

gpGeocoder Documentation



Documentation Overview

release notes gpGeocoder-1.8.25.0-SNAPSHOT

Geocoder Configuration and Profiles

The geocoder can be configured using parameters that have one of the types string, integer (i.e. long) or boolean. A set of parameters is called a profile and can be stored using an **XML** format. The gpGeocoder comes along with a number of predefined profiles that cover common use cases. You can find them in the directory **delivery/profiles**. With the dynamic library interface, they can be loaded using the methods `LoadProfile` or `LoadProfileFromString`. If the geocoder is statically linked, the parameters can be accessed using the class **CParameters**. The required string constants of the parameter keys can be found in **APICConst.h**. To give an example, here is an extract of such an **XML** file:

```
<LocatingEngineParameters>
  <SearchParameters>
    <Town.Active Value="1" Type="bool"/>
    <City.ReturnAllCity2 Value="0" Type="bool"/>
    <Engine.MaxResults Value="7500" Type="long"/>
    <!-- [...] -->
  </SearchParameters>
</LocatingEngineParameters>
```

Town Search Parameters

The geocoder can operate in two major modes: Town search or city search. The city search is the classic geocoder algorithm. Town search is an improved search algorithm that uses information from postcode, city and city2 input simultaneously. Town search can only be used if town indices are available in the used map. It must also be activated in the profile. This can be done with the parameter **Town.Active**. Some parameters exist twice because they have different values for town and city search. If town search is active, the town parameter is used, otherwise the city parameter is used. The town parameter is labeled with an additional prefix **Town** in the parameter name.

Incremental Profiles

It is possible to use incremental profiles that define only some of the parameters. For this, first a complete profile should be loaded. Then, one or more incremental Profiles can be loaded. They overwrite only the parameters they have, all other parameters remain as they were in the original profile. There is one exception: The parameter **Town.Active** is set to false if it is not defined in the incremental profile. The reason for this behaviour is to support older profiles that do not define the parameter **Town.Active**.

Parameter Overview

There is a large number of parameters. They can be distinguished in two groups: **internal** and **external** parameters. External parameters can configure the main features of the geocoder. They can be modified by every user to set up the geocoders behaviour. Until now, internal parameters should only be modified by users that have an intense internal knowledge of the geocoder algorithm. It is a shortcoming that they are still visible for everyone. In the following, a description for every parameter is given. Internal parameters are marked as such.

Parameter	Description
Town.Active <i>Only for town search</i> Type: <i>Bool</i> APIEnum: 186	Determines if town search should be used. If it is true, and map and engine support it, a town search will be executed. If the parameter is false, or missing, or map or engine do not support town search, the old city search will be used. See CLocatingEngine::hasProfileWithActivatedTownSearch() and CLocatingEngine::hasTownData() or the <code>HasProperty()</code> gpGeocoderDll method for how to find out which kind of search will be executed. Note that this parameter behaves special in conjunction with Incremental Profiles
AddressFormat Type: <i>String</i> APIEnum: 13	The address can be given as one single string. This parameter defines how to parse that string. For example the input <code>%Country%~%PostalCode%~%City%~%City2%~%Street%~%HouseNr%~%~</code> defines a format where the character <code>~</code> separates the fields.
AddressInputMode Type: <i>Long</i> APIEnum: 12	Configures the location of the input that is used for geocoding. There are two modes that can be configured. <ul style="list-style-type: none">> In standard mode (0) the input is retrieved from separate fields. That means separate keys for each field (City, PostalCode, ...) must be used to set the input values.> In address field mode (1) one address input is parsed to obtain the separate fields. The format of the second mode is configured with the parameter AddressFormat.
AsteriskMode <i>Only for city search</i> Type: <i>Long</i> APIEnum: 117	This is a deprecated parameter. We recommend to use the <code>pareto filter</code> instead. Information about it can be found in the section <code>ConceptParetoFilter</code> . An asterisk result has additional characters at the end. For example Karlsruhe is an asterisk result for the input Karl. This parameter configures if such results should be filtered. If it is enabled, asterisk results are only filtered if better results remain. <ul style="list-style-type: none">> If the asterisk mode is disabled (0), no filtering is done.> The mode exact fragments (1) allows additional words.> The mode exact (3) filters results with additional words.

