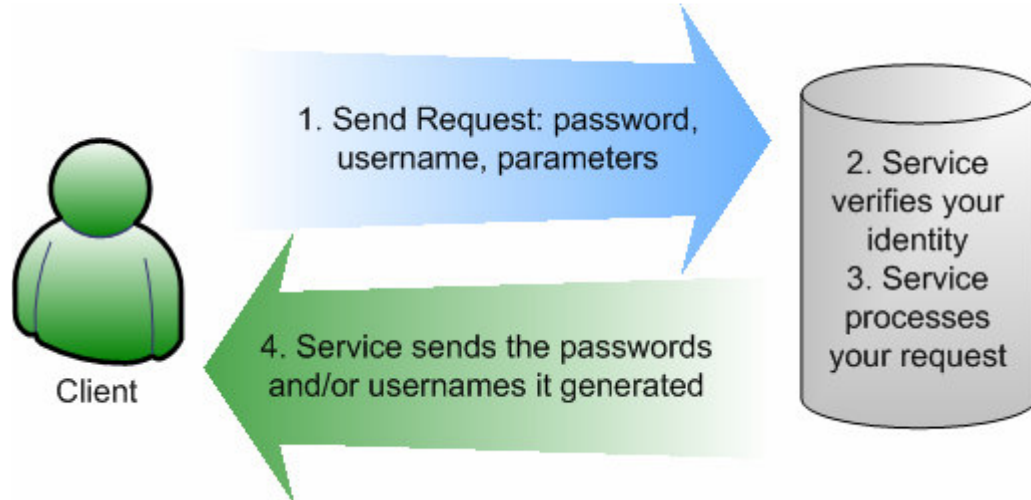


# Diplodock Password Generator Web Service Documentation

## Part I – Introduction

Password Generator Web Service is an online service that lets you generate random passwords, serial numbers, pins, and usernames 24/7. No matter where you are, which platform or application you are using, you will always be able to access this service! It is a professional, reliable, and secure tool for password and username generation that can be used on any computer, platform, or application! It is a platform and application-independent solution for controlled generation of secure random passwords and usernames.

The basic principle is that you send a request to the Password Generator Web Service in either HTTP POST (HTML Form), HTTP GET (URL), or SOAP form. The service then authenticates you and processes the request, if you supplied the correct username and password (which you will be given once you have paid for the service). Once it has processed the request, it sends back the passwords and/or usernames it generated, according to the criteria, you specified in your request.



Please note that you can use this service only during the time period for which you paid. Once that period has elapsed, the service will return “Your account has expired.” message.

## Part II – Calling the Service

Password Generator Web Service is located at <http://www.diplodock.com/PGWS/pgservice.asmx>. There are 3 ways to call the service: SOAP, HTTP GET, and HTTP POST. Detailed information about the syntax can be obtained at the above URL.

Here we will discuss only HTTP GET method of calling the Password Generator Web Service. Here is a sample request (note it is simply a URL):

```
http://www.diplodock.com/PGWS/pgservice.asmx/Generate?vUser=username&vPassword=\*\*\*\*\*&vConf=default&vCount=500&vMode=m&vMask=nnunu\[-\]ununu\[-\]uuunu&vUMask=Ann
```

This HTTP GET request consists of 4 parts (in fact, any request to the Password Generator Web Service will consist of 4 parts):

- The URL of the service (in blue) → Access the service.
- The function (operation) (in red) → You can call either Generate, GenerateRandom, GeneratePronounceable, and GenerateMask functions. They are described in detail later.
- Login information (in green) → You can only access the service if you paid for it and are a registered user, and your account has not been expired. If you provide incorrect login information, no passwords will be generated, and the service will return “Invalid username or password.” or “Your account has expired.”.
- Other parameters that depend on the function you are calling (in orange) → These are typically parameters that are not included in the configuration file.

### Part III – Four Functions

The Password Generator Web Service exposes 4 different functions that you can call to generate passwords. All 4 functions accept various parameters through which you customize how passwords and usernames will be generated (i.e. their length, character content, etc). All parameters are required. Overall, different functions have different parameters. However, some parameters are shared amongst all 4:

Parameter	Description	Values	Example
vUser	Your username and password to access the service	Whatever username and password you were assigned	username
vPassword			password
vCount	Number of passwords to generate	0<x<100,000; integer	outcfg1.ini

All other parameters are unique to each function and will be described below.

