

eFax Developer™

Universal User Guide for Outbound Processing

Revision Date: 03/01/2014

Contents

WELCOME	4
CONTENTS AT A GLANCE	4
PRODUCT INTRODUCTION	5
eFAX DEVELOPER™ OUTBOUND OVERVIEW	5
COMPONENTS	5
Security	5
eFax Developer™ Online Interface	5
Client Profile	5
XSD (XML Schema Definition)	5
Account Identifier	5
XML (Extensible Markup Language)	6
Base64 Encoding	6
URL Encoding	6
Desired Response Type (Optional)	6
PRODUCT INTEGRATION.....	7
INTEGRATION REQUIREMENTS	7
INTEGRATION PLANNING	7
PROGRAMMING.....	8
CLIENT-SIDE PROGRAMMING OVERVIEW	8
DYNAMIC FAX HEADERS	9
Overview	9
@Variables	9
Font Control	10
XML ELEMENT MAPPING	11
Outbound Request XML	11
Outbound Status Request XML	16
Outbound Response XML	17
Outbound Status Response XML	18
Outbound Disposition XML	21
HTML RESPONSE.....	22
OUTBOUND RESPONSE HTML	22
OUTBOUND STATUS RESPONSE HTML	23
XML RESPONSE.....	25
OUTBOUND RESPONSE XML	25
OUTBOUND STATUS RESPONSE XML	26
USER LEVEL ERROR MESSAGES.....	27
ERROR MESSAGES RETURNED BY eFAX DEVELOPER™ AUTHENTICATION	27
ERROR MESSAGES RETURNED BY XSD VALIDATION	31
SAMPLES	33
OUTBOUND REQUEST XML EXAMPLE	33
OUTBOUND STATUS REQUEST XML EXAMPLE	34
OUTBOUND DISPOSITION XML EXAMPLE	34
XSD (SCHEMATA) EXAMPLE	35

Notice

In all communications concerning this documentation, please refer to the revision date displayed on the front cover of this user's guide.

Copyright

The use, disclosure, reproduction, modification, transfer, or transmittal of this work for any purpose in any form or by any means without the written permission of j2 Global, Inc. is strictly prohibited.

Welcome

Welcome to *eFax Developer's Universal User Guide for Outbound Processing*. This guide was designed to assist you in integrating *eFax Developer™* into your current outbound processing.

This documentation was designed to assist clients who are required to build their own client-side API.

Before continuing, please consider using an *eFax Developer™* SDK if applicable. Our Software Developer's Kit is designed to ease a client's overall development timeline while giving client applications full functionality as described within this document.

Users who wish to "cut to the chase" can skip directly to the Programming section as a quick-start.

Contents at a glance

- **Product Introduction** will provide an introduction to *eFax Developer™ Outbound*.
- **Product Integration** will provide information for integrating with *eFax Developer™*.
- **Programming** will provide essential information for the client-side programmer.
- **HTML Response** will provide screen print examples of various HTML responses.
- **XML Response** will provide screen print examples of various XML responses.
- **User Level Error Messages** will cover common error messages and solutions.
- **Samples** will provide sample XML and XSD files.

Product Introduction

eFax Developer™ Outbound Overview

eFax Developer™ Outbound is a service that software applications can use to send multiple electronic documents to a specified fax number.

Based on XML technology, developers POST outbound requests to *eFax Developer™* through a URL passing an Account Identifier argument, an XML-formatted data argument, and; if desired; a response type as part of the POST request.

eFax Developer™ will authenticate the request, then immediately notify the client process whether or not the outbound request was successfully received.

Once validated, a fax transmission will be created and launched. Clients will be notified of a transmission's final disposition. Clients can control this notification through elements passed in the outbound request. A client can suppress notification, or generate notification upon success, failure, or both.

Components

Security

eFax Developer™ is designed with data security in mind. Requests are made using secure (HTTPS) protocol ensuring sender and receiver are the only parties able to decode the request. Additionally, each request will be authenticated against the requesting client's profile and XSD (schemata).

***eFax Developer™* Online Interface**

Clients can change various outbound settings through *eFax Developer™* online by clicking the "Settings" icon located on any page of the interface. Go to <https://secure.efaxdeveloper.com> to log into the *eFax Developer™* online interface.

Client Profile

eFax Developer™ client profile will consist of a user name, password and a unique Account Identifier. If desired, clients may choose to have all submitting IP Address' validated as an additional layer of security. Clients can access their profile via the *eFax Developer™* online interface.

XSD (XML Schema Definition)

To ensure data integrity, *eFax Developer™* will validate all requests against the client's XSD file. XSD files are client specific and generated at account setup time.

Account Identifier

Each client will be provided a unique Account Identifier. The Account Identifier is passed as part of the request. *eFax Developer™* will authenticate the Account Identifier against the requesting client's profile.

XML (Extensible Markup Language)

The XML-formatted data argument will contain client access and fax transmission data specific to the request. This argument will contain the Base64 encoded document to be transmitted.

*eFax Developer*TM supports the XML 1.0 specification as recommended by the World Wide Web consortium (<http://www.w3.org/TR/REC-xml>). All XML messages must comply with this specification.

Base64 Encoding

The document passed in the XML-formatted argument must be Base64 encoded. Base64 encoding will convert the document binary data into an ASCII text string.

An in depth RFC on *Base64 Content-Transfer-Encoding* can be found at the following addresses

- <http://www.freesoft.org/CIE/RFC/1521/7.htm>
- http://www.w3.org/Protocols/rfc1341/5_Content-Transfer-Encoding.html

URL Encoding

The XML-formatted argument must be URL-encoded prior to posting to *eFax Developer*TM. Based on World Wide Web consortium specifications, an argument's name and value should be URL-encoded with each name and value separated by an equal sign ("=") and each name/value pairing separated by an ampersand ("&"). Neither "=" nor "&" should be URL-encoded when used as a separator.

An in depth RFC on *URL Character Encoding Issues* can be found at the following addresses:

- <http://www.freesoft.org/CIE/RFC/1738/4.htm>
- http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.1

Desired Response Type (Optional)

Clients may choose to receive XML-formatted responses by passing a response type argument to *eFax Developer*TM otherwise, HTML-formatted responses are the chosen default.

Final Disposition notification is not affected by the response type argument. Final Disposition notification will be an XML-formatted response as described in this document.

Product Integration

Integration Requirements

To integrate *eFax Developer*[™] with your application, you must develop custom code that adapts your business logic and data to *eFax Developer*[™]. You can use any programming language that supports HTTP communication across internet socket and secure connections, such as Java, Visual Basic, or C++.

- You will need to know how to program URL or socket connections. The method you choose will depend on your application platform and programming language. For example, if you program in Java on any platform, you could use the URLConnection Class. Alternatively, Microsoft supplies several components for use with Visual Basic and ASP (VBScript).
- If you desire secure socket implementation, you will need to obtain, or write a secure socket implementation that supports the SSL standard for secure communications.
- You will need to know how to encode and decode XML documents. There are a number of tools, tutorials and editors freely available to assist you as needed.
- You will need to know how to encode binary data using Base64 encoding. There are a number of tools freely available to assist you with Base64 encoding.
- You will need to know how to URL-encode arguments passed in the POST request.
- You will need to parse the formatted response to determine a request's success or failure.

Integration Planning

To successfully integrate with *eFax Developer*[™], it is important to have the appropriate tools as well as a project plan prior to attempting integration. The following is provided to assist you in the integration planning process.

- Contact *eFax Developer*[™] Support to establish your account.
- Carefully review this documentation.
- If desired, obtain the proper software for SSL communications.
- Obtain software for XML encoding/decoding.
- Obtain software for Base64 encoding.
- Determine how your application will connect with the internet.
- Develop a plan for error handling.
- Develop your applications.
- Log into the *eFax Developer*[™] online interface to set your outbound settings as desired.
- Test your application's integration with *eFax Developer*[™].
- Make your application live.

