



EasyLink Messaging API

API Basic User Guide

A Guide for Beginning Users of the EasyLink

Messaging API.

Last Modified: 9/3/2013

V1.2 – MessagingAPI 2.6

Table of Contents

Table of Contents	2
Overview	5
Introduction	5
What is SOAP?	5
Namespaces	5
Building Request Headers	6
ReceiverKey	6
Authentication	6
Sample Request Header	7
Reading Response Headers	8
Sample Response Header	8
Working with Destination Lists	9
Upload a List	10
Building the Request.....	10
ListContents	10
Sample Request.....	11
Reading the Response.....	11
Status	11
ListId.....	12
Sample Response.....	12
Next Actions	12
Review Stored Lists	13
Building the Request.....	13
Sample Request	13
Reading the Response.....	13
Status	13
ListInfo.....	14
Sample Response.....	14
Next Actions	15
Retrieve a List	16
Building the Request.....	16
Sample Request	16
Reading the Response.....	17

Status	17
ListId	17
Table	17
Sample Response	18
Next Actions	18
Sending a Message	19
Send an Email Message	20
Building the Request	20
JobOptions	21
Destinations	21
Reports	22
<i>Contents</i>	22
Sample Request	22
Reading the Response	23
Next Actions	23
Send a Fax Message	24
Building the Request	24
JobOptions	24
Destinations	25
Reports	25
<i>Contents</i>	25
Sample Request	26
Reading the Response	27
Next Actions	27
Send an SMS Message	28
Building the Request	28
JobOptions	28
Destinations	29
Reports	29
<i>Contents</i>	29
Sample Request	30
Reading the Response	30
Next Actions	31
Send a Voice Message	32
Building the Request	32

JobOptions	33
Destinations.....	33
Reports	34
Contents	34
Sample Request	34
Reading the Response	35
Next Actions	35
Read the JobSubmitResult Element	36
Reading the Response	36
Status	36
MessageResult.....	36
Sample Response	37
Reviewing Jobs	38
Checking Delivery Status	39
Building the Request.....	39
Sample Request	39
Reading the Response	39
JobData.....	40
AttachmentInfo (email Only)	42
DeliveryDetail	42
Sample Response	44
Exclusive States	48
Checking a Job Summary	49
Building the Request.....	49
Sample Request	49
Reading the Response	49
JobSummaryData.....	50
AttachmentInfo (email Only)	52
Sample Response	53

Overview

Introduction

The EasyLink Messaging API allows EasyLink's customers to develop messaging applications that communicate with the EasyLink Messaging infrastructure to deliver Email, SMS, Voice, and Fax messages. This help file provides instructions for performing the most common tasks using the EasyLink Messaging framework, such as specifying recipients, sending messages, and checking delivery status.

All EasyLink Messaging messages are composed of a SOAP header and an XML-based request element. An application that uses the EasyLink Messaging API must be able to generate the SOAP messages expected by the EasyLink Messaging infrastructure and read the responses it provides. This help file walks the developer through the following tasks:

1. [Building Request Headers](#)
2. [Reading Response Headers](#)
3. [Working with Destination Lists](#)
4. [Sending Messages](#)
5. [Reviewing Jobs](#)

This help file provides an introduction to common tasks performed using the API. Please consult the latest [EasyLink Messaging API Manual](#) for complete documentation of API.

What is SOAP?

Simple Object Access Protocol (SOAP) is designed for exchanging web-based services over a network. In SOAP, a client requests a service by sending an XML request that includes important data structures. The type of action performed by the server depends on the service requested as well as the data provided. In all cases, the server replies to the client with an XML response.

This help file assumes that you have a working knowledge of SOAP. Refer to <http://www.w3.org/TR/soap/> for more information.

Namespaces

Each XML request must specify at least one namespace. The namespace identifies the formats being used in the message. The namespace will also depend on the specific request and response being processed. For example, a JobSubmit request for version 2.6 will utilize the namespace <http://ws.easylink.com/JobSubmit/2011/01>.

Building Request Headers

All SOAP request envelopes must include a request header that contains user authentication information and the URL for the response location. This topic describes how to develop a SOAP request header for use with the EasyLink Messaging API.

Within the SOAP envelope, include a **Header** element that contains the **Request** element. Normally this element has its own namespace. For example, the API 2.6 version request element utilizes the namespace <http://ws.easylink.com/RequestResponse/2011/01>. See example below.

See the individual EasyLink Messaging API Manuals for complete details about the Request element.

The **Request** element must contain a **ReceiverKey** element and an **Authentication** element.

ReceiverKey

The ReceiverKey element indicates the URL to which the request will be sent. It does not have any impact on the API processing, but we do advise ensuring that it matches the actual submission URL used to send the request.

Authentication

Authentication via the API can be performed using standard user name and password style credentials, or through the use of an X.509 certificate using [WS-Security](#) (Web Services Security). For details on using WS-Security with the API see [User Credentials and WS-Security](#) on the API manuals site.

To use password-based authentication, the **Authentication** element must include the **XDDSAuth** element. If you have a User ID and Password, you can populate the **XDDSAuth** element as follows:

Element	Description	Sub-elements
RequesterID	The User ID of the user sending the request.	None.
Password	Password associated with the RequesterID.	None.

Sample Request Header

The following sample request header authenticates **UserOne** with password **abc123** and indicates that the SOAP request was sent to **http://xxx/soap/xxx**.

```
<soapenv:Header>
  <ns1:Request
xmlns:ns1="http://ws.easylink.com/RequestResponse/2011/01">
    <ns1:ReceiverKey>http://xxx/soap/xxx</ns1:ReceiverKey>
    <ns1:Authentication>
      <ns1:XDDSAuth>
        <ns1:RequesterID>UserOne</ns1:RequesterID>
        <ns1:Password>abc123</ns1:Password>
      </ns1:XDDSAuth>
    </ns1:Authentication>
  </ns1:Request>
</soapenv:Header>
```

Reading Response Headers

All SOAP response envelopes will include a response header that provides the EasyLink-specific Processing ID for the request. This topic describes how to read a SOAP response header from the EasyLink Messaging infrastructure.

The SOAP envelope includes a **Header** element that contains the **Response** element.

See *the individual EasyLink Messaging API Manuals* for complete details about the *Request* element.

At a minimum, read the following elements in the **Response** element:

Element	Description	Sub-elements
ProcessingID	The EasyLink-generated processing ID for the request.	None.

Sample Response Header

The following sample response header provides the URI from a **JobSubmitRequest** and the EasyLink-generated **ProcessingID** associated with the request.

```
<soapenv:Header>
  <Response xmlns="http://ws.easylink.com/RequestResponse/2011/01">
    <SenderKey>http://messaging.easylink.com/2_6/JobSubmit</Senderkey>
    <ProcessingID>89EC01B8-1009-071126174657782</ProcessingID>
  </Response>
</soapenv:Header>
```

There also may be a <SenderKey> element, which gives two pieces of important information:

- The Web Server URL where the request was processed. In this example - http://messaging.easylink.com/2_6/JobSubmit
- The version of the API running on the web server. In this example – version 2.6.

Working with Destination Lists

EasyLink Messaging applications send messages to lists of recipients called destination lists. Specifying message destinations is one of the first actions an EasyLink Messaging application should perform. Once a destination list is uploaded, the EasyLink Messaging application can reference that list to send messages. In general, an EasyLink Messaging application should be able to do the following:

- [Upload a List](#)
- [Retrieve a List](#)
- [Review Stored Lists](#)

While storing lists is a useful way to maintain a collection of destinations, you are not required to use stored lists to send messages. You can send a list of destinations with any request to send a message. You may prefer to send the destination list with the message if your destination lists change significantly from message to message.

*Refer to the [Common Types and Elements](#) section in the *EasyLink Messaging API Manual* for more information about lists.*

Upload a List

This topic describes how a EasyLink Messaging application uploads a list of recipients for a message. After storing the list, you can use it in the Destination element of future message requests.

Building the Request

Upload a destination list by wrapping a **ListCreateRequest** element in a SOAP envelope. Be sure to include a [SOAP request header](#).

Refer to the [ListCreate](#) section in the *EasyLink Messaging API Manual* for complete details about the *ListCreateRequest* element.

At a minimum, include the following elements in the **ListCreateRequest** element:

Element	Description	Sub-elements
ListId	Identifies the stored list to create. Attributes include: liststore - Indicates which service contains the list. Valid values are: <i>XDDS</i> - voice and fax services <i>MR</i> - enhanced email service ownership - Indicates the ownership level. Valid values are: <i>user</i> - belongs to the user that created the list <i>customer</i> - any user belonging to the customer account can access the item	None.
ListContents	Specifies the contents of the list.	Table - Specifies a list of destinations in Comma-Separated Value (CSV) or Excel (XLS) format.
CreateOption	Identifies the method used to create the list. Valid values are: <i>CreateOnly</i> - Operation fails if the ListId is in use. <i>CreateOrReplace</i> - Operation replaces a duplicate ListId. <i>ReplaceOnly</i> - Operation fails if the ListId does not exist.	None.

ListContents

The **ListContents** element should contain one **Table** element, which allows you to use a CSV or XLS file to specify destinations. At a minimum, include the following element in the **Table** element:

Element	Description	Sub-elements
Document	Includes document location	DocType - The type of document sent.

and encoding information.

For uploading lists, specify *text*.

Filename- The file that contains the list of destinations.

URL- The URL location of the specified file.

CharacterSet- The character set used by the message. Most list uploads use *ISO-8859-1*.

Sample Request

The following sample request uploads the contents of the **ISO-8859-1** encoded **fax_lst1.txt** file located at **https://xxx/fax_lst1.txt** to the EasyLink Messaging infrastructure as **fax_lst1**. If **fax_lst1** already exists, it will be overwritten because **CreateOrReplace** is specified.

```
<ListCreateRequest xmlns="
http://ws.easylink.com/ListCreate/2011/01">
  <ListId liststore="XDDS" ownership="customer">fax_lst1</ListId>
  <ListContents>
    <Table>
      <Document>
        <DocType>text</DocType>
        <Filename>fax_lst1.txt</Filename>
        <DocUrl>http://xxx/fax_lst1.txt</DocUrl>
        <CharacterSet>ISO-8859-1</CharacterSet>
      </Document>
    </Table>
  </ListContents>
  <CreateOption>CreateOrReplace</CreateOption>
</ListCreateRequest>
```

Reading the Response

After processing the request, the EasyLink Messaging infrastructure sends a response. The response to a ListCreate request includes a [SOAP header](#) and a **ListCreateResult** element in a SOAP envelope.

