HireAbility ALEX^{sm}
Résumé and Job Order Parsing

Web Service API Documentation
SOAP & HTTP POST

Sales:
sales@hireability.com

Support:
support@hireability.com

Voice:
 +1 (603) 432 6653

https://www.hireability.com/

Introduction

Purpose of this document

This document is a development guide for integrating the **ALEX résumé and job description (job order) processing technology** into your applications via a **web service API**. It provides the technical information needed to access the HireAbility web services in flexible ways that will allow ALEX to work seamlessly inside your application.

Copyright

© 2000 – 2020, HireAbility, LLC
ALEX is a registered service mark of HireAbility

Londonderry, NH 03053
Voice: 603.432.6653
Fax: 603.434.5899

Microsoft, Microsoft Windows, MS-Word and Microsoft Word are either registered trademarks or trademarks of Microsoft Corporation in the United States or other countries.

Adobe, Acrobat, Postscript, Postscript 3 and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States or other countries.

All other trademarks listed in this document but not specifically listed above are the property of their respective owners.

Glossary

ALEX

ALEX (**A**utomated **L**inguistics **EX**pert) is a document processing technology that uses pattern recognition, language structure, and artificial intelligence to extract the content of a résumé or a job description (order) into a predefined format (e.g. HR-XML).

HR-XML

When ALEX processes a résumé, part of the resulting XML output is HR-XML-compliant. HireAbility is currently supporting HR-XML 2.5 and 3.2

JSON

HireAbility is currently supporting HR-XML JSON Lightweight Recruiting Standard.

Batch Processing Requests

Batch requests can be made via SOAP, HTTP POST and HTTP GET, Email or FTP. Completed batch results can be returned via a SOAP or HTTP GET request by your application, a HTTP POST to your servers, or by Email or FTP.

Batch Processing via Email

You can take advantage of batch processing through the use of email boxes in the following manner:

An email box is designated as the location from which documents are processed. HireAbility's end service is run at regular intervals (every 1, 3 or 5 minutes) and the emails are accessed and parsed. The results are stored on our servers until the user initiates the **retrieveCompletedDocs()** method (see below). The following information about the email box from which the resumes will be processed is required:

Location	mail.example.com
Port	110
Username	youraddress@example.com
Password	Yourpassword

Real-Time Processing Requests

Real-time requests can be made via SOAP, HTTP POST, and HTTP GET.

WSDL

WSDL is used in conjunction with SOAP, and is provided to your calling application in order to create the SOAP object. HireAbility's ALEX WSDL is located at http://processing.resumeparser.com/ParsingTools.soap?wsdl

Accepted File Types

ALEX can process many different document types including Word, RTF, Plain Text, HTML, and PDF documents. It is recommended that all files have valid extensions before processing. Ask your sales representative about any file extensions that are not mentioned in the list below.

Supported document types:

MS Word (all versions starting with 2.x: doc, docx)
Document Template (.dot)
PDF (all versions)
Text (.txt, .ans, .asc)
MS Office Word Open XML Format Document (.docm)
MS Word Perfect (.doc)
MS *Active Server Page* (.asp)
Rich Text Format (.rtf)
OpenOffice documents (odt)
Microsoft Excel (.xls, .xlsm)
VFC
Web Page HTML, HTM, XHTML and MHTML (.html, .htm, .xhtml and .mht)
Email (body and attachment)

Processing Status, Error and Return Codes

Processing errors and processing status messages are found between the
`<processingErrors></processingErrors>` tags. If more than one error occurs, then
the `<Error>` tags will repeat. If an error or status message occurs during
processing, the following values are populated:

```
<processingErrors> <Error>
            <ErrorCode>[integer error code]</ErrorCode>
            <ErrorMessage>[string error message]</ErrorMessage>
        </Error>
</processingErrors>
```

If there are **no** errors during processing, empty error tags (ex.
`<ErrorCode></ErrorCode>`, `<ErrorMessage>< /ErrorMessage>`) are returned. This
allows for a simple string comparison to determine if any errors occurred.

A complete list of status and return codes can be found in Appendix 1.

Secure Requests

Any of the requests to the SOAP object functions or HTTP Post web services can be
made secure by replacing **http** with **https** in the request for the WSDL file. We strongly
recommend that you use SSL while making requests to download customer stats, to
delete completed documents, or in retrieving completed documents.

Default ALEX Configuration

The default ALEX configuration processes real-time requests from your calling
application and returns HR-XML and the plain text version of the résumé or job
description (order) to your calling application. The HR-XML results and original résumé
or job description (order) will be removed from the HireAbility servers as soon as the
results are returned to your calling application.

In batch mode you will be responsible for checking, retrieving the processed
documents from our servers.

Processing Results

The parsed HR-XML results for each résumé and job description (order) is encapsulated in these HA-XML tags:

```
<Results>
    <HireAbilityXMLResults id="34343" rid="212121"
    usageLimitType="allocation" usageLimit="1000"
    currentUsage="550" remainingUsage="450"
    accountValidThrough="04-10-2008">

    </HireAbilityXMLResults>
</Results>
```

The **id** value is a unique identifier HireAbility supplies for each customer parsing request.

