FSET IMPLEMENTATION GUIDE

REGISTRATION

Credentials to access the web service will be assigned after registering with the E-Services Help Desk by filling out the contact information form and emailing it to us.

NOTE: If there are issues accessing the web service, please contact us for assistance.

INTERFACE OVERVIEW

The data is exchanged between the external Third Party Transmitters and the Louisiana Federal/State Employment Tax (LAFSET) system via Simple Object Access Protocol (SOAP) messages using the Web Services request-response model transport mechanism over an HTTPS SSL connection. The SOAP data structures are specified in this document.

The web service is provided in both a test and production environment at the following URL's:

- TEST - https://esweb.revenue.louisiana.gov/fsettest/fsetgatewaywebservice.asmx
- PRODUCTION - https://esweb.revenue.louisiana.gov/fset/fsetgatewaywebservice.asmx

The web service is built using the WSDL found on the TIGERS site. If there are issues generating a proxy class from the WSDL, you may need to update the targetNamespace in FSETGatewayService.xsd and FSETGatewayHelper.xsd from http://www.statemef.com/FSETGatewayService to https://www.statemef.com/state_gateway.shtml. All URL's referencing the LDR website (*.revenue.louisiana.gov) should have https. Release 4.4 is the current schema set used.

BASIC SOAP MESSAGE STRUCTURE

The LAFSET system uses Web Services to exchange SOAP-structured data. The following sections describe the basic SOAP message structure for LAFSET messages.

This section describes the logical structure of basic messages with a SOAP Header and SOAP Body blocks within a SOAP message Envelope. A SOAP message contains one SOAP Header and one SOAP Body within one SOAP envelope. The SOAP header contains the Web Services (WS) Addressing (WS-Addressing), WSSecurity, and FSETHeader/CustomFSETHeader elements. The SOAP Body contains the Message elements.

SOAP Header

The SOAP Header specification is to be provided by the state or transmitter application. Only one SOAP Header element structure is allowed in SOAP messages on the A2A channel:

FSET Header

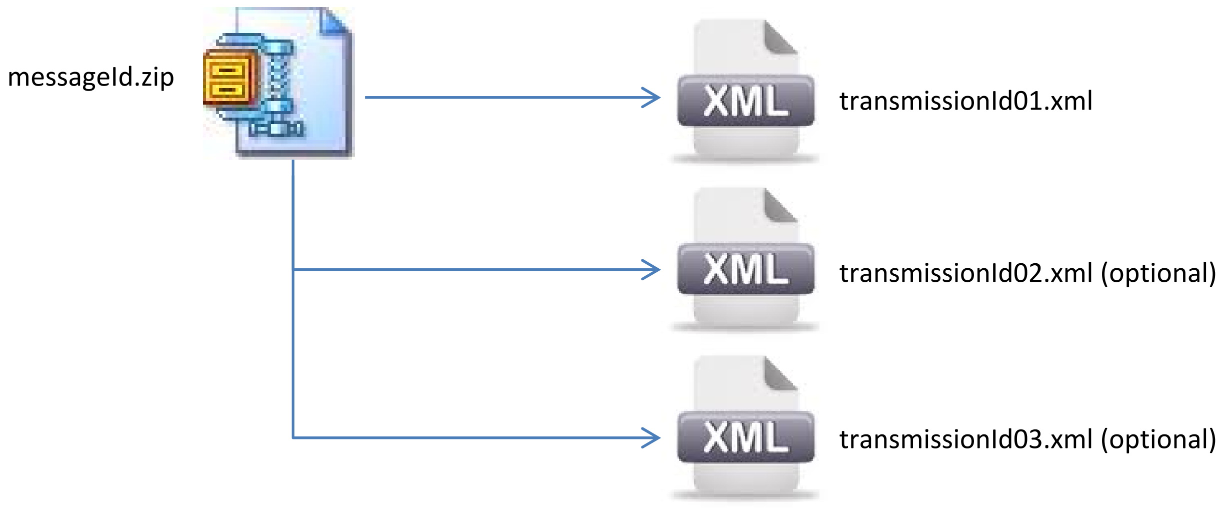
Every SOAP message to and from an A2A Web Service must contain an FSETHeader or CustomFSETHeader in the SOAP message header. The FSETHeader and CustomFSETHeader are described in the FSETHeader Elements section.