See *ListCreate* section of the *EasyLink Messaging API Manual* for complete details about the *ListCreateResult* element.

The **ListCreateResult** element includes a **Status** element for the request and a **ListId** element that identifies the created list.

Status

The contents of the **Status** element define success or failure of the request message as a whole. At a minimum, read the following elements from the **Status** element:

Element	Description	Sub-elements
StatusCode	Indicates the success or failure of the request. Valid values are: 0- Success 1- Failure 2- Partial success	None.

StatusMessage	Describes the StatusCode value for convenience. Do not rely on this field to indicate success or failure.	None.
SubmissionTime	Time the request message was received.	None.
CompletionTime	Time the request message was processed.	None.

See *ListCreate* section of the *EasyLink Messaging API Manual* for detailed information about the *Status* element.

ListId

The **ListId** element returns the identifying characteristics of the newly created list. The following table describes the ListId element:

Element	Description	Sub-elements
ListId	Identifies the stored list. Attributes include: liststore - Indicates which service contains the list. Valid values are: <i>XDDS</i> - voice and fax services <i>MR</i> - enhanced email service ownership - Indicates the ownership level (<i>customer</i> , <i>user</i>) of the list.	None.

Sample Response

The sample response below includes a **Status** for the **ListCreateRequest** message. The response indicates success (status code **0**) and returns the requested **ListId** information for reference.

```
<ListCreateResult xmlns="
http://ws.easylink.com/ListCreate/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-11-26T18:15:07.889Z</SubmissionTime>
    <CompletionTime>2009-11-26T18:15:09.610Z</CompletionTime>
  </Status>
  <ListId ownership="customer" liststore="XDDS">#fax_lst1</ListId>
</ListCreateResult>
```

Next Actions

If the response indicates success, your list is uploaded and ready to be used for [sending](#) messages.

If the response indicates failure, you may want to resend the list in a new request.

Review Stored Lists

This topic describes how to use the EasyLink Messaging API to review the lists stored on the EasyLink Messaging infrastructure. An EasyLink Messaging application might send this request as part of a recipient-management feature that allowed users to review a set of destination lists.

Building the Request

Upload a destination list by wrapping a **ListListsRequest** element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See [ListLists](#) of the *EasyLink Messaging API Manual* for complete details about the *ListListsRequest* element.

At a minimum, include the following element in the **ListListsRequest** element:

Element	Description	Sub-elements
Ownership	Indicates the ownership level. Valid values are: <i>user</i> - belongs to the user that created the list <i>customer</i> - any user belonging to the customer account can access the item	None.

Sample Request

The following sample request the lists visible at the **customer** ownership level.

```
<ListListsRequest xmlns=" http://ws.easylink.com/ListLists/2011/01 ">  
  <Ownership>customer</Ownership>  
</ListListsRequest>
```

Reading the Response

Once the request is received and processed, the EasyLink Messaging infrastructure sends a response. The response to a List Lists request includes a [SOAP response header](#) and a **ListListsResult** element in a SOAP envelope.

The **ListListsResult** element includes a **Status** element for the entire request and a **ListInfo** element for each list visible to the requested ownership level.

Status

The contents of the **Status** element define success or failure of the request message as a whole. At a minimum, read the following elements from the **Status** element:

Element	Description	Sub-elements
StatusCode	Indicates the success or failure of the request. Valid values	None.

	are: 0 - Success 1 - Failure 2 - Partial success	
StatusMessage	Describes the StatusCode value for convenience. Do not rely on this field to indicate success or failure.	None.
SubmissionTime	Time the request message was received.	None.
CompletionTime	Time the request message was processed.	None.

ListInfo

The contents of the **ListInfo** element provide the information needed to reference the list as well as the time the list was last modified. At a minimum, read the following elements from the **ListInfo** element:

Element	Description	Sub-elements
ListId	Identifies the stored list. Attributes include: liststore - Indicates which service contains the list. Valid values are: <i>XDDS</i> - voice and fax services <i>MR</i> - enhanced email service ownership - Indicates the ownership level (<i>customer or user</i>) of the list.	None.
ListModified	Indicates the time the list was last modified.	None.

Sample Response

The sample response below includes a **Status** for the **ListListsRequest** message. The response indicates success (status code **0**) and returns **ListInfo** elements for each list at the **user** ownership level.

```
<ListListsResult>
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-11-26T17:46:57.782Z</SubmissionTime>
    <CompletionTime>2009-11-26T17:46:59.725Z</CompletionTime>
  </Status>
  <ListInfo>
    <ListId ownership="user" liststore="MR">shilpa1</ListId>
    <ListModified>2008-09-14T21:25:27.000Z</ListModified>
  </ListInfo>
  <ListInfo>
    <ListId ownership="user" liststore="MR">jeannew2</ListId>
    <ListModified>2008-09-14T21:25:32.000Z</ListModified>
  </ListInfo>
</ListListsResult>
```

```
</ListInfo>
<ListInfo>
  <ListId ownership="user" liststore="MR">upload</ListId>
  <ListModified>2008-09-14T21:25:39.000Z</ListModified>
</ListInfo>
<ListInfo>
  <ListId ownership="user" liststore="MR">shilp11</ListId>
  <ListModified>2008-09-14T21:25:51.000Z</ListModified>
</ListInfo>
</ListListsResult>
```

Next Actions

If the response indicates success, you can use any of the returned lists for [sending](#) messages. Or you can use one of the returned List IDs to [retrieve a list](#).

If the response indicates failure, you may want to resend the request.

Retrieve a List

This topic describes how to use the EasyLink Messaging API to retrieve a list stored on the EasyLink Messaging infrastructure. An EasyLink Messaging application might use this request to display the detailed contents of a particular destination list.

Building the Request

Retrieve a destination list by wrapping a **ListRetrieveTableRequest** element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See [ListRetrieveTable](#) section of the *EasyLink Messaging API Manual* for complete details about the *ListRetrieveTableRequest* element.

At a minimum, include the following element in the **ListRetrieveTableRequest** element:

Element	Description	Sub-elements
SubmitId	Optional string to describe the request. Will be echoed in the response.	None.
ListId	Identifies the stored list to retrieve. Attributes include: liststore - Indicates which service contains the list. Valid values are: <i>XDDS</i> - voice and fax services <i>MR</i> - enhanced email service ownership - Indicates the ownership level. Valid values are: <i>user</i> - belongs to the user that created the list <i>customer</i> - any user belonging to the customer account can access the item	None.

Sample Request

The following sample request asks for the contents of the **testlist2** list, which is defined at the **user** level of the **XDDS** (voice and fax) list store. It identifies the request as **listretrieve 029**.

```
<ListRetrieveTableRequest>  
<SubmitId>listretrieve 029</ns1:SubmitId>  
<ListId ownership="user" liststore="XDDS">testlist2</ListId>  
</ListRetrieveTableRequest>
```


Reading the Response

Once the request is received and processed, the EasyLink Messaging infrastructure sends a response. The response to a List Retrieve Tables request includes a [SOAP response header](#) and a **ListRetrieveTableResult** element in a SOAP envelope.

The **ListRetrieveTableResult** element includes a **Status** element for the entire request, a **ListId** element to identify the returned list, and a **Table** element that provides the list contents.

Status

The contents of the **Status** element define success or failure of the request message as a whole. At a minimum, read the following elements from the **Status** element:

Element	Description	Sub-elements
StatusCode	Indicates the success or failure of the request. Valid values are: 0- Success 1- Failure 2- Partial success	None.
StatusMessage	Describes the StatusCode value for convenience. Do not rely on this field to indicate success or failure.	None.
SubmissionTime	Time the request message was received.	None.
CompletionTime	Time the request message was processed.	None.

ListId

The contents of the **ListId** element echo the ListId element in the request.

Table

The contents of the requested list are returned as a single table of data as comma-separated values in the encoding specified in the request message. At a minimum, read the following elements from the **Table** element:

Element	Description	Sub-elements
Document	Includes document location and encoding information. This element requires the following attribute: <i>ref</i> -The name of the document. This name is used to reference the document when building the Message element below.	DocType - The type of document. Example values include: <i>text,HTML,GIF</i> , etc. Filename - The file that contains the contents of the message. This is the data the recipient will receive. DocData - The contents of the file. This element requires the following attribute: <i>format</i> -The format of the information

provided in the DocData element.

CharacterSet- The character set used in the document.

Sample Response

The sample response below includes a **Status** for the **ListRetrieveTableRequest** message, identifies the returned testlist2, and provides the **base64, UTF-8**-encoded CSV file that defines the members of the destination list.

```
<ListRetrieveTableResult
xmlns="http://ws.easylink.com/ListLists/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <SubmissionTime>2007-05-16T19:19:22.300Z</SubmissionTime>
    <CompletionTime>2007-05-16T19:19:23.502Z</CompletionTime>
  </Status>
  <ListId liststore="XDDS" ownership="user">testlist2</ListId>
  <Table>
    <Document>
      <DocType>text</DocType>
      <DocData format="base64">IkFERFI ... idm9pY2Ui</DocData>
      <CharacterSet>UTF-8</CharacterSet>
    </Document>
  </Table>
</ListRetrieveTableResult>
```

Next Actions

An EasyLink Messaging application that retrieves a list usually does so at the request of the user. The application should read the list and determine what information to present.

If the response indicates failure, you may want to resend the retrieve request.