Programming

Client-Side Programming Overview

The following is provided as a high-level overview of client-side **outbound request** processing.

- Open a TCP/IP socket or URL connection using the URL address for *eFax Developer*TM. The secure *eFax Developer*TM URL is https://secure.efaxdeveloper.com/EFax_WebFax.serv.
- Set the request header section to specify the HTTP POST method, set the Content-type to be “application/x-www-form-urlencoded.”
- Create an XML-formatted message meeting all requirements as explained in this document.
- URL-encode the XML-formatted message.
- Send your request to *eFax Developer*TM. To send a request, an application writes a stream of data to *eFax Developer*TM via a TCP/IP socket or URL connection. The data stream must be in the form of an HTTP POST request message. The request will consist of two arguments, an Account Identifier (“id”); provided at setup time; and your XML-formatted data (“xml”). Optionally, you may send a third argument to indicate the desired response type (“respond”) of “HTML” (default) or “XML.” For example, the entire URL and query string should be formatted as follows...

https://secure.efaxdeveloper.com/EFax_WebFax.serv?id=999999999&xml=<the URL-encoded XML value goes here>

- Parse the formatted response returned by *eFax Developer*TM for success or failure.

The following is provided as a high-level overview of client-side **outbound status request** processing.

- Follow client-side **outbound request** processing guide-lines displayed above.
- Parse the formatted response returned by *eFax Developer*TM as desired.

The following is provided as a high-level overview of client-side **outbound disposition** processing, if used.

- If indicated, *eFax Developer*TM can send an XML-formatted disposition via HTTP POST to a URL specified in the “DispositionURL” element of the request XML. *eFax Developer*TM will make an initial attempt to deliver a disposition. If delivery fails, *eFax Developer*TM will retry the disposition until a total of 4 attempts have been made. With each attempt, an alert email will be sent to the client’s primary email address until a confirmation (“Post Successful”) is received back from the client-side “disposition response” process, or 4 total attempts have been tried.
- Create an application to accept and decode XML-formatted disposition data from the HTTP POST “xml” argument. Process the disposition data as desired then return an HTML response of “Post Successful” back to the *eFax Developer*TM.
- If desired, *eFax Developer*TM can be set to generate dispositions to one-or-many email addresses instead of via HTTP POST. If the email option is selected, the HTTP POST disposition will not be generated.

Dynamic Fax Headers

Overview

eFax Developer™ allows clients the ability to *override* the application generated fax header. This section is provided as a programmer's reference to this optional feature.

The following line is an example of a dynamic fax header line with a static company name inserted:

"@DATE1 @TIME3 **My Company Name** @ROUTETO{26} @RCVRFAX **Pg**%P/@SPAGES"

That dynamic header line once converted will display as follows:

10/15/03 11:11AM **My Company Name** Recipient Name Company 8581234567 **Pg** 1/2

Dynamic Fax Headers are freeform; the client is free to format the line as desired. The maximum length of this line is 80 characters for a single line. *@Variables* can be inserted in no particular order or removed completely. Static text can be added as desired, so "Pg 1/2" could have just as easily been made to display as "Page 1 of 2" or just "Page 1." *In the above example, bolded text represents static text.*

@Variables

The following describes variables available for use in the dynamic fax header string:

- @DATEx - from our server system date (Pacific Time)
 - @DATE0 yyyyymmdd (example: 20031015)
 - @DATE1 mm/dd/yy (example: 10/15/03)
 - @DATE2 dd/mm/yy (example: 15/10/03)
 - @DATE3 dd/xx/yy (example: 15/OC/03)
 - @DATE4 mm/dd/yyyy (example: 10/15/2003)
 - @DATE5 dd mon yyyy (example: 15 Oct 2003)
 - @DATE6 xxxxx dd, yyyy (example: October 15, 2003)
 - @DATE7 yy mm dd (example: 03 10 15)
 - @DATE8 yy-mm-dd (example: 03-10-15)
 - @DATE9 yymmdd (example: 031015)
- @TIMEx – from our server system time (Pacific Time)
 - @TIME1 hh:mm (example: 17:30)
 - @TIME2 hh:mm:ss (example: 17:30:00)
 - @TIME3 hh:mmxx (example: 05:30PM)
 - @TIME4 hhmm (example: 1730)
- @ROUTETO{n} – “*Recipient Name Recipient Company*” passed within the XML-formatted argument. *The number in brackets indicates the maximum width (characters) of this area within the header.*
- @RCVRFAX – “*Recipient Fax*” number passed within the XML-formatted argument.
- %P – Contains the current page number.
- @SPAGES – Contains the number of pages; including the cover sheet; to be sent.

Font Control

It is possible to control the font size within Dynamic Fax Headers through the use of one or many font control variables. The following describes this feature.

- `%nf` – “n” contains a number (0 through 3) that indicates the font size. “0” is the default value with each subsequent number indicating a slightly smaller font. Font control variables can be placed anywhere within the fax header line. Any text following a font control variable will be affected until another font control variable is encountered.

XML Element Mapping

Outbound Request XML

Element	Required/Cardinality	Type	Length	Description/Values
OutboundRequest/ Access Control	Yes/One	Container	N/A	Access Control Container
OutboundRequest/ AccessControl/ UserName	Yes/One	Alphanumeric	20	User Name
OutboundRequest/ AccessControl/ Password	Yes/One	Alphanumeric	20	User Password
OutboundRequest/ Transmission	Yes/One	Container	N/A	Transmission Container
OutboundRequest/ Transmission/ TransmissionControl	Yes/One	Container	N/A	Transmission Control Container
OutboundRequest/ Transmission/ TransmissionControl/ TransmissionID	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
OutboundRequest/ Transmission/ TransmissionControl/ CustomerID	No/Zero or One	Alphanumeric	50	Client specified customer identifier. The CustomerID is optional. It is not required or validated by <i>eFax Developer™</i> .
OutboundRequest/ Transmission/ TransmissionControl/ NoDuplicates	No/Zero or One	Alphanumeric	7	“ENABLE” “DISABLE” (default) By default, the system will allow duplicate transmission identifiers to be submitted. Enabling this option causes <i>eFax Developer™</i> to verify that the transmission identifier has not already been used by a past transmission. When enabled, <i>eFax Developer™</i> will fail the transmission if a duplicate transmission identifier exists.

Element	Required/Cardinality	Type	Length	Description/Values
OutboundRequest/ Transmission/ TransmissionControl/ Resolution	Yes/One	Alphanumeric	8	<p>“STANDARD” “FINE”</p> <p>Resolution desired for this transmission.</p> <p><u>Transmissions requested with “fine” resolution will incur a price premium.</u></p>
OutboundRequest/ Transmission/ TransmissionControl/ Priority	No/Zero or One	Alphanumeric	6	<p>“NORMAL” (Default) “HIGH”</p> <p>Priority desired for this transmission.</p> <p>By default, transmissions will be sent with normal priority.</p> <p><u>Transmissions requested with “high” priority will incur a price premium.</u></p>
OutboundRequest/ Transmission/ TransmissionControl/ SelfBusy	No/Zero or One	Alphanumeric	7	<p>“ENABLE” (Default) “DISABLE”</p> <p>By default, the system prevents multiple fax channels from simultaneously dialing the same fax number.</p> <p>Disabling this option will allow a single fax number to be dialed simultaneously by multiple fax channels.</p>
OutboundRequest/ Transmission/ TransmissionControl/ FaxHeader	No/Zero or One	Alphanumeric	80	<p>Dynamic Fax Header</p> <p>Client defined fax header to be used during this transmission.</p> <p>If no value is supplied, a default fax header will be used.</p> <p><u>Please review the section on “Dynamic Fax Headers” for more information.</u></p>

