

SSystemInitParams Structure

C++

```
struct SSystemInitParams {
    void * hInstance;
    void * hWnd;
    ILog * pLog;
    ILogCallback * pLogCallback;
    ISystemUserCallback * pUserCallback;
    const char * sLogFileName;
    IValidator * pValidator;
    IOutputPrintSink * pPrintSync;
    char szSystemCmdLine[2048];
    char szUserPath[256];
    char szBinariesDir[256];
    bool bEditor;
    bool bPreview;
    bool bTestMode;
    bool bDedicatedServer;
    bool bExecuteCommandLine;
    bool bUIFramework;
    bool bSkipFont;
    bool bSkipRenderer;
    bool bSkipConsole;
    bool bSkipNetwork;
    bool bSkipLiveCreate;
    bool bMinimal;
    bool bSkipInput;
    bool bTesting;
    bool bNoRandom;
    bool bShaderCacheGen;
    bool bUnattendedMode;
    ISystem * pSystem;
    IGameStartup * pGameStartup;
    void * pCheckFunc;
    ProtectedFunction pProtectedFunctions[eProtectedFuncsLast];
    SCvarsDefault * pCvarsDefault;
};
```

File

ISystem.h

Description

Structure passed to Init method of [ISystem](#) interface.

SSystemInitParams::bDedicatedServer Data Member

C++

```
bool bDedicatedServer;
```

Description

When running a dedicated server.

SSystemInitParams::bEditor Data Member

C++

```
bool bEditor;
```

Description

When running in Editor mode.

SSystemInitParams::bExecuteCommandLine Data Member

C++

```
bool bExecuteCommandLine;
```

Description

can be switched of to suppress the feature or do it later during the initialization.

SSystemInitParams::bMinimal Data Member

C++

```
bool bMinimal;
```

Description

Don't load banks

SSystemInitParams::bNoRandom Data Member

C++

```
bool bNoRandom;
```

Description

use fixed generator init/seed

SSystemInitParams::bPreview Data Member

C++

```
bool bPreview;
```

Description

When running in Preview mode (Minimal initialization).

SSystemInitParams::bShaderCacheGen Data Member

C++

```
bool bShaderCacheGen;
```

Description

When running in shadercache gen mode

SSystemInitParams::bSkipConsole Data Member

C++

```
bool bSkipConsole;
```

Description

Don't create console

SSystemInitParams::bSkipFont Data Member

C++

```
bool bSkipFont;
```

Description

Don't load CryFont.dll

SSystemInitParams::bSkipInput Data Member

C++

```
bool bSkipInput;
```

Description

do not load CryInput

SSystemInitParams::bSkipLiveCreate Data Member

C++

```
bool bSkipLiveCreate;
```

Description

Don't create LiveCreate

SSystemInitParams::bSkipNetwork Data Member

C++

```
bool bSkipNetwork;
```

Description

Don't create Network

SSystemInitParams::bSkipRenderer Data Member

C++

```
bool bSkipRenderer;
```

Description

Don't load Renderer

SSystemInitParams::bTesting Data Member

C++

```
bool bTesting;
```

Description

CryUnit

SSystemInitParams::bTestMode Data Member

C++

```
bool bTestMode;
```

Description

When running in Automated testing mode.

SSystemInitParams::bUIFramework Data Member

C++

```
bool bUIFramework;
```

SSystemInitParams::bUnattendedMode Data Member

C++

```
bool bUnattendedMode;
```

Description

When running as part of a build on build-machines: Prevent popping up of any dialog

SSystemInitParams::hInstance Data Member

C++

```
void * hInstance;
```

SSystemInitParams::hWnd Data Member

C++

```
void * hWnd;
```

SSystemInitParams::pCheckFunc Data Member

C++

```
void * pCheckFunc;
```

Description

Authentication function (must be set).

SSystemInitParams::pCvarsDefault Data Member

C++

```
SCvarsDefault * pCvarsDefault;
```

Description

to override the default value of some cvar

SSystemInitParams::pGameStartup Data Member

C++

```
IGameStartup * pGameStartup;
```

Description

Pointer to the calling GameStartup instance, to allow use of some game specific data during engine init. char szLocalIP[256]; // local IP address (needed if we have several servers on one machine)

SSystemInitParams::pLog Data Member

C++

```
ILog * pLog;
```

Description

You can specify your own [ILog](#) to be used by System.

SSystemInitParams::pLogCallback Data Member

C++

```
ILogCallback * pLogCallback;
```

Description

You can specify your own [ILogCallback](#) to be added on log creation (used by Editor).

SSystemInitParams::pPrintSync Data Member

C++

```
IOutputPrintSink * pPrintSync;
```

Description

Print Sync which can be used to catch all output from engine

SSystemInitParams::pProtectedFunctions Data Member

C++

```
ProtectedFunction pProtectedFunctions[eProtectedFuncsLast];
```

Description

Protected functions.

SSystemInitParams::pSystem Data Member

C++

```
ISystem * pSystem;
```

Description

Pointer to existing [ISystem](#) interface, it will be reused if not NULL.

SSystemInitParams::pUserCallback Data Member

C++

```
ISystemUserCallback * pUserCallback;
```

SSystemInitParams::pValidator Data Member

C++

```
IValidator * pValidator;
```

Description

You can specify different validator object to use by System.

SSystemInitParams::sLogFileName Data Member

C++

```
const char* sLogFileName;
```

Description

File name to use for log.

SSystemInitParams::szBinariesDir Data Member

C++

```
char szBinariesDir[256];
```

SSystemInitParams::szSystemCmdLine Data Member

C++

```
char szSystemCmdLine[2048];
```

Description

Command line.

SSystemInitParams::szUserPath Data Member

C++

```
char szUserPath[256];
```

Description

User alias path relative to My Documents folder.

SSystemInitParams::SSystemInitParams Constructor

Initialization defaults.

C++

```
SSystemInitParams();
```

SSystemInitParams::ProtectedFunction Nested Type

C++

```
typedef void* (* ProtectedFunction)(void *param1, void *param2);
```

In This Topic