The **rid** value is the optional parameter that can be supplied at the time of the parse request. If no **rid** value is supplied, it won't appear in this tag. The **rid** is most commonly used in batch requests so that results can be re-linked to a record in originating system.

Your **usageLimitType** is based on your account type, possible values are:

allocation – This is the default type for most accounts. When there is an allocation limit, then once you have processed that number of documents, the system will decline further requests for parsing. You can set up your own customers within the HireAbility system with their own

allocations. With an allocation usage type the following attributes will also be populated: **usageLimit currentUsage** and **remainingUsage**

based_on_agreement – The terms of this account are limited by the agreement between you and HireAbility. With this type **remainingUsage** will also be populated.

daily – This type has a daily parse limit. Once this limit is reached, further requests will be declined. With this type the following attributes will also be populated: **usageLimit currentUsage** and **remainingUsage**

weekly – This type has a weekly parse limit. Once this limit is reached, further requests will be declined. With this type the following attributes will also be populated: **usageLimit currentUsage** and **remainingUsage**

monthly – This type has a monthly parse limit. Once this limit is reached, further requests will be declined. With this type the following attributes will also be populated: **usageLimit currentUsage** and **remainingUsage**

yearly – This type has a yearly parse limit. Once this limit is reached, further requests will be declined. With this type the following attributes will also be populated: **usageLimit currentUsage** and **remainingUsage**

If the **usageLimit** attribute is populated it means that your account will process requests until your **currentUsage** value reaches the **useageLimit**. The **remainingUsage** attribute contains the number of parse requests remaining or other words simply your **useageLimit – currentUsage**.

If present, **accountValidThrough** means that your account is date sensitive and will only work until the end of the day listed in the **accountValidThrough** attribute.

Find the HR-XML parsed results between the tags:

```
<Resume xml:lang="EN" xmlns=http://ns.hr-xml.org/2007-04-15
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ns.hr-xml.org/2007-04-15 Resume.xsd">
        <StructuredXMLResume> <!—HR-XML
        content—> </StructuredXMLResume>
        <NonXMLResume>
        <TextResume><!—Base64 Encoded Text Version of the Original Document →
        </TextResume>
        </NonXMLResume>
</Resume>
```

The original document is not returned by default (since in real-time requests your calling application already has the original document). However, your account can be configured to return the original document, and in batch mode it is returned by default. The original document data is **Base64 encoded**. You will find the original document (if so configured) between these tags:

```
<!-- Original document is Base 64 Encoded -->
<originalDocument id="21219">
        <DocumentTitle>document title.doc</DocumentTitle>
        <DocumentData><!—Base64 Encoded Document—>
        </DocumentData>
</originalDocument >
```

where **id** is the HireAbility unique identifier.

Function and Web Service Parameter Definitions

Please, read through this section carefully as the definitions **are not repeated** elsewhere in the document.

string (128)	product_code	Replace with the product_code value provided by your sales rep.
string (128)	document_title	Title of the document to parse. This value is not required if the URL parameter is populated (recommended if the name of the document is not present in the URL). The document title is always to end with the correct file extension.
string (blob)	document_data	Base 64 encoded document to parse. This value is not required if the URL parameter is populated.
string (64)	rid	If provided, it is returned with the completed data in the tag: \<HireAbilityXMLResults rid ="value">. Use this parameter in batch mode so the XML data can be re-attached to the original document/profile. When supplied at the time of the parse request, you can do a lookup using that same value. While providing an rid, an id value is not needed.
string (128)	id	After submitting a batch request, the XML status message contains a unique id value from the HireAbility system. Using this value the completed document can be downloaded. When using id, rid is not required. The id is ignored if rid is present.
string (64)	customer_id	When provided, the request is tracked individually, allowing the function retrieveCustomerStatsNew to retrieve the number of completed parse requests per customer.
string (255)	URL	The URL of a document to parse. This value must be URL encoded. When URL is provided, the document_title and document_data parameters are not necessary and should be left empty (though document_title is still

		recommended).

string (blob)	rd	This parameter is used only in special instances that you and HireAbility agree upon first. This data can be integrated into the XML output or used in other ways to manipulate the output before it is returned.
string (16)	date_format	The format for interpreting both start_date and end_date. Possible date formats are: • MM-DD-YYYY • YYYY-MM-DD • UNIX_TIMESTAMP
string (16)	start_date	The start date of the date range. See the date_format parameter for options
string (16)	end_date	The end date of the date range. See the date_format parameter for options

Accessing ALEX via SOAP

WSDL Definition and Caching

HireAbility WSDL Definition can be found at
http://processing.resumeparser.com/ParsingTools.soap?wsdl
We recommend the use of WSDL caching in order to speed up SOAP requests.

Available Functions

Pass an empty string rather than a null value when passing unused parameters.