SOAP Body

The SOAP Body contains the request message and any attachment provided by the state or transmitter application. For responses, the SOAP body contains the response message and any attachment or a SOAP Fault returned by the service. The attachments must use the current production schema set (v4.4) from the TIGERS site. The business rules and field mapping (from Louisiana return to ReturnState fields) can be found near the end of this document.

Message Attachments Zip File Format

Only a single zip file will be accepted as an attachment sent by a Transmitter and only a single zip file will be returned as an attachment by a service. This zip file attachment is a container zip that contains one or more data items (transmissions or acknowledgements). For Transmitters, each transmission file must always be a valid XML document with an .xml file extension. The container zip file should be compressed. Submissions must follow the format below:


messageId.zip
XML
transmissionId01.xml
XML
transmissionId02.xml (optional)
XML
transmissionId03.xml (optional)

FSETHeader Elements

This section describes the elements in the FSETHeader message. LAFSET has extended the default FSETHeader and created a CustomFSETHeader that adds the username and password needed for authentication/authorization. Every SOAP message request from a Third-Party Transmitter on the FSET channel must contain a CustomFSETHeader message in the SOAP message header with the element tag “CustomFSETHeader”. Every SOAP message response to a Third-Party Transmitter on the FSET channel will contain an FSETHeader message in the SOAP message header with the element tag “FSETHeader”. The elements in the FSETHeader and CustomFSETHeader message are illustrated below. The individual elements are described in the sections that follow. Please note that the elements are the same for both FSETHeaderType and CustomHeaderType, except where noted for Username and Password in CustomHeaderType.

Element FSETHeaderType/MessageID

The MessageID element is mandatory. All request and response messages must have a globally unique MessageID provided by the message source. There are two types of messages to consider: request messages and response messages. The message ID formats are specified as follows.

- Request Message ID – To ensure global uniqueness of a MessageId, the following format is adopted for the request messages sent to the LAFSET system:
    - 6-digit EFIN + 4-digit year + 3-digit day of year + 7 characters

    - Example: 0135792012181000abc1
- Response Message ID – the following format is used by the LAFSET system:
    - LAFSET + 4-digit year + 3-digit day of year + T + 6-digit timestamp (HHMMss)

Element FSETHeaderType/RelatesTo

The RelatesTo element is optional. All response messages from the LAFSET system will populate this field with the value of the MessageId from the request message.

Element FSETHeaderType/Action

The Action element is mandatory. The action element identifies the Web Service endpoints to be used for end-to-end endpoint identification in messages. For all service requests, this element must match the service at the URL invoked. The Action element must be one of the enumerated values (Ping, ChangePassword, SendSubmission, GetNewAcks, GetAcks, GetAck, GetAcksByMsgID, GetNewDX, GetDX).

Element FSETHeaderType/Timestamp

The Timestamp element is mandatory. The Timestamp element identifies the Web Service date and time of message creation.

Element FSETHeaderType/ETIN

The ETIN element required though it is optional in the schema definition. The ETIN (assigned by IRS) element identifies the electronic transmitter identification number of the transmitter who sends the request message.

Element FSETHeaderType/EFIN

The EFIN element required though it is optional in the schema definition. The EFIN (assigned by IRS) element identifies the electronic filer identification number of the transmitter who sends the request message.

Element FSETHeaderType/TestIndicator

The TestIndicator element is mandatory. The TestIndicator element identifies if the request is a production request or test request. The expected values are “P” (to indicate that this is a production request) or “T” (to indicate that this is a test request).

Element FSETHeaderType/NotificationResponse

The optional NotificationResponse element is used to communicate secondary information back to the Third Party or State applications. It is intended for use on response messages only, not on input requests. If there is an error processing the request, information about the error can be found in the NotificationResponse.

Element FSETHeaderType/AppSysID

The AppSysID element is optional. The AppSysID element will be used at a later time with a different security model.

LAFSET Web Services Summary

Each LAFSET service request must contain a SOAP header, which must contain a CustomFSETHeader. The services available to Third Party Transmitters can be found in this section.

All of the services attempt to authenticate the request except for Ping. If authentication fails, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of “AUTHENTICATION_FAILURE”. Additionally, any errors not specified below will cause the NotificationResponse in the FSETHeader of the SOAP response to have a NotificationType of “GENERIC_ERROR - *error*” where *error* is the cause of the failure. Generic errors should be reported to the LAFSET team. It is important that the NotificationResponse of each response is reviewed. It will contain information on any errors generated by a request. If it is empty or null, there were no errors.

