Data On Call
CORPORATE MESSAGING SOLUTIONS

WebFaxAPI
for
WebFax Enterprise™

User Guide

Revision Date: 06/14/2005

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

Contents

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

Notice

In all communications with Data On Call 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 Data On Call is strictly prohibited.

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved.

Welcome

Welcome to Data On Call's *WebFaxAPI User's Guide*. This guide was designed to assist you in integrating *WebFax Enterprise*™ into your current processing.

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

Before continuing, please consider using one of Data On Call's easy to use SDK packages. Our Software Developer Kits are 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 the *WebFax Enterprise*™.
- **Product Integration** will provide information for establishing a *WebFaxAPI*.
- **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
- **Samples** will provide sample XML and XSD files.

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

Product Introduction

WebFax Enterprise™ Overview

Data On Call's *WebFax Enterprise*™ is a service that software applications can use to send multiple electronic documents to a specified fax number.

Based on XML technology, *WebFax Enterprise*™ is a simple solution for developers at all experience levels. Developers POST requests to *WebFax Enterprise*™ 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.

WebFax Enterprise™ will authenticate the request, then immediately notify the client whether or not the POST 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 POST request. A client can suppress notification, or generate notification upon success, failure, or both.

Components

Security

WebFax Enterprise™ is designed with data security in mind. If desired, requests can be made using the secure HTTPS POST method to ensure both senders and receivers are the only parties able to decode the request. Each request will be authenticated against the requesting client's profile and XSD (schemata).

Client Profile

WebFax Enterprise™ client profile will consist of a user name, password and a unique Account Identifier provided by Data On Call. If desired, clients may choose to have all submitting IP Address' validated as an additional layer of security.

XSD (XML Schema Definition)

To ensure data integrity, *WebFax Enterprise*™ will validate all requests against the client's XSD file. XSD files are client specific and generated by Data On Call 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. *WebFax Enterprise*™ 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.

WebFax Enterprise™ 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 *WebFax Enterprise*™. 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 *WebFax Enterprise*™. *WebFax Enterprise*™ defaults to HTML-formatted responses.

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 *WebFax Enterprise™* with your application, you must develop custom code that adapts your business logic and data to *WebFax Enterprise™*. 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 *WebFax Enterprise™*, 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 Data On Call to establish your account and receive your unique account identifier.
- 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.
- Test your application's integration with Data On Call.
- Make your application live.

Programming

Client-Side Programming Overview

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

- Open a TCP/IP socket or URL connection using the URL address for *WebFax Enterprise*™. The *WebFax Enterprise*™ URL is http://secureservices.dataoncall.com/WebFax.serv (or https://secureservices.dataoncall.com/WebFax.serv if you prefer to POST securely).

- 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 *WebFax Enterprise*™. To send a request, an application writes a stream of data to *WebFax Enterprise*™ 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 by Data On Call; 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."

- Parse the formatted response returned by *WebFax Enterprise*™ for success or failure.

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

- Follow client-side **request** processing guide-lines displayed above.

- Parse the formatted response returned by *WebFax Enterprise*™ as desired.

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

- If indicated, *WebFax Enterprise*™ can send an XML-formatted disposition via HTTP POST to a URL specified in the "DispositionURL" element of the request XML. *WebFax Enterprise*™ will make an initial attempt to deliver this disposition. If delivery is not successful, *WebFax Enterprise*™ will retry once a minute for 15 minutes, then continue at 15 minute intervals until 30 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 from the client-side "disposition response" process or our retry limit is exhausted.

- 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 *WebFax Enterprise*™.

- If desired, *WebFax Enterprise*™ 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

WebFax Enterprise™ 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 8587123637 **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	yyyymmdd	(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

POST Request XML

Element	Required/Cardinality	Type	Length	Description/Values
WebFaxAPIRequest/ **Access Control**	Yes/One	Container	N/A	Access Control Container
WebFaxAPIRequest/ AccessControl/ **UserName**	Yes/One	Alphanumeric	20	User Name
WebFaxAPIRequest/ AccessControl/ **Password**	Yes/One	Alphanumeric	20	User Password
WebFaxAPIRequest/ **Transmission**	Yes/One	Container	N/A	Transmission Container
WebFaxAPIRequest/ Transmission/ **TransmissionControl**	Yes/One	Container	N/A	Transmission Control Container
WebFaxAPIRequest/ Transmission/ TransmissionControl/ **TransmissionID**	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
WebFaxAPIRequest/ 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 *WebFax Enterprise*™ to verify that the transmission identifier has not already been used by a past transmission. When enabled, *WebFax Enterprise*™ will fail the transmission if a duplicate transmission identifier exists.

