

BLACK SCAR STUDIO

ScarCore Communication System

Setup Guide | Version 3.6

Multiplayer chat, voice (even on dedicated servers), quick messages, and ping.

ENGINE	Unreal Engine 4.27 and 5.0 - 5.7
SUBSYSTEMS	Steam Epic Online Services LAN (Null) Dedicated Servers
SETUP	Blueprint-only. One component, no C++, no input mappings.
GET IT ON	Fab Marketplace

1 What you get

ScarCore Communication System adds working multiplayer communication to your game: server-authoritative text chat, voice chat with push-to-talk, adjustable voice volume (master, per-player, and mic gain), a quick-messages panel, a per-player mute list, and a live ping readout. You add one component to your PlayerController and press Play. That's the whole setup.

Everything is Blueprint-friendly and nothing is hard-coded to a platform. Your DefaultEngine.ini decides whether you're on Steam, EOS, LAN, or a dedicated server - the plugin adapts to whatever it finds.

Class	What it's for
ScarCore Communication Component	Add this to your PlayerController. Holds all settings, creates the UI, exposes the Blueprint API.
ScarCoreBaseUI	The pre-built UI - the FULL feature set with zero designer work: a black-tinted chat panel with a slide button that collapses / expands it, mic / speaker buttons, a QUICK button + quick-messages list (seeded with default phrases), a PLAYERS button + per-player mute list, a master volume slider, and the ping label. Toggle any part off or restyle every color / label via properties (ScarCore BaseUI).
ScarCoreCommunicationWidget	Parent class for your own custom UI (Section 6).
ScarQuickMessageRow / ScarPlayerMuteRow	Row widgets for the two popups. Reparent a WBP to restyle them.

Under the hood: chat sender names are resolved on the server from PlayerState (no spoofing), and the profanity filter and rate limiter run server-side. Voice is captured locally, Opus-compressed, and relayed over the game's own replicated connection - so it works on **dedicated servers** and direct-IP (no-session) setups out of the box, not just listen servers. A legacy OnlineSubsystem-voice path is still built in for session-based games; one toggle (**Use Custom Voice Relay**) switches between them. You don't have to manage any of it.

2 Quick start

- 1. Install and enable.** In the Epic Games Launcher, install the plugin to your engine version (4.27 - 5.7) from your Fab Library. Then in your project: Edit > Plugins > search "ScarCore" > tick Enabled > restart.
- 2. Add the component.** Open your PlayerController Blueprint, click Add Component, and add **ScarCore Communication Component**.
- 3. Pick the UI.** Select the component and set **Main Widget Class** to **ScarCoreBaseUI**.
- 4. Set up your platform (one click).** Open **Edit > Project Settings > ScarCore Communication**, pick **Target Platform** (Steam / EOS / **Local / Null**), and click **Apply Platform Config**. It writes the DefaultEngine.ini entries for you - online subsystem plus voice. Restart the editor. (Prefer to paste it by hand? Section 3 has every block.)

Local / Null needs no credentials - it's the one to pick for LAN, dedicated servers, direct-IP, and single-machine standalone testing.

Press Play (2+ players, listen server). You'll see the chat panel bottom-left, mic and speaker buttons above it, and the ping top-right. Press **Enter** to type, **M** to turn your mic on, hold **V** to talk, and **N** to hear everyone else.

Players join muted on purpose. The mic starts OFF and the speaker starts MUTED, so nobody broadcasts or hears anything until they press M and N. If your game wants open comms from the first second, untick **Start With Speaker Muted** (and optionally tick **Start With Mic On**) on the component.

3 Online subsystem config (copy-paste)

The fast way: **Project Settings > ScarCore Communication > Apply Platform Config** writes all of this for you (Section 2, step 4). The blocks below are the manual equivalent and a reference for exactly what gets written - use whichever you prefer.

Pick the block for your platform and put it in **YourProject/Config/DefaultEngine.ini**. Restart the editor afterwards - Unreal loads the subsystem once at startup.

LAN / local testing (Null) - no accounts, no credentials

```
[Voice]
bEnabled=true

[OnlineSubsystem]
bHasVoiceEnabled=true
DefaultPlatformService=Null
```

Steam

Enable the **Online Subsystem Steam** plugin in Edit > Plugins first, and have the Steam client running when you test.

```
[Voice]
bEnabled=true

[OnlineSubsystem]
bHasVoiceEnabled=true
DefaultPlatformService=Steam

[OnlineSubsystemSteam]
bEnabled=true
SteamDevAppId=480
GameServerQueryPort=27015
bRelaunchInSteam=false
bVACEnabled=false

[/Script/OnlineSubsystemSteam.SteamNetDriver]
NetConnectionClassName="OnlineSubsystemSteam.SteamNetConnection"

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

App ID 480 is Valve's free test app - it works for development. Swap in your real App ID before shipping.

Epic Online Services (EOS)

Enable the **Online Subsystem EOS** plugin, create an artifact at dev.epicgames.com/portal, and fill in your own ids below. The encryption key must be exactly 64 hex characters.