CheckArtificialTownCharacterization Type: <i>Bool</i> APIEnum: 165	There are artificial towns in the data that are derived from regions. If this parameter is enabled, for all results the information if it is an artificial town is determined. Otherwise, this fact remains unknown in the result.
City.OmitCity2 Type: <i>Bool</i> APIEnum: 152	If this parameter is set, city2 results are not returned if their city is also part of the result and has the same postcode.
City.ReturnAllCity2 Type: <i>Bool</i> APIEnum: 100	This parameter specifies if for each result on City or City2 level also the list of its children shall be returned in the result list. In TownSearches, you can separately configure if postcodes or city2 children shall be returned using the parameters GetAllCity2 and GetAllPostcode.
Country.CodeType Type: <i>Long</i> APIEnum: 56	Parameter which specifies what input format is given for the country field. The output country field is also filled in this format. Possible values are: <ul style="list-style-type: none"> > 0 = Name > 1 = ISO2 > 2 = ISO3 > 3 = CountryCodePlate > 4 = DialingCode > 5 = RegionAbbreviation
Engine.CitySearchTimeout Type: <i>Long</i> APIEnum: 112	Milliseconds the search is allowed to run before the street search starts. If this part of the search takes longer it is aborted. The value -1 indicates that the time is not limited.
Engine.MaxResults Type: <i>Long</i> APIEnum: 123	The maximum number of results. If there are more results, all additional results are not returned. Note that this number refers not only to the final number of results, but also to the number of results of internal tasks. So, a low value of this parameter can produce results that can only be explained with knowledge about the internal geocoder algorithms.
Engine.StopOnResultListOverflow Type: <i>Bool</i> APIEnum: 124	Determines if the search should stop immediately if the capacity of the result list is exceeded. The capacity is configured in the parameter Engine.MaxResults .
Engine.StreetSearchTimeout Type: <i>Long</i> APIEnum: 102	Overall time in Milliseconds the search is allowed to run before it is aborted. The value -1 indicates that the time is not limited.
FieldWeight.City Town.FieldWeight.City Type: <i>Long</i> APIEnum: 74 / 207	<i>For internal use only.</i>
FieldWeight.City2 Town.FieldWeight.City2 Type: <i>Long</i> APIEnum: 75 / 208	<i>For internal use only.</i>
FieldWeight.HouseNr Town.FieldWeight.HouseNr Type: <i>Long</i> APIEnum: 78 / 211	<i>For internal use only.</i>
FieldWeight.Postcode Town.FieldWeight.Postcode Type: <i>Long</i> APIEnum: 76 / 209	<i>For internal use only.</i>
FieldWeight.Street Town.FieldWeight.Street Type: <i>Long</i> APIEnum: 77 / 210	<i>For internal use only.</i>
Town.GetAllCity2 <i>Only for town search</i> Type: <i>Bool</i> APIEnum: 194	Whether or not to get city2 records when returnAllCity2 is switched on. There are four kinds of records in the data: <ul style="list-style-type: none"> > Cities (without postcode), e.g. 76*** Karlsruhe > Postcodes of cities, e.g. 76131 Karlsruhe > City districts, e.g. 76*** Karlsruhe Durlach > Postcodes of city districts, e.g. 76227 Karlsruhe Durlach If the search finds the first one, and ReturnAllCity2 is turned on, then the parameter steers if the third and fourth may be in the result list. The second and fourth can only be in the result list if the parameter TownGetAllPostcode is switched on.
Town.GetAllPostcode <i>Only for town search</i> Type: <i>Bool</i> APIEnum: 193	Whether or not to get postcode records when returnAllCity2 is switched on. There are four kinds of records in the data: <ul style="list-style-type: none"> > Cities (without postcode), e.g. 76*** Karlsruhe > Postcodes of cities, e.g. 76131 Karlsruhe > City districts, e.g. 76*** Karlsruhe Durlach > Postcodes of city districts, e.g. 76227 Karlsruhe Durlach If the search finds the first one, and ReturnAllCity2 is turned on, then the parameter steers if the second and fourth may be in the result list. The third and fourth can only be in the result list if the parameter TownGetAllCity2 is switched

