new thread.
Stop	This method sets the status of the PreProcessor class to “Stopped”.
ProcessConfiguration	This method calls Configuration Parser class which reads and parses the ServiceConfig.xml file values It also reads other common configuration values such as connection string After reading the ServiceConfig.xml file it goes over each Pre-Processor configuration and calls RefreshProcessor method.
Refresh	This method sets the status of the PreProcessor class to “Started”.
RefreshProcessor	In this method service control creates a new PreProcessor class instance’s Start method in a new thread After creating new instance thread it puts it in the synchronized sorted list If a particular PreProcessor object already exists it is removed and a new instance is added.
Dispose	This method disposes off the service object.

25.3.2.2 PreProcessor Class

PreProcessor Class implements EDI_RunTimeObjectBase class and IDisposable interface PreProcessor class reads the configuration values from App.config (file name is PreProcessorService.exe.config during runtime), ServiceConfigl.xml files to get configuration values which are common to the entire service Based on those values it creates new Pre-Processor thread for each configuration and starts these threads While starting new thread it passes the value of which directory to monitor and what process handler to use When the PreProcessor class gets a call to stop the Pre-Processor Service then it is also responsible for killing each Pre-Processor handler thread which will finally stop the service.

File Name: PreProcessor.cs	**Class Name: PreProcessor**
Method Name	**Description**
PreProcessor	This is the default constructor of the class This method calls ProcessConfiguration class which parses the processor handler, watcher name and writer name for this PreProcessor instance Then it calls the GetHandler, GetWatcher and GetWriter methods to returns respective object instances to this method It also instantiates the new delegates MessageArrived of PreProcessor class and ExecuteHandler of the ProcessorHandlerBase class in ProcessorHandler project It passes values to ProcessorHandlerBase class by calling its delegates dlgProcessorHandler and dlgProcessorHandlerMessaging.
Start	This method starts the watcher object so it starts monitoring the configured directory at the configured polling interval.
GetHandler	This method reads the configuration to get the configured class name of the handler and based on that value it creates the instance of a specific handler and returns it to PreProcessor constructor
GetWatcher	This method reads the configurations to get the configured class name of the watcher and based on that value it creates the instance of a specific watcher and returns it to PreProcessor constructor.
GetWriter	This method reads the configuration to get the configured class name of the writer and based on that value it creates the instance of a specific writer and returns it to PreProcessor constructor.
MessageArrived	This method calls the dlgProcessorHandler delegate and passes the file location to it This starts the processing of a file in Pre-Processor service by a specific handler and its base In return to the delegate call it gets back the file location of the processed file If this location is not empty and if the file exists then it is moved by the writer object to the final location.

File Name: PreProcessor.cs	Class Name: PreProcessor
Method Name	**Description**
RefreshProcessor	In this method service control creates a new PreProcessor class instance's Start method in a new thread After creating new instance thread it puts it in the synchronized sorted list If a particular PreProcessor object already exists it is removed and a new instance is added.
Dispose	This method disposes off the service object.

25.3.2.3 Processor Configuration Class

This is a static class which is specifically made to read the configuration file and pass it to the Directory Manager class

File Name: ProcessorConfiguration.cs	Class Name: ProcessorConfiguration
Method Name	**Description**
ProcessorConfiguration	This is the default constructor of the class This is the method which actually reads the App.config file and ServiceConfig.xml file After reading the ServiceConfig.xml file it reads the handler, watcher and writer configurations
GetProcessors	This method loads the ServiceConfig.xml configuration file and reads all the "Preprocessor" record configurations, adds them to an ArrayList collection object and returns back the ArrayList.
Dispose	This method disposes off the service object.

25.3.2.4 IService Object Interface

This is the interface which is implemented by Directory Manager and EDI Runtime Object Base classes This class is present in other EDI services also so the implementation of all EDI services is consistent.

File Name: ConfigurationParser.cs	Class Name: ConfigurationParser
Method Name	**Description**
Start	Interface method
Stop	Interface method
Continue	Interface method
Refresh	Interface method
Pause	Interface method

25.3.2.5 EDI Runtime Object Base Class

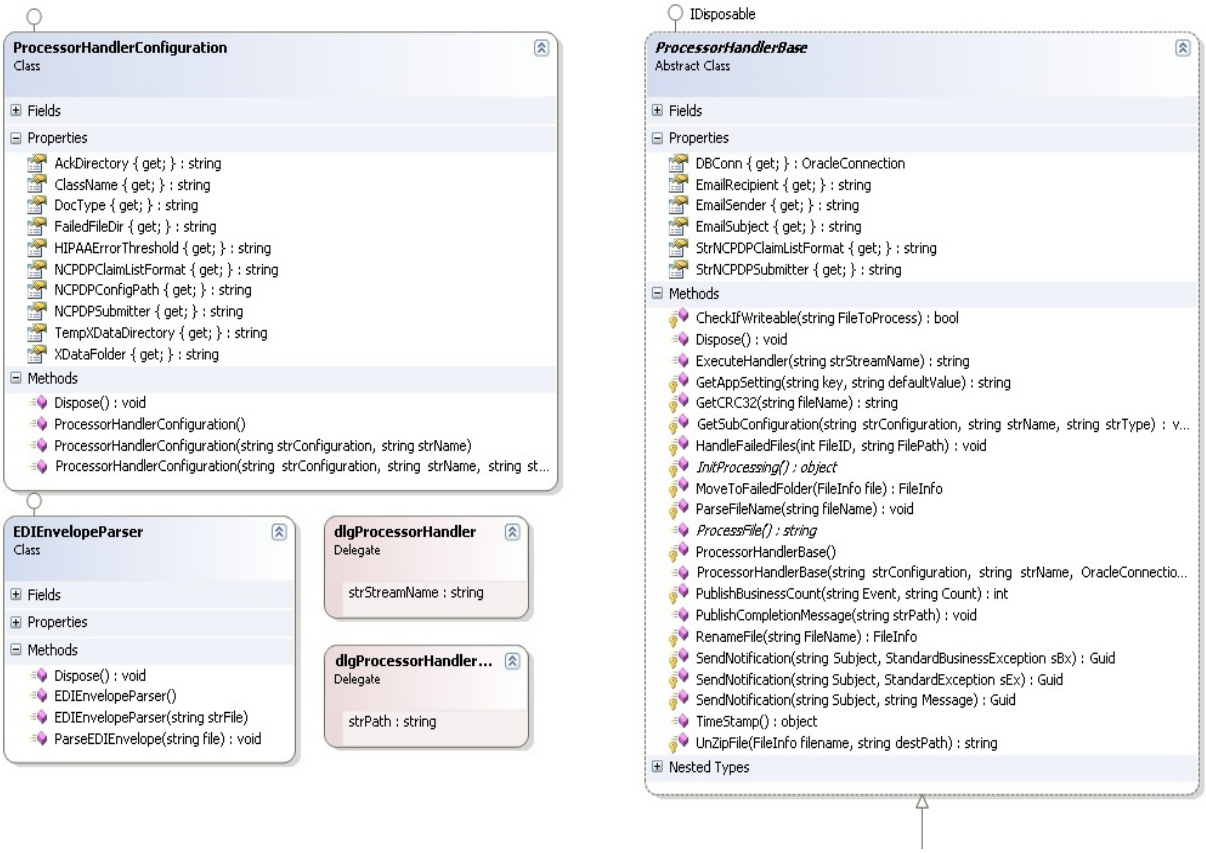
This is an abstract class which implements MarshalByRefObject class, IDisposable interface and IServiceObject interface The purpose of this class is to give a layer to add EDI subsystem specific code.

File Name: EDI_RunTimeObjectBase.cs	**Class Name: EDI_RunTimeObjectBase**
Method Name	**Description**
Start	Abstract method
Stop	Abstract method
Continue	Abstract method
Refresh	Abstract method
Pause	Abstract method

25.3.2.6 Processor Handler Project

This is the project which is actually responsible for processing a file and implementing all the business logic designed for Pre-Processor

25.3.2.7 Common Classes



25.3.2.7.1 Processor Handler Base Class

As the name says, this is the base class for all the Processor Handler classes It has all the common functions needed to perform Pre-Processor Service functions and is implemented by Processor Handler Inbound Base class and Processor Handler Outbound Base class.

File Name: ProcessorHandlerBase.cs	**Class Name: ProcessorHandlerBase**
Method Name	**Description**
ProcessorHandlerBase	This is the default constructor of the class This class reads the App.config and the ServiceConfig.xml to get the run time values.
CheckIfWriteable	This method tried to open the file with an exclusive lock on the file If the file opens in a writeable mode then this method returns true.
ExecuteHandler	This is the method which is called by the PreProcessor class when it starts a new thread This method opens a database connection using the connection string received from the configuration file Then it checks if the file is writeable by calling CheckIfWriteable If the file is writeable it unzips the file if needed by calling UnzipFile method If file is already unzipped it checks its integrity by checking its CRC32 code against the CRC32 value parsed from the file

File Name: ProcessorHandlerBase.cs	Class Name: ProcessorHandlerBase
Method Name	**Description**
	name and then it finally calls the InitProcessing method.
GetAppSettings	This method gets a key and a default value in its call and using the key it parses the configuration value of that key If there is an error in parsing it returns back the default value.
GetCRC32	This method calculates the Crc32 of the file using the ICSharpCode.SharpZipLib assembly's Crc32 class After getting the Crc32 value it converts the value from number format to hex format which is an eight digit alpha numeric value This value matches with the Crc32 value calculated by WinZip product.
GetSubConfiguration	This method gets the SubHandler record values for a PreProcessor record It parses the Xdata folder value and the document type value.
HandleFailedFiles	This method handles the failed files by checking if they exist and calling MoveToFailedFolder.
InitProcessing	This is an abstract method in this class which is implemented in the child classes
MoveToFailedFolder	This file moves all the failed files to the failed folder and if a file already exists it renames the older file name by appending current date and time.
ParseFileName	This method calls the FileNameParser assembly's EDIFileName class to parse values like CRC32, System assign Key (SAK), transport type, date, time, original file name and file extension from the file name.
ProcessFile	This is an abstract method in this class which is implemented in the child classes.
PublishBusinessCount	This method is used to save Business Counts of the file to the database This method uses SP_EVENT_TRACK_INS stored procedure of the PKG_TRACKING Oracle package.
PublishCompletionMessage	This method publishes the final event by calling event and then updates the file status by calling event publisher assembly
RenameFile	This method does the final renaming of the file before passing it to BizTalk.
SendNotification	This method attaches a custom notification object provided by Standard Exception Management Solution to the custom Standard Exception before the exception is

File Name: ProcessorHandlerBase.cs	Class Name: ProcessorHandlerBase
Method Name	**Description**
	persisted in database by calling Publish method of Exception Manager class When Notification object is published it sends out an email to all the configured email addresses.
TimeStamp	This is a common method to provide a date time stamp.
UnzipFile	This method uses ICSharpCode.SharpZipLib assembly's ZipOutputStream to compress a file It also calculates the file's Crc32 value and puts it in the zipped file header.
Dispose	This method disposes off this object.

25.3.2.7.2 Processor Handler Configuration Class

This is the main class which is responsible for parsing the Preprocessor and ProcessorHandler record values from the ServiceConfig.xml file.

File Name: ProcessorHandlerConfiguration.cs	Class Name: ProcessorHandlerConfiguration
Method Name	**Description**
ProcessorHandlerConfiguration	This is the default constructor of the class and is used to parse the values from the PreProcesor and ProcessorHandler records of the configuration file
Dispose	This method disposes off this object.

25.3.2.7.3 EDI Envelope Parser Class

This class implements IDisposable interface and is responsible to parse the key values of EDI document.

File Name: EDIEnvelopeParser.cs	Class Name: EDIEnvelopeParser
Method Name	**Description**
EDIEnvelopeParser	This is the default constructor of the class It can take the file pointer as an input variable and call ParseEDIEnvelope method.
ParseEDIEnvelope	This method is responsible for parsing all the key values of an EDI document Actual parsing of values is done by EDS.USGS.MED.EDI.NRM.X12Messages assembly This method parses the ISA, GS, ST segment values for all documents For claims it also parses BHT segment values For 997 documents it gets AK5 and AK9 values.
Dispose	This method disposes off this object.

25.3.2.8 Inbound Classes

25.3.2.8.1 Processor Handler Inbound Base Class

This class implements ProcessorHandlerBase class and acts as the base class for all the inbound (in respect to MMIS) direction document specific processor handler classes.

File Name: ProcessorHandlerInboundBase.cs	**Class Name: ProcessorHandlerInboundBase**
Method Name	**Description**
ProcessorHandlerInboundBase	This is the default constructor of the class
InitProcessing	This method overrides the base class InitProcessing method and when called is responsible to set up the handler It first checks if the document type is NCPDP and if it is then it starts NCPDP document tracking and returns back NCPDP handler object If document type is not NCPDP then it first cleans the file by removing all the carriage return and line feeds from the file and then sends it for EDI parsing Then it calls the trading partner validations and if the file is valid it continues processing After TPM validations it starts the document tracking for the X12 file It then gets the valid processor handler for the file depending upon its document type and as the last step of

File Name: ProcessorHandlerInboundBase.cs	**Class Name: ProcessorHandlerInboundBase**
Method Name	**Description**
	preparation it starts the XEngine
IsNCPDP	This method checks if the inbound file is a NCPDP document and if it is NCPDP document it then parses its key values.
ProcessFile	In this method file goes through following processing steps First the balancing count of the file is calculated and then the file is renamed Then the file is sent for HIPAA validation and then the business count after validation is calculated Finally EDI is converted into XDATA format and passed to BizTalk.
ParseEDIEnvelope	This method calls ParseEDIEnvelope method of the EDIEnvelopeParser class to parse key EDI values.
GetEDIData	This method copies all the parsed EDI values from the EDIEnvelopeParse object instance to local variables.
ValidateEDI	This method instantiates the EdifecsWrapper assembly's FileHandler class and calls its ValidateInboundX12 method so the inbound EDI file can be validated for HIPAA errors.
ConvertEDItoXDATA	This method is responsible for converting EDI file to XDATA by calling EdifecsWrapper assembly's FileHandler class Once the conversion is complete it then calls the PublishCompletionMessage method.
ValidateTradingPartner	This method is responsible for performing all the Trading Partner Management (TPM) specific validations on the file This method checks if the trading partner is a valid trading partner in the EDI system Then it checks if the file is received from the allowed transport type for the trading partner Then it checks if the current document type of the file is configured for the trading partner and finally if the processing file's Usage Indicator matches with the configured value for that document type.
SaveDocumentTrackingAndTransactionTracking	This method is responsible for starting Document tracking for the EDI file It also inserts document transaction (ST control values) values in the database
SaveNCPDPDocumentTracking	This method is responsible for starting Document tracking for the NCPDP file.
GetNCPDPBusinessCountAfterValidation	This method calls the GetKPIAndBusinessCount method and then persists these values in database by calling insert event tracking method of the event publisher This value is stored under the event code of

File Name: ProcessorHandlerInboundBase.cs	**Class Name: ProcessorHandlerInboundBase**
Method Name	**Description**
	"COUNT_AFTER_VALIDATION".
GetBusinessCountsBeforeValidation	This method calls the GetKPIAndBusinessCount method and then persists these values in database by calling insert event tracking method of the event publisher This value is stored under the event code of "COUNT_BEFORE_VALIDATION".
GetBusinessCountAfterValidation	This method calls the GetKPIAndBusinessCount method and then persists these values in database by calling insert event tracking method of the event publisher This value is stored under the event code of "COUNT_AFTER_VALIDATION".
GetKPIAndBusinessCount	This is a virtual method implemented by child classes.
CheckForEncounterErrorThreshold	This method is called only for encounter claims and is responsible for calculating the percentage of claims that failed HIPAA validations If this number is more than configured number then the entire file is rejected.
GetHandler	This method goes over the transaction type of the file and depending on the value it assigns the processor handler for the file.
GetBaseConfiguration	This method copies the base handler values to its local variables.
FormatCRLFTilde	This method goes over the entire file character by characters and if it finds any carriage return or line feed it removes those characters and creates a new file.
Dispose	This method disposes off this object.

25.3.2.8.2 Processor Handler Inbound 837P X12 Class

Processor Handler Inbound 837P X12 class is used to process Inbound EDI 837 professional claim files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_837_P_X12.cs	**Class Name: ProcessorHandlerInbound_837_P_X12**
Method Name	**Description**
ProcessorHandlerInbound_837_P_X12	This is the default constructor of the ProcessorHandlerInbound_837_P_X12 class.
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerInboundBase class In this method claim file goes through following processing steps First the

File Name: ProcessorHandlerInbound_837_P_X12.cs	Class Name: ProcessorHandlerInbound_837_P_X12
Method Name	**Description**
	balancing count of the file is calculated and then the file is renamed Then the file is sent for HIPAA validations and then the business count after validation is calculated If more than one valid claim exists then file processing continues It then checks for threshold errors by calling CheckForEncounterErrorThreshold method Finally EDI is converted into XDATA format and passed to BizTalk.

25.3.2.8.3 Processor Handler Inbound 837D X12 Class

Processor Handler Inbound 837D X12 class is used to process Inbound EDI 837 dental claim files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_837_D_X12.cs	Class Name: ProcessorHandlerInbound_837_D_X12
Method Name	**Description**
ProcessorHandlerInbound_837_D_X12	This is the default constructor of the ProcessorHandlerInbound_837_D_X12 class.
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerInboundBase class In this method claim file goes through following processing steps First the balancing count of the file is calculated and then the file is renamed Then the file is sent for HIPAA validations and then the business count after validation is calculated If more than one valid claim exists then file processing continues It then checks for threshold errors by calling CheckForEncounterErrorThreshold method Finally EDI is converted into XDATA format and passed to BizTalk.

25.3.2.8.4 Processor Handler Inbound 837I X12 Class

Processor Handler Inbound 837I X12 class is used to process Inbound EDI 837 institutional claim files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_837_I_X12.cs	Class Name: ProcessorHandlerInbound_837_I_X12
Method Name	**Description**
ProcessorHandlerInbound_837_I_X12	This is the default constructor of the ProcessorHandlerInbound_837_I_X12 class.
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerInboundBase class In this method claim file goes through following processing steps First the balancing count of the file is calculated and then the file is renamed Then the file is sent for HIPAA validations and

File Name: ProcessorHandlerInbound_837_I_X12.cs	**Class Name: ProcessorHandlerInbound_837_I_X12**
Method Name	**Description**
	then the business count after validation is calculated If more than one valid claim exists then file processing continues It then checks for threshold errors by calling CheckForEncounterErrorThreshold method Finally EDI is converted into XDATA format and passed to BizTalk.

25.3.2.8.5 Processor Handler Inbound 270 X12 Class

Processor Handler Inbound 270 X12 class is used to process Inbound EDI 270 batch eligibility request files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_270_X12.cs	**Class Name: ProcessorHandlerInbound_270_X12**
Method Name	**Description**
ProcessorHandlerInbound_270_X12	This is the default constructor of the ProcessorHandlerInbound_270_X12 class This method calls the base methods like GetEDIData, GetBaseConfiguration and GetSubConfiguration to get all the values for processing of the file.
GetKPIAndBusinessCount	This overrides GetKPIAndBusinessCount method of base ProcessorHandlerInboundBase class This method reads the entire file and then uses Regex methods to calculate the business count.

25.3.2.8.6 Processor Handler Inbound 276 X12 Class

Processor Handler Inbound 276 X12 class is used to process Inbound EDI 276 batch claim status inquiry files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_276_X12.cs	**Class Name: ProcessorHandlerInbound_276_X12**
Method Name	**Description**
ProcessorHandlerInbound_276_X12	This is the default constructor of the ProcessorHandlerInbound_276_X12 class This method calls the base methods like GetEDIData, GetBaseConfiguration and GetSubConfiguration to get all the values for processing of the file.
GetKPIAndBusinessCount	This overrides GetKPIAndBusinessCount method of base ProcessorHandlerInboundBase class This method reads the entire file and then uses Regex methods to calculate the business count.

25.3.2.8.7 Processor Handler Inbound NCPDP Class

Processor Handler Inbound NCPDP class is used to process Inbound NCPDP claim files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_NCPDP.cs	**Class Name: ProcessorHandlerInbound_NCPDP**
Method Name	**Description**
ProcessorHandlerInbound_NCPDP	This is the default constructor of the ProcessorHandlerInbound_NCPDP class This method calls the base GetBaseConfiguration to get all the values for processing of the file.
ConvertToSingleClaimList	This reads the NCPDP request Xml which is created by NCPDP translator while validating and transforming native NCPDP inbound file This method reads the response tag of each claim and collects the details about the claims which failed NCPDP validations These validation errors are put in a custom Standard Exception object and persisted in database by calling Publish method of the Exception Manager class While it goes over each claim, all the valid claims are collected and put under one claimlist XML tag and as a result claimlist Xml file is created which has only valid claims.
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerInboundBase class In this method claim file goes through following processing steps First the balancing count of the file is calculated and then the file is renamed Then the file is sent for NCPDP validations and then the business count after validation is calculated If a single claimlist Xml is requested then the ConvertToSingleClaimList method is called to convert the file to single claimlist Finally business count after validation is published and processing is finished.

25.3.2.8.8 Processor Handler Inbound NSF Class

Processor Handler Inbound NSF class is used to process Inbound NSF claim files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_NSF.cs	**Class Name: ProcessorHandlerInbound_NSF**
Method Name	**Description**
ProcessorHandlerInbound_NSF	This is the default constructor of the ProcessorHandlerInbound_NSF class.
InitProcessing	This method publishes an event stating that Pre-Processor is processing the file.

File Name: ProcessorHandlerInbound_NSF.cs	**Class Name: ProcessorHandlerInbound_NSF**
Method Name	**Description**
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerInboundBase class This method first parses all the key values of the file and uses them to do trading partner validations If validation succeeds then processing continues Then it starts the document tracking and gets the business count before validation After getting business count it renames the file and passes it for NSF validations After NSF validations another business count is taken and then published to the database If no valid claims exist then an error notification email is generated which is sent using Exception Manager's Publish method.
GetKPIAndBusinessCount	This overrides GetKPIAndBusinessCount method of base ProcessorHandlerInboundBase class This method reads the entire file and then uses Regex methods to calculate the business count.
ValidateTP	This method is responsible for performing Trading Partner Management (TPM) specific validations for NSF file This method checks if the trading partner is a valid trading partner in the EDI system Then it checks if the file is received from the allowed transport type for the trading partner If both validations are successful then a true response is sent back.

25.3.2.8.9 Processor Handler Inbound 997 X12 Class

Processor Handler Inbound 997 X12 class is used to process Inbound EDI 997 Acknowledgement files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_997_X12.cs	**Class Name: ProcessorHandlerInbound_997_X12**
Method Name	**Description**
ProcessorHandlerInbound_997_X12	This is the default constructor of the ProcessorHandlerInbound_997_X12 class This method calls the base methods like GetEDIData, GetBaseConfiguration and GetSubConfiguration to get all the values for processing of the file.
ProcessFile	This is overrides ProcessFile method of base ProcessorHandlerInboundBase class In this method acknowledgement file goes through following processing steps First key values are parsed from the 997 acknowledgement Then the acknowledgement response value is stored in database by calling SaveFunctionalAcknowledgement of the TPMLibrary assembly's TPM class Then the acknowledgement is

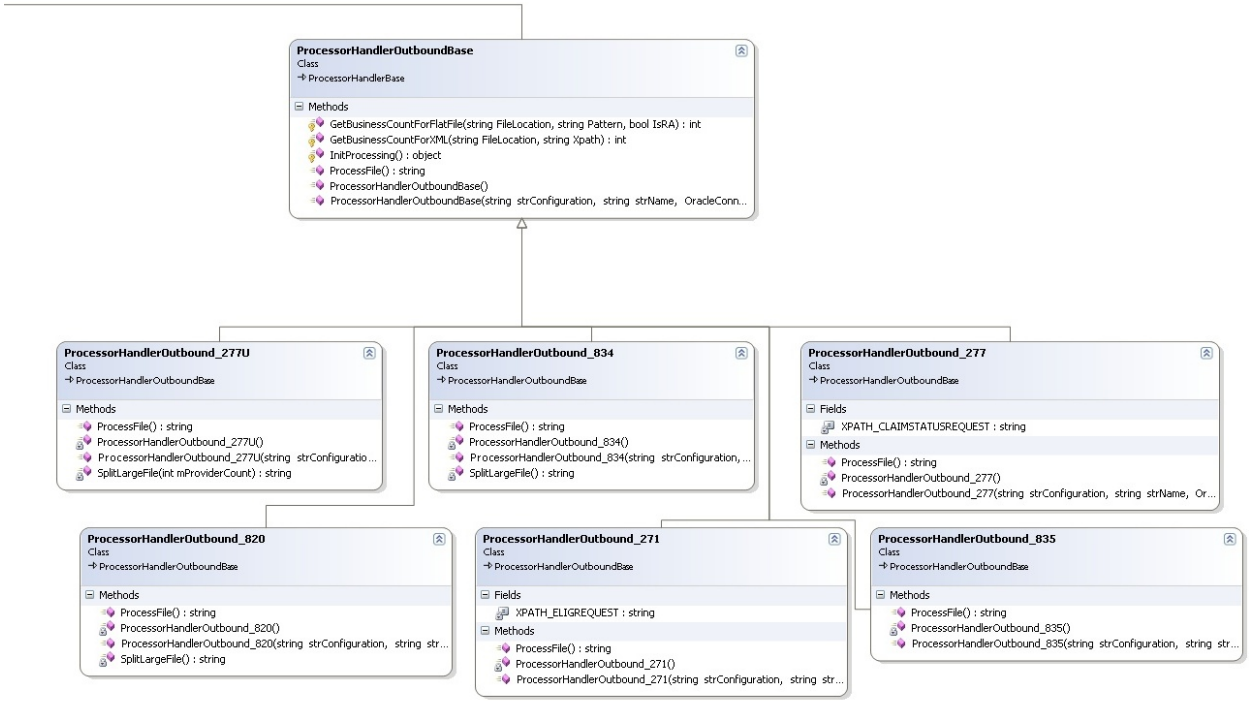
File Name: ProcessorHandlerInbound_997_X12.cs	**Class Name: ProcessorHandlerInbound_997_X12**
Method Name	**Description**
	correlated to the original inbound X12 document by calling the CorrelateFunctionalAckInsert method of the TPM class.

25.3.2.8.10Processor Handler Inbound TA1 X12 Class

Processor Handler Inbound TA1 X12 class is used to process Inbound EDI TA1 Acknowledgement files This class implements ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_TA1_X12.cs	**Class Name: ProcessorHandlerInbound_TA1_X12**
Method Name	**Description**
ProcessorHandlerInbound_997_X12	This is the default constructor of the ProcessorHandlerInbound_TA1_X12 class This method calls the base methods like GetEDIData, GetBaseConfiguration and GetSubConfiguration to get all the values for processing of the file.
ProcessFile	This is overrides ProcessFile method of base ProcessorHandlerInboundBase class In this method acknowledgement file goes through following processing steps First key values are parsed from the TA1 acknowledgement Then the acknowledgement response value is stored in database by calling SaveFunctionalAcknowledgement of the TPMLibrary assembly's TPM class Then the acknowledgement is correlated to the original inbound X12 document by calling the CorrelateFunctionalAckInsert method of the TPM class.

25.3.2.9 Inbound Classes



25.3.2.9.1 Processor Handler Outbound Base Class

This class implements ProcessorHandlerBase class and act as the base class for all the outbound (in respect to MMIS) direction document specific processor handler classes.

File Name: ProcessorHandlerOuboundBase.cs	**Class Name: ProcessorHandlerOuboundBase**
Method Name	**Description**
ProcessorHandlerOuboundBase	This is the default constructor of the ProcessorHandlerOuboundBase class and it sets the direction of the file to Outbound.
InitProcessing	This method publishes an event saying that file is being processed by Pre-Processor Service.
ProcessFile	This method overrides the base class method is responsible for publishing the business count and renaming the file Currently all the document type specific handlers override this method.
GetBusinessCountForFlatFile	This method takes the file path and the search pattern string as the input parameters and then it opens the file and uses String methods to get the business count.
GetBusinessCountForXML	This method takes the file path and the Xpath string as the input parameters and then it opens the file and uses Xml

File Name: ProcessorHandlerOuboundBase.cs	**Class Name: ProcessorHandlerOuboundBase**
Method Name	**Description**
	Document methods to get the business count.

25.3.2.9.2 Processor Handler Outbound 271 Class

Processor Handler Outbound 271 class is used to process outbound 270 Xml file which is in interchange format This class implements ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound271.cs	**Class Name: ProcessorHandlerOutbound271**
Method Name	**Description**
ProcessorHandlerOutbound271	This is the default constructor of the ProcessorHandlerOutbound271 class
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerOutboundBase class This method first gets the business counts by calling GetBusinessCountForXML method of the base class and then it publishes the counts using event publisher When this is done it returns the file name as the response.

25.3.2.9.3 Processor Handler Outbound 277 Class

Processor Handler Outbound 277 class is used to process outbound 277 Xml file which is in interchange format This class implements ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound277.cs	**Class Name: ProcessorHandlerOutbound277**
Method Name	**Description**
ProcessorHandlerOutbound277	This is the default constructor of the ProcessorHandlerOutbound277 class
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerOutboundBase class This method first gets the business counts by calling GetBusinessCountForXML method of the base class and then it publishes the counts using event publisher When this is done it returns the file name as the response.

25.3.2.9.4 Processor Handler Outbound 277U Class

Processor Handler outbound 277U class is used to process outbound 277U flat file which is in interchange format This class implements ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound277U.cs	**Class Name: ProcessorHandlerOutbound277U**
Method Name	**Description**
ProcessorHandlerOutbound277U	This is the default constructor of the ProcessorHandlerOutbound277U class
ProcessFile	This is overrides ProcessFile method of base ProcessorHandlerOutboundBase class This method first calculates the business counts by reading the record types and then it publishes the counts using event publisher It then checks if the file has more than five hundred provider records then it calls SplitLargeFile function to split the file in smaller parts.
SplitLargeFile	This method is called to logically split the file into smaller parts while still maintaining the integrity of the file This method separates the records in two header records and body records Header consists of constant records which occur only once in the file while body consists of multiple provider and other loops This method loops over each record of the file and splits the file at every five hundred provider record count and puts them in a new file with the header on top When it is done splitting the file it returns all the new file names separated by “~”.

25.3.2.9.5 Processor Handler Outbound 820 Class

Processor Handler outbound 820 class is used to process outbound 820 flat file which is in interchange format This class implements ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound820.cs	**Class Name: ProcessorHandlerOutbound820**
Method Name	**Description**
ProcessorHandlerOutbound820	This is the default constructor of the ProcessorHandlerOutbound820 class
ProcessFile	This is overrides ProcessFile method of base ProcessorHandlerOutboundBase class This method first calculates the business counts by calling GetBusinessCountForFlatFile method and then it publishes the counts using event publisher It then checks if the business count is more than two thousand and five hundred then it calls SplitLargeFile function to split the file in smaller parts.

File Name: ProcessorHandlerOutbound820.cs	**Class Name: ProcessorHandlerOutbound820**
Method Name	**Description**
SplitLargeFile	This method is called to logically split the file into smaller parts while still maintaining the integrity of the file This method loops over each record of the file and splits the file at every two thousand five hundred record count and puts them in a new When it is done splitting the file it returns all the new file names separated by “~”.

25.3.2.9.6 Processor Handler Outbound 834 Class

Processor Handler outbound 834 class is used to process outbound 834 flat file which is in interchange format This class implements ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound834.cs	**Class Name: ProcessorHandlerOutbound834**
Method Name	**Description**
ProcessorHandlerOutbound834	This is the default constructor of the ProcessorHandlerOutbound834 class
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerOutboundBase class This method first calculates the business counts by calling GetBusinessCountForFlatFile method and then it publishes the counts using event publisher It then checks if the business count is more than five hundred then it calls SplitLargeFile function to split the file in smaller parts.
SplitLargeFile	This method is called to logically split the file into smaller parts while still maintaining the integrity of the file This method separates the records in two header records and body records Header consists of constant records which occur only once in the file while body consists of multiple member and other loops This method loops over each record of the file and splits the file at every five hundred member record count and puts them in a new file with the header on top When it is done splitting the file it returns all the new file names separated by “~”.

