



## Job Posting Web Service - v04

Proprietary information notice

This material is strictly confidential and proprietary to Madgex. Any use or disclosure for purposes other than the intent of the original distribution of this document is prohibited.

© Madgex 2014-16. All rights reserved. It cannot be reproduced without the permission of Madgex Limited.

# Contents

- Contents
  - Introduction
- Setup
- Process Overview
- Method Details & Additional Classes
  - Logon method
  - Logoff method
  - AddJob Method
  - UpdateJob Method
  - UpdateJobProperties Method
  - LogonUpdateJobPropertiesLogoff Method
  - ArchiveJob Method
  - LogonAddJobLogoff Method
  - LogonUpdateJobLogoff Method
  - LogonArchiveJobLogoff Method
  - GetActiveUpsellPackages Method
  - GetApplicationMethodTypes
  - GetAvailableRecruiters Method
  - GetCategories Method
  - GetCategoryTerms Method
  - GetJobProperties Method
  - GetLocations Method
  - WSJobProperty Class
  - WSCategory Class
  - WSTerm Class
  - WSRecruiter Class
  - WSUpsell Class
  - WSJobPropertyValue Class
  - WSLocation Class
  - WSSelectedTerms Class
  - GetHeirarchicalCategoryTerms Method
  - GetSynonyms Method
  - LogonRepublishArchivedJobLogoff Method
  - RepublishArchivedJob Method
  - LogonExpireJobLogoff Method
  - ExpireJob Method
  - LogonAddJobLogoffWithReturnJobDetail Method
  - AddJobWithReturnJobDetail Method
  - WSAddJobResult Class

- Job Categorisation IDs
- Exception Handling
- Code Example
- Location Changes
  - New Location SOAP Call
  - Getting the Location IDs

## Introduction

This document details how to use the Job Posting Web Service which can be used to post jobs to the Job Board. This is an extension of the Job Posting Web Service.

All examples are given in Microsoft's C# programming language.

# Setup

The namespace of the webservice is `http://<SITENAME>-webservice.madgex.co.uk`

The endpoint for the test service is `http://<SITENAME>-webservice.madgexjbttest.com`

The endpoint for the live service is `http://<SITENAME>-webservice.madgexjb.com`

When you start developing against the web service you must code against the test system. You will be provided with login credentials for both test and live systems, along with the IDs of the recruiters for whom you will add, edit and archive jobs (Note: recruiter IDs will vary between test and live systems).

When developing on the test system you will be able to see the jobs you have posted by viewing the test system version of your jobs site, the URL for which is `http://<SITENAME>-web.madgexjbttest.com/`

Please also remember that other uploaders may be using the system, so do not post offensive material.

# Process Overview

The general process of using the webservice is as follows:

Start a web service session by calling the 'Logon' method, storing the returned string that is required for all subsequent method calls.

Utilise the AddJob, UpdateJob, and ArchiveJob etc methods as required.

Call the Logoff method to end your web service session.

NB – additional methods have been added to the webservice that allow logging on, adding a job (for instance), and logging off all in one go. Where possible it is encouraged to use the separate method calls as many job postings can be done under one session which makes for better reporting in the CMS.

# Method Details & Additional Classes

The tables below specify the methods of the webservice. Please note that a parameter marked as being optional must still be passed, but can be an empty string (i.e. you must not omit it from your method call). Parameters marked as mandatory cannot be passed as an empty string. Please refer to the "Exception Handling" section for details of capturing each method's possible errors.

## Logon method

Logs on to the web service using the supplied credentials and creates a session in which an uploader can add, update and archive jobs. If successful this method returns uiSecurityToken (string), which should be held for the duration of the session by the client application, as it is required by all other methods.

Parameter	Description
sUploaderID	Mandatory string identifying the uploader.
sPassword	Mandatory string. Uploader's password.

## Logoff method

Ends the current session. The absence of a thrown exception indicates a successful call of this method. If this method is not called a session will automatically expire after five hours of not being used.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

## AddJob Method

Posts a new job immediately to the website using the given parameters. The absence of a thrown exception indicates a successful call of this method.