```
[Voice]
bEnabled=true

[OnlineSubsystem]
bHasVoiceEnabled=true
DefaultPlatformService=EOS

[OnlineSubsystemEOS]
bEnabled=true

[/Script/OnlineSubsystemEOS.NetDriverEOS]
bIsUsingP2PSockets=true

[/Script/Engine.GameEngine]
!NetDriverDefinitions=ClearArray
+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="OnlineSubsystemEOS.NetDriverEOS",DriverClassNameFallback="OnlineSubsystemUtils.IpNetDriver")

[/Script/OnlineSubsystemEOS.EOSSettings]
DefaultArtifactName=DefaultArtifact
bEnableOverlay=true
bEnableSocialOverlay=true
bUseEAS=true
bUseEOSConnect=true
bUseEOSSessions=true
+Artifacts=(ArtifactName="DefaultArtifact",ClientId="<id>",ClientSecret="<secret>",ProductId="<id>",SandboxId="<id>",DeploymentId="<id>",EncryptionKey="<64-hex-chars>")
```

Dedicated servers (optional tuning)

No special setup is needed - the component detects dedicated servers and runs no UI or timers there. **Voice works on dedicated (and direct-IP) servers by default:** the custom relay sends compressed frames over the game connection, so no OnlineSubsystem session is required. The values below are just sensible network tuning if you want them:

```
[/Script/OnlineSubsystemUtils.IpNetDriver]
NetServerMaxTickRate=60
MaxClientRate=30000
MaxInternetClientRate=30000

[/Script/Engine.GameSession]
MaxPlayers=64
```

Mic sensitivity (optional)

```
[Voice]
SilenceDetectionThreshold=0.01
MicNoiseGateThreshold=0.01
MicInputGain=1.0
MicOutputGain=1.0
```

Mobile permissions (only if you ship there)

```
; Android - DefaultEngine.ini
[/Script/AndroidRuntimeSettings.AndroidRuntimeSettings]
+ExtraPermissions=android.permission.RECORD_AUDIO
+ExtraPermissions=android.permission.MODIFY_AUDIO_SETTINGS
+ExtraPermissions=android.permission.INTERNET
+ExtraPermissions=android.permission.ACCESS_NETWORK_STATE