Ping

Third Party Transmitters will have authorization to programmatically request a Web Service to query the status of the web service. When the Ping service is called, the TestIndicator is reviewed and the web service will generate a response of whether the respective production or test service is online or offline.

SendSubmissions

Third Party Transmitters will have authorization to programmatically request a Web Service to send submissions for a specific set of returns. The requesting system will provide a SendSubmissionsRequest message in the SOAP body containing the count and a list of submission IDs and a single zip file attachment containing the submissions. If successful, the service will return a SendSubmissionsResponse with a populated SubmissionReceiptList with count and submissionId/Timestamps. Please note that there is a maximum of 1000 submissions allowed per SendSubmissionRequest.

During SendSubmissions, each transmission file is reviewed against multiple rules to ensure that it adheres to the specifications and will be accepted. Please note that the file being accepted does not mean that the returns are accepted. They will be checked against business rules during separate processing in the LAFSET system. The transmission file will be reviewed the following ways:

- *Loaded as a valid XML document with a root element* – If document load fails, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of “SCHEMA_VALIDATION_FAILURE – *transmissionFilename* : *There was an issue loading the xml file or identifying the root element.*” where *transmissionFilename* is the name of the XML transmission file that failed the condition.
- *Root element is Transmission* – The root element of the transmission file must be Transmission. If this condition fails, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of “SCHEMA_VALIDATION_FAILURE – *transmissionFilename* : The root of the transmission must be Transmission (instead of <*root*>).” where *transmissionFilename* is the name of the XML transmission file that failed the condition and *root* is the root element.
- *Incorrect Namespace* – The namespace of the transmission file must be correct. If this condition fails, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of “SCHEMA_VALIDATION_FAILURE – *transmissionFilename* : The transmission is not using the correct namespace <*namespace*>.” where *transmissionFilename* is the name of the XML transmission file that failed the condition and *namespace* is the correct namespace.
- *Validated against the FSET schema* - If schema validation fails, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of “SCHEMA_VALIDATION_FAILURE – *transmissionFilename* : *schemaFailures*” where *transmissionFilename* is the name of the XML transmission file that failed schema validation and *schemaFailures* is a list of reasons that schema validation failed.
- *Duplicate Transmission* - If the transmissionId used in the transmission file has already been processed, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of “TRANSMISSION_ALREADY_RECEIVED – *transmissionId* HAS ALREADY BEEN

RECEIVED ON *date*." where *transmissionId* is the duplicate TransmissionId and *date* is the time that it was previously received.

- *RecordCount matches Submission Count* – If the recordCount attribute in the transmission file doesn't match the number of submissions in the transmission file, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of "RECORD_COUNT_MISMATCH".

GetAck

Third Party Transmitters will have authorization to programmatically request a Web Service to retrieve an acknowledgement for a specified SubmissionId. The requesting system will provide a GetAckRequest message in the SOAP body containing the submissionId. If successful, the service will return a GetAckResponse with an AckListAttachmentMTOM containing the acknowledgement. If an acknowledgement is not available for the submissionId, the GetAckResponse will contain the SubmissionError element with the SubmissionId, ErrorClassification of "REQUEST_ERROR" and ErrorMessage of "AN ACKNOWLEDGMENT COULD NOT BE FOUND."

GetAcks

Third Party Transmitters will have authorization to programmatically request a Web Service to retrieve acknowledgements for a specified list of SubmissionIds. The requesting system will provide a GetAcksRequest message in the SOAP body containing a count and the submissionIds. If successful, the service will return a GetAckResponse with an AckListAttachmentMTOM containing the acknowledgements. If the count supplied does not match the count of submissionId's requested, the NotificationResponse in the FSETHeader of the SOAP response will have a NotificationType of "RECORD_COUNT_MISMATCH". If an acknowledgement is not available for any of the submissionIds, the GetAckResponse will contain the SubmissionError element with the SubmissionId, ErrorClassification of "REQUEST_ERROR" and ErrorMessage of "AN ACKNOWLEDGMENT COULD NOT BE FOUND." in the ErrorList element, along with a count of the number of acknowledgements that were unavailable.

ChangePassword/GetNewAcks/GetAcksByMsgID/GetNewDX/GetDX

Some of these services will be implemented in the future.