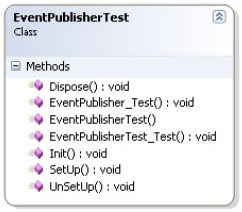
25.3.2.9.7 Processor Handler Outbound 835 Class

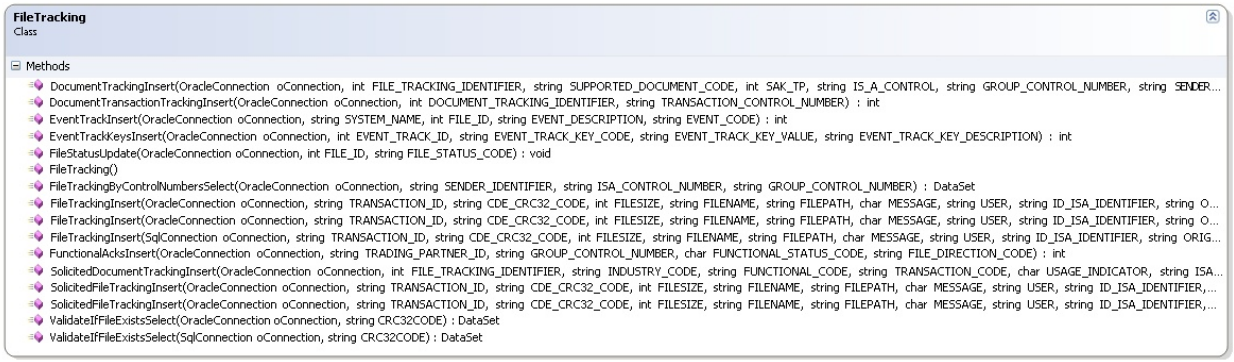
Processor Handler outbound 835 class is used to process outbound 835flat file which is in interchange format This class implements ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound835.cs	**Class Name: ProcessorHandlerOutbound835**
Method Name	**Description**
ProcessorHandlerOutbound835	This is the default constructor of the ProcessorHandlerOutbound835 class
ProcessFile	This overrides ProcessFile method of base ProcessorHandlerOutboundBase class This method first calculates the business counts by reading the record types and then it publishes the counts using event publisher.

25.3.3 File Tracking Library Project

This project is used to create classes which help EDI services to start and continue file tracking through the n-layer solution spanning over multiple servers This library is accessed from File Transfer Service, pre-processor service, post-processor service and Event Publisher.





25.3.3.1 File Tracking Class

This class has all the functions which provide the ability to add, modify or continue File tracking.

File Name: FileTracking.cs	**Class Name: FileTracking**
Method Name	**Description**
FileTracking	This is the default constructor of the class

File Name: FileTracking.cs	Class Name: FileTracking
Method Name	**Description**
DocumentTrackingInsert	This method is used to create a new Document Tracking record This method is called to start the file tracking of any file This method uses SP_DOCUMENT_TRANSACTIONS_INS stored procedure of the PKG_TRACKING Oracle package.
DocumentTransactionTrackingInsert	This method is used to create a new Document Transaction Tracking record This method is called to start the file tracking of any file This method uses SP_DOCUMENT_TRACK_INS stored procedure of the PKG_TRACKING Oracle package.
EventTrackInsert	This method is used to save events (audit steps) of the file processing to the database This method uses SP_EVENT_TRACK_INS stored procedure of the PKG_TRACKING Oracle package.
FileStatusUpdate	This method is used to update file status in the database This method uses SP_FILE_TRACK_UPD_STATUS stored procedure of the PKG_TRACKING Oracle package.
FileTrackingByControlNumbersSelect	This method is not used by Pre-Processor Service.
FileTrackingInsert	This method is not used by Pre-Processor Service.
FunctionalAckInsert	This method is used to update file status in the database This method uses SP_FUNCTIONAL_ACKS_STATUS stored procedure of the PKG_TRACKING Oracle package.
SolicitedDocumentTrackingInsert	This method is not used by Pre-Processor Service.
SolicitedFileTrackingInsert	This method is not used by Pre-Processor Service.
ValidateFileExistsSelect	This method checks the existing T_FILE_TRACK table records to match the Crc32 value of the file This is used during the duplicate check of the file.

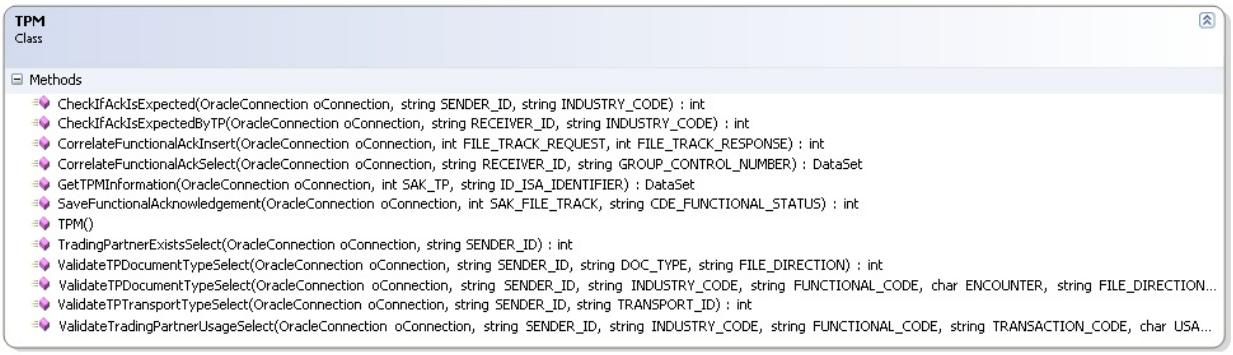
25.3.3.2 File Exception Class

This class provides the ability to correlate exception data persisted in database using Exception Manager Publish method with the File Tracking record.

File Name: FileException.cs	Class Name: FileException
Method Name	**Description**
FileException	This is the default constructor of the class
FileTrackingExceptionsInsert	This method downloads a specific file from FTP servers to the local location.

25.3.4 TPM Library Project

This project is responsible for performing Trading Partner Management (TPM) specific validation on the inbound and outbound EDI documents This assembly is accessed by both Pre-Processor Service and Post-Processor Service.



25.3.4.1 TPM Class

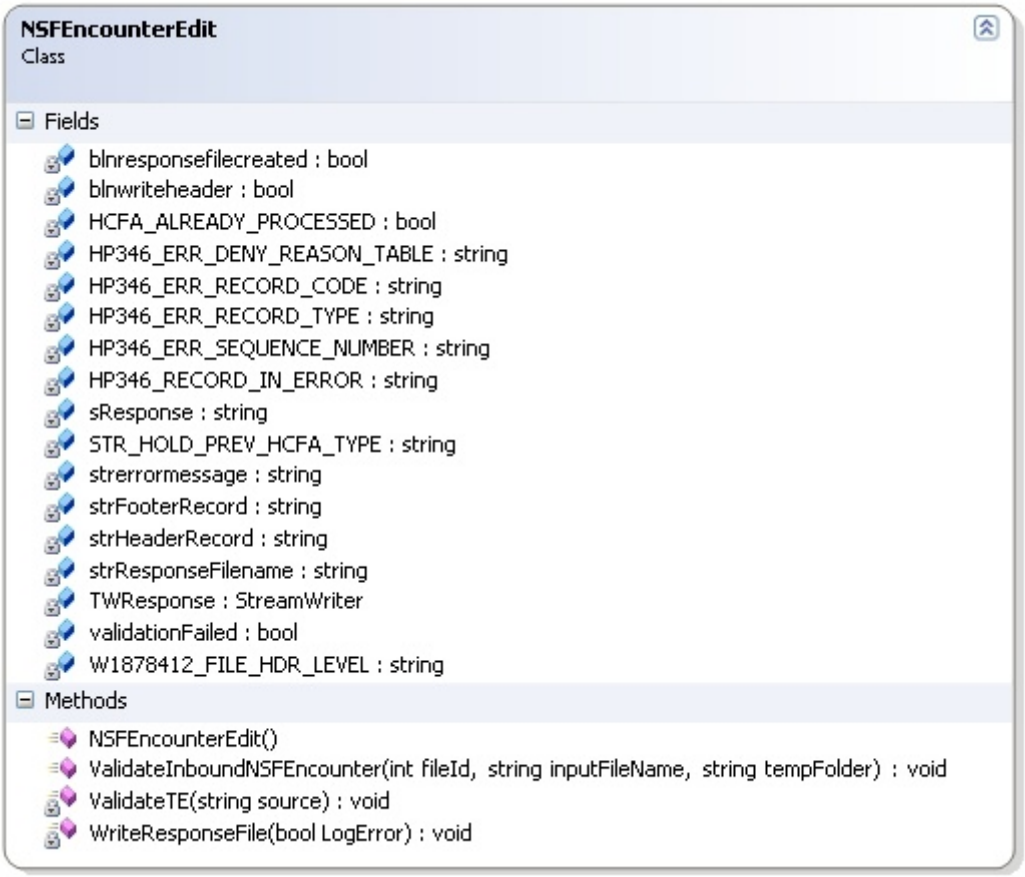
This class is the main class which provides the methods to do TPM validations or TPM specific data inquiry.

File Name: TPM.cs	**Class Name: TPM**
Method Name	**Description**
TPM	This is the default constructor of the class
GetTPMInformation	This method is not used by Pre-Processor Service.
TradingPartnerExistsSelect	This function checks the T_TP table to see if the trading partner exists with active status during the run time of the method It returns SAK_TP as the return value.
ValidateTradingPartnerUsageSelect	This function selects data against T_TP, T_TP_CONFIG, T_CDE_SUPPORTED_DOC and T_CDE_FILE_DIRECTION tables to see if the document type exists with the correct usage indicator passed as the method parameter It returns the SAK_TP_CONFIG as the return parameter.
CorrelateFunctionalAckSelect	This method is not used by Pre-Processor Service.
SaveFunctionalAcknowledgement	This method takes the File ID and Functional Acknowledgement value as the input parameters Based on File ID it searches for SAK_DCOUMENT_TRACK Then its gets the next unique value for SAK_FUNCTIONAL_ACK using the sequence key With these values it inserts a record in T_FUNCTIONAL_ACK table and returns SAK_FUNCTIONAL_ACK value.
ValidateTPDocumentTypeSelect	This function selects data against T_TP, T_TP_CONFIG, T_CDE_SUPPORTED_DOC and T_CDE_FILE_DIRECTION tables to see if the document type is configured for the given trading partner It returns

File Name: TPM.cs	**Class Name: TPM**
Method Name	**Description**
	the SAK_TP_CONFIG as the return parameter.
ValidateTPTransportTypeSelect	This function selects data against T_TP, T_TP_ALLOWED_TRANSPORT, and T_CDE_TRANSPORT tables to see if the trading partner is configured for the given transport type through which file was received It returns the SAK_ALLOWED_TRANSPORT as the return parameter.
CorrelateFunctionalAckInsert	This method is used to correlate two File IDs usually in the case of request response files This method uses SP_FILE_CORRELATION_INS stored procedure of the PKG_TRACKING Oracle package.
CheckIfAckIsExpected	This function selects data against T_TP, T_TP_CONFIG and T_CDE_SUPPORTED_DOC tables to see if the received document type is configured to receive 997 functional acknowledgements It returns the SAK_TP_CONFIG as the return parameter.
CheckIfAckIsExpectedByTP	This method is not used by Pre-Processor Service.
ValidateNCPDPTradingPartner	This function selects data against T_TP, T_TP_CONFIG, T_CDE_SUPPORTED_DOC and T_CDE_FILE_DIRECTION tables to see if the NCPDP document type is configured for a trading partner It returns the SAK_TP_CONFIG as the return parameter.

25.3.5 NSF Encounter Edit Project

This project is a custom code to validate Nation Standard format (NSF) claims received from Transportation cabinet This project also helps building NSF response file.



25.3.5.1 NSF Encounter Edit Class

This class is the main class which has methods for validating NSF transportation cabinet claim file and logic of splitting large batch NSF file to individual NSF claim file This class is responsible for generating NSF claims acknowledgement response file during validating NSF transportation cabinet claim file.

File Name: NSFEncounterEdit.cs	**Class Name: NSFEncounterEdit**
Method Name	**Description**
NSFEncounterEdit	This is the default constructor of the class
ValidateInboundNSFEncounter	This is the only public method which is called by ProcessorHandlerInbound_NSF class for validating and splitting the NSF Transportation claim file This method takes FileID, inputFileName, tempFolder, SplitFilesFolder as input parameters and makes GetBusinessCountForFlatFile and ValidateTE method calls.

File Name: NSFEncounterEdit.cs	**Class Name: NSFEncounterEdit**
Method Name	**Description**
ValidateTE	This method is used in validating and splitting NSF Transportation claim file This method takes source file name string as input parameter This method has business validations for validating the NSF file and generates the NSF Validation response file While validating the NSF claim file this method makes CreateInvalidClaimsNSFFile, Split LargeFile, DeleteSplitFile method calls.
WriteResponseFile	This method is used for writing the NSF response acknowledgement file during the process of Validation and splitting
GetBusinessCountForFlatFile	This method takes file name as input parameter and is used in calculating the Business count.
SplitLargeFile	This method is used in splitting the Large Batch NSF Transportation file
CreateInvalidClaimsNSFFile	This method is used during Validating NSF Transportation file for creating the invalid claims NSF file.
DeleteSplitFile	This method is used in deleting the split files which gets created during Validation and splitting of NSF Transportation file.

25.4 Pre-Processor Service Configuration File

Configuration files are responsible for providing configuration data to Pre-Processor Service during run time These files constitute data which needs to be changed often and is needed only during run time.

25.4.1 App.Config

This is the default configuration file that can be built along with the service component This file access is supported by default .Net components The purpose of this file is to host values which are common to all service components.

Sample App.config file:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
      <appSettings>
            <add key="RemotingChannel" value="9088" />
            <add key="InstanceName" value="EDS.PREPROC.TCP" />
            <add key="Configuration" value="C:\Program
Files\EDS\PreProcessorService\ServiceConfig.xml" />
            <add key="ConfigIsFile" value="true" />
            <add key="ConnString" value="Data Source=kymisa1;User ID=xxxxx;
Password=xxxxx; Integrated Security=no;" />
            <add key="WMIIPAddress" value="SBG" />
            <add key="WMIPortNo" value="9090" />
            <add key="WMIRuntimeObjectName" value="EDS.MSMQ.TCP" />
            <add key="RecipientAddress" value="JoneDoe@eds.com" />
            <add key="SenderAddress" value=" JoneDoe@eds.com" />
            <add key="EmailSubject" value="Pre-processor SBG " />
            <add key="ServerName" value="SBG" />
            <add key="ComponentSeparator" value="0x3a" />
            <add key="SegmentTerminator" value="0x7e" />
      </appSettings>
</configuration>
```

File Name: App.config	
Attribute Name	**Description**
RemotingChannel	This attribute holds the value through which we can remote into FTS This is not yet implemented.
InstanceName	This is the channel name under which it will be registered This is not yet implemented
Configuration	This attribute value holds the location of the ServiceConfig.xml configuration file.
ConfigIsFile	This attribute takes true or false value and helps service to determine if it should look for a file for configuration.
ConnString	This attribute value holds the database connection string.
WMIIPAddress	This attribute holds the value of the IPAddress or server

File Name: App.config	
Attribute Name	**Description**
	name where MSMQ Listener Service is installed.
WMIPortNo	This attribute holds the value of the port number to remote in to MSMQ Listener Service.
WMIRuntimeObjectName	This is the channel name under which MSMQ remoting stub is registered.
RecipientAddress	This is the global recipient email address where error notification emails are sent.
SenderAddress	This is the global sender email address used to send error notification emails.
EmailSubject	This is the common text that can be attached to the notification email.
ServerName	This is the name of the server where FTS is deployed.
ComponentSeparator	This is default value of the component separator It is configured to be 0x3a.
SegmentTerminator	This is default value of the component separator It is configured to be 0x7e.

25.4.2 ServiceConfig.xml

This is the main configuration file which holds the configuration details about which directory to monitor, where to deliver, what file names to pick, how to rename at destination, where to archive etc There is supposed to be at least one configuration record for each file transfer If the source and destination locations are both accessed through FTP or FTPS transport type then such transfers are accomplished using two configuration records, one to download the file from the resource server and the other to upload the file to the destination server.

Sample FTSConfigLocal.xml file:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
      <PreProcessors>
            <PreProcessor name="Inbound_X12_Generic"
watcherName="Inbound_X12_Generic" writerName="Inbound_X12_Generic"
processorHandlerName="Inbound_X12_Generic" />
            <PreProcessor name="Inbound_NCPDP_Passport"
watcherName="NCPDP_Passport" writerName="NCPDP_Passport"
processorHandlerName="Inbound_NCPDP_Passport" />
            <PreProcessor name="PreProcessor21"
watcherName="277U_OUT_Passport" writerName="277U_OUT_Passport"
processorHandlerName="Outbound_passport_277U" />
      </PreProcessors>
      <SubHandlers>
            <SubHandler name="Inbound_X12_Generic">
```

```
                  <XDATA type="837P" Doctype="837_P_X12_BATCH"
XDataFolder="C:\Edifecs_testing\FTS\Generic\Claim837P\Inbound837P\XData"/>
            </SubHandler>
      </SubHandlers>
      <ProcessorHandlers>
            <ProcessorHandler name="Inbound_X12_Generic">
                  <Assembly name="C:\Program
Files\EDS\PreProcessorService\EDS.USGS.MED.EDI.PreProcessors.ProcessorHandler
.dll" />
                  <Class
name="EDS.USGS.MED.EDI.PreProcessors.ProcessorHandlers.ProcessorHandlerInboun
dBase" />
                  <DocType value="" />
                  <FailedFilesDirectory value="C:\Edifecs_testing\FTS\FAILED"
/>
                  <XDATA
AckDirectory="C:\Edifecs_testing\FTS\Generic\ACKs\Outbound\X12"
TempXDataDirectory="C:\Program Files\EDS\PreProcessorService\TempFolder"
HIPAAErrorThreshold="5" XDataFolder=""/>
            </ProcessorHandler>
            <ProcessorHandler name="Inbound_NCPDP_Passport">
                  <Assembly name="C:\Program
Files\EDS\PreProcessorService\EDS.USGS.MED.EDI.PreProcessors.ProcessorHandler
.dll" />
                  <Class
name="EDS.USGS.MED.EDI.PreProcessors.ProcessorHandlers.ProcessorHandlerInboun
d_NCPDP" />
                  <DocType value="NCPDP_GENERIC" />
                  <FailedFilesDirectory value="C:\Edifecs_testing\FTS\FAILED"
/>
                  <XDATA AckDirectory="" TempXDataDirectory=""
HIPAAErrorThreshold="5"  XDataFolder=""/>
                  <NCPDPSubmitter value="Passport" />
                  <NCPDPConfigPath
value="C:\Edifecs_testing\FTS\Builds\NCPDP\TranslatorProcess\NCPDPBatchConfig
uration.xml" />
                  <NCPDPClaimListFormat value="Single" />
            </ProcessorHandler>
            <ProcessorHandler name="Outbound_passport_277U">
                  <Assembly name="C:\Program
Files\EDS\PreProcessorService\EDS.USGS.MED.EDI.PreProcessors.ProcessorHandler
.dll" />
                  <Class
name="EDS.USGS.MED.EDI.PreProcessors.ProcessorHandlers.ProcessorHandlerOutbou
nd_277U" />
                  <DocType value="277_U_X12_BATCH" />
                  <FailedFilesDirectory value="C:\Edifecs_testing\FTS\FAILED"
/>
                  <XDATA AckDirectory="" TempXDataDirectory=""
HIPAAErrorThreshold="5"  XDataFolder=""/>
            </ProcessorHandler>
      </ProcessorHandlers>
      <Watchers>
            <Watcher name="Inbound_X12_Generic" type="UNCWatcherType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.UNCWatcher" />
                  <TriggeringInterval value="5000" />
```

```
                  <TriggeringType value="PollingBased" />
                  <WatcherLocation
value="C:\Edifecs_testing\FTS\Generic\X12\Inbound" />
            </Watcher>
            <Watcher name="NCPDP_Passport" type="UNCWatcherType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.UNCWatcher" />
                  <TriggeringInterval value="5000" />
                  <TriggeringType value="PollingBased" />
                  <WatcherLocation
value="C:\Edifecs_testing\fts\Passport\NCPDP\Inbound" />
            </Watcher>
            <Watcher name="277U_OUT_Passport" type="UNCWatcherType">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.UNCWatcher" />
                  <TriggeringInterval value="5000" />
                  <TriggeringType value="PollingBased" />
                  <WatcherLocation
value="C:\Edifecs_testing\FTS\Generic\RemittanceAdvice277U\Outbound277U\Inter
Change" />
            </Watcher>
      </Watchers>
      <Writers>
            <Writer name="Inbound_X12_Generic" type="UNCWriter">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Writers.UNCWriter" />
                  <WriteLocation
value="C:\Edifecs_testing\FTS\Generic\Claim837P\Inbound837P\XData" />
                  <WriteAsync value="true" />
            </Writer>
            <Writer name="NCPDP_Passport" type="UNCWriter">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Writers.UNCWriter" />
                  <WriteLocation
value="C:\Edifecs_testing\fts\Passport\NCPDP\Preprocessor" />
                  <WriteAsync value="true" />
            </Writer>
            <Writer name="277U_OUT_Passport" type="UNCWriter">
                  <ClassName
value="EDS.USGS.MED.EDI.Common.Writers.UNCWriter" />
                  <WriteLocation
value="C:\Edifecs_testing\FTS\Generic\RemittanceAdvice277U\Outbound277U\Prepr
ocessor" />
                  <WriteAsync value="true" />
            </Writer>
      </Writers>
      <WatcherTypes>
            <WatcherType name="UNCWatcherType">
                  <WildcardFilter value="*.*" enabled="true" />
                  <RegexFilter value="^([A-Za-z0-9.])*$" enabled="false" />
                  <IncludeSubFolders value="false" />
                  <EventTriggerTypes>
                        <Creation enabled="true" />
                        <Deletion enabled="true" />
                        <Renaming enabled="true" />
                        <Changing enabled="true" />
                  </EventTriggerTypes>
```

```
            </WatcherType>
        </WatcherTypes>
</configuration>
```

25.4.2.1 PreProcessor Record

Parent Record Name: PreProcessors	
Attribute Name	**Description**
PreProcessor	This is the parent element name.
name	This attribute provides the unique name for this PreProcessor configuration.
WatcherName	This attribute gives the name of the watcher configuration to use for this PreProcessor.
WriterName	This attribute gives the name of the writer configuration to use for this PreProcessor.
processorHandlerName	This attribute gives the name of the ProcessorHandler configuration to use for this PreProcessor.

25.4.2.2 Sub-Handler Record

Parent Record Name: SubHandlers	
Attribute Name	**Description**
SubHandler	This is the parent element name.
name	This attribute provides the unique name for this SubHandler configuration.
type	This attribute provides the short name of the document type for which this configuration is being set This value is used in code calculations.
DocType	This configuration provides the identification code of the document as set in the database This value should match with the CDE_IDENTIFICATION column in the T_CDE_SUPPORTED_DOC table.
XDataFolder	This attribute gives the folder location where the XDATA files for this document type are put by Pre-Processor Service.

25.4.2.3 Processor Handler Record

Parent Record Name: ProcessorHandlers	
Attribute Name	**Description**
ProcessorHandler	This is the parent element name.
name	This attribute provides the unique name for this

Parent Record Name: ProcessorHandlers	
Attribute Name	**Description**
	ProcessorHandler configuration.
Assembly	This lists the assembly name of the ProcessorHandler class.
Class	This lists the class name of the ProcessorHandler to use for this configuration.
DocType	This configuration provides the identification code of the document as set in the database This value should match with the CDE_IDENTIFICATION column in the T_CDE_SUPPORTED_DOC table.
FailedFilesDirectory	This folder lists the directory name where the file should be kept if there were an error in processing the file.
AckDirectory	This attribute gives the folder location where the acknowledgement TA1 or 997 files for the inbound X12 files are placed.
TempXDataDirectory	This attribute lists the folder name where temporary files are kept during HIPAA validations.
HIPAAErrorThreshold	This attribute value is used only for encounter claims and it lists the percentage of the error threshold above which encounter processing stops and the entire file is rejected.
XDataFolder	This attribute gives the folder location where the XDATA files for this document type are put by Pre-Processor Service.

25.4.2.4 Watcher Record

Parent Record Name: Watchers	
Attribute Name	**Description**
Watcher	This is the parent element name.
name	This attribute provides the unique name for this Watcher configuration.
type	This attribute defines the type of the watcher to use Currently “UNCWatcherType” value is the only value that is supported.
ClassName	This attribute gives the name of the class to use for this watcher.
TriggeringInterval	This is the polling interval of the watcher This value sets the time interval at which the folder being watched is

Parent Record Name: Watchers	
Attribute Name	**Description**
	checked for files.
TriggeringType	This is the type of the trigger to use for the watcher Currently "PollingBased" is the only supported value.
WatcherLocation	This attribute gives the folder name which is supposed to be watched by this watcher.

25.4.2.5 Writer Record

Parent Record Name: Writers	
Attribute Name	**Description**
Writer	This is the parent element name.
name	This attribute provides the unique name for this Writer configuration.
type	This attribute defines the type of the writer to use Currently “UNCWriter” value is the only value that is supported.
ClassName	This attribute gives the name of the class to use for this writer.
WriteLocation	This attribute gives the folder name where writer is supposed to create the new file.
WriteAsync	This element determines if the writer threads are supposed to be synchronous or asynchronous.

26 Appendix X – Post-Processor Service Documentation_V1

26.1 Introduction

Post-Processor Service is a windows service component which is used to process EDS ODC Batch BizTalk Servers output files. Some of the main functions performed by Post-Processor Service before it passes the file to destination sub system are document balancing, event tracking and Key Performance Indicator (KPI) tracking. For Outbound X12 files Post Processor merges the split files and performs HIPAA validations by making calls to Edifecs XEngine to convert X12 XData files to EDI files.

26.1.1 Post-Processor Service Features

26.1.1.1 Document Balancing and Tracking

When the file is received by Post-processor service publishes Tracking audit events against the existing File ID of the file. Post-Processor Service also counts the number of transactions in each file received and adds these values to event tracking.

26.1.1.2 HIPAA Validation and Transformation

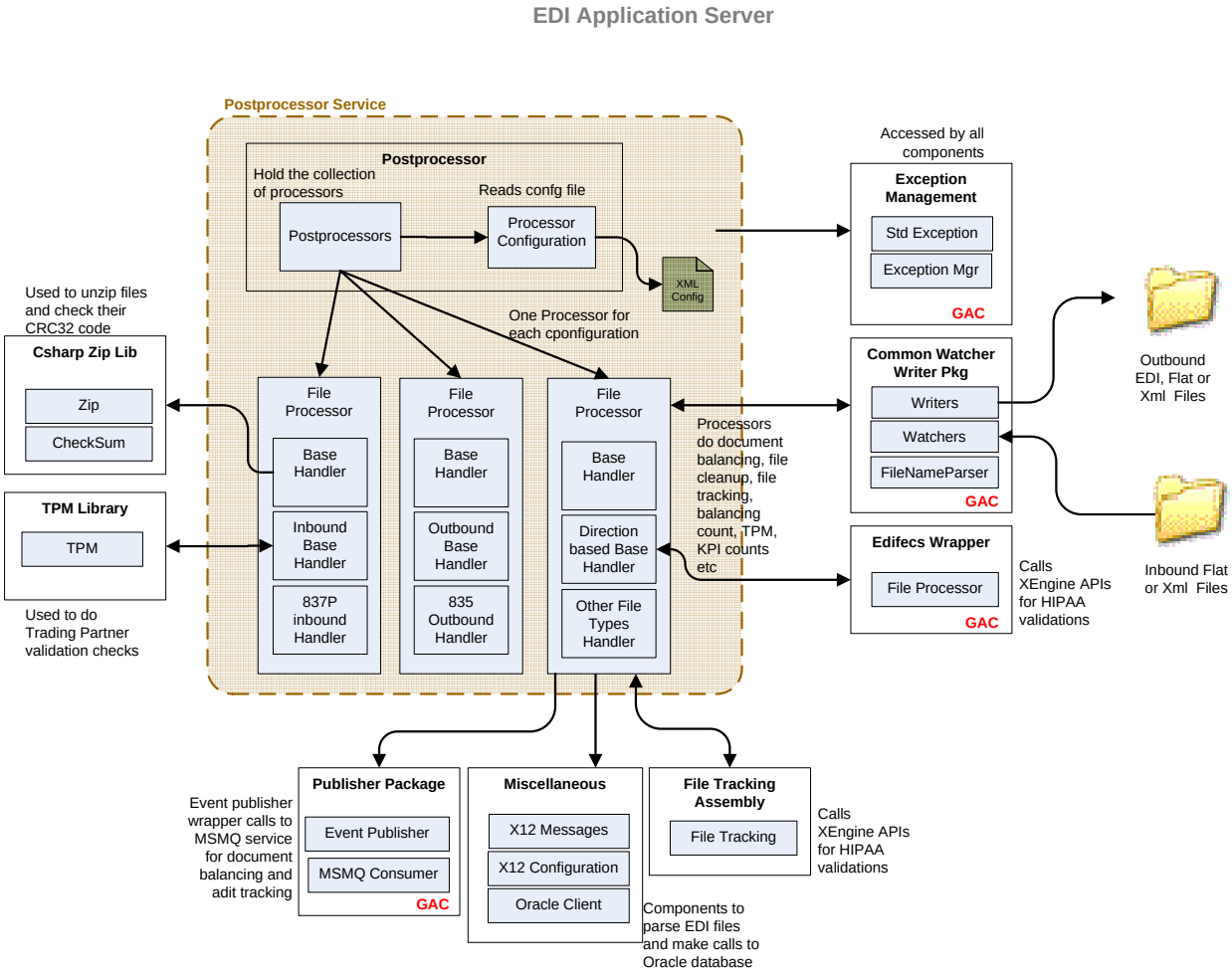
All outbound X12 documents passed to Edifecs XEngine for HIPAA validations. This is done by calling Edifecs Wrapper which in turn calls XEngine. XEngine is also responsible for transforming XDATA (proprietary Edifecs Xml) to EDI X 12 formats. Another step that happens in XEngine is that X12 documents are broken into smaller files depending upon the Split Rules set in XEngine configurations. Currently for KY MMIS all claims are split at claim level leading to one XDATA file for each claim while all other document types are split at ST-SE transaction level leading to one XDATA file for each ST-SE transaction.

26.1.1.3 File Size Optimization

Post-Processor Service is designed to optimize the size of large outbound files before these are passed to Edifecs XEngine. In Pre Processor If the file size is bigger than the pre-configured size for that document type then the file is split into smaller size files and renamed with the file part information (for example: 1 of 10 which would mean part of the file which is split into 10 parts). This leads to faster processing of files in BizTalk server. These split files are processed in BizTalk server are received in post processor and merged into single file before converting to EDI X12 formats in Edifecs XEngine.

26.2 Post Processor Service Component Diagram

Below diagram shows all the components used by Post Processor Service and how to interact with each other. In later section this document will go over each project related to Post Processor solution in detail.