; iOS - DefaultEngine.ini
[/Script/IOSRuntimeSettings.IOSRuntimeSettings]
AdditionalPlistData=<key>NSMicrophoneUsageDescription</key><string>Voice chat with other players.</string>
```

4 Default controls

The widget binds these keys on its own - no Project Settings > Input work, no Blueprint wiring. Every key is a property under the widget's Class Defaults (ScarCore | Input | Auto-Bindings) if you want different ones, and **Auto Register Key Bindings = false** turns the whole thing off so you can wire your own.

Key	Does
Enter	Focus the chat box / send what you typed / restore the chat UI after Escape hid it.
V (hold)	Push-to-talk. Only transmits while the mic toggle is ON. (Component setting.)
M	Mic toggle on / off.
N	Speaker toggle - mute / unmute all incoming voice.
C	Open / close the quick-messages panel.
Up / Down	Move the quick-message selection (wraps around).
Escape	Close the quick-messages panel; press again to hide the chat UI.
Tab	Open / close the per-player mute list.
Mouse wheel	Scroll the chat. While the quick-messages panel is open, moves the selection instead.

While the chat box has focus, keys type into it - Enter sends and hands control back to the game.

5 The settings you'll actually touch

Everything lives on the component, in the Details panel under **ScarCore | ...** categories. These are the ones that matter day-to-day; the rest (colors, rate limits, UI visibility toggles) are right next to them and self-explanatory.

Setting	What it does	Default
Main Widget Class	Which UI to spawn. ScarCoreBaseUI or your own widget.	(none)
Start With Mic On	Spawn with the mic already transmitting-ready.	false
Start With Speaker Muted	Spawn deaf until the player presses N.	true
Use Push To Talk	On = hold V to talk. Off = open mic while the toggle is ON.	true
Push To Talk Key	The PTT key.	V
Use Custom Voice Relay	On = self-hosted Opus relay over the game connection (works on dedicated / direct-IP servers, no session). Off = legacy OnlineSubsystem voice (listen-server / session setups).	true
Voice Mode	Global = hear everyone. Proximity = hear only nearby players.	Global
Voice Output Volume	Starting master volume for incoming voice (a slider / Blueprint can change it live).	1.0
Mic Input Gain	Multiplier on this player's own mic before it's sent.	1.0
Proximity Radius	Hearing range in cm for Proximity mode.	1200
Max Message Length	Chat messages are truncated past this.	256
Profanity Filter Enabled + Blocked Words	Server-side word filter. The shipped word list is a placeholder - add your own.	on

Setting	What it does	Default
Chat Font Size	Text size in the chat log.	14
Ping Good / Medium Threshold	Color steps for the ping label (green / yellow / red).	50 / 100

6 Build your own UI

Optional. **ScarCoreBaseUI** already ships every feature - chat, mic/speaker, quick messages, the per-player mute list, a volume slider, and ping. Only build a custom widget if you want a different look; otherwise skip this section.

This is where you make the chat system look like *your* game. You design a normal Widget Blueprint with any layout, art, and animations you want - the plugin finds its widgets by **name** and wires them up automatically.

1. Content Browser > right-click > User Interface > **Widget Blueprint**, and pick **ScarCoreCommunicationWidget** as the parent class. (For an existing WBP: open it, File > Reparent Blueprint.)
2. Design your layout. Wherever a feature should live, add the matching widget from the table below and rename it to the **exact name** shown.
3. Tick **Is Variable** on each of those widgets (top of the Details panel).
4. On the component, set **Main Widget Class** to your new widget. Done - chat, voice, popups, and key bindings all work in your layout.

Names are **case-sensitive** and every widget is **optional**. Skip what you don't need - a missing widget just means that feature has no UI. Nothing breaks.

The named widgets

Name it exactly	Widget type	What it powers
ChatScrollBox	Scroll Box	Chat messages appear here, auto-scrolled.
ChatInputTextBox	Editable Text Box	The typing field. Enter sends.
SendButton	Button	Sends the typed text.
ChatPanel	Any panel	Optional wrapper around the chat - hidden when chat UI is disabled.
MicButton	Button	Mic toggle. Gets your on / off icons stamped on it.
SpeakerButton	Button	Speaker toggle. Gets your on / off icons too.
VoiceVolumeSlider	Slider	Optional. Master incoming-voice volume - auto-wires, no Blueprint. Set Min 0 / Max 1 (or 2 for boost).
VoicePanel	Any panel	Optional wrapper around the voice buttons.
PingTextBlock	Text Block	Live ping, color-coded by your thresholds.
QuickMessagesPanel	Vertical Box	Quick-message rows spawn in here. Set it Collapsed by default; C opens it.
QuickMessagesButton	Button	Optional on-screen button that toggles the panel.
PlayerListPanel	Vertical Box	Mute-list rows spawn in here.

Name it exactly	Widget type	What it powers
PlayerListContainer	Border (or any widget)	Optional styled wrapper around PlayerListPanel - the whole thing shows / hides together. Set it Collapsed by default; Tab opens it.
OpenPlayerListButton	Button	Optional on-screen button for the mute list. Hides while the speaker is muted (configurable).

A layout that works (copy this hierarchy)

