



CFX_iMSMail for inFusion Mail Server

Copyrights

inFusion Mail Server® is a product of On-Line Data Solutions, Inc.

Trademarks

inFusion Mail Server® is a registered trademark of On-Line Data Solutions, Inc.

iMS® is a registered trademark of On-Line Data Solutions, Inc.

ColdFusion is a trademark of Macromedia Corporation

BlueDragon is a trademark of New Atlanta Corporation

Suggestions and Support

Your suggestions for inFusion Mail Server and CFX_iMSMail are welcomed.
Please email your comments to support@CoolFusion.com.

Technical support for iMS, excluding FusionMail and iMS-Lite, can be obtained via:

- email: support@coolfusion.com
- the iMS Support list

Details on joining the support list can be found at:
<http://www.coolfusion.com/support>.

Support for freeware products, including FusionMail and iMS-Lite, require paid support:

<http://www.coolfusion.com/commerce/index.cfm?category=6>

On-Line Data Solutions, Inc.
1919 Middle Country Road STE 204
Centereach, NY 11720-3501

Phone: (631) 737-4668

FAX: (631) 737-9539

Sales: sales@CoolFusion.com

Technical Support: support@CoolFusion.com



Copyright © 1999-2006 On-Line Data Solutions, Inc.
Manual revision 4.7

CFX_iMSMail v 4.7 Introduction.....	1
Feature overview	1
Installation.....	5
Feature details	6
Inserts mail directly into the iMS POST queue.....	6
Complete header control	7
Support for HTML and multipart mail	8
Optional token definitions	8
Support for SMTP Authentication on sent mail	8
Email recipient Exclusions	8
Automatic rollback	9
Email address validator	9
Direct ODBC queries	9
Direct ODBC queries for exclusions	9
Sending time designation	9
Multiple Priority Queues.....	9
Personalization	11
Tag Usage	13
Tag Syntax.....	14
Syntax Details	16
Return Codes and Information.....	27
Example output	28
Result Display Example	29
Automatic MIME type detection	30
Tag usage examples.....	35
Basic Example.....	36
Basic Example With Personalization	37
Notes	37
Qtokens Example (specifying tokens)	38
Dynamic HTML content example	40
Directly Querying an ODBC database.....	41
Querying recipients and exclusions from an ODBC database.....	42
Preconfiguring Datasources.....	43
Editing a Datasource	44

CFX_iMSMail v 4.7 Introduction

CFX_iMSMail is an add-on tag for both Macromedia ColdFusion (including CFMX) and BlueDragon from New Atlanta. CFX_iMSMail allows you to easily send email using iMS (inFusion Mail Server) and is not only a great replacement for CFMAIL but can also do many more things than CFMAIL can.

The CFX_iMSMail ColdFusion custom tag allows you to put email into the iMS POST queue with minimal effort. This tag is installed into the iMS installation directory. You can move the tag to any directory that you choose as long as you specify the proper location in the ColdFusion or BlueDragon administrator.

Feature overview

- Inserts mail directly into the iMS POST queue.
- Dynamic html documents with embedded content (images, etc)
- Supports load balancing across multiple POST Servers.
- Supports tokenization from queries. You can therefore, send mass customized emails quickly and easily.
- Supports dynamic and fixed emails. You can create an email using the specified tag attributes or you can specify a raw email file as well. Using raw email files allows you to take any email received by the iMS SMTP server and forward to any number of recipients (a good application for this is an email list server).
- Supports any number of email headers. This is a great innovation over other CFX email tags. You need only preface the attribute name with "header_" to add any header that you choose. For example, if you wanted to add a reply-to header you simply set header_Reply_To="<some email address>".
- Supports addition of multiple recipients into the same control file when the destination domain is the same. This saves on disk space and greatly improves POST processing efficiency.

- Optionally writes to a log file. This is particularly helpful when tracing log files for a particular mail. For example, a user says that they sent a mail to a particular email address but the mail never arrived. First, you look in the SMTP log file and find the record for the particular mail. Then, you look in the CFX_iMSMail log file (the log files that have file names beginning with "CFXI" and in that log file you can find the unique message ID. Finally, you check the POST log files to see the status of that message according to the message ID.
- Supports HTML and multipart mail.
- Supports RFC822-style attachments (to attach the contents of an entire email in the new email)
- Can optionally include token definitions.
- Can save a copy of the sent email to a specified directory.
- Can automatically delete the raw file after sending mail.
- Support for SMTP Authentication on sent mail.
- Supports sending mail to one or more other iMS POST Servers (full UNC support)
- Exclusions parameter to remove exclusions from a recipients list.
- Qtokens are automatically created from query columns.
- MaxAttempts and SendingIP parameters
- Render plain text from HTML source
- Force sending of plain text or HTML exclusively
- Improved email MIME encoding

New Features in version 4.7

New BinaryEncodeHeaders and BinaryEncodeBody parameters

New Features in version 4.6

none

New Features in version 4.5

New attachments from memory option

New Features in version 4.4

New UTF-8 encoding via EncodeTo parameter

New Features in version 4.3

CFX_iMSMail version 4.3 adds the RemoveDupes parameter which guarantees that no email addresses are duplicated in the output.

(Note: version 4.2 was an internal version that was not released)

New Features in version 4.1

CFX_iMSMail version 4.1 was updated to support the CF_MAIL tag set which emulates the ColdFusion and BlueDragon CFMAIL tag (see the CF_MAIL tag set documentation for details). The new parameters (which are outlined in the parameters section of this manual) are:

- TokenNames
- TokenValues
- StartRow
- MaxRows

New Features in version 4.0

Version 4.0 is a major rewrite of the previous version. The new version uses significantly less memory and also adds the ability to utilize ODBC datasources directly. The important new features are:

- Faster processing for large mailings
- Lower memory usage
- Automatic rollback
- New email address validator
- Rejected recipients list

- ODBC Queue support (for iMS 3.0)
- Direct ODBC queries
- Direct ODBC queries for exclusions
- Sending time designation
- Enhanced Logging
- Virtually unlimited priorities (for iMS 2.8 or higher)
- Flexible delivery retry times (for iMS 2.8 or higher)

Installation

The CFX tag will work with ColdFusion 2 through 5, CFMX and BlueDragon. The automated installer will attempt to register the tag. To register the tag with ColdFusion or BlueDragon manually, refer to the associated product documentation.

Feature details

CFX_iMSMail, along with iMS, provides a host of advanced features and functionality that can be utilized for sending all types of emails including standard person to person emails, list emails, personalized emails (great for CRM applications) and automated emails. iMS and CFX_iMSMail are extremely flexible and fast. This section details the most important features.

Inserts mail directly into the iMS POST queue

Unlike other CFMAIL replacements that use SMTP to send email, the CFX_iMSMail tag inserts the email job directly into the iMS POST Server queue (the iMS POST server is the mail sending engine behind iMS) thus alleviating the additional overhead required when sending the email twice (once to get the email to the mail server and again when the mail server sends the mail to the final destination). It is also important to note that iMS is usually configured to send mail directly to its final destination so no external email server is required to set up and maintain.