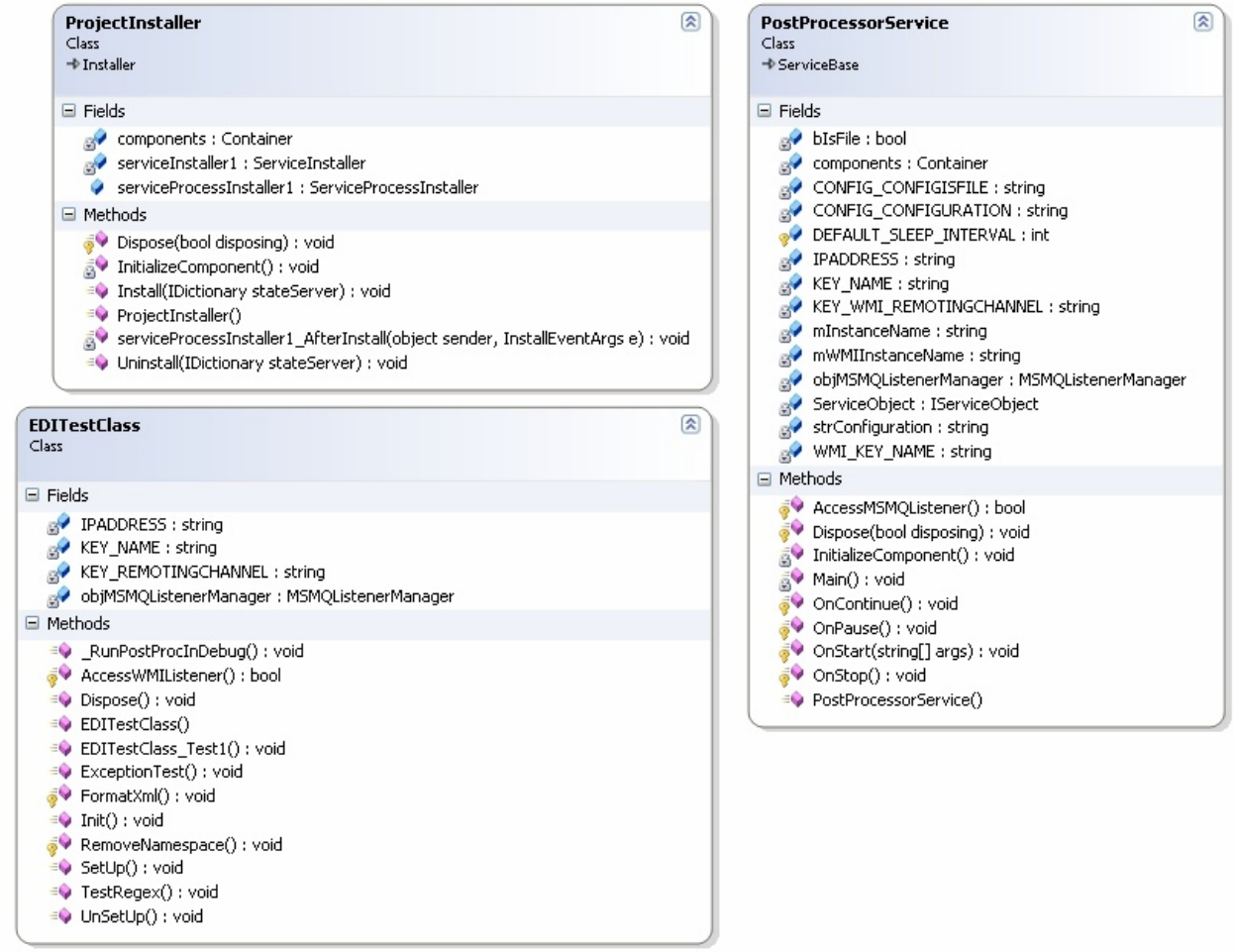


26.3 Projects Description

This section will explain all the projects involved in development of the Post-Processor Service. It will also go in to explaining the individual projects, their classes and the methods involved in supporting the required functionality of the Post-Processor Service.

26.3.1 Post-Processor Service Project

Post-Processor Service project is the main project which holds the actual windows service component. This windows service component is responsible for calling other assembly projects to perform its designed functions. This project also incorporates the test harness class (EDITestClass) which allows the service to be run in a debug mode using NUnit tool.



26.3.1.1 Post-Processor Service Class

Post-Processor Service is the file which has the main class, “PostProcessorService”, of the “PostProcessorService” project where the windows service gets the control to Start, Stop, Pause or Restart itself.

File Name: PostProcessorService.cs	**Class Name: PostProcessorService**
Method Name	**Description**
PostProcessorService	This is the default constructor of the class

File Name: PostProcessorService.cs	**Class Name: PostProcessorService**
Method Name	**Description**
Main	This is the entry point for the service control. PostProcessorService is instantiated in this method and is set to run.
InitializeComponent	This method is called by the constructor and is used to initialize the name of the class.
OnStart	This is the method which is called to start the service. This method reads the App.config to get the configuration file and passes it to Postprocessors class of Postprocessor project. It also calls the AccessWMIListenerService to check if the MSMQ Listener Service is up and running. Finally it then calls the Start method of the instantiated Postprocessors class.
OnStop	Before stopping, this method calls the above instantiated Postprocessors object Stop method. If successful it stops running the Post-Processor Service.
AccessMSMQListener	This method reads the configurations require to remote into MSMQ Listener Service and then remotes into it to check if the service is running and is accepting remote calls.
Dispose	This method disposes off the service object.

26.3.1.2 Project Installer Class

This class is required to create a proper deployment (msi) package for the Post-Processor service. This msi package is used to deploy the service in run time environment.

File Name: ProjectInstaller.cs	**Class Name: ProjectInstaller**
Method Name	**Description**
ProjectInstaller	This is the default constructor of the class and it calls InitializeComponent.
InitializeComponent	This is the designer generated code and should not be modified by developer using code editor.
Install	This method is called during install process of the windows service deployment package and responsible for installing the service component. This method is also used to setup the install properties of the service like service Name, service description and registry settings.
Uninstall	This method is called during uninstall process of the windows service. This method is used to remove the service related registry settings and finally uninstalls the service component.

File Name: ProjectInstaller.cs	**Class Name: ProjectInstaller**
Method Name	**Description**
Dispose	This method disposes off the service object.

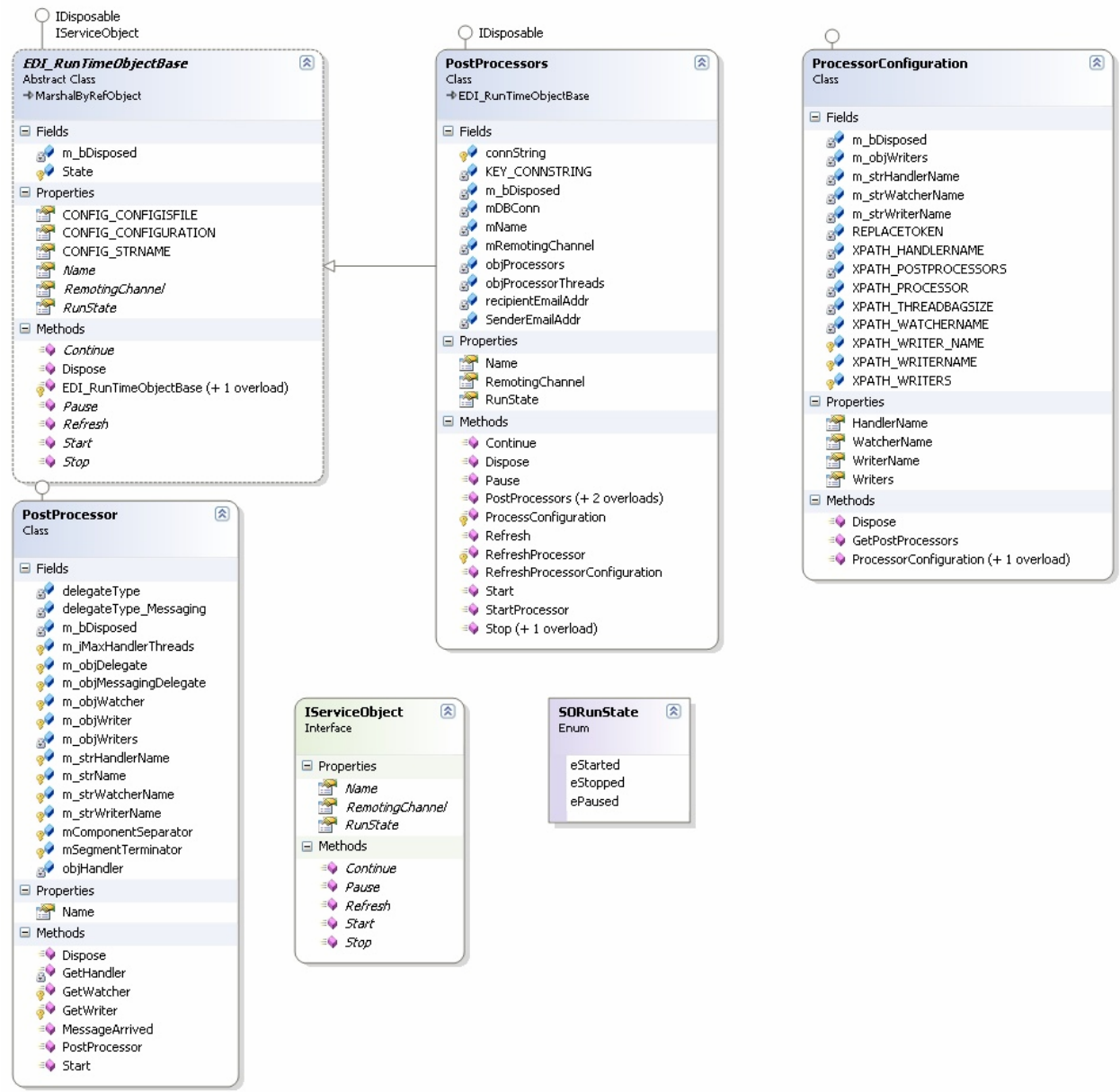
26.3.1.3 EDI Test Class

This class is a Text Fixture class of the NUnit tool and helps the developer to run the windows service in the debug mode.

File Name: EDITestClass.cs	**Class Name: EDITestClass**
Method Name	**Description**
EDITestClass_Test	This is the NUnit test that is run using the NUnit tool to start the Post-Processor Service in the Debug mode. This replaces the need to install the service component. It performs the same functionality of getting the configuration file and then it calls the Postprocessors class Start method.

26.3.2 Postprocessor Project

Postprocessor Project is called by the PostProcessorService class and is responsible for setting up the runtime threads and handlers of the service. In the Postprocessors project Post-processor Service accesses the configuration files and reads the configuration attributes needed to initializes Postprocessor handlers for all supported document types. After reading the configuration values it starts each Post-Processor Handler in a separate thread. This project is also responsible for cleaning up all the handlers before Post-Processor Service shuts down.



26.3.2.1 Postprocessors Class

Postprocessors Class implements EDI_RunTimeObjectBase class and IDisposable interface. Postprocessors class reads the configuration values from App.config (file name is PostProcessorservice.exe.config during runtime), ServiceConfigl.xml files to get configuration values which are common to the entire service. Based on those values it creates new Pre-Processor thread for each configuration and starts these threads. While starting new thread it passes the value of which directory to monitor and what process handler to use. When the Postprocessors class gets a call to stop the Pre-Processor Service then it is also responsible to kill each Pre-Processor handler thread which will finally stop the service.

File Name: Postprocessors.cs	**Class Name: Postprocessors**
Method Name	**Description**
Postprocessors	This is the default constructor of the class. This method calls ProcessConfiguration method.
Start	This method sets the service state to “Started” and calls StartProcessor method.
StartProcessor	This method goes over each Postprocessor instance in the Postprocessor sorted list and calls its Start method which starts the new thread.
Stop	This method sets the status of the Postprocessor class to “Stopped”.
ProcessConfiguration	This method calls Configuration Parser class which reads and parses the ServiceConfigl.xml file values. It also reads other common configuration values such as connection string. After reading the ServiceConfigl.xml file it goes over each Post-Processor configuration and calls RefreshProcessor method.
Refresh	This method sets the status of the Postprocessor class to “Started”.
RefreshProcessor	In this method service control creates a new Postprocessor class instance’s Start method in a new thread. After creating new instance thread it puts it in the synchronized sorted list. If a particular Postprocessor object already exists it is removed and a new instance is added.
Dispose	This method disposes off the service object.

26.3.2.2 Postprocessor Class

Postprocessors Class implements EDI_RunTimeObjectBase class and IDisposable interface. Postprocessors class reads the configuration values from App.config (file name is PostProcessorservice.exe.config during runtime), ServiceConfigl.xml files to get configuration values which are common to the entire service. Based on those values it creates new Post-Processor thread for each configuration and starts these threads. While starting new thread it passes the value of which directory to monitor and what process handler to use. When the Postprocessors class gets a call to stop the Post-Processor Service then it is also responsible to kill each Post-Processor handler thread which will finally stop the service.

File Name: PostProcessor.cs	**Class Name: Postprocessor**
Method Name	**Description**
Postprocessor	This is the default constructor of the class. This method calls ProcessConfiguration class which parses the processor handler, watcher name and writer name for this Postprocessor instance. Then it calls the GetHandler, GetWatcher and GetWriter methods to returns respective object instances to this method. It also instantiates the new delegates MessageArrived of Postprocessor class and ExecuteHandler of the ProcessorHandlerBase class in ProcessorHandler project. It passes values to ProcessorHandlerBase class by calling its delegates dlgProcessorHandler and dlgProcessorHandlerMessaging.
Start	This method starts the watcher object so it starts monitoring the configured directory at the configured polling interval.
GetHandler	This method reads the configuration to get the configured class name of the handler and based on that value it creates the instance of a specific handler and returns it to Postprocessor constructor.
GetWatcher	This method reads the configurations to get the configured class name of the watcher and based on that value it creates the instance of a specific watcher and returns it to Postprocessor constructor.
GetWriter	This method reads the configuration to get the configured class name of the writer and based on that value it creates the instance of a specific writer and returns it to Postprocessor constructor.
MessageArrived	This method calls the dlgProcessorHandler delegate and passes the file location to it. This starts the processing of a file in Post-Processor service by a specific handler and its base. In return to the delegate call it gets back the file location of the processed file. If this location is not empty and if the file exists then it is moved by the writer object to the final location.

File Name: PostProcessor.cs	Class Name: Postprocessor
Method Name	**Description**
Dispose	This method disposes off the service object.

26.3.2.3 Processor Configuration Class

This is a static class which is specifically made to read the configuration file and pass it to the Directory Manager class.

File Name: ProcessorConfiguration.cs	Class Name: ProcessorConfiguration
Method Name	**Description**
ProcessorConfiguration	This is the default constructor of the class. This is the method which actually reads the App.config file and ServiceConfigl.xml file. After reading the ServiceConfigl.xml file it read the handler, watcher and writer configurations.
GetPostProcessors	This method loads the ServiceConfigl.xml configuration file and reads all the “Postprocessor” record configurations, adds them to an ArrayList collection object and returns back the ArrayList.
Dispose	This method disposes off the service object.

26.3.2.4 IService Object Interface

This is the interface which is implemented by Directory Manager and EDI Runtime Object Base classes. This class is present in other EDI services also so the implementation of all EDI services is consistent.

File Name: ConfigurationParser.cs	Class Name: ConfigurationParser
Method Name	**Description**
Start	Interface method.
Stop	Interface method.
Continue	Interface method.
Refresh	Interface method.
Pause	Interface method.

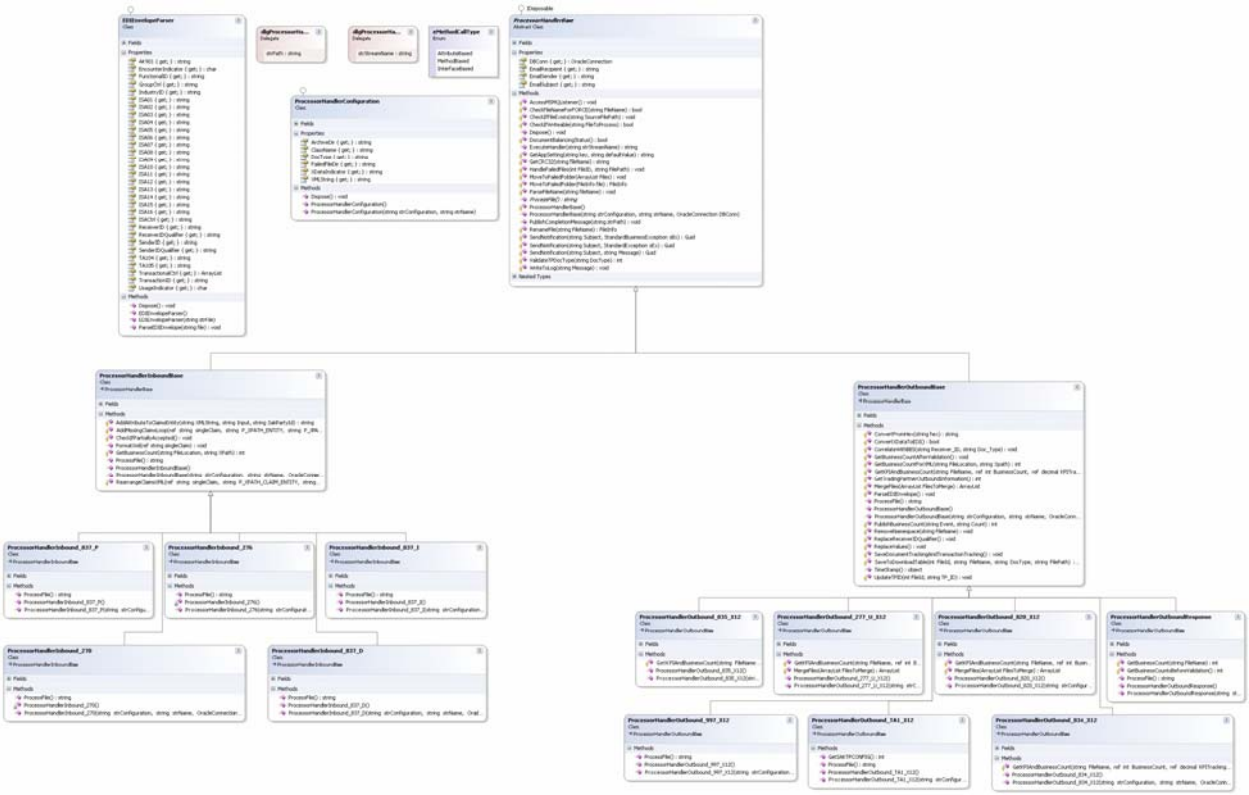
26.3.2.5 EDI Runtime Object Base Class

This is an abstract class which implements MarshalByRefObject class, IDisposable interface and IServiceObject interface. The purpose of this class is to give a layer to add EDI subsystem specific code.

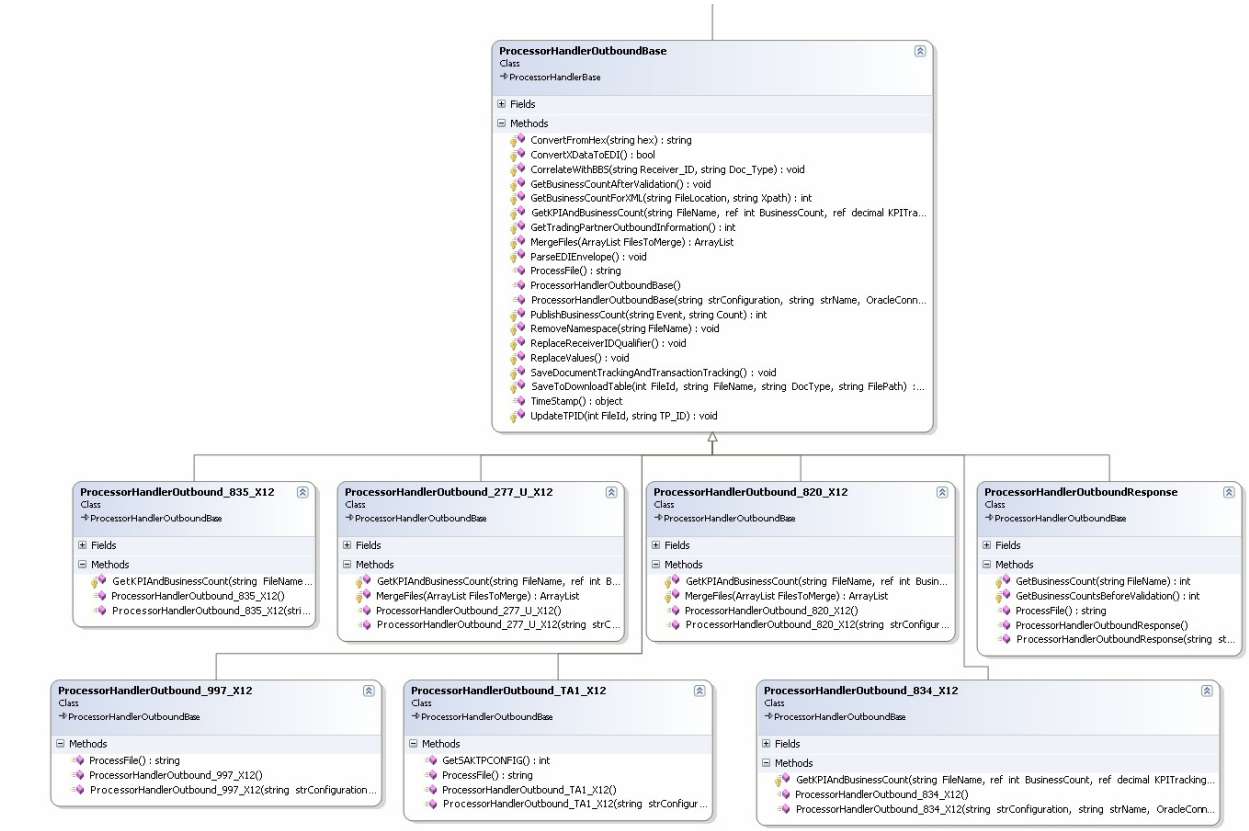
File Name: EDI_RunTimeObjectBase.cs	**Class Name: EDI_RunTimeObjectBase**
Method Name	**Description**
Start	Abstract method.
Stop	Abstract method.
Continue	Abstract method.
Refresh	Abstract method.
Pause	Abstract method.
Dispose	This method disposes off the service object.

26.3.3 Processor Handler Project

This is the project which is actually responsible for processing a file and implementing all the business logic designed for Post-Processor.



EDIEnvelopeParser
Class
Fields
Properties
AK901 { get; } : string
EncounterIndicator { get; } : char
FunctionalID { get; } : string
GroupCtrl { get; } : string
IndustryID { get; } : string
ISA01 { get; } : string
ISA02 { get; } : string
ISA03 { get; } : string
ISA04 { get; } : string
ISA05 { get; } : string
ISA06 { get; } : string
ISA07 { get; } : string
ISA08 { get; } : string
ISA09 { get; } : string
ISA10 { get; } : string
ISA11 { get; } : string
ISA12 { get; } : string
ISA13 { get; } : string
ISA14 { get; } : string
ISA15 { get; } : string
ISA16 { get; } : string
ISACtrl { get; } : string
ReceiverID { get; } : string
ReceiverIDQualifier { get; } : string
SenderID { get; } : string
SenderIDQualifier { get; } : string
TA104 { get; } : string
TA105 { get; } : string
TransactionalCtrl { get; } : ArrayList
TransactionID { get; } : string
UsageIndicator { get; } : char
Methods
Dispose() : void
EDIEnvelopeParser()
EDIEnvelopeParser(string strFile)
ParseEDIEnvelope(string file) : void
dlgProcessorHa...
Delegate
strPath : string
dlgProcessorHa...
Delegate
strStreamName : string
eMethodCallType
Enum
AttributeBased
MethodBased
InterfaceBased
ProcessorHandlerConfiguration
Class
Fields
Properties
ArchiveDir { get; } : string
ClassName { get; } : string
DocType { get; } : string
FailedFileDir { get; } : string
XDataIndicator { get; } : string
XMLString { get; } : string
Methods
Dispose() : void
ProcessorHandlerConfiguration()
ProcessorHandlerConfiguration(string strConfiguration, string strName)
IDisposable
ProcessorHandlerBase
Abstract Class
Fields
Properties
DBConn { get; } : OracleConnection
EmailRecipient { get; } : string
EmailSender { get; } : string
EmailSubject { get; } : string
Methods
AccessMSMQListener() : void
CheckFileNameForFORCE(string FileName) : bool
CheckIfFileExists(string SourceFilePath) : void
CheckIfWriteable(string FileToProcess) : bool
Dispose() : void
DocumentBalancingStatus() : bool
ExecuteHandler(string strStreamName) : string
GetAppSetting(string key, string defaultValue) : string
GetCRC32(string fileName) : string
HandleFailedFiles(int FileID, string FilePath) : void
MoveToFailedFolder(ArrayList Files) : void
MoveToFailedFolder(FileInfo file) : FileInfo
ParseFileName(string fileName) : void
ProcessFile() : string
ProcessorHandlerBase()
ProcessorHandlerBase(string strConfiguration, string strName, OracleConnection DBConn)
PublishCompletionMessage(string strPath) : void
RenameFile(string FileName) : FileInfo
SendNotification(string Subject, StandardBusinessException stEx) : Guid
SendNotification(string Subject, StandardException sEx) : Guid
SendNotification(string Subject, string Message) : Guid
ValidateTPDocType(string DocType) : int
WriteToLog(string Message) : void
Nested Types
ProcessorHandlerInboundBase
Class
ProcessorHandlerBase
Fields
Methods
AddAttributeToClaimsEntity(string XMLString, string Input, string SakPartyId) : string
AddMissingClaimsLoop(ref string singleClaim, string P_XPATH_ENTITY, string P_XPA...
CheckIfPartiallyAccepted() : void
FormatXml(ref string singleClaim) : void
GetBusinessCount(string FileLocation, string XPath) : int
ProcessFile() : string
ProcessorHandlerInboundBase()
ProcessorHandlerInboundBase(string strConfiguration, string strName, OracleConnec...
RearrangeClaimsXML(ref string singleClaim, string P_XPATH_CLAIM_ENTITY, string...
ProcessorHandlerInbound_837_P
Class
ProcessorHandlerInboundBase
Fields
Methods
ProcessFile() : string
ProcessorHandlerInbound_837_P()
ProcessorHandlerInbound_837_P(string strConfigu...
ProcessorHandlerInbound_276
Class
ProcessorHandlerInboundBase
Fields
Methods
ProcessFile() : string
ProcessorHandlerInbound_276()
ProcessorHandlerInbound_276(string strConfigurat...
ProcessorHandlerInbound_837_I
Class
ProcessorHandlerInboundBase
Fields
Methods
ProcessFile() : string
ProcessorHandlerInbound_837_I()
ProcessorHandlerInbound_837_I(string strConfiguration...
ProcessorHandlerInbound_270
Class
ProcessorHandlerInboundBase
Fields
Methods
ProcessFile() : string
ProcessorHandlerInbound_270()
ProcessorHandlerInbound_270(string strConfiguration, string strName, OracleConnection...
ProcessorHandlerInbound_837_D
Class
ProcessorHandlerInboundBase
Fields
Methods
ProcessFile() : string
ProcessorHandlerInbound_837_D()
ProcessorHandlerInbound_837_D(string strConfiguration, string strName, Orad...



26.3.3.1 ProcessorHandlerConfiguration Class

ProcessorHandlerConfiguration class implements IDisposable interface and is used in post processor class for reading the “PostProcessorservice.exe.config” xml configuration file. This class is responsible for reading the handler configuration information from the xml configuration file.

File Name: ProcessorHandlerConfiguration.cs	**Class Name: ProcessorHandlerConfiguration**
Method Name	**Description**
ProcessorHandlerConfiguration	This is the overloaded constructor of the ProcessorHandlerConfiguration class. It takes PostProcessorservice.exe.config xml file, handler name as input parameter and has code for reading the config file and returning handler configuration information.
Dispose	This method disposes of the service object.

26.3.3.2 ProcessorHandlerBase Class

ProcessorHandlerBase is a base class for all handler classes defined in ProcessorHandler project. This class is responsible for initializing all variables required for processing the EDI files using handler classes. ProcessorHanlderBase implements the IDisposable interface.

File Name: ProcessorHandlerBase.cs	**Class Name: ProcessorHandlerBase**
Method Name	**Description**
ProcessorHandlerBase	This is the overloaded constructor of the ProcessorHandlerBase class. It takes PostProcessorservice.exe.config xml file, handler name as input parameter and makes call to ProcessorHandlerConfiguration class for reading handler configuration information.
ExecuteHandler	This method takes file to be processed has input parameter and calls the ParseFileName method. After parsing FileID and Eventid information from filename to be processed calls the DocumentBalancingStatus method. This is the main method for handling files in the post processor.
ParseFileName	This method takes file name to be processed has input parameter, calls the EDIFileName.ParseFileName_PostProcessor method and gets the FileID, mDocTypeFromFileName, EventID, mFilePartInfo, mFileName, mExtension information from the input file name.
CheckIfWriteable	This method takes file name to be processed has input parameter and checks if input file can be accessible for writing using Filestream.canwrite () method call.
GetCRC32	This method takes file name to be processed has input parameter, calls the ICSharpcode.sharziplib.checksums.crc32 class for getting the crc32 code.
AccessMSMQListener	This method is used to get the reference to the MSMQListenerManager class instance using reflection.
PublishCompletionMessage	This method takes file name as input parameter and calls the Publisher.UpdateFileStatus, Publisher.PublishWMIEvent for updating the file completion and File Audit Status.
GetAppSetting	GetAppSetting method takes Key, default Value as input parameters and gets the value by calling ConfigurationSettings.AppSettings using the passed key.
ValidateTPDocType	ValidateTPDocType method takes DocType as input parameter and calls the TP Data base call to validate

File Name: ProcessorHandlerBase.cs	**Class Name: ProcessorHandlerBase**
Method Name	**Description**
	Trading partner by passing ReceiverID, DocType and FileDirection.
SendNotification	This method is implemented with multiple combinations of input parameters. Each SendNotification method creates the objects of Notification, Standard Exception class and publishes them by creating object of Exception Manager class. All overloaded SendNotification methods returns Guid.
RenameFile	This method takes file name as input parameter, Calls the CheckIfFileExists by passing rename file name parameter and renames the input file name to FileID_ DocTypeFromFileName_ FileName + mFile.Extension format in the defined Directory.
DocumentBalancingStatus	This method is used in ExecuteHandler method. This method calls the MSMQListenerManager. GetBalancingCountStatus by passing FileID and get the BalancingcountStatus value. Depending upon Balancing count Status Post Processor will call the Publisher.UpdateFileStatus, Publisher.PublishWMIEvent updates file status and file audit.
HandleFailedFiles	This method takes FileID and file Path as input parameters and calls the MoveToFailedFolder () method.
MoveToFailedFolder	This method takes File info has input parameter and moves the passed file to failed folder. Once moving files to failed folder this method publishes “PASSED_TO_FAILEDFOLDER” event.
CheckFileNameForFORCE	This method takes file name as input parameter and calls Regex.Match, makes search if file name has “.Force.” Key word returns true/false depending upon found results.
CheckIfFileExists	This method takes file name as input parameter and check if file exist in the specified location. If file exist the method renames the passed file name to “sourcefilename + datetime “format.
WriteToLog	This method takes message as input parameter and logs the passed message to log file.
Dispose	This method disposes of the service object.

26.3.3.3 ProcessorHandlerInbound_270 Class

ProcessorHandlerInbound_270 class is used while processing the 270 Interchange eligVerification xml file. This class inherits ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_270.cs	**Class Name: ProcessorHandlerInbound_270**
Method Name	**Description**
ProcessorHandlerInbound_270	This is the default constructor of the ProcessorHandlerInbound_270 class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerInboundBase class. This method calls the ProcessorHandlerInboundBase. GetBusinessCount () method, gets the business count and calls ProcessorHandlerInboundBase. ProcessFile () method.

26.3.3.4 ProcessorHandlerInbound_276 Class

ProcessorHandlerInbound_276 class is used while processing the 276 interchange claimStatus276 xml file. This class inherits ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_276.cs	**Class Name: ProcessorHandlerInbound_276**
Method Name	**Description**
ProcessorHandlerInbound_276	This is the default constructor of the ProcessorHandlerInbound_276 class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerInboundBase class. This method calls the ProcessorHandlerInboundBase. GetBusinessCount () method, gets the business count and calls ProcessorHandlerInboundBase. ProcessFile () method.

26.3.3.5 ProcessorHandlerInbound_278 Class

ProcessorHandlerInbound_278 class is used while processing the 278 Interchange priorAuths278 xml file. This class inherits ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_278.cs	**Class Name: ProcessorHandlerInbound_278**
Method Name	**Description**
ProcessorHandlerInbound_278	This is the default constructor of the ProcessorHandlerInbound_278 class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerInboundBase class. This method calls the ProcessorHandlerInboundBase.

File Name: ProcessorHandlerInbound_278.cs	**Class Name: ProcessorHandlerInbound_278**
Method Name	**Description**
	GetBusinessCount () method, gets the business count and calls ProcessorHandlerInboundBase. ProcessFile () method.

26.3.3.6 ProcessorHandlerInbound_837_D Class

ProcessorHandlerInbound_837_D class is used while processing the 837 Dental interchange claim list xml file. This class inherits ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_837_D.cs	**Class Name: ProcessorHandlerInbound_837_D**
Method Name	**Description**
ProcessorHandlerInbound_837_D	This is the default constructor of the ProcessorHandlerInbound_837_Dclass.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerInboundBase class. This method is responsible for reading the 837 Dental interchange claim list xml file, rearrange the claimlist xml to make proper xml file and count the no. of claims and calls ProcessorHandlerInboundBase.ProcessFile() method.