	on.
HNr.DetectRanges Type: <i>Bool</i> APIEnum: 149	Determines if the geocoder should look for house number ranges in the input.
HNr.FilterDuplicates Type: <i>Bool</i> APIEnum: 153	It is possible, that the geocoder can find the same house number more than once. This parameter determines if these duplicates should be filtered and only one of them remains.
HNr.Offset Type: <i>Long</i>	This offset determines the distance in meters for the displacement of the house number coordinates. House numbers coordinates are obtained by an interpolation of its street polygon. With this, we get a coordinate on the street. Then, the coordinate is displaced to one side, depending on the house number structure (odd or even)
HNr.RangeSeparators Type: <i>String</i> APIEnum: 150	Determines the characters sequences that are used to separate house number ranges. The separators itself are here separated with the special character "␣" (0xA4).
Intersections.Enable Type: <i>Bool</i> APIEnum: 129	Specifies whether the search for crossings is enabled. A crossing search also requires that one of the separators is somewhere in the middle of the street input string. The separators can be specified in Paramter Intersections.Separators .
Intersections.Separators Type: <i>String</i> APIEnum: 130	Specifies the text that can act as an crossing separator. Contains a list of strings separated by Character '␣'(0xA4) that, if entered in the street field indicate that a crossing is sought. If, for example the separators are "\␣/" then 'KarlAllee' and 'Karl/Allee' will trigger a crossing search, but 'Karl @ Allee' will not. This requires that Intersections.Enable is set to true
MultiwordIndex.Enable Type: <i>Bool</i> APIEnum: 109	Specifies whether or not the multiword index shall be used. Affects the search for city, city2 and streetnames. However, the search for towns in the town search uses always the multiword index.If the multiword index is not used, the input may not have additional words at the beginning or in the middle. This parameter is usually always true.
ParetoFilter.TreatAdditionalWordsAsExact Town.ParetoFilter.TreatAdditionalWordsAsExact Type: <i>Bool</i> APIEnum: 182 / 212	Specifies whether or not the pareto filter categories AdditionalWord and Exact shall be the same. For additional information about pareto filtering see ConceptParetoFilter.
ParetoFilter.TreatAsteriskAsAdditonalWord Town.ParetoFilter.TreatAsteriskAsAdditonalWord Type: <i>Bool</i> APIEnum: 183 / 213	Specifies whether or not the pareto filter categories Asterisk and AdditionalWord shall be the same. For additional information about pareto filtering see ConceptParetoFilter.
ParetoFilter.TreatAsteriskWithAdditionalWordsAsAsterisk Town.ParetoFilter.TreatAsteriskWithAdditionalWordsAsAsterisk Type: <i>Bool</i> APIEnum: 184 / 214	Specifies whether or not the pareto filter categories AsteriskWithAdditionalWord and Asterisk shall be the same. For additional information about pareto filtering see ConceptParetoFilter.
ParetoFilter.TreatNoMatchAsAsteriskWithAdditionalWords Town.ParetoFilter.TreatNoMatchAsAsteriskWithAdditionalWords Type: <i>Bool</i> APIEnum: 185 / 215	Specifies whether or not the pareto filter categories NoMatch and AsteriskWithAdditionalWord shall be the same. For additional information about pareto filtering see ConceptParetoFilter.
Penalty.AddCity2ToResultList <i>Only for city search</i> Type: <i>Long</i> APIEnum: 51	<i>For internal use only.</i>
Penalty.AddCityToResultList <i>Only for city search</i> Type: <i>Long</i> APIEnum: 50	<i>For internal use only.</i>
Penalty.AddHouseNrToResultList Town.Penalty.AddHouseNrToResultList Type: <i>Long</i> APIEnum: 54 / 202	<i>For internal use only.</i>
Penalty.AddIntersectionToResultList Town.Penalty.AddIntersectionToResultList Type: <i>Long</i> APIEnum: 135 / 201	<i>For internal use only.</i>
Penalty.AddPostcodeToResultList <i>Only for city search</i> Type: <i>Long</i> APIEnum: 53	<i>For internal use only.</i>
Penalty.AddStreetToResultList Town.Penalty.AddStreetToResultList Type: <i>Long</i> APIEnum: 52 / 199	<i>For internal use only.</i>
Town.Penalty.AddTownToResultList <i>Only for town search</i>	<i>For internal use only.</i>