Sending a Message

Sending email, fax, SMS, and voice messages is the primary purpose of the EasyLink Messaging API. EasyLink Messaging treats all send requests as jobs, and EasyLink Messaging applications can request jobs using the [JobSubmit](#) element in a SOAP request. To request a specific type of job, refer to the following topics:

- [Send an Email Message](#)
- [Send a Fax Message](#)
- [Send an SMS Message](#)
- [Send a Voice Message](#)

The EasyLink Messaging infrastructure responds to each of these JobSubmit requests with a JobSubmitResult response that includes status information for the request. See [Read the JobSubmitResult Element](#) for instructions on interpreting the response.

Send an Email Message

This topic describes how to use the EasyLink Messaging API to send email messages. Before using these instructions, you must do the following:

- Prepare a CSV or XLS file that contains the destinations for the email, as well as additional information about each destination.

Building the Request

Send email messages by wrapping a `JobSubmitRequest` element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See [JobSubmit](#) section of the *EasyLink Messaging API Manual* for complete details about the `JobSubmitRequest` element.

The **JobSubmitRequest** must contain at least one of each of the following elements:

- **DocumentSet**- Contains file information for each of the documents used in the email message.
- **Message**- Contains all the information necessary to transmit the email message.

At a minimum, include the following element in the **DocumentSet** element:

Element	Description	Sub-elements
Document	Includes document location and encoding information. This element requires the following attribute: <i>ref</i> -The name of the document. This name is used to reference the document when building the Message element below. <i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i>	DocType - The type of document. Example values include: <i>text</i> , <i>HTML</i> , <i>GIF</i> , etc. Filename - The name of the specified document. This name is used when sending the file to the recipient(s). DocData - The contents of the file. This element requires the following attribute: format -The format of the information provided in the DocData element. CharacterSet - The character set used in the document.

At a minimum, include the following elements in the **Message** element:

Element	Description	Sub-elements
JobOptions	Includes delivery and message-specific options.	Delivery - Specifies priority and scheduling for the job. EnhancedEmailOptions - Includes email-specific job options.
Destinations	Specifies the email message recipients <i>See Common Types and Elements section of the</i>	Table - Defines a set of destinations using a properly formatted file.

	<i>EasyLink Messaging API Manual for more information.</i>	
Reports	Identifies which reports should be returned to the sender by the messaging infrastructure.	Sub-elements depend on the desired report(s).
Contents	Includes the contents of the fax message.	Part - Includes the message to be sent and treatment options for the message.

JobOptions

The contents of the **JobOptions** element define the characteristics of the message. Here are some sub-elements you might wish to include in **JobOptions**:

Element	Description	Key Sub-elements
BillingCode	Job-specific billing code specified by your messaging application, intended for use in invoice preparation.	None.
CustomerReference	Customer Reference code specified by your messaging application, intended for use as the customer reference number for reporting and results messages.	None.
EnhancedEmailOptions	Includes email-specific job options. If you are initiating an email job, you must include this element, even if its contents are empty.	Subject - The email subject. When a Table is used with inserts to specify recipient names, use parenthesis to reference an insert value. For example: "Hello, (1)" FromDisplayName - The name to display as the From address. HTMLOpenTracking - Indicates the type of receipt tracking to use for HTML messages. Valid values are: <i>top</i> - Places the tracking element at the top of the message. <i>bottom</i> - Places the tracking element at the end of the message. <i>none</i> - Do not use a tracking element.

Destinations

The contents of the **Destinations** element define the recipients of the message. When using the **Table** mechanism to specify your destinations, include the following:

Element	Description	Sub-elements
Table	Defines a set of destinations using a properly formatted file. By including a CSV or XLS document in the DocumentSet , you can send email to multiple recipients. <i>ref</i> - The name of the table. This name is used	DocRef - Name of the document to use. This name corresponds to the <i>ref</i> value of a Document element in the DocumentSet .

to reference the table elsewhere in the **Message** element.

Reports

The contents of the **Reports** element define what message-related reports to send, and where. For most messages, you should request a delivery report. To do so, include the following element in the **Reports** element:

Element	Description	Sub-elements
DeliveryReport	Requests a delivery report.	DeliveryReportType - Type of delivery report requested. Specify <i>detail</i> for the most information.

Contents

The **Contents** element must contain *at least one* **Part** element. The **Part** element allows you to select a document to include in the message and specify what part of the message the document should fill. At a minimum, include the following elements for each **Part** element:

Element	Description	Sub-elements
DocRef	The name of the email document to send. This should match the document defined above in the DocumentSet element.	None.
Treatment	Specifies what part of the message the document should be treated as. Valid values are: <i>body</i> - Email body. <i>attachment</i> - Sends the document as an attachment with the email. <i>pullfile</i> - Stores the document on the EasyLink Messaging infrastructure and links to the document in the email.	None.

Sample Request

The following sample request sends the contents provided in the **email_text** and **email_html_text** documents to the destinations provided in the **docREF_CSV** document and requests a **Detail** delivery report.

```
<JobSubmitRequest
xmlns="http://ws.easylink.com/JobSubmit/2011/01">
  <DocumentSet>
    <Document ref="email_text">
      <DocType>text</DocType>
      <DocData format="text">Hi (i1) (i2), This is a generic message
please reply
      </DocData>
    </Document>
    <Document ref="email_html_text">
      <DocType>HTML</DocType>
      <DocData format="base64">PGJvZH ... vYm9keT4=</DocData>
```

```

</Document>
<Document ref="docREF_CSV">
  <DocType>text</DocType>
  <Filename>email_test.csv</Filename>
  <DocData format="base64">QUREUixU ... GJveAOK</DocData>
</Document>
</DocumentSet>
<Message>
  <JobOptions>
    <BillingCode>Enhanced Email Job</BillingCode>
    <CustomerReference>Enhanced Email Job</CustomerReference>
    <EnhancedEmailOptions>
      <Subject>Hello (il) EasyLink Messaging Test</Subject>
      <FromDisplayName>EasyLink</FromDisplayName>
      <ReplyTo>apitest@easylink.com</ReplyTo>
      <HTMLOpenTracking>top</HTMLOpenTracking>
    </EnhancedEmailOptions>
  </JobOptions>
  <Destinations>
    <Table ref="tblREF_CSV">
      <DocRef>docREF_CSV</DocRef>
    </Table>
  </Destinations>
  <Reports>
    <DeliveryReport>
      <DeliveryReportType>detail</DeliveryReportType>
    </DeliveryReport>
  </Reports>
  <Contents>
    <Part>
      <DocRef>email_text</DocRef>
      <Treatment>body</Treatment>
    </Part>
    <Part>
      <DocRef>email_html_text</DocRef>
      <Treatment>body</Treatment>
    </Part>
  </Contents>
</Message>
</JobSubmitRequest>

```

Reading the Response

After processing the request, the EasyLink Messaging infrastructure sends a response. Refer to the [JobSubmit Response](#) topic for instructions on reading this response.

Next Actions

If the response indicates success, your message was submitted correctly. Successful delivery depends on a number of factors.

If the response indicates failure, you may want to resend the message in a new request or check the status of the message.

See [Checking Delivery Status](#) for further instructions.

Send a Fax Message

This topic describes how to use the EasyLink Messaging API to send fax messages. Before using these instructions, you must do the following:

- [Upload a list](#) of recipients to the EasyLink Messaging infrastructure.
- Store the desired contents of the fax message at a public URL accessible by the EasyLink Messaging infrastructure.

Once the fax message is ready and the list of recipients is uploaded, you are ready to send a fax message request.

Building the Request

Send fax messages by wrapping a JobSubmitRequest element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See *JobSubmit* section the *EasyLink Messaging API Manual* for complete details about the *JobSubmitRequest* element.

The **JobSubmitRequest** element must contain *at least one Message* element. The Message element contains all the information necessary to transmit the fax message. At a minimum, include the following elements in the **Message** element:

Element	Description	Sub-elements
JobOptions	Includes delivery and message-specific options.	Delivery - Specifies priority and scheduling for the job. Fax Options - Includes fax-specific job options.
Destinations	Specifies the fax message recipient <i>See Common Types and Element section of the EasyLink Messaging API Manual for more information.</i>	List - Specifies a list of destinations already stored on the EasyLink Messaging infrastructure.
Reports	Identifies which reports should be returned to the sender by the messaging infrastructure.	Sub-elements depend on the desired report(s).
Contents	Includes the contents of the fax message.	Part - Includes the message to be sent and treatment options for the message.

JobOptions

The contents of the **JobOptions** element define the characteristics of the message. Here are some sub-elements you might wish to include in **JobOptions**:

Element	Description	Sub-elements
Delivery	Includes delivery and	Schedule - Indicates when the message is to be delivered. To deliver as soon as

	scheduling options.	possible, specify <i>express</i> .
FaxOptions	Includes fax-specific job options. When sending a fax job, you must include this element, even if its contents are empty. Be sure to include only one modality-specific element in your input.	FaxCoversheet - Includes the <i>UseCoversheet</i> , <i>CoversheetTo</i> , and <i>CoversheetAttn</i> elements that define how and if a cover sheet is appended to the fax. Include these three elements and specify <i>no</i> in the <i>UseCoversheet</i> element. Leave the remaining elements empty. BannerFX - Includes the <i>UseBannerFX</i> element that defines if the BannerFX features of XDDS will be used. Include this element and specify <i>no</i> . FaxMode - Indicates the vertical image resolution for the fax. Valid values are: <i>standard</i> - Standard resolution. <i>fine</i> - High resolution.

Destinations

The contents of the **Destinations** element define the recipients of the message. At a minimum, include the following element in the **Destinations** element:

Element	Description	Sub-elements
List	A list of destinations stored on the EasyLink Messaging infrastructure by a previous ListCreate request .	Name - Name of the destinations list to use.

See *Common Types and Elements* section of the *EasyLink Messaging API Manual* for detailed information about the *Destinations* element, which is of the *DeliveryItem* type.

Reports

The contents of the **Reports** element define what message-related reports to send, and where. For most messages, you should request a delivery report. To do so, include the following element in the **Reports** element:

Element	Description	Sub-elements
DeliveryReport	Requests a delivery report.	DeliveryReportType - Type of delivery report requested. Specify <i>detail</i> for the most information.

See *JobSubmit Overview* section of the *EasyLink Messaging API Manual* for more information about the *Reports* element.