Dynamic html documents with embedded content (images, etc)

CFX_iMSMail provides the capability to specify images or other content that is to be embedded in the email (as opposed to linking to an image served from a remote web server). The tag will automatically encode and embed the content into the email and will include each item only once, even if it is referenced multiple times in the HTML content.

Support load balancing across multiple POST Servers.

Email jobs can be submitted to multiple iMS POST Servers for processing. Each iMS server will receive a proportional amount of the total job. In order to do this, ColdFusion or BlueDragon must be configured to log in as a user with rights to the remote iMS servers. In addition, the *SPOOLDIR* parameter will need to be configured, and UNC path names must be used.

Personalization via Query data.

iMS and CFX_iMSMail support the concept of Qtokens which are used to send personalized emails to recipients. Tokenization means that emails can be sent with special embedded placeholders that will be interpreted by the iMS server when sending mail to a specific recipient. In this way, each recipient will, for example, see his or her name, or other personal information, in the text of the email.

Tokens can be obtained from a query that has been generated from within the CFML template, or the tag can optionally specify a query using the built-in ODBC query support to create dynamic content. Each column from a query is treated as a Qtoken, so information from any column can be used in a customized email.

For example, with the following query:

```
select email, name from table1
```

the *name* column is automatically made a Qtoken that can be embedded into the email.

Dear <:name:>

Thank you for your interest in our services.

The CFX_iMSMail tag and iMS have utilized the “name: column from the database query automatically. You must delineate the Qtoken in your email using the proper start and end tag text. The default start and end tags are “<:” and “:;>” however, any characters may be used for start and end tokens, provided they are defined using the *TOKENSTART* and *TOKENEND* parameters.

Complete header control

CFX_iMSMail supports any number of email headers. This is a great innovation over other CFX email tags. Headers are created by prefacing the attribute name with *header_*. For example, to add a reply-to header, simply set:

```
header_Reply_To="<some email address>".
```

The example above assumes that the email is being created within the tag, using the *BODY* and/or *HTML* attributes. If an email source file is being used (e.g., you are relaying a mail to other recipients but want to customize the headers) the *TOKEN_ADDHEADER* and/or *TOKEN_SETHEADER* parameters must be used.

Support for HTML and multipart mail

The CFX_iMSMail tag can create plain text, HTML or multipart emails. Multipart emails are emails that contain both a plain text and HTML version of the content. An email that is in multipart format is rendered by the recipient's email program in either the plain text or HTML version depending on the end user preference. Multipart emails can be sent with CFX_iMSMail by specifying both the *BODY* and *HTML* attributes. An alternate method would be to supply just the *HTML* attribute and use the *RENDERPLAINTEXT* attribute to instruct the CFX tag to create a plain text rendering of the HTML portion.

Optional token definitions

CFX_iMSMail supports global tokens in addition to Qtokens. Global tokens are common to all recipients of an email. For example, to specify a token that contains the next date/time of a meeting which will be a common item in all of the received emails:

```
<cfset nextmeeting="March 1, 2005 at 8PM">
```

```
<cfx_imsmail Token_NextMeeting="#NextMeeting#"
body="Please be advised that our next meeting will be <:NextMeeting:>">
```

Support for SMTP Authentication on sent mail

Although iMS can send mail directly to the recipient's email server it is also possible for iMS to relay mail through one or more local servers. If the local servers require SMTP authentication, use the AUTHUSER and AUTHPASS parameters to allow iMS to authenticate to the relay mail server.

Email recipient Exclusions

Sometimes it is convenient to send mass emailings by selecting a group of addresses from a database without having to worry about recipients that have specifically asked to be excluded from future mailings. In cases like these it is possible to:

- specify a list of email addresses to exclude from a mailing, using the *EXCLUSIONS* parameter

or

- specify a query for the CFX_iMSMail tag to use when compiling a list of email addresses to exclude, using the *EXODBCQUERY* parameter.

Automatic rollback

If an error occurs while sending a mailing (e.g., disk is full) then all of the files created up to that point will automatically be deleted.

Email address validator

The CFX tag contains an internal email address validator which is enabled by default but may be disabled by setting *VALIDATEADDRESSES*="NO" in the CFX tag parameters. The validator will weed out addresses that are not valid and will return the list of invalid email addresses in the *iMSMail_RejectedList* return query.

Direct ODBC queries

The tag has the ability to query an ODBC database directly, using the *ODBCQUERY* parameter. This parameter can be used alone, or in addition to the *QUERY* parameter, which designates a query that has been executed in the CFML template. It may also be used with the *SMTPTO* parameter, which is a comma-separated list of email addresses.

Direct ODBC queries for exclusions

The tag has the ability to query an ODBC database directly for email exclusions, using the *EXODBCQUERY* parameter. Email exclusions are lists of email addresses to which the email should not be delivered. This parameter can be used alone, or in addition to the *EXCLUSIONS* parameter, which is a comma-separated list of email addresses.

Sending time designation

A delivery date and time can be specified for the email job, using the *START* parameter.

Multiple Priority Queues

The CFX_imsMail tag and iMS allow prioritization of the email jobs that are submitted. A higher priority email, designated by a lower priority number, will take delivery precedence over a lower priority email. iMS version 2.8 or higher allows a virtually unlimited number of priorities (0 to 2,147,483,647) while older versions of iMS have 10 priorities (0 to 9). Priority 0 is the highest level priority.

Flexible delivery retry times (for iMS 2.8 or higher)

When attempting to deliver an email there may be instances where the sending needs to be retried (the recipient mailbox is full, for example).

iMS version 2.8 or higher allows the designation of a retry time (in minutes) for every email that is sent, using the *RETRYINTERVAL* parameter. If this parameter is not set, the default retry time configured in the POST Server settings will be used.

Personalization

Personalization is one of the most important aspects of email delivery. Personalization makes email discussion lists, CRM applications, etc., possible. iMS and CFX_iMSMail were designed from the ground up with flexible personalization in mind.

iMS and CFX_iMSMail use tokens for email personalization. Tokens are small, textual representations that are replaced with the actual database values when the email is sent. For example, a token called NAME would be replaced with the actual name of the recipient that was stored in the database.

Tokens can be global or user-specific.

Global tokens will have the same value for all recipients of a particular email. Global tokens are specified in the tag parameters with a prefix of “*Token_*”. For example, a global token might be designated by:

```
Token_sender="sales@example.com"
```

If the token “*sender*” is included in the email, the token will be replaced by the sales email address shown above.

User-specific tokens - Qtokens, or “Query Tokens” - will be customized for each recipient.

When sending personalized mail, use the *QUERY* and/or *ODBCQUERY* parameters of the tag. The CFX tag will automatically utilize each column in the query for Qtokens. For example, if the database has a column called “name” there will be an available Qtoken called NAME. Please refer to the examples later on in this manual.

