

Files

7abe42d

Go to file





















- .config
- .github
- docs
- eng
- src
  - coreclr
    - scripts
    - src
      - .nuget
      - System.Private.CoreLib
      - ToolBox
      - binder
      - classlibnative
      - debug
      - dlls
      - gc
      - gcdump
      - gcinfo
      - hosts
      - ilasm
      - ildasm
      - inc

```
1 // Licensed to the .NET Foundation under one or more agreements.
2 // The .NET Foundation licenses this file to you under the MIT license.
3 // See the LICENSE file in the project root for more information.
4 //
5 // -----
6 // CLRConfig.h
7 //
8
9 //
10 // Unified method of accessing configuration values from environment variables, registry, etc.
11 // This class replaces all GetConfigDWORD and GetConfigString methods in EEConfig and R
12 // flag, add an entry in the table in file:CLRConfigValues.h.
13 //
14 //
```

Code Blame 215 lines (183 loc) · 8.83 KB

Raw Copy Download

```
17 // -----
18
19
20 #include "utilcode.h"
21 #include "holder.h"
22
23 #ifndef __CLRConfig_h__
24 #define __CLRConfig_h__
25
26 class CLRConfig
27 {
28 public:
29     //
30     // Types
31     //
32
33     // Setting each option results in some change to the config value lookup method. De
34     // the following order):
35     // * Look at environment variables (prepending COMPlus to the name)
36     // * Look at the framework registry keys (HKCU\Software\Microsoft\NETFramework the
37     //     HKLM\Software\Microsoft\NETFramework)
38     // * Look at the available config files (system, application, host and user). For d
39     //     Link to BOTR documentation
40     enum LookupOptions {
41         // If set, don't look in environment variables.
42         IgnoreEnv = 0x1,
43         // If set, do not prepend "COMPlus_" when doing environment variable lookup.
44         DontPrependCOMPlus_ = 0x2,
45         // If set, don't look in HKLM in the registry.
46         IgnoreHKLM = 0x4,
47         // If set, don't look in HKCU in the registry.
48         IgnoreHKCU = 0x8,
49         // If set, look only in the system config file, ignoring other config files.
50         // (This option does not affect environment variable and registry lookups)
51         ConfigFile_SystemOnly = 0x40,
52         // If set, reverse the order of config file lookups (application config file fi
53         // (This option does not affect environment variable and registry lookups)
54         ConfigFile_ApplicationFirst = 0x80,
55         // Remove any whitespace at beginning and end of value. (Only applicable for
56         // *string* configuration values.)
57         TrimWhiteSpaceFromStringValue = 0x100,
```

