Product Solutions Resources Open Source Enterprise Pricing Sign in Sign up

dotnet / runtime Public

Notifications Fork 4.7k Star 15.2k

Code Issues 5k+ Pull requests 324 Discussions Actions Projects 6 Security 37 Insights

runtime / src / coreclr / src / inc / clrconfigvalues.h

jkoritzinsky Remove built-in WinRT support from the runtime (#36715) f62e934 · 4 years ago History

```
// Licensed to the .NET Foundation under one or more agreements.
// The .NET Foundation licenses this file to you under the MIT license.
// See the LICENSE file in the project root for more information.
//*****************************************************************************
// CLRConfigValues.h
//

//
// Unified method of accessing configuration values from environment variables,
// registry and config file.
//
//*****************************************************************************

// IMPORTANT: Before adding a new config value, please read up on naming conventions (s
// code:#NamingConventions)
//
```

Files

f62e934

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

runtime / src / coreclr / src / inc / clrconfigvalues.h Top

Code Blame 814 lines (746 loc) · 75.6 KB Raw

```
// * How to access config values (see code:#Access)
// * Naming conventions (see code:#NamingConventions)
//
//
// ==================================
// #Define - Use one of the following macros to define config values. (See code:#DWORDs
// ==================================
//
// By default, all macros are DEBUG ONLY. Add the "RETAIL_" prefix to make the config v
//
// #DWORDs:
// --------------------------------------------------------------------------
// CONFIG_DWORD_INFO(symbol, name, defaultValue, description)
// --------------------------------------------------------------------------
// Use this macro to define a basic DWORD value. CLRConfig will look in environment var
// COMPlus_ to the name), the registry (HKLM and HKCU), and all the config files for th
// where CLRConfig looks, use the extended version of the macro below. IMPORTANT: pleas
// code:#NamingConventions for the symbol and the name!
//
// Example: CONFIG_DWORD_INFO(INTERNAL_AllowCrossModuleInlining, W("AllowCrossModuleInl
//
// --------------------------------------------------------------------------
// CONFIG_DWORD_INFO_EX(symbol, name, defaultValue, description, lookupOptions)
// --------------------------------------------------------------------------
// To customize where CLRConfig looks to get a DWORD, use the extended (_EX) version of
// of options and their descriptions, see code:CLRConfig.LookupOptions
//
// Example: CONFIG_DWORD_INFO_EX(INTERNAL_EnableInternetHREFexes, W("EnableInternetHREF
// (CLRConfig::LookupOptions) (CLRConfig::IgnoreEnv | CLRConfig::IgnoreHKCU))
//
// #Strings:
// --------------------------------------------------------------------------
// CONFIG_STRING_INFO(symbol, name, description)
// --------------------------------------------------------------------------
// Defines a string value. Same rules apply as DWORDs.
//
//
```