---

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)
sRecruiterID	Mandatory ID of the recruiter
sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sJobID	Mandatory string (20) which should be unique amongst all of that recruiters jobs. This ID, coupled with the specified recruiter allows you to call the update and archive methods for the same job at a later date.
sStartDateTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the current time will be used.
sEndDateTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.

sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sApplicationEmail	<p>Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.</p>
sExternalApplicationURL	<p>Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.</p>

<p>jobProperties</p>	<p>Array of WSJobPropertyValue objects specifying job properties</p> <p>Currently this includes</p> <p>JobTitle - string(100). Title of the job.</p> <p>JobDescription</p> <p>JobLocations - string(255), CSV of location Ids Max:4 (use GetLocations() to obtain Ids)</p> <p>Salary Description - string(100). Salary description shown on job details</p> <p>This is subject to change so to obtain the full list of available properties including non-mandatory call the GetJobProperties method</p>
<p>jobCategorization</p>	<p>Array of WSSelectedTerms objects specifying job categorizations</p> <p>This includes some mandatory categories which are site specific. Call GetCategories to obtain a list of the categories applicable</p>
<p>jobUpsells</p>	<p>Array of WSUpsell objects specifying job upsells</p>
<p>productDefinitionID</p>	<p>Optional int specifying the product. If not supplied, it will assume the basic job posting product. Product ID's can be obtained from the GetAvailableProducts method.</p>

jobLogo	<p>Optional. Contains two sub-elements</p> <ol style="list-style-type: none"> <li>1. AssetID - If ID of a logo is known then this will assign this asset/logo to the newly created job.</li> <li>2. Base64BlobString - Base 64 encoding of the logo bytes.</li> </ol> <p>If jobLogo is specified then at least one of the above sub-elements must be specified. If both are specified and the AssetID is valid then the logo relating to that AssetID will be used. Otherwise a new Logo will be created using the Base64BlobString. If both elements are specified and empty or in anyway invalid then an error will be thrown with the message Unable to process Job Logo</p>
---------	---

## UpdateJob Method

Updates the specified job on the website with immediate effect using the given parameters. The absence of a thrown exception indicates a successful call of this method.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)
sRecruiterID	Mandatory ID of the recruiter
sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to update.

sEndDateTime	Optional string specifying the end date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.
sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sApplicationEmail	Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.
sExternalApplicationURL	Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.
jobProperties	Array of WSJobPropertyValue objects specifying job properties

jobCategorization	Array of WSSelectedException objects specifying job categorizations
-------------------	---

## UpdateJobProperties Method

Updates the individual job properties on the website with immediate effect using the given parameters. The absence of a thrown exception indicates a successful call of this method.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)
sRecruiterID	Mandatory ID of the recruiter
sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to update.
sInternalJobID	Optional string which can be used instead of the sJobID. If this is used, then the sRecruiterID does not need to be supplied. This is the Madgex internal job ID.
sEndTime	Optional string specifying the end date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.

sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sApplicationEmail	<p>Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.</p>
sExternalApplicationURL	<p>Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.</p>
jobProperties	<p>Array of WSJobPropertyValue objects specifying job properties</p>
jobCategorization	<p>Array of WSSelectedTerms objects specifying job categorizations</p>

## LogonUpdateJobPropertiesLogoff Method

Logs in and updates the individual job properties on the website with immediate effect using

the given parameters. The absence of a thrown exception indicates a successful call of this method.

Parameter	Description
sUploaderID	Mandatory string identifying the uploader.
sRecruiterID	Mandatory ID of the recruiter
sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to update.
sInternalJobID	Optional string which can be used instead of the sJobID. If this is used, then the sRecruiterID does not need to be supplied. This is the Madgex internal job ID.
sEndTime	Optional string specifying the end date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.

sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sApplicationEmail	<p>Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.</p>
sExternalApplicationURL	<p>Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.</p>
jobProperties	<p>Array of WSJobPropertyValue objects specifying job properties</p>
jobCategorization	<p>Array of WSSelectedTerms objects specifying job categorizations</p>

## ArchiveJob Method

Archives the specified job on the website with immediate effect. The absence of a thrown

exception indicates a successful call of this method.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)
sRecruiterID	Mandatory ID of the recruiter
sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to act upon.