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

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

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

WebFaxAPIRequest/ Transmission/ **DispositionControl**	Yes/One	Container	N/A	Disposition Control Container
WebFaxAPIRequest/ Transmission/ DispositionControl/ **DispositionURL**	No/Zero or One	Any URI	100	Contains the destination URL to receive final disposition notifications.
WebFaxAPIRequest/ Transmission/ DispositionControl/ **DispositionLevel**	Yes/One	Alphanumeric	7	“ERROR” “SUCCESS” “BOTH” “NONE” 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.
WebFaxAPIRequest/ Transmission/ DispositionControl/ **DispositionMethod**	No/Zero or One	Alphanumeric	5	“POST” “EMAIL” Defines delivery method for the final disposition.
WebFaxAPIRequest/ Transmission/ DispositionControl/ **DispositionEmails**	No/Zero or One	Container	N/A	Disposition Emails Container will hold one-to-many disposition email containers.
WebFaxAPIRequest/ Transmission/ DispositionControl/ DispositionEmails/ **DispositionEmail**	No/Zero or Many	Container	N/A	Disposition Email Container
WebFaxAPIRequest/ Transmission/ DispositionControl/ DispositionEmails/ DispositionEmail/ **DispositionRecipient**	No/Zero or One	Alphanumeric	ANY	The recipient’s *friendly* name to be displayed as part of the disposition email address.
WebFaxAPIRequest/ Transmission/ DispositionControl/ DispositionEmails/ DispositionEmail/ **DispositionAddress**	Yes/One	Alphanumeric	ANY	The recipient’s disposition email address.

WebFaxAPIRequest/ Transmission/**Recipients**	Yes/One	Container	N/A	Recipients Container
WebFaxAPIRequest/ Transmission/Recipients/ **Recipient**	Yes/One	Container	N/A	Recipient Container
WebFaxAPIRequest/ Transmission/Recipients/ Recipient/ **RecipientName**	No/Zero or One	Alphanumeric	50	The recipient’s name when supplied will be merged into the fax header line. Please review the section on “Dynamic Fax Headers” when using that option.
WebFaxAPIRequest/ 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. Please review the section on “Dynamic Fax Headers” when using that option.
WebFaxAPIRequest/ 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 (“+”).
WebFaxAPIRequest/ Transmission/**Files**	Yes/One	Container	N/A	Files Container will hold one-to-many File containers.
WebFaxAPIRequest/ Transmission/Files/**File**	Yes/One to Many	Container	N/A	File Container
WebFaxAPIRequest/ Transmission/Files/File/ **FileContents**	Yes/One	Base64 encoded text	ANY	Base64 encoded file contents.
WebFaxAPIRequest/ Transmission/Files/File/ **FileType**	Yes/One	Alphanumeric	4	“doc” “xls” “tif” “pdf” “txt” “html” “htm” “rtf” Defines the file type being transmitted.

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

Status Request XML

Element	Required/Cardinality	Type	Length	Description/Values
WebFaxAPIStatus/ Access Control	Yes/One	Container	N/A	Access Control Container
WebFaxAPIStatus/ AccessControl/ **UserName**	Yes/One	Alphanumeric	20	User Name
WebFaxAPIStatus/ AccessControl/ **Password**	Yes/One	Alphanumeric	20	User Password
WebFaxAPIStatus/ **Transmission**	Yes/One	Container	N/A	Transmission Container
WebFaxAPIStatus/ Transmission/ **TransmissionControl**	Yes/One	Container	N/A	Transmission Control Container
WebFaxAPIStatus/ Transmission/ TransmissionControl/ **TransmissionID**	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
WebFaxAPIStatus/ Transmission/ TransmissionControl/ **DOCID**	No/Zero or One	Alphanumeric	N/A	Unique Data On Call specified transmission identifier.

POST Response XML

Attribute	Required	Type	Length	Description/Values
WebFaxAPIResponse/ **Transmission**	Yes/One	Container	N/A	Transmission Container
WebFaxAPIResponse/ Transmission/ **TransmissionControl**	Yes/One	Container	N/A	Transmission Control Container
WebFaxAPIResponse/ Transmission/ TransmissionControl/ **TransmissionID**	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
WebFaxAPIResponse/ Transmission/ TransmissionControl/ **DOCID**	No/Zero or One	Alphanumeric	8	Unique Data On Call specified transmission identifier.
WebFaxAPIResponse/ Transmission/ **Response**	Yes/One	Container	N/A	Response Container
WebFaxAPIResponse/ Transmission/ Response/ **StatusCode**	Yes	Alphanumeric	1	“1” Success “2” Failure Requests return status
WebFaxAPIResponse/ Transmission/ Response/ **StatusDescription**	Yes	Alphanumeric	7	“Success” “Failure” Requests return status value
WebFaxAPIResponse/ 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 Data On Call intervention.
WebFaxAPIResponse/ Transmission/ Response/ **ErrorMessage**	No/Zero or One	Alphanumeric	ANY	The generated error message.