Tokens, whether global or Qtokens, always require a prefixed and post fixed delimiter. The default delimiters in iMS are “<:” and “:;>”. However, these default delimiters may be configured with different values in the iMS server configuration.

If you are unsure of the settings in the iMS server, you should use the *TOKENSTART* and *TOKENEND* parameters to specify the token delimiters that you wish to use. Do not set the *TOKENSTART* and *TOKENEND* parameters to the same value.

There are also several predefined tokens that can be used in mailings. These predefined tokens are as follows:

SMTPTO
This is the email address of the recipient
HEADER_XXXX
These are the headers that are in the email. For example, you can reference a token called HEADER_SUBJECT and, as long as the email contains a subject header, the text of the subject will replace the token in the email.

Tag Usage

Once the CFX tag is installed it can be utilized in CFML templates as a replacement for the CFMAIL tag. There are many advantages in using the CFX tag as opposed to the standard CFMAIL tag and, if iMS installed on the server, there is really no reason to ever use CFMAIL again.

It is fairly simple to alter the syntax of a CFML template when using the CFX_iMSMail tag. For example, here is a simple email that is sent using CFMAIL:

```
<cfmail to="joe@example.com" from="null@example.com"  
subject="Registration confirmation">
```

```
You are now registered!  
</cfmail>
```

The same email can be sent using the CFX tag as follows:

```
<cfx_imsmail smtpo="joe@example.com" smtpfrom="null@example.com"  
header_to="joe@example.com" header-from="null@example.com"  
header_subject="Registration confirmation"  
body="You are now registered!">
```

The biggest difference between CFMAIL and CFX_iMSMail in the above example is that the parameter count is slightly higher for the CFX tag. The reason for this is that the CFX tag allows you complete control over the headers of the mail and who the mail is to and from in the SMTP protocol (SMTP protocol is the language that used by mail senders and receivers). It is important to note that the headers in the mail above denoting the sender and receiver (which are displayed when you view the mail) are not necessarily the same as the sender and receiver specified when the email is transmitted. Because of this, we can set the header_to="Loyal customer" and set the smtpo="joe@example.com". In this case, joe@example.com still gets the email but the "To" address when Joe views the email will show "Loyal Customer."

CFX_iMSMail supports several attributes as outlined on the next page:

Tag Syntax

<cfx_iMsmail

ATTACHMENTS="<attachment(s)>"
AUTHPASS="<Password for SMTP Authentication>"
AUTHUSER="<User name for SMTP Authentication>"
BinaryEncodeHeaders="YES/NO"
BinaryEncodeHeaders="YES/NO"
BODY="<email body>"
BOUNDARY="<MIME Boundary>"
CHARSET="<character set>"
CREATEMAILFILEONLY="YES/NO"
DATASOURCE="<ODBC Datasource name>"
DELETEMAILFILE="YES/NO"
EMAILFILE="<raw email file>"
ENABLEWORDWRAP="YES/NO"
ENCODETO="<Character set>"
EXCLUSIONS="<recipient exclusions>"
EXODBCQUERY="<ODBC Query for exclusions>"
EXODBCQUERYFIELD="<Column that contains the email address>"
FAILTO="<email address>"
HEADER_XXXXX="<header value>"
HOSTNAME="<Host name for delivery>"
HTML="<html encoded email body>"
HTMLONLY="YES/NO"
JOBID="<User-defined Job ID>"
LOG="YES/NO"
LOGDIR="<Log directory>"
MAXATTEMPTS="<maximum delivery attempts>"
MAXRECIPIENTS="<Max. recipients per control file>"
MAXROWS="<maximum query rows>"
MX="<Mail Server(s) to send mail to>"
ODBCQUERY="<ODBC Query for recipients>"
ODBCQUERYFIELD="<Column name that contains the email address>"
PACKAGE="YES/NO"
PARSETOKENS="YES"
PASSWORD="<ODBC Password>"
PLAINTEXTONLY="YES/NO"
PRIORITY="<mail priority>"
QTOKEN_XXXXX="<query field>"
QUERY="<email query>"
QUERYFIELD="<field in query>"
QUEUETYPE="ODBC"
QUOTEDPRINTABLE="YES/NO"
REMOVEDUPES="YES/NO"
RENDERPLAINTEXT="YES/NO"

RETRYINTERVAL="<minutes between retries>"
RFC822ATTACHMENT="<email attachment>"
SAVECOPY="YES/NO"
SAVEDIR="<directory>"
SENDINGIP="<IP Address to use for sending mail>"
SMTPFROM="<return address>"
SMTPPORT="<SMTP Port of Mail Server to send mail to>"
SMTPTO="<email recipient(s)>"
SPOOLDIR="<spool directories>"
START="YYYYMMDDHHNNSS"
STARTROW="<row number>"
TOKEN_XXXXX="<token definition>"
TOKEN_ADDHEADER_XXXXX="<header value>"
TOKEN_SETHEADER_XXXXX="<header value>"
TOKENEND="<Token delimiter end>"
TOKENNAMES="<delimited list of names>"
TOKENSTART="<Token delimiter start>"
TOKENVALUES="<delimited list of values>"
USERNAME="<ODBC User Name>"
VALIDATEADDRESSES="YES/NO"
WARNTO="<email address>"
WORDWRAP="<maximum length of line>"
>

Syntax Details

The following table details all of the CFX_imsMail parameters. Although there are many available parameters, mail can be sent by utilizing only a few. Please refer to the examples later on in this manual.

ATTACHMENTS
<p>A comma-delimited list of file attachments. This parameter is used only for dynamically generated emails (i.e. if the <i>EMAILFILE</i> parameter is not specified). The MIME type may also be entered (the CFX will include a MIME type if you omit the type specifically). Note that each entry in the list should be the complete path to the file and that double quotes should be used around the file name if the path or file name contains spaces.</p> <p>attachments="<full path to attachment> <MIME type>"</p> <p>ex: attachments="c:\test.jpg image/jpeg"</p> <p>Version 4.5 enhancement</p> <p>Version 4.5 of the CFX tag also allows you to include attachments from ColdFusion variables instead of disk files. Due to a limitation of the CFX API the value of the variable must be included as an attribute in the tag and must also be Base64 encoded if the value is binary.</p> <p>For example, suppose that you have a binary image in memory referenced by a ColdFusion variable called "myimage" :</p> <pre><cfx_imsmail ... image=ToBase64(myimage) attachments="var:image" ></pre> <p>Please refer to the examples section for additional details.</p>
AUTHPASS
<p>The password, if authenticated SMTP is required for the receiving mail server. This parameter is only used for relaying mail through another email server that requires authentication and is normally not used if IMS is configured to send email directly to the destination.</p>
AUTHUSER
<p>The user name, if authenticated SMTP is required for the receiving mail server. This parameter is only used for relaying mail through another email server that requires authentication and is normally not used if IMS is configured to send email directly to the destination.</p>