- inc
  - 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

```
// ---------------------------------------------------------------------------
// CONFIG_STRING_INFO_EX(symbol, name, description, lookupOptions)
// ---------------------------------------------------------------------------
// Extended version of the String macro. Again, similar to the DWORD extended macro.
//
//
// ==============================================================
// #Access - Use the following overloaded method to access config values.
// ==============================================================
// From anywhere, use CLRConfig::GetConfigValue(CLRConfig::<symbol>) to access any valu
// file.
//
//
// ==============================================================
// #NamingConventions
// ==============================================================
// ----------------
// #Symbol - used to access values from the source. (using CLRConfig::<symbol>)
// ----------------
// The symbol for each config value is named as such:
// ### <Class>_<feature area>_<name> ###
//
// <Class> indicates which of the following buckets the value is in:
// * INTERNAL ? this value is for internal (CLR team) use only
// * UNSUPPORTED ? this value is available to partners/developers, but is not officiall
// * EXTERNAL ? this value is available for anyone to use and is publicly documented
//
// Examples:
// * INTERNAL_Security_FullAccessChecks
// * UNSUPPORTED_Security_DisableTransparency
// * EXTERNAL_Security_LegacyHMACMode
//
// ----------------
// #Name - the name of the registry value or environment variable that CLRConfig looks
// ----------------
// The name of each value is the same as the symbol, with one exception. Names of exter
// contain the EXTERNAL prefix.
//
// For compatibility reasons, current names do not follow the convention.
//
// Examples:
// * W("INTERNAL_Security_FullAccessChecks")
// * W("UNSUPPORTED_Security_DisableTransparency")
// * W("Security_LegacyHMACMode") <---------------------- (No EXTERNAL prefix)

///
/// AppDomain
///
CONFIG_DWORD_INFO(INTERNAL_ADDumpSB, W("ADDumpSB"), 0, "Not used")
CONFIG_DWORD_INFO(INTERNAL_ADForceSB, W("ADForceSB"), 0, "Forces sync block creation fo
CONFIG_DWORD_INFO(INTERNAL_ADLogMemory, W("ADLogMemory"), 0, "Superseded by test hooks"
CONFIG_DWORD_INFO(INTERNAL_ADTakeDHSnapShot, W("ADTakeDHSnapShot"), 0, "Superseded by t
CONFIG_DWORD_INFO(INTERNAL_ADTakeSnapShot, W("ADTakeSnapShot"), 0, "Superseded by test
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_EnableFullDebug, W("EnableFullDebug"), "Heavy-

///
/// Jit Pitching
///
RETAIL_CONFIG_DWORD_INFO(INTERNAL_JitPitchEnabled, W("JitPitchEnabled"), (DWORD)0, "Set
RETAIL_CONFIG_DWORD_INFO(INTERNAL_JitPitchMemThreshold, W("JitPitchMemThreshold"), (DWO
RETAIL_CONFIG_DWORD_INFO(INTERNAL_JitPitchMethodSizeThreshold, W("JitPitchMethodSizeThr
RETAIL_CONFIG_DWORD_INFO(INTERNAL_JitPitchTimeInterval, W("JitPitchTimeInterval"), (DWO
RETAIL_CONFIG_DWORD_INFO(INTERNAL_JitPitchPrintStat, W("JitPitchPrintStat"), (DWORD)0,
RETAIL_CONFIG_DWORD_INFO(INTERNAL_JitPitchMinVal, W("JitPitchMinVal"), (DWORD)0, "Do Ji
RETAIL_CONFIG_DWORD_INFO(INTERNAL_JitPitchMaxVal, W("JitPitchMaxVal"), (DWORD)0xffffffff

///
/// Assembly Loader
///
CONFIG_DWORD_INFO_EX(INTERNAL_GetAssemblyIfLoadedIgnoreRidMap, W("GetAssemblyIfLoadedIg

///
/// Conditional breakpoints
///
RETAIL_CONFIG_DWORD_INFO_EX(UNSUPPORTED_BreakOnBadExit, W("BreakOnBadExit"), 0, "", CLR
CONFIG_STRING_INFO(INTERNAL_BreakOnClassBuild, W("BreakOnClassBuild"), "Very useful for
CONFIG_STRING_INFO(INTERNAL_BreakOnClassLoad, W("BreakOnClassLoad"), "Very useful for d
CONFIG_STRING_INFO(INTERNAL_BreakOnComToClrNativeInfoInit, W("BreakOnComToClrNativeInfo