Element	Required/Cardinality	Type	Length	Description/Values
OutboundRequest/ Transmission/ TransmissionControl/ TSID	No/One	Alphanumeric	20	Contains the Transmitting Subscriber Identification or TSID. If no value is provided, eFax Developer™ will transmit the account's fax number as the TSID.
OutboundRequest/ Transmission/ DispositionControl	Yes/One	Container	N/A	Disposition Control Container
OutboundRequest/ Transmission/ DispositionControl/ DispositionURL	No/Zero or One	Any URI	100	Contains the destination URL to receive final disposition notifications.
OutboundRequest/ Transmission/ DispositionControl/ DispositionLevel	Yes/One	Alphanumeric	7	"ERROR" "SUCCESS" "BOTH" "NONE" (default) Defines the level at which final disposition notifications should be received. Clients can suppress the notification completely, or generate the notification upon success, failure or both.
OutboundRequest/ Transmission/ DispositionControl/ DispositionMethod	No/Zero or One	Alphanumeric	5	"POST" "EMAIL" "NONE" (default) Defines delivery method for the final disposition.
OutboundRequest/ Transmission/ DispositionControl/ DispositionLanguage	No/Zero or One	Alphanumeric	2	The ISO 639-1 language code for final disposition emails. <div> "en" English (default) "de" German "es" Spanish "fr" French "it" Italian "nl" Dutch "pl" Polish "pt" Portuguese </div>

Element	Required/Cardinality	Type	Length	Description/Values
OutboundRequest/ Transmission/ DispositionControl/ DispositionEmails	No/Zero or One	Container	N/A	Disposition Emails Container will hold one-to-many disposition email containers.
OutboundRequest/ Transmission/ DispositionControl/ DispositionEmails/ DispositionEmail	No/Zero or Many	Container	N/A	Disposition Email Container
OutboundRequest/ Transmission/ DispositionControl/ DispositionEmails/ DispositionEmail/ DispositionRecipient	No/Zero or One	Alphanumeric	ANY	The recipient's <i>friendly</i> name to be displayed as part of the disposition email address.
OutboundRequest/ Transmission/ DispositionControl/ DispositionEmails/ DispositionEmail/ DispositionAddress	Yes/One	Alphanumeric	ANY	The recipient's disposition email address.
OutboundRequest/ Transmission/ Recipients	Yes/One	Container	N/A	Recipients Container
OutboundRequest/ Transmission/Recipients/ Recipient	Yes/One	Container	N/A	Recipient Container
OutboundRequest/ Transmission/Recipients/ Recipient/ RecipientName	No/Zero or One	Alphanumeric	50	The recipient's name when supplied will be merged into the fax header line. <u>Please review the section on "Dynamic Fax Headers" when using that option.</u>
OutboundRequest/ Transmission/Recipients/ Recipient/ RecipientCompany	No/Zero or One	Alphanumeric	50	The recipient's company name when supplied will be merged into the fax header line. <u>Please review the section on "Dynamic Fax Headers" when using that option.</u>

Element	Required/Cardinality	Type	Length	Description/Values
OutboundRequest/ Transmission/Recipients/ Recipient/ RecipientFax	Yes/One	Alphanumeric	25	The recipient's fax number. Designate international fax numbers by using the "011" international dialing prefix or start the fax number with a plus sign ("+").
OutboundRequest/ Transmission/ Files	Yes/One	Container	N/A	Files Container will hold one-to-many File containers.
OutboundRequest/ Transmission/Files/ File	Yes/One to Many	Container	N/A	File Container
OutboundRequest/ Transmission/Files/File/ FileContents	Yes/One	Base64 encoded text	ANY	Base64 encoded file contents.
OutboundRequest/ Transmission/Files/File/ FileType	Yes/One	Alphanumeric	4	Defines the file type being transmitted. <div> <div>doc</div> <div>docx</div> <div>xls</div> <div>xlsx</div> <div>ppt</div> <div>pptx</div> <div>html/htm</div> <div>tif/tiff</div> <div>jpg/jpeg</div> <div>txt</div> <div>pdf</div> <div>rtf</div> <div>snp</div> <div>png</div> <div>gif</div> </div>

Outbound Status Request XML

Element	Required/Cardinality	Type	Length	Description/Values
OutboundStatus/ Access Control	Yes/One	Container	N/A	Access Control Container
OutboundStatus/ AccessControl/ UserName	Yes/One	Alphanumeric	20	User Name
OutboundStatus/ AccessControl/ Password	Yes/One	Alphanumeric	20	User Password
OutboundStatus/ Transmission	Yes/One	Container	N/A	Transmission Container
OutboundStatus/ Transmission/ TransmissionControl	Yes/One	Container	N/A	Transmission Control Container
OutboundStatus/ Transmission/ TransmissionControl/ TransmissionID	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
OutboundStatus/ Transmission/ TransmissionControl/ DOCID	No/Zero or One	Alphanumeric	N/A	<i>eFax Developer™</i> transmission identifier.

Outbound Response XML

Attribute	Required	Type	Length	Description/Values
OutboundResponse/ Transmission	Yes/One	Container	N/A	Transmission Container
OutboundResponse/ Transmission/ TransmissionControl	Yes/One	Container	N/A	Transmission Control Container
OutboundResponse/ Transmission/ TransmissionControl/ TransmissionID	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
OutboundResponse/ Transmission/ TransmissionControl/ DOCID	No/Zero or One	Alphanumeric	8	<i>eFax Developer</i> TM transmission identifier.
OutboundResponse/ Transmission/ Response	Yes/One	Container	N/A	Response Container
OutboundResponse/ Transmission/ Response/ StatusCode	Yes	Alphanumeric	1	“1” Success “2” Failure Requests return status
OutboundResponse/ Transmission/ Response/ StatusDescription	Yes	Alphanumeric	7	“Success” “Failure” Requests return status value
OutboundResponse/ Transmission/ Response/ ErrorLevel	No/Zero or One	Alphanumeric	6	“User” “System” Defines the level of error. A “User” level can be handled by the client, while “System” generated errors will require <i>eFax Developer</i> TM Support intervention.
OutboundResponse/ Transmission/ Response/ ErrorMessage	No/Zero or One	Alphanumeric	ANY	The generated error message.

Outbound Status Response XML

Attribute	Required	Type	Length	Description/Values
OutboundStatusResponse/ Transmission	Yes/One	Container	N/A	Transmission Container
OutboundStatusResponse/ Transmission/ TransmissionControl	Yes/One	Container	N/A	Transmission Control Container
OutboundStatusResponse/ Transmission/ TransmissionControl/ TransmissionID	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
OutboundStatusResponse/ Transmission/ TransmissionControl/ CustomerID	No/Zero or One	Alphanumeric	50	Client specified customer identifier.
OutboundStatusResponse/ Transmission/ Recipients	Yes/One	Container	N/A	Recipients Container
OutboundStatusResponse/ Transmission/Recipients/ Recipient	Yes/One	Container	N/A	Recipient Container
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ DOCID	No/Zero or One	Alphanumeric	8	<i>eFax Developer™</i> transmission identifier.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ Name	No/Zero or One	Alphanumeric	50	The recipient's name.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ Company	No/Zero or One	Alphanumeric	20	The recipient's company name.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ Fax	No/Zero or One	Alphanumeric	25	The recipient's fax number.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ Status	Yes/One	Container	N/A	Status Container
OutboundStatusResponse/ Transmission/Recipients/ Recipient/Status/ Message	Yes/One	Alphanumeric	100	The status message for this fax transmission.