Contents

The **Contents** element must contain *at least one Part* element. The **Part** element allows you to select a document to include in the message and specify what part of the message the document should fill. At a minimum, include the following elements in the **Part** element:

Element	Description	Sub-elements
Document	Includes document location and encoding information.	DocType - The type of document sent. For fax messages, specify <i>text</i> .

Note: The document URL must be public so that the EasyLink Messaging infrastructure can access the file.

Filename- The file that contains the contents of the message. This is the data the recipient will receive.

URL- The URL location of the specified file.

CharacterSet- The character set used by the message. Most fax messages use *UTF-8*.

Sample Request

The following sample request sends the contents of the **UTF-8**-encoded **coldel.txt** file located at <https://xxx/coldel.txt> as a fax to all recipients in the **#fax_tst1** list and requests a detailed delivery report. Because the specified delivery schedule is **express**, transmission of the message begins immediately.

```
<JobSubmitRequest
xmlns="http://ws.easylink.com/JobSubmit/2011/01">
  <Message>
    <JobOptions>
      <Delivery>
        <Schedule>express</Schedule>
      </Delivery>
      <FaxOptions>
        <FaxCoversheet>
          <UseCoversheet>no</UseCoversheet>
        </FaxCoversheet>
        <BannerFX>
          <UseBannerFx>no</UseBannerFx>
        </BannerFX>
        <FaxMode>standard</FaxMode>
        <DeliveryRetryPattern>1</DeliveryRetryPattern>
      </FaxOptions>
    </JobOptions>
    <Destinations>
      <List>
        <Name>#fax_lst1</Name>
      </List>
    </Destinations>
    <Reports>
      <DeliveryReport>
        <DeliveryReportType>detail</DeliveryReportType>
      </DeliveryReport>
    </Reports>
    <Contents>
      <Part>
        <Document>
          <DocType>text</DocType>
          <Filename>coldel.txt</Filename>
          <DocUrl>https://xxx/coldel.txt</DocUrl>
          <CharacterSet>UTF-8</CharacterSet>
        </Document>
      </Part>
    </Contents>
  </Message>
</JobSubmitRequest>
```

Reading the Response

After processing the request, the EasyLink Messaging infrastructure sends a response. Refer to the [JobSubmit Response](#) topic for instructions on reading this response.

Next Actions

If the response indicates success, your message was submitted correctly. Successful delivery depends on a number of factors.

If the response indicates failure, you may want to resend the message in a new request or check the status of the message.

See [Checking Delivery Status](#) for further instructions.

Send an SMS Message

This topic describes how to use the EasyLink Messaging API to send text (SMS) messages. Before using these instructions, you must do the following:

- [Upload a list](#) of recipients to the EasyLink Messaging infrastructure.
- Store the desired contents of the SMS message at a public URL accessible by the EasyLink Messaging infrastructure.

Once the SMS message is ready and the list of recipients is uploaded, you are ready to send an SMS message request.

Building the Request

Send SMS messages by wrapping a JobSubmitRequest element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See *JobSubmit* of the *EasyLink Messaging API Manual* for complete details about the *JobSubmitRequest* element.

The **JobSubmitRequest** element must contain *at least one Message* element. The Message element contains all the information necessary to transmit the SMS message. At a minimum, include the following elements in the **Message** element:

Element	Description	Sub-elements
JobOptions	Includes delivery and message-specific options.	Delivery - Specifies priority and scheduling for the job. SMSOptions - Includes SMS-specific job options.
Destinations	Specifies the SMS message recipient <i>See Common Types and Element section of the EasyLink Messaging API Manual for more information.</i>	List - Specifies a list of destinations already stored on the EasyLink Messaging infrastructure.
Reports	Identifies which reports should be returned to the sender by the messaging infrastructure.	Sub-elements depend on the desired report(s).
Contents	Includes the contents of the SMS message.	Part - Includes the message to be sent and treatment options for the message. SMS messages only need one Part.

JobOptions

The contents of the **JobOptions** element define the characteristics of the message. Here are some sub-elements you might wish to include in **JobOptions**:

Element	Description	Sub-elements
Delivery	Includes delivery and scheduling options.	Schedule - Indicates when the message is to be delivered. To deliver as soon as possible, specify <i>express</i> .
SMS Options	Includes SMS-specific job options. When sending an SMS job, you must include this element, even if its contents are empty. Be sure to include only one modality-specific element in your input.	ExpirationDays - Number of days before the message expires on the messaging infrastructure. CharacterSet - The character set used by the message. Most SMS messages use <i>UTF-8</i> .

Destinations

The contents of the **Destinations** element define the recipients of the message. At a minimum, include the following element in the **Destinations** element:

Element	Description	Sub-elements
List	A list of destinations stored on the EasyLink Messaging infrastructure by a previous ListCreate request .	Name - Name of the destinations list to use.

See *Common Types and Elements* section of the *EasyLink Messaging API Manual* for detailed information about the *Destinations* element, which is of the *DeliveryItem* type.

Reports

The contents of the **Reports** element define what message-related reports to send, and where. For most messages, you should request a delivery report. To do so, include the following element in the **Reports** element:

Element	Description	Sub-elements
DeliveryReport	Requests a delivery report.	DeliveryReportType - Type of delivery report requested. Specify <i>detail</i> for the most information.

See *JobSubmit* section of the *EasyLink Messaging API Manual* for more information about the *Reports* element.

Contents

The **Contents** element must contain *at least one Part* element. The **Part** element allows you to select a document to include in the message and specify what part of the message the document should fill. At a minimum, include the following elements in the **Part** element:

Element	Description	Sub-elements
Document	Includes document location and encoding information. Note: The document URL must be public so that the EasyLink Messaging infrastructure can access the file.	DocType - The type of document sent. For SMS messages, specify <i>text</i> . Filename - The file that contains the contents of the message. This is the data the recipient will receive. URL - The URL location of the specified file. CharacterSet - The character set used by the message. Most SMS messages

		use <i>UTF-8</i> .
Treatment	Specifies what part of the message the document should be treated as. For an SMS message, specify <i>body</i> .	None.

Sample Request

The following sample request sends the contents of the **UTF-8**-encoded **coldel.txt** file located at **https://xxx/coldel.txt** as an SMS message to all recipients in the **#sms_tst1** list and requests a detailed delivery report. Because the specified delivery schedule is **express**, transmission of the message is treated with priority.

```
<JobSubmitRequest
xmlns="http://ws.easylink.com/JobSubmit/2011/01">
  <Message>
    <JobOptions>
      <Delivery>
        <Schedule>express</Schedule>
      </Delivery>
      <SmsOptions>
        <ExpirationDays>1</ExpirationDays>
        <CharacterSet>UTF-8</CharacterSet>
      </SmsOptions>
    </JobOptions>
    <Destinations>
      <List>
        <Name>#sms_tst1</Name>
      </List>
    </Destinations>
    <Reports>
      <DeliveryReport>
        <DeliveryReportType>detail</DeliveryReportType>
      </DeliveryReport>
    </Reports>
    <Contents>
      <Part>
        <Document>
          <DocType>text</DocType>
          <Filename>coldel.txt</Filename>
          <DocUrl>https://xxx/coldel.txt</DocUrl>
          <CharacterSet>UTF-8</CharacterSet>
        </Document>
        <Treatment>body</Treatment>
      </Part>
    </Contents>
  </Message>
</JobSubmitRequest>
```

Reading the Response

After processing the request, the EasyLink Messaging infrastructure sends a response. Refer to the [JobSubmit Response](#) topic for instructions on reading this response.

Next Actions

If the response indicates success, your message was submitted correctly. Successful delivery depends on a number of factors.

If the response indicates failure, you may want to resend the message in a new request or check the status of the message.

See [Checking Delivery Status](#) for further instructions.

Send a Voice Message

This topic describes how to use the EasyLink Messaging API to send voice messages. Before using these instructions, you must do the following:

- [Upload a list](#) of recipients to the EasyLink Messaging infrastructure.
- Record WAV files for each part of the voice message.

Once the voice messages are ready and the list of recipients is uploaded, you are ready to send a voice message request.

Building the Request

Send voice messages by wrapping a **JobSubmitRequest** element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See *JobSubmit* section of the *EasyLink Messaging API Manual* for complete details about the *JobSubmitRequest* element.

The **JobSubmitRequest** must contain at least one of each of the following elements:

- **DocumentSet**- Contains file information for each of the voice documents used in the message.
- **Message**- Contains all the information necessary to transmit the voice message.

At a minimum, include the following element in the **DocumentSet** element:

Element	Description	Sub-elements
Document	Includes document location and encoding information. This element requires the following attribute: <i>ref</i> -The name of the document. This name is used to reference the document when building the Message element below.	DocType - The type of document sent. For voice messages, specify <i>WAV</i> . Filename - The file that contains the contents of the message. This is the data the recipient will receive. DocData - The contents of the file. This element requires the following attribute: <i>format</i> -The format of the information provided in the DocData element. For WAV files, use <i>base64</i> . CharacterSet - The character set used by the message. Most WAV files use <i>UTF-8</i> .

At a minimum, include the following elements in the **Message** element:

Element	Description	Sub-elements
JobOptions	Includes delivery and message-specific options.	Delivery - Specifies priority and scheduling for the job. VoiceOptions - Includes voice-specific job options.

Destinations	Specifies the voice message recipient <i>See Common Types and Element section of the EasyLink Messaging API Manual for more information.</i>	List- Specifies a list of destinations already stored on the EasyLink Messaging infrastructure.
Reports	Identifies which reports should be returned to the sender by the messaging infrastructure.	Sub-elements depend on the desired report(s).
Contents	Includes the contents of the fax message.	Part- Includes the message to be sent and treatment options for the message.

JobOptions

The contents of the **JobOptions** element define the characteristics of the message. Here are some sub-elements you might wish to include in **JobOptions**:

Element	Description	Sub-elements
BillingCode	Job-specific billing code specified by your messaging application.	None.
CustomerReference	Customer Reference code specified by your messaging application.	None.
VoiceOptions	Includes voice-specific job options. When sending an SMS job, you must include this element, even if its contents are empty. Be sure to include only one modality-specific element in your input.	VoiceDeliveryMethod- Indicates when to begin playback of the voice message. Valid values are: <i>silence-</i> Playback begins after first two seconds of silence. <i>PAMD-</i> Playback begins after a recorded voicemail or answering machine message answers the call. <i>live_only-</i> Playback begins only if a person answers the call. <i>voicemail_only-</i> Playback begins only after a recorded voicemail or answering machine message answers the call.