AUTOTOKENS
By default, the CFX tag will generate Qtokens based on the input query columns. This behavior can be disabled by setting this parameter to "NO".
BINARYENCODEHEADERS
Forces the "To", "From" and "Subject" headers to be base64 encoded instead of the default quoted-printable or plain text.
BINARYENCODEBODY
Forces the body and html portions of the email to be base64 encoded instead of the default quoted-printable or plain text.
BODY
The body of the email, if the email is dynamically generated. The body may contain one or more tokens.
BOUNDARY
The text to use as the boundary in a MIME encoded email. The tag creates boundaries internally if this parameter is not specified.
CHARSET
The character set of the email. The default is "iso-8859-1". Please note that the tag does not encode the actual email – the email body should already be encoded with the character set specified.
CREATEMAILFILEONLY
Use this parameter to create the mail file only. No control files will be created, so no mails will be sent.
DATASOURCE
The ODBC Datasource name that is to be used when querying a database for recipients and/or email exclusions. This parameter is used in conjunction with the USERNAME and PASSWORD parameters to specify the credentials for accessing the datasource. You can also use the included <i>ODBC credentials editor</i> to store the username and password so that they can be omitted from your CFML scripts.
DELETEMILEFILE
If this parameter is set to "YES", the specified <i>EMAILFILE</i> will be deleted after the CFX tag has been executed.
EMAILFILE
The complete path and file name of the raw email file. This parameter overrides any of the dynamic email parameters (such as the <i>BODY</i> parameter and all of the <i>HEADER</i> parameters).
ENABLEWORDWRAP
Turns automatic word wrapping on/off. Default is "on."
ENCODETO
This parameter specifies the character set that should be used to encode the email text. Currently, the CFX supports encoding to UTF-8 only.
EXCLUSIONS
This is a comma-delimited list of email addresses to be removed from the list of recipients. Note that you can also use the <i>EXODBCQUERY</i> parameter to

retrieve a list of exclusions from a database.
EXODBCQUERY
This parameter specifies a SQL query that is to be used to retrieve a list of email addresses that should not receive the mailing. The <i>DATASOURCE</i> parameter is required when using this parameter.
EXODBCQUERYFIELD
The name of the column that contains the email address of the excluded recipients. Default is "email".
FAILTO
The address to send delivery failure notifications to. This setting overrides the SMTPFROM parameter when returning undeliverable mail. Note that the FAILTO parameter is used exclusively by the iMS server to send notifications. To specify an address that the remote server should use to send bounces to, specify a return path using the <i>header_return_path</i> parameter.
HEADER_xxxxxx
This is any number of header parameters that you may specify for a dynamically generated email. For example, to set a "TO" header which is normally required for an email, enter: HEADER_TO=joe@example.com The tag automatically sets the proper case for each header name in the email, so entering the header in upper or lower case is irrelevant.
HOSTNAME
The host name that the server sends in the helo/ehlo stage of the SMTP protocol. This is normally not needed, the iMS server will use its default. If used, this host name should match the RDNS of the sending IP.
HTML
The body of the email in HTML format if the email is dynamically generated (i.e. the <i>EMAILFILE</i> parameter is not specified).
HTMLONLY
The sent email should contain only the HTML version of the email even if the <i>BODY</i> parameter was specified. The default is "NO" but this can be set to "YES" for an HTML-only email. Note that this parameter will have no effect on an email unless the <i>HTML</i> parameter is also specified.
JOBID
A user-defined JOBID that will be inserted into the ODBC Queue table. This parameter applies only to iMS 3.0 or higher when using an ODBC Queue.

LOG
Set this parameter to "YES" to enable the CFX_iMSMail log file function. The default is disabled. If used, the <i>LOGDIR</i> parameter is required as well.
LOGDIR
The full path to the folder in which to store log files.
MAXATTEMPTS
Maximum number of sending attempts. The default is 0, which means limited only by the maximum delivery time set in the server.
MAXRECIPIENTS
The maximum number of recipients per control file. The default is 100. Note: The POST Server will connect to the remote mail server and deliver mail with as many recipients as possible (to save on bandwidth). If the remote server has limitations on the number of recipients then the POST Server will batch the number of recipients accordingly. So, if you have several recipients in a single control file then this is the best scenario. However, if you have several hundred or thousands of recipients in a control file, then the efficiency will decrease and it would be better to have the recipients spread out over several control files.
MAXROWS
This is the maximum number of rows to process in the QUERY parameter. The default is all rows. You can set this value to any number greater than 0 to limit the number of records processed from the query. Note that this setting applies only to the query passed by the QUERY parameter and has no affect on the ODBCQUERY parameter.
MX
Comma-delimited list of mail servers to send this mail to. If not specified then iMS will do an MX lookup on the recipient domain.
ODBCQUERY
This parameter specifies a SQL query that is to be used to retrieve a list of email recipients. The query must return a column of email addresses and, optionally, may contain one or more additional columns to be used for personalization (Qtokens). The DATASOURCE parameter is required when using this parameter.
ODBCQUERYFIELD
The name of the column that contains the email address of the excluded recipients. Default is "email".
PACKAGE
Instructs the CFX tag to create packages for the iMS POST Server. Packages (available in iMS 2.5d or higher) greatly increase mail handling when there are several recipients for a particular mailing (list mail, for example). The default is YES. Note: In order to use the tag with iMS 2.5c or lower it is mandatory to set this

parameter to "NO".
PARSETOKENS
Force the POST Server to parse tokens. Note: It is not necessary to specify this parameter if tokens are also specified. This parameter is supplied mainly to deal with mail files that use standard (built-in) tokens but no other tokens are specified. (Used when <i>AUTOTOKENS</i> is set to "NO").

PASSWORD
The password that should be used for ODBC queries. You can either specify the password and username parameters or, for added security, you can use the included ODBC Credentials editor to store the username and password instead of designating the credentials in your CFML script.
PLAINTEXTONLY
The sent email should contain only the plain text version of the email even if the <i>HTML</i> parameter was specified. The default is "NO" but can be set to "YES" for a text-only email. Note that this parameter will have no effect on an email unless the <i>BODY</i> parameter is specified or the <i>RENDERPLAINTEXT</i> parameter has been specified along with the <i>HTML</i> parameter.
PRIORITY
The priority of the mail. May be set from 0 to 9 (5 is the default if not specified). 0 is the highest priority. Please note that iMS 2.8 and higher allow a virtually unlimited number of priorities (0- 2,147,483,647). An email with a higher priority (a lower numeric value) will be delivered before an email with a lower priority.
QTOKEN_xxxxxx
This is any number of token definitions that may be specified for an outgoing e-mail. These tokens are retrieved from the specified query (See "QUERY" below). CFX_iMSMail will automatically use the specified query column when creating token substitutions). For example, to set a parameter called "name" enter the token: QTOKEN_NAME="username" The CFX tag will automatically URL-Encode the text. Please note that version 3.3.2 (and higher) of the CFX tag allows you to use Qtokens without defining them. All Qtokens will be generated automatically from the supplied query. If, for example, a query contained a column called "address" they you could use a Qtoken value of "address" in your email body. This behavior can be disabled with the <i>AUTOTOKENS</i> parameter.
QUERY
Optional name of an existing query from which to retrieve recipients. The field of the query that contains the email addresses is in the <i>QUERYFIELD</i> parameter.
QUERYFIELD
The field name in the query from which to retrieve the email addresses. This parameter is required only when the <i>QUERY</i> parameter is specified and the column in the query containing the email address is not called "email."
QUEUETYPE
The type of Queue in which to store the email job. The default is the standard iMS queue but you can optionally specify a QUEUETYPE of "ODBC" for iMS 3.0 or higher.
QUOTEDPRINTABLE
Causes CFX_iMSMail to encode the email text using quoted printable encoding.