// Send a parse request - real-time or batch mode
string ParseDocNew(*string* product_code, *string* document_title, *string*
document_data[,*string* rid[, *string* customer_id[, *string* URL[, string rd]]]])

// Send a parse request - real-time or batch mode
string ParseDocNew_fromURL(*string* product_code, *string* URL[, *string*
document_title [,*string* rid[, *string* customer_id[, *string* rd]]]])

// Send a parse request - real-time or batch mode
string ParseJobDescriptionNew(*string* product_code, *string* document_title, *string*
document_data[, *string* rid[, *string* customer_id[, *string* URL[, string rd]]]])

// Send a parse request - real-time or batch mode
string ParseJobDescriptionNew_fromURL(*string* product_code, *string* URL[, *string*
document_title [, *string* rid[, *string* customer_id[, *string* rd]]]])

// Retrieve customer stats (*pass your customer_id values at processing time*)
string retrieveCustomerStatsNew(*string* product_code, *string* start_date, *string*
end_date, *string* date_format[, *string* customer_id])

// Retrieve parent customer stats (*pass your customer_id values at processing time*)
string retrieveCustomerStatsParentNew(*string* product_code, *string* start_date,
string end_date, *string* date_format [, *string* customer_id])

// Check the status of a document - batch mode
string checkDocStatusNew(*string* product_code, *string* rid, *string* id)

// Retrieve complete document(s) - batch mode
string retrieveCompletedDocsNew(*string* product_code, *string* rid, *string* id)

// Delete a complete document - batch mode
string deleteDocumentsNew(*string* product_code[, *string* rid[, *string* id]])

How to Send a Parsing Request

You can make any of the requests to the following SOAP object functions secure by replacing **http** with **https** in the request for the WSDL file. We strongly recommend that you use SSL while making requests to download customer stats, to delete completed documents, or in retrieving completed documents.

To process a résumé or job description (order), create a new SOAP object using the WSDL provided. Call the **ParseDocNew** or **ParseJobDescriptionNew** function, passing the parameter value for *product_code* and the required document data as shown in the table below. You will use these functions to parse documents in both real-time and batch mode, using a different product_code for each mode.

If you plan to track your individual customers, be sure to populate the *customer_id* parameter. In batch mode, if the document that you are processing already exists in your database, it is useful to send HireAbility the primary key identifier for the document as the *rid* value. This will allow you to attach the results to the original document or account easily upon document completion.

Required parameters are red, recommended parameters are blue, optional parameters are black.

string ParseDocNew(*string* product_code, *string* document_title , *string* document_data, *string* rid, *string* customer_id, *string* URL, *string* rd)

or

string ParseJobDescriptionNew(*string* product_code, *string* document_title, *string* document_data, *string* rid, *string* customer_id, *string* URL, *string* rd)

string ParseDocNew_fromURL(*string* product_code, *string* URL, *string* document_title, *string* rid, *string* customer_id, *string* URL, *string* rd)

or

string ParseJobDescriptionNew_fromURL(*string* product_code, *string* URL, *string* document_title, *string* rid, *string* customer_id, *string* rd)

How to Process Request Results

During a real-time request**, ParseDocNew** and **ParseJobDescriptionNew** will return an XML string with HR-XML and some HA-XML data. HA-XML data includes: the plain text document, and customer and request identifiers.

During a batch request, **ParseDocNew** and **ParseJobDescriptionNew** will return HA-XML containing the status of your batch request. By checking this status, found in the SOAP Body response, you will be able to verify if your request has been received successfully. Because the returned XML processing results are inside of the SOAP Body response, e.g:

```
<ns1:ParseDocResponse><results xsi:type="xsd:string"><!—HireAbility Data-
➔ </results></ns1:ParseDocResponse>
```

the HireAbiliy XML data is an HTML entity encoded by SOAP before the transfer. Therefore, you may need to decode the HTML entities in order to load the XML tree.

How to Check the Processing Status for a Document

While processing documents **in batch mode** there is a need to check the status of a document in processing to see if it has completed, is still in the queue, etc.

Required parameters are red, recommended parameters are blue, optional parameters are black.

---

string checkDocStatusNew(*string* product_code, *string* rid, *string* id)

The SOAP Body response will look like this:

```
<?xml version="1.0" encoding="UTF-8" ?> <Status
id="2915040242" rid="3ff3dx3">
        <StatusCode>752</StatusCode> <StatusMessage>STATUS[752]: This
        document has completed
processing. You will receive the results based on your custom
rules.</StatusMessage>
        <CurrentlyProcessing>true</CurrentlyProcessing>
        <State>Completed - In Packaging</State>
</Status>
```

How to Retrieve Completed Documents

In batch mode, documents that have completed processing can be retrieved using the **retrieveCompletedDocsNew** function. **SSL is required to access this function.**

Required parameters are red, recommended parameters are blue, optional parameters are black.

string retrieveCompletedDocsNew(*string* product_code, *string* rid, *string* id)