Attribute	Required	Type	Length	Description/Values
OutboundStatusResponse/ Transmission/Recipients/ Recipient/Status/ Classification	No/Zero or One	Alphanumeric	ANY	The status classification. May or may not exist depending on the current status of the transmission.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/Status/ Outcome	No/Zero or One	Alphanumeric	100	The outcome text message. May or may not exist depending on the current status of the transmission.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ LastAttempt	Yes/One	Container	N/A	Last Attempt Container
OutboundStatusResponse/ Transmission/Recipients/ Recipient/LastAttempt/ LastDate	No/Zero or One	Alphanumeric	10	Last attempt date for this fax. May or may not exist depending on the current status of the transmission. Format is mm/dd/yyyy.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/LastAttempt/ LastTime	No/Zero or One	Alphanumeric	8	Last attempt time for this fax. May or may not exist depending on the current status of the transmission. Time zone is PST. Format is hh:mm:ss (24 hour).
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ NextAttempt	Yes/One	Container	N/A	Next Attempt Container
OutboundStatusResponse/ Transmission/Recipients/ Recipient/LastAttempt/ NextDate	No/Zero or One	Alphanumeric	10	Next attempt date for this fax. May or may not exist depending on the current status of the transmission. Format is mm/dd/yyyy.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/NextAttempt/ NextTime	No/Zero or One	Alphanumeric	8	Next attempt time for this fax. May or may not exist depending on the current status of the transmission. Time zone is PST. Format is hh:mm:ss (24 hour).

Attribute	Required	Type	Length	Description/Values
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ Pages	Yes/One	Container	N/A	Pages Container
OutboundStatusResponse/ Transmission/Recipients/ Recipient/Pages/ Scheduled	No/Zero or One	Alphanumeric	ANY	Scheduled number of pages for this fax transmission. May or may not exist depending on the current status of the transmission.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/Pages/ Sent	No/Zero or One	Alphanumeric	ANY	Number of pages successfully sent for this fax transmission. May or may not exist depending on the current status of the transmission.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ BaudRate	No/Zero or One	Alphanumeric	8	Baud Rate used for this fax transmission. May or may not exist depending on the current status of the transmission.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ Duration	No/Zero or One	Alphanumeric	10	Actual duration in minutes for this fax transmission. May or may not exist depending on the current status of the transmission.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ Retries	No/Zero or One	Alphanumeric	3	Actual number of retry attempts for this fax transmission. May or may not exist depending on the current status of the transmission.
OutboundStatusResponse/ Transmission/Recipients/ Recipient/ RemoteCSID	No/Zero or One	Alphanumeric	ANY	The CSID transmitted to. May or may not exist depending on the current status of the transmission.

Outbound Disposition XML

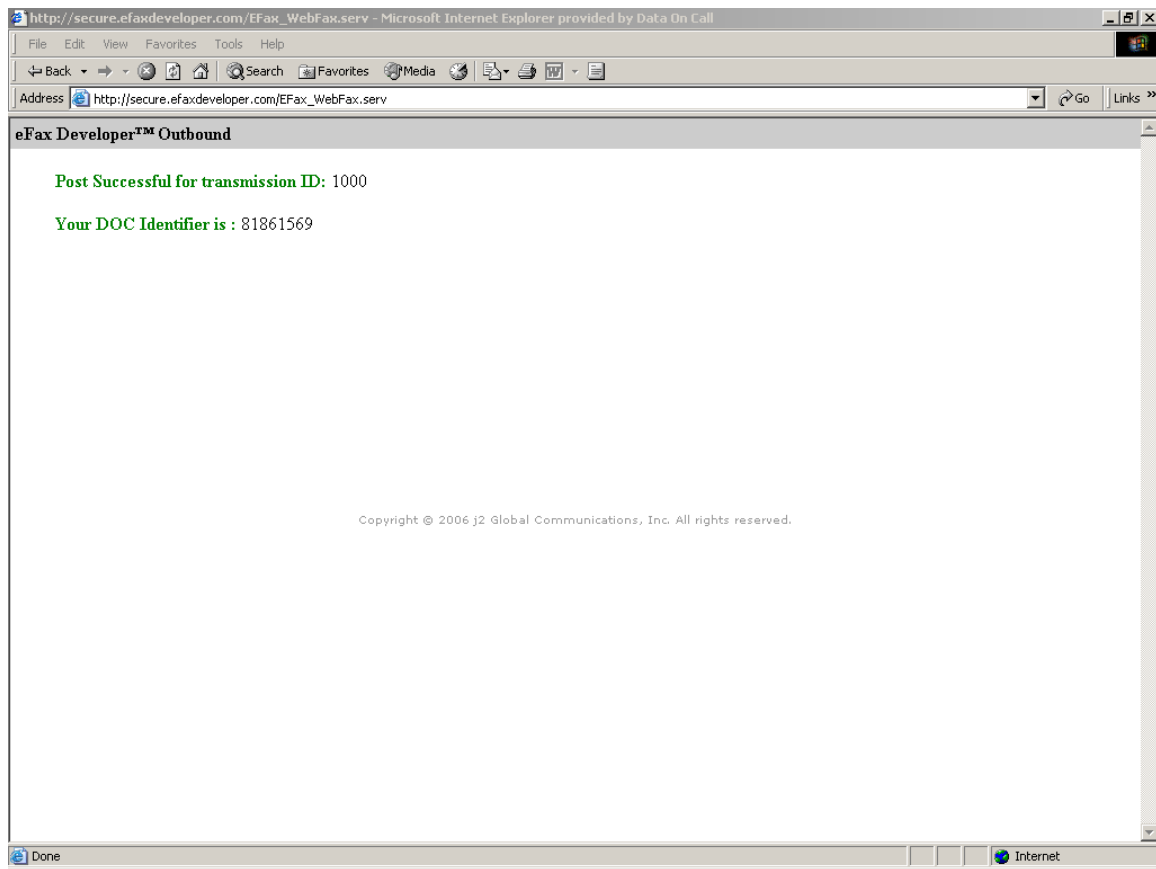
Attribute	Required	Type	Length	Description/Values
OutboundDisposition/ UserName	Yes	Alphanumeric	20	User Name
OutboundDisposition/ Password	Yes	Alphanumeric	20	User Password
OutboundDisposition/ TransmissionID	No	Alphanumeric	15	Unique client specified transmission identifier.
OutboundDisposition/ DOCID	Yes	Alphanumeric	8	<i>eFax Developer</i> TM transmission identifier.
OutboundDisposition/ FaxNumber	Yes	Alphanumeric	25	The recipient's fax number.
OutboundDisposition/ CompletionDate	Yes	Alphanumeric	20	Fax completion date and time. Time zone is PST. Format is yyyy-mm-dd hh:mm:ss (24 hour).
OutboundDisposition/ FaxStatus	Yes	Numeric	5	Numeric field indicating the fax status. "0" indicates a successful transmission while all other values indicate an error code which can be cross-referenced with an <i>eFax Developer</i> TM supplied table.
OutboundDisposition/ RecipientCSID	No	Alphanumeric	20	The station identifier, when supplied by the receiving fax machine upon successful transmission.
OutboundDisposition/ Duration	No	Alphanumeric	5	Transmission time in minutes.
OutboundDisposition/ PagesSent	No	Numeric	3	The number of pages sent.
OutboundDisposition/ NumberOfRetries	No	Numeric	2	Number of times the fax was attempted before success or failure.