Default is "NO." The tag automatically uses quoted-printable for lines that exceed the WordWrap parameter. Turn on quoted printable by setting this parameter to "YES"

REMOVEDUPES
If this parameter is set to "YES" then the tag will remove all duplicate email addresses from the output. Enabling this parameter will cause the tag to use more memory as the tag needs to retain a lookup list of all email addresses. The tag will also process slower due to the additional overhead. It is therefore recommended that you select unique addresses in your query code rather than using this parameter. Default is "NO".
RENDERPLAINTEXT
Instructs the tag to create a plain text version of the <i>HTML</i> parameter. Note that this will cause the text in the <i>BODY</i> parameter to be overwritten. Default is "NO".
RETRYINTERVAL
The number of minutes between retries. The default is 0, which means that the server retry interval will be used. This feature is available in iMS version 2.8 or higher.
RFC822ATTACHMENT
Specifies an e-mail file to embed in the new e-mail file. Use this parameter to send a copy of an e-mail to a recipient as an attachment.
SAVECOPY
Saves a copy of the sent email in the specified <i>SAVEDIR</i> directory.
SAVEDIR
The directory that the raw mail should be saved to.
SENDINGIP
The IP address to use for sending. The default is blank which means that iMS should use its default address. This can be set to any valid IP on the server. Choosing an invalid IP will cause the server to use the Windows default address.
SMTPFROM
The address that the email is from in the SMTP protocol. If this parameter is omitted, an address of <i>NULL</i> is used automatically.
SMTPPORT
The SMTP port of the receiving mail server. The default is the standard SMTP port, 25.
SMTPTO
A comma-delimited list of email addresses of the recipient or recipients. This field is required if you are not using the <i>QUERY</i> or <i>ODBCQUERY</i> parameters.
SPOOLDIR
The location of the iMS POST spool directory. CFX_iMSMail automatically determines the location of the iMS POST spool directory so this parameter is not normally used. Non-local spool directories should be referenced using UNC paths. A comma-delimited list of spool directories - local or on another machine – may also be specified, to implement load balancing. The CFX tag will evenly distribute control files among the specified directories.

START

The start date and time for the email job. The default is the current date and time. This parameter must be in the following format:

yyyymmddhhnnss

where

yyyy = 4-digit year

mm = 2-digit month

dd = 2-digit day

hh = 2-digit hour (military time)

nn = 2-digit minute

ss = 2-digit second

example

To post a mail to the IMS queue that is to be delivered on March 1, 2005 at 3PM then, set *Start* to:

20050301150000

STARTROW

This is the row at which to begin processing the QUERY parameter. The default is row 1. You can set this value to any number greater than 0 set the starting row higher than the default. Note that this setting applies only to the query passed by the QUERY parameter and has no affect on the ODBCQUERY parameter.

TOKEN_XXXXXX

This is any number of token definitions that you may specify for an outgoing e-mail. For example, to set a parameter called "name", enter the token:

TOKEN_NAME="John Smith"

Please note that for email personalization it is better to use Qtokens.

TOKEN_ADDHEADER
<p>A special token that is used to add a header to an existing email. For example, you might have an email source file that is received from a list member and you want to relay the mail to all of the list recipients but you want to add an X-List header signifying the name of the list. Use the <i>EMAILFILE</i> parameter to specify the email source file and then use the <i>TOKEN_ADDHEADER</i> parameter to add a list header as follows:</p> <pre>TOKEN_ADDHEADER_X_LIST="MyList"</pre> <p>Note that tokens can also be used within the header values. For example, to add a header designating the email address of a list member:</p> <pre>TOKEN_ADDHEADER_X_LISTMEMBER= "<:SMTPTO:>"</pre> <p>There is no limit to the number of <i>TOKEN_ADDHEADER</i> parameters.</p>
TOKEN_SETHEADER
<p><i>TOKEN_SETHEADER</i> is a special designation like <i>TOKEN_ADDHEADER</i>. However, instead of adding the header, <i>TOKEN_SETHEADER</i> will only modify a header that already exists. Suppose, for example, you want to create a threaded list archive that is based on the MESSAGE-ID of the email. You would want to remove the message-id header that was created by the email program of the list member and replace it with one of your own. In this case you would do the following:</p> <pre>TOKEN_SETHEADER_MESSAGE_ID="12345"</pre> <p>Note that you can also remove existing headers from an email by setting the value to a blank string. For example, if a list member sends mail at a high priority you may want to remove that designation. You can do that as follows:</p> <pre>Token_setheader_X_MSMail_Priority="" Token_setheader_Importance="" Token_setheader_X_priority=""</pre>
TOKENNAMES / TOKENVALUES
<p>The <i>TOKENNAMES</i> and <i>TOKENVALUES</i> parameters are used in conjunction to pass multiple token name and value pairs to the tag. The lists for each should be TAB-delimited but you can also use comma-delimited lists (tabs are recommended). The number of items in each list should be the same.</p>
TOKENSTART / TOKENEND
<p>The token starting and ending delimiters. Default is "<:" and ":>". Note: Version 1.5e and higher of iMS allows you to specify default start and end delimiters. These parameters allow you to override the default token delimiters. POST Server 1.6 is required for this functionality.</p>

USERNAME
The username that should be used for ODBC queries. You can either specify the password and username parameters or, for added security, you can use the included ODBC Credentials editor to store the username and password instead of designating the credentials in your CFML script.
VALIDATEADDRESSES
Tells the tag to remove email addresses that are not in the proper format. Default is "YES"
WORDWRAP
Column to enforce word wrapping. Default is 70. Set to a value less than 20 to disable word wrap. This parameter is only valid for dynamically generated emails.
WARNTO
The address to send delivery warning notifications to.

Return Codes and Information

CFX_iMSMail provides information pertaining to every aspect of the email job that was created. This information can be used to report status to users or to track the number of mails sent, number of email addresses that were rejected, etc. The following table details the return codes and information from the CFX tag.