The SOAP Body response will look like this:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Results>
  <processingStatus rid="25">
  <statusCode>729</statusCode>
  <statusMessage>STATUS[729]: Valid request for completed client data.
  There are currently 5 completed documents associated with this
  client.</statusMessage>
  <requestSuccess>True</requestSuccess>
  <completedDocuments>2</completedDocuments>
  <totalClientDocuments>7</totalClientDocuments>
  </processingStatus>
  <processingResults>
    <HireAbilityXMLResults id="2942" rid="25"><!-data->
  </HireAbilityXMLResults>
    <HireAbilityXMLResults id="29" rid="26"><!-data->
  </HireAbilityXMLResults>
  </processingResults>
</Results>
```

How to Delete Documents from the System

In the default configuration documents are automatically removed as soon as they are returned to the calling application. If you choose to have the documents remain on the server until you remove them please let HireAbility know so that your account can be configured appropriately. This functionality can be configured to limit access to specific or partial IP addresses with an additional username/password combination. **SSL is required to access this function.**

Required parameters are red, recommended parameters are blue, optional parameters are black.

string deleteDocumentsNew(*string* product_code, *string* rid, *string* id)

The SOAP Body response will look something like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Results>
    <statusCode>768</statusCode>
    <statusMessage>STATUS[768]: Selected records have been
        deleted.</statusMessage>
    <deleteStatus>
        <RecordDeleted id="22341" rid="22341">true</RecordDeleted>
        <RecordDeleted id="22342" rid="22342">true</RecordDeleted>
    </deleteStatus>
</Results>
```

How to Retrieve Customer Statistics

If the *customer_id* parameter is populated when making parse requests, HireAbility tracks the number of parses that that customer completes. Using the **retrieveCustomerStatsNew** function, customer parse statistics can be retrieved for any given date range. **Requests to this function must be made via SSL,** so be sure to reference the WSDL properly with https.

Required parameters are red, recommended parameters are blue, optional parameters are black.

string retrieveCustomerStatsNew(*string* product_code, *string* date_format, *string* start_date, *string* end_date, *string* customer_id)

If you are intending to retrieve stats for all of your customers at once or have only a single *product_code* value across your configurations, then use the **retrieveCustomerStatsParentNew** function.

string retrieveCustomerStatsParentNew(*string* product_code, *string* date_format, *string* start_date, *string* end_date, *string* customer_id)

The SOAP Body response will look like this:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Results>
        <totalDocumentCount>5</totalDocumentCount>
        <totalClientCount>1</totalClientCount> <queryRange>
                <startDate>05-01-2004</startDate>
                <endDate>05-01-2005</endDate>
                <dateFormat>MM-DD-YYYY</dateFormat>
        </queryRange>
        <customerStats>
          <documentCount customer_id="2341">112</documentCount>
          <documentCount customer_id="638">88990</documentCount>
        </customerStats>
</Results>
```

Accessing ALEX via HTTP POST

Available Web Services

// Send a parse request in either real-time or batch mode
// Returns structured résumé data and extended information
Location: processing.resumeparser.com/requestprocessing.html
Input Parameters: *string* product_code, *string* document_title, *string* document_data[, *string* rid[, *string* customer_id[, *string* URL[, string rd]]]

// Returns structured job description (order) data and extended information
Location: processing.resumeparser.com/requestprocessingjd.html Input Parameters: *string* product_code, *string* document_title, *string* document_data[, *string* rid[, *string* customer_id[, *string* URL[, string rd]]]

// Check the status of a request
Location: processing.resumeparser.com/checkrequest.html Input Parameters: *string* product_code, *string* rid, *string* id

// Retrieve complete document(s) - batch mode
Location: processing.resumeparser.com/getcompleteddocs.html Input Parameters: *string* product_code, *string* rid, *string* id

// Retrieve customer stats (pass your customer_id values at processing time)
Location: processing.resumeparser.com/customerstats.html
Input Parameters: *string* product_code, *string* start_date, *string* end_date, *string* date_format[, *string* customer_id]

// Delete a complete document - batch mode
Location: processing.resumeparser.com/deletedocs.html
Input Parameters: *string* product_code[, *string* rid[, *string* id]]

How to Send a Parsing Request

You can create the HTTP POST using a programming language of choice, but if you'd like to use an HTML form as a preliminary test request, your file browse variable should be named *document* and the form definition should have in it enctype="multipart/form-data". The variables *document_data* and *document_title* will be ignored in an HTML form request.

requestprocessing web service is primarily used to initiate **résumé** parse requests and **requestprocessingjd** web service is primarily used to initiate **job description (order)** parse requests.
Required parameters are red, recommended parameters are blue, optional parameters are black.

Protocol: http POST on port 80 or https POST on port 443
Server location: processing.resumeparser.com
Script location: /requestprocessing.html or /requestprocessingjd.html
Parameters: *string* product_code, *string* document_title, *string* document_data, *string* rid, *string* customer_id, *string* URL, *string* rd

How to Process Request Results

During a real-time request, **requestprocessing.html** will return an XML string comprised of HR-XML and some HA-XML data. HA-XML data includes: the plain text résumé, and customer and request identifiers.

During a batch request, **requestprocessing.html** will return an HA-XML string containing the status of your batch request. By checking this status you will be able to verify if your request for batch processing has been received successfully.

How to Check the Processing Status for a Document

SSL is required to access this function.

Required parameters are red, recommended parameters are blue, optional parameters are black.

Protocol: http GET on port 80 or https GET on port 443
Server location: processing.resumeparser.com
Script location: /checkrequest.html
Parameters: *string* product_code, *string* rid, *string* id

The HTTP response will look like this:

