



CROWN NOIR

The Official House Archive

Digital Jewelry Guide Book for GTA V / GTA VI / FiveM

OFFICIAL CROWN NOIR PRODUCT

Built to Shine · Made to Last · Made to Lead

CROWN NOIR

The Official House Archive

A digital jewelry guide book for creators, server owners and players in GTA V, GTA VI and FiveM.

— Built to Shine • Made to Last • Made to Lead —

© 2026 Maison Crown Noir. All rights reserved.
This is an official Crown Noir product. Unauthorized redistribution or resale is prohibited.

Not affiliated with Rockstar Games or Cfx.re.

Contents of the Archive

Founder Letter
From the Maison

Chapter 01 — The Atelier
3D Creation Pipeline

Chapter 02 — The Mechanism
FiveM Lua Scripting

Chapter 03 — The House
Monetization & Legacy

Chapter 04 — The Wearer
Player Guide

— From the Founder —

Crown Noir is not a shop. It is a House. Every piece we release is a document of craftsmanship — modelled, engraved and inscribed for those who understand that the truest measure of luxury is what remains after the moment has passed.

This archive opens the House to you. Inside you will find the complete path from a raw 3D mesh to a wearable, sellable item inside a FiveM server. You will learn how to price it, protect it, and how players will use it once it lives in their world.

Read slowly. Return often. Build something that outlasts the trend.

— The Maison —

CHAPTER 01

3D Creation Pipeline

From Blender mesh to a streamable FiveM prop.

— Crafted for Legacy —

What you're actually building

A piece of digital jewelry in GTA / FiveM is a prop — a small 3D model (a .ydr drawable) with its textures (.ytd) and metadata (.ymt) that the game can stream in, attach to a bone on the player, and render like any other prop.

There is no 'jewelry' file format. You are building the same asset type Rockstar uses for a coffee cup or a suitcase, then attaching it to the neck, wrist or finger of the player skeleton.

Tools you need

- Blender 4.x — modeling and UV unwrapping.
- Sollumz — Blender addon that exports directly to GTA V formats (YDR, YDD, YTD).
- CodeWalker — inspect models, verify collisions, preview in-game before streaming.
- OpenIV — texture packing and RPF inspection (optional but useful).
- Substance Painter / Photoshop / GIMP — texture authoring (metals, gems, roughness maps).

The pipeline, step by step

01. Model low-poly

Jewelry sits close to the camera, but it's still a small object on-screen most of the time. Target 2k-8k tris for a full chain. Rings and studs: under 1k tris.

02. Bevel the edges

Sharp jewelry looks CG. Bevel every visible edge slightly (0.2-0.5 mm scale) so highlights catch the light. This is the single biggest visual win.

03. UV unwrap cleanly

Metals need clean UVs so the roughness / normal map wraps predictably. Diamonds and gems get their own material with a separate UV island.

04. Author PBR textures

Base color, metallic (near 1.0 for gold/silver), roughness (0.1-0.3 for polished, 0.5-0.7 for brushed), normal. GTA V uses a diffuse + spec workflow — Sollumz converts PBR on export.

05. Set up the Sollumz shader

In Blender, add a Sollumz Drawable and assign the vehicle_paint or normal_spec shader depending on the material. Set diffuse and spec samplers.

06. Export YDR + YTD

Sollumz > Export Drawable. You get a .ydr (mesh) and a .ytd (texture dictionary) — the two files FiveM will stream.

07. Preview in CodeWalker

Drop the .ydr into CodeWalker's RPF explorer or a test .ymap and confirm it renders, has correct scale, and no missing textures.

08. Package for streaming

Place files in your FiveM resource's stream/ folder. FiveM auto-loads any .ydr / .ytd found there.

