{steamAPI Registercallresult}

The current user ID is the numerical unique identifier for a user currently in the Steam client. Steam IDs are not linked to real-world identity. See User . CSteamID GetSteamID();. Get the Steam ID of the current user. This is commonly called the 'current user', or 'local user' . {Steamworks CSteamID} The id number for a user who is currently logged into the Steam client. {Steamworks CSteamID} {Steamworks CSteamID} A randomly generated 64-bit number for use as a multiplayer matchmakingID in the Gamespy game network. {Steamworks CSteamID} Example: CSteamID m_steamIDUser; CSteamID m_steamIDUser; Example: m_steamIDUser = CSteamID(765611979604354233); m_steamIDUser = '765611979604354233'; m_steamIDUser = new CSteamID(); m_steamIDUser->Set("765611979604354233"); m_steamIDUser->nAppID = 123; m_steamIDUser->nProtocolAppID = 987; m_steamIDUser->nAccountID = 123; m_steamIDUser->m_steamID64 = 123; m_steamIDUser->m_bAnonUser = false; m_steamIDUser->m_bSelfUser = false; m_steamIDUser->m_eRetrievedLocally = false; m_steamIDUser = '765611979604354233'; The single most important part of a Steamworks API call is the SteamID. The user's (and hence game) device SteamID (64-bit numbers) is incredibly important to Steam. A user can not play a game without the correct SteamID . { int nAppID = 987; CSteamID steamID = CSteamID(765611979604354233); // Set the new AppID steamID.nAppID = nAppID; // Get the CallResult of the method if(steamID.IsValid()) { // Put the new AppID in the steamID

void SteamAPI_AddCallResult(string result);. Adds a call result from an API call you made. The steam_api. dll file does not support callbacks, so we must use Callback results. Unlike functions in steam_api, which support call results, this function call errors and will cause an exception to be thrown. The Steam API API is mainly to provide you game more functionality in details to enjoy the Steam service. API Version 1. Regarding to the current Steamworks API release version.Q: Compare two structs in C What is the best way to compare two structs to see if they are equal? I have two structures. Both have members with the same type (int) and each are populated with values from a for loop with a certain number of iterations. I would like to see if the two structs are equal to each other by printing out the two of the ints in both structures. Here is my simple code. #include #include #include #define SIZE 5 #define EOF 0 typedef int valtype; struct valstructure{ valtype i; struct valstructure *next; }; void fill_valstructure(struct valstructure * s){ valtype i=0; for(i=0;ii=i; s->next=NULL; printf("%d ",i); } } int main(void) { struct valstructure * s1; struct valstructure * s2; s1=malloc(sizeof(struct valstructure)*SIZE); s2=malloc(sizeof(struct valstructure)*SIZE); fill_valstructure(s1); fill_valstructure(s2); printf("%d ",s1[0].i); printf("%d ",s2[0].i); if 55cdc1ed1c