```
<?xml version="1.0" encoding="UTF-8" ?> <Status
id="2915040242" rid="3ff3dx3">
  <StatusCode>752</StatusCode> <StatusMessage>STATUS[752]: This
  document has completed
  processing. You will receive the results based on your custom
  rules.</StatusMessage>
  <CurrentlyProcessing>true</CurrentlyProcessing> <State>Completed
  - In Packaging</State>
</Status>
```

How to Retrieve Completed Documents

SSL is required to access this function.

Required parameters are red, recommended parameters are blue, optional parameters are black.

Protocol: https GET on port 443
Server location: processing.resumeparser.com
Script location: /getcompleteddocs.html
Parameters: *string* product_code, *string* rid, *string* id

The HTTP response will look like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Results>
  <processingStatus rid="25">
  <statusCode>729</statusCode>
  <statusMessage>STATUS[729]: Valid request for completed client data.
  There are currently 5 completed documents associated with this
  client.</statusMessage>
   <requestSuccess>True</requestSuccess>
  <completedDocuments>2</completedDocuments>
```

```xml
  <totalClientDocuments>7</totalClientDocuments>
  </processingStatus>
  <processingResults>
    <HireAbilityXMLResults id="2942" rid="25"><!-data->
  </HireAbilityXMLResults>
    <HireAbilityXMLResults id="2943" rid="26"><!-data->
  </HireAbilityXMLResults>
  </processingResults>
</Results>
```

How to Delete Documents from the System

Processed documents are removed automatically as soon as they are returned to the calling application. If you choose to have the documents remain on the server until you remove them, please let HireAbility know so the proper configuration can be made. This functionality can be configured to limit access to specific or partial IP addresses with an additional username/password combination. **SSL is required to access this web service.**

Required parameters are red, recommended parameters are blue, optional parameters are black.

Protocol: https POST or GET on port 443
Server location: processing.resumeparser.com
Script location: /deletedocs.html
Parameters: *string* product_code, *string* rid, *string* id

The HTTP response will look something like this:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Results>
    <statusCode>768</statusCode> <statusMessage>STATUS[768]:
    Selected records have been
      deleted.</statusMessage>
    <deleteStatus>
      <RecordDeleted id="22341"
        rid="22341">true</RecordDeleted>
      <RecordDeleted id="22342"
        rid="22342">true</RecordDeleted>
    </deleteStatus>
</Results>
```

How to Retrieve Customer Statistics

If the *customer_id* parameter is populated when making parse requests, HireAbility tracks the number of parses that the customer completes. If you are intending to retrieve stats for all of your customers at once or have only a single *product_code* value, then it is recommended that you make requests to **customerstatsparent.html**. **SSL is required to access this web service.**

Required parameters are red, recommended parameters are blue, optional parameters are black.

Protocol: https GET or POST on port 443 Server location: processing.resumeparser.com
Script location: /customerstats.html or /customerstatsparent.html Parameters: *string* product_code, *string* date_format, *string* start_date, *string* end_date, *string* customer_id

---

The HTTP response will look like this