Folder layout inside your FiveM resource

TEXT

my-jewelry/

├─ fxmanifest.lua

├─ stream/

```
|   ├── jewel_chain_cuban_gold.ydr
|   ├── jewel_chain_cuban_gold.ytd
|   ├── jewel_ring_signet.ydr
|   ├── jewel_ring_signet.ytd
|   └── jewel_watch_ap_gold.ydr
├── client/
|   └── attach.lua
├── server/
|   └── main.lua
└── shared/
    └── catalog.lua
```

Minimal fxmanifest.lua

LUA

```
fx_version 'cerulean'

game 'gta5'

lua54 'yes'

author 'Your Studio'

description 'Custom jewelry props + wear system'

version '1.0.0'

files {
    'shared/catalog.lua',
}

shared_script 'shared/catalog.lua'

client_script 'client/attach.lua'
```

```
server_script 'server/main.lua'
```

```
-- FiveM auto-streams anything in stream/, no manifest entry needed
```

Attachment bones — pick the right one

Jewelry is attached to skeleton bones via `AttachEntityToEntity`. The most common bones: `SKEL_Head` — earrings, grillz (with an offset); `SKEL_Neck_1` (bone id 39317) — chains, pendants; `SKEL_L_Hand` / `SKEL_R_Hand` — rings, bracelets; `BONETAG_L_FINGER1`..`BONETAG_L_FINGER5` — individual finger rings.

What this guide does NOT cover

This is a scripting-first guide. Deep 3D authoring — Blender fundamentals, UV theory, Substance material graphs — has better dedicated resources on YouTube and the Sollumz wiki. Once your `.ydr` and `.ytd` exist, jump to Chapter 02 — Scripting to wire them into the game.

CHAPTER 02

FiveM Lua Scripting

Turn a prop file into a wearable, sellable item.

— Built to Shine —

Architecture

The jewelry resource has three responsibilities:

- Catalog — a shared table describing every piece (model, bone, offset, price).
- Inventory — items players can own (usually via ox_inventory / qb-inventory).
- Wear system — client-side attachment of the prop to the player's bone, synced to nearby players.

Shared catalog

Every piece lives in one shared file so client and server agree on the metadata.

LUA

```
-- shared/catalog.lua

Catalog = {

    ['chain_cuban_gold'] = {

        label    = 'Cuban Link Chain - Gold',

        model    = 'jewel_chain_cuban_gold',
```

```

        bone    = 39317,          -- SKEL_Neck_1

        pos     = vec3(0.0, 0.08, 0.0),

        rot     = vec3(0.0, 0.0, 0.0),

        slot    = 'neck',

        price   = 25000,

    },

    ['ring_signet'] = {

        label    = 'Signet Ring',

        model    = 'jewel_ring_signet',

        bone     = 26610,          -- BONETAG_L_FINGER1

        pos      = vec3(0.0, 0.0, 0.0),

        rot      = vec3(0.0, 0.0, 0.0),

        slot     = 'finger_l',

        price    = 8000,

    },

    ['watch_ap_gold'] = {

        label    = 'AP Royal Oak – Gold',

        model    = 'jewel_watch_ap_gold',

        bone     = 60309,          -- SKEL_L_Hand

        pos      = vec3(0.02, 0.0, 0.0),

        rot      = vec3(0.0, 90.0, 0.0),

        slot     = 'wrist_l',

        price    = 120000,

    },

}

```

ox_inventory item registration

Register each catalog entry as an inventory item so it can be traded, dropped and stored.

LUA

```
-- ox_inventory/data/items.lua (partial)

['chain_cuban_gold'] = {

    label = 'Cuban Link Chain - Gold',

    weight = 200,

    stack = false,

    close = true,

    client = {

        image = 'chain_cuban_gold.png',

        event = 'jewelry:client:equip',

        slot = 'neck',

    },

},
```

Client-side wear system

Each player tracks what they're wearing in a local table. Attaching is CreateObject + AttachEntityToEntity. State replicates via a state bag so other players see the prop.

