

ScarMultiSession

Plug-and-Play Multiplayer Session Plugin

Complete Setup and Integration Guide

Version 2.1 | Unreal Engine 5.2+

Studio	Black Scar Studio
Product	ScarMultiSession v2.1
Engine Support	Unreal Engine 5.2 (and newer)
Backend Support	Steam, EOS, LAN (NULL), and any OSS
Configuration	One central Project Settings page
Document	Setup and Integration Guide

WHAT IS NEW IN 2.1 A single Project Settings page now drives the whole flow, the player count is one number, and a built-in Setup Doctor validates your project on Play. See Sections 6 and 10.

Table of Contents

1	Introduction and Overview What the plugin is and what it provides
2	Who Is This Plugin For Target audience and use cases
3	Supported Backends Steam, EOS, LAN, and any OSS provider
4	Installation Installing and enabling the plugin
5	DefaultEngine.ini Backend Configuration Per-backend INI setup (Steam / EOS / NULL)
6	Central Configuration (Project Settings) The ONE place you configure everything
7	Widget Setup Creating the lobby and quick-join widgets
8	Game Mode and Map Setup Lobby / game GameModes and the three maps
9	Blueprint Integration The single node that drives the whole flow
10	Setup Doctor and Debugging Self-validation, logging, and console commands
11	Plugin Classes Reference Every class and its public API
12	Common Issues and Fixes Troubleshooting setup and runtime problems
13	Quick Reference Card One-page summary of everything

1 Introduction and Overview

ScarMultiSession is a plug-and-play multiplayer session management plugin for Unreal Engine 5, by Black Scar Studio. It handles the full flow - hosting, smart matchmaking, a replicated lobby with a live player count and countdown, and travel to your playable map - so you focus on your game, not Online Subsystem plumbing.

In version 2.1 the plugin is configured from a single Project Settings page. You set your three maps, the match size, the match type and your lobby widget once, and the menu and lobby read everything from there. A built-in Setup Doctor checks your project the moment you press Play and tells you exactly what to fix.

What the Plugin Provides

UScarMultiSessionSettings NEW. A Project Settings page (Plugins > Scar MultiSession) - the single place to configure maps, match size, match type, lobby widget, and an optional loading screen.

UMultiplayerSessionsSubsystem Auto-initialising GameInstance subsystem exposing Create, Find, Join, Destroy and Start session operations with dynamic multicast delegates.

UQuickJoinMenu Single-button smart matchmaking widget. Finds an open session, or hosts one if none exists. This is the entry menu.

ULobbyMenu Lobby waiting-room widget: live player count, optional host-only Force Start, and a synced countdown. Auto-destroyed when the match starts.

AScarLobbyGameMode Lobby GameMode. Spawns the lobby UI on each player's own client, replicates the count to everyone, and auto-travels to the game map when the lobby is full.

AScarLobbyPlayerController NEW. Spawns the lobby widget on the owning client (fixes the "Only Local Player Controllers can be assigned to widgets" error) and relays Force Start.

AScarLobbyGameState NEW. Replicates the live player count, required count and countdown to every client.

AScarMultiSessionGameMode Base GameMode for your game level. Subclass it for team assignment, pawn customisation, etc.

FScarSetupDoctor NEW. Development-time validation that reports misconfigurations with clear fix-it guidance.

2 Who Is This Plugin For

ScarMultiSession is for Unreal developers who want a working session and lobby layer without spending days fighting the Online Subsystem API.

- **Indie developers** who need a working host / join / lobby system in under an hour.
- **Prototypers and game-jam teams** who need multiplayer running fast, with near-zero configuration.
- **Intermediate Unreal developers** comfortable with Blueprints but new to the OSS API.
- **Commercial studios** wanting a backend-agnostic session layer (Steam, EOS, LAN) with no vendor lock-in.
- **Educators** - the source is fully commented and ASCII-clean for readability.
- **Plugin / system authors** who need session awareness without rebuilding the session layer.

3 Supported Backends

The plugin is built on Unreal's abstract Online Subsystem interface, so it works with any OSS provider with no code changes. Switching backends is a DefaultEngine.ini edit (Section 5).

Steam - OnlineSubsystemSteam

The common choice for PC. Real lobby matchmaking, friend sessions, presence and invites. Needs the OnlineSubsystemSteam plugin enabled and a Steam App ID. App ID 480 (Spacewar) works for testing.

Epic Online Services (EOS) - OnlineSubsystemEOS

Free cross-platform services - matchmaking, lobbies, voice and achievements. Needs the OnlineSubsystemEOS plugin and an Epic Developer Portal product. The plugin enables the lobby path EOS requires.