Destinations

The contents of the **Destinations** element define the recipients of the message. At a minimum, include the following element in the **Destinations** element:

Element	Description	Sub-elements
List	A list of destinations stored on the EasyLink Messaging infrastructure by a previous ListCreate request .	Name- Name of the destinations list to use.

See *Common Types and Elements section of the EasyLink Messaging API Manual* for detailed information about the *Destinations* element, which is of the *DeliveryItem* type.

Reports

The contents of the **Reports** element define what message-related reports to send, and where. For most messages, you should request a delivery report. To do so, include the following element in the **Reports** element:

Element	Description	Sub-elements
DeliveryReport	Requests a delivery report.	DeliveryReportType - Type of delivery report requested. Specify <i>detail</i> for the most information.

See *JobSubmit* section of the *EasyLink Messaging API Manual* for more information about the *Reports* element.

Contents

The **Contents** element must contain *at least one Part* element. The **Part** element allows you to select a document to include in the message and specify what part of the message the document should fill. For example, you could specify one **Part** element to define messages to deliver when a person answers the phone and second **Part** element to define messages to deliver when a voicemail system answers the phone. At a minimum, include the following elements in the **Part** element:

Element	Description	Sub-elements
DocRef	The name of the voice document to send. This should match the document defined above in the DocumentSet element.	None.
Treatment	Specifies what part of the message the document should be treated as. For voice messages, this determines when the message plays. Valid values are: <i>voice_all</i> - Plays in all circumstances. <i>voice_live</i> - Only plays when a real person answers the call. <i>voice_voicemail</i> - Only plays when a voicemail system answers the call. <i>voice_whisper</i> - Only plays to the recipient of a transferred call. <i>voice_call_control</i> - Plays at the beginning or end of the message to provide call control options.	None.

Sample Request

The following sample request defines the **WAVMessage** document using the **base64,UTF-8**-encoded audio data from the **TestMessage.wav** file. The **WAVMessage** document is sent to all the recipients in the **#VoiceTest** list and a **detail** delivery report is requested. Because the specified delivery method is **PAMD**, the message only plays if the call is received by a voicemail system or answering machine.

Note: In this example, the contents of the DocData element are truncated for display purposes.

```

<JobSubmitRequest
xmlns="http://ws.easylink.com/JobSubmit/2011/01">
<DocumentSet>
<Document ref="WAVMessage">
<DocType>WAV</DocType>
<Filename>TestMessage.wav</Filename>
<DocData format="base64">Uk ... gn6BgYCAgA==</DocData>
<CharacterSet>UTF-8</CharacterSet>
</Document>
</DocumentSet>
<Message>
<JobOptions>
<BillingCode>EasyLink WAV Voice Job</BillingCode>
<CustomerReference>EasyLink WAV Voice Job</CustomerReference>
<VoiceOptions>
<VoiceDeliveryMethod>PAMD</VoiceDeliveryMethod>
</VoiceOptions>
</JobOptions>
<Destinations>
<List>
<Name>#VoiceTest</Name>
</List>
</Destinations>
<Reports>
<DeliveryReport>
<DeliveryReportType>detail</DeliveryReportType>
</DeliveryReport>
</Reports>
<Contents>
<Part>
<DocRef>WAVMessage</DocRef>
<Treatment>voice_all</Treatment>
</Part>
</Contents>
</Message>
</JobSubmitRequest>

```

Reading the Response

After processing the request, the EasyLink Messaging infrastructure sends a response. Refer to the [JobSubmit Response](#) topic for instructions on reading this response.

Next Actions

If the response indicates success, your message was submitted correctly. Successful delivery depends on a number of factors.

If the response indicates failure, you may want to resend the message in a new request or check the status of the message.

See [Checking Delivery Status](#) for further instructions.

Read the JobSubmitResult Element

EasyLink Messaging answers job submissions with a JobSubmitResult response. This topic describes how to read the response to determine the status of a Send Message request. Before using these instructions, you must do the following:

- [Send a message](#) using the EasyLink Messaging API.
- Receive a SOAP response from the EasyLink Messaging infrastructure.

Reading the Response

After processing the request, the EasyLink Messaging infrastructure sends a response. The response to a Job Submit request includes a [SOAP response header](#) and a **JobSubmitResult** element in a SOAP envelope.

See *JobSubmit* section of the *EasyLink Messaging API Manual* for complete details about the *JobSubmitResult* element.

The **JobSubmitResponse** element includes a **Status** element for the entire request and a **MessageResult** element for each **Message** element in the request.

Status

The contents of the **Status** element define success or failure of the request message as a whole. At a minimum, read the following elements from the **Status** element:

Element	Description	Sub-elements
StatusCode	Indicates the success or failure of the request. Valid values are: 0- Success 1- Failure 2- Partial success	None.
SubmissionTime	Time the request message was received.	None.
CompletionTime	Time the request message was processed.	None.

See *Common Types and Elements* section of the *EasyLink Messaging API Manual* for detailed information about the *Status* element.

MessageResult

Each **MessageResult** element responds to a **Message** element in the request. At a minimum, include the following elements in the **JobOptions** element:

Element	Description	Sub-elements
Status	Includes status information for the processing of the Message element in the request. <i>See Common Types and</i>	StatusCode - Indicates delivery success or failure. Valid values are: 0- Success 1- Failure

	<i>Elements section of the EasyLink Messaging API Manual for more information.</i>	<p>StatusMessage- Describes the StatusCode value for convenience. Do not rely on this field to indicate success or failure.</p> <p>SubmissionTime-Time the request message was received.</p> <p>CompletionTime- Time the request message was processed and the message was sent.</p>
JobId	<p>Includes Job ID information for the message.</p> <p><i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i></p>	<p>XDN- The EasyLink Messaging Domain Name for the message.</p> <p>MRN- The Message Reference Number for the message.</p>

Sample Response

The sample response below includes a **Status** for the request message as a whole. The **MessageResult** element includes status and job information for the **Message** element in the request. In both cases, the response indicates success (status code0). Because only one message was submitted in the request, one **MessageResult** is returned in this sample.

```
<JobSubmitResult
xmlns=" http://ws.easylink.com/JobSubmit/2011/01 ">
  <Status>
    <StatusCode>0</StatusCode>
    <SubmissionTime>2007-12-18T22:58:25.262Z</SubmissionTime>
    <CompletionTime>2007-12-18T22:58:26.069Z</CompletionTime>
  </Status>
  <MessageResult>
    <Status>
      <StatusCode>0</StatusCode>
      <StatusMessage>OK</StatusMessage>
      <SubmissionTime>2007-12-18T22:58:25.901Z</SubmissionTime>
      <CompletionTime>2007-12-18T22:58:26.068Z</CompletionTime>
    </Status>
    <JobId>
      <XDN>xsi</XDN>
      <MRN>35540992</MRN>
    </JobId>
  </MessageResult>
</JobSubmitResult>
```

Reviewing Jobs

The EasyLink Messaging API has several methods for you to use to review the status of jobs that you have submitted.

Use the following message to review jobs that you have sent to the EasyLink Messaging API:

- [JobDeliveryStatus](#) - Use this message to review detailed information about the status of a job broken down by individual destinations.
- [JobSummary](#) - Use this message to review summarized counts of the status and results of a job.

When EasyLink Messaging responds to a JobSubmitRequest, it provides a JobId with two parts: the EasyLink Messaging Domain Name (XDN) for the message and a Message Reference Number (MRN). A messaging application can use the JobId to request delivery status or job summaries from the messaging infrastructure. Refer to [Read the JobSubmitResult Element](#) for more information about receiving the JobId.

Checking Delivery Status

This topic describes how to submit a delivery status request using a job's JobId.

Building the Request

Send a delivery status request by wrapping a **JobDeliveryStatusRequest** element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See [JobDeliveryStatus](#) section of the *EasyLink Messaging API Manual* for complete details about the *JobDeliveryStatusRequest* element.

The **JobDeliveryStatusRequest** element must contain *at least one* **JobId** element. The JobId element contains all the information necessary to identify which job the application is requesting status for. At a minimum, include the following elements in the **JobId** element:

Element	Description	Sub-elements
XDN	The EasyLink Messaging Domain Name for the message as returned in the original JobSubmitResult response.	None.
MRN	The Message Reference Number for the message as returned in the original JobSubmitResult response.	None.

Sample Request

The sample below requests delivery status for **xsi** message number **35540992**.

```
<JobDeliveryStatusRequest
xmlns="http://ws.easylink.com/JobDeliveryStatus/2011/01">
  <JobId>
    <XDN>xsi</XDN>
    <MRN>35540992</MRN>
  </JobId>
</JobDeliveryStatusRequest>
```

Reading the Response

Once the request is received and processed, the EasyLink Messaging infrastructure sends a response. The response to a delivery status request includes a [SOAP response header](#) and a **JobDeliveryStatusResult** element in a SOAP envelope.

See [JobDeliveryStatus](#) section of the *EasyLink Messaging API Manual* for complete details about the *JobDeliveryStatusResult* element.