HTML Response

Outbound Response HTML

eFax Developer™ will send a response back to the calling application. It is the client's responsibility to parse this response HTML to determine whether or not the request was successfully received. *This response is always generated by eFax Developer™ and should not be confused with the final disposition.*

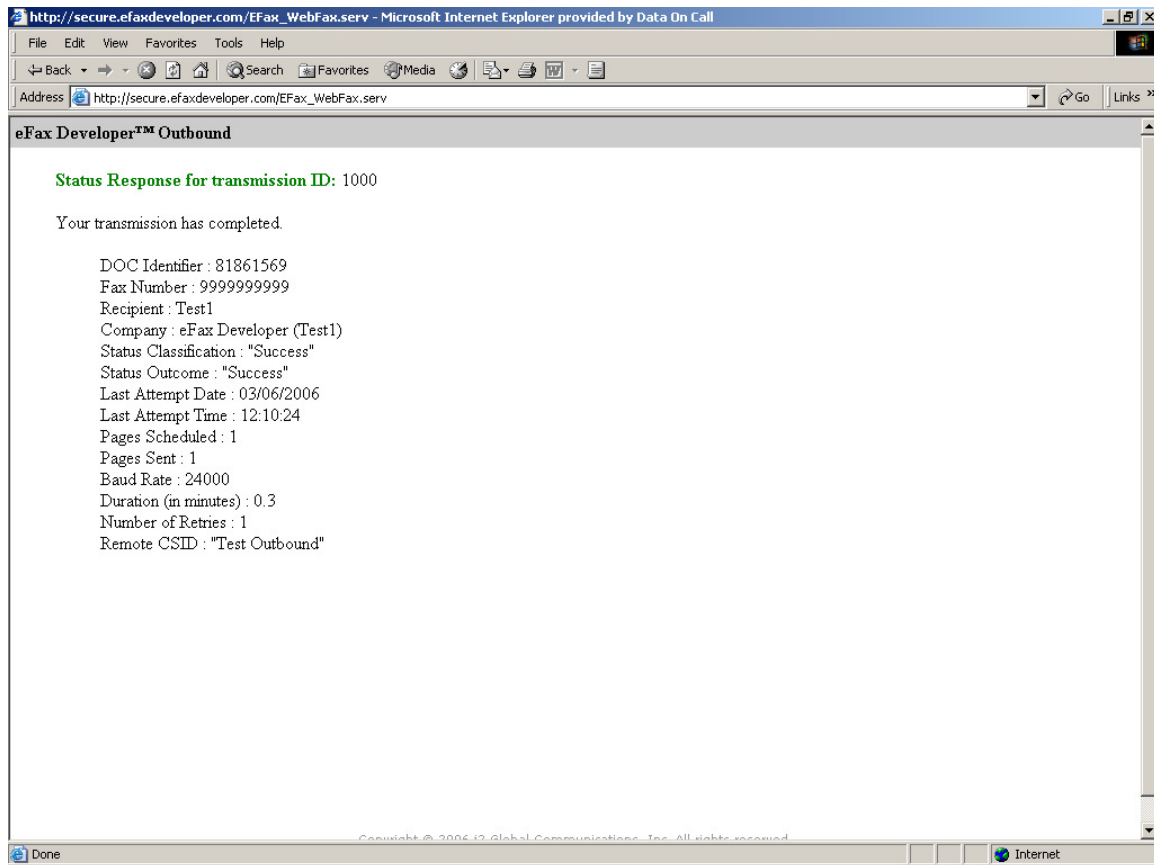
The following HTML will be returned to the client when the outbound request is successfully received.



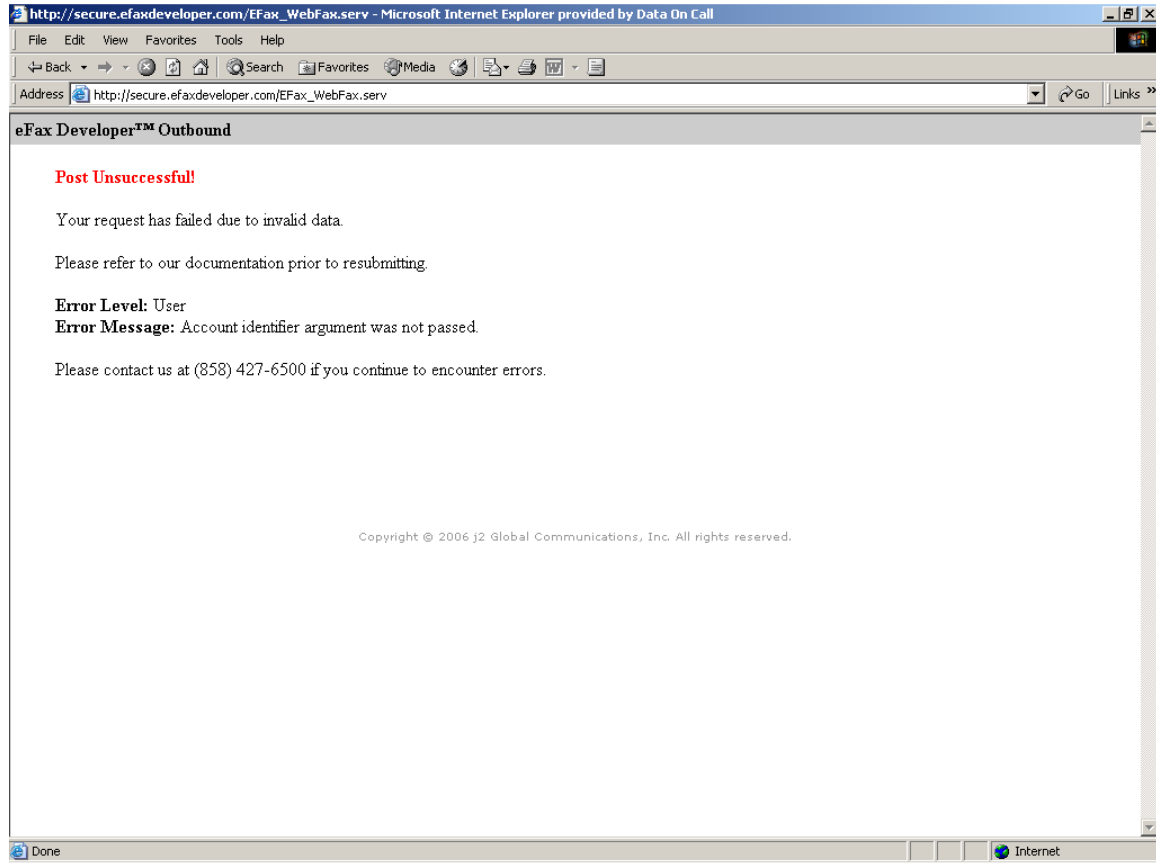
Outbound Status Response HTML

Upon request, *eFax Developer*TM will send an outbound status response back to the calling application. It is the client's responsibility to parse the status response HTML.

The following HTML will be returned to the client process when an outbound status request is successfully received.



The following HTML will be returned to the client when Outbound or Status requests fail.