The four functions and their unique parameters are outlined below:

- **Generate** – This is the most customizable function. That it gives you the most flexibility and control over passwords and usernames generated. It has the following parameters:  
**vUser=string&vPassword=string&vConf=string&vCount=string&vMode=string&vMask=string&vUMask=string**

Parameter	Description	Values	Example
vConf	Name of the configuration file that defines the rules for password generation. The actual file is stored on the server. You can create your own configuration files with rules to suit your needs. There is a “default” configuration file that all users can use	One of the names of the configuration files that you created, or the “default”.	myconf file

vMode	Generation mode	“p” – password “m” – mask “n” – pronounceable passwords	p
vMask	If vMode parameter is set to “m”, this mask will be used to generate passwords	Characters allowed: 1, 2, 3, 4, 5, 6, 7, 8, 9, l, u, s, n, A, B, C, D, [, ], any single character within []	nuun[-]llun
vUMask	If this is undefined, no usernames will be generated. If this parameter is given, usernames will be generated according to mask specified. Note, if usernames will be generated, in the output array password-username combination will be stored as “password::::username”, i.e. separated by “::::”.		Ann

- GenerateRandom** – Use this function, if you need to generate random passwords, without using masks or dictionaries. It has the following parameters:
   
**vUser=string&vPassword=string&vCount=string&vMinLen=string&vMaxLen=string&vGroupOrder=string&vUCaseChars=string&vUCaseDensity=string&vLCaseChars=string&vLCaseDensity=string&vNChars=string&vNDensity=string&vSChars=string&vSDensity=string&vSimilarPasswords=string&vNoSimChars=string&vRandomSeed=string&vCPFilter=string**

Parameter	Description	Values	Example
vMinLen	Minimum and maximum lengths of a password	[4-100]	5
vMaxLen			8
vGroupOrder	Determines how characters from character groups are arranged in a password and whether they are mixed together. Parentheses indicate that characters will be mixed.	Characters allowed: l, u, n, s, 1, 2, 3, 4, 5, 6, 7, 8, 9, (, )	(lus)n
vUCaseChars, vLCaseChars, vNChars, vSChars	These determine what characters will appear in a password for upper case, lower case, numerical and special character groups respectively.	Any set of characters	0123569
vUCaseDensity,	These determine how	[0-10]	5

vLCaseDensity, vNDensity, vSDensity	often characters from upper case, lower case, numerical and special character groups respectively, will appear in a password.		
vSimilarPasswords	Similar passwords will be generated.	True/False	False
vNoSimChars	No similar characters like o, O, and 0 will appear in passwords	True/False	False
vRandomSeed	Random seed to use to generate passwords	Any integer	123
vCPFilter	Determines if passwords will be checked against a weak password filter	True/False	True

- GeneratePronounceable** – Use this function if you want to generate easy-to-remember pronounceable passwords. It has the following parameters:  
**vUser=string&vPassword=string&vCount=string&vMinLen=string&vMaxLen=string  
&vVowels=string&vConsonants=string&vDoubleConsonants=string&vNumbers=string  
&vNumbersCount=string&vNumbersAfter=string&vLetterCaseMask=string&vNoSimChars=string&vRandomSeed=string&vCPFilter=string**

Parameter	Description	Values	Example
vMinLen		See above	
vMaxLen			
vVowels, vConsonants, VDoubleConsonants, vNumbers	Similar to vUCaseChars – see above		
vNumbersCount	Determines how many numbers, if any, will be included in a password.	Integer $0 \leq x \leq (\text{min. length} - 3)$	2
vNumbersAfter	If True, numbers will appear at the end, otherwise at the beginning	True/False	True
vLetterCaseMask	This tweaks the case of letters in a password according to mask.	Characters allowed: A, a, (, )	a (A) a
vNoSimChars, vRandomSeed, vCPFilter	See above		

- GenerateMask** – Use this function, if you need to generate passwords, according to mask. It has the following parameters:

**vUser=string&vPassword=string&vCount=string&vMask=string&vUCCaseChars=string&vLCaseChars=string&vNChars=string&vSChars=string&vNoSimChars=string&vRandomSeed=string&vCPFilter=string**

All these parameters have been already described. See above for details.