At a minimum, read the following elements from the **JobDeliveryStatusResult** element:

Element	Description	Sub-elements
Status	Includes processing success or failure. <i>See Common Types and Elements</i>	StatusCode - Indicates delivery success or failure. Valid values are: 0- Success

	<p><i>section of the EasyLink Messaging API Manual for more information.</i></p>	<p>1- Failure</p> <p>StatusMessage- Describes the StatusCode value for convenience. Do not rely on this field to indicate success or failure.</p> <p>SubmissionTime-Time the request message was received.</p> <p>CompletionTime- Time the request message was processed and the message was sent.</p>
JobDeliveryStatus	Includes job status information.	<p>JobID- Includes Job ID information for the message. Sub-elements include:</p> <p><i>XDN-</i> The EasyLink Messaging Domain Name for the message.</p> <p><i>MRN-</i> The Message Reference Number for the message.</p> <p><i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i></p> <p>RecStatus- The status of job data retrieval. Requires the following attribute:</p> <p><i>code-</i> Numeric code describing whether or not delivery status data retrieval was successful. 0 indicates success.</p> <p><i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i></p> <p>JobData- Includes multiple sub-elements that contain job-specific information. In particular, the DeliveryGroup element includes status information for each message recipient.</p>

JobData

The **JobData** element provides comprehensive information about the job. Here are some of the sub-elements of the **JobData** element:

Element	Description	Sub-elements
JobId	Includes Job ID information for the message. <i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i>	XDN- The EasyLink Messaging Domain Name for the message. MRN- The Message Reference Number for the message.
BillingCode	Job-specific billing code specified by your messaging application.	None.
CustomerReference	Customer Reference code specified by your messaging application.	None.

DeliveryPriority	Delivery priority specified when the message was sent.	None.
EntryTime	Time the job entered the system.	None.
ListsUsed	Identifies any stored list used for the job.	ListId - Identifies a list. Attributes include: <i>liststore</i> - Indicates which service contains the list. <i>ownership</i> - Indicates the ownership level (<i>system, customer, group, user</i>) of the list. <i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i>
ExpirationTime	Time the email service was completed.	None.
Subject	The subject of the message, if any.	None.
UserID	The User ID of the job's owner.	None.
PostedTime	Time all primary deliveries were completed.	None.
JobState	Indicates the job's current state. Valid values are: <i>Awaiting Setup</i> <i>Setup Processing</i> <i>Awaiting Delivery</i> <i>Delivering</i> <i>Posted</i> <i>Delivered</i> <i>Reporting</i> <i>Complete</i> <i>Error</i> <i>Cancelled</i> Note that multiple JobState elements may be present.	None.
DestinationCount	Number of destinations for the message.	None.
ServiceType	Type of Service performed.	None.
JobFlags	Flags on the job, if any.	None.
EmailFrom (Not used for Voice messages)	The From address of an email, SMS, or fax message.	None.
Filename	Name of the file used in the message, if any.	None.

FaxMode (Fax only)	Fax mode specified when the message was sent.	None.
Class	Class of message sent. Valid values include: <i>bcast</i> - broadcast <i>p2p</i> - peer to peer	None.
CharacterSet	Encoding used in the message.	None.
DeliveryGroup	Includes information about a specific set of destinations.	DeliveryGroupId - Identifies the group. AttachmentInfo (email only) - Includes detailed information about attachments to an email message. DeliveryDetail - Includes information about a specific delivery target. The number of DeliveryDetail elements should match the DestinationCount .

AttachmentInfo (email Only)

The **AttachmentInfo** element provides information about a file attached to the job. This element is usually only applicable to email jobs. At a minimum, read the following elements from the **AttachmentInfo** element:

Element	Description	Sub-elements
FileName	Name of the file used as an attachment.	None.
Size	Size of the attached file in bytes.	None.
AttachmentFlags	Flags on the attachment, if any. Valid values include: <i>pull</i> <i>open</i> <i>trackable</i>	None.
Security	Degree of encryption. Valid values are: <i>none</i> <i>40bit</i> <i>128bit</i>	None.
ContentType	MIME type of the attachment.	None.
Total Pulls	Number of times a pull attachment has been accessed.	None.
DistinctPulls	Number of times a pull attachment has been accessed from distinct locations.	None.

DeliveryDetail

Each **DeliveryDetail** element within the DeliveryGroup provides information about an individual destination. At a minimum, read the following elements from the **DeliveryDetail** element:

Element	Description	Sub-elements
Ref	The destination's unique reference identifier.	None.
Destination	The destination described in normal form. Format depends on the following attribute: <i>type</i> - Type of destination, such as <i>internet, fax, sms, or voice</i> .	None.
State	Indicates the destination's current state. Valid values are: <i>Pending</i> <i>Submitted</i> <i>InProcess</i> <i>Error</i> <i>Cancelled</i> <i>Held</i> <i>Sent</i> <i>Expired</i>	None.
PrimeAlt	Indicates what level of alternate destination was used. If the primary destination succeeded, this value is 0. Otherwise, this value indicates which alternate destination was used (in order of priority).	None.
ListId	Identifies what list contains this destination. Attributes include: <i>liststore</i> - Indicates which service contains the list. <i>ownership</i> - Indicates the ownership level (<i>system, customer, group, user</i>) of the list. <i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i>	None.
DocumentPageCount (Fax only)	Number of pages in the document delivered, if applicable.	None.
FirstAttemptTime	Time of the first delivery attempt.	None.
LastAttemptTime	Time of the last delivery attempt.	None.
Attempts	Number of delivery attempts made.	None.
DeliveryUnits	Number of delivery units from	None.

	the last delivery. Attributes include: <i>unit</i> - Type of unit counted, such as <i>bytes</i> or <i>seconds</i> .	
RawDeliveryUnits	Number of raw delivery units from the last delivery. Attributes include: <i>unit</i> - Type of unit counted, such as <i>bytes</i> or <i>seconds</i> .	None.
CumulativeUnits	Number of delivery units from all delivery attempts. Attributes include: <i>unit</i> - Type of unit counted, such as <i>bytes</i> or <i>seconds</i> .	None.
RawCumulativeUnits	Number of raw delivery units from all delivery attempts. Attributes include: <i>unit</i> - Type of unit counted, such as <i>bytes</i> or <i>seconds</i> .	None.
BaudRate	Indicates the delivery baud rate, if applicable.	None.
LocalizedDestination	The destination described in localized form. Format depends on the following attribute: <i>type</i> - Type of destination, such as <i>internet</i> , <i>fax</i> , <i>sms</i> , or <i>voice</i> .	None.
TransferNumber (Voice only)	Indicates a transfer number used in the job, if applicable.	None.

Sample Response

Sample responses are provided below for each message type:

[Email Sample](#)

```
<JobDeliveryStatusResult
xmlns="http://ws.easylink.com/JobDeliveryStatus/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-12-18T22:58:25.901Z</SubmissionTime>
    <CompletionTime>2009-12-18T22:58:26.068Z</CompletionTime>
  </Status>
  <JobDeliveryStatus
xmlns="http://ws.easylink.com/JobDeliveryStatus/2011/01">
    <JobId>
      <XDN>usme</XDN>
      <MRN>6705182</MRN>
    </JobId>
    <RecStatus code="0">OK</RecStatus>
```

```

<JobData>
  <JobId>
    <XDN>usme</XDN>
    <MRN>6705182</MRN>
  </JobId>
  <BillingCode>EasyLink MessageREACH Job</BillingCode>
  <CustomerReference>EasyLink MessageREACH Job</CustomerReference>
  <DeliveryPriority>express</DeliveryPriority>
  <EntryTime>2009-12-18T23:00:11.000Z</EntryTime>
  <ExpirationTime>2008-01-01T23:00:11.000Z</ExpirationTime>
  <Subject>Hello (il) EasyLink Messaging Test</Subject>
  <UserId>ml/rechichar</UserId>
  <PostedTime>2009-12-18T23:05:00.000Z</PostedTime>
  <JobState>Delivering</JobState>
  <JobState>Posted</JobState>
  <DestinationCount>11</DestinationCount>
  <ServiceType>MR</ServiceType>
  <JobFlags>8194</JobFlags>
  <EmailFrom>EasyLink Customer</EmailFrom>
  <Class>bcast</Class>
  <CharacterSet>ISO-8859-1</CharacterSet>
  <DeliveryGroup type="Normal">
    <DeliveryGroupId>2</DeliveryGroupId>
    <AttachmentInfo attachmentNum="-1">
      <FileName>none</FileName>
      <Size>43</Size>
      <AttachmentFlags>pull</AttachmentFlags>
      <AttachmentFlags>trackable</AttachmentFlags>
      <AttachmentFlags>open</AttachmentFlags>
      <Security>none</Security>
      <ContentType>image/gif</ContentType>
      <TotalPulls>2</TotalPulls>
      <DistinctPulls>2</DistinctPulls>
    </AttachmentInfo>
    <DeliveryDetail xqn="1">
      <Ref>My Hotmail</Ref>
      <Destination type="internet">bdrec@hotmail.com</Destination>
      <State code="7">Sent</State>
      <PrimeAlt>0</PrimeAlt>
      <ListId liststore="MR" ownership="user">adhoc</ListId>
      <FirstAttemptTime>2009-12-18T23:00:25.000Z</FirstAttemptTime>
      <LastAttemptTime>2009-12-18T23:00:25.000Z</LastAttemptTime>
      <Attempts>1</Attempts>
      <RawDeliveryUnits unit="bytes">1749</RawDeliveryUnits>
      <RawCumulativeUnits unit="bytes">1</RawCumulativeUnits>
      <BaudRate>2400</BaudRate>
      <LocalizedDestination type="internet">
        bdrec@hotmail.com</LocalizedDestination>
    </DeliveryDetail>
    <DeliveryDetail xqn="2">
      ...
    </DeliveryDetail>
    ...
    <DeliveryDetail xqn="11">
      ...
    </DeliveryDetail>
  </DeliveryGroup>