XML Response

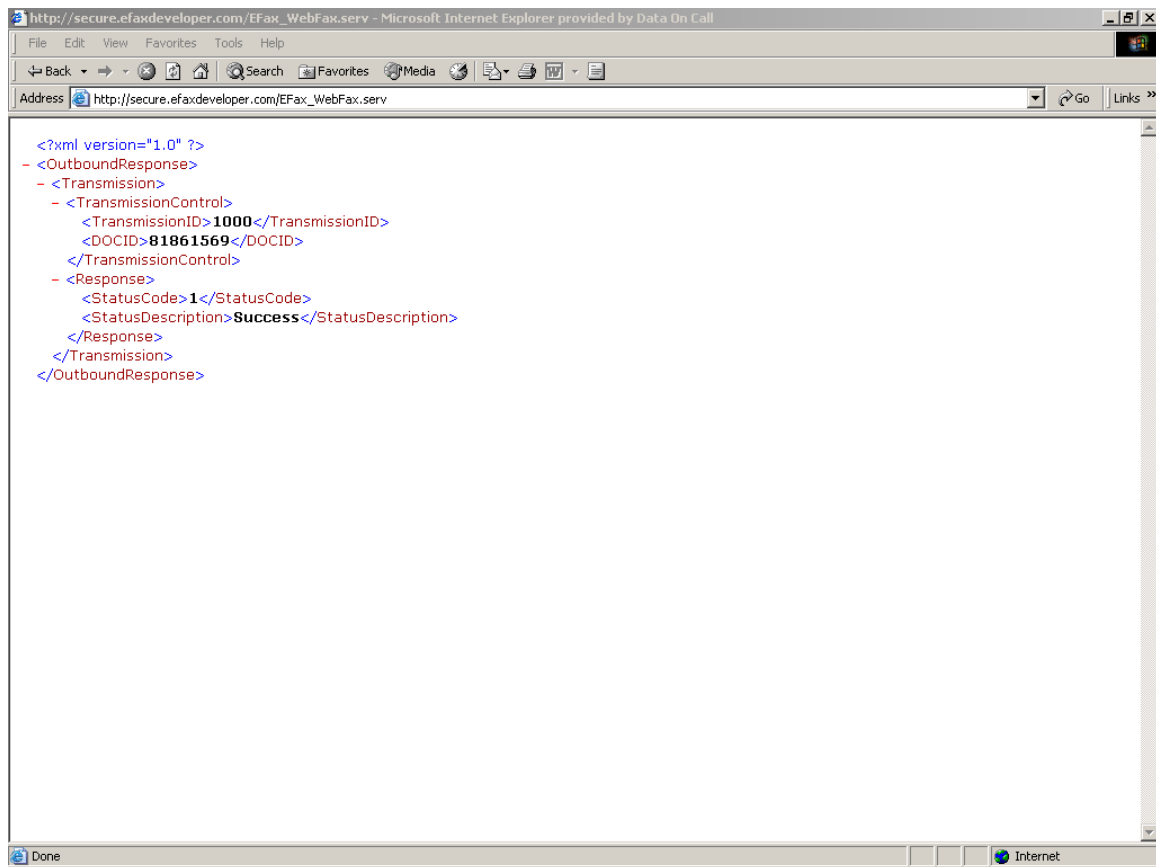
Outbound Response XML

If desired, *eFax Developer*[™] can send an XML-formatted response back to the calling application. It is the client's responsibility to parse this response to determine whether or not the request was successfully received.

A *StatusCode* of "1" indicates a successful request, while a *StatusCode* of "2" indicates failure. The *StatusDescription* element will contain "Success" or "Failure" depending on the *StatusCode* received.

An *ErrorLevel* and *ErrorMessage* will be returned upon a failed request. These fields can be parsed to determine the level of error (*System* or *User*) and the reason for the error.

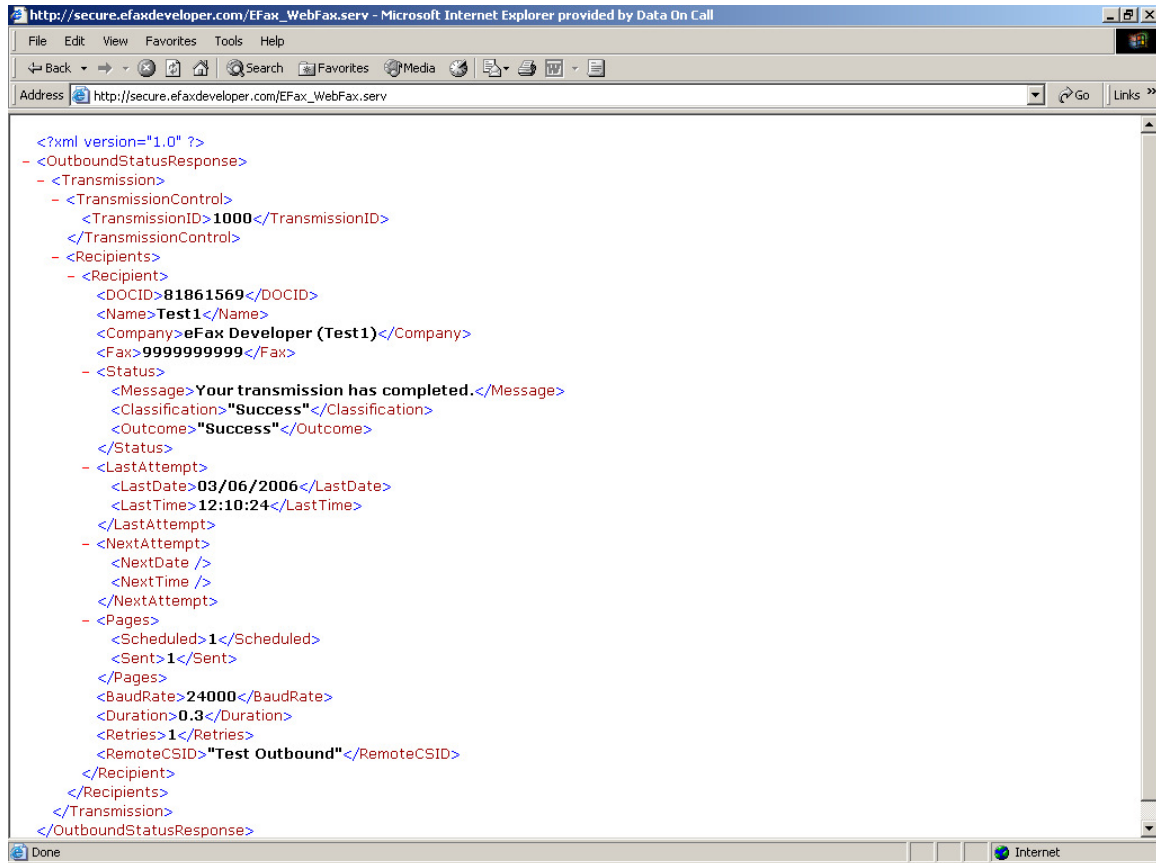
The following example shows an XML-formatted response returned to the client when a POST request is successfully received and the client passed the "respond=XML" argument.



Outbound Status Response XML

Upon request, *eFax Developer*TM can send an XML-formatted status response back to the calling application. It is the client's responsibility to parse the status response XML as desired.

The following example shows an XML-formatted status response returned to the client when a POST request is successfully received and the client passed the "respond=XML" argument.



User Level Error Messages

Requests sent to *eFax Developer™* will be first validated against an XSD before being authenticated by *eFax Developer™* processing. If the request fails validation or authentication, a response will be returned back to the client indicating why the request failed. It is up to the client to decide how these error messages should be handled.

Error Messages Returned by eFax Developer™ Authentication

The following error messages are returned by *eFax Developer™* authentication. When a well-formed request is received by *eFax Developer™*, the request will be authenticated against the client's online profile. If the request fails authentication for any reason, an error message will be returned to the client.

Error Message: Account identifier argument was not passed.

Problem:

The *eFax Developer™* URL did not include an "id" parameter or the "id" parameter value was empty. The "id" parameter value is the account identifier which in most cases will be the primary fax number for the account.

Solution:

Be sure that an "id" parameter and value is included in the URL...

https://secure.efaxdeveloper.com/EFax_WebFax.serv?id=9999999999&xml=<the URL-encoded XML value goes here>

Error Message: XML data argument was not passed.

Problem:

The *eFax Developer™* URL did not include an "xml" parameter or the "xml" parameter value was empty. The "xml" parameter value contains the URL-encoded XML string.

Solution:

Be sure that an "xml" parameter and URL-encoded value is included in the URL...

https://secure.efaxdeveloper.com/EFax_WebFax.serv?id=9999999999&xml=<the URL-encoded XML value goes here>

Error Message: Login not successful. UserName, Password or IP Address failed login validation.

Problem:

The account could not be validated using the username, password or IP address provided.

Solution:

Log in at <https://secure.efaxdeveloper.com>.

From the “Services” page, click on the [eFax Developer](#) link.

From the “Inbox” page, click the “Settings” icon.

From the “Outbound Settings” tab, make sure the username and password is in sync with the <UserName> and <Password> elements passed in the XML.

If the account is configured to validate sending IP addresses, make sure the sending IP is included in the list of Approved IP Addresses.

