Gizmondo Digital Signature Specifications

* Strictly Confidential *

Revision v1.0

Revision	Description	Author
1.0	Initial draft.	PH

Gizmondo Digital Signatures
(Last Updated 21-01-2005)

Preamble

The Gizmondo device operates in the Windows CE trusted security mode. All modules (both EXE and DLL) must be digitally signed in order to run on the device.

NOTE: Unsigned modules will fail to be loaded by the Kernel.

Four signing levels are implemented:

Developer run mode
Developer trusted mode*****
User run mode
User trusted mode*****

In **run** mode, the application is trusted to run, but is prevented from making any privileged function calls. In **trusted** mode the application is fully trusted to perform any operation.

Distribution of developer keys is managed by Gizmondo.

User Run Modes
By default all retail Gizmondo units will only be able to run applications signed with either of the user mode keys. Gizmondo is the sole signing authority for user mode keys and that signing will form part of the final GCR game approvals process, prior to manufacture.

Developer Run Modes
Developers Gizmondo units will also have the ability to run applications signed with the developer's keys. These keys will be distributed to developers by Gizmondo and signing will need to be incorporated into the developers build processes.

***Trusted Developer and User Modes**
Trusted keys are **not** distributed with the Gizmondo Developer Kits by default, as the trusted keys allow much lower (systems) level access to the device by the developer's application.

Generally, most gaming applications won't have a need to call any of the trusted API functions, however please read through the "Trusted API" section of this document and if you feel your product will require access to the privileged functions please fill in the

attached “Trusted Key Request Form” in order to request the required trusted certificates.

Further detailed documentation on configuring the digital certificates and implementing the signing process automatically from your build process is distributed with the developer key manifest.

Trusted APIs

In addition to the OEM functions, the CeGetCurrentTrust and CeGetCallerTrust APIs enable a DLL to query the trust level of a calling application. You can use these functions to verify the trust levels of the applications.

The following (hyperlinked) table shows the application programming interfaces (APIs) that can be called **only** by trusted applications:

AllocPhysMem	ReadRegistryFromOEM
CeSetThreadPriority	RegCopyFile
CeSetThreadQuantum	RegReplaceKey
CheckPassword	RegRestoreFile
ContinueDebugEvent	RegSaveKey
CryptUnprotectData	SetCleanRebootFlag
DebugActiveProcess	SetCurrentUser
ForcePageout	SetInterruptEvent
FreeIntChainHandler	SetKMode
FreePhysMem	SetPassword
InterruptDisable	SetPasswordStatus
InterruptDone	SetProcPermissions
InterruptInitialize	SetSystemMemoryDivision
KernelLibIoControl	SetUserData
LoadDriver	UnlockPages
LoadIntChainHandler	VirtualCopy
LoadKernelLibrary	VirtualSetPageFlags
LockPages	WaitForDebugEvent
PowerOffSystem	WriteProcessMemory
ReadProcessMemory	WriteRegistryToOEM

The following (hyperlinked) table shows file-based APIs that are influenced by the SYSTEM attribute that can be set on a file.

CreateFile	CreateFileForMapping
CopyFile	DeleteFile
DeleteAndRenameFile	MoveFile
RemoveDirectory	SetFileAttributes

For more information, see File System Security.

The following (hyperlinked) table shows database APIs that are influenced by the SYSTEM attribute that can be set on a database.

CeMountDBVol	CeCreateDatabaseEx2
CeOpenDatabaseEx2	CeDeleteDatabaseEx
CeSetDatabaseInfoEx2	

For more information, see Database Security.

In addition, the debug flags DEBUG_ONLY_THIS_PROCESS and DEBUG_PROCESS of the CreateProcess API are restricted. If these flags are used by a non-trusted application, the identified process will still launch but no debugging will occur.

Debug flags, DEBUG_ONLY_THIS_PROCESS and DEBUG_PROCESS, in the **CreateProcess** API are restricted as well.

The secure registry architecture in Windows CE allows only trusted applications that you have identified to modify keys and values in protected portions of the registry.

Because most of the registry is unprotected, OEMs must place all important registry information in one of the protected keys.

Note All applications have read-only access to all registry keys and values.

In Windows CE .NET, the following registry root keys and their subkeys are protected from untrusted applications:

- **HKEY_LOCAL_MACHINE\Comm**
- **HKEY_LOCAL_MACHINE\Drivers**
- **HKEY_LOCAL_MACHINE\HARDWARE**
- **HKEY_LOCAL_MACHINE\Init**
- **HKEY_LOCAL_MACHINE\Services**
- **HKEY_LOCAL_MACHINE\SYSTEM**
- **HKEY_LOCAL_MACHINE\WDMDrivers**

Untrusted applications are also not allowed to modify protected data. They receive the ERROR_ACCESS_DENIED return value if they attempt to use the following registry functions:

- **RegSetValueEx**
- **RegCreateKeyEx**
- **RegDeleteKey**
- **RegDeleteValue**

For more information on digital signatures and the trusted APIs check our Microsoft's MSDN database on the Internet;

http://msdn.microsoft.com/library/default.asp?url=/library/enus/wcedsn40/html/cgcontrustedapis.asp