## LogonAddJobLogoff Method

Logs in. Posts a new job immediately to the website using the given parameters. The absence of a thrown exception indicates a successful call of this method. Logs off.

Parameter	Description
sUploaderID	Mandatory string identifying the uploader.
sPassword	Mandatory string. Uploader's password.
sRecruiterID	Mandatory ID of the recruiter
sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sJobID	Mandatory string(20) which should be unique amongst all of that recruiters jobs. This ID, coupled with the specified recruiter allows you to call the update and archive methods for the same job at a later date.
sStartDateTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the current time will be used.

sEndDateTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.
sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sApplicationEmail	Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.
sExternalApplicationURL	Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.
jobProperties	Array of WSJobPropertyValue objects specifying job properties

jobCategorization	Array of WSSelectedExceptions objects specifying job categorizations
jobUpsells	Array of WSUpsell objects specifying job upsells
productDefinitionID	Optional int specifying the product. If not supplied, it will assume the basic job posting product. Product ID's can be obtained from the GetAvailableProducts method.
jobLogo	<p>Optional. Contains two sub-elements</p> <ol style="list-style-type: none"> <li>1. AssetID - If ID of a logo is known then this will assign this asset/logo to the newly created job.</li> <li>2. Base64BlobString - Base 64 encoding of the logo bytes.</li> </ol> <p>If jobLogo is specified then at least one of the above sub-elements must be specified. If both are specified and the AssetID is valid then the logo relating to that AssetID will be used. Otherwise a new Logo will be created using the Base64BlobString. If both elements are specified and empty or in anyway invalid then an error will be thrown with the message Unable to process Job Logo</p>

## LogonUpdateJobLogoff Method

Logs in. Updates the specified job on the website with immediate effect using the given parameters. The absence of a thrown exception indicates a successful call of this method. Logs off.

Parameter	Description
sUploaderID	Mandatory string identifying the uploader.
sPassword	Mandatory string. Uploader's password.
sRecruiterID	Mandatory ID of the recruiter

sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sJobID	Mandatory string which, coupled with the liRecruiterID parameter, uniquely identifies a job as the job to update.
sEndTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.
sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sApplicationEmail	Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.

sExternalApplicationURL	Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.
jobProperties	Array of WSJobPropertyValue objects specifying job properties
jobCategorization	Array of WSSelectedTerms objects specifying job categorizations

## LogonArchiveJobLogoff Method

Logs in. Archives the specified job on the website with immediate effect. The absence of a thrown exception indicates a successful call of this method. Logs off.

Parameter	Description
sUploaderID	Mandatory string identifying the uploader.
sPassword	Mandatory string. Uploader's password.
sRecruiterID	Mandatory ID of the recruiter
sJobID	Mandatory string which, coupled with the liRecruiterID parameter, uniquely identifies a job as the job to update.

## GetActiveUpsellPackages Method

Returns an array of WSUpsell objects that represent the available upsell packages for a job.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

## GetApplicationMethodTypes

Returns a list of application method types.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

## GetAvailableRecruiters Method

Returns an array of WSRecruiter objects that represent the recruiters that the uploader can post jobs for.

Parameter	Descripton
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

## GetCategories Method

Returns an array of WSCategory objects that represent the available categories that can be set for a job.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

## GetCategoryTerms Method

Returns an array of WSTerm objects that represent the available category terms that can be set for a job.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

sCategoryID	Mandatory string identifying the category ID for which to fetch terms (obtained from the GetCategories method)
-------------	--

## GetJobProperties Method

Returns an array of WSJobProperty objects that represent the available properties that can be set for a job.

Note: JobLocations is the new property that has been exposed to take a CSV list of location ID's which are to be used for the new location offering. The ID's can be found using the GetLocations method described below.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

## GetLocations Method

Returns an array of WSLocation objects that represent the locationID and the label for the locations in the geography database. These location ID's can be used in conjunction with the "JobLocation" property which can be used on the add and update job methods to set the location for the job.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)
searchText	Mandatory string of at least the first 3 characters must be supplied. This is used to filter the return list of locations. The list will be ordered in country preference (defined by the config), followed by largest region (e.g. county, town etc) followed by an alphabetical ordering.