LAN and NULL - OnlineSubsystemNull

Zero-configuration local play. No accounts, SDKs or internet. LAN mode is detected and enabled automatically when the NULL subsystem is active.

Any custom OSS - IOnlineSession

Because the plugin only talks to the IOnlineSession interface, any compliant custom or third-party OSS works automatically.

NOTE You can have several OSS plugins installed. Only DefaultPlatformService in DefaultEngine.ini decides which one is active at runtime - the rest of the plugin is identical.

4 Installation

Install and enable

- Purchase ScarMultiSession on Fab / the Epic Games Marketplace; it appears in your library.
- In the Epic Games Launcher, install it to your engine version (5.2 or newer).
- Open your project, go to **Edit > Plugins**, search **ScarMultiSession**, tick **Enabled**, and restart.
- ScarMultiSession declares OnlineSubsystem and OnlineSubsystemUtils as dependencies, so they are enabled for you.

Manual (zip) install

- Extract the zip; it contains a folder named ScarMultiSession_UE5_X.
- Rename it to **ScarMultiSession** (drop the version suffix).
- Copy it to [YourProject]/Plugins/ScarMultiSession/.
- Reopen the project; if prompted to rebuild, click Yes. Enable it via Edit > Plugins.

NOTE If your project is Blueprint-only, there is nothing else to link - the plugin's modules load automatically. Continue to Section 5.

5 DefaultEngine.ini Backend Configuration

Pick the backend you are shipping. The file is at [YourProject]/Config/DefaultEngine.ini.

IMPORTANT Keep every .ini entry on ONE line. Never split +NetDriverDefinitions=(...) or +Artifacts=(...) across lines, or Unreal reports "Missing closing parenthesis", the GameNetDriver never builds, and Play as Client fails.

Steam

```
[OnlineSubsystem]
DefaultPlatformService=Steam

[OnlineSubsystemSteam]
bEnabled=true
SteamDevAppId=480           ; replace 480 with your real App ID to ship

[/Script/Engine.Engine]
!NetDriverDefinitions=ClearArray
+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="OnlineSubsystemSteam.SteamNetDriver"
,DriverClassNameFallback="OnlineSubsystemUtils.IpNetDriver")
```

Epic Online Services (EOS)

```
[OnlineSubsystem]
DefaultPlatformService=EOS

[OnlineSubsystemEOS]
bEnabled=true

[/Script/OnlineSubsystemEOS.EOSSettings]
CacheDir=EOSCache
DefaultArtifactName=MyProduct
+Artifacts=(ArtifactName="MyProduct",ClientId="...",ClientSecret="...",ProductId="...",SandboxId="...
.",DeploymentId="...",EncryptionKey="")
```

SECURITY Never commit your EOS Client Secret to source control. Use environment variables or a secrets manager in your build pipeline.

LAN / Local Testing (NULL)

```
[OnlineSubsystem]
DefaultPlatformService=NULL

[OnlineSubsystemNull]
bEnabled=true

[/Script/OnlineSubsystemUtils.IpNetDriver]
NetServerMaxTickRate=60
```

TIP For LAN / NULL you do not need any NetDriverDefinitions. To test on one PC, package the game (or use Standalone with "Run Under One Process" unchecked) and launch two copies; single-process PIE will not discover NULL sessions.

6 Central Configuration (Project Settings)

THIS IS THE ONE PLACE YOU CONFIGURE THE PLUGIN Open **Edit > Project Settings > Plugins > Scar MultiSession**. The menu and lobby read everything from here, so you never set the same value twice.

Setting	What it does
Main Menu Map	Map the Leave button returns to. Pick your menu level.
Lobby Map	Waiting-room map hosted as a listen server after Quick Join.
Game Map	Playable map travelled to once the lobby is full.
Required Players	The match size - ONE number. Sizes the session AND starts the match when this many gather.
Countdown Duration	Seconds before travelling once the lobby is full. 0 = instant.
Lobby Widget Class	Your WBP_LobbyMenu (parented to ULobbyMenu).
Match Type	Tag that keeps different modes from matching together (e.g. FreeForAll).
Loading Screen Widget Class	Optional. Shown during travel so players never see a frozen frame.

ONE NUMBER FOR PLAYER COUNT Required Players is the only count you set. The session is sized to hold exactly that many, and the match auto-starts the moment they are gathered. Set 2 for a two-player match, 4 for four.

TIP The three maps are picked as asset references, so selecting them here also makes them cook into packaged builds automatically - no more "missing map" travel failures.