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

Status Response XML

Attribute	Required	Type	Length	Description/Values
WebFaxAPIStatusResponse/ **Transmission**	Yes/One	Container	N/A	Transmission Container
WebFaxAPIStatusResponse/ Transmission/ **TransmissionControl**	Yes/One	Container	N/A	Transmission Control Container
WebFaxAPIStatusResponse/ Transmission/ TransmissionControl/ **TransmissionID**	No/Zero or One	Alphanumeric	15	Unique client specified transmission identifier.
WebFaxAPIStatusResponse/ Transmission/**Recipients**	Yes/One	Container	N/A	Recipients Container
WebFaxAPIStatusResponse/ Transmission/Recipients/ **Recipient**	Yes/One	Container	N/A	Recipient Container
WebFaxAPIStatusResponse/ Transmission/Recipients/ Recipient/**DOCID**	Yes	Alphanumeric	8	Unique Data On Call specified transmission identifier.
WebFaxAPIStatusResponse/ Transmission/Recipients/ Recipient/**Name**	No/Zero or One	Alphanumeric	50	The recipient's name.
WebFaxAPIStatusResponse/ Transmission/Recipients/ Recipient/**Company**	No/Zero or One	Alphanumeric	20	The recipient's company name.
WebFaxAPIStatusResponse/ Transmission/Recipients/ Recipient/**Fax**	Yes/One	Alphanumeric	25	The recipient's fax number.
WebFaxAPIStatusResponse/ Transmission/Recipients/ Recipient/**Status**	Yes/One	Container	N/A	Status Container
WebFaxAPIStatusResponse/ Transmission/Recipients/ Recipient/Status/**Message**	Yes/One	Alphanumeric	100	The status message for this fax transmission.
WebFaxAPIStatusResponse/ 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.

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

WebFaxAPIStatusResponse/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.
WebFaxAPIStatusResponse/Transmission/Recipients/Recipient/**LastAttempt**	Yes/One	Container	N/A	Last Attempt Container
WebFaxAPIStatusResponse/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.
WebFaxAPIStatusResponse/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).
WebFaxAPIStatusResponse/Transmission/Recipients/Recipient/**NextAttempt**	Yes/One	Container	N/A	Next Attempt Container
WebFaxAPIStatusResponse/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.
WebFaxAPIStatusResponse/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).

WebFaxAPIStatusResponse/ Transmission/Recipients/ Recipient/**Pages**	Yes/One	Container	N/A	Pages Container
WebFaxAPIStatusResponse/ 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.
WebFaxAPIStatusResponse/ 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.
WebFaxAPIStatusResponse/ 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.
WebFaxAPIStatusResponse/ 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.
WebFaxAPIStatusResponse/ 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.
WebFaxAPIStatusResponse/ 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.