## WSJobProperty Class

Property	Description
MaxLen	Integer specifying the maximum length of the job property
Desc	String description of the job property
Mandatory	Boolean specifying whether the job property must be included or not when posting a job
Name	Name of the job property

## WSCategory Class

Property	Description
MultiSelect	Boolean specifying if the category is multi or single select.
ID	Integer specifying the ID of the category
Mandatory	Boolean specifying whether the category must be included or not when posting a job
Name	Name of the category

## WSTerm Class

Property	Description
ID	Integer specifying the ID of the term
Name	Name or value of the term

## WSRecruiter Class

Property	Description
----------	-------------

ID	Integer specifying the ID of the recruiter
Name	Name of the recruiter

## WSUpsell Class

Property	Description
ID	Integer specifying the ID of the upsell
Name	Name of the upsell

## WSJobPropertyValue Class

Property	Description
Name	Name of the upsell
Value	String value for the job property

## WSLocation Class

Property	Description
LocationID	Integer specifying the ID of the location
Label	Label that will be assigned to that location.

## WSSelectedTerms Class

Property	Description
CategoryID	Integer specifying the ID of the category to which the terms belong
TermIDs	Integer array of the Ids of the terms the job will be assigned to

## GetHeirarchicalCategoryTerms Method

Returns an array of WSHierarchicalTerm objects that represent the available category terms that can be set for a job. This differs from the "GetCategoryTerms" method in that the child terms are marked as such in the response as opposed to a straight list of terms.

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)
sCategoryID	Mandatory string identifying the category ID for which to fetch terms (obtained from the GetCategories method)

## GetSynonyms Method

Returns an array of WSSynonym objects that represent the synonyms available within the site. This will also show the relationships between the synonyms

Parameter	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)

## LogonRepublishArchivedJobLogoff Method

This will allow you to republish an archived job (assuming it has been paid for and has not passed its end date). If it has passed its end date, then the job will be set to the "Expired" status.

Property	Description
sUploaderID	Mandatory string identifying the uploader.
sPassword	Mandatory string. Uploader's password.
sRecruiterID	Mandatory ID of the recruiter

sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to republish.
--------	---

## RepublishArchivedJob Method

This will allow you to republish an archived job (assuming it has been paid for and has not passed its end date). If it has passed its end date, then the job will be set to the "Expired" status.

Property	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method).
sRecruiterID	Mandatory ID of the recruiter
sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to republish.

## LogonExpireJobLogoff Method

This will expire the job by setting the end date of the job to "Now" (the same effect as the end date passing). The job will return a 200 response code on the website, which means that search engines can still get to the job if they have it in their index. The jobseeker cannot apply or perform any actions against the job.

Property	Description
sUploaderID	Mandatory string identifying the uploader.
sPassword	Mandatory string. Uploader's password.
sRecruiterID	Mandatory ID of the recruiter
sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to republish.

## ExpireJob Method

This will expire the job by setting the end date of the job to "Now" (the same effect as the end date passing). The job will return a 200 response code on the website, which means that search engines can still get to the job if they have it in their index. The jobseeker cannot apply or perform any actions against the job.

Property	Description
uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method).
sRecruiterID	Mandatory ID of the recruiter
sJobID	Mandatory string which, coupled with the sRecruiterID parameter, uniquely identifies a job as the job to republish.

## LogonAddJobLogoffWithReturnJobDetail Method

Logs in. Posts a new job immediately to the website using the given parameters. Returns a WSAddJobResult class. This is essentially the same as LogonAddJobLogoff method but more information about the job is returned.

