



1.0 Post Request File to the Broker

**Prepared by the
Information Technology Support Center**

REQUIREMENTS BASELINE

24 December 2008

1.0 Post Request File to the Broker

Revision History

Date	Version	Description	Author
11/09/06	1.0	Draft Version	Lou Ansaldi (ITSC)
11/15/06	2.0	Updated as of 11/15 Meeting	Jason Hase (ITSC)
2/21/08	3.0	Draft for 2/28/08 Requirements Meeting	Lou Ansaldi (ITSC)
3/10/08	4.0	<p>1. Updated section numbering. First number of each section is now Use Case number</p> <p>2. Rework of Basic Flow. Major changes include steps for handling of state generated GUID, and the checks for “participant”, “active” of both State and Destination Employer</p> <p>3. Updated alternate and exception flows to describe handling when each check performed in Basic flow fails. Updated Acknowledgement process.</p> <p>4. Commenced list of business rules used in this Use Case</p>	Lou Ansaldi (ITSC), Jason Holzbach
3/18/08	5.0	Updated after Consortium review	Lou Ansaldi (ITSC), Jason Holzbach
3/20/08	6.0	Updated after Consortium review	Lou Ansaldi (ITSC), Jason Holzbach
3/31/08	7.0	Updated after internal review	Lou Ansaldi (ITSC), Jason Holzbach
3/31/08	10	<p>Updated to show Broker requirements that affect endpoint requirements</p> <p>Updated to Version 10 for 4/3/08 Requirements Review</p>	Lou Ansaldi (ITSC), Jason Holzbach
4/7/08	11	<p>Updated after consortium review</p> <p>Changed 1.5.8 to 1.5.10</p> <p>Changed 1.5.9 to 1.5.11</p> <p>Moved Alternate and Exception Flows to the correct place</p>	Lou Ansaldi (ITSC), Jason Holzbach
4/9/08	13	<p>No Changes</p> <p>Updated version number to Version 13</p>	Lou Ansaldi (ITSC), Jason Holzbach

Version 18 – Requirements Baseline – 12-24-2008

		for Final Review Draft	
4/14/08	14	Updated after Consortium Review Made Post Conditions Consistent with the text in the alternate/exception flows Added text to 1.5.12 to make consistent with Use Case 2 Added deletion and logging to 1.7.5 REQUIREMENTS BASELINE ISSUE	Lou Ansaldi (ITSC), Jason Holzbach
5/6/08	15	SIGNED OFF VERSION	Lou Ansaldi (ITSC), Jason Holzbach
7/3/08	16	No Changes Updated version number to Version 16 for updated Requirements Baseline reflecting approved changes from CCB #1 document.	Lou Ansaldi (ITSC), Jason Holzbach
7/24/08	17	No Changes Updated version number to Version 17 for updated Requirements Baseline reflecting approved changes from CCB #2 document.	Lou Ansaldi (ITSC), Jason Holzbach
12/24/08	18	No Changes Updated version number to Version 18 for updated Requirements Baseline reflecting approved changes from CCB #3 document.	Lou Ansaldi (ITSC), Jason Holzbach

Table of Contents

1.1 Brief Description.....	5
1.2 Actors.....	5
1.3 Pre-Conditions	5
1.4 Post-Conditions.....	5
1.5 Basic Flow	6
1.6 Alternate Flows.....	7
1.7 Exception Flows.....	7
1.8 Business Rules	9
1.9 Correspondence.....	10
1.10 Screen Layouts.....	10
1.11 Other Notes	10
1.12 Unresolved Notes.....	10
1.13 Glossary	10

USE CASE NAME: 1.0 Post Request File to Broker

1.1 Brief Description

This Use Case describes the steps taken by the State Connector in transmitting Request files to the Broker.