```
<?xml version="1.0" encoding="UTF-8" ?>
<Results>
        <totalDocumentCount>5</totalDocumentCount>
        <totalClientCount>1</totalClientCount> <queryRange>
                <startDate>05-01-2004</startDate>
                <endDate>05-01-2005</endDate>
                <dateFormat>MM-DD-YYYY</dateFormat>
        </queryRange>
        <customerStats>
          <documentCount customer_id="2341">112</documentCount>
          <documentCount customer_id="11341">543</documentCount>
        </customerStats>
</Results>
```

Appendix 1 Error Codes

// 690 -> 699: External Server Connection Errors

// 698 - proximate
698   =>   "We were unable to connect to the location @var1:@var3@var2 with your
              completed results. We will keep the completed results on our server until we can re-
              establish a connection to your servers."

// 699 - direct
699   =>   "We were unable to connect to the location @var1:@var3@var2 with your
              completed results. Check with your system administrator as to when your servers will be
              available, then resubmit your document(s) for processing."

// 700 -> 719: Data Capture Status
700   =>   "We have successfully received your document(s) for processing.
              You data will complete processing shortly. The completed data will be
              removed from the system if you fail to retrieve   the data."

701   =>   "The record ( ) was not found. Please contact HireAbility support at "

702   =>   "This account is not currently active. This account exists, but is not currently
              activated. Please contact for inquiries regarding the non-active status of this account. "

704   =>   "Unauthorized IP trying to connect
              Cannot continue processing because your IP address is not allowed connect was refused
              because of wrong IP: "

706   =>   "Unknown Parse Type request. Please contact HireAbility support at "

707   =>   "There was an error trying to upload your document. Please resubmit your
               document for processing. If the problem persists contact HireAbility support at"

708   =>   "There was an error trying to upload your document via ...
              It's likely that an incorrect variable name was passed. "

709   =>   "The URL you supplied did not contain a valid document.
              Please check the URL you supplied and resubmit your parsing request."

710   =>   "The value you supplied as a file to process was empty or too short to process.
              Please check that you selected a valid file before submitting a processing request."

711   =>   "There was an error trying to download your document via a URL. @var1
              With the URL that you supplied, we could not find any data.
              Please check the URL and try submitting again."

712   =>   "There was a problem adding your files for processing.
              Please check that the file you are trying to submit is valid. ".

713   =>   "This account ( ) has run out of credits.
              In order to continue processing, more credits must be purchased. "

714   =>   "Account ( ) must provide a customer_id in order to process requests. "

716   =>   "Account ( ) is not currently active because the expiration date for the account
               has passed. Please contact HireAbility or your Parsing Reseller to re-activate your account."

// 720 -> 729: HTTP GET Completed Data Errors
720 => "Requests to retrieve completed data must be SSL encrypted on port 443. Request for
               completed documents under ( ) cannot continue."

721   =>   "You must provide your cid,pid and did values in order to download completed
               documents. Request for completed documents ( ) cannot continue."

722   =>   "The document rules associated with ( ) do not allow this method of retrieval for
               completed documents. Request for completed documents cannot continue. Please contact
               HireAbility support at "

723   =>   "The client associated with cid:@var2 is not currently active.
               Request for completed documents cannot continue. Please contact HireAbility
               support at "

725   =>   "No Client Setup rules record with the values ( ) could be found.
               Request for completed documents cannot continue. Please contact HireAbility support at
               "

726   =>   "Requests for this clients documents are IP address restricted.
               Your IP address is not valid and request for completed documents cannot continue.
               Please contact HireAbility support at "

727   =>   "Requests for this clients documents are username/password restricted.
               Your username/password is not valid and request for completed documents cannot
               continue. Please contact HireAbility support at "

728   =>   "Valid request for completed client data.
               There are currently no completed documents associated with this client."

729   =>   "Valid request for completed client data.
               There are currently @var1 completed documents associated with this client."

// 730 -> 749: Sub Client Parse Count Stats
730 => "Requests to download customer stats data must be SSL encrypted on port 443. Request for
               stats ( ) cannot continue."

731   =>   "You must provide your cid value in order to download parsing stats.
               Request for stats ( ) cannot continue."

732   =>   "No valid parse purchase found under ( ). Request for stats cannot continue."

733   =>   "No date range given. Request for stats cannot continue. ( )"

734   =>   "No date format specified. Valid date formats are (@var3). Request for stats
               cannot continue. ( ) "

735   =>   "Incorrect date format specified. Valid date formats are (@var3). Request for
               stats cannot continue. ( )"

740   =>   "Requests to download customer usage data must be SSL encrypted on port"
             443. Request for stats ( )."

741   =>   "You must provide your product_code value in order to download usage
              information. Request for stats ( ) cannot continue."

742   =>   "You must provide your product_code value in order to download usage
              information. Request for stats ( ) cannot continue."

743   =>   "No valid parse purchase found under ( ). Request for usage info cannot
              continue."

// 750 -> 759: Document Request Status
750 => "This document has completed processing and is no longer in our system. The results have
                been packaged and returned."

751   =>   "This document is currently in processing."

752   =>   "This document has completed processing.
              You will receive the results based on your custom rules."

755   =>   "You must provide a did value to do a lookup with an rid.