Parameter	Description
jobProperties	Array of WSJobPropertyValue objects specifying job properties
jobCategorization	Array of WSSelectedTerms objects specifying job categorizations
jobUpsells	Array of WSUpsell objects specifying job upsells
sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sRecruiterID	Mandatory ID of the recruiter
sUploaderID	Mandatory string identifying the uploader.

sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sJobID	<p>Mandatory string(20) which should be unique amongst all of that recruiters jobs. This ID, coupled with the specified recruiter allows you to call the update and archive methods for the same job at a later date.</p>
sPassword	<p>Mandatory string. Uploader's password.</p>
productDefinitionID	<p>Optional int specifying the product. If not supplied, it will assume the basic job posting product. Product ID's can be obtained from the GetAvailableProducts method.</p>
sStartDateTime	<p>Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the current time will be used.</p>

sEndTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.
sExternalApplicationURL	Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.
sApplicationEmail	Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.
jobLogo	Optional. Contains two sub-elements <ul style="list-style-type: none"> <li>1. AssetID - If ID of a logo is known then this will assign this asset/logo to the newly created job.</li> <li>2. Base64BlobString - Base 64 encoding of the logo bytes.</li> </ul> <p>If jobLogo is specified then at least one of the above sub-elements must be specified. If both are specified and the AssetID is valid then the logo relating to that AssetID will be used. Otherwise a new Logo will be created using the Base64BlobString. If both elements are specified and empty or in anyway invalid then an error will be thrown with the message Unable to process Job Logo</p>

## AddJobWithReturnJobDetail Method

Posts a new job immediately to the website using the given parameters. Returns a WSAddJobResult class. This is essentially the same as AddJob method but more information about the job is returned.

Parameter	Description
-----------	-------------

uiSecurityToken	Mandatory string identifying your web service session (obtained from the logon method)
sRecruiterID	Mandatory ID of the recruiter
sEmployerID	ID of the employer the job is for, normally 0 (zero) as this functionality is not used from the webservice.
sJobID	Mandatory string (20) which should be unique amongst all of that recruiters jobs. This ID, coupled with the specified recruiter allows you to call the update and archive methods for the same job at a later date.
sStartDateTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the current time will be used.
sEndDateTime	Optional string specifying the start date and time for the job to appear on the web site. This string must be in the format "dd/mm/yyyy hh:nn". If left blank the default will be used.

sApplicationMethod	<p>Mandatory string specifying the method by which applications for this job should be made. Typically the options are:</p> <ul style="list-style-type: none"> <li>• AutoForwardedApplications</li> <li>• ResponseManagement</li> <li>• ExternalRedirect</li> <li>• OfflineApplications</li> </ul> <p>The options may vary between implementations, so check which ones are available using the GetApplicationMethodTypes method as described later in this document.</p> <p>Note: If an application method of 'ExternalRedirect' is specified, the sExternalApplicationURL parameter must be supplied.</p> <p>If an application method of 'AutoForwardedApplications' is specified, the sApplicationEmail parameter must be supplied.</p>
sApplicationEmail	<p>Optional string(500). Must be supplied if application method is set to 'AutoForwardedApplication'. Multiple email addresses can be supplied when separated by a comma.</p>
sExternalApplicationURL	<p>Optional string(255). External URL to which a potential applicant is directed when applying for the job. Must be supplied if application method is set to 'ExternalRedirect'.</p>
jobProperties	<p>Array of WSJobPropertyValue objects specifying job properties</p>
jobCategorization	<p>Array of WSSelectedTerms objects specifying job categorizations</p>
jobUpsells	<p>Array of WSUpsell objects specifying job upsells</p>

productDefinitionID	Optional int specifying the product. If not supplied, it will assume the basic job posting product. Product ID's can be obtained from the GetAvailableProducts method.
jobLogo	<p>Optional. Contains two sub-elements</p> <ol style="list-style-type: none"> <li>1. AssetID - If ID of a logo is known then this will assign this asset/logo to the newly created job.</li> <li>2. Base64BlobString - Base 64 encoding of the logo bytes.</li> </ol> <p>If jobLogo is specified then at least one of the above sub-elements must be specified. If both are specified and the AssetID is valid then the logo relating to that AssetID will be used. Otherwise a new Logo will be created using the Base64BlobString. If both elements are specified and empty or in anyway invalid then an error will be thrown with the message Unable to process Job Logo</p>

## WSAddJobResult Class

Property	Description
JobStatus	JobStatus of the job (e.g. live, awaiting_approval)
JobId	Int specifying the id of the job

# Job Categorisation IDs

Job categorisation data is available through supplied method calls of the webservice, you will need to use these to get the correct categorisation ids.