1.2 Actors

1.2.1 State Connector

1.2.2 Broker

1.3 Pre-Conditions

1.3.1 The State has built the XML Request files.

1.4 Post-Conditions

1.4.1 The State Connector has successfully transmitted an XML Request file to the Broker

1.4.2 The following alternate flows and exceptions have been handled:

- The Broker determines that the posting State is not active
- The Broker determines that the destination Employer/TPA is not active
- The Broker fails to authenticate the posting State
- The Broker determines that the destination Employer/TPA is not a participant
- The Broker cannot decrypt the file
- The State cannot connect to the Broker when posting a Request file
- The Broker determines that the Request XML file is malformed
- The Broker parses the XML Request file and there is at least one record failure
- The Broker is unable to transmit Acknowledgement File for Request File from State in a configurable amount of time
- The Broker transmits the Acknowledgement File for Request File from State SOAP message but the State does not receive it

1.5 Basic Flow

- 1.5.1 The State Connector builds, encrypts with the public/private key, the SOAP Post message to transmit a request file to the Broker (Affects Endpoint(s) Requirements)
- 1.5.2 The State Connector transmits the SOAP Post message with file payload to the Broker (Affects Endpoint(s) Requirements)
- 1.5.3 The Broker authenticates the SOAP message.
- 1.5.4 The Broker receives and dates the SOAP Post message with file payload from the State Connection.
- 1.5.5 The Broker decrypts the SOAP message.
- 1.5.6 The Broker inspects the State generated Request file GUID (StateRequestFileGUID) and records if duplicate
- ~~1.5.7 The Broker ensures that the posting State is a participant by inspecting the SOAP header~~
- 1.5.8 The Broker ensures that the destination Employer/TPA is a participant by inspecting the SOAP header
- 1.5.9 The Broker determines that the posting State is Active
- 1.5.10 The Broker determines that the destination Employer/TPA is Active
- 1.5.11 The Broker validates that SOAP message and XML file are well formed.
- 1.5.12 The Broker parses and validates Request file against its XSD. (see Business Rules for the Request XSD)
- 1.5.13 The Broker assigns a unique BrokerRecordTransactionNumber to each valid record in the XML Request file.
- 1.5.14 The Broker inspects the State generated record-level GUID (StateRequestRecordGUID) and records if duplicate
- 1.5.15 The Broker has processed all records in the Request file.
- 1.5.16 The Broker transmits an acknowledgement to the sending State Connector of receipt of SOAP message with the Request Payload (Affects Endpoint(s) Requirements)
- 1.5.17 The Broker dates each valid record.
- 1.5.18 The valid records are ready for the Employer/TPA to Pull or receive through a Push.
- 1.5.19 Use Case ends

1.6 Alternate Flows

- 1.6.1 The Broker determines that the posting State is not active
 - 1.6.1.1 The Broker logs the event
 - 1.6.1.2 The Broker notifies the Broker Administrator
 - 1.6.1.3 Return to 1.5.10

- 1.6.2 The Broker determines that the destination Employer/TPA is not active (Post or Push mode for the Employer/TPA)
 - 1.6.2.1 The Broker logs the event
 - 1.6.2.2 The Broker notifies the Broker Administrator
 - 1.6.2.3 Return to 1.5.11

1.7 Exception Flows

- 1.7.1 The Broker fails to authenticate the posting State
 - 1.7.1.1 The Broker deletes the file
 - 1.7.1.2 The Broker logs the event
 - 1.7.1.3 The Broker notifies the Broker Administrator
 - 1.7.1.4 Use Case ends

- 1.7.2 The Broker determines that the destination Employer/TPA is not a participant
 - 1.7.2.1 The Broker deletes the file
 - 1.7.2.2 The Broker logs the event
 - 1.7.2.3 The Broker notifies the Broker Administrator
 - 1.7.2.4 The Broker sends an Acknowledgement to the State with an appropriate error code (Affects Endpoint(s) Requirements)
 - 1.7.2.5 Use Case ends

- 1.7.3 The Broker cannot decrypt the file
 - 1.7.3.1 The Broker deletes the file