26.3.3.7 ProcessorHandlerInbound_837_I Class

ProcessorHandlerInbound_837_I class is used while processing the 837 Institutional interchange claim list xml file. This class inherits ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_837_I.cs	**Class Name: ProcessorHandlerInbound_837_I**
Method Name	**Description**
ProcessorHandlerInbound_837_I	This is the default constructor of the ProcessorHandlerInbound_837_I class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerInboundBase class. This method is responsible for reading the 837 Institutional interchange claim list xml file, rearrange the claimlist xml to make proper xml file and count the no. of claims and calls ProcessorHandlerInboundBase.ProcessFile() method.

26.3.3.8 ProcessorHandlerInbound_837_P Class

ProcessorHandlerInbound_837_P class is used while processing the 837 Professional interchange claim list xml file. This class inherits ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_837_P.cs	**Class Name: ProcessorHandlerInbound_837_P**
Method Name	**Description**
ProcessorHandlerInbound_837_P	This is the default constructor of the ProcessorHandlerInbound_837_P class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerInboundBase class. This method is responsible for reading the 837 Professional interchange claim list xml file, rearrange the claimlist xml to make proper xml file and count the no. of claims and call ProcessorHandlerInboundBase.ProcessFile() method.

26.3.3.9 ProcessorHandlerInbound_NSF Class

ProcessorHandlerInbound_NSF class is used while processing the NSF 837 Professional interchange claim list xml file. This class inherits ProcessorHandlerInboundBase class.

File Name: ProcessorHandlerInbound_NSF.cs	**Class Name: ProcessorHandlerInbound_NSF**
Method Name	**Description**
ProcessorHandlerInbound_NSF	This is the default constructor of the ProcessorHandlerInbound_NSF class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerInboundBase class. This method is responsible for reading the NSF 837 Professional interchange claim list xml file, rearrange the claimlist xml to make proper xml file and count the no. of claims and call ProcessorHandlerInboundBase.ProcessFile() method.

26.3.3.10 ProcessorHandlerInboundBase Class

ProcessorHandlerInboundBase class is base inbound handler class used for processing inbound EDI Transactions.

File Name: ProcessorHandlerInboundBase.cs	**Class Name: ProcessorHandlerInboundBase**
Method Name	**Description**
ProcessorHandlerInboundBase	This is the default constructor of the ProcessorHandlerInbound_NSF class.
ProcessFile	This method is responsible for publishing "COUNT_BEFORE_SENDING_TO_IC" event with EventID and Balancing count information. This method even calls

File Name: ProcessorHandlerInboundBase.cs	**Class Name: ProcessorHandlerInboundBase**
Method Name	**Description**
	CheckIfPartiallyAccepted () method.
GetBusinessCount	This method takes file location and Xpath string as input parameters and returns count of matching Xpath nodes as Balancing Count information.
RearrangeClaimsXML	This method is used in Re arranging 837 claimlist xmls ClnEntity Node information.
AddAttributeToClaimsEntity	This method is used in Re arranging 837 claim List xmls ClnEntity Node information
AddMissingClaimsLoop	This method is used in Re arranging 837 claim List xmls ClnEntity Node information
FormatXml	This method is used in Re arranging 837 claim List xmls ClnEntity Node information
CheckIfPartiallyAccepted	This method is used to check if Balancing count before sending to Interchange is less than Balance count after BizTalk Transformation and if it is publishes the "PARTIALLY_ACCEPTED" file Status.

26.3.3.11 ProcessorHandlerOutbound_277_U_X12 Class

ProcessorHandlerOutbound_277_U_X12 class is used while processing the out bound 277 EDI X12 file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_277_U_X12.cs	**Class Name: ProcessorHandlerOutbound_277_U_X12**
Method Name	**Description**
ProcessorHandlerOutbound_277_U_X12	This is the default constructor of the ProcessorHandlerOutbound_277_U_X12 class.
GetKPIAndBusinessCount	GetKPIAndBusinessCount is used to read the processing file name using stream Reader and calculate the Business count by calling Regex.Match method.
MergeFiles	MergeFiles method takes FilesToMerge Arraylist as the input parameter and merges received files to single file.

26.3.3.12 ProcessorHandlerOutbound_278_X12 Class

ProcessorHandlerOutbound_278_X12 class is used while processing the out bound 278 EDI X12 file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_278_X12.cs	**Class Name: ProcessorHandlerOutbound_278_X12**
Method Name	**Description**
ProcessorHandlerOutbound_278_X12	This is the default constructor of the ProcessorHandlerOutbound_277_U_X12 class.
GetKPIAndBusinessCount	GetKPIAndBusinessCount is used to read the processing file using stream Reader and calculate the Business count by calling Regex.Match method.

26.3.3.13 ProcessorHandlerOutbound_820_X12 Class

ProcessorHandlerOutbound_820_X12 class is used while processing the out bound 820 EDI X12 file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_820_X12.cs	**Class Name: ProcessorHandlerOutbound_820_X12**
Method Name	**Description**
ProcessorHandlerOutbound_820_X12	This is the default constructor of the ProcessorHandlerOutbound_820_X12 class.
GetKPIAndBusinessCount	GetKPIAndBusinessCount is used to read the processing file using stream Reader and calculate the Business count by calling Regex.Match method.
MergeFiles	MergeFiles method takes FilesToMerge Arraylist as the input parameter and merges received files to a single file.

26.3.3.14 ProcessorHandlerOutbound_834_X12 Class

ProcessorHandlerOutbound_834_X12 class is used while processing the out bound 834 EDI X12 file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_834_X12.cs	**Class Name: ProcessorHandlerOutbound_834_X12**
Method Name	**Description**
ProcessorHandlerOutbound_834_X12	This is the default constructor of the ProcessorHandlerOutbound_834_X12 class.
GetKPIAndBusinessCount	GetKPIAndBusinessCount is used to read the processing file using stream Reader and calculate the Business count by calling Regex.Match method.

26.3.3.15 ProcessorHandlerOutbound_835_X12 Class

ProcessorHandlerOutbound_835_X12 class is used while processing the out bound 835 EDI X12 file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_835_X12.cs	**Class Name: ProcessorHandlerOutbound_835_X12**
Method Name	**Description**
ProcessorHandlerOutbound_835_X12	This is the default constructor of the ProcessorHandlerOutbound_835_X12 class.
GetKPIAndBusinessCount	GetKPIAndBusinessCount is used to read the processing file using stream Reader and calculate the Business count by calling Regex.Match method.

26.3.3.16 ProcessorHandlerOutbound_997_X12 Class

ProcessorHandlerOutbound_997_X12 class is used while processing the out bound 997 EDI X12 file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_997_X12.cs	**Class Name: ProcessorHandlerOutbound_997_X12**
Method Name	**Description**
ProcessorHandlerOutbound_997_X12	This is the default constructor of the ProcessorHandlerOutbound_997_X12 class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerOutboundBase class. This method is responsible for reading the 997 EDI X12 file, call ParseEDIEnvelope () method. This method publishes events required for File Tracking, functional Acknowledgement and File Audit.

26.3.3.17 ProcessorHandlerOutbound_NCPDP Class

ProcessorHandlerOutbound_NCPDP class is used while processing the out bound NCPDP file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_NCPDP.cs	**Class Name: ProcessorHandlerOutbound_NCPDP**
Method Name	**Description**
ProcessorHandlerOutbound_NCPDP	This is the default constructor of the ProcessorHandlerOutbound_NCPDP class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerOutboundBase class. This method is responsible for reading the NCPDP Configuration and move the processing file to configured location where NCPDP Batch Translation is configured to watch. This method calls NCPDPBatchResponseBuilder.

File Name: ProcessorHandlerOutbound_NCPDP.cs	**Class Name: ProcessorHandlerOutbound_NCPDP**
Method Name	**Description**
	ProcessXMLFile () method for translating and does Business count after validation.

26.3.3.18 ProcessorHandlerOutbound_NSF_Response Class

ProcessorHandlerOutbound_NSF_Response class is used while processing the NSF response file. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutbound_NSF_Response.cs	**Class Name: ProcessorHandlerOutbound_NSF_Response**
Method Name	**Description**
ProcessorHandlerOutbound_NSF_Response	This is the default constructor of the ProcessorHandlerOutbound_NSF_Response class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerOutboundBase class. This method is responsible for moving NSF response file to the submitter and publishing events required for Tracking, File Audit, Correlating the response file to main processing NSF file in File Tracking.

26.3.3.19 ProcessorHandlerOutbound_TA1_X12 Class

ProcessorHandlerOutbound_TA1_X12 class is used while processing the TA1 X12 file. This class inherits ProcessorHandlerOutboundBase class.

File Name: DeleteSplitFile	**Class Name: ProcessorHandlerOutbound_TA1_X12**
Method Name	**Description**
ProcessorHandlerOutbound_TA1_X12	This is the default constructor of the ProcessorHandlerOutbound_TA1_X12 class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerOutboundBase class. This method is responsible for parsing TA1 EDI X12 file, moving the TA1 to the configured folder location and publishing the required events for saving functional acknowledgements, File Tracking Events, File Audit and Correlate TA1 acknowledgements to the main processing file.
GetSAKTPCONFIG	This method is responsible for getting the trading partner configuration from TP Database.

26.3.3.20 ProcessorHandlerOutboundBase Class

ProcessorHandlerOutboundBase class is the base class for processing Outbound EDI Transactions. ProcessorHandlerOutboundBase class inherits ProcessorHandlerBase class.

File Name: ProcessorHandlerOutboundBase.cs	**Class Name: ProcessorHandlerOutboundBase**
Method Name	**Description**
ProcessorHandlerOutboundBase	This is the default constructor of the ProcessorHandlerOutboundBase class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerOutboundBase class. This method is responsible for parsing TA1 EDI X12 file, move the TA1 files to the configured folder location and publish the required events for saving functional acknowledgements, File Tracking Events, File Audit and Correlate TA1 acknowledgements to the main processing file.
ConvertFromHex	This method takes hex string input parameter and is responsible for converting the received value to hexadecimal value.
SaveDocumentTrackingAndTransactionTracking	This method is responsible for making document tracking and document transaction tracking.
GetTradingPartnerOutboundInformation	This method is responsible for querying the trading partner information. This method makes objTPM.GetTPMInformation () method call by passing SAK_TP and ReceiverID of the processing outbound file.
ConvertXDataToEDI	This method is responsible for converting outbound xml version of X12 file to EDI X12 file by calling Edifecs Wrapper class.
ParseEDIEnvelope	This method is responsible for parsing the EDI envelope information of the processing EDI file. This method makes object of EDIEnvelopeParser class by passing the file name and reads the EDI envelope information to the local variables.
ReplaceReceiverIDQualifier	This method is responsible for replacing the ISA ReceiverID qualifier to the trading partner configured Receiver ID Qualifier value in the processing edi file.
GetBusinessCountAfterValidation	This is virtual method which is overridden in NCPDP handler class. This method calls GetKPIAndBusinessCount () for getting the KPI value and Business count values and publishes COUNT_AFTER_VALIDATION, KPI events.
GetKPIAndBusinessCount	This is virtual method which is overridden in individual handler classes. This method takes FileName,

File Name: ProcessorHandlerOutboundBase.cs	**Class Name: ProcessorHandlerOutboundBase**
Method Name	**Description**
	BusinessCount, KPITrackingValue, CDE_Event_Key as input parameters.
PublishBusinessCount	This method takes Event and Count as input parameters and is responsible for making database call for inserting the business count values.
RemoveNamespace	This method takes file name as input parameter and is responsible for removing name spaces from the input file name.
Timestamp	This method returns the formatted date using DateTime.Now.ToString("dd-MMM-yy hh.mm.ss.fffffff tt") method call.
SaveToDownloadTable	This method takes FileID, FileName, DocType, and FilePath as input parameters and is responsible for making database calls for inserting values into T_DOWNLOAD table.
CorrelateWithBBS	This method takes Receiver_ID, Doc_Type as input parameters and checks if there is matching ID_PROVIDER and CDE_IDENTIFICATION in T_BBS_CORRELATION table. If there is match found this method replaces the ISA08 with retrieved ID_ISA_RECEIVER value in the processing file.
GetBusinessCountForXML	This method takes FileLocation, Xpath as string input parameters and returns the matching business count in xml using DocumentElement.SelectNodes (Xpath) method call.
MergeFiles	This method takes FilesToMerge ArrayList as input parameter and merges the files to single file.
UpdateTPID	This method takes FileId, TP_ID as input parameters and is responsible for making update database call, updates the T_FILE_TRACK table with ID_ISA_IDENTIFIER = :P_TP_ID where SAK_FILE_TRACK = : FileId.

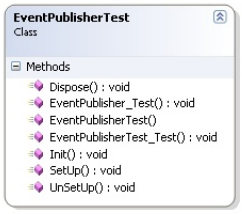
26.3.3.21 ProcessorHandlerOutboundResponse Class

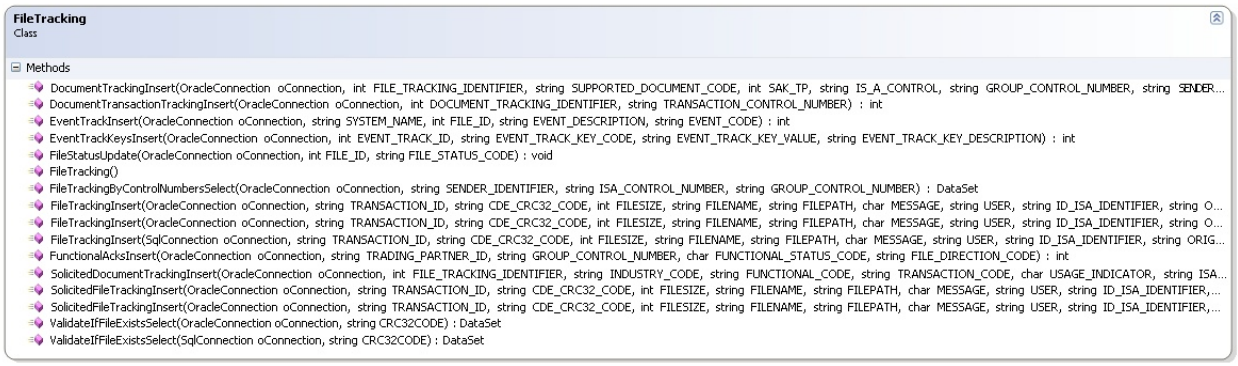
ProcessorHandlerOutboundResponse class is used while processing the 271, 277 X12 response file transactions. This class inherits ProcessorHandlerOutboundBase class.

File Name: ProcessorHandlerOutboundResponse.cs	**Class Name: ProcessorHandlerOutboundResponse**
Method Name	**Description**
ProcessorHandlerOutboundRespons e	This is the default constructor of the ProcessorHandlerOutboundResponse class.
ProcessFile	This is an overridden method of ProcessFile method of base ProcessorHandlerOutboundBase class. This method is responsible for processing the 271,277 EDI response transactions and publishes the required events for saving functional acknowledgements, File Tracking Events, File Audit and Correlate response files to the main processing file.
GetBusinessCount	This method takes FileName as input parameter and is responsible for calculating the business count using Regex.Match () method call. Once Business count is calculated, method publishes the “COUNT_AFTER_VALIDATION” event using event id and count value.
GetBusinessCountsBeforeValidation	This method calculates the validation count by calling GetBusinessCountForXML () and publishes "COUNT_BEFORE_VALIDATION" event using EventID and validation count value.

26.3.4 File Tracking Library Project

This project is used to create classes which help EDI service to start and continue file tracking through the n-layer solution spanning over multiple servers. This library is accessed from File Transfer Service, pre-processor service, post-processor service and Event Publisher.





26.3.4.1 File Tracking Class

This class has all the functions which provide the ability to add, modify or continue File tracking.

File Name: FileTracking.cs	**Class Name: FileTracking**
Method Name	**Description**
FileTracking	This is the default constructor of the class.
EventTrackInsert	This method is used to save events (audit steps) of the file processing to the database. This method uses SP_EVENT_TRACK_INS stored procedure of the PKG_TRACKING Oracle package.
FileStatusUpdate	This method is used to update file status in the database. This method uses SP_FILE_TRACK_UPD_STATUS stored procedure of the PKG_TRACKING Oracle package.
FileTrackingInsert	This method is used to create a new File Tracking record. This method is called to start the file tracking of any file. This method uses SP_FILE_TRACK_INS stored procedure of the PKG_TRACKING Oracle package
ValidateFileExistsSelect	This method checks the existing T_FILE_TRACK table records to match the Crc32 value of the file. This is used during the duplicate check of the file.

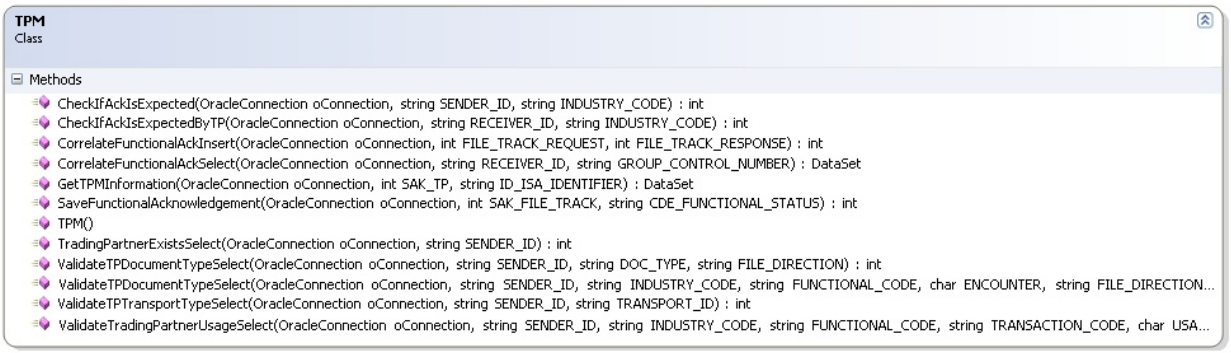
26.3.4.2 File Exception Class

This class provides the ability to correlate exception data persisted in database using Exception Manager Publish method with the File Tracking record.

File Name: FileException.cs	**Class Name: FileException**
Method Name	**Description**
FileException	This is the default constructor of the class.
FileTrackingExceptionsInsert	This method downloads a specific file from FTP servers to the local location.

26.3.5 TPM Library Project

This project is to perform Trading Partner Management (TPM) specific validation on the inbound and outbound EDI documents. This assembly is accessed by both Pre-Processor Service and Post-Processor Service.



26.3.5.1 TPM Class

This class is the main class which provides the methods to do TPM validations or do TPM specific data inquiry.

File Name: TPM.cs	**Class Name: TPM**
Method Name	**Description**
TPM	This is the default constructor of the class.
GetTPMInformation	This method is used to get the TPM information. This function checks the `T_TP table to see if TP information exists`.
TradingPartnerExistsSelect	This function checks the T_TP table to see if the trading partner exists with active status during the run time of the method. It returns SAK_TP as the return value.
ValidateTradingPartnerUsageSelect	This function selects data against T_TP, T_TP_CONFIG, T_CDE_SUPPORTED_DOC and T_CDE_FILE_DIRECTION tables to see if the document type exists with the correct usage indicator passed as the method parameter. It returns the SAK_TP_CONFIG as the

File Name: TPM.cs	Class Name: TPM
Method Name	**Description**
	return parameter.
CorrelateFunctionalAckSelect	This method is takes RECEIVER_ID, GROUP_CONTROL_NUMBER as input parameters and checks in T_DOCUMENT_TRACK table to see if records exist and it returns NUM_ORGN_GROUP_CNTRL, ID_ORGN_SND_QUALIFY, ID_ORGN_SND columns.
SaveFunctionalAcknowledgement	This method takes the File ID and Functional Acknowledgement value as the input parameters. Based on File ID it searches for SAK_DCOUMENT_TRACK. Then its gets the next unique value for SAK_FUNCTIONAL_ACK using the sequence key. With these values it inserts a record in T_FUNCTIONAL_ACK table and returns SAK_FUNCTIONAL_ACK value.
ValidateTPDocumentTypeSelect	This function selects data against T_TP, T_TP_CONFIG, T_CDE_SUPPORTED_DOC and T_CDE_FILE_DIRECTION tables to see if the document type is configured for the given trading partner. It returns the SAK_TP_CONFIG as the return parameter.
ValidateTPTransportTypeSelect	This function selects data against T_TP, T_TP_ALLOWED_TRANSPORT, and T_CDE_TRANSPORT tables to see if the trading partner is configured for the given transport type through which file was received. It returns the SAK_ALLOWED_TRANSPORT as the return parameter.
CorrelateFunctionalAckInsert	This method is used to correlate two File IDs usually in the case of request response files. This method uses SP_FILE_CORRELATION_INS stored procedure of the PKG_TRACKING Oracle package.
CheckIfAckIsExpected	This function selects data against T_TP, T_TP_CONFIG and T_CDE_SUPPORTED_DOC tables to see if the received document type is configured to receive 997 functional acknowledgements. It returns the SAK_TP_CONFIG as the return parameter.
CheckIfAckIsExpectedByTP	This method is not used by Pre-Processor Service.
ValidateNCPDPTradingPartner	This function selects data against T_TP, T_TP_CONFIG, T_CDE_SUPPORTED_DOC and T_CDE_FILE_DIRECTION tables to see if the NCPDP document type is configured for a trading partner. It returns the SAK_TP_CONFIG as the return parameter.

26.4 Configuration Description

26.4.1 App.Config

```
<configuration>
        <appSettings>
                <add key="RemotingChannel" value="9087" />
                <add key="InstanceName" value="EDS.POSTPROC.TCP" />
                <add key="Configuration" value="C:\Program
Files\EDS\PostProcessorService\ServiceConfigl.xml" />
                <add key="ConfigIsFile" value="true" />
                <add key="ConnString" value="Data Source= KYDDI_TR;User
Id=ky_tracker_mod;Password=ky_tracker_mod;Integrated Security=no;" />
                <add key="RecipientAddress" value="test-eds@eds.com" />
                <add key="SenderAddress" value="test-eds@eds.com" />
                <add key="EmailSubject" value="Post-processor SBG " />
                <add key="AckPath" value="\ACKs\Outbound\X12" />
                <add key="ServerName" value="SBG" />
                <add key="WMIIPAddress" value="MSSBG" />
                <add key="WMIPortNo" value="9099" />
                <add key="WMIRuntimeObjectName" value="EDS.MSMQ.TCP" />
                <add key="ComponentSeparator" value="0x3a" />
                <add key="SegmentTerminator" value="0x7e" />
        </appSettings>
</configuration>
```

File Name: App.config	
Attribute Name	**Description**
RemotingChannel	This attribute holds the value through which we can remote into Post processor. This is not yet implemented.
InstanceName	This is the channel name under which it will be registered. This is not yet implemented.
Configuration	This attribute value holds the location of the ServiceConfig.xml configuration file.
ConfigIsFile	This attribute takes true or false value and helps service to determine it should look for a file for configuration.
ConnString	This attribute value holds the database connection string.
WMIIPAddress	This attribute holds the value of the IpAddress or server name where MSMQ Listener Service is installed.
WMIPortNo	This attribute holds the value of the port number to remote in to MSMQ Listener Service.
WMIRuntimeObjectName	This is the channel name under which MSMQ Remoting stub is registered.
RecipientAddress	This is the global recipient email address where error

File Name: App.config	
Attribute Name	**Description**
	notification emails are sent.
SenderAddress	This is the global sender email address used to send error notification emails.
EmailSubject	This is the common text that can be attached to the notification email.
ServerName	This is the name of the server where FTS is deployed.
ComponentSeparator	This is default value of the component separator. It is configured to be 0x3a.
SegmentTerminator	This is default value of the component separator. It is configured to be 0x7e.

26.4.2 ServiceConfig.xml

This is the main configuration file which holds the configuration details about which directory to monitor, where to deliver, what file names to pick, how to rename at destination, where to archive and so on. There is supposed to be at least one configuration record for each file transfer. If the source and destination locations are both accessed through FTP or FTPS transport type then such transfers are accomplished using two configuration records, one to download the file from the resource server and the other to upload the file to the destination server.

Sample ServiceConfig.xml file:

```
<?xml version="1.0" encoding="utf-8" ?>

<configuration>

        <PostProcessors>

        <PostProcessor name="PostProcessor15" watcherName="837D_IN_Generic"
processorHandlerName="Inbound_837_D">

                <Writers>

                        <Writer name="837D_IN_Generic" />

                </Writers>

        </PostProcessor>

        <PostProcessor name="PostProcessor16" watcherName="837I_IN_Generic"
processorHandlerName="Inbound_837_I">

                <Writers>

                        <Writer name="837I_IN_Generic" />
```

```
                </Writers>
        </PostProcessor>
        <PostProcessor name="PostProcessor17" watcherName="837P_IN_Generic"
processorHandlerName="Inbound_837_P">
                <Writers>
                        <Writer name="837P_IN_Generic" />
                </Writers>
        </PostProcessor>
</PostProcessors>
        <ProcessorHandlers>
                <ProcessorHandler name="Inbound_837_D">
                        <Assembly name="C:\Program
Files\EDS\PostProcessorService\EDS.USGS.MED.EDI.PostProcessors.ProcessorHandler.dll"
/>
                        <Class
name="EDS.USGS.MED.EDI.PostProcessors.ProcessorHandlers.ProcessorHandlerInbound_8
37_D" />
                        <DocType value="837_D_X12_BATCH" />
                        <XDATA Indicator="Y" />
                        <XMLString value="(&lt;/claimList&gt;(.){0,6}&lt;[?]xml
version=&quot;1.0&quot; encoding=&quot;utf-8&quot;[?]&gt;(.){0,6}&lt;claimList&gt;)" />
                        <Directory FailedDir="\\10.40.13.12\FTS_UAT\FTS\FAILED"
ArchiveDir=""/>
                </ProcessorHandler>
                <ProcessorHandler name="Inbound_837_I">
                        <Assembly name="C:\Program
Files\EDS\PostProcessorService\EDS.USGS.MED.EDI.PostProcessors.ProcessorHandler.dll"
/>
                        <Class
name="EDS.USGS.MED.EDI.PostProcessors.ProcessorHandlers.ProcessorHandlerInbound_8
37_I" />
                        <DocType value="837_I_X12_BATCH" />
                        <XDATA Indicator="Y" />
                        <XMLString value="(&lt;/claimList&gt;(.){0,6}&lt;[?]xml
version=&quot;1.0&quot; encoding=&quot;utf-8&quot;[?]&gt;(.){0,6}&lt;claimList&gt;)" />
```

```
                <Directory FailedDir="\\10.40.13.12\FTS_UAT\FTS\FAILED"
ArchiveDir=""/>

        </ProcessorHandler>

        <ProcessorHandler name="Inbound_837_P">

                <Assembly name="C:\Program
Files\EDS\PostProcessorService\EDS.USGS.MED.EDI.PostProcessors.ProcessorHandler.dll"
/>

                <Class
name="EDS.USGS.MED.EDI.PostProcessors.ProcessorHandlers.ProcessorHandlerInbound_8
37_P" />

                <DocType value="837_P_X12_BATCH" />

                <XDATA Indicator="Y" />

                <XMLString value="(&lt;/claimList&gt;(.){0,6}&lt;[?]xml
version=&quot;1.0&quot; encoding=&quot;utf-8&quot;[?]&gt;(.){0,6}&lt;claimList&gt;)" />

                <Directory FailedDir="\\10.40.13.12\FTS_UAT\FTS\FAILED"
ArchiveDir=""/>

        </ProcessorHandler>

    </ProcessorHandlers>

    <Watchers>

        <Watcher name="277U_OUT_BBS" type="UNCWatcherType">

                <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.UNCWatcher" />

                <TriggeringInterval value="5000" />

                <TriggeringType value="PollingBased" />

                <WatcherLocation
value="\\10.40.13.12\FTS_UAT\FTS\BBS\RemittanceAdvice277U\Outbound277U\XData" />

        </Watcher>

        <Watcher name="277U_OUT_Passport" type="UNCWatcherType">

                <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.UNCWatcher" />

                <TriggeringInterval value="5000" />

                <TriggeringType value="PollingBased" />

                <WatcherLocation
value="\\10.40.13.12\FTS_UAT\FTS\Passport\RemittanceAdvice277U\Outbound277U\XData"
/>
```

```
            </Watcher>
            <Watcher name="997_OUT_BBS" type="UNCWatcherType_997">
                <ClassName
value="EDS.USGS.MED.EDI.Common.Watchers.UNCWatcher" />
                <TriggeringInterval value="5000" />
                <TriggeringType value="PollingBased" />
                <WatcherLocation
value="\\10.40.13.12\FTS_UAT\FTS\BBS\ACKs\Outbound\X12" />
            </Watcher>
        </Watchers>
        <Writers>
            <Writer name="837D_IN_Generic" type="UNCWriter">
                <ClassName value="EDS.USGS.MED.EDI.Common.Writers.UNCWriter"
/>
                <WriteLocation
value="\\10.40.13.12\FTS_UAT\FTS\Generic\Claim837D\Inbound837D\Postprocessor" />
                <WriteAsync value="true" />
            </Writer>
            <Writer name="837I_IN_Generic" type="UNCWriter">
                <ClassName value="EDS.USGS.MED.EDI.Common.Writers.UNCWriter"
/>
                <WriteLocation
value="\\10.40.13.12\FTS_UAT\FTS\Generic\Claim837I\Inbound837I\Postprocessor" />
                <WriteAsync value="true" />
            </Writer>
            <Writer name="837P_IN_Generic" type="UNCWriter">
                <ClassName value="EDS.USGS.MED.EDI.Common.Writers.UNCWriter"
/>
                <WriteLocation
value="\\10.40.13.12\FTS_UAT\FTS\Generic\Claim837P\Inbound837P\Postprocessor" />
                <WriteAsync value="true" />
            </Writer>
        </Writers>
```

</configuration>

26.4.2.1 PostProcessors Record

Parent Record Name: PostProcessors	
Attribute Name	**Description**
PostProcessor	This is the parent element name.
name	This attribute provides the unique name for this PostProcessor configuration.
WatcherName	This attribute gives the name of the watcher configuration to use for this PostProcessor.
WriterName	This attribute gives the name of the writer configuration to use for this PostProcessor.
processorHandlerName	This attribute gives the name of the ProcessorHandler configuration to use for this PostProcessor.

26.4.2.2 Processor Handler Record

Parent Record Name: ProcessorHandlers	
Attribute Name	**Description**
ProcessorHandler	This is the parent element name.
name	This attribute provides the unique name for this ProcessorHandler configuration.
Assembly	This lists the assembly name of the ProcessorHandler class.
Class	This lists the class name of the ProcessorHandler to use for this configuration.
DocType	This configuration provides the identification code of the document as set in the database. This value should match with the CDE_IDENTIFICATION column in the T_CDE_SUPPORTED_DOC table.
FailedDir	This folder lists the directory name where the file should be kept if there were an error in processing the file.
ArchiveDir	This attribute gives the folder location where the file is archived during Post Processor processing.

26.4.2.3 Watcher Record

Parent Record Name: Watchers	
Attribute Name	**Description**
Watcher	This is the parent element name.

