

## ICVar Structure

### C++

```
struct ICVar {  
    enum EConsoleLogMode {  
        eCLM_Off,  
        eCLM_ConsoleAndFile,  
        eCLM_FileOnly,  
        eCLM_FullInfo  
    };  
};
```

### File

IConsole.h

### Description

This interface is the 1:1 "C++ representation" of a console variable.

### Notes

A console variable is accessible in C++ through this interface and in all scripts as global variable (with the same name of the variable in the console)

## ICVar::EConsoleLogMode Enumeration

### C++

```
enum EConsoleLogMode {  
    eCLM_Off,  
    eCLM_ConsoleAndFile,  
    eCLM_FileOnly,  
    eCLM_FullInfo  
};
```

### File

IConsole.h

### Members

Members	Description
eCLM_Off	off
eCLM_ConsoleAndFile	normal info to console and file
eCLM_FileOnly	normal info to file only
eCLM_FullInfo	full info to file only

## ICVar::~~ICVar Destructor

C++

```
virtual ~ICVar();
```

## ICVar::AddOnChangeFunctor Method

C++

```
virtual void AddOnChangeFunctor(const SFunctor& pChangeFunctor) = 0;
```

### Description

Adds a new on change functor to the list. It will add from index 1 on (0 is reserved).

## ICVar::ClearFlags Method

C++

```
virtual void ClearFlags(int flags) = 0;
```

### Description

clear the specified bits in the flag field

## ICVar::DebugLog Method

C++

```
virtual void DebugLog(const int iExpectedValue, const EConsoleLogMode mode) const;
```

### Description

only useful for CVarGroups log difference between expected state and real state

## ICVar::ForceSet Method

C++

```
virtual void ForceSet(const char* s) = 0;
```

### Description

Force to set the string value of the variable - can be called from inside code only @param s string representation the value

## ICVar::GetDataProbeString Method

C++

```
virtual const char * GetDataProbeString() const = 0;
```

## Description

Return the data probe string value of the variable, don't store pointer as multiple calls to this function might return same memory ptr  
@return the value

---

## ICVar::GetFlags Method

C++

```
virtual int GetFlags() const = 0;
```

---

## Description

return the variable's flags @return the variable's flags

## See Also

[EVarFlags](#)

---

## ICVar::GetFVal Method

C++

```
virtual float GetFVal() const = 0;
```

---

## Description

Return the float value of the variable @return the value

---

## ICVar::GetHelp Method

C++

```
virtual const char* GetHelp() = 0;
```

---

## Description

return the variable's help text @return the variable's help text, can be 0 if no help is available

---

## ICVar::GetI64Val Method

C++

```
virtual int64 GetI64Val() const = 0;
```

---

## Description

Return the 64-bit integer value of the variable @return the value

---

## ICVar::GetIVal Method

C++

---

```
virtual int GetIVal() const = 0;
```

## Description

Return the integer value of the variable @return the value

## ICVar::GetMemoryUsage Method

C++

```
virtual void GetMemoryUsage(class ICrySizer* pSizer) const = 0;
```

## ICVar::GetName Method

C++

```
virtual const char* GetName() const = 0;
```

## Description

return the variable's name @return the variable's name

## ICVar::GetNumberOfOnChangeFunctors Method

C++

```
virtual uint64 GetNumberOfOnChangeFunctors() const = 0;
```

## Description

Returns the number of registered on change functors.

## ICVar::GetOnChangeCallback Method

C++

```
virtual ConsoleVarFunc GetOnChangeCallback() const = 0;
```

## Description

Get the current callback function.

## ICVar::GetOnChangeFunctor Method

C++

```
virtual const SFuncutor& GetOnChangeFunctor(uint64 nFuncutorIndex) const = 0;
```

## Description

Returns the number of registered on change functors.

---

## ICVar::GetRealIVal Method

**C++**

```
virtual int GetRealIVal() const = 0;
```

### Returns

value that would represent the state, -1 if the state cannot be found

### Description

only useful for CVarGroups, other types return [GetIVal\(\)](#) CVarGroups set multiple other CVars and this function returns the integer value the CVarGroup should have, when looking at the controlled cvars

---

## ICVar::GetString Method

**C++**

```
virtual const char * GetString() const = 0;
```

### Description

Return the string value of the variable, don't store pointer as multiple calls to this function might return same memory ptr @return the value

---

## ICVar::GetType Method

**C++**

```
virtual int GetType() = 0;
```

### Description

return the primary variable's type @return the primary variable's type e.g. [CVAR\\_INT](#), [CVAR\\_FLOAT](#), [CVAR\\_STRING](#)

---

## ICVar::IsConstCVar Method

**C++**

```
virtual bool IsConstCVar() const = 0;
```

### Description

return if the variable may be modified in config files @return if the variable may be modified in config files

---

## ICVar::Release Method

**C++**

```
virtual void Release() = 0;
```

---

## Description

delete the variable

## Notes

the variable will automatically unregister itself from the console

---

## ICVar::RemoveOnChangeFunctor Method

C++

```
virtual bool RemoveOnChangeFunctor(const uint64 nElement) = 0;
```

---

## Description

Removes an on change functor returns true if removal was successful.

---

## ICVar::Set Method (char\*)

C++

```
virtual void Set(const char* s) = 0;
```

---

## Description

set the string value of the variable @param s string representation the value

---

## ICVar::Set Method (float)

C++

```
virtual void Set(const float f) = 0;
```

---

## Description

set the float value of the variable @param s float representation the value

---

## ICVar::Set Method (int)

C++

```
virtual void Set(const int i) = 0;
```

---

## Description

set the float value of the variable @param s integer representation the value

---

## ICVar::SetFlags Method

---

**C++**

```
virtual int SetFlags(int flags) = 0;
```

## Description

[Set](#) the variable's flags

## See Also

[EVarFlags](#)

---

## ICVar::SetOnChangeCallback Method

**C++**

```
virtual void SetOnChangeCallback(ConsoleVarFunc pChangeFunc) = 0;
```

## Description

[Set](#) a new on change function callback. Deprecated function. The functor should be preferred.

## In This Topic