LUA

```
-- client/attach.lua

local worn = {} -- [slot] = objectHandle

local function equip(itemName)

    local piece = Catalog[itemName]

    if not piece then return end

    -- Remove anything already in this slot

    if worn[piece.slot] then
```

```

        DeleteObject(worn[piece.slot])

        worn[piece.slot] = nil

    end

    local model = joaat(piece.model)

    lib.requestModel(model, 5000)

    local ped = PlayerPedId()

    local obj = CreateObject(model, 0.0, 0.0, 0.0, true, true, false)

    AttachEntityToEntity(

        obj, ped, GetPedBoneIndex(ped, piece.bone),

        piece.pos.x, piece.pos.y, piece.pos.z,

        piece.rot.x, piece.rot.y, piece.rot.z,

        true, true, false, true, 1, true

    )

    worn[piece.slot] = obj

    SetModelAsNoLongerNeeded(model)

    -- Sync to other players via state bag

    LocalPlayer.state:set('jewelry', worn_to_state(), true)

end

RegisterNetEvent('jewelry:client:equip', function(data)

    equip(data.name)

end)

RegisterNetEvent('jewelry:client:unequip', function(slot)

    if worn[slot] then

        DeleteObject(worn[slot])

    end

end)

```

```

        worn[slot] = nil

        LocalPlayer.state:set('jewelry', worn_to_state(), true)

    end

end)

-- Rebuild worn pieces on nearby players

AddStateBagChangeHandler('jewelry', nil, function(bagName, _, value)

    local ply = GetPlayerFromStateBagName(bagName)

    if ply == 0 or ply == PlayerId() then return end

    rebuild_remote(ply, value)

end)

-- Clean up on death / respawn

AddEventHandler('playerSpawned', function()

    for slot, _ in pairs(worn) do worn[slot] = nil end

    -- re-equip from server-authoritative state here

end)

```

Server-side validation

LUA

```

-- server/main.lua

local ox = exports.ox_inventory

RegisterNetEvent('jewelry:server:equip', function(itemName)

    local src = source

    local piece = Catalog[itemName]

    if not piece then return end

    -- Player must actually own the item

```

```

    local count = ox:GetItemCount(src, itemName)

    if count < 1 then

        return -- silently reject; log if you care

    end

    TriggerClientEvent('jewelry:client:equip', src, { name = itemName })

end)

RegisterNetEvent('jewelry:server:unequip', function(slot)

    TriggerClientEvent('jewelry:client:unequip', source, slot)

end)

```

Trust nothing from the client

Every equip request must be validated server-side against the player's actual inventory. A leaked client-only version of your resource will let users spawn any piece for free otherwise.

Framework variants

- ESX — replace `ox:GetItemCount` with `xPlayer.getInventoryItem(name).count`.
- QBCore — `Player.Functions.GetItemByName(name)`.
- QBox — same as QBCore but the export is `exports.qbx_core`.
- Standalone — persist wear state in a KVP or database table keyed by license identifier.

CHAPTER 03

Monetization

Turn your jewelry pack into a repeat-revenue product.

— **Luxury Lives Forever** —

Where creators actually sell

- Tebex (Cfx.re Store) — the only officially sanctioned marketplace. Supports escrow (buyer can use the resource but can't read the Lua). Required for anything sold to server owners publicly.
- Discord community stores — direct-sale via Tebex-hosted checkout is fine. Selling via PayPal F&F; to skirt Tebex is a Cfx.re ToS violation and gets your assets DMCA'd off servers.
- Commissions — custom pieces for individual servers, sold with a limited-use license.

Pricing brackets (2025 market)

Pack size	Typical price	What it includes
Single piece	\$5 - \$15	One chain / watch / ring, one texture variant.
Starter pack (10-20 pieces)	\$25 - \$60	Mix of chains, rings, grillz — no wear system.
Full pack (50+ pieces)	\$75 - \$150	Curated collection + wear/equip script.
Custom / branded server piece	\$150 - \$500+	Bespoke logo pieces, per-server license.