IMSMAIL_DEBUG
Not used
IMSMAIL_INPUTRECIPIENTS
The total number of recipients that were submitted.
IMSMAIL_MESSAGE_SIZE
The size of the email in bytes (this value is the same as <i>IMSMAIL_STOREDMESSAGE_SIZE</i> and is provided for backward compatibility).
IMSMAIL_PROCESSTIME
The time (in milliseconds) that the job took to run.
IMSMAIL_REJECTEDLIST
A query with a single column (email), which lists all of the email addresses that were rejected if the address were an invalid format or existed in the exclusions list. The query consists of a single column called EMAIL and contains the email address that was rejected.
IMSMAIL_RESULT
The result of the processing. A result of 0 is equivalent to a successful process. Any other result indicates that an error occurred during processing.
IMSMAIL_RESULTTEXT
A textual representation of the result. This value will be set to "Success" for a successful delivery and the error details if an error occurred during processing.
IMSMAIL_SAVEDFILENAME
The name of the file if the SAVECOPY parameter is used. This parameter will not exist if the SAVECOPY parameter was not specified.
IMSMAILSTOREDFILENAME
The name of the email file that was saved to the iMS queue.
IMSMAIL_STOREDMESSAGE_SIZE
The size of the email in bytes.
IMSMAIL_TOTALEXCLUSIONS
The total number of recipients that were rejected due to exclusion listing.
IMSMAIL_TOTALRECIPIENTS
The total number of recipients that were added to the iMS queue.
IMSMAIL_VERSION
The version of the CFX_iMSMail tag.

Example output

The example output, below, was obtained by including the following code after the call to CFX_iMSMail.

```
<cfdump var="#variables#">
```

struct									
IMSMAIL_DEBUG	[empty string]								
IMSMAIL_MESSAGESIZE	111								
IMSMAIL_PROCESSTIME	47								
IMSMAIL_REJECTEDLIST	<table border="1"><thead><tr><th colspan="2">query</th></tr><tr><th></th><th>EMAIL</th></tr></thead><tbody><tr><td>1</td><td>joe@</td></tr><tr><td>2</td><td>fred@example</td></tr></tbody></table>	query			EMAIL	1	joe@	2	fred@example
query									
	EMAIL								
1	joe@								
2	fred@example								
IMSMAIL_RESULT	0								
IMSMAIL_RESULTTEXT	Success								
IMSMAIL_STOREDFILENAME	E0D66155D42C534EBBCA143C199EF052.mbx								
IMSMAIL_STOREDMESSAGESIZE	111								
IMSMAIL_TOTALEXCLUSIONS	0								
IMSMAIL_TOTALINPUTRECIPIENTS	4								
IMSMAIL_TOTALRECIPIENTS	2								
IMSMAIL_VERSION	4.0								

The most notable information in the example above pertains to the email processing results for the submitted recipients. A total of four recipients were submitted but two were rejected. The two email addresses that were rejected were rejected due to bad formatting as shown by the resultant IMSMAIL_REJECTEDLIST query.

Result Display Example

There may be times when you want to present the sender of the mail with information pertaining to the results of the CFX tag processing. The following code is a simple example that can be expanded to suit your needs:

```
<cfoutput>Your email was sent to #iMSMail_TotalRecipients# recipients  
from a total of #iMSMail_TotalInputRecipients#.</cfoutput><p>
```

```
<cfif isdefined("iMSMail_rejectedlist")>
```

```
There were <cfoutput>#iMSMail_rejectedlist.recordcount# rejections including  
#iMSMail_TotalExclusions# exclusions as follows:</cfoutput>  
<p>
```

```
<cfoutput query="iMSMail_rejectedlist">  
#email#<br>  
</cfoutput>
```

```
<cfelse>  
<cfoutput>There were <cfif iMSMail_TotalExclusions is  
0>no<cfelse>#iMSMail_TotalExclusions#</cfif> exclusions.</cfoutput>  
</cfif>
```

Automatic MIME type detection

CFX_iMSMail inserts MIME types automatically if they are not specified for attachments and <iMSCID> in-line objects. The MIME type that is inserted is dependent upon the file extension according to the following table.

File Extension	MIME Type
ai	application/postscript
aif	audio/x-aiff
aifc	audio/x-aiff
aiff	audio/x-aiff
asc	text/plain
au	audio/basic
avi	video/x-msvideo
bcpio	application/x-bcpio
bin	application/octet-stream
c	text/plain
cc	text/plain
ccad	application/clariscad
cdf	application/x-netcdf
class	application/octet-stream
cpio	application/x-cpio
cpt	application/mac-compactpro
csh	application/x-csh
css	text/css
dcr	application/x-director
dir	application/x-director
dms	application/octet-stream
doc	application/msword
drw	application/drafting
dvi	application/x-dvi
dwg	application/acad
dxg	application/dxf
dxr	application/x-director
eps	application/postscript
etx	text/x-setext
exe	application/octet-stream
ez	application/andrew-inset
f	text/plain
f90	text/plain
fli	video/x-fli
gif	image/gif
gtar	application/x-gtar

File Extension	MIME Type
gz	application/x-gzip
h	text/plain
hdf	application/x-hdf
hh	text/plain
hqx	application/mac-binhex40
htm	text/html
html	text/html
ice	x-conference/x-cooltalk
ief	image/ief
iges	model/iges
igs	model/iges
ips	application/x-ipscript
ipx	application/x-ipix
jpe	image/jpeg
jpeg	image/jpeg
jpg	image/jpeg
js	application/x-javascript
kar	audio/midi
latex	application/x-latex
lha	application/octet-stream
lsp	application/x-lisp
lzh	application/octet-stream
m	text/plain
man	application/x-troff-man
me	application/x-troff-me
mesh	model/mesh
mid	audio/midi
midi	audio/midi
mif	application/vnd.mif
mime	www/mime
mov	video/quicktime
movie	video/x-sgi-movie
mp2	audio/mpeg
mp3	audio/mpeg
mpe	video/mpeg
mpeg	video/mpeg
mpg	video/mpeg
mpga	audio/mpeg
ms	application/x-troff-ms
msh	model/mesh
nc	application/x-netcdf

File Extension	MIME Type
oda	application/oda
pbm	image/x-portable-bitmap
pdb	chemical/x-pdb
pdf	application/pdf
pgm	image/x-portable-graymap
pgn	application/x-chess-pgn
png	image/png
pnm	image/x-portable-anymap
pot	application/mspowerpoint
ppm	image/x-portable-pixmap
pps	application/mspowerpoint
ppt	application/mspowerpoint
ppz	application/mspowerpoint
pre	application/x-freelance
prt	application/pro_eng
ps	application/postscript
qt	video/quicktime
ra	audio/x-realaudio
ram	audio/x-pn-realaudio
ras	image/cmu-raster
rgb	image/x-rgb
rm	audio/x-pn-realaudio
roff	application/x-troff
rpm	audio/x-pn-realaudio-plugin
rtf	text/rtf
rtx	text/richtext
scm	application/x-lotusscreencam
set	application/set
sgm	text/sgml
sgml	text/sgml
sh	application/x-sh
shar	application/x-shar
silo	model/mesh
sit	application/x-stuffit
skd	application/x-koan
skm	application/x-koan
skp	application/x-koan
skt	application/x-koan
smi	application/smil
smil	application/smil
snd	audio/basic