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnDebugBreak, W("BreakOnDebugBreak"), 0, "Allows an 
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnDILoad, W("BreakOnDILoad"), 0, "Allows an assert w
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnDumpToken, W("BreakOnDumpToken"), 0xffffffff, "Bre
RETAIL_CONFIG_DWORD_INFO_EX(UNSUPPORTED_BreakOnEELoad, W("BreakOnEELoad"), 0, "", CLRCo
CONFIG_DWORD_INFO(INTERNAL_BreakOnEEShutdown, W("BreakOnEEShutdown"), 0, "")
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnExceptionInGetThrowable, W("BreakOnExceptionInGetT
CONFIG_DWORD_INFO(INTERNAL_BreakOnFindMethod, W("BreakOnFindMethod"), 0, "Breaks in fin
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnFirstPass, W("BreakOnFirstPass"), 0, "", CLRConfig
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnHR, W("BreakOnHR"), 0, "Debug.cpp, IfFailxxx use t
CONFIG_STRING_INFO(INTERNAL_BreakOnInstantiation, W("BreakOnInstantiation"), "Very usef
CONFIG_STRING_INFO(INTERNAL_BreakOnInteropStubSetup, W("BreakOnInteropStubSetup"), "Thr
CONFIG_STRING_INFO_EX(INTERNAL_BreakOnInteropVTableBuild, W("BreakOnInteropVTableBuild"
CONFIG_STRING_INFO(INTERNAL_BreakOnMethodName, W("BreakOnMethodName"), "Very useful for
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnNotify, W("BreakOnNotify"), 0, "", CLRConfig::EECo
RETAIL_CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnRetailAssert, W("BreakOnRetailAssert"), 0,
CONFIG_DWORD_INFO_EX(INTERNAL_BreakOnSecondPass, W("BreakOnSecondPass"), 0, "", CLRConf
RETAIL_CONFIG_DWORD_INFO_EX(UNSUPPORTED_BreakOnSO, W("BreakOnSO"), 0, "", CLRConfig::EE
CONFIG_STRING_INFO(INTERNAL_BreakOnStructMarshalSetup, W("BreakOnStructMarshalSetup"),
```

```
// DO NOT ADD ANY MORE CONFIG SWITCHES TO THIS SECTION!
// **
CONFIG_DWORD_INFO_EX(INTERNAL_ActivatePatchSkip, W("ActivatePatchSkip"), 0, "Allows an
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_AlwaysUseMetadataInterfaceMapLayout, W("Always
CONFIG_DWORD_INFO(INTERNAL_AssertOnUnneededThis, W("AssertOnUnneededThis"), 0, "While t
CONFIG_DWORD_INFO_EX(INTERNAL_AssertStacktrace, W("AssertStacktrace"), 1, "", CLRConfig
CONFIG_DWORD_INFO_EX(INTERNAL_clearNativeImageStress, W("clearNativeImageStress"), 0, "
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_CPUFamily, W("CPUFamily"), "")
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_CPUFeatures, W("CPUFeatures"), "")
RETAIL_CONFIG_DWORD_INFO_EX(EXTERNAL_DisableConfigCache, W("DisableConfigCache"), 0, "U
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(EXTERNAL_DisableStackwalkCache, W("DisableStackw
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(UNSUPPORTED_DoubleArrayToLargeObjectHeap, W("Dou
CONFIG_STRING_INFO(INTERNAL_DumpOnClassLoad, W("DumpOnClassLoad"), "Dumps information a
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_ExpandAllOnLoad, W("ExpandAllOnLoad"), "")
CONFIG_STRING_INFO_DIRECT_ACCESS(INTERNAL_ForcedRuntime, W("ForcedRuntime"), "Verify ve
CONFIG_DWORD_INFO_EX(INTERNAL_ForceRelocs, W("ForceRelocs"), 0, "", CLRConfig::EEConfig
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_GenerateLongJumpDispatchStubRatio, W("Generate
CONFIG_DWORD_INFO_EX(INTERNAL_HashStack, W("HashStack"), 0, "", CLRConfig::EEConfig_def
CONFIG_DWORD_INFO(INTERNAL_HostManagerConfig, W("HostManagerConfig"), (DWORD)-1, "")
CONFIG_DWORD_INFO(INTERNAL_HostTestThreadAbort, W("HostTestThreadAbort"), 0, "")
RETAIL_CONFIG_DWORD_INFO_EX(UNSUPPORTED_IgnoreDllMainReturn, W("IgnoreDllMainReturn"),
CONFIG_STRING_INFO(INTERNAL_InvokeHalt, W("InvokeHalt"), "Throws an assert when the giv
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_MaxStackDepth, W("MaxStackDepth"), "")
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_MaxStubUnwindInfoSegmentSize, W("MaxStubUnwind