- >  clr
- >  clr\_std
- >  llvm
- >  mpl
- >  obj
- >  shim
- >  winrt
-  CMakeLists.txt
-  CrstTypeTool.cs
-  CrstTypes.def
-  MSCOREE.IDL
-  OpCodeGen.pl
-  allocacheck.h
-  appxutil.h
-  arrayholder.h
-  arraylist.h
-  bbsweep.h
-  bitmask.h
-  bitmask.inl
-  bitposition.h

```
57         minimumWhitespaceFromStringValue = 0x100,
58
59         // Legacy EEConfig-style lookup.
60         EEConfig_default = 0,
61     };
62
63     // Struct used to store information about where/how to find a Config DWORD.
64     // NOTE: Please do NOT create instances of this struct. Use the macros in file:CLRConfigValues.h
65     typedef struct ConfigDWORDInfo
66     {
67         LPCWSTR name;
68         DWORD defaultValue;
69         LookupOptions options;
70     } ConfigDWORDInfo;
71
72     // Struct used to store information about where/how to find a Config String.
73     // NOTE: Please do NOT create instances of this struct. Use the macros in file:CLRConfigValues.h
74     typedef struct ConfigStringInfo
75     {
76         LPCWSTR name;
77         LookupOptions options;
78     } ConfigStringInfo;
79
80     //
81     // Declaring structs using the macro table in CLRConfigValues.h
82     //
83
84     // These macros declare ConfigDWORDInfo structs.
85     #define RETAIL_CONFIG_DWORD_INFO(symbol, name, defaultValue, description) \
86         static const ConfigDWORDInfo symbol;
87     #define RETAIL_CONFIG_DWORD_INFO_EX(symbol, name, defaultValue, description, lookupOptions) \
88         static const ConfigDWORDInfo symbol;
89
90     // These macros declare ConfigStringInfo structs.
91     #define RETAIL_CONFIG_STRING_INFO(symbol, name, description) \
92         static const ConfigStringInfo symbol;
93     #define RETAIL_CONFIG_STRING_INFO_EX(symbol, name, description, lookupOptions) \
94         static const ConfigStringInfo symbol;
95
96     // TEMPORARY macros that declare strings. These are used for config value accesses that have been
97     // moved over to CLRConfig yet. Once all accesses have been moved, these macros (and their
98     // instantiations in file:../utilcode/CLRConfig.cpp) should be removed.
99     #define RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(symbol, name, description) \
100         static const LPCWSTR symbol;
101     #define RETAIL_CONFIG_STRING_INFO_DIRECT_ACCESS(symbol, name, description) \
102         static const LPCWSTR symbol;
103
104     //
105     // Debug versions of the macros
106     //
107     #ifdef _DEBUG
108     #define CONFIG_DWORD_INFO(symbol, name, defaultValue, description) \
109         static const ConfigDWORDInfo symbol;
110     #define CONFIG_DWORD_INFO_EX(symbol, name, defaultValue, description, lookupOptions) \
111         static const ConfigDWORDInfo symbol;
112     #define CONFIG_STRING_INFO(symbol, name, description) \
113         static const ConfigStringInfo symbol;
114     #define CONFIG_STRING_INFO_EX(symbol, name, description, lookupOptions) \
115         static const ConfigStringInfo symbol;
116     #define CONFIG_DWORD_INFO_DIRECT_ACCESS(symbol, name, description) \
117         static const LPCWSTR symbol;
118     #define CONFIG_STRING_INFO_DIRECT_ACCESS(symbol, name, description) \
```

```
142         #undef CONFIG_STRING_INFO_DIRECT_ACCESS
143
144         //
145         // Methods to do config value (DWORD and String) lookups.
146         //
147         static BOOL IsConfigEnabled(const ConfigDWORDInfo & info);
148
149         // Look up a DWORD config value.
150         static DWORD GetConfigValue(const ConfigDWORDInfo & info);
151
152         // Look up a DWORD config value.
153         static DWORD GetConfigValue(const ConfigDWORDInfo & info, bool acceptExplicitDefault);
154
155         // Look up a string config value.
156         // You own the string that's returned.
157         static LPWSTR GetConfigValue(const ConfigStringInfo & info);
158
159         // Look up a string config value, passing it out through a pointer reference. Report
160         // errors (HRESULT E_OUTOFMEMORY).
161         // You own the string that's returned.
162         static HRESULT GetConfigValue(const ConfigStringInfo & info, __deref_out_z LPWSTR * pwszTrimmed);
163
164         //
165         // Check whether an option is specified (e.g. explicitly listed) in any of the CLR
166         // locations: environment or registry (with or without COMPlus_) or any config file.
167         // The result is therefore a conservative approximation (some settings do not actually
168         // take effect everywhere and no setting is valid both with and without COMPlus_)
169         //
170         static BOOL IsConfigOptionSpecified(LPCWSTR name);
171
172         // Free a string returned by GetConfigValue
173         static void FreeConfigString(__in __in_z LPWSTR name);
174
175     private:
176
177         // Helper method to translate LookupOptions to REGUTIL::CORConfigLevel
178         static REGUTIL::CORConfigLevel GetConfigLevel(LookupOptions options);
179
180         //
181         // Helper methods.
182         //
183
184         // Helper method to check if a certain option is set in a ConfigDWORDInfo struct.
185         static inline BOOL CheckLookupOption(const ConfigDWORDInfo & info, LookupOptions option)
186         {
187             LIMITED_METHOD_CONTRACT;
188             return ((info.options & option) == option) ? TRUE : FALSE;
189         }
190
191         // Helper method to check if a certain option is set in a ConfigStringInfo struct.
192         static inline BOOL CheckLookupOption(const ConfigStringInfo & info, LookupOptions option)
193         {
194             LIMITED_METHOD_CONTRACT;
195             return ((info.options & option) == option) ? TRUE : FALSE;
196         }
197
198         // Helper method to check if a certain option is set in an options enum.
199         static inline BOOL CheckLookupOption(LookupOptions infoOptions, LookupOptions optionToCheck)
200         {
201             LIMITED_METHOD_CONTRACT;
202             return ((infoOptions & optionToCheck) == optionToCheck) ? TRUE : FALSE;
203         }
204
205         static HRESULT TrimWhiteSpace(LPCWSTR wszOrig, __deref_out_z LPWSTR * pwszTrimmed);
206     };
```

```
207  
208     inline CLRConfig::LookupOptions operator|(CLRConfig::LookupOptions lhs, CLRConfig::Look  
209     {  
210         return static_cast<CLRConfig::LookupOptions>(static_cast<DWORD>(lhs) | static_cast<  
211     }  
212  
213     typedef Wrapper<LPWSTR, DoNothing, CLRConfig::FreeConfigString, NULL> CLRConfigStringHo  
214  
215     #endif //__CLRConfig_h__
```