File Extension	MIME Type
sol	application/solids
spl	application/x-futuresplash
src	application/x-wais-source
step	application/STEP
stl	application/SLA
stp	application/STEP
sv4cpio	application/x-sv4cpio
sv4crc	application/x-sv4crc
swf	application/x-shockwave-flash
t	application/x-troff
tar	application/x-tar
tcl	application/x-tcl
tex	application/x-tex
texi	application/x-texinfo
texinfo	application/x-texinfo
tif	image/tiff
tiff	image/tiff
tr	application/x-troff
tsi	audio/TSP-audio
tsp	application/dsptype
tsv	text/tab-separated-values
txt	text/plain
unv	application/i-deas
ustar	application/x-ustar
vcd	application/x-cdlink
vda	application/vda
viv	video/vnd.vivo
vivo	video/vnd.vivo
vrml	model/vrml
wav	audio/x-wav
wrl	model/vrml
xbm	image/x-bitmap
xlc	application/vnd.ms-excel
xll	application/vnd.ms-excel
xlm	application/vnd.ms-excel
xls	application/vnd.ms-excel
xlw	application/vnd.ms-excel
xml	text/xml
xpm	image/x-xpixmap
xwd	image/x-xwindowdump

File Extension	MIME Type
xyz	chemical/x-pdb
zip	application/zip
Default if there is no match	application/octet-stream

Tag usage examples

The best way to learn about CFX_iMSMail is by example. This section will outline several mail sending examples, from basic to advanced. When testing the tag there are several things to keep in mind:

1 – It is often helpful to add a CFDUMP tag to the test template in order to see all of the information that is returned by the CFX tag. For example:

```
<cfdump var="#variables#">
```

2 – It is often helpful to see the control files (these are plain text INI files that contain the delivery information, tokens, recipients, etc.) and the generated email file that the CFX creates. Designate a folder in which to create these files by using the *SPOOLDIR* parameter.

The examples on the following pages illustrate the basic concepts of using CFX_iMSMail.

Basic Example

In this example we're sending mail to someone who has signed up for our web site. The user has submitted a form. One of the form fields is the email address of the user so we use that information to send the email.

```
<cfx_iMSMail
smtpo="#form.email#"
header_to="#form.email#"
header_subject="Thanks for registering!"
header_from="admin@example.com"
body="Thanks for joining our web site! Please visit
www.example.com/confirm.cfm?id=12345 in order to confirm your registration."
html="<html><body bgcolor=""blue""><font face=""arial"" color=""white"">Thanks
for joining our web site!</font><p> Please visit <a
href=""http://www.example.com/confirm.cfm?id=12345"">http://www.example.co
m/confirm.cfm?id=12345</a> in order to confirm your
registration.</body></html>"
>
```

Basic Example With Personalization

The first example is good for sending a quick email to a single recipient. This example will show you how to send personalized mail to one or more recipients. All of the recipient information is contained in a database and the database includes the email address as well as some personal information about each recipient. In this basic example we'll personalize the email with the email address of the recipient, their account number and their name.

```
<cfquery name="recipients" datasource="list">
  select email,name, account from customers
</cfquery>
```

```
<cfx_iMSMail
query="recipients"
header_to=""<:name:>"" <:smtpto:>
header_subject="Monthly specials"
header_from=""ACME Sales"" <sales@example.com>
header_x_account=""<:account:>""
body="Dear <:name:>,
Please visit www.example.com for our monthly specials."
html=""<html><body bgcolor=""blue""><font face=""arial"" color=""white"">Dear
<:name:>,</font><p> Please visit <a
href=""http://www.example.com/"">http://www.example.com/</a> for our monthly
specials.</body></html>""
>
```

This example uses the QUERY parameter instead of the SMTPTO parameter in the previous example. This allows us to send customized mail from a database query. The tag automatically creates tokens from the query that can be used in your code (tokens are delimited markers that are replaced with actual values).

Notes

Since our query has a column called "email", we don't need to use the *QUERYFIELD* parameter.

This example is using a predefined token (SMTPTO) that is always equivalent to the email address of the recipient.

All of the tokens are delimited with the default start and end token tags of "<:" and ":>". Your iMS server may be configured to use different delimiters so, to be safe, you may want to specify the token delimiters with the TOKENSTART and TOKENEND parameters.

Qtokens Example (specifying tokens)

Older versions of the CFX_iMSMail tag required that you specify the name of each column in the query that you wanted to use as a Qtoken. Also, you may not want to use all of the columns of your query (to save on the disk space required to store the token definitions) by setting the AUTOTOKENS parameter to "NO".

The following example illustrates this principle.

```
<cfquery name="test" datasource="imsqueue">
  select * from email
</cfquery>
<cfx_imsmail query="test" queryfield="email" qtoken_name="name"
Autotokens="NO"
qtoken_address="address"
header_to="recipients" header_from="The Big Kahuna"
header_subject="Big Kahuna Burgers - the planet's best burgers!"
body="Dear <:name:>,"
```

We now deliver Big Kahuna Burgers to <:address:>! Try one today!"

>

The above code performs a query of the recipient information and then calls the cfx tag specifying the fields from the query with which to build the dynamic tokens. QToken_name="name" tells the tag to get the token called "name" from a query column called "name." The second QToken parameter fills in the recipient's address.

Version 3.3.2 (and higher) of the tag also allows the creation of dynamic mail without specifying *Qtokens* or the *QueryField* parameters. The column in the query that contains the email address should be called "Email" but, if not, the *QueryField* parameter can be used to tell the CFX which column to use. The example above could be rewritten as:

```
<cfquery name="test" datasource="imsqueue">
  select * from email
</cfquery>
```

```
<cfx_imsmail query="test"
header_to="recipients" header_from="The Big Kahuna"
header_subject="Big Kahuna Burgers - the planet's best burgers!"
body="Dear <:name:>,"
```

```
We now deliver Big Kahuna Burgers to <:address:>! Try one today!"
>
```

Dynamic HTML content example

CFX_iMSMail allows you to insert embedded content into the generated html. This allows you to include images that are embedded into the rendered html instead of inserting links to content.

Example:

```
<cfx_iMSMail smtpTo="joe@example.com" header_to="joe@example.com"
header_subject="test" header_from="james@example.com"
html="<html><body bgcolor=blue><font face=arial color=white>this is a dynamic
test!</font><p>
Here is an image:<p>
<img src=""<imscid=e:\configservices.jpg|image/jpeg>""><p>
</body></html>"
>
```

Notice that in the example above we have a new tag called “iMSCID” which is the tag that causes the cfx to insert the file as an embedded attachment. The example shows that a file called “configservices.jpg” is being inserted and that the content type is “image/jpeg.”