Disposition Response XML

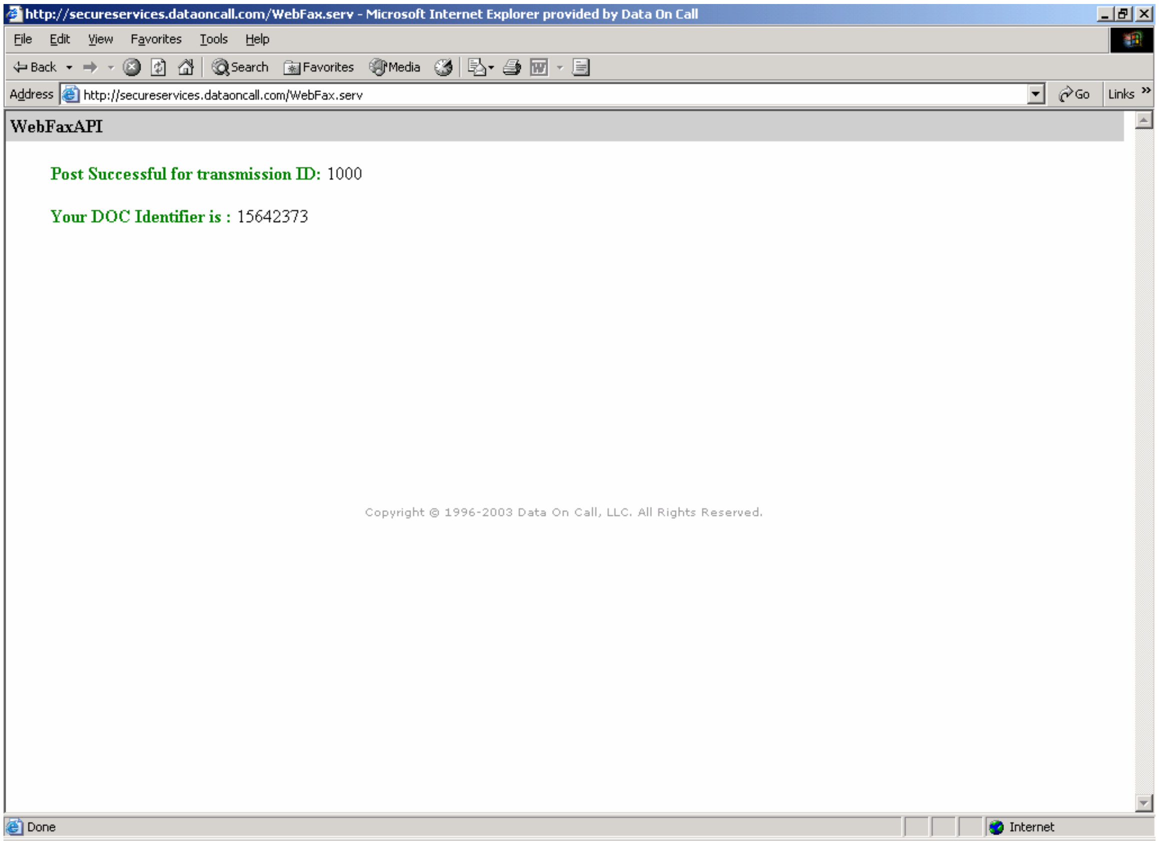
Attribute	Required	Type	Length	Description/Values
WebFaxAPIDisposition/ **UserName**	Yes	Alphanumeric	20	User Name
WebFaxAPIDisposition/ **Password**	Yes	Alphanumeric	20	User Password
WebFaxAPIDisposition/ **TransmissionID**	No	Alphanumeric	15	Unique client specified transmission identifier.
WebFaxAPIDisposition/ **DOCID**	Yes	Alphanumeric	8	Unique Data On Call specified transmission identifier.
WebFaxAPIDisposition/ **FaxNumber**	Yes	Alphanumeric	25	The recipient's fax number.
WebFaxAPIDisposition/ **CompletionDate**	Yes	Alphanumeric	20	Fax completion date and time. Time zone is PST. Format is yyyy-mm-dd hh:mm:ss (24 hour).
WebFaxAPIDisposition/ **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 a Data On Call supplied table.
WebFaxAPIDisposition/ **RecipientCSID**	No	Alphanumeric	20	The station identifier, when supplied by the receiving fax machine upon successful transmission.
WebFaxAPIDisposition/ **Duration**	No	Alphanumeric	5	Transmission time in minutes.
WebFaxAPIDisposition/ **PagesSent**	No	Numeric	3	The number of pages sent.
WebFaxAPIDisposition/ **NumberOfRetries**	No	Numeric	2	Number of times the fax was attempted before success or failure.

HTML Response

POST Response HTML

WebFax Enterprise™ 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 WebFax Enterprise™ and should not be confused with the final disposition.*

The following HTML will be returned to the client when the POST request was successfully received.


http://secureservices.dataoncall.com/WebFax.serv - Microsoft Internet Explorer provided by Data On Call
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Address
http://secureservices.dataoncall.com/WebFax.serv
Go
Links
Done
Internet