# Exception Handling

SOAP web services in .Net raise errors through SoapExceptions. For example the flow of code will often be something along the lines of:

```
try
{
    //do some web service stuff
}
catch ( SoapException s )
{
    //use the SoapException object to process the error somehow
}
```

The SOAP specification ("<http://www.w3.org/TR/soap/>") specifies that the 'Details' property of a SoapException object should contain an xml document, and this is precisely how the webservice operates. Within this xml there are two nodes to check if a SoapException is thrown by the webservice. These are 'ERRORNUMBER' and 'ERRORMESSAGE' (nb – these node names are case sensitive). The following two lines demonstrate how to obtain the error information:

```
int iNum = Int32.Parse(s.Detail["ERRORNUMBER"].InnerText);
string sMsg = s.Detail["ERRORMESSAGE"].InnerText;
```

You may wish to then perform a Switch statement on the error number and process it in a way you see fit. In the event that you receive an 'Internal Server Error', you should record as much of the information as possible that caused the error (date, time, your credentials, other parameters etc) and email your contact indicating which webservice method you were trying to call.

# Code Example

The following code excerpt gives an example of calling the “Logon” method

```
try
{
    // Create the proxy object
    JobsWebService.JobsWebService_test wsJob;
    w = new JobsWebService.JobsWebService_test();
    string sLoginID = "D845A85D-4BF9-42F2-9CDC-FCCA49B14824";
    string sPassword = "49C20F07-E5A8-4661-8870-068164A4BD1B";
    // Logon and store the returned web service session ID for subsequent calls
    string sSessionID = wsJob.Logon(sLoginID,sPassword);
}
catch ( SoapException s )
{
    int iNum = Int32.Parse(s.Detail["ERRORNUMBER"].InnerText);
}
```

# Location Changes

Madgex have recently changed the way that the location service works. If you are using the new location offering from Madgex, then the soap calls you make need to be changed as per the below.

## Legacy Location SOAP Call

Even if the job board has the new location offering enabled, the original SOAP job posting request will still work (unless a brand new site, which will not have existing location terms anyhow).

This is an example of the existing SOAP call:

```
<job:jobCategorization>
...
<!-- Start: Location Category -->
<job:WSSelectedTerms>
  <job:CategoryID>101</job:CategoryID>
  <job:TermIDs>
    <job:int>10125</job:int>
    <job:int>10137</job:int>
  </job:TermIDs>
</job:WSSelectedTerms>
<!-- End: Location Category -->
</job:jobCategorization>
```

In this example:

- Location Category ID = 101
- Term ID 10125 = London (Central)
- Term ID 10137 = Brighton

Note: These category and term ID's are specific to each jobboard

## New Location SOAP Call

In order to make use of the new locations, you must use the JobLocations properties inside the jobProperties section of the call.

```
<job:jobProperties>
...
<!-- New Location Ids -->
<job:WSJobPropertyValue>
  <job:Name>JobLocations</job:Name>
  <job:Value>1500,1655</job:Value>
</job:WSJobPropertyValue>
</job:jobProperties>
```

In this example:

- Term ID 1500 = London (Central)
- Term ID 1655 = Brighton

Note: There is no need to supply the category ID and the term ID's are global, so they would be the same for any Madgex job board that is running the latest location offering. By default, up to 4 locations can be added for each job, but this can be changed per job board.

You need not supply a value in the <job:jobCategorization> node as this is the old method of adding locations.

## Getting the Location IDs

You can access the ID for a specific location by calling the GetLocations method. This method will allow you to search the location based on at least the first 3 letters of the location. The more of the location word you search on, the more likely you are to get an accurate result.

You can also request that Madgex send you an XML feed of the latest version of the Global Geography list if you want to maintain your own database of Madgex location ID's. The ID's will not change, but more may be added by Madgex over time in order to accommodate more countries and locations that have been added to specific countries.