Error Message: Request content length (nnnnnnnn) has exceeded our 30MB ceiling.

Problem:

The total content length of the POST request exceeded 30MB and was rejected.

Solution:

Break up the transmission into separate requests if possible. Please contact *eFax Developer*TM Support if you continue to encounter this exception.

Error Message: This account is no longer active. Please contact eFax Developer Customer Support.

Problem:

The account is no longer active or closed.

Solution:

Contact *eFax Developer*TM Customer Support for further assistance.

Error Message: Demo account has reached its maximum allotment. Please contact eFax Developer Customer Support.

Problem:

A demo account is expired or reached the maximum number of allowed faxes for the demo.

Solution:

Contact *eFax Developer*TM Customer Support for further assistance.

Error Message: This account has been placed on hold. Please contact eFax Developer Customer Support.

Problem:

The account is on hold.

Solution:

Contact *eFax Developer*TM Customer Support for further assistance.

Error Message: Requests from this account are prohibited. Please contact eFax Developer Customer Support.

Problem:

The account has been blocked from submitting requests.

The account is continuously submitting bad requests or bad attachments, or is misusing the service in some manner. This is an extreme measure imposed by eFax DeveloperTM Support after all other avenues of communication with the primary account holder have been exhausted.

Solution:

Contact *eFax Developer*TM Customer Support for further assistance.

Error Message: TransmissionID required when NoDuplicates option is enabled.

Problem:

A <NoDuplicates> tag was included but a <TransmissionID> element was missing or empty.

Solution:

Add a <TransmissionID> element within the <TransmissionControl> container or remove the <NoDuplicates> tag from the request XML.

Error Message: Duplicate TransmissionID not allowed when NoDuplicates option is enabled.

Problem:

A <NoDuplicates> tag was included in the request XML, and the transmission identifier passed in the <TransmissionID> element was already submitted by an earlier request.

Solution:

Handle as desired in the client-side process.

Error Message: A transmission or DOC identifier is required for a status request.

Problem:

The status request XML did not contain a <TransmissionID> or <DOCID> element.

Solution:

Add a <TransmissionID> or <DOCID> element within the <TransmissionControl> container of the status request XML.

Error Messages Returned by XSD Validation

The following error messages are returned by *eFax Developer*™ XSD validation. XML that contains unexpected or unrecognized elements either by name or location in the XML structure, or contains elements with values that fail to meet criteria defined in this document will be rejected.

Error Message: the value is not a member of the enumeration: (“XXX/XXX/XXX”)

Problem:

An element held an unrecognized value.

Solution:

Make sure the following element values meet the specifications outlined in this document.

<Resolution>	("FINE"/"STANDARD")
<Priority>	("NORMAL"/"HIGH")
<SelfBusy>	("ENABLE"/"DISABLE")
<DispositionLevel>	("BOTH"/"SUCCESS"/"ERROR"/"NONE")
<DispositionMethod>	("POST"/"EMAIL"/"NONE")

Error Message: the value is not a member of the enumeration.

Problem:

An element held an unrecognized value.

Solution:

Make sure the following element values meet the specifications outlined in this document.

<FileType>
<DispositionLanguage>

Error Message: the length of the value is *nnn*, but the required minimum is *nnn*.

Problem:

The <RecipientFax> element held a value that did not meet minimum length requirements.

Solution:

Make sure the <RecipientFax> element value meets the minimum required length outlined in this document.

Error Message: the length of the value is *nnn*, but the required maximum is *nnn*.

Problem:

An element held a value that exceeded a maximum length requirement.

Solution:

Make sure the following element values meet the specifications outlined in this document.

<CustomerID>
<TransmissionID>
<DispositionURL>
<RecipientFax>

Error Message: Content of element “XXXX” is incomplete

Problem:

The element specified in the error message did not contain a value.

Solution:

Make sure the value inserted into the specified element is not an empty string.

Error Message: “XXXX” does not satisfy the “base64Binary” type

Problem:

A <FileContents> element did not contain a Base64 encoded value as required.

Solution:

Make sure all <FileContents> element values are Base64 encoded strings.

Libraries that support Base64 encoding are available in most programming languages. Additional information regarding Base64 Content-Transfer-Encoding can be found at the following links...

- <http://www.freesoft.org/CIE/RFC/1521/7.htm>
- http://www.w3.org/Protocols/rfc1341/5_Content-Transfer-Encoding.html

Samples

Outbound Request XML Example

The following XML shows a two file transmission with multi-email disposition and a dynamic fax header.

```
<?xml version="1.0"?>
<OutboundRequest>
  <AccessControl>
    <UserName>aaa</UserName>
    <Password>aaa</Password>
  </AccessControl>
  <Transmission>
    <TransmissionControl>
      <TransmissionID>1000</TransmissionID>
      <Resolution>STANDARD</Resolution>
      <Priority>NORMAL</Priority>
      <SelfBusy>ENABLE</SelfBusy>
      <FaxHeader>"@DATE1 @TIME3 @ROUTETO{26} @RCVRFAX Pg%P/@TPAGES"</FaxHeader>
    </TransmissionControl>
    <DispositionControl>
      <DispositionURL></DispositionURL>
      <DispositionLevel>BOTH</DispositionLevel>
      <DispositionMethod>EMAIL</DispositionMethod>
      <DispositionEmails>
        <DispositionEmail>
          <DispositionRecipient>Email Recipient One</DispositionRecipient>
          <DispositionAddress>First recipient email address here...</DispositionAddress>
        </DispositionEmail>
        <DispositionEmail>
          <DispositionRecipient>Email Recipient Two</DispositionRecipient>
          <DispositionAddress>Another recipient email address here...</DispositionAddress>
        </DispositionEmail>
      </DispositionEmails>
    </DispositionControl>
    <Recipients>
      <Recipient>
        <RecipientName>aaaaaaaa</RecipientName>
        <RecipientCompany>aaaaaaaa</RecipientCompany>
        <RecipientFax>9999999999</RecipientFax>
      </Recipient>
    </Recipients>
    <Files>
      <File>
        <FileContents>Base64 encoded file contents here...</FileContents>
        <FileType>doc</FileType>
      </File>
      <File>
        <FileContents>Base64 encoded file contents here...</FileContents>
        <FileType>tif</FileType>
      </File>
    </Files>
  </Transmission>
</OutboundRequest>
```

Outbound Status Request XML Example

The following XML shows an outbound status request passing the client specified transmission identifier along with the *eFax Developer*™ transmission identifier.

```
<?xml version="1.0"?>
<OutboundStatus>
  <AccessControl>
    <UserName>aaa</UserName>
    <Password>aaa</Password>
  </AccessControl>
  <Transmission>
    <TransmissionControl>
      <TransmissionID>1000</TransmissionID>
      <DOCID>15642373</DOCID>
    </TransmissionControl>
  </Transmission>
</OutboundStatus>
```

Outbound Disposition XML Example

The following XML shows a standard outbound disposition notification.

```
<OutboundDisposition UserName="aaa" Password="aaa" TransmissionID="1000" DOCID="15642373"
FaxNumber="9999999999" CompletionDate="2003-05-19 13:18:08" FaxStatus="0"
RecipientCSID="MJC Inbound" Duration="0.8" PagesSent="2" NumberOfRetries="1"/>
```