              Please provide the proper did value in order to do a document status lookup."

756   =>   "You must provide an id or a did and rid value to do a document status lookup."

// 760 -> 769: Client Delete Record Errors
760   =>   "Requests to delete completed data must be SSL encrypted on port 443.
              Request to delete records under cid:@var2,pid:@var1,did@var3 cannot continue."

761   =>   "You must provide your cid,pid and did values in order to delete records.
              Request to delete records (cid:@var2,pid:@var1,did@var3) cannot continue."

762   =>   "You must provide id or rid values in order to remove records.
              Request to delete records cannot continue ( )."

763   =>   "The client associated with product_code ( ) is not currently active.
              Request to delete records cannot continue. Please contact HireAbility support"

764   =>   "The parse purchase associated with pid:@var1 is not currently active.
              Request to delete records cannot continue. Please contact HireAbility support "

765   =>   "No Client Setup rules record with the values ( ) could be found.
              Request to delete records cannot continue. Please contact HireAbility support "

766   =>   "Requests for this clients documents are IP address restricted.
              Your IP address is not valid and request to delete records cannot continue. Please contact
              HireAbility support at "

767   =>   "Requests to delete this clients documents are username/password restricted.
              Your username/password is not valid and request for deletion cannot continue. Please
              contact HireAbility support at "

| 768 | => | "Selected records have been deleted." |
| 769 | => | "Selected records have previously been deleted." |

// 3300 -> 3400: Disable parsing on event errors

3310 =>   "This account (pid:) has reached its yearly processing limits. In order to continue

processing documents this year, additional credits may be purchased by contacting sales@hireability.com ".

// 8000 -> 8050 - Document conversion errors

8000 => "Document @name failed to convert to into plain text format which is required for parsing. The document may be in a non-standardized format. Try converting this document into text and then resubmitting it for processing.",

8001 =>   "Document @name contains no data and therefore cannot be converted into plain text format, which is required for parsing."

8002 =>   "There was a problem converting the document @var1 to text. Please check if @name is a valid document and if so try saving the file in plain text format and resubmitting it for processing."

8003 =>   "Document @name failed to convert to plain text format which is required for parsing. We tried more than one way to convert @name to plain text without success. This file is likely corrupted or incomplete."

8004 =>   "PDF Document @name is protected and therefore cannot be converted to plain text format which is required for parsing. The document must be submitted in an unprotected format for processing.",

8005 =>   "PDF Document @name failed to convert to plain text format which is required for parsing. Some PDF's are images rather than text or perhaps this PDF is protected. Please resumbit this document in plain text format."

8006 =>   "This document converted successfully to text but was found to contain no data. This document is most likely a valid but empty word document. Check that the document contains data."

8007 =>   "Document: @name is not a supported document type. Please refer to http://www.hireability.com/ALEX/system_message.html?show=processing_file_types to see a list of supported file types."

8009 =>   "This document appears to be password protected. It cannot be opened for conversion nto plain text which is required for parsing. The document must be submitted in an unprotected format for processing."

8010 =>   "This document is too large to be converted. Please limit documents to 2MB. The Document must be no larger than 2MB."

Appendix 2

A1. Sample Résumé Parsing Output File

This is an example of the structured XML output that ALEX will return after processing a résumé. See Appendix A3 for an example of all HR-XML tags currently supported by ALEX resume processing.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
- <!-- [xml content created: 2-23-2006 09:09:29] -->
- <HireAbilityXMLResults id="29126002631" rid="pansy Trellis.doc">
<Resume xml:lang="EN" xmlns="http://ns.hr-xml.org/2007-04-15"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ns.hr-xml.org/2007-04-15 Resume.xsd">
- <StructuredXMLResume>
- <ContactInfo>
- <PersonName>
  <FormattedName>Pansy Trellis</FormattedName>
  <GivenName>Pansy</GivenName>
  <FamilyName>Trellis</FamilyName>
    </PersonName>
- <ContactMethod>
- <PostalAddress>
  <CountryCode>US</CountryCode>
  <PostalCode>02210</PostalCode>
  <Region>MA</Region>
  <Municipality>Boston</Municipality>
- <DeliveryAddress>
  <AddressLine>12 Garden Lane</AddressLine>
    </DeliveryAddress>
    </PostalAddress>
    </ContactMethod>
- <ContactMethod>
  <Use>personal</Use>
  <Location>home</Location>
  <WhenAvailable>unknown</WhenAvailable>
- <Telephone>
  <TelcomCountryCode>1</TelcomCountryCode>
  <AreaCityCode>617</AreaCityCode>
  <SubscriberNumber>555-1212</SubscriberNumber>
    </Telephone>
    </ContactMethod>
- <ContactMethod>
  <Use>business</Use>
  <Location>office</Location>
```

```xml
<WhenAvailable>daytime</WhenAvailable>
- <Telephone>

<TelcomCountryCode>1</TelcomCountryCode>
<AreaCityCode>617</AreaCityCode>
<SubscriberNumber>555-2121</SubscriberNumber>

</Telephone>
</ContactMethod>
- <ContactMethod>
<Use>personal</Use>
<Location>onPerson</Location>
<WhenAvailable>anytime</WhenAvailable>
- <Mobile> <TelcomCountryCode>1</TelcomCountryCode>
<AreaCityCode>617</AreaCityCode>
<SubscriberNumber>555-2211</SubscriberNumber>

</Mobile>
</ContactMethod>
- <ContactMethod>
<InternetEmailAddress>pansytrellis@msn.com</InternetEmailAddress>
</ContactMethod>
- <ContactMethod>
<InternetWebAddress>www.greenthumb.com</InternetWebAddress>

</ContactMethod>
</ContactInfo>
- <EmploymentHistory>
- <EmployerOrg>
<EmployerOrgName>Knight Communications</EmployerOrgName>
- <PositionHistory>
<Title>EJB Developer</Title>
- <OrgInfo>
- <PositionLocation> <Region>VA</Region>
<Municipality>Vienna</Municipality>

</PositionLocation>
</OrgInfo>

<Description>Web/e-commerce applications Digital Interactive TV As a member of
the project team I designed and coded some of the Framework classes used in
the Data Driven Menu.</Description>
- <StartDate> <AnyDate>2003-
09</AnyDate>
</StartDate>
- <EndDate> <AnyDate>2009-
02</AnyDate>
</EndDate>
</PositionHistory>
</EmployerOrg>
- <EmployerOrg>
<EmployerOrgName>eInteractive, Inc.</EmployerOrgName>
- <PositionHistory>
<Title>Java Consultant/J2EE Developer</Title>
```