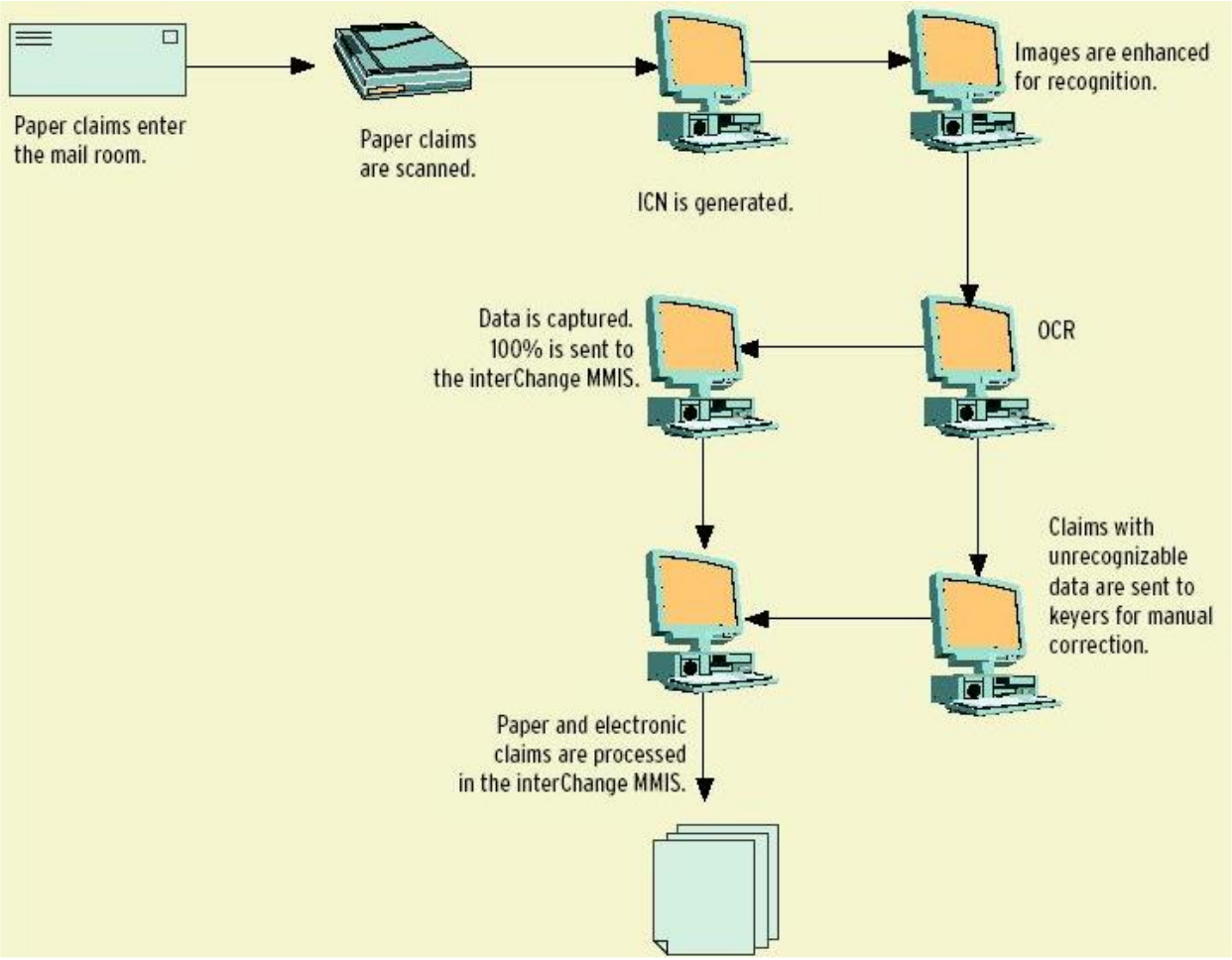
Parent Record Name: Watchers	
Attribute Name	**Description**
name	This attribute provides the unique name for this Watcher configuration.
type	This attribute defines the type of the watcher to use. Currently “UNCWatcherType” value is the only value that is supported.
ClassName	This attribute gives the name of the class to use for this watcher.
TriggeringInterval	This is the polling interval of the watcher. This value sets the time interval at which the folder being watched is checked for files.
TriggeringType	This is the type of the trigger to use for the watcher. Currently "PollingBased" is the only supported value.
WatcherLocation	This attribute gives the folder name which is supposed to be watched by this watcher.

26.4.2.4 Writer Record

Parent Record Name: Writers	
Attribute Name	**Description**
Writer	This is the parent element name.
name	This attribute provides the unique name for this Writer configuration.
type	This attribute defines the type of the writer to use. Currently “UNCWriter” value is the only value that is supported.
ClassName	This attribute gives the name of the class to use for this writer.
WriteLocation	This attribute gives the folder name where writer is supposed to create the new file.
WriteAsync	This element determines if the writer threads are supposed to be synchronous or asynchronous.

27 Data Capture Detailed System Design

27.1 Data Capture System Flow



27.2 Subsystem Description

27.2.1 Introduction to Data Capture

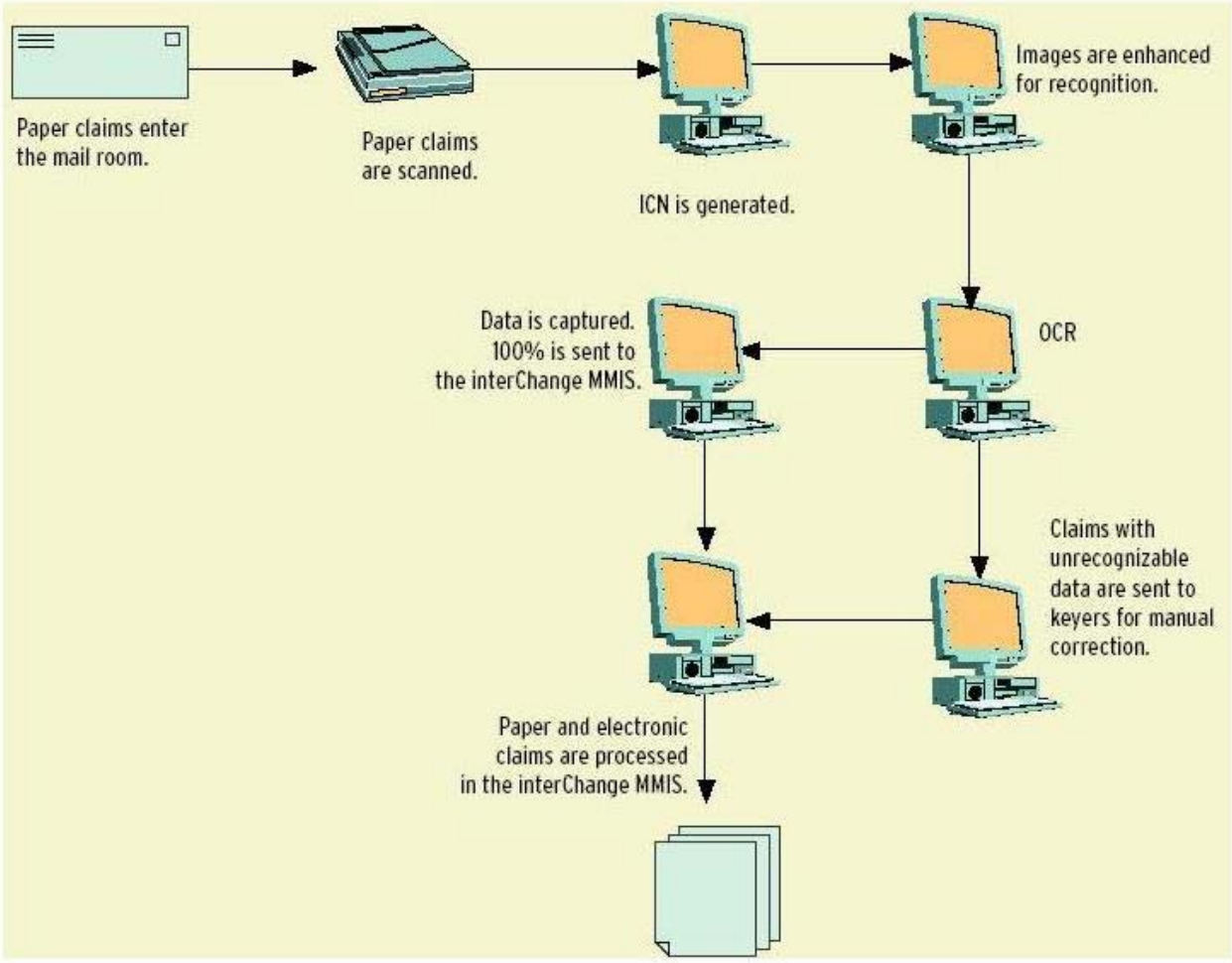
Data Capture uses Commercial Off-The-Shelf (COTS) by Captiva for the imaging of documents. For documents on a standard form, such as claims, this solution offers data capture through ICR/OCR. The Captiva Image Character Recognition/Optical Character Recognition (ICR/OCR) solution provides a high level of accuracy, repeated data validation to confirm the accuracy of data using reference files, and processing speed. The output of the ICR/OCR process is a formatted claim input file to the claim adjudication process, an image file, and the associated data indices for storage and later retrieval of the claim image through OnBase. For nonstandard form paper documents, such as provider correspondence, the software creates images which the user reviews, then manually enters and creates the necessary data indices for easy access and retrieval using OnBase.

27.3 Processes

The Data Capture subsystem handles only paper data entry. The subsystem supplies claims input files to interChange, and receives periodic extracts of interChange reference files used for cursory data entry editing. These files are noted under the Internal and External Interfaces section of this document.

27.4 Data Model

The Data Capture subsystem does not have its own data model, as it is an interface into the MMIS for paper claims. Paper claim forms are scanned, data is captured from the scanned images via OCR and/or data entry, claim data is exported from Captiva into the MMIS data format. The MMIS accepts the transactions, performs the necessary processing and updates the MMIS database as necessary. The following diagram depicts at a high level how the Data Capture subsystem interfaces interacts with the MMIS database.



27.4.1 External System Interfaces

The Data Capture subsystem has no external system interfaces.

27.5 Job Scripts

The Data Capture subsystem does not have any Job Scripts.

27.6 Programs

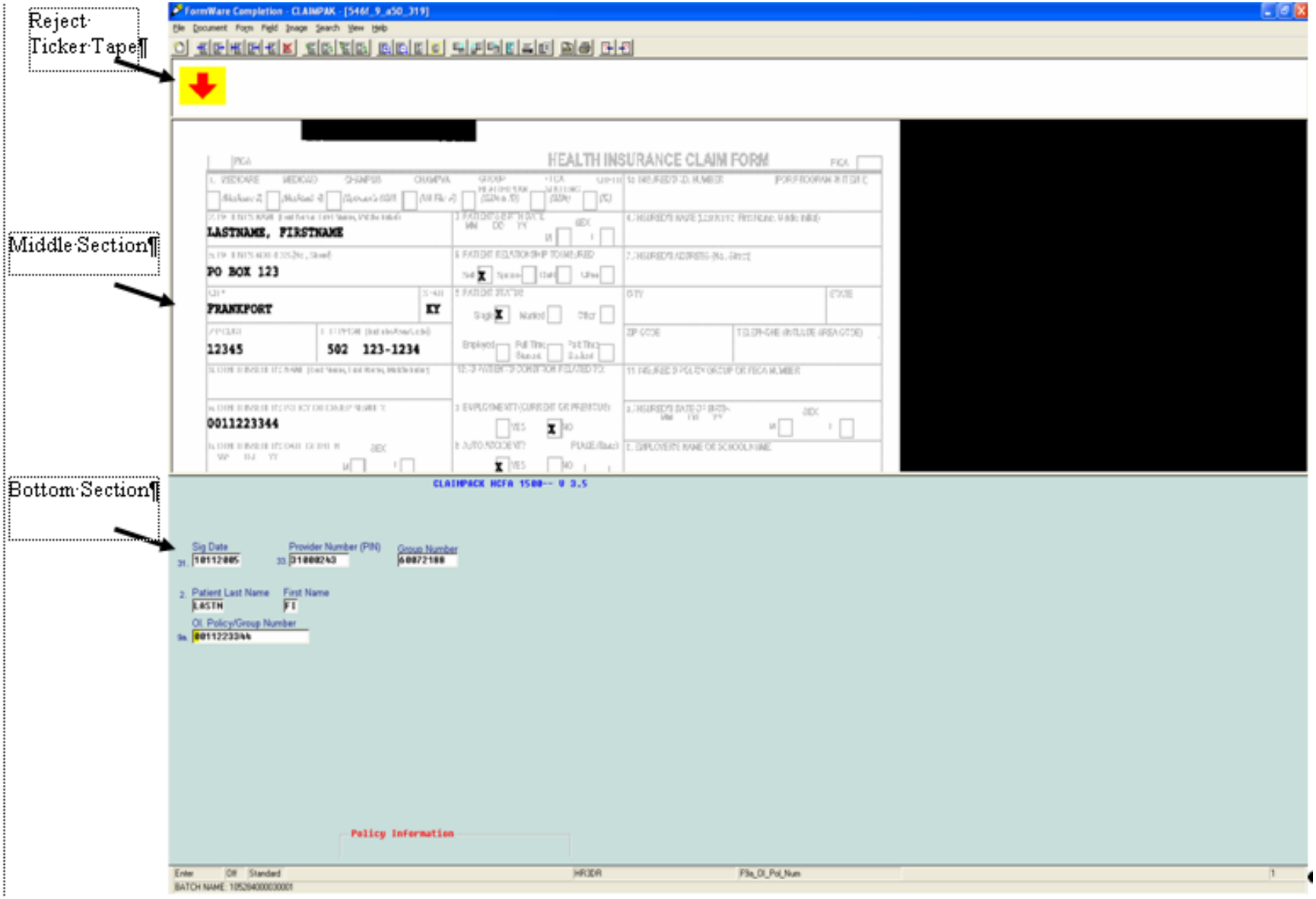
The Data Capture subsystem does not have any Programs.

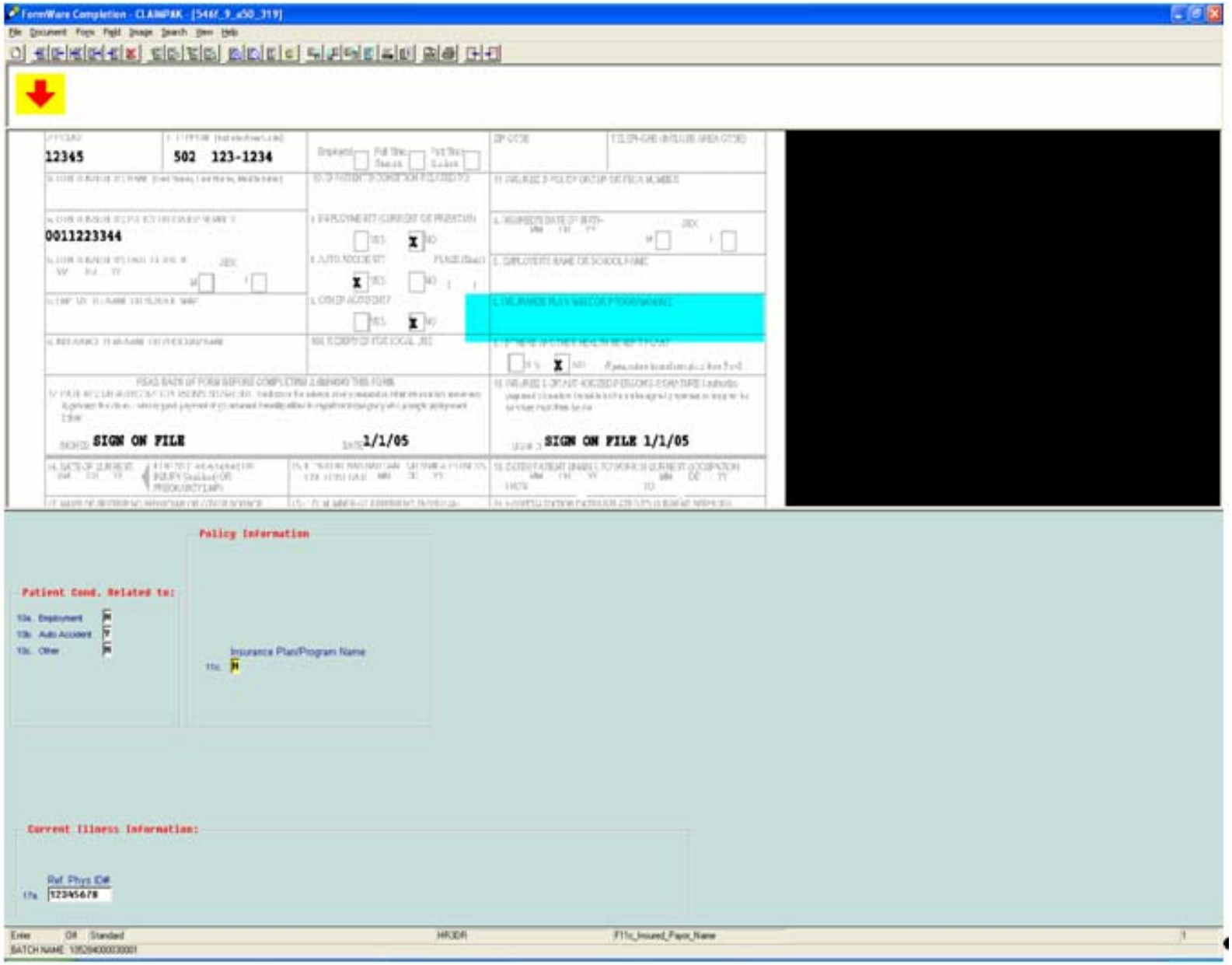
27.7 Internal Web Pages

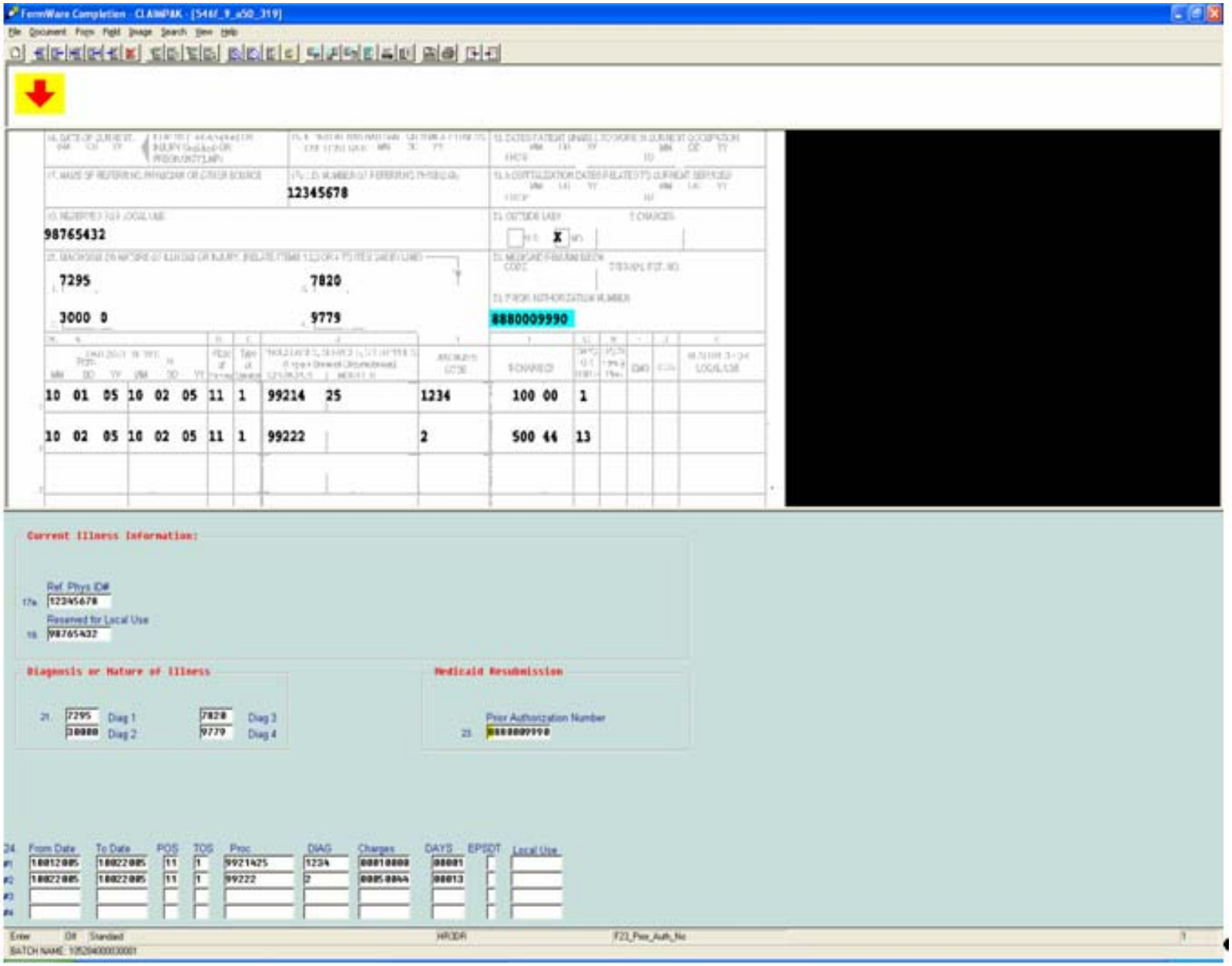
Data Capture Captiva pages do not follow other interChange pages/panels for formats and documentation. The following layouts are representative of Captiva form views for CMS 1500, UB92, UB04 and Dental Claims.

For readability, the images appear on the following pages.

27.7.1 KY MMIS HCFA 1500 Completion Screens




12345
502 123-1234
0011223344
SIGN ON FILE
1/1/05
SIGN ON FILE 1/1/05
Policy Information
Patient Cond. Related to:
Insurance Plan/Program Name
Current Illness Information:
Ref. Phys ID#
12345678


FormWare Completion - CLAIMPAK - [546f_9_a50_319]
File Document Form Field Image Search View Help
12345678
98765432
7295
7820
3000 0
9779
8880009990
10 01 05 10 02 05 11 1 99214 25 1234 100 00 1
10 02 05 10 02 05 11 1 99222 2 500 44 13
Current Illness Information:
Ref. Phys ID#
17a 12345678
Reserved for Local Use
19 98765432
Diagnosis or Nature of Illness
21 7295 Diag 1
30000 Diag 2
7820 Diag 3
9779 Diag 4
Medicaid Resubmission
Prior Authorization Number
23 8880009990
24 From Date To Date POS TOS Proc DIAG Charges DAYS EPSDT Local Use
#1 10012005 10022005 11 1 9921425 1234 00010000 00001
#2 10022005 10022005 11 1 99222 2 00050044 00013
#3
#4
Enter Off Standard HRDDR F23_Prior_Auth_No
BATCH NAME: 105204000030001

FormWare Completion - CLAIMPAK - [546f_9_a50_319]

File Document Pages Field Image Search View Help

PAT ACCT NUM

X

600 44 | 0 | 600 44

FRANKFORT MEDICAL CTR
PO BOX 0000
FRANKFORT, KY 12345

SIGN ON FILE 10/11/05

31000243 60072188

PLEASE PRINT OR TYPE

24	From Date	To Date	POS	TOS	Proc	DIAG	Charges	DAYS	EPSDT	Local Use
#1	10012005	10022005	11	1	9921425	1234	00010000	00001		
#2	10022005	10022005	11	1	99222	2	0005 0044	00013		
#3										
#4										
#5										
#6										
#7										
#8										
#9										
#10										
#11										
#12										

Patient's Account No. 26 PATACCTNUM

Total Charge 28 0006 0044

Amount Paid 29 00000000

Balance Due 30 0006 0044

Last Charge

27.7.2 HCFA 1500 Data Form Fields

SCREEN FIELD NAME:	**PROVIDER NUMBER (PIN)**
FORM FIELD NUMBER:	33
FIELD DESCRIPTION:	Kentucky Medicaid Provider Number
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	This is a Mandatory field for a claim to be processed. Provider must have 8 digits or 10 digits. If less than 8 or invalid after two tries the claim must be rejected for missing PIN.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The Provider Number (PIN) field is validated against a Provider Stub File.
- If Provider (PIN#) is missing or invalid, edit alert of: MISSING/INVALID PROVIDER NUMBER. PLEASE ENTER VALID PROVIDER NUMBER OR REJECT THE CLAIM is received at the bottom of the screen. Verify information on image; if data is missing or invalid, the claim must be rejected from the file to be Returned to Provider (RTP'd).
- If Alpha value is entered, you will receive a Dialog Box stating "Invalid Character". Click OK. If Alpha character is present in field, the claim must be rejected from the file to be Returned to Provider (RTP'd).

SCREEN FIELD NAME:	**GROUP NUMBER**
FORM FIELD NUMBER:	33
FIELD DESCRIPTION:	Kentucky Medicaid Provider Number
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Must be 8 or 10 digits if present. Looks like a provider number. Do not enter provider PIN number in this field. If blank continue to next field. DO NOT enter zip codes or phone numbers in this field if that type of value is present on the claim in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The Provider Number (GRP) field is validated against a Provider Stub File.
- If Provider (GRP#) is invalid, edit alert of: MISSING/INVALID GROUP NUMBER. PLEASE ENTER VALID GROUP NUMBER is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**PATIENT LAST NAME**
FORM FIELD NUMBER:	2
FIELD DESCRIPTION:	Recipient Last Name
FORMAT:	Up to 5 Byte Alpha value
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter first 5 letters of Patient Last Name in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID is validated against a Recipient Stub File.
- If Patient Last Name is missing, edit alert of: PATIENT LAST NAME SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Numeric value is entered, you will receive a Dialog Box stating “Alpha Field Type”. Click OK. Numeric value is not allowed in this field; enter all values present except numeric values. If entire field contains numeric values, leave field blank.

SCREEN FIELD NAME:	**FIRST NAME**
FORM FIELD NUMBER:	2
FIELD DESCRIPTION:	Recipient First Name
FORMAT:	Up to 2 Byte Alpha value
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter first 2 letters of Patient First Name in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID is validated against a Recipient Stub File.
- If Patient First Name is missing, edit alert of: PATIENT FIRST NAME SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Numeric value is entered, you will receive a Dialog Box stating “Alpha Field Type”. Click OK. Numeric value is not allowed in this field; enter all values present except numeric values. If entire field contains numeric values, leave field blank.

SCREEN FIELD NAME:	**OI. POLICY/GROUP NUMBER**
FORM FIELD NUMBER:	9a
FIELD DESCRIPTION:	Recipient's Medical Assistance Identification (MAID) Number
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	If the number present contains a letter, key only the numbers. If no number is present or is unreadable leave blank and continue to next field. MAID number coincides with the patient name, if number is red flagged check name and number for errors. If no errors are present shift enter and continue.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID are validated against a Recipient Stub File.
- If MAID is missing/invalid, edit alert of: MAID MISSING/INVALID. MAID SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image, if data is missing or invalid shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If MAID ID entered does not match the stub file lookup being performed against the Recipient's Name/ID, edit alert of: LOOPKUP FAILED. MAID MISSING/INVALID is received at the bottom of the screen. The cursor will be positioned in the Patient Last Name Field. Validation of the Patient Last Name, First Name and MAID will need to occur. You can either re-key the information or if values present on data form are correct press shift/enter to override the error message.
- If Alpha value is entered, you will receive a Dialog Box stating "Invalid Character". Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	**PATIENT'S DOB**
FORM FIELD NUMBER:	3
FIELD DESCRIPTION:	Patient Date of Birth
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter DOB in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If DOB is missing, edit alert of: "PATIENT DOB SHOULD NOT BE BLANK" is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating "Numeric Field Type". Click OK. Alpha value is not allowed in this field; enter all values present except alpha values. If entire field contains alpha values, leave field blank.
- If a date in the format of MMDDYY is received, an alert will be presented to validate the century.
- If date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is after scan date, edit alert of: DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on data image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**PATIENT COND. RELATED TO** **EMPLOYMENT** **AUTO ACCIDENT** **OTHER**
FORM FIELD NUMBER:	10a, 10b, 10c
FIELD DESCRIPTION:	Worker's Comp, Auto Accident, Other Accident Indicators
FORMAT:	Each are 1 Byte Alpha Fields
VALUES:	Y or N
BUSINESS RULES:	This is a Yes/No field. If marked Yes, place a Y, if No press enter and an N will be populated. If X is between Yes and No, mark the box No.

- If, there is no check mark in the Yes or No box, enter value of "N" or press enter key and default value of "N" will be populated.
- If, there is a check mark in the No box, enter value of "N" or press enter key and default value of "N" will be populated.
- If, there is a check mark in the Yes box, enter value of "Y".
- If, there is a check mark in both boxes for a single field (i.e. Other has both Yes/No checked), enter value of "Y".
- If invalid value is received Dialog Box stating "Invalid Character", along with: Y-YES, N-NO EDIT ALERT: PLEASE REVIEW DATA is received at the bottom of the screen. Verify information on image, if unexpected value present, press enter key to allow default value of "N".

SCREEN FIELD NAME:	**FOR LOCAL USE**
FORM FIELD NUMBER:	10d
FIELD DESCRIPTION:	Physician Assistant
FORMAT:	10 bytes numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	This field is for entering Provider Number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The value within the For Local Use Field when used as Referring Provider Number is validated against a Provider Stub File. Referring Provider Numbers are a length of 8 or 10 bytes.
- If enter any number different than 8 or 10 digits, edit alert of “PROVIDER NUMBER INVALID: LENGTH” is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Referring Provider is invalid, edit alert of: “PROVIDER NUMBER NOT ON FILE” is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**Date of Current Illness**
FORM FIELD NUMBER:	14
FIELD DESCRIPTION:	Accident Date
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter Accident Date in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If “Y” is not present in any of Fields 10A-C, this field will be protected and cleared. If “Y” is present, the accident date should be entered in this field.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha value is not allowed in this field; enter all values present except alpha values. If entire field contains alpha values, leave field blank.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is after scan date, edit alert of: DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on data image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**REF. PHYS ID#**
FORM FIELD NUMBER:	17a
FIELD DESCRIPTION:	Unique Physician Identification Number (UPIN) of the Referring Physician.
FORMAT:	10 Byte Alphanumeric Field
VALUES:	No specific value editing is done.
BUSINESS RULES:	Contains Phys ID numbers. Ex: I20776, G70795, 64155344, 30609099, etc. If the field contains both types of values use the ID starting with the letter.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).

SCREEN FIELD NAME:	**RESERVED FOR LOCAL USE**
FORM FIELD NUMBER:	19
FIELD DESCRIPTION:	ID Number of Referring Physician (KENPAC)
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Used as: ID Number of Referring Physician If present, must contain an 8 digit Provider Number. Ex: 64155344, 30609099, etc.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The value within the Reserved for Local Use Field when used as Referring Provider Number is validated against a Provider Stub File. Referring Provider Numbers are a length of 8 or 10 bytes.
- If Referring Provider is invalid, edit alert of: REFERRING PROVIDER NUMBER is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**DIAGNOSIS OR NATURE OF ILLNESS**
FORM FIELD NUMBER:	21
FIELD DESCRIPTION:	ICD-9 (International Classification of Diseases) DIAGNOSIS CODE
FORMAT:	6 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Key what you see in designated fields. Do not key decimals, however you need to pick up the value after the decimal if present.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Diagnosis Codes are validated against a Stub File.
- If the first Diagnosis Code is missing, edit alert of: DIAGNOSIS CODE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- Up to 4 diagnosis codes can be present. I.E. 21-Diag 1, 21-Diag 2 can be present with nothing present in 21-Diag 3 and 21-Diag 4.
- If any of the Diagnosis Codes are invalid, edit alert of: DIAGNOSIS CODE NOT IN TABLE is received at the bottom of the screen. Verify information on image, if data was captured correctly shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If the diagnosis codes are not received in sequential order, I.E. 21-Diag 1 and 21-Diag 3 are present however 21-Diag 2 is missing, edit alert of: PREVIOUS DIAGNOSIS CODES CANNOT BE BLANK is received at the bottom of the screen. Verify information from image, if data is present and not in sequential order, shift/enter to override error message.

SCREEN FIELD NAME:	**PRIOR AUTHORIZATION NUMBER**
FORM FIELD NUMBER:	23
FIELD DESCRIPTION:	PRIOR AUTHORIZATION NUMBER or TIME OF PICKUP
FORMAT:	10 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Used as: Prior Authorization Number If the series of numbers contains a letter (ex: 18D, etc) do not key, leave blank. If no Prior Authorization number is present, continue to next field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Prior Authorization Numbers can be up to 10 bytes.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	**PRIOR AUTHORIZATION NUMBER**
FORM FIELD NUMBER:	23
FIELD DESCRIPTION:	PRIOR AUTHORIZATION NUMBER or TIME OF PICKUP
FORMAT:	10 Byte Numeric
VALUES:	Positions 1-2 equal 00-24 Positions 3-4 equal 00-59
BUSINESS RULES:	Used as: Time of Pickup If the value present is a 4-digit number for provider types 55 or 56 the value represent is time of pickup. Enter only the 4-digits do not enter the seconds if present. Ex: 18:41:22

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Time of Pickup is used for Provider Type 55 and 56.
- If Time of Pickup is invalid, edit alert of: MISSING/INVALID TIME FORMAT. PLEASE ENTER VALID FORMAT is received at the bottom of the screen. Verify information from image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**FROM DATE**
FORM FIELD NUMBER:	24A
FIELD DESCRIPTION:	FROM DATE OF SERVICE IS THE BEGINNING DATE OF SERVICE FOR THE PROCEDURE
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter date on claim form. FDOS may not equal the TDOS. If no date is present in FDOS, however there is data in other fields on that detail line, enter 00000000 in this date field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If the From Date is missing from the First Detail Line, edit alert of: FIRST FROM DATE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing claim must be rejected from file and RTP'd. If data is present and was captured incorrectly enter value from image.
- If From Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If From Date is after scan date, edit alert of: SERVICE FROM DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.