```

Canvas Panel
ChatPanel Vertical Box bottom-left
ChatScrollBox Scroll Box
HorizontalBox
ChatInputTextBox Editable Text Box
SendButton Button
VoicePanel Horizontal Box above the chat
MicButton Button
SpeakerButton Button
PingTextBlock Text Block top-right
QuickMessagesPanel Vertical Box centered, Collapsed
PlayerListContainer Border right side, Collapsed
PlayerListPanel Vertical Box
OpenPlayerListButton Button near the voice row

```

Icons and styling

In your widget's Class Defaults you'll find texture slots for every state: **Mic On / Off Icon**, **Speaker On / Off Icon** (ScarCore | Voice | Icons), icons for the open-list button, and per-row speaker icons for the mute list. Assign a texture and the plugin stamps it onto the button whenever the state changes, with hover and pressed tints applied for free - one texture per state is enough. Leave a slot empty to keep whatever brush you designed.

Custom popup rows

The quick-message and mute-list rows are built procedurally by default, styled from properties on your main widget (font size, colors, padding, row spacing). To go fully custom:

- **Quick messages:** make a WBP with parent **ScarQuickMessageRow**, add a Button named **RowButton** and a Text Block named **RowLabel**, then set **Quick Message Row Class** on your main widget.
- **Mute list:** make a WBP with parent **ScarPlayerMuteRow**, add a Button named **MuteButton** and a Text Block named **NameText**, then set **Player Mute Row Class**.

Animation hooks

Four events fire on your widget so you can play UMG animations: **BP On Player List Show / Hide Requested** when Tab opens or closes the mute list, and **BP On Chat UI Hide / Show Requested** when Escape hides the chat and Enter brings it back. Visibility and the mouse cursor are already handled by the time they fire - the animation is pure polish.

7 Quick messages and the mute list

Quick messages

Canned phrases players send without typing. Fill the **Quick Messages** array in your widget's Class Defaults (ScarCore | QuickMessages) - one entry per row, in order. In game: **C** opens the panel, **Up / Down** picks, **Enter** sends, **Escape** closes. Messages go through the normal chat pipeline, filter and rate limit included. The panel

auto-closes after sending and shows the mouse cursor while open (both configurable).

Per-player mute list

Tab opens a roster of every other player in the session with a speaker icon per row - click to mute or unmute that player *locally*. The muted player isn't notified, nobody else is affected, and the choice sticks for the whole match (even across a reconnect). The list refreshes itself every second while open, and works on clients as well as the host.

From Blueprint: **Mute Player**, **Is Player Muted**, and **Get Active Remote Voice Players** on the component let you build the same feature into your own scoreboard instead.

8 Blueprint API

Component functions

Function	Does
Send Chat Message	Send text to all players (default colors).
Send Chat Message Colored	Send with custom name / body colors - team chat, announcements.
Add Local Message	Show a message on this screen only. Never replicated.
Add Blocked Word / Set Profanity Filter Enabled	Manage the word filter at runtime.
Toggle Mic / Toggle Speaker	Same as pressing M / N.
Hard Mute Mic / Unhard Mute Mic	Force-silence the mic (overrides everything). For settings menus.
Set Voice Mode	Switch Global / Proximity at runtime.
Set / Get Voice Output Volume	Master volume for all incoming voice on this client. Wire to a slider (0-1).
Set / Get Player Voice Volume	Per-player incoming volume by net id - turn individual players up or down. Local, persists across their reconnect.
Set / Get Mic Input Gain	How loud this player's own mic transmits (multiplier).
Set Debug Voice Loopback	Hear your own mic looped back through the full pipeline. One-machine voice test; leave OFF for real play.
Mute Player / Is Player Muted	Local per-player mute by unique net id.
Get Active Remote Voice Players	Ids + display names of everyone else in the session.
Is Mic Toggle Enabled / Is Mic Hard Muted / Is Speaker Muted / Is Transmitting	State queries.
Get Current Ping Ms	Ping in ms. 0 on the listen-server host (no round trip - that's correct).

Widget functions

Function	Does
Focus / Unfocus Chat Input, On Enter Pressed	Drive the chat flow from your own input events.

Function	Does
Toggle / Open / Close Quick Messages	Control the quick-message panel.
Select Previous / Next, Send Selected Quick Message	Keyboard navigation, your way.
Open / Close / Toggle Player List Panel, Refresh Player Mute List	Control the mute popup.
Scroll Up / Scroll Down	Chat scrolling (or selection movement while the panel is open).
Apply Feature Visibility	Re-apply chat / voice / ping visibility after changing component toggles.

Events

Event	Fires when
On Message Received (component + widget)	A chat message arrived. Play a sound, flash the HUD.
On Mic State Changed / On Speaker State Changed	The toggles flipped. Update your own indicators.
On Remote Player Talking Changed (widget)	A remote player started / stopped talking.
BP On Player List Show / Hide Requested (widget)	Mute popup opened / closed - play your animation.
BP On Chat UI Hide / Show Requested (widget)	Escape hid the chat / Enter restored it.

9 Five-minute recipes

- **Welcome message:** PlayerController BeginPlay > branch Is Local Player Controller > Delay 0.5s > Add Local Message with sender "System".
- **Server announcement:** a Run-on-Server event that calls Send Chat Message Colored with your announcement color.
- **Mute checkbox in your settings menu:** checked = Hard Mute Mic, unchecked = Unhard Mute Mic.