Type: <i>Long</i> APIEnum: 198	
Penalty.AsteriskResult <i>Only for city search</i> Type: <i>Long</i> APIEnum: 144	<i>For internal use only.</i>
Penalty.BinaryCity2Search <i>Only for city search</i> Type: <i>Long</i> APIEnum: 47	<i>For internal use only.</i>
Penalty.BinaryCitySearch <i>Only for city search</i> Type: <i>Long</i> APIEnum: 43	<i>For internal use only.</i>
Penalty.BinaryGlobalCity2Search <i>Only for city search</i> Type: <i>Long</i> APIEnum: 44	<i>For internal use only.</i>
Penalty.BinaryGlobalPostcodeSearch <i>Only for city search</i> Type: <i>Long</i> APIEnum: 45	<i>For internal use only.</i>
Penalty.BinaryIntersectionStreetSearch Town.Penalty.BinaryIntersectionStreetSearch Type: <i>Long</i> APIEnum: 131 / 196	<i>For internal use only.</i>
Penalty.BinaryPostcodeSearch <i>Only for city search</i> Type: <i>Long</i> APIEnum: 46	<i>For internal use only.</i>
Town.Penalty.BinarySearch <i>Only for town search</i> Type: <i>Long</i> APIEnum: 187	<i>For internal use only.</i>
Penalty.BinaryStreetSearch Town.Penalty.BinaryStreetSearch Type: <i>Long</i> APIEnum: 48 / 195	<i>For internal use only.</i>
Penalty.FuzzyCity2Search <i>Only for city search</i> Type: <i>Long</i> APIEnum: 63	<i>For internal use only.</i>
Penalty.FuzzyCitySearch <i>Only for city search</i> Type: <i>Long</i> APIEnum: 61	<i>For internal use only.</i>
Penalty.FuzzyGlobalCity2Search <i>Only for city search</i> Type: <i>Long</i> APIEnum: 60	<i>For internal use only.</i>
Penalty.FuzzyIntersectionStreetSearch Town.Penalty.FuzzyIntersectionStreetSearch Type: <i>Long</i> APIEnum: 133	<i>For internal use only.</i>
Penalty.FuzzyPostcodeSearch <i>Only for city search</i> Type: <i>Long</i> APIEnum: 64	<i>For internal use only.</i>
Town.Penalty.FuzzySearch <i>Only for town search</i> Type: <i>Long</i> APIEnum: 218	<i>For internal use only.</i>
Penalty.FuzzyStreetSearch Town.Penalty.FuzzyStreetSearch Type: <i>Long</i> APIEnum: 62	<i>For internal use only.</i>
Penalty.GetAllCity2 <i>Only for city search</i> Type: <i>Long</i> APIEnum: 115	<i>For internal use only.</i>
Penalty.GetAllPostcode <i>Only for city search</i> Type: <i>Long</i>	<i>For internal use only.</i>