Business Rules

	Code	Xpath	Error Type	Severity
	//ReturnState/ReturnHeaderState/TaxPeriodBeginDate			
	R1.1.1	Must have a value	Required	Reject
	V1.1.1	Must be one of 2012-01-01, 2012-04-01, 2012-07-01, 2012-10-01	IsValid	Reject
	V1.1.2	Must be correct value according to TaxPeriodEndDate	IsValid	Reject
	//ReturnState/ReturnHeaderState/TaxPeriodEndDate			
	R1.2.1	Must have a value	Required	Reject
	V1.2.1	Must be one of 2012-03-31, 2012-06-30, 2012-09-30, 2012-12-31	IsValid	Reject
	//ReturnState/ReturnHeaderState/TaxYear			
	V1.3.1	Must have a value of 2012	IsValid	Reject
	//ReturnState/ReturnHeaderState/Filer/StateEIN			
	R.1.4.1	Must have a value	Required	Reject
	V.1.4.1	StateEIN must be a valid 10-digit LDR withholding account number	IsValid	Reject
	//ReturnState/ReturnHeaderState/ReturnQuarter			
	R1.5.1	Must have a value	Required	Reject
	V1.5.1	Must be correct value according to TaxPeriodEndDate	IsValid	Reject
	//ReturnState/ReturnHeaderState/Form			
	V1.6.1	Must have a value of L1	IsValid	Reject
	//ReturnState/ReturnHeaderState/FilingAction/Action			
	V1.7.1	Must have a value from: Original, Amended	IsValid	Reject
	//ReturnState/ReturnHeaderState/AmendedReturnIndicator			
	V1.8.1	Should have a value of X if FilingAction is Amended	IsValid	Warning
	//ReturnState/ReturnDataState/StateWH/SubmissionId			
	V2.0.1	Must be a unique SubmissionId	IsValid	Reject
	//ReturnState/ReturnDataState/StateWH/Month1Liability			
	V2.1.1	Must be a non-negative value	IsValid	Reject
	//ReturnState/ReturnDataState/StateWH/Month2Liability			
	V2.2.1	Must be a non-negative value	IsValid	Reject
	//ReturnState/ReturnDataState/StateWH/Month3Liability			
	V2.3.1	Must be a non-negative value	IsValid	Reject
	//ReturnState/ReturnDataState/StateWH/TotalQuarterLiability			
	V2.4.1	Must be a non-negative value	IsValid	Reject
	C2.4.1	Should equal the sum of Month1Liability, Month2Liability, and Month3Liability	Calculation	Warning
	//ReturnState/ReturnDataState/StateWH/WHAmountRemitted			
	V2.5.1	Must be a non-negative value	IsValid	Reject
	//ReturnState/ReturnDataState/StateWH/WHBalanceDue			
	V2.6.1	Must be a non-negative value	IsValid	Reject
	C2.6.1	Should equal the larger of 0 and WHAmountRemitted from TotalQuarterLiability	Calculation	Warning
	//ReturnState/ReturnDataState/StateWH/WHOverpayment/AmountOfOverpayment			
	V2.7.1	Must be a non-negative value	IsValid	Reject
	C2.7.1	Should equal the larger of 0 and TotalQuarterLiability from WHAmountRemitted	Calculation	Warning
	//ReturnState/ReturnDataState/StateWH/WHOverpayment/RefundElect			
	V2.8.1	Should have a value of X ifAmountOfOverpayment is greater than 0	IsValid	Warning
	//ReturnState/FinancialTransaction/StatePayment/RequestedPaymentDate			
	R3.1.1	If no value is provided, the value of Timestamp will be used	Required	Informational

L1 Field Mappings

The grid below maps the data fields from the return to their corresponding XPath.

Line	XPath
1	//ReturnState/ReturnDataState/StateWH/Month1Liability
2	//ReturnState/ReturnDataState/StateWH/Month2Liability
3	//ReturnState/ReturnDataState/StateWH/Month3Liability
4	//ReturnState/ReturnDataState/StateWH/TotalQuarterLiability
5	//ReturnState/ReturnDataState/StateWH/WHAmountRemitted
6	//ReturnState/ReturnDataState/StateWH/WHBalanceDue
7	//ReturnState/ReturnDataState/StateWH/WHOverpayment/AmountOfOverpayment
*	//ReturnState/ReturnDataState/StateWH/WHOverpayment/RefundElect

*- If AmountOfOverpayment is greater than 0, check RefundElect