- If From Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on data image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If From Date is older than a year from current date, edit alert of: DATE IS OLDER THAN 1 YEAR is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**TO DATE**
FORM FIELD NUMBER:	24A
FIELD DESCRIPTION:	TO DATE OF SERVICE IS THE ENDING DATE OF SERVICE FOR THE PROCEDURE
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter date on claim form. TDOS may not equal the FDOS. If the TDOS equals the FDOS you can enter through the field and the FDOS will be populated in the TDOS field. If the TDOS does not equal the FDOS, enter the date on the claim form.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If the To Date is missing, press Enter key. The value in the From Date will be populated in the To Date field.
- If To Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If To Date is before From Date, edit alert of: TO DATE OLDER THAN FROM DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If To Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.

- If To Date is older than a year from current date, edit alert of: DATE IS OLDER THAN 1 YEAR is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**POS**
FORM FIELD NUMBER:	24b
FIELD DESCRIPTION:	Place the service was performed.
FORMAT:	2 Byte Alphanumeric
VALUES:	01, 03-08, 11-15, 20-26, 31-34, 41-42, 49-57, 60-62, 65, 71-72, 81 and 99
BUSINESS RULES:	Enter what is on the claim form (Ex: 11, 21, 22 etc). If no data is present press the enter key to continue to the next field.

- If POS is invalid, edit alert of: PLACE OF SERVICE NOT IN TABLE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**TOS**
FORM FIELD NUMBER:	24c
FIELD DESCRIPTION:	Type of service that was used to perform the treatment.
FORMAT:	2 Byte Alphanumeric
VALUES:	1-6, 01-06.
BUSINESS RULES:	Enter what is on the claim form (Ex: 01, 04, 09, etc). If no data is present press the enter key to continue to the next field.

- If TOS is invalid, edit alert of: TYPE OF SERVICE CODE NOT IN TABLE is received at the bottom of the screen. Verify information on image/data form; if correct shift/enter to accept data. If data is not correct, enter value from image.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).

SCREEN FIELD NAME:	**PROC**
FORM FIELD NUMBER:	24d
FIELD DESCRIPTION:	Procedure Code identifies what service was performed.
FORMAT:	13 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Procedure codes can start with a letter followed by numbers or all numbers. Modifier codes are 2 letters or numbers that follow the procedure code, they are not always present. (Ex: QX, HN, 25, 26 etc). These codes must be entered if present on the claim.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If the Procedure Code is missing, edit alert of: PROCEDURE CODE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message. If data is present and captured incorrectly enter value from image.
- If the Procedure Code is invalid, edit alert of: LOOKUP FAILED. PROCEDURE CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data is present and invalid shift/enter to override error message.
- If the value equals a length of 5, it will be validated against a Procedure Code Stub File.
- If the value equals a length of 7, the first 5 positions will be validated against a Procedure Code Stub File, the next 2 positions will be validated against a Modifier Stub File.
- If the value equals a length of 9, the first 5 positions will be validated against a Procedure Code Stub File, positions 6 and 7 and positions 8 and 9 will both be validated against a Modifier Stub File.

- If the value equals a length of 11, the first 5 positions will be validated against a Procedure Code Stub File, positions 6 and 7, positions 8 and 9 and positions 10 and 11 will all be validated against a Modifier Stub File.
- If the value equals a length of 13, the first 5 positions will be validated against a Procedure Code Stub File, positions 6 and 7, positions 8 and 9, positions 10 and 11, and positions 12 and 13 will all be validated against a Modifier Stub File.

SCREEN FIELD NAME:	**DIAGNOSIS CODE**
FORM FIELD NUMBER:	24e
FIELD DESCRIPTION:	Diagnosis Code Pointer value references the diagnosis for which the recipient is being treated as it relates to field 21.1, 21.2, 21.3 and 21.4.
FORMAT:	1 Byte Alphanumeric
VALUES:	1, 2, 3, 4
BUSINESS RULES:	These numbers refer to the Diagnosis Codes in box 21.1 to 21.4 on the claim. Enter value present on claim form. There can be more than one value present. (Ex: 1, 2, 3, 4 or 4, 2, 1, 3) – key what you see.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If the Diagnosis Code Pointer is missing, edit alert of: INVALID DIAG CODE POINTER. PLEASE VERIFY is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If the Diagnosis Code Pointer is invalid, edit alert of: DIAGNOSIS POINTER IS NOT VALID is received at the bottom of the screen. Verify information on image/data form; if data is present and invalid shift/enter to override error message.

SCREEN FIELD NAME:	**CHARGES**
FORM FIELD NUMBER:	24f
FIELD DESCRIPTION:	Detail amount billed for a specific procedure.
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	This is the charge for the procedure code. Key the entire dollar amount (including cents) without the decimal point. It is very important that these amounts are keyed correctly. If keyed wrong, it will cause delay in processing of the claim. Do not key negative charge lines.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If the detail Charge is missing, edit alert of: INVALID CHARGES. PLEASE VERIFY is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message.
- If the detail Charge exceeds length allowed you will receive a Completion box that states “Field Boundary”. Click OK, and then reject the claim from the file. If the length received exceeds the length of 8, the claim must be RTP’d for exceeding length allowed.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field, leave field blank. You will receive, edit alert of: INVALID CHARGES. PLEASE VERIFY at the bottom of the screen. Press enter key, field will be left blank.

SCREEN FIELD NAME:	**DAYS**
FORM FIELD NUMBER:	24g
FIELD DESCRIPTION:	Number of Days or Units being billed for a specific procedure.
FORMAT:	5 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter number of days or units.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, press enter key and field will be defaulted with a value of 1.

SCREEN FIELD NAME:	**EPSDT**
FORM FIELD NUMBER:	24h
FIELD DESCRIPTION:	Early and Periodic Screening Treatment (EPSDT) and Family Planning Indicator.
FORMAT:	1 Byte Alphanumeric
VALUES:	E, F or B
BUSINESS RULES:	Enter PDSDT/Family Planning Indicator.

- If the EPSDT indicator is invalid, edit alert of: INVALID EPSDT/FAMILY PLANNING INDICATOR is received at the bottom of the screen. Verify information on image/data form, if data is invalid, press enter out of field. Field will be left blank.

SCREEN FIELD NAME:	**LOCAL USE**
FORM FIELD NUMBER:	24k
FIELD DESCRIPTION:	Various uses. Please see comments under Business Rules.
FORMAT:	8 Byte Alphanumeric
VALUES:	Immunization sequence: 01-99, rendering provider is validated against a stub file.
BUSINESS RULES:	Enter KY specific data.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If Provider Type is 21 and data is present in this field, length should equal 3. If 3 bytes is not received, edit alert of “Please review data” is received at the bottom of the screen. Verify information on the image/data form; if data is present and valid shift/enter to override error message.
- If Provider Type is 30 and data is present in this field, length should equal 4. If 4 bytes is not received, edit alert of “Please review data” is received at the bottom of the screen. Verify information on the image/data form; if data is present and valid shift/enter to override error message.
- If a value of “03” is received in field 24B, and data is present in this field, length should equal 7. If 7 bytes is not received, edit alert of “Please review data” is received at the bottom of the screen. Verify information on the image/data form; if data is present and valid shift/enter to override error message.
- If data in this field is present and numeric with a length of 2, 4 or 6 bytes, the value should be 01-99 in each two byte value. If the value is not 01-99 in each, an edit alert of : Invalid Disposition/ Immunization sequence code is received at the bottom of the screen. Verify information on the image/data form; if data is present and valid shift/enter to override error message.

- If data in this field is present and alpha with a length of 2, 4, 6 or 8 bytes, the value should be valid in each two byte value. If the value is not valid in each, an edit alert of : Invalid Disposition/ Immunization sequence code is received at the bottom of the screen. Verify information on the image/data form; if data is present and valid shift/enter to override error message.
- If data is present in this field, and does not match any of the above criteria, the value is assumed to be a rendering provider number and validated against the stub file. If the value is not found on the stub file, an edit alert of: RENDERING IS MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data is present and valid shift/enter to override error message.

SCREEN FIELD NAME:	**PATIENT'S ACCOUNT NO**
FORM FIELD NUMBER:	26
FIELD DESCRIPTION:	Account number assigned by the Provider for the recipient.
FORMAT:	20 Byte Alphanumeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter first 20 positions of Patient's Account Number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.

SCREEN FIELD NAME:	**TOTAL CHARGES**
FORM FIELD NUMBER:	28
FIELD DESCRIPTION:	Sum of all detail charges.
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter total charges.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If the Total Charge is invalid (does not equal the sum of all details), edit alert of: TOTAL CLAIM CHARGE <$ AMOUNT > NOT EQUAL TO TOTAL LINE ITEM AMOUNTS <SUM OF DETAIL(s) AMOUNT> is received at the bottom of the screen. Verify information on image/data form; if data is correct press shift/enter. You will receive a Dialog Box stating “Total claim charge <xxx>not equal to total line item amounts <xxx>”. Click OK. You will then be taken back to the first detail to validate value. If the value is correct, press shift/enter. You will be taken through each detail line present on the claim to validate values. Correct those that are wrong. Once the claim is in balance you will be taken back to the Total Charge field. If the claim is truly out of balance, once you are taken back to the Total Charge field, press shift/enter to override error message. If for some reason when you re-enter values in the charges field (and they were different from what was originally entered) and the claim still does not balance you will receive another edit alert of: CLAIM STILL OUT OF BALANCE. PLEASE RE-KEY AMOUNT. Verify information on image/data form; if data is valid press shift/enter to override error message, if invalid correct data.
- If the Total Charge is missing, edit alert of: TOTAL CHARGE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image/data form; if data is missing press shift/enter. You will receive the TOTAL CLAIM CHARGE error mentioned above. Follow steps mentioned above to validate values.

- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field, leave field blank. You will receive, edit alert of: TOTAL CHARGE SHOULD NOT BE BLANK at the bottom of the screen. Press enter key, field will be blank and out of balance error will be received. Follow steps mentioned above to validate data.

SCREEN FIELD NAME:	**AMOUNT PAID**
FORM FIELD NUMBER:	29
FIELD DESCRIPTION:	Amount that has been paid by other health insurance
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter amount paid by other insurance.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.

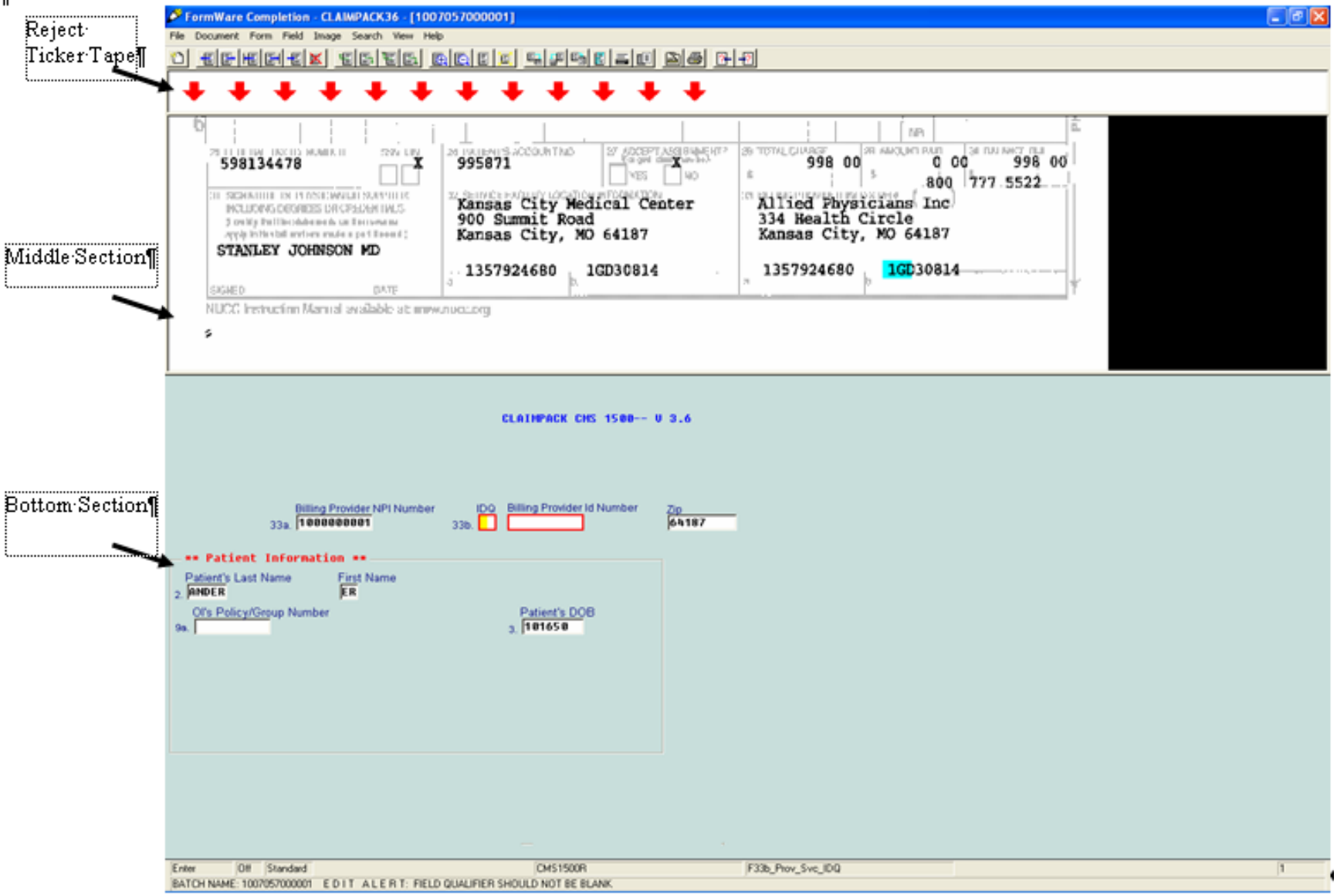
SCREEN FIELD NAME:	**BALANCE DUE**
FORM FIELD NUMBER:	30
FIELD DESCRIPTION:	Sum of all detail charges.
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter balance due.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If Balance Due does not equal Total Charge minus Amount Paid edit alert of: BALANCE DUE < $> TOTAL – AMOUNT PAID is received at the bottom of the screen. Verify information on image/data form; if data is correct press shift/enter. You will be taken back to the Amount Paid field. If information on data form is still correct press shift/enter to override error message. If data was invalid correct data.

SCREEN FIELD NAME:	**LAST CHANCE**
FORM FIELD NUMBER:	N/A
FIELD DESCRIPTION:	Last opportunity to verify claim information
FORMAT:	1 Byte
VALUES:	No specific value editing is done.
BUSINESS RULES:	Hit enter when ready to move to the next claim.

- This is the last chance to correct ANYTHING on this claim. Once any key is pressed the document can not be retrieved to make any modifications.
- Prior to pressing any keying, scroll up on the claim to see if there are ANY handwritten fields that were not captured from OCR. The most common fields that are handwritten and not captured are: 17a, 19, 23, 24c and 24k.

27.7.3 KY MMIS CMS 1500 Completion Screens



FormWare Completion - CLAIMPACK36 - [1007057000001]

File Document Form Field Image Search View Help

13450325

ANDERSON ERICK T

10 16 50

ANDERSON ERICK T

444 CENTER DR

444 CENTER DR

KANSAS CITY MO

KANSAS CITY MO

Ot's Policy/Group Number

9a.

Patient's DOB

3. 09151923

Patient Cond. Related to:

10a. Employment N

10b. Auto Accident N

10c. Other Accident N

For Local Use

10d.

Enter Ott Standard CMS1500R F3_Pat_DOB 1

BATCH NAME: 1007057000001 EDIT ALERT: QUALIFIER ID NOT IN TABLE

FormWare Completion - CLAIMPACK36 - [1007057000001]

File Document Form Field Image Search View Help

162 2 518 5

401 93 V15 84

Begin 1245 End 1415 Time 90 minutes

12 13 05 12 13 05 24 31600 12 755 00 1 N 1G D30814 1357924680 1G D30814

Reserved for Local Use

Diagnosis or Nature of Illness or Injury

21. 1622 Diagnosis 1 Diagnosis 3

Diagnosis 2 Diagnosis 4

Medicaid Resubmission

Prior Authorization Number

23.

24. From Date	To Date	POS	EMG	CPT/HCPCS	Local Use	Diagnosis Pointer	Charges	Days or Units	EPSDT	IDQ	Rendering Provider Id #
Line #1 Supplemental Information											
#1 12132005	12132005	24		31600		12	0000075500	1	N	1G NPI	D30814 1357924680
Line #2 Supplemental Information											
#2 12132005	12132005	24		31615		12	0000024300	1	N	1G NPI	D30814 1357924680
Line #3 Supplemental Information											
#3										NPI	
Line #4 Supplemental Information											
										NPI	

Enter Ctrl Standard CMS1500R F21_Diag_Left_2 1

BATCH NAME: 1007057000001 EDIT ALERT: DATE IS OLDER THAN 1 YEARS

Begin 1245 End 1415 Time 90 minutes

1	12	13	05	12	13	05	24	31600	12	755	00	1	N	1G	D30814	1357924680
2	12	13	05	12	13	05	24	31615	12	243	00	1	N	1G	D30814	1357924680
3																
4																

Line #2 Supplemental Information
#2 24 31615 1 0000010000 001 NPI

Line #3 Supplemental Information
#3 NPI

Line #4 Supplemental Information
#4 NPI

Line #5 Supplemental Information
#5 NPI

Line #6 Supplemental Information
#6 NPI

Patient's Account No.
26. 995871

Total Charge
28. 0000099800

Amount Paid
29.

Balance Due
30. 0000099800

Last Charge

Enter Off Standard CMS1500R F24a_Svc_From_Date_2 1

27.7.4 CMS1500 Data Form Fields

SCREEN FIELD NAME:	BILLING PROVIDER NPI NUMBER
FORM FIELD NUMBER:	33a
FIELD DESCRIPTION:	Billing Provider National Provider Identification Number
FORMAT:	10 digits - Numeric Only
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Must be 10 digits if present. Do not enter provider number in this field. If blank continue to next field. This field is mandatory, if provider number is missing, for a claim to be processed. Otherwise RTP the claim.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The 10 positions are validated against a Provider Stub File if present.
- If NPI is invalid, edit alert of “PLEASE ENTER VALID NPI” is received at the bottom of the screen. The field cannot be overridden when 33b does not contain a valid provider number.
- If enter less than 10 digits, edit alert of “PLEASE ENTER 10 DIGITS” is received at the bottom of the screen.
- If this field is blank, valid provider number must be present otherwise the claim must be rejected from the file to be Returned to Provider (RTP’d)
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**IDQ**
FORM FIELD NUMBER:	33b
FIELD DESCRIPTION:	Qualifier ID Code
FORMAT:	2 Alpha-numeric characters
VALUES:	1D and ZZ
BUSINESS RULES:	If NPI is present, the value for this field must be ZZ otherwise is 1D. It is required when provider number is present.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The field is validated against an OtherID table.
- If IDQ is missing, edit alert of “FIELD QUALIFIER SHOULD NOT BE BLANK” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If IDQ is invalid, edit alert of “QUALIFIER ID NOT IN TABLE” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If IDQ is missing when NPI is present, edit alert of “NPI IS PRESENT. PLEASE ENTER CORRECT QUALIFIER CODE.” is received at the bottom of the screen.
- If IDQ is invalid when NPI is present, edit alert of “INVALID IDQ. PLEASE ENTER VALID IDQ.” is received at the bottom of the screen.

SCREEN FIELD NAME:	BILLING PROVIDER ID NUMBER
FORM FIELD NUMBER:	33b
FIELD DESCRIPTION:	Kentucky Medicaid Provider Number
FORMAT:	8 or 10 digits - Numeric only
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	This is a Mandatory field, if NPI is missing, for a claim to be processed. If both provider number and NPI are missing, the claim must be RTP'd.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If IDQ is "1D"
  - The Provider Number (PIN) field is validated against a Provider Stub File.
  - If enter any number different than 8 digits or 10 digits, edit alert of "PLEASE ENTER CORRECT LENGTH CODE" is received at the bottom of the screen.
  - If Provider (PIN#) is missing or invalid, edit alert of "MISSING/INVALID PROVIDER NUMBER. PLEASE ENTER VALID PROVIDER NUMBER OR REJECT THE CLAIM" is received at the bottom of the screen. Verify information on image; if data is missing or invalid, the claim must be rejected from the file to be Returned to Provider (RTP'd) if valid NPI is missing.
- If IDQ is "ZZ"
  - The field is validated against a Taxonomy Stub File.
  - If taxonomy is missing or invalid, edit alert of "PLEASE ENTER VALID TAXONOMY CODE" is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**Zip**
FORM FIELD NUMBER:	33a
FIELD DESCRIPTION:	Kentucky Medicaid Provider Zip Code
FORMAT:	5 or 9 digits - Numeric only
VALUES:	No specific value editing is done.
BUSINESS RULES:	This field is entered when NPI is received on the claim.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If enter any number different than 5 digits or 9 digits, edit alert of “PLEASE ENTER VALID ZIP CODE” is received at the bottom of the screen.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; do not enter alpha characters.

SCREEN FIELD NAME:	PATIENT LAST NAME
FORM FIELD NUMBER:	2
FIELD DESCRIPTION:	Recipient Last Name
FORMAT:	Up to 5 Byte Alpha value
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter first 5 letters of Patient Last Name in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID is validated against a Recipient Stub File.
- If Patient Last Name is missing, edit alert of: PATIENT LAST NAME SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Numeric value is entered, you will receive a Dialog Box stating “Alpha Field Type”. Click OK. Numeric value is not allowed in this field; enter all values present except numeric values. If entire field contains numeric values, leave field blank.

SCREEN FIELD NAME:	FIRST NAME
FORM FIELD NUMBER:	2
FIELD DESCRIPTION:	Recipient First Name
FORMAT:	Up to 2 Byte Alpha value
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter first 2 letters of Patient First Name in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID is validated against a Recipient Stub File.
- If Patient First Name is missing, edit alert of: PATIENT FIRST NAME SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Numeric value is entered, you will receive a Dialog Box stating “Alpha Field Type”. Click OK. Numeric value is not allowed in this field; enter all values present except numeric values. If entire field contains numeric values, leave field blank.

SCREEN FIELD NAME:	OI. POLICY/GROUP NUMBER
FORM FIELD NUMBER:	9a
FIELD DESCRIPTION:	Recipient's Medical Assistance Identification (MAID) Number
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	If the number present contains a letter, key only the numbers. If no number is present or is unreadable leave blank and continue to next field. MAID number coincides with the patient name, if number is red flagged check name and number for errors. If no errors are present shift enter and continue.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID are validated against a Recipient Stub File.
- If MAID is missing/invalid, edit alert of: MAID MISSING/INVALID. MAID SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image, if data is missing or invalid shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If MAID ID entered does not match the stub file lookup being performed against the Recipient's Name/ID, edit alert of: LOOPKUP FAILED. MAID MISSING/INVALID is received at the bottom of the screen. The cursor will be positioned in the Patient Last Name Field. Validation of the Patient Last Name, First Name and MAID will need to occur. You can either re-key the information or if values present on data form are correct press shift/enter to override the error message.
- If Alpha value is entered, you will receive a Dialog Box stating "Invalid Character". Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	**PATIENT'S DOB**
FORM FIELD NUMBER:	3
FIELD DESCRIPTION:	Patient Date of Birth
FORMAT:	6 digit date (MMDDYY) 8 digit date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter DOB in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If DOB is missing, edit alert of: "PATIENT DOB SHOULD NOT BE BLANK" is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating "Numeric Field Type". Click OK. Alpha value is not allowed in this field; enter all values present except alpha values. If entire field contains alpha values, leave field blank.
- If a date in the format of MMDDYY is received, an alert will be presented to validate the century.
- If date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is after scan date, edit alert of: DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on data image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**PATIENT COND. RELATED TO** **EMPLOYMENT** **AUTO ACCIDENT** **OTHER**
FORM FIELD NUMBER:	10a, 10b, 10c
FIELD DESCRIPTION:	Worker's Comp, Auto Accident, Other Accident Indicators
FORMAT:	Each are 1 Byte Alpha Fields
VALUES:	Y or N
BUSINESS RULES:	This is a Yes/No field. If marked Yes, place a Y, if No press enter and an N will be populated. If X is between Yes and No, mark the box No.

- If, there is no check mark in the Yes or No box, enter value of "N" or press enter key and default value of "N" will be populated.
- If, there is a check mark in the No box, enter value of "N" or press enter key and default value of "N" will be populated.
- If, there is a check mark in the Yes box, enter value of "Y".
- If, there is a check mark in both boxes for a single field (i.e. Other has both Yes/No checked), enter value of "Y".
- If invalid value is received Dialog Box stating "Invalid Character", along with: Y-YES, N-NO EDIT ALERT: PLEASE REVIEW DATA is received at the bottom of the screen. Verify information on image, if unexpected value present, press enter key to allow default value of "N".

SCREEN FIELD NAME:	FOR LOCAL USE
FORM FIELD NUMBER:	10d
FIELD DESCRIPTION:	Provider Number
FORMAT:	8 or 10 digits
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	This field is for entering Provider Number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The value within the For Local Use Field when used as Referring Provider Number is validated against a Provider Stub File. Referring Provider Numbers are a length of 8 bytes.
- If enter any number different than 8 or 10 digits, edit alert of “PROVIDER NUMBER INVALID: LENGTH” is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Referring Provider is invalid, edit alert of: “PROVIDER NUMBER NOT ON FILE” is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**Date of Current Illness**
FORM FIELD NUMBER:	14
FIELD DESCRIPTION:	Accident Date
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter Accident Date in this field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If "Y" is not present in any of Fields 10A-C, this field will be protected and cleared. If "Y" is present, the accident date should be entered in this field.
- If Alpha value is entered, you will receive a Dialog Box stating "Numeric Field Type". Click OK. Alpha value is not allowed in this field; enter all values present except alpha values. If entire field contains alpha values, leave field blank.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is after scan date, edit alert of: DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on data image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**IDQ**
FORM FIELD NUMBER:	17a
FIELD DESCRIPTION:	Qualifier Code
FORMAT:	2 characters code – alpha-numeric – “ZZ”
VALUES:	ZZ
BUSINESS RULES:	Required if 17b NPI

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If qualifier code is invalid, edit alert of: “QUALIFIER ID NOT IN TABLE” is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	REF. PHYS ID#
FORM FIELD NUMBER:	17a
FIELD DESCRIPTION:	Referring Physician taxonomy number.
FORMAT:	10 Byte Alphanumeric Field
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	If qualifier code is 'ZZ', taxonomy is required.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If IDQ is a value "ZZ"
- The field is validated against a Taxonomy Stub File.
- If enter any number different than 10 digits or taxonomy is missing, edit alert of "PLEASE ENTER CORRECT LENGTH " is received at the bottom of the screen.
- If taxonomy is invalid, edit alert of "MISSING/INVALID TAXONOMY CODE. PLEASE ENTER VALID TAXONOMY CODE" is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**NPI**
FORM FIELD NUMBER:	17b
FIELD DESCRIPTION:	National Provider Identification Number
FORMAT:	10 digits - Numeric Only
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Must be 10 digits if present. Do not enter provider number in this field. If blank continue to next field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The 10 positions are validated against a Provider Stub File if present.
- If NPI is invalid, edit alert of “PLEASE ENTER VALID NPI” is received at the bottom of the screen. The field cannot be overridden
- If enter less than 10 digits, edit alert of “PLEASE ENTER 10 DIGITS” is received at the bottom of the screen.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	RESERVED FOR LOCAL USE
FORM FIELD NUMBER:	19
FIELD DESCRIPTION:	ID Number of Referring Physician (KenPac)
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Used as: ID Number of Referring Physician If present, must contain an 8 or 10 digit Provider Number. Ex: 64155344, 30609099, etc.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The value within the Reserved for Local Use Field when used as Referring Provider Number is validated against a Provider Stub File. Referring Provider Numbers are a length of 8 or 10 bytes.
- If Referring Provider is invalid, edit alert of: REFERRING PROVIDER NUMBER is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	DIAGNOSIS OR NATURE OF ILLNESS
FORM FIELD NUMBER:	21
FIELD DESCRIPTION:	ICD-9 (International Classification of Diseases) DIAGNOSIS CODE
FORMAT:	6 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Key what you see in designated fields. Do not key decimals, however you need to pick up the value after the decimal if present.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Diagnosis Codes are validated against a Stub File.
- If the first Diagnosis Code is missing, edit alert of: DIAGNOSIS CODE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- Up to 4 diagnosis codes can be present. I.E. 21-Diag 1, 21-Diag 2 can be present with nothing present in 21-Diag 3 and 21-Diag 4.
- If any of the Diagnosis Codes are invalid, edit alert of: DIAGNOSIS CODE NOT IN TABLE is received at the bottom of the screen. Verify information on image, if data was captured correctly shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If the diagnosis codes are not received in sequential order, I.E. 21-Diag 1 and 21-Diag 3 are present however 21-Diag 2 is missing, edit alert of: PREVIOUS DIAGNOSIS CODES CANNOT BE BLANK is received at the bottom of the screen. Verify information from image, if data is present and not in sequential order, shift/enter to override error message.

SCREEN FIELD NAME:	**PRIOR AUTHORIZATION NUMBER**
FORM FIELD NUMBER:	23
FIELD DESCRIPTION:	PRIOR AUTHORIZATION NUMBER
FORMAT:	10 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Used as: Prior Authorization Number If the series of numbers contains a letter (ex: 18D, etc) do not key, leave blank. If no Prior Authorization number is present, continue to next field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Prior Authorization Numbers can be up to 10 bytes.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	**FROM DATE**
FORM FIELD NUMBER:	24A
FIELD DESCRIPTION:	FROM DATE OF SERVICE IS THE BEGINNING DATE OF SERVICE FOR THE PROCEDURE
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter date on claim form. FDOS may not equal the TDOS. If no date is present in FDOS, however there is data in other fields on that detail line, enter 00000000 in this date field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If the From Date is missing from the First Detail Line, edit alert of: FIRST FROM DATE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing claim must be rejected from file and RTP'd. If data is present and was captured incorrectly enter value from image.
- If From Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If From Date is after scan date, edit alert of: SERVICE FROM DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If From Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on data image; if correct shift/enter to accept data. If data is not correct, enter value from image.

- If From Date is older than a year from current date, edit alert of: DATE IS OLDER THAN 1 YEAR is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**TO DATE**
FORM FIELD NUMBER:	24A
FIELD DESCRIPTION:	TO DATE OF SERVICE IS THE ENDING DATE OF SERVICE FOR THE PROCEDURE
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter date on claim form. TDOS may not equal the FDOS. If the TDOS equals the FDOS you can enter through the field and the FDOS will be populated in the TDOS field. If the TDOS does not equal the FDOS, enter the date on the claim form.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If the To Date is missing, press Enter key. The value in the From Date will be populated in the To Date field.
- If To Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If To Date is before From Date, edit alert of: TO DATE OLDER THAN FROM DATE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If To Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.
- If To Date is older than a year from current date, edit alert of: DATE IS OLDER THAN 1 YEAR is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.

- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**POS**
FORM FIELD NUMBER:	24b
FIELD DESCRIPTION:	Place the service was performed.
FORMAT:	2 Byte Alphanumeric
VALUES:	01, 03-08, 11-15, 20-26, 31-34, 41-42, 49-57, 60-62, 65, 71-72, 81 and 99
BUSINESS RULES:	Enter what is on the claim form (Ex: 11, 21, 22 etc). If no data is present press the enter key to continue to the next field.

- If POS is invalid, edit alert of: PLACE OF SERVICE NOT IN TABLE is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**EMG**
FORM FIELD NUMBER:	24c
FIELD DESCRIPTION:	Emergency Indicator
FORMAT:	1 character
VALUES:	Y, N or Blank
BUSINESS RULES:	Enter what is on the claim form. If no data is present press the enter key to continue to the next field.

- If EMG is invalid, edit alert of: "INVALID EMERGENCY INDICATOR. VALID VALUES ARE Y, N, OR BLANK" is received at the bottom of the screen. Verify information on image/data form; if correct shift/enter to accept data. If data is not correct, enter value from image.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).