Tebex packaging checklist

- Enable escrow on the release version so the Lua ships encrypted.
- Include a README.md: framework requirements, dependencies (ox_inventory, ox_lib), install steps.
- Ship a preview video — a 30s in-game clip converts far better than static renders.
- Provide versioned changelogs. Buyers judge maintenance quality by cadence.
- Set an allowed CFX identifier per license to prevent resale.
- Offer a free demo pack (2-3 pieces, non-escrow) as a funnel.

License template (short version)

Every pack should ship with a license file. Minimum clauses:

- Grants use on one FiveM server (CFX ID pinned).
- No redistribution, resale, or public repository upload.
- Modification allowed for internal use only; modified copies inherit the license.
- Original 3D source files (.blend, .psd) are not included unless separately purchased.
- Refunds within 24h if the pack is unused and un-imported.

Anti-leak reality check

Escrow raises the cost of leaking — it does not prevent it. Assume every pack will eventually leak. Compete on cadence: frequent updates, custom pieces, and a healthy support Discord will keep paying customers even after a leak hits the illicit hubs.

What sells best

- Brand pieces — Cuban links, tennis chains, iced watches, grillz. High-recognition = high conversion.
- Themed collections — 'Street Kings', 'Yakuza', 'Cartel', 'Old Money'. Servers buy identities, not props.
- Female-fit pieces — chronically under-served. Fewer competing packs.
- Wear scripts bundled — a pack that 'just works' out of the box outsells raw model packs 3-5x.

CHAPTER 04

Player Guide

Buy it, equip it, flex it. Don't get robbed.

— Worn by Kings —

How to get jewelry in-game

On most FiveM servers, jewelry comes from one of four places:

- In-game jewelry store — usually marked on the map. Walk in, open the shop menu, pay cash or bank.
- NPC vendor / pop-up shop — hidden or event-based, higher-end pieces.
- Player-run business — a jeweler character who commissions or resells pieces from other players.
- Robbery loot — snatched chains and watches from other players (see below).

Equipping a piece

Once the item is in your inventory, the exact command depends on the server, but 90% follow one of these patterns:

- Right-click in ox_inventory → 'Wear' or 'Equip'.
- Chat command: /wear chain, /jewelry to open a menu.
- Radial menu: hold Z or F1 (server-dependent) → Jewelry → pick the piece.

To take it off: same menu, 'Unequip', or drop the item.

Slots — what can you wear at once?

A typical wear system enforces one piece per slot:

- Neck (1 chain)
- Left wrist / right wrist (1 watch or bracelet each)
- Left hand / right hand fingers (rings, up to 5 per hand on some servers)
- Ears (studs / hoops)
- Teeth (grillz)

Stacking two chains? Rare, and usually only on servers with a 'layered jewelry' plugin.

Roleplay etiquette

- Chains are targets. On serious-RP servers, walking through the hood in a \$50k Cuban link is an invitation. That's the RP — engage with it.
- Value your piece in-character. If you paid \$25k in-game, don't hand it over for a \$500 threat. Also don't die on the hill either — fear RP matters.
- Chain snatching is a mechanic on many servers: /snatch or an inventory action transfers the piece from victim to robber. Learn if your server has it.
- Don't metagame the model name. 'OMG a jewel_chain_cuban_gold' is out-of-character. Describe it in-character: 'iced-out gold rope, ain't hers'.

If your jewelry disappears

Most likely cause: you respawned and the wear system didn't restore state. Re-equip from your inventory. If the item itself is gone from inventory, contact server support with your character name and approximate time — server logs will show it.

Showing off (the fun part)

- Use the in-game camera / /cam for close-ups.
- Pose menus (rpemotes, dpemotes) have 'show chain' and 'wristflex' animations.
- Screenshot at golden hour in-game (around 6-7 PM in-game clock) — the specular highlights on gold pop.

— The End of the First Volume —

Thank you for reading The Official House Archive.
Return to the archive as your catalog grows. The House is never finished.

Built to Shine · Made to Last · Made to Lead

© 2026 Maison Crown Noir. Official product. Not affiliated with Rockstar Games or Cfx.re.