```

```
</JobData>
</JobDeliveryStatus>
</JobDeliveryStatusResult>
```

SMS Sample

```
<JobDeliveryStatusResult
xmlns="http://ws.easylink.com/JobDeliveryStatus/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-12-18T22:58:25.901Z</SubmissionTime>
    <CompletionTime>2009-12-18T22:58:26.068Z</CompletionTime>
  </Status>
  <JobDeliveryStatus
xmlns="http://ws.easylink.com/JobDeliveryStatus/2011/01">
    <JobId>
      <XDN>usme</XDN>
      <MRN>6705182</MRN>
    </JobId>
    <RecStatus code="0">OK</RecStatus>
    <JobData>
      <JobId>
        <XDN>usme</XDN>
        <MRN>6705182</MRN>
      </JobId>
      <DeliveryPriority>express</DeliveryPriority>
      <EntryTime>2009-12-18T11:16:00.000Z</EntryTime>
      <ExpirationTime>2009-12-21T11:16:00.000Z</ExpirationTime>
      <Subject>Scheduled Report</Subject>
      <UserId>ml/rechichar</UserId>
      <PostedTime>2009-12-18T11:16:00.000Z</PostedTime>
      <JobState>Delivered</JobState>
      <JobState>Posted</JobState>
      <DestinationCount>1</DestinationCount>
      <ServiceType>MR</ServiceType>
      <JobFlags>8202</JobFlags>
      <EmailFrom>Bruce Rechichar</EmailFrom>
      <Filename>SrTu0315.rpt</Filename>
      <FaxMode>standard</FaxMode>
      <Class>P2P</Class>
      <CharacterSet>ISO-8859-1</CharacterSet>
      <DeliveryGroup type="Normal">
        <DeliveryGroupId>2</DeliveryGroupId>
        <DeliveryDetail xqn="1">
          <Ref>INTERNET</Ref>
          <Destination type="internet">
            kelly@easylink.com</Destination>
          <State gammacode="6500" code="5">Cancelled</State>
          <PrimeAlt>0</PrimeAlt>
          <ListId liststore="MR" ownership="user">adhoc</ListId>
          <FirstAttemptTime>2009-12-18T11:16:01.000Z</FirstAttemptTime>
          <LastAttemptTime>2009-12-18T11:16:01.000Z</LastAttemptTime>
          <Attempts>1</Attempts>
          <LocalizedDestination type="internet">
            kelly@easylink.com</LocalizedDestination>
```

```
</DeliveryDetail>
</DeliveryGroup>
</JobData>
</JobDeliveryStatus>
</JobDeliveryStatusResult>
```

Voice Sample

```
<JobDeliveryStatus
xmlns="http://ws.easylink.com/JobDeliveryStatus/2011/01">
  <JobId>
    <XDN>xsi</XDN>
    <MRN>35540992</MRN>
  </JobId>
  <RecStatus code="0">OK</RecStatus>
  <JobData>
    <JobId>
      <XDN>xsi</XDN>
      <MRN>35540992</MRN>
    </JobId>
    <BillingCode>EasyLink WAV Voice Job</BillingCode>
    <CustomerReference>EasyLink WAV Voice Job</CustomerReference>
    <DeliveryPriority>express</DeliveryPriority>
    <EntryTime>2009-12-18T22:58:26.000Z</EntryTime>
    <ListsUsed>
      <ListId liststore="XDDS" ownership="user">voicetest</ListId>
    </ListsUsed>
    <UserId>myuser</UserId>
    <JobState>Processing</JobState>
    <JobState>Error</JobState>
    <DestinationCount>1</DestinationCount>
    <ServiceType>VR</ServiceType>
    <JobFlags>40960</JobFlags>
    <Filename>TestMessage.wav</Filename>
    <Class>p2p</Class>
    <CharacterSet>ISO-8859-1</CharacterSet>
    <DeliveryGroup type="Normal">
      <DeliveryGroupId>2</DeliveryGroupId>
      <DeliveryDetail xqn="1">
        <Ref>Bruce Test</Ref>
        <Destination type="voice">9092661830</Destination>
        <State code="1">Pending</State>
        <PrimeAlt>0</PrimeAlt>
        <ListId liststore="XDDS" ownership="user">VoiceTest</ListId>
        <LocalizedDestination type="voice">9092661830</LocalizedDestination>
        <TransferNumber>9096936393</TransferNumber>
      </DeliveryDetail>
    </DeliveryGroup>
  </JobData>
</JobDeliveryStatus> </JobDeliveryStatusResult>
```

Exclusive States

There are “exclusive” states, which every job is in one and only one exclusive state at any time, and “non-exclusive” states, which a job can have between zero up to possibly three non-exclusive states.

Exclusive states are:

- Awaiting Setup
- Setting Up
- Processing
- Awaiting Delivery
- Delivering
- Delivered
- Complete

Examples of non-exclusive states are:

- Cancelled
- Posted
- Reporting
- Preview
- Paused

Checking a Job Summary

This topic describes how to submit a job summary request using a job's JobId.

Building the Request

Send a job summary request by wrapping a **JobSummaryRequest** element in a SOAP envelope. Be sure to include a [SOAP request header](#).

See [JobSummary](#) section of the *EasyLink Messaging API Manual* for complete details about the *JobSummaryRequest* element.

The **JobSummaryRequest** element must contain *at least one* **JobId** element. The **JobId** element contains all the information necessary to identify which job the application is requesting status for. At a minimum, include the following elements in the **JobId** element:

Element	Description	Sub-elements
XDN	The EasyLink Messaging Domain Name for the message as returned in the original JobSubmitResult response.	None.
MRN	The Message Reference Number for the message as returned in the original JobSubmitResult response.	None.

Sample Request

The sample below requests job summary information for **xsi** message number **35540992**.

```
<JobSummaryRequest
xmlns="http://ws.easylink.com/JobSummary/2011/01">
  <JobId>
    <XDN>xsi</XDN>
    <MRN>35540992</MRN>
  </JobId>
</JobSummaryRequest>
```

Reading the Response

Once the request is received and processed, the EasyLink Messaging infrastructure sends a response. The response to a delivery status request includes a [SOAP response header](#) and a **JobSummaryResult** element in a SOAP envelope.

See [JobSummary](#) section of the *EasyLink Messaging API Manual* for complete details about the *JobSummaryResult* element.

At a minimum, read the following elements from the **JobSummaryResult** element:

Element	Description	Sub-elements
Status	Includes status information for the processing of the summary request. <i>See Common Types and Elements</i>	StatusCode - Indicates delivery success or failure. Valid values are: 0- Success

	<p>section of the <i>EasyLink Messaging API Manual</i> for more information.</p>	<p>1- Failure</p> <p>StatusMessage- Describes the StatusCode value for convenience. Do not rely on this field to indicate success or failure.</p> <p>SubmissionTime-Time the request message was received.</p> <p>CompletionTime- Time the request message was processed and the message was sent.</p>
JobSummaryData	Includes job summary information.	<p>Includes multiple sub-elements that contain job-specific information, as well as the following elements:</p> <p>JobState- Includes status information for each state the job has gone through.</p> <p>AttachmentInfo- Includes information about attachments associated with the job.</p>

JobSummaryData

The **JobSummaryData** element provides comprehensive information about the job. Here are some of the sub-elements:

Element	Description	Sub-elements
JobId	<p>Includes Job ID information for the message.</p> <p><i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i></p>	<p>XDN- The EasyLink Messaging Domain Name for the message.</p> <p>MRN- The Message Reference Number for the message.</p>
Status	<p>Includes status information for the processing of the job.</p> <p><i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i></p>	<p>StatusCode- Indicates delivery success or failure. Valid values are:</p> <p>0- Success</p> <p>1- Failure</p> <p>StatusMessage- Describes the StatusCode value for convenience. Do not rely on this field to indicate success or failure.</p> <p>SubmissionTime-Time the request message was received.</p> <p>CompletionTime- Time the request message was processed and the message was sent.</p>
BillingCode	Job-specific billing code specified by your messaging application.	None.
CustomerReference	Customer Reference code specified by your messaging application.	None.

DeliveryPriority	Delivery priority specified when the message was sent.	None.
EntryTime	Time the job entered the system.	None.
ListsUsed	Identifies any stored list used for the job.	ListId - Identifies a list. Attributes include: <i>liststore</i> - Indicates which service contains the list. <i>ownership</i> - Indicates the ownership level (<i>system, customer, group, user</i>) of the list. <i>See Common Types and Elements section of the EasyLink Messaging API Manual for more information.</i>
ExpirationTime	Time the email service was completed.	None.
Subject	The subject of the message, if any.	None.
UserID	The User ID of the job's owner.	None.
PostedTime	Time all primary deliveries were completed.	None.
JobState	Provides information about a current or previous job state. Attributes include: <i>state</i> - The state of the job. Valid values are: <i>Awaiting Setup</i> <i>Setup Processing</i> <i>Awaiting Delivery</i> <i>Delivering</i> <i>Posted</i> <i>Delivered</i> <i>Reporting</i> <i>Complete</i> <i>Error</i> <i>Cancelled</i> Note that multiple JobState elements may be present.	JobSummaryCounts - Includes destination information for the <i>Delivering, Posted, Reporting, and Complete</i> states. Not valid for other JobStates. Sub-elements include: <i>Total</i> - Total number of destinations. <i>Pending</i> - Number of destinations remaining for delivery. <i>Sent</i> - Number of successfully sent destinations. <i>Error</i> - Number of non-deliveries caused by error. <i>Cancelled</i> - Number of non-deliveries caused by cancellation or expiration. <i>TotalDSN</i> - Number of DSNs detected. <i>NegativeDSN</i> - Number of DSNs indicating non-delivery. <i>UnconfirmedDSN</i> - Number of DSNs that have not returned delivery or non-delivery indications. <i>Blocked</i> - Number of destinations/deliveries blocked. <i>Read</i> - Number of deliveries that have been read. <i>Pulls</i> - Number of file pulls.
DestinationCount	Number of destinations for the message.	None.

ServiceType	Type of Service performed.	None.
JobFlags	Flags on the job, if any.	
EmailFrom	The From address of an Email, SMS, or Fax message.	None.
Filename	Name of the file used in the message, if any.	None.
FaxMode (Fax only)	Fax Mode specified when the message was sent.	None.
Class	Class of message sent. Valid values include: <i>bcast</i> - broadcast <i>p2p</i> - peer to peer	None.
CharacterSet	Encoding used in the message.	None.
DeliveryGroup	Includes information about a specific set of destinations.	DeliveryGroupId - Identifies the group. AttachmentInfo (email only) - Includes detailed information about attachments to an email message. DeliveryDetail - Includes information about a specific delivery target. The number of DeliveryDetail elements should match the DestinationCount .