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_MaxThreadRecord, W("MaxThreadRecord"), "")
CONFIG_DWORD_INFO(INTERNAL_MessageDebugOut, W("MessageDebugOut"), 0, "")
RETAIL_CONFIG_DWORD_INFO_EX(EXTERNAL_NativeImageRequire, W("NativeImageRequire"), 0, ""
CONFIG_DWORD_INFO_EX(INTERNAL_NestedEhOom, W("NestedEhOom"), 0, "", CLRConfig::EEConfig
#define INTERNAL_NoGuiOnAssert_Default 1
RETAIL_CONFIG_DWORD_INFO_EX(INTERNAL_NoGuiOnAssert, W("NoGuiOnAssert"), INTERNAL_NoGuiO
RETAIL_CONFIG_DWORD_INFO_EX(EXTERNAL_NoProcedureSplitting, W("NoProcedureSplitting"), 0
CONFIG_DWORD_INFO_EX(INTERNAL_NoStringInterning, W("NoStringInterning"), 1, "Disallows
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(EXTERNAL_NotifyBadAppCfg, W("NotifyBadAppCfg"),
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_PauseOnLoad, W("PauseOnLoad"), "Stops in Syste
CONFIG_DWORD_INFO(INTERNAL_PerfAllocsSizeThreshold, W("PerfAllocsSizeThreshold"), 0x3FF
CONFIG_DWORD_INFO(INTERNAL_PerfNumAllocsThreshold, W("PerfNumAllocsThreshold"), 0x3FFFF
CONFIG_STRING_INFO(INTERNAL_PerfTypesToLog, W("PerfTypesToLog"), "Log facility LF_GCALL
RETAIL_CONFIG_DWORD_INFO(EXTERNAL_Prepopulate1, W("Prepopulate1"), 1, "")
CONFIG_STRING_INFO(INTERNAL_PrestubGC, W("PrestubGC"), "")
CONFIG_STRING_INFO(INTERNAL_PrestubHalt, W("PrestubHalt"), "")
RETAIL_CONFIG_STRING_INFO(EXTERNAL_RestrictedGCStressExe, W("RestrictedGCStressExe"), "
CONFIG_DWORD_INFO_EX(INTERNAL_ReturnSourceTypeForTesting, W("ReturnSourceTypeForTesting
RETAIL_CONFIG_DWORD_INFO_EX(UNSUPPORTED_RSStressLog, W("RSStressLog"), 0, "Allows turni
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_SaveThreadInfo, W("SaveThreadInfo"), "")
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_SaveThreadInfoMask, W("SaveThreadInfoMask"), "
CONFIG_DWORD_INFO(INTERNAL_SBDumpOnNewIndex, W("SBDumpOnNewIndex"), 0, "Used for Syncbl
CONFIG_DWORD_INFO(INTERNAL_SBDumpOnResize, W("SBDumpOnResize"), 0, "Used for Syncblock
CONFIG_DWORD_INFO(INTERNAL_SBDumpStyle, W("SBDumpStyle"), 0, "Used for Syncblock debugg
RETAIL_CONFIG_STRING_INFO_DIRECT_ACCESS(UNSUPPORTED_ShimDatabaseVersion, W("ShimDatabas
RETAIL_CONFIG_DWORD_INFO(UNSUPPORTED_SleepOnExit, W("SleepOnExit"), 0, "Used for lrak d
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_StubLinkerUnwindInfoVerificationOn, W("StubLin
RETAIL_CONFIG_DWORD_INFO_EX(UNSUPPORTED_SuccessExit, W("SuccessExit"), 0, "", CLRConfig
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(EXTERNAL_SymbolReadingPolicy, W("SymbolReadingPo
RETAIL_CONFIG_DWORD_INFO(UNSUPPORTED_TestDataConsistency, W("TestDataConsistency"), FAL
RETAIL_CONFIG_DWORD_INFO_EX(EXTERNAL_ThreadGuardPages, W("ThreadGuardPages"), 0, "", CL
RETAIL_CONFIG_DWORD_INFO_EX(EXTERNAL_Timeline, W("Timeline"), 0, "", CLRConfig::EEConfi
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(UNSUPPORTED_TotalStressLogSize, W("TotalStressLo

#ifdef _DEBUG
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(EXTERNAL_TraceIUnknown, W("TraceIUnknown"), "")
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(EXTERNAL_TraceWrap, W("TraceWrap"), "")
#endif
```

```
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(EXTERNAL_TURNOFFDEBUGINFO, W("TURNOFFDEBUGINFO")
RETAIL_CONFIG_DWORD_INFO(EXTERNAL_UseMethodDataCache, W("UseMethodDataCache"), FALSE, "
RETAIL_CONFIG_DWORD_INFO(EXTERNAL_UseParentMethodData, W("UseParentMethodData"), TRUE,
CONFIG_DWORD_INFO_DIRECT_ACCESS(INTERNAL_VerifierOff, W("VerifierOff"), "")
RETAIL_CONFIG_DWORD_INFO_DIRECT_ACCESS(EXTERNAL_VerifyAllOnLoad, W("VerifyAllOnLoad"),
// **
// PLEASE MOVE ANY CONFIG SWITCH YOU OWN OUT OF THIS SECTION INTO A CATEGORY ABOVE
//
// DO NOT ADD ANY MORE CONFIG SWITCHES TO THIS SECTION!
// **
//---------------------------------------------------------------------------------
```