7 Widget Setup

Create Widget Blueprint subclasses of the plugin's widget classes and add the required named widgets. **Names are case-sensitive.**

WBP_LobbyMenu (parent: ULobbyMenu)

- Create a Widget Blueprint with parent class **ULobbyMenu**; name it **WBP_LobbyMenu**.
- Required: **StatusText** (TextBlock), **LeaveButton** (Button).
- Optional: **ForceStartButton** (Button, auto-hidden on clients), **CountdownText** (TextBlock).
- No properties to set - the menu map for Leave comes from Project Settings.
- Assign this widget to **Lobby Widget Class** in Project Settings.

WBP_QuickJoinMenu (parent: UQuickJoinMenu)

- Create a Widget Blueprint with parent class **UQuickJoinMenu**; name it **WBP_QuickJoinMenu**.
- Required: **QuickJoinButton** (Button). Optional: **StatusText** (TextBlock).

WBP_LoadingScreen (optional, any UUserWidget)

- Create any Widget Blueprint (a full-screen image / "Loading..." text is enough).
- Assign it to **Loading Screen Widget Class** in Project Settings to enable it. Leave blank to disable.
- Editor PIE skips loading screens by design; it is visible in Standalone and packaged builds.

8 Game Mode and Map Setup

Lobby GameMode (BP_LobbyGameMode)

- Content Browser > Blueprint Class, search **ScarLobbyGameMode**, name it **BP_LobbyGameMode**.
- There is nothing to configure on it - all lobby settings live in Project Settings (Section 6).
- Open your Lobby map > World Settings > set **GameMode Override** to BP_LobbyGameMode.

NOTE Do not override Player Controller Class or Game State Class on BP_LobbyGameMode. The plugin sets AScarLobbyPlayerController and AScarLobbyGameState automatically; overriding them breaks the client-side lobby UI.

Game-Level GameMode (BP_GameGameMode)

- Blueprint Class > **ScarMultiSessionGameMode**, name it **BP_GameGameMode** (optional but recommended).
- Set its Default Pawn Class to your character; override PostLogin / Logout for game logic.
- Open your Game map > World Settings > set GameMode Override to BP_GameGameMode.

The three maps

Map	GameMode	Notes
Main Menu	Any (or None)	Hosts your WBP_QuickJoinMenu. Set as Main Menu Map in settings.
Lobby	BP_LobbyGameMode	Waiting room. Set as Lobby Map in settings.
Game	BP_GameGameMode	Your gameplay map. Set as Game Map in settings.

9 Blueprint Integration

The entire entry flow is one node. In your Main Menu map's Level Blueprint (or a menu Actor), on BeginPlay:

- Create Widget (Class = **WBP_QuickJoinMenu**).
- Call **Quick Join Menu Setup** on it. **It takes no parameters** - match type, lobby map and match size all come from Project Settings.
- Add to Viewport. Clicking the button finds or hosts a game and walks the player through the lobby into the match.

TIP On NULL/LAN, have player 1 reach the lobby before player 2 clicks - if both search at the exact same instant, each hosts its own lobby (the classic matchmaking race). Steam and EOS resolve this with real matchmaking.

Calling the subsystem directly (advanced)

- Get Game Instance > Get Subsystem (Class = **MultplayerSessionsSubsystem**).
- Call CreateSession / FindSessions / JoinSession / DestroySession / StartSession.
- Bind the matching delegate (e.g. MultiplayerOnFindSessionsComplete) BEFORE calling the function.

10 Setup Doctor and Debugging

Development builds include a self-check that catches the things that silently break multiplayer.

Setup Doctor

- Runs automatically when your Main Menu map loads (the start of every play session).
- Checks: Online Subsystem active, all three maps set and cooked, lobby widget set, sane player count.
- Prints a PASS / FIX report to the screen and to the Output Log under category **LogScarSession**.
- Run it any time from the console (press ~): `scar.RunDoctor`

```
==== Scar MultiSession Setup Doctor (map: menu_map) ====  
[Setup Doctor] PASS: Online Subsystem 'NULL' is active.  
[Setup Doctor] PASS: Lobby map '/Game/Maps/Lobby' is present.  
[Setup Doctor] PASS: Lobby Widget Class is set.  
[Setup Doctor] PASS: Required Players = 2.  
[Setup Doctor] All checks passed.
```

Logging and debug toggle

- **LogScarSession** (Output Log, or the packaged Saved/Logs/*.log) traces the full flow: create, find, join, the resolved connect address, and travel.
- On-screen status messages are gated by a console variable. Ship clean with `scar.ShowDebug 0` (default 1 = on).