AttachmentInfo (email Only)

The **AttachmentInfo** element provides information about a file attached to the job. This element is usually only applicable to email jobs. At a minimum, read the following elements from the **AttachmentInfo** element:

Element	Description	Sub-elements
FileName	Name of the file used as an attachment.	None.
Size	Size of the attached file in bytes.	None.
AttachmentFlags	Flags on the attachment, if any. Valid values include: <i>pull</i> <i>open</i> <i>trackable</i>	None.
Security	Degree of encryption. Valid values are: <i>none</i> <i>40bit</i> <i>128bit</i>	None.
ContentType	MIME type of the attachment.	None.

Total Pulls	Number of times a pull attachment has been accessed.	None.
DistinctPulls	Number of times a pull attachment has been accessed from distinct locations.	None.

Sample Response

The sample response below includes a **Status** for the request message as a whole. The **MessageResult** element includes status and job information for the **Message** element in the request. In both cases, the response indicates success (status code **0**). Because only one message was submitted in the request, one **MessageResult** is returned in this sample.

```
<JobSummaryResult
xmlns="http://ws.easylink.com/JobSummary/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-12-18T23:57:04.654Z</SubmissionTime>
    <CompletionTime>2009-12-18T23:57:05.080Z</CompletionTime>
  </Status>
  <JobSummaryData jobGroup="false">
    <JobId>
      <XDN>xsi</XDN>
      <MRN>35540992</MRN>
    </JobId>
    <Status>
      <StatusCode>0</StatusCode>
      <StatusMessage>OK</StatusMessage>
      <SubmissionTime>2009-12-18T23:57:05.000Z</SubmissionTime>
      <CompletionTime>2009-12-18T23:57:05.000Z</CompletionTime>
    </Status>
    <UserId>ml/rechichar</UserId>
    <EntryTime>2009-12-18T22:58:26.000Z</EntryTime>
    <BillingCode>EasyLink WAV Voice Job</BillingCode>
    <CustomerReference>EasyLink WAV Voice Job</CustomerReference>
    <DeliveryPriority>express</DeliveryPriority>
    <ListsUsed>
      <ListId liststore="XDDS" ownership="user">voicetest</ListId>
    </ListsUsed>
    <DestinationCount>1</DestinationCount>
    <ServiceType>VR</ServiceType>
    <JobFlags>40960</JobFlags>
    <Filename>TestMessage.wav</Filename>
    <Class>p2p</Class>
    <CharacterSet>ISO-8859-1</CharacterSet>
    <JobState state="Processing">
      <JobSummaryCounts>
        <Total>1</Total>
        <Pending>1</Pending>
        <Sent>0</Sent>
        <Error>0</Error>
        <Cancelled>0</Cancelled>
      </JobSummaryCounts>
    </JobState>
  </JobSummaryData>
</JobSummaryResult>
```

```

<JobState state="Error">
  <ErrorInfo>
    <ErrorCategory>R</ErrorCategory>
    <ErrorProcessingCode>R230</ErrorProcessingCode>
    <ErrorDetail>The Wave Conversion software (SoX) failed</ErrorDetail>
  </ErrorInfo>
</JobState>
</JobSummaryData>
</JobSummaryResult>

```

Additional sample response messages for the remaining message types are shown below:

Email Sample

```

<JobSummaryResult
xmlns="http://ws.easylink.com/JobSummary/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-12-18T23:59:16.373Z</SubmissionTime>
    <CompletionTime>2009-12-18T23:59:16.784Z</CompletionTime>
  </Status>
  <JobSummaryData jobGroup="false">
    <JobId>
      <XDN>usme</XDN>
      <MRN>6705182</MRN>
    </JobId>
    <Status>
      <StatusCode>0</StatusCode>
      <StatusMessage>OK</StatusMessage>
      <SubmissionTime>2009-12-18T23:59:16.000Z</SubmissionTime>
      <CompletionTime>2009-12-18T23:59:16.000Z</CompletionTime>
    </Status>
    <UserId>ml/rechichar</UserId>
    <EntryTime>2009-12-18T23:00:11.000Z</EntryTime>
    <BillingCode>EasyLink MessageREACH Job</BillingCode>
    <CustomerReference>EasyLink MessageREACH Job</CustomerReference>
    <DeliveryPriority>express</DeliveryPriority>
    <DestinationCount>11</DestinationCount>
    <PostedTime>2009-12-18T23:05:00.000Z</PostedTime>
    <ExpirationTime>2010-01-01T23:00:11.000Z</ExpirationTime>
    <ServiceType>MR</ServiceType>
    <JobFlags>8194</JobFlags>
    <Subject>Hello (il) EasyLink Messaging Test</Subject>
    <EmailFrom>EasyLink Customer</EmailFrom>
    <Class>bcast</Class>
    <CharacterSet>ISO-8859-1</CharacterSet>
    <JobState state="Delivering">
      <JobSummaryCounts>
        <Total>11</Total>
        <Pending>2</Pending>
        <Sent>8</Sent>
        <Error>0</Error>
        <Cancelled>1</Cancelled>
        <TotalDSN>0</TotalDSN>
      </JobSummaryCounts>
    </JobState>
  </JobSummaryData>
</JobSummaryResult>

```

```

    <NegativeDSN>0</NegativeDSN>
    <UnconfirmedDSN>8</UnconfirmedDSN>
    <Blocked>1</Blocked>
    <Read>0</Read>
    <Pulls>2</Pulls>
  </JobSummaryCounts>
</JobState>
<JobState state="Posted"></JobState>
<AttachmentInfo attachmentNum="-1">
  <FileName>none</FileName>
  <Size>43</Size>
  <AttachmentFlags>pull</AttachmentFlags>
  <AttachmentFlags>trackable</AttachmentFlags>
  <AttachmentFlags>open</AttachmentFlags>
  <Security>none</Security>
  <ContentType>image/gif</ContentType>
  <TotalPulls>2</TotalPulls>
  <DistinctPulls>2</DistinctPulls>
</AttachmentInfo>
</JobSummaryData>
</JobSummaryResult>

```

Fax Sample

```

<JobSummaryResult
xmlns="http://ws.easylink.com/JobSummary/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-12-18T23:51:12.516Z</SubmissionTime>
    <CompletionTime>2009-12-18T23:51:12.802Z</CompletionTime>
  </Status>
  <JobSummaryData jobGroup="false">
    <JobId>
      <XDN>xsi</XDN>
      <MRN>35513943</MRN>
    </JobId>
    <Status>
      <StatusCode>0</StatusCode>
      <StatusMessage>OK</StatusMessage>
      <SubmissionTime>2009-12-18T23:51:12.000Z</SubmissionTime>
      <CompletionTime>2009-12-18T23:51:12.000Z</CompletionTime>
    </Status>
    <UserId>ml/rechichar</UserId>
    <EntryTime>2009-12-18T22:14:23.000Z</EntryTime>
    <CompletionTime>2009-12-18T22:17:00.000Z</CompletionTime>
    <BillingCode>Billing Code Fax Test</BillingCode>
    <CustomerReference>Fax Test</CustomerReference>
    <DeliveryPriority>express</DeliveryPriority>
    <ListsUsed>
      <ListId liststore="XDDS" ownership="user">brucef2m</ListId>
    </ListsUsed>
    <DestinationCount>1</DestinationCount>
    <PostedTime>2009-12-18T22:16:00.000Z</PostedTime>
    <ServiceType>FR</ServiceType>
    <JobFlags>8194</JobFlags>
    <Filename>Word1pg.doc</Filename>

```

```

<FaxMode>standard</FaxMode>
<Class>p2p</Class>
<CharacterSet>ISO-8859-1</CharacterSet>
<JobState state="Complete">
  <JobSummaryCounts>
    <Total>1</Total>
    <Pending>0</Pending>
    <Sent>1</Sent>
    <Error>0</Error>
    <Cancelled>0</Cancelled>
    <TotalCompletedDeliveryTime>53</TotalCompletedDeliveryTime>
  </JobSummaryCounts>
</JobState>
<JobState state="Posted"></JobState>
</JobSummaryData>
</JobSummaryResult>

```

SMS Sample

```

<JobSummaryResult
xmlns="http://ws.easylink.com/JobSummary/2011/01">
  <Status>
    <StatusCode>0</StatusCode>
    <StatusMessage>OK</StatusMessage>
    <SubmissionTime>2009-12-18T23:52:28.338Z</SubmissionTime>
    <CompletionTime>2009-12-18T23:52:28.633Z</CompletionTime>
  </Status>
  <JobSummaryData jobGroup="false">
    <JobId>
      <XDN>usme</XDN>
      <MRN>6643798</MRN>
    </JobId>
    <Status>
      <StatusCode>0</StatusCode>
      <StatusMessage>OK</StatusMessage>
      <SubmissionTime>2009-12-18T23:52:28.000Z</SubmissionTime>
      <CompletionTime>2009-12-18T23:52:28.000Z</CompletionTime>
    </Status>
    <UserId>ml/rechichar</UserId>
    <EntryTime>2009-12-18T11:16:00.000Z</EntryTime>
    <DeliveryPriority>express</DeliveryPriority>
    <DestinationCount>1</DestinationCount>
    <PostedTime>2009-12-18T11:16:00.000Z</PostedTime>
    <ExpirationTime>2009-12-21T11:16:00.000Z</ExpirationTime>
    <ServiceType>MR</ServiceType>
    <JobFlags>8202</JobFlags>
    <Subject>Scheduled Report</Subject>
    <EmailFrom>Bruce Rechichar</EmailFrom>
    <Filename>SrTu0315.rpt</Filename>
    <FaxMode>standard</FaxMode>
    <Class>P2P</Class>
    <CharacterSet>ISO-8859-1</CharacterSet>
    <JobState state="Delivered">
      <JobSummaryCounts>
        <Total>1</Total>
        <Pending>0</Pending>

```



```
<Sent>0</Sent>
<Error>0</Error>
<Cancelled>1</Cancelled>
<TotalDSN>0</TotalDSN>
<NegativeDSN>0</NegativeDSN>
<UnconfirmedDSN>0</UnconfirmedDSN>
<Blocked>1</Blocked>
<Read>0</Read>
<Pulls>0</Pulls>
</JobSummaryCounts>
</JobState>
<JobState state="Posted"></JobState>
</JobSummaryData>
</JobSummaryResult>
```