- 1.7.3.2 The Broker logs the event
- 1.7.3.3 The Broker notifies the Broker Administrator
- 1.7.3.4 The Broker sends an Acknowledgement to the State with an appropriate error code (Affects Endpoint(s) Requirements)
- 1.7.3.5 Use Case ends
- 1.7.4 The State cannot connect to the Broker when posting a Request file (Broker is down) (Affects Endpoint(s) Requirements)
 - 1.7.4.1 The State receives an HTTPS error.
 - 1.7.4.2 The State Connector logs to their own system and notifies Broker Administrator via the Help Desk (Affects Endpoint(s) Requirements) [Consortium Process]
 - 1.7.4.3 Use Case Ends
- 1.7.5 The Broker determines that the Request XML file is malformed
 - 1.7.5.1 The Broker rejects the XML file
 - 1.7.5.2 The Broker logs the event.
 - 1.7.5.3 The Broker deletes the file
 - 1.7.5.4 The Broker transmits (response) to the Posting State a SOAP message with a message code indicating XML file rejection (Affects Endpoint(s) Requirements). See Rules and Requirements #86.
 - 1.7.5.5 Use Case Ends
- 1.7.6 The Broker parses the XML Request file and there is at least one record failure (See Request XSD)
 - 1.7.6.1 The Broker builds an XML file of record failure(s), including failed attachments
 - 1.7.6.2 The Broker transmits the acknowledgement to the Posting State, which includes the failed records and the Message Code indicating “File Success with Rejected Records” (Affects Endpoint(s) Requirements)
 - 1.7.6.3 Use Case Ends

- 1.7.7 The Broker is unable to transmit Acknowledgement File for Request File from State in a configurable amount of time (initial setting = 15 minutes)
 - 1.7.7.1 The Broker deletes the Request file (The State will send the same file again) (Affects Endpoint(s) Requirements)
 - 1.7.7.2 The Broker logs the event
 - 1.7.7.3 The Broker generates a report for the Administrator and Posting State
 - 1.7.7.4 Use Case Ends

- 1.7.8 The Broker transmits the Acknowledgement File for Request File from State SOAP message but the State does not receive it (Affects Endpoint(s) Requirements)
 - 1.7.8.1 The State resends the same file up to two (2) more times (Affects Endpoint(s) Requirements)
 - 1.7.8.2 The State logs each file transmission event (Affects Endpoint(s) Requirements)
 - 1.7.8.3 The State notifies its Administrator (Affects Endpoint(s) Requirements) [Consortium Process]
 - 1.7.8.4 Use Case Ends

1.8 Business Rules

State generates GUID for each record in the request file (StateRequestRecordGUID) (Affects Endpoint(s) Requirements)

The State and Employer/TPA logs duplicate files and records (Affects Endpoint(s) Requirements)

When Broker able to send an Ack, and the State receives it the State will send the next file if there is one (Affects Endpoint(s) Requirements)

When Broker able to send an Ack and the State does not receive it, the State will resend the same file up to 3 times (Affects Endpoint(s) Requirements)

When Broker is unable to send an ack, and therefore the State will not receive it, the State will resend the same file up to 3 times. (Affects Endpoint(s) Requirements)

When Broker is unable to send an ack where it knows that the ack was not sent (e.g. a session termination), the file will be deleted in anticipation of receiving the same file again and to prevent duplicates from being passed on to the Employer/TPA.

Date will be the calendar date plus time to the second (mm/dd/yyyy hh:MM:ss)

The system configurable value for the transaction timeout placed on a Post or Pull transaction is 15 minutes. (Affects Endpoint(s) Requirements)

The maximum system configurable value for the transaction timeout placed on a Post or Pull transaction is 60 minutes. (Affects Endpoint(s) Requirements)

The system configurable value for the transaction timeout applies to the whole system. Each entity does not get it's own timeout value.

1.9 Correspondence

1.10 Screen Layouts

1.11 Other Notes

1.12 Unresolved Notes

1.13 Glossary