APIEnum: 116	
Penalty.GlobalPostcodeAddMissing0Base Town.Penalty.GlobalPostcodeAddMissing0Base Type: <i>Long</i> APIEnum: 156 / 203	<i>For internal use only.</i>
Penalty.GlobalPostcodeAddMissing0Increment Town.Penalty.GlobalPostcodeAddMissing0Increment Type: <i>Long</i> APIEnum: 157 / 204	<i>For internal use only.</i>
Penalty.GlobalPostcodeCuttingBase Town.Penalty.GlobalPostcodeCuttingBase Type: <i>Long</i> APIEnum: 125 / 205	<i>For internal use only.</i>
Penalty.GlobalPostcodeCuttingIncrement Town.Penalty.GlobalPostcodeCuttingIncrement Type: <i>Long</i> APIEnum: 126 / 206	<i>For internal use only.</i>
Penalty.HouseNumberRangeFirstNr Type: <i>Long</i> APIEnum: 146	<i>For internal use only.</i>
Penalty.HouseNumberRangeSecondNr Type: <i>Long</i> APIEnum: 147	<i>For internal use only.</i>
Penalty.HouseNumberSearch Type: <i>Long</i> APIEnum: 49	<i>For internal use only.</i>
Penalty.MaxDynamic Type: <i>Long</i> APIEnum: 79	<i>For internal use only.</i>
Penalty.PhoneticCity2Search <i>Only for city search</i> Type: <i>Long</i> APIEnum: 69	<i>For internal use only.</i>
Penalty.PhoneticCitySearch <i>Only for city search</i> Type: <i>Long</i> APIEnum: 67	<i>For internal use only.</i>
Penalty.PhoneticGlobalCity2Search <i>Only for city search</i> Type: <i>Long</i> APIEnum: 68	<i>For internal use only.</i>
Penalty.PhoneticIntersectionStreetSearch Town.Penalty.PhoneticIntersectionStreetSearch Type: <i>Long</i> APIEnum: 132 / 220	<i>For internal use only.</i>
Town.Penalty.PhoneticSearch <i>Only for town search</i> Type: <i>Long</i> APIEnum: 188	<i>For internal use only.</i>
Penalty.PhoneticStreetSearch Town.Penalty.PhoneticStreetSearch Type: <i>Long</i> APIEnum: 66 / 219	<i>For internal use only.</i>
Penalty.PostcodeAddMissing0Base <i>Only for city search</i> Type: <i>Long</i> APIEnum: 154	<i>For internal use only.</i>
Penalty.PostcodeAddMissing0Increment <i>Only for city search</i> Type: <i>Long</i> APIEnum: 155	<i>For internal use only.</i>
Penalty.PostcodeCuttingBase <i>Only for city search</i> Type: <i>Long</i> APIEnum: 113	<i>For internal use only.</i>
Penalty.PostcodeCuttingIncrement <i>Only for city search</i> Type: <i>Long</i> APIEnum: 114	<i>For internal use only.</i>
Town.Penalty.PostcodeOnlySearch <i>Only for town search</i> Type: <i>Long</i>	<i>For internal use only.</i>

APIEnum: 190	
Penalty.StateMismatch Type: <i>Long</i> APIEnum: 128	<i>For internal use only.</i>
Town.Penalty.StreetAsteriskResult <i>Only for town search</i> Type: <i>Long</i> APIEnum: 192	<i>For internal use only.</i>
Penalty.StreetHNrCombination.SplitHNr Town.Penalty.StreetHNrCombination.SplitHNr Type: <i>Long</i> APIEnum: 136	<i>For internal use only.</i>
Penalty.StreetHNrCombination.Street1AddOn Town.Penalty.StreetHNrCombination.Street1AddOn Type: <i>Long</i> APIEnum: 137	<i>For internal use only.</i>
Penalty.StreetHNrCombination.Street2AddOn Town.Penalty.StreetHNrCombination.Street2AddOn Type: <i>Long</i> APIEnum: 138	<i>For internal use only.</i>
Penalty.StreetHNrCombination.UseCompleteInput Town.Penalty.StreetHNrCombination.UseCompleteInput Type: <i>Long</i> APIEnum: 139	<i>For internal use only.</i>
Town.Penalty.StreetUnionSearch <i>Only for town search</i> Type: <i>Long</i> APIEnum: 217	<i>For internal use only.</i>
Penalty.SwapAndSplit.SingleAffix <i>Only for city search</i> Type: <i>Long</i> APIEnum: 143	<i>For internal use only.</i>
Penalty.SwapAndSplit.Split <i>Only for city search</i> Type: <i>Long</i> APIEnum: 98	<i>For internal use only.</i>
Penalty.SwapAndSplit.Swap <i>Only for city search</i> Type: <i>Long</i> APIEnum: 99	<i>For internal use only.</i>
Town.Penalty.TownAsteriskResult <i>Only for town search</i> Type: <i>Long</i> APIEnum: 191	<i>For internal use only.</i>
Town.Penalty.UnionSearch <i>Only for town search</i> Type: <i>Long</i> APIEnum: 189	<i>For internal use only.</i>
Postcode.Aggregate Type: <i>Bool</i> APIEnum: 105	<i>For internal use only.</i>
Postcode.LengthDifferenceMalus Type: <i>Long</i> APIEnum: 84	<i>For internal use only.</i>
Postcode.Missing0Malus Type: <i>Long</i> APIEnum: 83	<i>For internal use only.</i>
Postcode.NumericalMalus Type: <i>Long</i> APIEnum: 81	<i>For internal use only.</i>
Postcode.RepPostcodeMalus Type: <i>Long</i> APIEnum: 85	<i>For internal use only.</i>
Postcode.SpaceMalus Type: <i>Long</i> APIEnum: 82	<i>For internal use only.</i>
Preprocessor.WordSeparators Type: <i>String</i> APIEnum: 16	A list of characters that are treated as separators.
Region.CheckState Type: <i>Bool</i> APIEnum: 127	<i>For internal use only.</i>