Note that the CFX tag will automatically set the correct MIME type for the most popular file types if omitted.

Version 4.5 enhancement

Version 4.5 of the CFX tag also allows you to include attachments from ColdFusion variables instead of disk files. Due to a limitation of the CFX API the value of the variable must be included as an attribute in the tag and must also be Base64 encoded if the value is binary.

Example:

```
<cffile action="readbinary" file=" e:\configservices.jpg " variable="myimage">

<cfx_iMSMail smtpTo="joe@example.com" header_to="joe@example.com"
header_subject="test" header_from="james@example.com"
myimage=#ToBase64(myimage)#
html="<html><body bgcolor=blue><font face=arial color=white>this is a dynamic
test!</font><p>
Here is an image:<p>
<img src=""<imscid=var:myimage>""><p>
</body></html>"
>
```

Directly Querying an ODBC database

The `CFX_iMSMail` allows you to query an ODBC datasource directly using the `ODBCQUERY` parameter. Both the `DATASOURCE` and `ODBCQUERY` parameters are required. The `USERNAME` and `PASSWORD` parameters are required unless they have been predefined with the ODBC Credentials editor.

```
<cfx_imsmail datasource="postqueue" username="sa" password="xxx"  
odbcquery="select email,name,address from emails" odbcqueryfield="email"  
header_to="list member" header_from="list@coolfusion.com"  
header_subject="test"  
body="this is a test">
```

In this example we are querying from a table called "emails" in the specified datasource "postqueue." The tag will automatically produce tokens from the columns returned by the query.

Querying recipients and exclusions from an ODBC database

In addition to querying an ODBC datasource directly for recipients, it is also possible to query a datasource for a list of email addresses to exclude from a mailing. This is a convenient way of utilizing a list of opt-out addresses that should not receive mail. In this example we're keeping a list of opt-out addresses in a table called EXEMAILS.

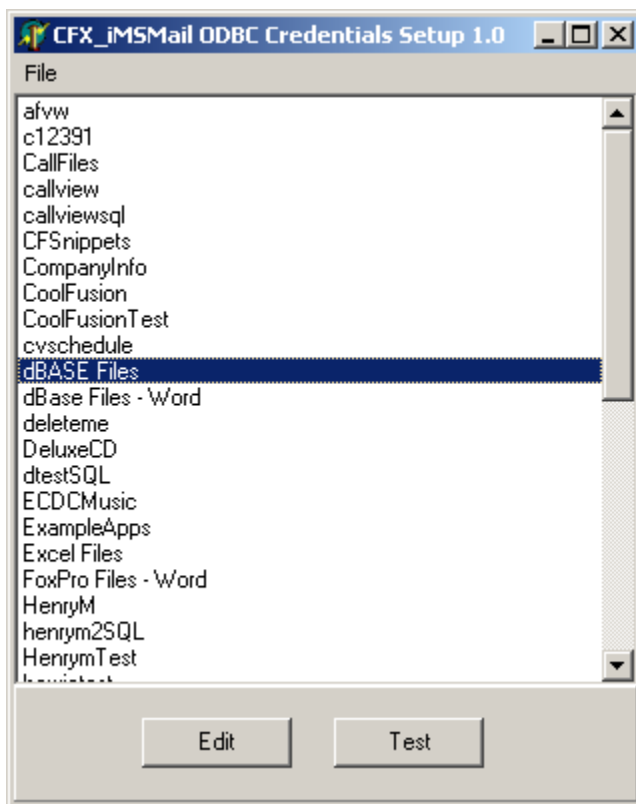
```
<cfx_imsml datasource="postqueue" username="sa" password="xxx"  
odbcquery="select email,name,address from emails" odbcqueryfield="email"  
exodbcquery="select email from exemails" exodbcqueryfield="email"  
header_to="list member" header_from="list@coolfusion.com"  
header_subject="test"  
body="this is a test">
```

In this example we are querying from a table called "emails" in the specified datasource "postqueue." The tag will automatically produce tokens from the columns returned by the query. Any emails returned by the exclusions query will not receive the mail.

Preconfiguring Datasources

In order to provide additional security when directly querying datasources for email recipients or exclusions, the CFX tag has the ability to read the datasource credentials from saved values store in the system registry. The user name and password are stored in the registry using a lightweight encryption that can be quickly decoded by the CFX tag.

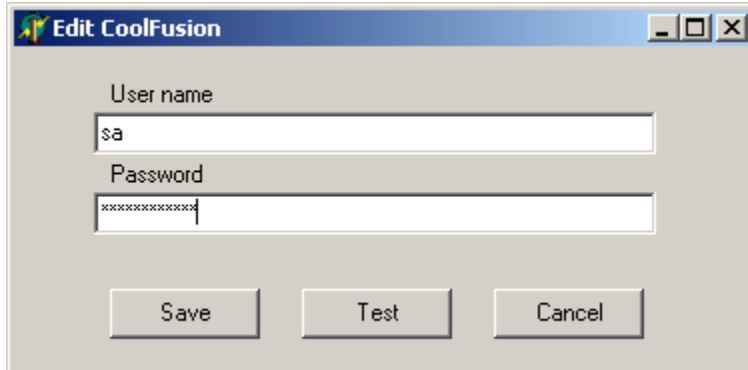
The datasource credentials are edited using the included ODBC Credentials editor. The credentials editor is accessed from the iMS program group (or the CFX_iMSMail program group if you installed the CFX separately). The credentials editor has a simple interface as shown below:



The listbox lists all of the datasources on the system. Use the *Edit* or *Test* buttons to edit the credentials or test the saved credentials assigned to the highlighted datasource.

Editing a Datasource

To edit the credentials of a datasource simply highlight the datasource and click on the Edit button.

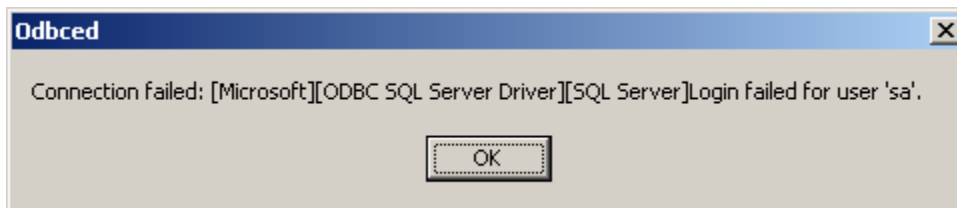


The edit box will display the existing credentials. Another test button is included on the edit box so that you can test the datasource credentials prior to saving.

If the datasource credentials are configured properly the following popup will appear when the *Test* button is clicked.



If the credentials are not correct, the following will appear:



If the datasource is not configured properly an error, such as the one below, will be displayed:



Once you have finished editing your datasource credentials click on the *Save* button and the credentials will be encrypted and stored in the system registry.