SCREEN FIELD NAME:	**PROC**
FORM FIELD NUMBER:	24d
FIELD DESCRIPTION:	Procedure Code identifies what service was performed.
FORMAT:	13 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Procedure codes can start with a letter followed by numbers or all numbers. Modifier codes are 2 letters or numbers that follow the procedure code, they are not always present. (Ex: QX, HN, 25, 26 etc). These codes must be entered if present on the claim.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If the Procedure Code is missing, edit alert of: PROCEDURE CODE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message. If data is present and captured incorrectly enter value from image.
- If the Procedure Code is invalid, edit alert of: LOOKUP FAILED. PROCEDURE CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data is present and invalid shift/enter to override error message.
- If the value equals a length of 5, it will be validated against a Procedure Code Stub File.
- If the value equals a length of 7, the first 5 positions will be validated against a Procedure Code Stub File, the next 2 positions will be validated against a Modifier Stub File.
- If the value equals a length of 9, the first 5 positions will be validated against a Procedure Code Stub File, positions 6 and 7 and positions 8 and 9 will both be validated against a Modifier Stub File.
- If the value equals a length of 11, the first 5 positions will be validated against a Procedure Code Stub File, positions 6 and 7, positions 8 and 9 and positions 10 and 11 will all be validated against a Modifier Stub File.

- If the value equals a length of 13, the first 5 positions will be validated against a Procedure Code Stub File, positions 6 and 7, positions 8 and 9, positions 10 and 11, and positions 12 and 13 will all be validated against a Modifier Stub File.

SCREEN FIELD NAME:	**DIAGNOSIS CODE**
FORM FIELD NUMBER:	24e
FIELD DESCRIPTION:	Diagnosis Code Pointer value references the diagnosis for which the recipient is being treated as it relates to field 21.1, 21.2, 21.3 and 21.4.
FORMAT:	1 Byte Alphanumeric
VALUES:	1, 2, 3, 4
BUSINESS RULES:	These numbers refer to the Diagnosis Codes in box 21.1 to 21.4 on the claim. Enter value present on claim form. There can be more than one value present. (Ex: 1, 2, 3, 4 or 4, 2, 1, 3) – key what you see.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If the Diagnosis Code Pointer is missing, edit alert of: INVALID DIAG CODE POINTER. PLEASE VERIFY is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If the Diagnosis Code Pointer is invalid, edit alert of: DIAGNOSIS POINTER IS NOT VALID is received at the bottom of the screen. Verify information on image/data form; if data is present and invalid shift/enter to override error message.

SCREEN FIELD NAME:	CHARGES
FORM FIELD NUMBER:	24f
FIELD DESCRIPTION:	Detail amount billed for a specific procedure.
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	This is the charge for the procedure code. Key the entire dollar amount (including cents) without the decimal point. It is very important that these amounts are keyed correctly. If keyed wrong, it will cause delay in processing of the claim. Do not key negative charge lines.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If the detail Charge is missing, edit alert of: INVALID CHARGES. PLEASE VERIFY is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message.
- If the detail Charge exceeds length allowed you will receive a Completion box that states “Field Boundary”. Click OK, and then reject the claim from the file. If the length received exceeds the length of 8, the claim must be RTP’d for exceeding length allowed.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field, leave field blank. You will receive, edit alert of: INVALID CHARGES. PLEASE VERIFY at the bottom of the screen. Press enter key, field will be left blank.

SCREEN FIELD NAME:	**DAYS**
FORM FIELD NUMBER:	24g
FIELD DESCRIPTION:	Number of Days or Units being billed for a specific procedure.
FORMAT:	5 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter number of days/units.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, press enter key and field will be defaulted with a value of 1.

SCREEN FIELD NAME:	**EPSDT**
FORM FIELD NUMBER:	24h
FIELD DESCRIPTION:	Early and Periodic Screening Treatment (EPSDT) and Family Planning Indicator.
FORMAT:	1 Byte Alphanumeric
VALUES:	E, F or B
BUSINESS RULES:	Enter EPSDT/Family Planning Indicator.

- If the EPSDT indicator is invalid, edit alert of: INVALID EPSDT/FAMILY PLANNING INDICATOR is received at the bottom of the screen. Verify information on image/data form, if data is invalid, press enter out of field. Field will be left blank.

SCREEN FIELD NAME:	**NPI**
FORM FIELD NUMBER:	24j
FIELD DESCRIPTION:	National Provider Identification Number
FORMAT:	10 digits - Numeric Only
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Must be 10 digits if present. Do not enter provider number in this field. If blank continue to next field.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The 10 positions are validated against a Provider Stub File if present.
- If NPI is invalid, edit alert of “PLEASE ENTER VALID NPI” is received at the bottom of the screen.
- If enter less than 10 digits, edit alert of “PLEASE ENTER 10 DIGITS” is received at the bottom of the screen.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**IDQ**
FORM FIELD NUMBER:	24i (Shaded)
FIELD DESCRIPTION:	Qualifier Code
FORMAT:	2 character code – alpha-numeric
VALUES:	1D, ZZ
BUSINESS RULES:	Required if 24i NPI or 24j (shaded) rendering provider ID is present

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If qualifier code is invalid, edit alert of: “QUALIFIER ID NOT IN TABLE” is received at the bottom of the screen. Verify information on image; if correct shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**RENDERING PROVIDER ID#**
FORM FIELD NUMBER:	24j (shaded)
FIELD DESCRIPTION:	Rendering provider number
FORMAT:	10 Byte Alphanumeric Field
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	If qualifier code is ‘1D’, provider ID is required. If qualifier code is ‘ZZ’, taxonomy is required.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If IDQ is “1D”
- The provider ID field is validated against a Provider Stub File.
- If enter any number different than 8 digits or 10 digits or ID is missing, edit alert of “PLEASE ENTER CORRECT LENGTH ” is received at the bottom of the screen.
- If provider ID is missing, edit alert of “THE QUALIFIER IS NOT BLANK. PLEASE ENTER VALID CODE.” is received at the bottom of the screen. Verify information on image, if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If provider ID is invalid, edit alert of “MISSING/INVALID PROVIDER NUMBER. PLEASE ENTER VALID PROVIDER NUMBER” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If IDQ is “ZZ”
- The field is validated against a Taxonomy Stub File.

- If enter any number different than 10 digits or taxonomy is missing, edit alert of “PLEASE ENTER CORRECT LENGTH ” is received at the bottom of the screen.
- If taxonomy ID is missing, edit alert of “THE QUALIFIER IS NOT BLANK. PLEASE ENTER VALID CODE.” is received at the bottom of the screen. Verify information on image, if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If taxonomy is invalid, edit alert of “PLEASE ENTER VALID TAXONOMY CODE” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**EMG (shaded)**
FORM FIELD NUMBER:	24c (shaded)
FIELD DESCRIPTION:	Use for entering number of students in a group
FORMAT:	1 digit
VALUES:	1-6
BUSINESS RULES:	Enter what is on the image

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If number of student is invalid, edit alert of “PLEASE REVIEW DATA” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	**CPT/HCPCS (shaded)**
FORM FIELD NUMBER:	24d (shaded)
FIELD DESCRIPTION:	Use for entering immunization sequence
FORMAT:	Up to 6 digits
VALUES:	01-99
BUSINESS RULES:	Up to 3 occurrences allowed

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- If immunization sequence is invalid, edit alert of “INVALID IMMUNIZATION CODE. PLEASE VERIFY” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	LOCAL USE (shaded)
FORM FIELD NUMBER:	24d (shaded)
FIELD DESCRIPTION:	Use for entering school ID, employee ID, and disposition code
FORMAT:	Alpha-numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Up to 4 occurrences of disposition allowed

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- For Provider Type 20
  - If school ID is invalid, edit alert of “INVALID SCHOOL ID. PLEASE REVIEW DATA” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- For Provider Type 21
  - If 3 digit Employee ID is invalid, edit alert of “INVALID EMPLOYEE ID. PLEASE REVIEW DATA” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- For Provider Type 30
  - If 4 digit Employee ID is invalid, edit alert of “INVALID EMPLOYEE ID. PLEASE REVIEW DATA” is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.

- For Provider Type 40
  - If disposition code is invalid, edit alert of “DISPOSITION CODE X IS INVALID” (X = number of disposition code 1st, 2nd , 3rd , or 4th) is received at the bottom of the screen. Verify information on image, if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.

SCREEN FIELD NAME:	Diagnosis Pointer (Shaded)
FORM FIELD NUMBER:	24e (Shaded)
FIELD DESCRIPTION:	Use for entering time of pickup
FORMAT:	6 digits (HHMMSS)
VALUES:	HH: 00-23, MM: 00-59, SS: 00-59
BUSINESS RULES:	Enter time of pickup.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).
- For provider type 55/56
- If time is invalid, edit alert "MISSING/INVALID TIME FORMAT. PLEASE ENTER VALID TIME FORMAT." is received at the bottom of the screen. Verify information on image, if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	PATIENT'S ACCOUNT NO
FORM FIELD NUMBER:	26
FIELD DESCRIPTION:	Account number assigned by the Provider for the recipient.
FORMAT:	20 Byte Alphanumeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter first 20 positions of Patient's Account No.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.

SCREEN FIELD NAME:	TOTAL CHARGES
FORM FIELD NUMBER:	28
FIELD DESCRIPTION:	Sum of all detail charges.
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter total charges for all detail lines.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If the Total Charge is invalid (does not equal the sum of all details), edit alert of: TOTAL CLAIM CHARGE <$ AMOUNT > NOT EQUAL TO TOTAL LINE ITEM AMOUNTS <SUM OF DETAIL(s) AMOUNT> is received at the bottom of the screen. Verify information on image/data form; if data is correct press shift/enter. You will receive a Dialog Box stating "Total claim charge <xxx>not equal to total line item amounts <xxx>". Click OK. You will then be taken back to the first detail to validate value. If the value is correct, press shift/enter. You will be taken through each detail line present on the claim to validate values. Correct those that are wrong. Once the claim is in balance you will be taken back to the Total Charge field. If the claim is truly out of balance, once you are taken back to the Total Charge field, press shift/enter to override error message. If for some reason when you re-enter values in the charges field (and they were different from what was originally entered) and the claim still does not balance you will receive another edit alert of: CLAIM STILL OUT OF BALANCE. PLEASE RE-KEY AMOUNT. Verify information on image/data form; if data is valid press shift/enter to override error message, if invalid correct data.
- If the Total Charge is missing, edit alert of: TOTAL CHARGE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image/data form; if data is missing press shift/enter. You will receive the TOTAL CLAIM CHARGE error mentioned above. Follow steps mentioned above to validate values.
- If Alpha value is entered, you will receive a Dialog Box stating "Numeric Field Type". Click OK. Alpha character is not allowed in this field, leave field blank. You will receive, edit alert of: TOTAL CHARGE SHOULD NOT BE

BLANK at the bottom of the screen. Press enter key, field will be blank and out of balance error will be received. Follow steps mentioned above to validate data.

SCREEN FIELD NAME:	AMOUNT PAID
FORM FIELD NUMBER:	29
FIELD DESCRIPTION:	Amount that has been paid by other health insurance
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter amount paid by other insurance.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.

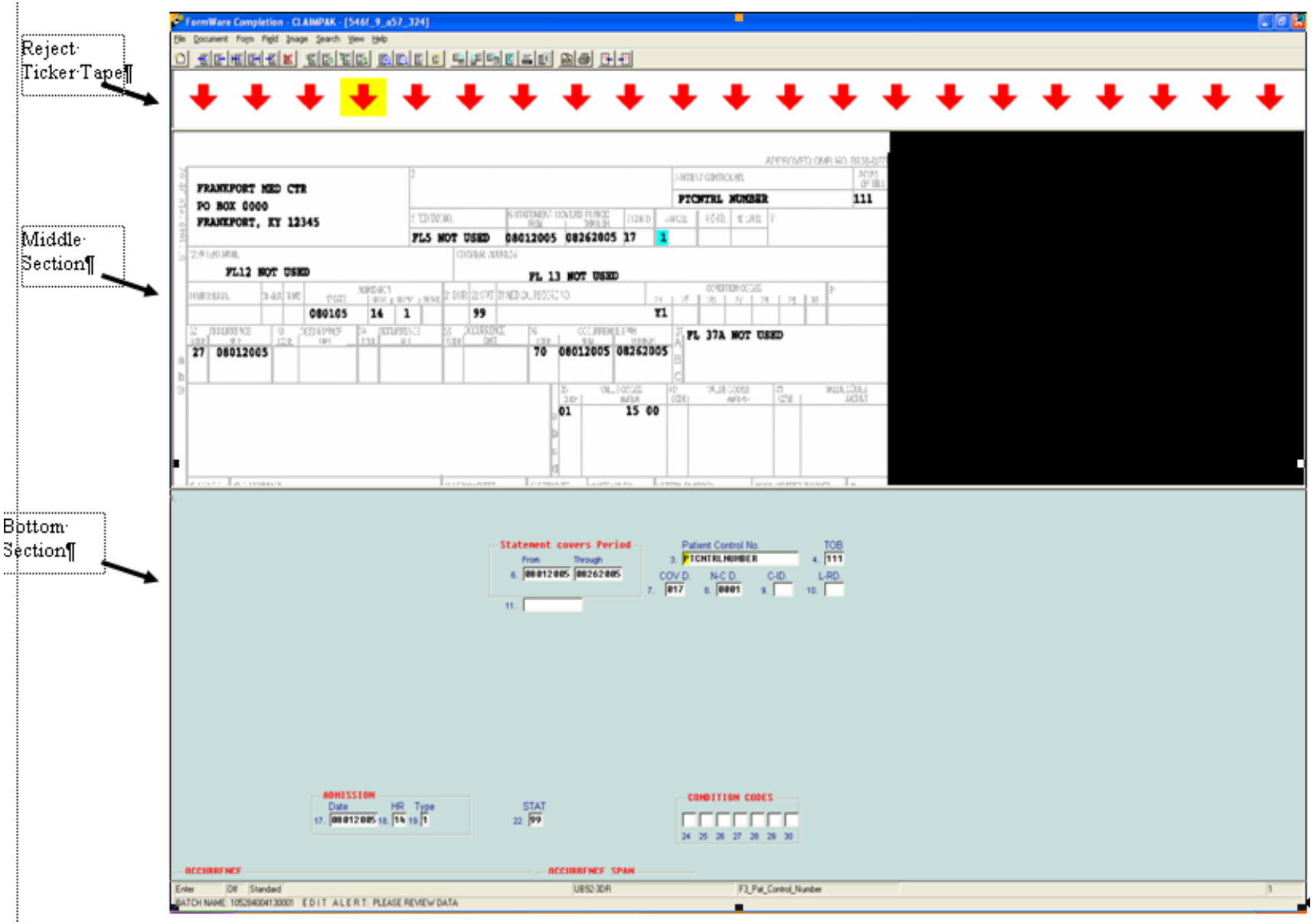
SCREEN FIELD NAME:	BALANCE DUE
FORM FIELD NUMBER:	30
FIELD DESCRIPTION:	Sum of all detail charges.
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter balance due.

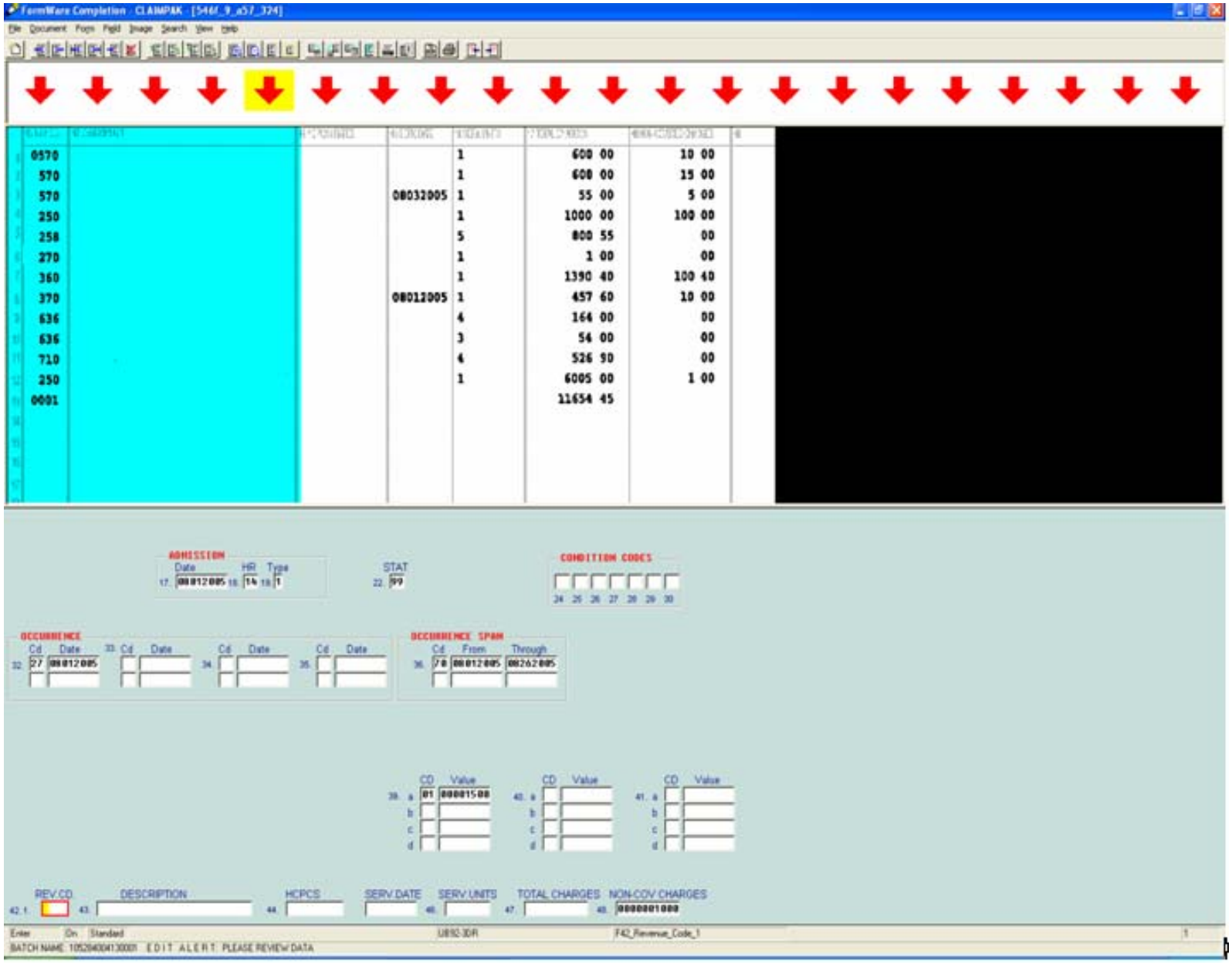
- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If Balance Due does not equal Total Charge minus Amount Paid edit alert of: BALANCE DUE < $> TOTAL – AMOUNT PAID is received at the bottom of the screen. Verify information on image/data form; if data is correct press shift/enter. You will be taken back to the Amount Paid field. If information on data form is still correct press shift/enter to override error message. If data was invalid correct data.

SCREEN FIELD NAME:	LAST CHANCE
FORM FIELD NUMBER:	N/A
FIELD DESCRIPTION:	Last opportunity to verify claim information
FORMAT:	1 Byte
VALUES:	No specific value editing is done.
BUSINESS RULES:	Hit enter when ready to move to the next claim.

- This is the last chance to correct ANYTHING on this claim. Once any key is pressed the document can not be retrieved to make any modifications.
- Prior to pressing any keying, scroll up on the claim to see if there are ANY handwritten fields that were not captured from OCR. The most common fields that are handwritten and not captured are: 17a, 19, 23, and 24 shaded fields.

27.7.5 KY MMIS UB92 Completion claim information:




FormWare Completion - CLAIMPAK - [546f_9_a57_324]
File Document Form Field Image Search View Help
0570 1 600 00 10 00
570 1 600 00 15 00
570 08032005 1 55 00 5 00
250 1 1000 00 100 00
258 5 800 55 00
270 1 1 00 00
360 1 1390 40 100 40
370 08012005 1 457 60 10 00
636 4 164 00 00
636 3 54 00 00
710 4 526 90 00
250 1 6005 00 1 00
0001 11654 45
ADMISSION
Date HR Type
17. 08012005 18. 14 19. 1
STAT
22. 99
CONDITION CODES
24 25 26 27 28 29 30
OCCURRENCE
Cd Date
32. 27 08012005
33. Cd Date
34. Cd Date
35. Cd Date
OCCURRENCE SPAN
Cd From Through
36. 70 08012005 08262005
CD Value
39. a 01 00001500
40. a
41. a
REV.CD
42. 1
DESCRIPTION
43.
HCPCS
44.
SERV.DATE
SERV.UNITS
46.
TOTAL CHARGES
47.
NON-COV CHARGES
48. 0000001000
Enter On Standard
U892-3DR
F42_Revenue_Code_1
BATCH NAME 105284004130001 EDIT ALERT PLEASE REVIEW DATA

FormWare Completion - CLAIMPAK - [546(_9_a57_324)]

File Document Form Field Image Search View Help

01 15 00

0570				1	600 00	10 00
570				1	600 00	15 00
570			08032005	1	55 00	5 00
250				1	1000 00	100 00
258				5	800 55	00
270				1	1 00	00
360				1	1390 40	100 40
370			08012005	1	457 60	10 00
636				4	164 00	00
636				3	54 00	00
710				4	526 90	00
250				1	6005 00	1 00
0001					11654 45	

	CD	Value		CD	Value		CD	Value
39. a	01	00001500	40. a			41. a		
b			b			b		
c			c			c		
d			d			d		

	REV.CD.	DESCRIPTION		HCPCS		SERV.DATE		SERV.UNITS		TOTAL CHARGES		NON-COV CHARGES
42.1.	0570	43.	44.		45.		46.	1	47.	0000060000	48.	0000001000
2.	0570							1		0000060000		0000001500
3.	0570					08032005		1		0000005500		0000000500
4.	0570							1		0000100000		0000010000
5.	0250							5		0000080055		0000000000

	REV.CD.	DESCRIPTION		HCPCS		SERV.DATE		SERV.UNITS		TOTAL CHARGES		NON-COV CHARGES
42.6.	0258	43.	44.		45.		46.	1	47.	0000000100	48.	0000000000
7.	0270							1		0000139040		0000010040
8.	0360					08012005		1		0000045760		0000001000
9.	0370							4		0000016400		0000000000

Enter Off Standard UB92-3DR F40_Line_Non_Covered_5 1

BATCH NAME: 1052640041 30001

FormWare Completion - CLAIMPAK - [546(_9_a57_324)]

File Document Form Field Image Search View Help

6	270				1	1	00	00	
7	360				1	1390	40	100 40	
8	370			08012005	1	457	60	10 00	
9	636				4	164	00	00	
10	636				3	54	00	00	
11	710				4	526	90	00	
12	250				1	6005	00	1 00	
13	0001					11654	45		

	REV.CD.	DESCRIPTION	HCPCS	SERV.DATE	SERV.UNITS	TOTAL CHARGES	NON-COV CHARGES
42.6.	0258	43.	44.	45.	46. 1	47. 0000000100	48. 0000000000
7.	0270				1	0000139040	0000010040
8.	0360			08012005	1	0000045760	0000001000
9.	0370				4	0000016400	0000000000
10.	0636				3	0000005400	0000000000
11.	0636				4	0000052690	0000000000
12.	0710				1	0000060500	0000000100
13.	0250					0000000000	
14.							
15.							
16.							
17.							

	REV.CD.	DESCRIPTION	HCPCS	SERV.DATE	SERV.UNITS	TOTAL CHARGES	NON-COV CHARGES
42.18.		43.	44.	45.	46.	47.	48.
19.							
20.							
21.							
22.							

Enter Off Standard UB92-3DR F48_Line_Non_Covered_17 1

BATCH NAME: 1052940041 30001

FormWare Completion - CLAIMPAK - [546f_9_a57_324]

File Document Form Field Image Search View Help

TPL OTHERCARRIER N Y 597 36
KENTUCKY MEDICAID 01022078 N Y
FL 57 NOT USED DUE FROM PATIENT 15 00
ALLEN, M 01 OTHERCARRIER ID
ALLEN, MICHAEL 01 4024699714
63015000
3439 7845 7070 3439
FL 82ATTENDING PROVIDER

Insured's Last Name First
58. A.
B.
C. ALLEN MI

CERT.-SSN-HIC.-ID NO.
60. W024699714

Treatment Autho. CD
63.A.
B.
C.

PRIN. DIAG.CD. Code Code Code Code Code Code Code Code Code ADM. DIAG.CD.
67. 3439 68. 7845 69. 7070 70. 71. 72. 73. 74. 75. 76. 3439

Principal Proc. Code Date
80. 2309 08012005

Other Procedure Code Date Code Date
81. A. B.

Attending Phys. ID
82. FL82

Enter Off Standard UB92-3DR FSB_Insured_Last_Name_3 1
BATCH NAME: 10528400413 0001

FormWare Completion - CLAIMPAK - [5461_9_a57_324]

File Document Form Field Image Search View Help

ALLEN, M 01 OTHERCARRIER ID
ALLEN, MICHAEL 01 4024699714

63015000

3439 7845 7070 3439

9 2309 08012005

FL 82ATTENDING PROVIDER

SIGNATURE ON FILE 090105

Principal Proc.

	Code	Date
80.	2309	08012005

Other Procedure

	Code	Date		Code	Date
81. A.			B.		
C.			D.		
E.					

Attending Phys. ID
82. FL82

Other Phys. ID
83.

Other Phys. ID

Provider Rep. Date
85. 10102005

Last Chance

Enter On Standard U892-3DR F_LastChance 1

BATCH NAME: 105284004130001

27.7.6 UB92 Data Form Fields

SCREEN FIELD NAME:	PATIENT CONTROL NO
FORM FIELD NUMBER:	3
FIELD DESCRIPTION:	Account number assigned by the Provider for the recipient.
FORMAT:	20 Byte Alphanumeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter first 20 positions of Patient's Control Number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.

SCREEN FIELD NAME:	**TOB**
FORM FIELD NUMBER:	4
FIELD DESCRIPTION:	Code indicating the specific type of bill (inpatient, outpatient, nursing home (LTC), etc.
FORMAT:	3 Numeric
VALUES:	Position 1 and 2: 1 - 9 Position 3: 0 - 9
BUSINESS RULES:	Enter type of bill.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If TOB is missing, edit alert of: TYPE OF BILL SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to leave field blank. If data is present and was captured incorrectly enter value from image.
- If TOB does not equal a length of 3 or positions 1 and 2 contain a value of 0, edit alert: TOB MISSING/INVALID BLANK is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and was captured incorrectly enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	STATEMENT COVERS PERIOD (FROM)
FORM FIELD NUMBER:	6
FIELD DESCRIPTION:	The beginning service dates of the period included on this claim.
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter from date of service.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If the From Date is missing, edit alert of: STATEMENT FROM DATE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to leave field blank. If data is present and was captured incorrectly enter value from image.
- If From Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If From Date is after scan date, edit alert of: FDOS IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If From Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.

- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**STATEMENT COVERS PERIOD (THROUGH)**
FORM FIELD NUMBER:	6
FIELD DESCRIPTION:	The ending service dates of the period included on this claim.
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter to date of service.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If the Through Date is missing, edit alert of: STATEMENT TO DATE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to leave field blank. If data is present and was captured incorrectly enter value from image.
- If Through Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Through Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**COV D**
FORM FIELD NUMBER:	7
FIELD DESCRIPTION:	Total number of days covered for the accommodations billed on this claim.
FORMAT:	3 byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter covered days.

- If Covered Days are greater than number of days between From Date-Through Date (Field 6), edit alert: COVERED DAYS MISSING/INVALID BLANK is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**N-C D**
FORM FIELD NUMBER:	8
FIELD DESCRIPTION:	Total of Non-Covered Days for the accommodations billed on this claim.
FORMAT:	4 byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter non-covered days.

- If Non-Covered Days are greater than number of Covered Days plus Non-Covered Days between From Date-Through Date (Field 6), edit alert: COVERED DAYS PLUS NON-COVERED DAYS INVALID BLANK is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**C-I D**
FORM FIELD NUMBER:	9
FIELD DESCRIPTION:	Coinsurance Days – The Inpatient Medicaid days occurring after the 60th day and before the 91st day in a single spell of illness.
FORMAT:	4 byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter Coinsurance number of days.

- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**L-R D**
FORM FIELD NUMBER:	10
FIELD DESCRIPTION:	Lifetime Reserve Days – Additional days of Inpatient hospital services after using 90 days of inpatient hospital services during a spell of illness.
FORMAT:	3 byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter lifetime reserve days.

- If Alpha value is entered, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. If entire field contains alpha values, leave field blank.

SCREEN FIELD NAME:	**11**
FORM FIELD NUMBER:	11 (Unlabeled Field)
FIELD DESCRIPTION:	Nursing Facility Provider Number of the facility where the hospice recipient resides.
FORMAT:	10 byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter nursing facility provider number.

- If TOB (field 4) first two positions equal 81 or 82 field 11 is used for Nursing Facility Provider Number. If a value is present it will be validated against a Provider Stub File. First 8 positions of the value are validated. Note: Field 11 is only validated if TOB equals 81 or 82.
- If TOB (field 4) first two positions equal 81 or 82 and value present does not equal a valid Nursing Facility Provider Number, edit alert: NURSING FACILITY ID (FL 11 MISSING/INVALID BLANK is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**Admission Date**
FORM FIELD NUMBER:	17
FIELD DESCRIPTION:	The date the recipient was admitted to the provider for inpatient care, outpatient care or start of care.
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter admission date.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If Admission Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Admission Date is after scan date, edit alert of: ADMISSION DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Admission Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**Admission Hour**
FORM FIELD NUMBER:	18
FIELD DESCRIPTION:	The hour during which the recipient was admitted for inpatient or outpatient care.
FORMAT:	2 bytes Numeric
VALUES:	00-23, 99 or Blank
BUSINESS RULES:	Enter admission hour.

- If Admission Hour is invalid, edit alert of: ADMISSION HOUR SHOULD BE 00-23, 99 is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “INVALID CHARACTER”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You will then receive, edit alert of: ADMISSION HOUR MISSING/INVALID at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.

SCREEN FIELD NAME:	**Admission Type**
FORM FIELD NUMBER:	19
FIELD DESCRIPTION:	A code indicating the priority of admission
FORMAT:	1 byte Numeric
VALUES:	1-4, 9 or Blank
BUSINESS RULES:	Enter admission type.

- If Admission Type is invalid, edit alert of: ADMISSION TYPE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “INVALID CHARACTER”. Click OK. Verify information on image/data form; if value contains Alpha, leave field blank, otherwise enter value on data form.

SCREEN FIELD NAME:	**STAT**
FORM FIELD NUMBER:	22
FIELD DESCRIPTION:	Indicates the recipient's disposition at the time of billing for that period of care.
FORMAT:	2 bytes Numeric
VALUES:	01-99
BUSINESS RULES:	Enter patient's status.

- If Alpha value is entered, you will receive a Dialog Box stating "INVALID CHARACTER". Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters.

SCREEN FIELD NAME:	CONDITION CODES
FORM FIELD NUMBER:	24 – 30
FIELD DESCRIPTION:	Codes used to identify conditions relating to this bill that may affect payer processing.
FORMAT:	2 bytes Numeric
VALUES:	01-99, A0-ZZ
BUSINESS RULES:	Enter condition codes.

- If Condition Code is invalid, edit alert of: CONDITION CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If the Condition Codes are not received in sequential order, I.E. 25 and 26 are present however 24 is missing OR 24 and 27 are present and 25 and 26 are missing, edit alert of: CONDITION CODE MISSING/INVALID. PLEASE CHECK THE PREVIOUS CONDITION CODE AS IT IS BLANK is received at the bottom of the screen. Verify information on image/data form, if data is present and values are valid or invalid, and the fields received are not in sequential order, shift/enter to override error message.