## Part IV - Configuration Files

Configuration files store various options and settings for password and username generation in a convenient way. They give you the most control over passwords and usernames being generated. There is a “default” configuration file that all users can use. However, you can define your own configuration files in the User Center at <http://www.diplodock.com/Products/PasswordGenerator/WebService/usercenter.aspx>, to which you should login using the username and password you were provided with at the time you purchased the service.

These files use standard ini file syntax, and can be easily edited with any text editor.

Syntax:

```
[SectionName]
KeyName = Value //Comment
```

Below is a description of parameters used in configuration files.

Section	Key	Description	Values	Example
Options	RandomSeed	Random seed that will be used to generate passwords and usernames	An integer larger than -1	0
	NoSimilarCharacters	No similar characters like o, O, and 0 will appear in passwords	“0” – False “1” – True	1
	ConcatenateBefore	Concatenate words according to concatenation mask before applying letter case mask	“0” – False “1” – True	1
	UseCPFilter	Determines if common passwords filter will be used to filter out possibly weak or common passwords	“0” – False “1” – True	1
	UniqueUsernames	Determines if only unique usernames will be generated	“0” – False “1” – True	0
	UniquePasswords	Determines if only unique passwords	“0” – False “1” – True	0

		will be generated		
	SimilarPasswords	Determines if similar passwords will be generated. More information about this option is available in Password Generator 2004 Professional Documentation.	“0” – False “1” – True	0
PasswordOptions	ExactApproximation	These properties are described in the Password Generator 2004 Professional Documentation	“0” – False “1” – True	1
	PasswordLength		Min,max [4-100],[4-100]	5, 8
	GroupOrder		Characters allowed: 1, u, n, s, 1, 2, 3, 4, 5, 6, 7, 8, 9, (, )	(lu)ns
PronounceableOptions	PasswordLength	property as for PasswordOptions		
	LetterCaseMask	This property is described in the Password Generator 2004 Professional Documentation	Characters allowed: A, a, (, )	a
	NumbersCount	This property determines how many numbers will appear in the pronounceable password	Integer $0 \leq x \leq (\text{min. length} - 3)$	2
	NumbersPosition	This property determines whether numbers will appear before or after the letters in a pronounceable password	“0” – Before “1” – After	1
	Vowels	Characters that will be included in a password	Any character (depends on the type of the property).	aeiouy
	Consonants			bcd fghjklmnp qrstvwxyz
	DoubleConsonants			cd fglmnp rst
	Numbers			0123456789
LowerCase / UpperCase / Numbers / Special [built-in character groups]	Characters	Characters that will be included in the character group	Any character (depends on the type of character group). → <b>Note:</b> You can use one character more than once; - such character will appear more often.	abcdefghijklmnopqrstuvwxyz

	Density	This determines how often character group's characters will appear in the password	An integer from 1 to 10.	5
1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 [custom character groups]	Characters & Density properties as for built-in character groups			
	Name	Name of the custom character group	Any one word	Vowels
dDefault / dGieven / dFamily / dMovie [built-in dictionaries]	Load	This determines if dictionary will be loaded. This can reduce / increase load speed of a program	"0" – do not load "1" - load	1
	WordLength	These properties are described in the Password Generator 2004 Professional Documentation	Min,max [0-100],[0-100]	0, 100
	Filter		* - denotes any character	a*
	LetterCaseMask		Characters allowed: A, a, (, )	a
	Concatenate		Before,after (!)[0-100],(!)[0-100]	0, ! 3

Note: custom dictionaries are currently unsupported!

## Part V - Output

Here is an example of what Password Generator Web Service will return:

```
<?xml version="1.0" encoding="utf-8" ?>
- <ArrayOfString xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.diplodock.com/PGWS">
<string>UbaNr9o19Y</string>
<string>UEa818Vw1f</string>
<string>pRG4hfK5x5</string>
...
<string>trI4CG12ur</string>
<string>WAvyr3F2v8</string>
<string>C6cK8xH9ds</string>
</ArrayOfString>
```