Region.CodeType Type: <i>Long</i> APIEnum: 151	Parameter which specifies what input format is given for the state field. The output state field is also filled in this format. Possible values are: > 0 = Name > 1 = ISO2 > 2 = ISO3 > 3 = CountryCodePlate > 4 = DialingCode > 5 = RegionAbbreviation
Result.Language Type: <i>String</i> APIEnum: 58	Determines the language of the result addresses. If given language is available this language will be used. Otherwise the default language is used. Parameter is expected in PTVLanguageCode which is similar to MARCLanguage code (e.g. DFT=default, GER=german, FRE=french).
Score.MinFieldScore Type: <i>Long</i> APIEnum: 14	<i>For internal use only.</i>
Score.MinPostcodeScore Type: <i>Long</i> APIEnum: 86	<i>For internal use only.</i>
Search.Binary Type: <i>Bool</i> APIEnum: 70	Determines if the binary search should be used. With this, only results that match exact can be found.
Search.Fuzzy Type: <i>Bool</i> APIEnum: 71	Determines if the fuzzy search should be used. With this, arbitrary spelling errors can be corrected. This search type is only supported by city search. Town search is not supported.
Search.Phonetic Type: <i>Bool</i> APIEnum: 72	Determines if the phonetic search should be used. With this, phonetic spelling errors can be corrected.
Town.SingleFieldSeparators <i>Only for town search</i> Type: <i>String</i> APIEnum: 223	TODO: Add documentation
Street.HNrPosition Type: <i>Long</i> APIEnum: 94	Position where the house number is searched for: 0= there is a separate field for the house number 1= before the street or in a separate field 2= after the street or in a separate field 3= before or after the street or in a separate field
Street.ReturnAllHNr Type: <i>Bool</i> APIEnum: 55	This parameter specifies if for each result on street level also the list of all its house numbers shall be returned in the result list.
Street.SplitHNr Type: <i>Bool</i> APIEnum: 93	This parameter specifies if the street field of the input address may contain house numbers In this case the algorithm will try to separate the house number from the street. You can use the parameter Street.HNrPosition to specify where in the street field the house numbers are sought.
SwapAndSplitMode Type: <i>Long</i> APIEnum: 95	Parameter to specify which swap and split combinations shall be used. Possible values are: 0=no swap and split operation at all 1=swap city and city2 2=split city and city2 3=swap and split city and city2
Weight.MinSubwordWeight Type: <i>Long</i> APIEnum: 15	<i>For internal use only.</i>
Weight.MissingWordFactor Type: <i>Long</i> APIEnum: 158	<i>For internal use only.</i>
Weight.Threshold Type: <i>Long</i> APIEnum: 110	<i>For internal use only.</i>