WebFaxAPI

Post Successful for transmission ID: 1000

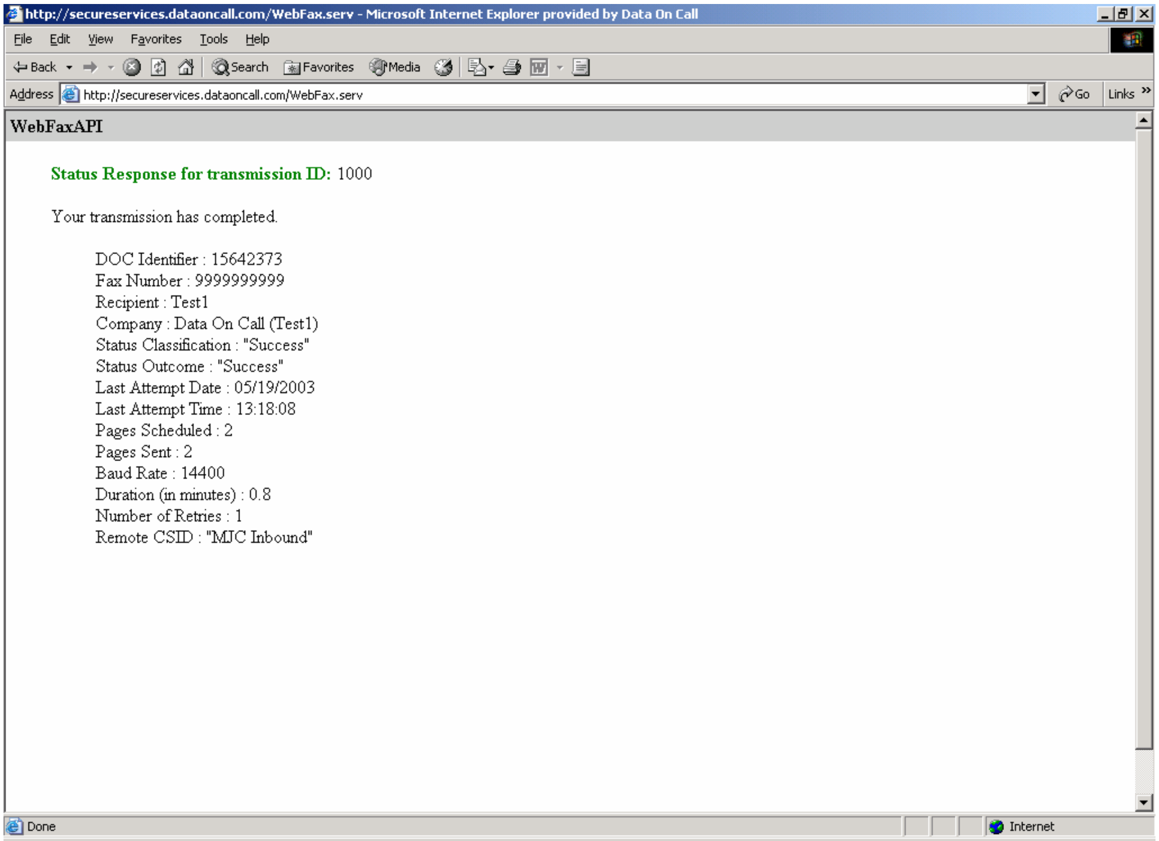
Your DOC Identifier is : 15642373

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved.

Status Response HTML

Upon request, *WebFaxEnterprise*™ will send a 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 when a status request is successfully received.


http://secureservices.dataoncall.com/WebFax.serv - Microsoft Internet Explorer provided by Data On Call
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://secureservices.dataoncall.com/WebFax.serv
Go Links
Done
Internet

WebFaxAPI

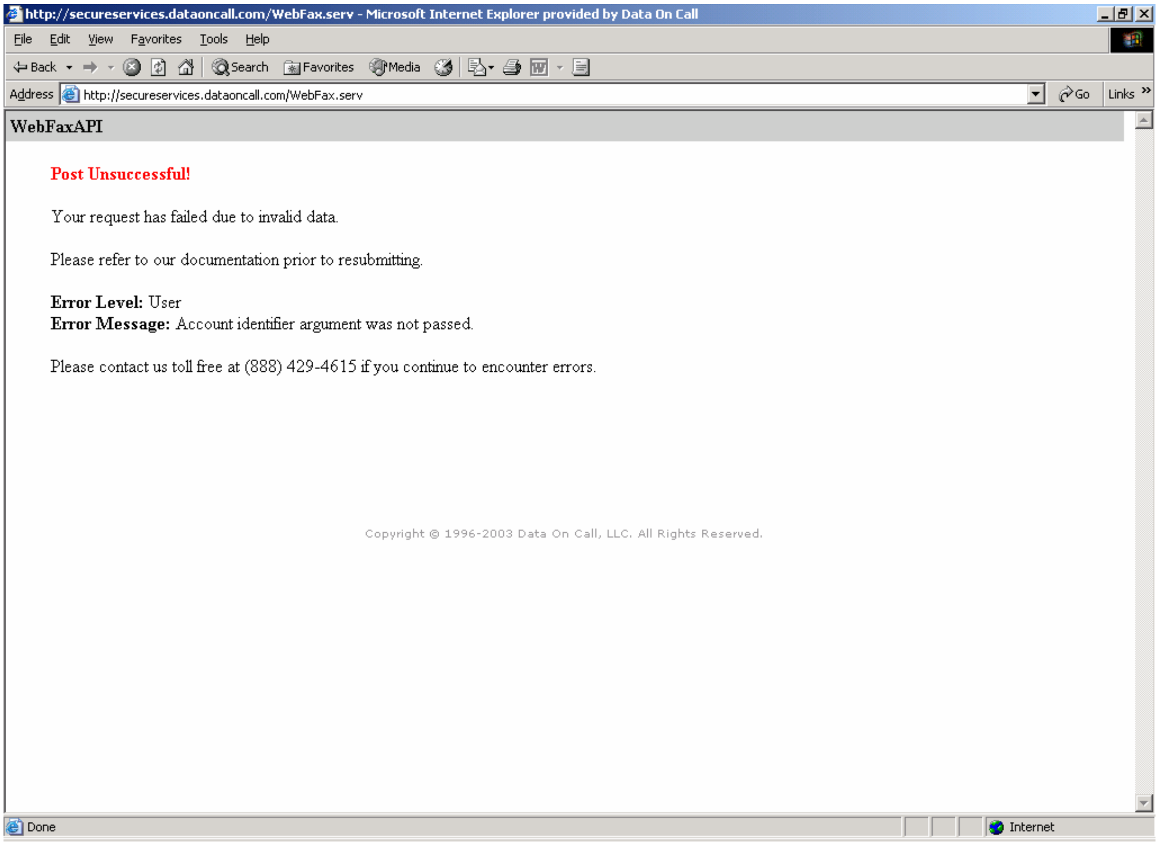
Status Response for transmission ID: 1000