SCREEN FIELD NAME:	**OCCURRENCE (Cd and Date)**
FORM FIELD NUMBER:	32 – 35 (lines a/b)
FIELD DESCRIPTION:	Code and Date defining a significant event relating to this claim that may affect payer processing.
FORMAT:	Code : 2 bytes Alphanumeric Date : 6 Digit Date (MMDDYY), 8 Digit Date (MMDDCCYY)
VALUES:	01-99, A0-ZZ MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter occurrence codes and dates.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received within the date field, once you press the enter key the century will be automatically populated.
- If Occurrence Code is invalid, edit alert of: OCCURRANCE CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Occurrence Code is present and Occurrence Date is missing, edit alert of: OCCURANCE DATE MISSING/INVALID. OCCURANCE DATE REQUIRED IF CODE PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Occurrence Date is present and Occurrence Code is missing, edit alert of: OCCURANCE DATE MISSING/INVALID. OCCURANCE CODE REQUIRED IF DATE PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.

- If Occurrence Date is invalid, edit alert of: OCCURANCE DATE MISSING/INVALID. PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Occurrence Date is after scan date, edit alert of: OCCURRANCE DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Occurrence Date is a greater than current date, edit alert of: OCCURANCE DATE MISSING/INVALID. DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is received in the Occurrence Date field, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	OCCURRENCE SPAN (Cd and From Through)
FORM FIELD NUMBER:	36 (lines a/b)
FIELD DESCRIPTION:	Code and From/Through Date defining a significant event relating to this claim that may affect payer processing.
FORMAT:	Code : 2 bytes Alphanumeric From/Through Dates : 6 Digit Date (MMDDYY), 8 Digit Date (MMDDCCYY)
VALUES:	70-79, M0-M4 MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter occurrence span codes and dates.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received within the date field, once you press the enter key the century will be automatically populated.
- If Occurrence Span Code is invalid, edit alert of: OCCURRENCE SPAN CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Occurrence Span Code is present and Occurrence From Date is missing, edit alert of: OCCURRANCE SPAN FROM DATE MISSING/INVALID. OCCURANCE DATE REQUIRED IF SPAN PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Occurrence Span Code and Occurrence From Date are present and Occurrence Through Date is missing, edit alert of: OCCURRENCE SPAN THROUGH DATE MISSING/INVALID is received at the bottom of the screen. Verify information on

image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image

- If Occurrence Span Code is missing and Occurrence From Date is present, edit alert of: OCCURRENCE SPAN CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Occurrence Span From or Through Date is invalid, edit alert of: OCCURRENCE SPAN FROM DATE MISSING/INVALID or OCCURRANCE SPAN TO DATE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Occurrence Span From or Through Date is after scan date, edit alert of: OCCURRENCE SPAN FROM or THROUGH DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Occurrence Span From or Through Date is a greater than current date, edit alert of: OCCURRANCE SPAN FROM or THROUGH DATE MISSING/INVALID. DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is received in the Occurrence Span From or Through Date fields, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**CD and Value**
FORM FIELD NUMBER:	39 - 41 (lines a/d)
FIELD DESCRIPTION:	A code structure to relate amounts or values to identified data elements necessary to process this claim as qualified by the payer organization.
FORMAT:	Code : 2 bytes Alphanumeric Amount: 8 bytes Numeric
VALUES:	01-99, A0-ZZ
BUSINESS RULES:	Enter value code and amounts.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If Value Code is invalid, edit alert of: VALUE CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Value Code is present and Value Amount is missing, edit alert of: VALUE AMOUNT REQUIRED IF CODE IS PRESENT, is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Value Amount is present and Value Code is missing, edit alert of: VALUE CODE REQUIRED IF AMOUNT PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Alpha value is received in the Value Amount field, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; leave field blank.

Detail Line Fields: **42 to 48**	The UB92 claim allows up to 23 details on a single page. The last detail on the claim must contain a value of 001 or 0001 in Field 42. When Field 42 contains 001 or 0001, the value in Field 47 represents the total charge for all details billed on that claim. If there is no 001 or 0001 in FL 42, and there are not 23 details present on the claim, completion operator will enter 001 or 0001 in FL 42, and no value will be entered in FL 47. If there is no 001 or 0001 in FL 42, there are 23 details present on the claim and it is not a multi-page claim, completion operator will reject the claim from the file. The claim must be RTP'd for no total charge on claim.

SCREEN FIELD NAME:	REV. CD
FORM FIELD NUMBER:	42
FIELD DESCRIPTION:	A code which identifies a specific accommodation, ancillary service or billing calculation.
FORMAT:	4 bytes Numeric
VALUES:	001-999, 0001-0999
BUSINESS RULES:	Enter revenue codes.

- If the Revenue Code is invalid, edit alert of: REVENUE CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message. If data is present and captured incorrectly enter value from image.
- If the Revenue Code is missing from the first detail line, edit alert of: FIRST DETAIL LINE CANNOT BE BLANK. PLEASE ENTER REVENUE CODE is received at the bottom of the screen. If the claim is being entered KFI, and no detail line information is present on the first detail, enter the information from the next available detail on the first detail line, and enter the next available detail on the second detail and so on. If there are no detail lines present, RTP the claim.
- If the value in the Revenue Code field is a length of 3 it will be autoplugged with a leading “0”.
- If Alpha value is received in the Revenue Code field, you will receive a Dialog Box stating “Invalid Character”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: REVENUE CODE MISSING/INVALID at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	DESCRIPTION
FORM FIELD NUMBER:	43
FIELD DESCRIPTION:	Detail Service Date if first two positions of field 4 (TOB) equal 81 or 82 (otherwise field 43 not used).
FORMAT:	6 Digit Date (MMDDYY) and 6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY) and 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	The business rules for field 43 apply only if the first two positions of TOB (Field 4) contain “81” or “82”. All other TOB, the field will be cleared and bypassed.

- If no value is present in this field, edit alert of: KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data is missing shift/enter to override error message. If data is present and captured incorrectly enter value from image.
- If an invalid value is present in this field, edit alert of: DATE IS INVALID. KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.
- If valid date(s) are received, however they are not in-between the Statement covers Period (Field 6), edit alert of: FIRST DATE IS NOT BETWEEN F6 FROM AND TO DATE. KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.

- If valid dates are received, however the second date is before the first date, edit alert of: SECOND DATE IS BEFORE FIRST DATE. KETY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.
- If valid dates are received, however the second date is not in-between the Statement covers Period (Field 6), edit alert of: SECOND DATE IS NOT BETWEEN F6 FROM AND TO DATE. KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.

SCREEN FIELD NAME:	HCPCS
FORM FIELD NUMBER:	44
FIELD DESCRIPTION:	HCPCS code that identifies a specific accommodation and/or ancillary service billing calculation. OR If the first two positions of TOB (Field 4) contain “81” or “82” this field may be used for a date.
FORMAT:	11 bytes Alphanumeric 6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	Value entered is compared against stub file. MM: 01-12, DD: 01-31
BUSINESS RULES:	If revenue code is 100-219, clear and bypass this field when TOB is not “81” or “82”.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If the first two positions of TOB (Field 4) DO NOT contain “81” or “82” and an invalid value is received, edit alert of: HCPCS/RATE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.
- If the first two positions of TOB (Field 4) DO NOT contain “81” or “82” and the value equals a length of 5, it will be validated against a HCPCS Code Stub File.
- If the first two positions of TOB (Field 4) DO NOT contain “81” or “82” and the value equals a length of 7 or more, the first 5 positions will be validated against a HCPCS Code Stub File, each of the next 2 positions will be validated against a Modifier Stub File.

- If first two positions of TOB (Field 4) contain “81” or “82 and an invalid date is present in this field, edit alert of: DATE IS INVALID. KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.

- If first two positions of TOB (Field 4) contain “81” or “82 and a valid date is received, however it is not in-between the Statement covers Period (Field 6), edit alert of: DATE IS NOT BETWEEN F6 FROM AND TO DATE. KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.

- If first two positions of TOB (Field 4) contain “81” or “82 and a valid date is received, however it is a future date, edit alert of: DATE IS NOT ON OR BEFORE SCAN DATE. KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.

- If first two positions of TOB (Field 4) contain “81” or “82 and a valid date is received, however it is older than a year from current date, edit alert of: DATE IS OLDER THAN ONE YEAR. KEY EITHER TWO 6 OR 8 DIGIT DATES SEPARATED BY A SPACE OR A SINGLE 6 OR 8 DIGIT DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.

SCREEN FIELD NAME:	**SERV. DATE**
FORM FIELD NUMBER:	45
FIELD DESCRIPTION:	Date Services were provided
FORMAT:	6 Digit Date (MMDDYY) 8 Digit Date (MMDDCCYY)
VALUES:	MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter date of service.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received, once you press the enter key the century will be automatically populated.
- If Service Date is invalid, edit alert of: PLEASE ENTER A VALID DATE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Service Date is after scan date, edit alert of: SERVICE DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Service Date is a greater than current date, edit alert of: DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Service Date is not in-between the Statement covers Period (Field 6), edit alert of: SERVICE DATE DOES NOT FALL BETWEEN F6 DATES is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.

- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	SERV. UNITS
FORM FIELD NUMBER:	46
FIELD DESCRIPTION:	Quantity or number of treatments being bill for this revenue code.
FORMAT:	7 bytes Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter service units.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered and any value after the punctuation is dropped.
- For claims that are KFI, DO NOT enter any value after the punctuation.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters.

SCREEN FIELD NAME:	TOTAL CHARGES
FORM FIELD NUMBER:	47
FIELD DESCRIPTION:	Total amount billed for each revenue code on claim. (Detail line charges when revenue code does not equal 001 or 0001. If revenue code equals 001 or 0001 this value is the sum of all details).
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter total charges.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If the Total Charges is missing, edit alert of: BLANK LINE CHARGES. PLEASE REVIEW is received at the bottom of the screen. Verify information on image/data form; if data is missing press shift/enter.
- Where Revenue Code 001 or 0001 is present field 47 equals the sum of all detail Total Charges. If the Total Charges is invalid (does not equal the sum of all details), edit alert of: TOTAL CLAIM CHARGE <$ AMOUNT > NOT EQUAL TO TOTAL LINE ITEM AMOUNTS <SUM OF DETAIL(s) AMOUNT> is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. You will receive a Dialog Box stating “Total claim charge <xxx>not equal to total line item amounts <xxx>”. Click OK. You will then be taken back to the first detail to validate value. If the value is correct, press shift/enter. You will be taken through each detail line present on the claim to validate values. Correct those that are wrong. Once the claim is in balance you will be taken back to the Total Charge field. If the claim is truly out of balance, once you are taken back to the Total Charge field, press shift/enter to override error message. If for some reason when you re-enter values in the charges field (and they were different from what was originally entered) and the claim still does not balance you will receive another edit alert of: CLAIM STILL OUT OF BALANCE. PLEASE RE-KEY AMOUNT. Verify information on image/data form; if data is valid press shift/enter to override error message, if invalid correct data.

- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field, leave field blank. You will receive, edit alert of: TOTAL CHARGE SHOULD NOT BE BLANK at the bottom of the screen. Press enter key, field will be blank and out of balance error will be received. Follow steps mentioned above to validate data.

SCREEN FIELD NAME:	NON-COV CHARGES
FORM FIELD NUMBER:	48
FIELD DESCRIPTION:	Charges that are non-covered that correspond to a specific revenue code
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter non-covered charges.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If the Non-Covered Charge is greater than the Total Charge, edit alert of: NON COVERED LINE CHARGES MISSING/INVALID. GREATER THAN CHARGES FIELD is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field, leave field blank.
- Fields 50 through Field 66 (Lines A-C) contain Payer and Recipient Information.
- The data form allows for entry of up to 3 lines on a single field (FL x-a, FL x-b, FL x-c). For each line (a, b, or c), all fields are associated together. For example, if 50b contains Kentucky Medicaid, 51b would contain the Medicaid Provider Number, 58b would contain the recipient last/first name, 60b would contain the MAID.
- The word Medicaid must be present on line 50 a, b or c on the form. (Note: “Kentucky Medicaid”, “KY Medicaid”, “Medicaid”, “Medicaid 19”, “KY Medicaid Out” “KY MCD” are examples of values that may be in FL 50a, b or c.

- If there is other payer information present on the claim, you must key “T” corresponding to the name of that payer from field 50. For example: If the recipient has Medicare you would enter “T” in field 50. Currently the only other field that has to be entered for other payers is field 54 (Prior Payments).

SCREEN FIELD NAME:	**PAYER**
FORM FIELD NUMBER:	50 (a-c)
FIELD DESCRIPTION:	Name of Payer Organization from which the provider may expect some payment for the claim.
FORMAT:	25 Bytes Alphanumeric
VALUES:	Line 50a, b or c – “D” for Medicaid, “E” for Medicare and “T” for any other values
BUSINESS RULES:	Enter value of payer.

- If the form of the word “Medicaid” is not present in 50 a, b or c and there is not a valid KY MMIS Provider number in 51 a, b or c, the claim must be rejected from the file and RTP’d for missing/invalid PIN#.
- If you get to 50c, and the word of “Medicaid” is not found on the form (“D” keyed on the screen), edit alert of: MEDICAID PAYER MISSING/INVALID is received at the bottom of the screen. This edit can not be overridden. If a Valid Medicaid Provider number is NOT found in the associated field 51, the claim must be rejected from the file and RTP’d.
- If an invalid value is entered, you will receive a Dialog Box stating “MEDICAID PAYER MISSING/INVALID”. Validate the appropriate code has been entered.

SCREEN FIELD NAME:	**PROVIDER NO.**
FORM FIELD NUMBER:	51 (a-c)
FIELD DESCRIPTION:	Kentucky Medicaid Provider Number
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter provider number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 8 or 10 positions within the Provider No. where field 50 equals “Medicaid” are validated against a Provider Stub File.
- If Provider Number is missing or invalid, edit alert of: PROVIDER NUMBER MISSING/INVALID. PLEASE ENTER VALID PROVIDER NUMBER OR REJECT THE CLAIM is received at the bottom of the screen. Verify information on image; if data is missing or invalid, the claim must be rejected from the file to be Returned to Provider (RTP’d).
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. If Alpha character is present in field, the claim must be rejected from the file to be Returned to Provider (RTP’d).

SCREEN FIELD NAME:	PRIOR PAYMENTS
FORM FIELD NUMBER:	54 (a-c)
FIELD DESCRIPTION:	Amount which has been paid by services other than Medicaid such as other insurance amounts.
FORMAT:	8 Byte Numeric
VALUES:	No specific value editing is done.
BUSINESS RULES:	Enter payments by other insurance.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered. KY MMIS will assume decimal point before the last two bytes present in the field.
- If Alpha value is entered, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field, leave field blank.

SCREEN FIELD NAME:	**Unlabeled Field (KENPAC NUMBER)**
FORM FIELD NUMBER:	56
FIELD DESCRIPTION:	The recipient's KenPAC provider number, if applicable.
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter Kenpac number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 8 or 10 positions within this field are validated against a Provider Stub File.
- If KENPAC Number is invalid, edit alert of: KENPAC PROVIDER NUMBER INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is present and captured incorrectly enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating "Invalid Character". Click OK. If Alpha character is present in field, leave field blank.

SCREEN FIELD NAME:	**INSURED'S LAST NAME**
FORM FIELD NUMBER:	58 (a-c)
FIELD DESCRIPTION:	Recipient Last Name
FORMAT:	Up to 5 Byte Alpha value
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter first 5 positions of last name.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID is validated against a Recipient Stub File.
- If Insured's Last Name is missing, edit alert of: INSURED LAST NAME SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Numeric value is entered, you will receive a Dialog Box stating "Alpha Field Type". Click OK. Numeric value is not allowed in this field; enter all values present except numeric values. If entire field contains numeric values, leave field blank.

SCREEN FIELD NAME:	**FIRST**
FORM FIELD NUMBER:	58 (a-c)
FIELD DESCRIPTION:	Recipient First Name
FORMAT:	Up to 2 Byte Alpha value
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter first 2 positions of first name.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID is validated against a Recipient Stub File.
- If Insured's First Name is missing, edit alert of: INSURED FIRST NAME SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to leave field blank. If data is not correct, enter value from image.
- If Numeric value is entered, you will receive a Dialog Box stating "Alpha Field Type". Click OK. Numeric value is not allowed in this field; enter all values present except numeric values. If entire field contains numeric values, leave field blank.

SCREEN FIELD NAME:	**CERT.-SSN-HIC.-ID NO.**
FORM FIELD NUMBER:	60 (a-c)
FIELD DESCRIPTION:	Recipient's Medical Assistance Identification (MAID) Number
FORMAT:	10 Byte Numeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter member's number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- The first 5 bytes of the Recipient Last Name, first 2 bytes of the First Name along with the MAID ID are validated against a Recipient Stub File.
- If MAID ID entered does not match the stub file lookup being performed against the Recipient's Name/ID, edit alert of: LOOPKUP FAILED. MAID MISSING/INVALID is received at the bottom of the screen. The cursor will be positioned in the Insured's Last Name Field. Validation of the Insured's Last Name, First Name and MAID will need to occur. You can either re-key the information or if values present on data form are correct press shift/enter to override the error message. If data is not correct, enter value from image.
- If MAID is missing, edit alert of: CERTIFICATE NUMBER SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image, if data is missing or invalid shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If Alpha value is entered, you will receive a Dialog Box stating "Invalid Character". Click OK. Alpha character is not allowed in this field; leave field blank.

SCREEN FIELD NAME:	**TREATMENT AUTHO. CD**
FORM FIELD NUMBER:	63 (a-c)
FIELD DESCRIPTION:	Prior Authorization Number that has been assigned by the Payer
FORMAT:	10 Bytes Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter prior authorization number.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Prior Authorization Numbers can be up to 10 bytes.
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).

SCREEN FIELD NAME:	PRIN. DIAG. CODE
FORM FIELD NUMBER:	67
FIELD DESCRIPTION:	Principal ICD-9 (International Classification of Diseases) DIAGNOSIS CODE
FORMAT:	6 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter principal diagnosis code.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Diagnosis Codes are validated against a Stub File.
- If the Principal Diagnosis Code is missing, edit alert of: PRINCIPAL DIAGNOSIS CODE SHOULD NOT BE BLANK is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If the Principal Diagnosis Code is invalid, edit alert of: PRINCIPAL DIAGNOSIS CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image; if data is missing shift/enter to override error message. If data is present and was captured incorrectly enter value from image.

SCREEN FIELD NAME:	Code
FORM FIELD NUMBER:	68 - 75
FIELD DESCRIPTION:	ICD-9 (International Classification of Diseases) DIAGNOSIS CODE
FORMAT:	6 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter other diagnosis codes.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Diagnosis Codes are validated against a Stub File.
- If any of the Diagnosis Codes are invalid, edit alert of: OTHER DIAGNOSIS CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image, if data was captured correctly shift/enter to override error message. If data is present and was captured incorrectly enter value from image.

SCREEN FIELD NAME:	ADM. DIAG. CD
FORM FIELD NUMBER:	76
FIELD DESCRIPTION:	Admitting ICD-9 (International Classification of Diseases) DIAGNOSIS CODE
FORMAT:	6 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter admission diagnosis codes.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Diagnosis Codes are validated against a Stub File.
- If Admitting Diagnosis Code is invalid, edit alert of: ADMITTING DIAGNOSIS CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image, if data was captured correctly shift/enter to override error message. If data is present and was captured incorrectly enter value from image.

SCREEN FIELD NAME:	E. Code
FORM FIELD NUMBER:	76
FIELD DESCRIPTION:	External Cause of Injury ICD-9 (International Classification of Diseases) DIAGNOSIS CODE
FORMAT:	6 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter E-code.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- Diagnosis Codes are validated against a Stub File.
- If E Code is invalid, edit alert of: E CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image, if data was captured correctly shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If E Code does not begin with E, edit alert of: E CODE SHOULD START WITH E is received at the bottom of the screen. Verify information on image, if data was captured correctly shift/enter to override error message. If data is present and was captured incorrectly enter value from image.
- If E Code is received on a non-inpatient claim, edit alert of: E CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image, if data was captured correctly shift/enter to override error message. If data is present and was captured incorrectly enter value from image.

SCREEN FIELD NAME:	**PRINCIPAL PROC (Code and Date)**
FORM FIELD NUMBER:	80
FIELD DESCRIPTION:	Principal Procedure Code and Date defining the primary surgical procedure rendered.
FORMAT:	Code : 7 bytes Alphanumeric Date : 6 Digit Date (MMDDYY), 8 Digit Date (MMDDCCYY)
VALUES:	Value entered is compared against stub file. MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter principal procedure code and date.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received within the date field, once you press the enter key the century will be automatically populated.
- If Principal Procedure Code is invalid, edit alert of: PRINCIPAL PROCEDURE CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Principal Procedure Code is present and Principal Procedure Date is missing, edit alert of: PRINCIPAL PROCEDURE DATE REQUIRED IF CODE IS PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Principal Procedure Date is present and Principal Procedure Code is missing, edit alert of: PRINCIPAL PROCEDURE DATE MISSING/INVALID. PROCEDURE CODE REQUIRED IF DATE PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data.
- If Principal Procedure Date is invalid, edit alert of: PRINCIPAL PROCEDURE DATE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data.

- If Principal Procedure Date is after scan date, edit alert of: PRINCIPAL PROCEDURE DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data.
- If Principal Procedure Date is a greater than current date, edit alert of: PRINCIPAL PROCEDURE DATE MISSING/INVALID. DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data
- If Alpha value is received in the Principal Procedure Date field, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**Other Procedure (Code and Date)**
FORM FIELD NUMBER:	81 a-e
FIELD DESCRIPTION:	Code that describes the secondary surgical procedure(s) provided.
FORMAT:	Code : 7 bytes Alphanumeric Date : 6 Digit Date (MMDDYY), 8 Digit Date (MMDDCCYY)
VALUES:	Value entered is compared against stub file. MM: 01-12, DD: 01-31
BUSINESS RULES:	Enter other procedure code and date.

- Do not enter punctuation. For claims that are OCR, punctuation is filtered.
- If a 6 digit date is received within the date field, once you press the enter key the century will be automatically populated.
- If Other Procedure Code is invalid, edit alert of: OTHER PROCEDURE CODE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Other Procedure Code is present and Principal Procedure Date is missing, edit alert of: OTHER PROC DATE MISSING/INVALID. PROCEDURE DATE REQUIRED IF CODE PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.
- If Other Procedure Date is present and Other Procedure Code is missing, edit alert of: OTHER PROC DATE MISSING/INVALID. PROCEDURE CODE REQUIRED IF DATE PRESENT is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data was not captured correctly, enter value from image.

- If Other Procedure Date is invalid, edit alert of: OTHER PROC DATE MISSING/INVALID is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Other Procedure Date is after scan date, edit alert of: OTHER PROCEDURE DATE IS AFTER SCAN DATE. PLEASE REKEY is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Other Procedure Date is a greater than current date, edit alert of: OTHER PROC DATE MISSING/INVALID. DATE IS IN FUTURE is received at the bottom of the screen. Verify information on image/data form; if data was captured correctly shift/enter to accept data. If data is not correct, enter value from image.
- If Alpha value is received in the Other Procedure Date field, you will receive a Dialog Box stating “Numeric Field Type”. Click OK. Alpha character is not allowed in this field; enter all values present except alpha characters. You may receive edit alert of: PLEASE ENTER A VALID DATE at the bottom of the screen, shift/enter to accept data.

SCREEN FIELD NAME:	**Attending Phys. ID**
FORM FIELD NUMBER:	82
FIELD DESCRIPTION:	The number for the physician who has primary responsibility for the recipient's medical treatment. I.E.: UPIN (unique identification number).
FORMAT:	10 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter attending physician provider id.

- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).

SCREEN FIELD NAME:	**Other Phys. ID (2 occurrences of this field)**
FORM FIELD NUMBER:	83 a-b
FIELD DESCRIPTION:	The number for the physician who has other responsibilities for the recipient's medical treatment. I.E.: UPIN (unique identification number).
FORMAT:	10 Byte Alphanumeric
VALUES:	Value entered is compared against stub file.
BUSINESS RULES:	Enter other physician provider id.

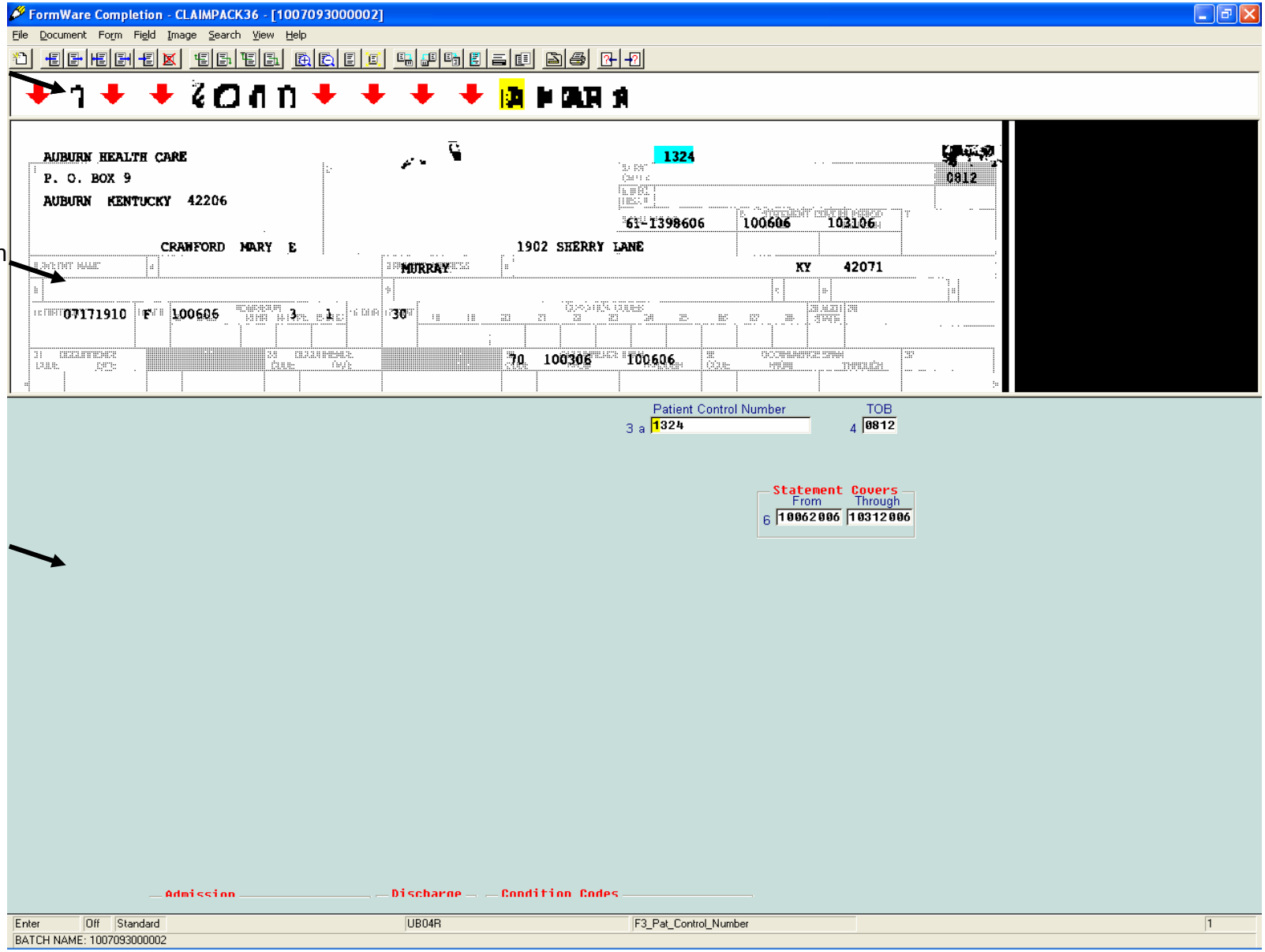
- OCR may or may not capture this field if it is handwritten. If the claim is OCR, this field will need to be checked for a handwritten value on every claim (it will not automatically stop in this field).

27.7.7 KY MMIS UB04 Completion Screens

Reject Ticker

Middle Section

Bottom Section



FormWare Completion - CLAIMPACK36 - [1007093000002]

File Document Form Field Image Search View Help

AUBURN KENTUCKY 42206

61-1398606 100606 103106

CRAWFORD MARY E 1902 SHERRY LANE

MURRAY KY 42071

07171910 F 100606 3 1 30

70 100306 100606

A2 714 00 30 2600 82 6 00

Pat. DOB
10 07172010

Admission
Date HR Type Src
12 10062006 13 14 3 15 1

Discharge
Stat
17 30

Condition Codes
18 19 20 21 22 23 24 25 26 27 28

Occurrence
Cd Date Cd Date Cd Date Cd Date
31 32 33 34

Occurrence Span
Cd From Through Cd From Through
35 70 10032006 10062006 36

Value Codes
Cd Amount Cd Amount Cd Amount
39 a A2 1000 40 a 41 a

Enter Off Standard UB04R F10_Pat_DOB 1

BATCH NAME: 1007093000002

FormWare Completion - CLAIMPACK36 - [1007093000002]

File Document Form Field Image Search View Help

0022		RHC07	101906	12	
0120	ROOM-BOARD/SEMI	140.00		26	3640 00
0250	PHARMACY			1	2364 46
0270	MED-SUR SUPPLIES			1	6 90
0300	LAB			5	537 00
0310	X-RAY			1	271 44
0420	PHYSICAL THERAPY			19	534 60
0430	OCCUPATIONAL THERAPY			30	811 80
0400	SPEECH THERAPY			20	541 80

	Rev.Cd.	Description	HCPCS/Rate	Service Date	Service Units	Line Charges	Non-Covered Charges
1	0022		RNX11	10102006	14		
2	0022		RHC17	10192006	12		
3	0120				26	0000364000	
4	0250				1	0000236446	
5	0270				1	0000000690	
6	0300				5	0000053700	
7	0310				1	0000027144	
8	0420				19	0000053460	
9	0430				30	0000081180	
10	0400				20	0000054180	
11							

	Rev.Cd.	Description	HCPCS/Rate	Service Date	Service Units	Line Charges	Non-Covered Charges
12							
13							
14							
15							
16							

Enter | Off | Standard | UB04R | F42_Revenue_Code_7 | 1

BATCH NAME: 1007093000002

FormWare Completion - CLAIMPACK36 - [1007093000002]

File Document Form Field Image Search View Help

1 1 032607 8708 00

PAGE OF CREATION DATE 1164403564

MEDICARE 185049 Y X 7837 74

KY MEDICAID 12504296 Y Y 714 00

CRAWFORD MARY E 01 402099658A

	Rev.Cd.	Description	HCPCS/Rate	Service Date	Service Units	Line Charges	Non-Covered Charges	
42	12	43	44	45	46	47	48	12
	13							13
	14							14
	15							15
	16							16
	17							17
	18							18
	19							19
	20							20
	21							21
	22							22

Rev.Cd. 23 001

Line Charges Totals >>> 0000001164 23

Provider NPI 56

Payer D

Health Plan Id 01000009

Prior Payments 0000783774

Other Provider Id

Enter Off Standard UB04R F42_Revenue_Code_Totals 1

BATCH NAME: 1007093000002

FormWare Completion - CLAIMPACK36 - [1007093000002]

File Document Form Field Image Search View Help

1 1 032607 8708 00

PAGE ___ OF ___ CREATION DATE PAID 1164403564

MEDICARE 185049 Y Y 7837 74

KY MEDICAID 12504296 Y Y 714 00

CRAWFORD MARY E 01 402099658A

CRAWFORD MARY E 01 4020996585

Rev.Cd. 23 001

Totals >>> Line Charges 0000001164 23

Provider NPI 56

Payer 50 A D

Health Plan Id 51 01000009

Prior Payments 54 0000783774

Other Provider Id 57 A

B B

C C

Insured's Last Name First Name 58 A MITCH CL

Insured's Unique Id 60 4073341515

Insured's Last Name First Name B

Enter Off Standard UB04R F51_Health_Plan_ID_2 1

BATCH NAME: 1007093000002

FormWare Completion - CLAIMPACK36 - [1007093000002]

File Document Form Field Image Search View Help

42731 4019 36250 4149 71590 82100 436 72887 43882

42731

1902885759 1G H12882

SINGH KAMAL

Treatment Authorization Code

63 A

B

C

Principal Diagnosis

67

Other Diagnosis

Code A Code B Code C Code D Code E Code F Code G Code H

Other Diagnosis

Code I Code J Code K Code L Code M Code N Code O Code P Code Q

Admit DX

69 42731

ECI

Code

72 a

Principal Procedure

Code Date

74

Other Procedure

Code Date a Code Date b

Attending Physician

NPI

76

IdQ Other Id

1G H12882

Enter Off Standard UB04R F67_Other_Diag_Code_I 1

BATCH NAME: 1007093000002