- **Talking indicator over heads:** override On Remote Player Talking Changed, find the matching PlayerState in the GameState's Player Array, toggle an icon on their pawn.
- **Mute from your scoreboard:** Get Active Remote Voice Players for the roster, Is Player Muted for current state, Mute Player on your button.
- **Ping in your own HUD:** a 1s timer that calls Get Current Ping Ms and writes it to your text block.
- **Voice volume slider:** drop a Slider named **VoiceVolumeSlider** (Min 0, Max 1) into your widget - it auto-controls master voice volume, no wiring. For a settings menu, bind your own slider's On Value Changed to **Set Voice Output Volume** instead.
- **Per-player volume:** build a row list from Get Active Remote Voice Players, give each row a slider, and call **Set Player Voice Volume** (player id, value). Use **Set Mic Input Gain** for the player's own transmit level.

10 Test it, then ship it

Testing chat + voice on one PC

Play dropdown > **Number of Players = 2, Net Mode = Play As Listen Server**. For **chat**, keeping Run Under One Process is fine - type in one window, watch it in both. For **voice**, untick **Run Under One Process** so each player is its own standalone process (they'd otherwise share one audio session).

Both windows share your single mic, so split the roles: in the **host** window press **M** (mic on), and in the **client** window press **N** (speaker on). Talk - the client plays your voice back. With **Use Push To Talk** off, M/N behave like a normal game (mic on = talk, speaker on = hear); with it on, hold **V** to talk.

Hear yourself with no second player. Call **Set Debug Voice Loopback (true)** on the component (or tick **Debug Voice Loopback** in its Details) and just talk - it starts the mic on its own and echoes you through the full capture > Opus > decode > playback path. No mic toggle, no push-to-talk, no server needed. Turn it OFF for real matches.

This is the #1 thing that fools people. A background (unfocused) window is muted by default, so on one PC you hear your own loopback (focused window) but never the OTHER player's voice (it plays in the unfocused window). **Apply Platform Config now writes the fix for you (UnfocusedVolumeMultiplier=1.0 under [/Script/Engine.Engine]).** If you configured the ini by hand, add that line yourself.

Shipping checklist

- DefaultEngine.ini has the voice block plus your platform block (Section 3).
- Steam: real App ID (not 480) and the Online Subsystem Steam plugin enabled.
- EOS: real artifact ids and a 64-character hex encryption key.
- Component on the PlayerController your shipped GameMode actually uses, Main Widget Class set.
- Custom UI: parent class is ScarCoreCommunicationWidget, named widgets spelled exactly, Is Variable ticked.
- Startup states reviewed: Start With Mic On / Start With Speaker Muted match your design.
- Voice path chosen: Use Custom Voice Relay stays ON for dedicated / direct-IP servers; flip OFF only for session-based OSS voice. Debug Voice Loopback is OFF.
- Blocked Words filled with a real list; Quick Messages filled if you use the panel.
- Default keys don't clash with your game's bindings (rebind in Class Defaults if they do).
- Mic permissions added for Android / iOS if you ship there (Section 3).
- One Shipping build, two real players, chat and voice confirmed.

11 When something's off

I can't hear anyone

The speaker starts muted by design - press N. Also check Tab: you may have muted them earlier.

No one hears me

The mic starts OFF - press M first. PTT only transmits while V is held and the mic is ON. Check Windows mic permissions (Settings > Privacy > Microphone).

Voice doesn't work at all

The default custom relay still needs [Voice] bEnabled=true and a DefaultPlatformService (for player identity), and an editor restart after editing the ini - but it does NOT need an online session. Check Windows mic permissions. Legacy path only (Use Custom Voice Relay = false): also needs bHasVoiceEnabled=true, a live session, and (Steam) the client running / plugin enabled.

No UI appears

Main Widget Class not set, Auto Create UI unticked, or the component isn't on a PlayerController. The Output Log (filter: LogScarCoreCommunication) will tell you which.

Chat only shows on my screen

You're in single-player. Set Number of Players to 2+ and Net Mode to Play As Listen Server.

Some players are silent in Proximity mode

They're outside Proximity Radius. That's the feature - raise the radius or switch to Global.

My custom widget's buttons do nothing

A name is off by one character (they're case-sensitive), Is Variable isn't ticked, or the parent class isn't ScarCoreCommunicationWidget.

Tab shows an empty list

The list excludes you, so it's empty until someone else joins. In a custom widget the container must be named PlayerListPanel exactly.

Quick messages don't show up

The Quick Messages array is empty, or your panel isn't named QuickMessagesPanel.

Keys clash with my game

Every key is rebindable in the widget's Class Defaults, or set Auto Register Key Bindings = false and call the widget functions from your own input events.

12 License and support

You can: use the plugin in unlimited personal and commercial games, customise it through Blueprint children, and ship it inside your packaged game on any platform.

You can't: resell or redistribute the plugin itself, upload it anywhere, or bundle it in tools or asset packs meant for other developers.

Stuck? Check Section 11 first, then look for **LogScarCoreCommunication** lines in the Output Log - the plugin says what it's doing.

- Discord: <https://discord.gg/a3QdMX9JZj>
- Web: <https://www.blackscarstudio.com>
- Fab: post on the product page's Questions tab.