Your transmission has completed.

DOC Identifier : 15642373
Fax Number : 9999999999
Recipient : Test1
Company : Data On Call (Test1)
Status Classification : "Success"
Status Outcome : "Success"
Last Attempt Date : 05/19/2003
Last Attempt Time : 13:18:08
Pages Scheduled : 2
Pages Sent : 2
Baud Rate : 14400
Duration (in minutes) : 0.8
Number of Retries : 1
Remote CSID : "MJC Inbound"

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

The following HTML will be returned to the client when a POST or Status request fails.


http://secureservices.dataoncall.com/WebFax.serv - Microsoft Internet Explorer provided by Data On Call
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Address
http://secureservices.dataoncall.com/WebFax.serv
Go
Links
Done
Internet

WebFaxAPI

Post Unsuccessful!

Your request has failed due to invalid data.

Please refer to our documentation prior to resubmitting.

Error Level: User
Error Message: Account identifier argument was not passed.

Please contact us toll free at (888) 429-4615 if you continue to encounter errors.

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved.

XML Response

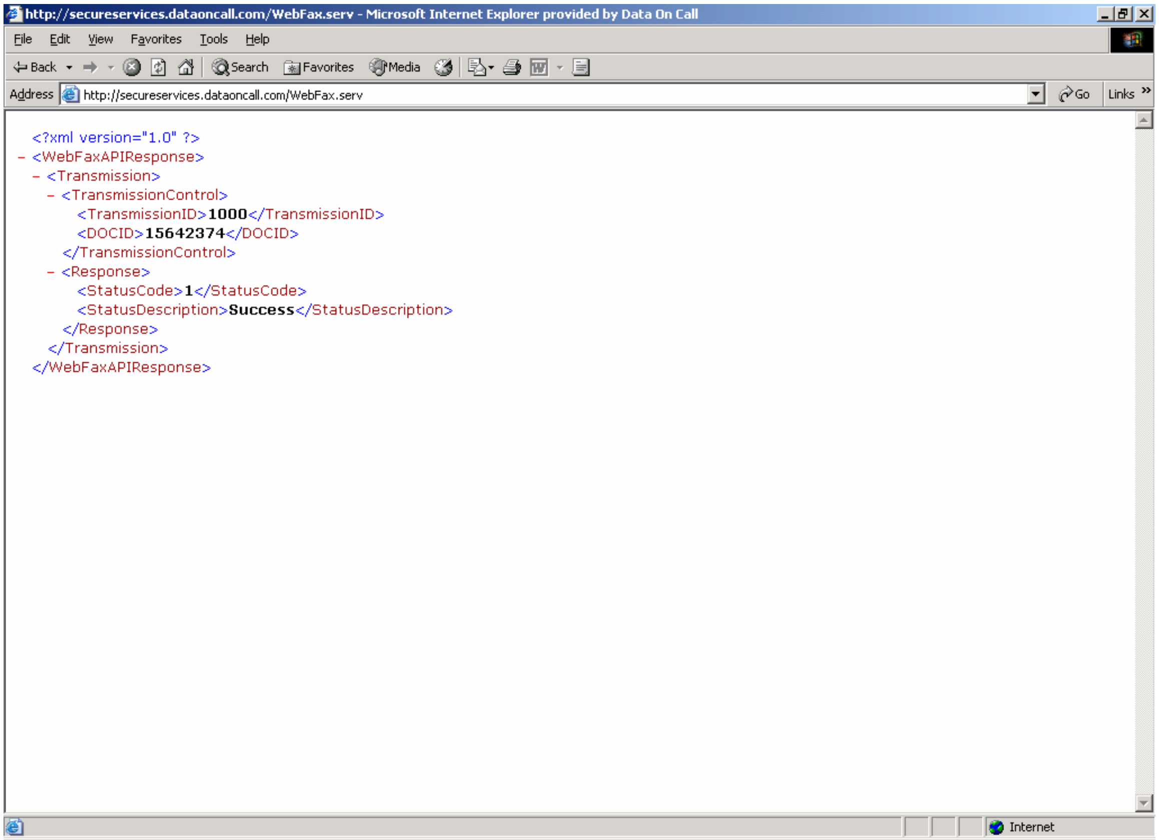
POST Response XML

If desired, *WebFax Enterprise*™ 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.


http://secureservices.dataoncall.com/WebFax.serv - Microsoft Internet Explorer provided by Data On Call
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://secureservices.dataoncall.com/WebFax.serv
Go
Links
Internet

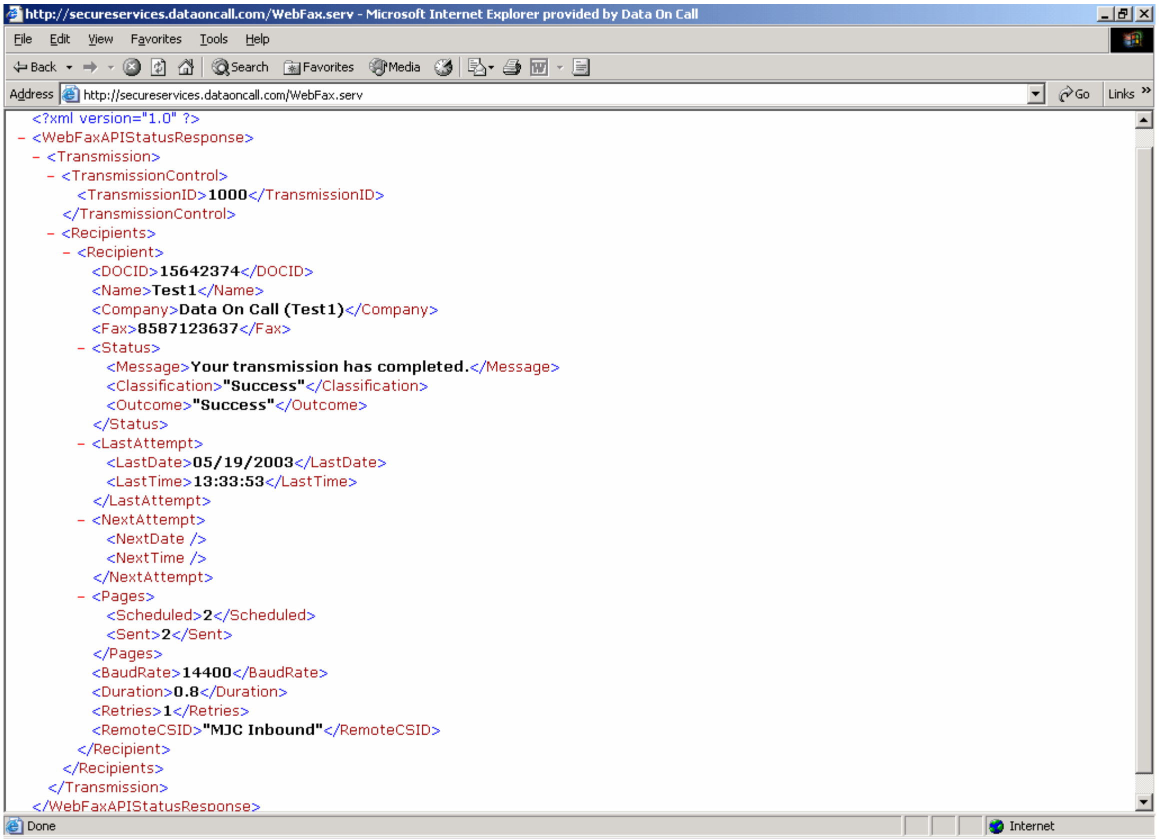
```
  <?xml version="1.0" ?>
- <WebFaxAPIResponse>
  - <Transmission>
    - <TransmissionControl>
        <TransmissionID>1000</TransmissionID>
        <DOCID>15642374</DOCID>
      </TransmissionControl>
    - <Response>
        <StatusCode>1</StatusCode>
        <StatusDescription>Success</StatusDescription>
      </Response>
    </Transmission>
  </WebFaxAPIResponse>
```