11 Plugin Classes Reference

UScarMultiSessionSettings UDeveloperSettings (Project Settings)

- MainMenuMap / LobbyMap / GameMap (soft map references)
- RequiredPlayers, CountdownDuration, MatchType
- LobbyWidgetClass, LoadingScreenWidgetClass
- Get() ; GetLobbyURL() / GetGameURL() / GetMainMenuURL()

UMultiplayerSessionsSubsystem GameInstanceSubsystem (auto-init)

- CreateSession(int32 NumPublicConnections, FString MatchType)
- FindSessions(int32 Max) ; JoinSession(Result)
- QuickJoinOrHost(int32 NumPublicConnections, FString MatchType)
- DestroySession() ; StartSession() ; IsSessionInterfaceValid()
- Delegates: OnCreate / OnFind / OnJoin / OnDestroy / OnStart Complete

UQuickJoinMenu UUserWidget (entry menu)

- QuickJoinMenuSetup() - no parameters; reads Project Settings
- Bound button: QuickJoinButton

ULobbyMenu UUserWidget (lobby waiting room)

- No public API; driven by the lobby GameState
- Bound: StatusText, LeaveButton, ForceStartButton?, CountdownText?

AScarLobbyGameMode AGameModeBase (lobby levels)

- ForceStartMatch() // host only
- GetCurrentPlayerCount() ; GetRequiredPlayerCount()
- OnPlayerCountChanged(Current, Required)

AScarLobbyPlayerController APlayerController (lobby) - NEW

- Client_SetupLobbyUI(WidgetClass) // spawns UI on the owning client
- Server_RequestForceStart() // host-validated

AScarLobbyGameState AGameStateBase (lobby) - NEW

- Replicated: CurrentPlayers, RequiredPlayers, bMatchStarting, CountdownEndServerTime

AScarMultiSessionGameMode AGameModeBase (game levels)

- Subclass in BP or C++; override PostLogin / Logout

FScarSetupDoctor C++ helper (development) - NEW

- Run(UWorld*) - validates project setup; also via "scar.RunDoctor"

12 Common Issues and Fixes

First, run the Setup Doctor

Press ~ and type `scar.RunDoctor` (or read the screen / Output Log on Play). It pinpoints most setup problems - missing maps, OSS not active, unset lobby widget - in one report.

Match never starts / both players host their own lobby

On NULL/LAN both players searched before either advertised. Let player 1 reach the lobby first, then have player 2 click. Always test NULL with two separate processes (packaged or Standalone, not single-process PIE).

Travel fails / "missing map" in a packaged build

Make sure the three maps are picked in Project Settings (they are soft references and cook automatically). You can also add them under Project Settings > Packaging > List of Maps to Include.

"Only Local Player Controllers can be assigned to widgets"

Resolved in 2.1 - the lobby UI is created on each client by `AScarLobbyPlayerController`. Ensure `BP_LobbyGameMode` does not override Player Controller Class.

Cannot move after entering the game map

Resolved in 2.1 - input is restored to Game mode on lobby exit. If you use a custom pawn/controller, ensure it sets Game input mode on `BeginPlay`.

"Couldn't Launch PIE Client" / NetDriver failure

Your `+NetDriverDefinitions` line is split across multiple lines in `DefaultEngine.ini`. Put it back on one line under `[/Script/Engine.Engine]`, or delete it for LAN/NULL. Restart the editor afterwards.

Find Sessions returns zero results

Steam/EOS: both machines online and on the same backend. LAN: same network, separate processes. `MatchType` must match between create and find (it comes from settings, so it will).

ForceStartButton shows on clients

Host detection uses authority on the owning controller. Confirm Lobby Widget Class points to `WBP_LobbyMenu` and that `ForceStartButton` exists with that exact name.

13 Quick Reference Card

Everything on one page.

Item	Value / Action
Plugin folder name	ScarMultiSession
Enable via	Edit > Plugins > search ScarMultiSession
Configure everything in	Project Settings > Plugins > Scar MultiSession
Entry widget parent	UQuickJoinMenu (call QuickJoinMenuSetup(), no params)
Lobby widget parent	ULobbyMenu
Lobby GameMode parent	AScarLobbyGameMode (no per-map config)
Game GameMode parent	AScarMultiSessionGameMode
Match size	Required Players (one number = session size + start)
Validate setup	console: scar.RunDoctor
Silence on-screen debug	console: scar.ShowDebug 0
Diagnostic log category	LogScarSession
Pick backend	DefaultEngine.ini DefaultPlatformService = NULL / Steam / EOS

SUPPORT Questions or issues? Join the Black Scar Studio Discord or use the support link on the product page. Thank you for using ScarMultiSession.