XSD (Schemata) Example

```
<?xml version="1.0"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="OutboundRequest" type="RequestComplexType">

    <xsd:annotation>
      <xsd:documentation xml:lang="en">
        eFax Developer Outbound Request.
        Copyright 2014 j2 Global, Inc. All rights reserved.
      </xsd:documentation>
    </xsd:annotation>

  </xsd:element>

  <xsd:complexType name="RequestComplexType">
    <xsd:sequence>
      <xsd:element name="AccessControl" type="AccessComplexType"/>
      <xsd:element name="Transmission" type="TransmissionComplexType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="AccessComplexType">
    <xsd:sequence>
      <xsd:element name="UserName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="Password" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="Version" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="TransmissionComplexType">
    <xsd:sequence>
      <xsd:element name="TransmissionControl" type="TransmissionControlComplexType"/>
      <xsd:element name="DispositionControl" type="DispositionControlComplexType"/>
      <xsd:element name="Recipients" type="RecipientsComplexType"/>
      <xsd:element name="Files" type="FilesComplexType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="TransmissionControlComplexType">
    <xsd:sequence>

      <xsd:element name="TransmissionID" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="15"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>

      <xsd:element name="CustomerID" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:maxLength value="50"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>

      <xsd:element name="NoDuplicates" type="xsd:string" minOccurs="0"/>
      <xsd:element name="Resolution" type="ResolutionValues" default="STANDARD"/>
      <xsd:element name="Priority" type="PriorityValues" default="NORMAL" minOccurs="0"/>
      <xsd:element name="SelfBusy" type="SelfBusyValues" default="ENABLE" minOccurs="0"/>
      <xsd:element name="FaxHeader" type="xsd:string" minOccurs="0"/>
      <xsd:element name="TSID" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>

  </xsd:complexType>

</xsd:schema>
```

```

</xsd:complexType>

<xsd:complexType name="DispositionControlComplexType">
  <xsd:sequence>

    <xsd:element name="DispositionURL" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="100"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>

    <xsd:element name="DispositionLevel" type="DispositionLevelValues" default="NONE"/>
    <xsd:element name="DispositionMethod" type="DispositionMethodValues" default="POST" minOccurs="0"/>
    <xsd:element name="DispositionLanguage" type="LanguageISOCodeValues" default="en" minOccurs="0"/>
    <xsd:element name="DispositionEmails" type="DispositionEmailsComplexType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DispositionEmailsComplexType">
  <xsd:sequence>
    <xsd:element name="DispositionEmail" minOccurs="0" maxOccurs="3">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="DispositionRecipient" type="xsd:string" minOccurs="0"/>
          <xsd:element name="DispositionAddress" type="xsd:string" minOccurs="1"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RecipientsComplexType">
  <xsd:sequence>
    <xsd:element name="Recipient" minOccurs="1" maxOccurs="1">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="RecipientName" type="xsd:string" minOccurs="0"/>
          <xsd:element name="RecipientCompany" type="xsd:string" minOccurs="0"/>
          <xsd:element name="RecipientFax" type="FaxNumberFormat" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FilesComplexType">
  <xsd:sequence>
    <xsd:element name="File" minOccurs="1" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="FileContents" type="xsd:base64Binary" minOccurs="1" maxOccurs="1"/>
          <xsd:element name="FileType" type="FileTypeValues" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="ResolutionValues">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="STANDARD"/>
    <xsd:enumeration value="FINE"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PriorityValues">
  <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="NORMAL"/>
        <xsd:enumeration value="HIGH"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="SelfBusyValues">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ENABLE"/>
        <xsd:enumeration value="DISABLE"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="LanguageISOCodeValues">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="en"/>
        <xsd:enumeration value="de"/>
        <xsd:enumeration value="es"/>
        <xsd:enumeration value="fr"/>
        <xsd:enumeration value="it"/>
        <xsd:enumeration value="nl"/>
        <xsd:enumeration value="pl"/>
        <xsd:enumeration value="pt"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="FaxNumberFormat">
    <xsd:restriction base="xsd:string">
        <xsd:minLength value="5"/>
        <xsd:maxLength value="25"/>
        <xsd:pattern value="[()+-0123456789]{5,25}"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="FaxNumberDomesticOnly">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="([1]{1})?(\\s|-)?(\\([?1-9]{3}\\))?(\\s|-)?[0-9]{3}-?[0-9]{4}" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="FileTypeValues">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="doc"/>
        <xsd:enumeration value="docx"/>
        <xsd:enumeration value="ppt"/>
        <xsd:enumeration value="pptx"/>
        <xsd:enumeration value="xls"/>
        <xsd:enumeration value="xlsx"/>
        <xsd:enumeration value="tif"/>
        <xsd:enumeration value="tiff"/>
        <xsd:enumeration value="pdf"/>
        <xsd:enumeration value="txt"/>
        <xsd:enumeration value="html"/>
        <xsd:enumeration value="htm"/>
        <xsd:enumeration value="jpg"/>
        <xsd:enumeration value="jpeg"/>
        <xsd:enumeration value="gif"/>
        <xsd:enumeration value="rtf"/>
        <xsd:enumeration value="snp"/>
        <xsd:enumeration value="png"/>
        <xsd:enumeration value="DOC"/>
        <xsd:enumeration value="DOCX"/>
        <xsd:enumeration value="PPT"/>
        <xsd:enumeration value="PPTX"/>
        <xsd:enumeration value="XLS"/>
        <xsd:enumeration value="XLSX"/>
        <xsd:enumeration value="TIF"/>
        <xsd:enumeration value="TIFF"/>
        <xsd:enumeration value="PDF"/>
        <xsd:enumeration value="TXT"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

        <xsd:enumeration value="HTML"/>
        <xsd:enumeration value="HTM"/>
        <xsd:enumeration value="JPG"/>
        <xsd:enumeration value="JPEG"/>
        <xsd:enumeration value="GIF"/>
        <xsd:enumeration value="RTF"/>
        <xsd:enumeration value="SNP"/>
        <xsd:enumeration value="PNG"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="DispositionLevelValues">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ERROR"/>
        <xsd:enumeration value="SUCCESS"/>
        <xsd:enumeration value="BOTH"/>
        <xsd:enumeration value="NONE"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="DispositionMethodValues">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="POST"/>
        <xsd:enumeration value="EMAIL"/>
        <xsd:enumeration value="NONE"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:element name="OutboundDisposition">

    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            eFax Developer Outbound Disposition.
            Copyright 2014 j2 Global, Inc. All rights reserved.
        </xsd:documentation>
    </xsd:annotation>

    <xsd:complexType>
        <xsd:attribute name="UserName" type="xsd:string" use="required" fixed="user_name_heren"/>
        <xsd:attribute name="Password" type="xsd:string" use="required" fixed="password_here"/>
        <xsd:attribute name="TransmissionID" type="xsd:string" use="required"/>
        <xsd:attribute name="DOCID" type="xsd:string" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:length value="8"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="FaxNumber" type="xsd:string" use="required"/>
        <xsd:attribute name="CompletionDate" type="xsd:string" use="required"/>
        <xsd:attribute name="FaxStatus" type="xsd:integer" use="required"/>
        <xsd:attribute name="recipientCSID" type="xsd:string" use="optional">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="20"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="duration" type="xsd:string" use="optional"/>
        <xsd:attribute name="pagesSent" type="xsd:integer" use="optional"/>
        <xsd:attribute name="numberOfRetries" type="xsd:integer" use="optional"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="OutboundStatus" type="StatusComplexType">

    <xsd:annotation>
        <xsd:documentation xml:lang="en">

```

```

    eFax Developer Outbound Status.
    Copyright 2014 j2 Global, Inc. All rights reserved.
</xsd:documentation>
</xsd:annotation>

</xsd:element>

<xsd:complexType name="StatusComplexType">
  <xsd:sequence>
    <xsd:element name="AccessControl" type="AccessComplexTypeStatus"/>
    <xsd:element name="Transmission" type="TransmissionComplexTypeStatus"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AccessComplexTypeStatus">
  <xsd:sequence>
    <xsd:element name="UserName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="Password" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TransmissionComplexTypeStatus">
  <xsd:sequence>
    <xsd:element name="TransmissionControl" type="TransmissionControlComplexTypeStatus"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TransmissionControlComplexTypeStatus">
  <xsd:sequence>
    <xsd:element name="TransmissionID" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DOCID" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```