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

Status Response XML

Upon request, *WebFax Enterprise*™ 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.



http://secureservices.dataoncall.com/WebFax.serv - Microsoft Internet Explorer provided by Data On Call
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://secureservices.dataoncall.com/WebFax.serv
Go Links
Done
Internet

```
   <?xml version="1.0" ?>
 - <WebFaxAPIStatusResponse>
   - <Transmission>
     - <TransmissionControl>
         <TransmissionID>1000</TransmissionID>
       </TransmissionControl>
     - <Recipients>
       - <Recipient>
           <DOCID>15642374</DOCID>
           <Name>Test1</Name>
           <Company>Data On Call (Test1)</Company>
           <Fax>8587123637</Fax>
         - <Status>
             <Message>Your transmission has completed.</Message>
             <Classification>"Success"</Classification>
             <Outcome>"Success"</Outcome>
           </Status>
         - <LastAttempt>
             <LastDate>05/19/2003</LastDate>
             <LastTime>13:33:53</LastTime>
           </LastAttempt>
         - <NextAttempt>
             <NextDate />
             <NextTime />
           </NextAttempt>
         - <Pages>
             <Scheduled>2</Scheduled>
             <Sent>2</Sent>
           </Pages>
           <BaudRate>14400</BaudRate>
           <Duration>0.8</Duration>
           <Retries>1</Retries>
           <RemoteCSID>"MJC Inbound"</RemoteCSID>
         </Recipient>
       </Recipients>
     </Transmission>
   </WebFaxAPIStatusResponse>
```

Copyright © 1996-2003 Data On Call, LLC. All Rights Reserved

Samples

POST Request XML Example

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

```
<?xml version="1.0"?>
<WebFaxAPIRequest>
  <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>one@dataoncall.com</DispositionAddress>
              </DispositionEmail>
              <DispositionEmail>
                <DispositionRecipient>Email Recipient Two</DispositionRecipient>
                <DispositionAddress>two@dataoncall.com</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>
</WebFaxAPIRequest>
```

Status Request XML Example

The following XML shows a status request passing the client specified transmission identifier along with the Data On Call transmission identifier.

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

Disposition XML Example

The following XML shows a standard disposition notification.

```
<WebFaxAPIDisposition 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="WebFaxAPIRequest" type="RequestComplexType">
   <xsd:annotation>
    <xsd:documentation xml:lang="en">
     Data On Call WebFaxAPI Request.
     Copyright 2003 Data On Call. 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: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"    type="xsd:string" minOccurs="0"/>
     <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:sequence>
   </xsd:complexType>

   <xsd:complexType name="DispositionControlComplexType">
    <xsd:sequence>
     <xsd:element name="DispositionURL"          type="xsd:string" minOccurs="0"/>
     <xsd:element name="DispositionLevel"        type="DispositionLevelValues"  default="NONE"/>
     <xsd:element name="DispositionMethod"       type="DispositionMethodValues" default="POST"/>
     <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="unbounded">
    <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="FaxFormat" 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="FaxFormat">
    <xsd:restriction base="xsd:integer">
      <xsd:pattern value="[1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="FileTypeValues">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="doc"/>
      <xsd:enumeration value="xls"/>
      <xsd:enumeration value="tif"/>
      <xsd:enumeration value="pdf"/>
      <xsd:enumeration value="txt"/>
      <xsd:enumeration value="html"/>
      <xsd:enumeration value="htm"/>
    </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="WebFaxAPIDisposition">

 <xsd:annotation>
  <xsd:documentation xml:lang="en">
   Data On Call WebFaxAPI Disposition.
   Copyright 2003 Data On Call. All rights reserved.
  </xsd:documentation>
 </xsd:annotation>

 <xsd:complexType>
  <xsd:attribute name="UserName"          type="xsd:string" use="required" fixed="user_name_here"/>
  <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" 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" 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="WebFaxAPIStatus" type="StatusComplexType">

   <xsd:annotation>
    <xsd:documentation xml:lang="en">
     Data On Call WebFaxAPI Status.
     Copyright 2003 Data On Call. 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>
```