– <OrgInfo>

– <PositionLocation>
<Region>**VA**</Region>

<Municipality>**Falls   Church**</Municipality>
</PositionLocation>
</OrgInfo>

<Description>**This project was carried out on the supplier's site (I was contracted by DHL for a consultant/developer role as part of the supplier's team) to work on an n-tier Interactive Web Application.**</Description>

– <StartDate> <AnyDate>**2002-11**</AnyDate>
</StartDate>

– <EndDate> <AnyDate>**2003-07**</AnyDate>
</EndDate>
</PositionHistory>
</EmployerOrg>

– <EmployerOrg>
<EmployerOrgName>**Galaxy Systems**</EmployerOrgName>

– <PositionHistory>
<Title>**J2EE Application Architect**</Title>

– <OrgInfo>

– <PositionLocation> <Region>**D.C**</Region>
<Municipality>**Washington**</Municipality>
</PositionLocation>
</OrgInfo>

<Description>**Architectural work carried out for a major ISP to provide an Internet Self-Care application project.**</Description>

– <StartDate> <AnyDate>**2002-08**</AnyDate>
</StartDate>

– <EndDate> <AnyDate>**2001-10**</AnyDate>
</EndDate>
</PositionHistory>
</EmployerOrg>
</PositionHistory>
</EmployerOrg>
</EmploymentHistory>

– <EducationHistory>

– <SchoolOrInstitution schoolType="**university**">

– <School>
<SchoolName>**Harvard   University**</SchoolName>
</School>

– <Degree degreeType="**masters**">
<DegreeName>**Masters**</DegreeName>

– <DegreeDate>
<AnyDate>**2002**</AnyDate>

```xml
          </DegreeDate>
     <DegreeMajor>Computer Science</DegreeMajor>
          </Degree>
          </SchoolOrInstitution>
 <SchoolOrInstitution schoolType="university">
 <School>

     <SchoolName>Harvard        University</SchoolName>
          </School>
 <Degree degreeType="bachelors">
 <DegreeName>Bachelors</DegreeName>
 <DegreeDate>
 <AnyDate>2000</AnyDate>
          </DegreeDate>

     <DegreeMajor>Computer        Science</DegreeMajor>
          </Degree>
          </SchoolOrInstitution>
          </EducationHistory>
 <Qualifications>

     <Competency name="Programming Tools" /> <Competency
     name="Java Programming Language" /> <Competency
     name="Jdk" />
     <Competency name="Enterprise Java Beans" />
 <Competency name="JDBC">

 <CompetencyEvidence name="JDBC" typeId="JDBC" lastUsed="2002"> <NumericValue
 description="Range in years of experience">3</NumericValue>

          </CompetencyEvidence>
          </Competency>

     <Competency name="Communications" />
     <Competency name="Text Editors" /> <Competency
     name="System Administration" /> <Competency
     name="Manufacturing" />

          </Qualifications>
          </StructuredXMLResume>


 <NonXMLResume>

     <textResume id="29126002631"> <DocumentTitle>pansy
     Trellis.doc.txt</DocumentTitle>

     <DocumentData><!—Base64 Encoded Plain Text Résumé→</DocumentData> </textResume>
 </NonXMLResume>
     </Resume>
 <processingErrors>
 <Error id="29126002631">
 <ErrorCode /> <ErrorMessage />

 </Error>
 </processingErrors>
 </HireAbilityXMLResults>
```

A2. Sample Job Description Parsing Output File

This is an example of the structured XML output that ALEX will return after processing a job description (order).

```xml
<?xml version="1.0" encoding="UTF-8" ?>
- <Results>
- <HireAbilityXMLResults id="2916255462311" rid=""
    usageLimitType="based_on_agreement" currentUsage="1045">
<PositionOpening xmlns="http://ns.hr-xml.org/2007-04-15"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ns.hr-xml.org/2007-04-15
    PositionOpening.xsd">
- <PositionPostings>
- <PositionPosting>
- <Id>
  <IdValue>1111</IdValue>
    </Id>
  <Title>Associate Project Manager</Title>
    </PositionPosting>
    </PositionPostings>
- <PositionProfile>
- <PositionDetail>
  <IndustryCode classificationName="management" />
- <JobCategory>
  <TaxonomyName          version="4.0">ONET</TaxonomyName>
    </JobCategory>
  <PositionTitle>Associate Project Manager</PositionTitle>
- <Competency name="Information Technology">
  <UserArea>* This Associate IT Project Management position will be responsible
    for managing the GI Intranet Revamp project.</UserArea>
    </Competency>
- <Competency name="Status Report">
  <UserArea>* This position is expected to provide timely status reports to all
    stakeholders</UserArea>
    </Competency>
- <Competency     name="Problem     Solving">
  <UserArea>problem solving</UserArea>
    </Competency>
- <Competency name="Leadership">
  <UserArea>*Proven    leadership    abilities</UserArea>
    </Competency>
    </PositionDetail>
    </PositionProfile>
  <NumberToFill>1</NumberToFill>
- <UserArea>
  <MinimumEducationLevel school="University" degree="bachelors" />
  <RequiredWorkExperience workDuration="year">2</RequiredWorkExperience>
```

<TextJobDescription><!—Base64 Encoded Plain Text Job Description →
</TextJobDescription>
</UserArea>
</PositionOpening>
– <processingErrors>
– <Error   id="**2916255462311**">
<ErrorCode />
<ErrorMessage /> </Error>
</processingErrors>
</HireAbilityXMLResults>
</Results>

A3. Supported Résumé/CV HR-XML and JSON Tags

Below you will find a complete list of the ALEX supported HR-XML and JSON tags. The XML and JSON in this example is intended only as a structure guide for the HR-XML or JSON your calling application will receive. It contains the HR-XML or JSON content only. HR-XML tags that contain no data will not be returned or will be returned in a <tag></tag> format.

Click here for Supported HR-XML Resume Tags

Click here for Supported JSON Resume/CV Tags

A4. Currently Supported Job Order HR-XML and JSON Tags

Click here for Supported HR-XML Job Order Tags

Click here for Supported JSON Job Order Tags