

Settlement and Misc Implementation Checklist

This document list the key points to take care of to ensure a settlement is properly implemented.

Do a first pass of QA

As the first person to check the level design, outside the original level designer itself, now is a good time to do a first pass of generic quality assurance:

Ensure there are no **level design mistakes**: no gaps, floating items, misaligned elements, or havok issues in the area you work on. Also look for any potential over downscaled statics which aren't supposed to be there, like very small statues (common new level designer mistake).

Check for **lighting issues** (both shadow caster count, and how many lights per mesh). AKA flickering lights.

Is the cell **optimized** with room bounds/occlusion planes? Only required for large cells such as castles and all dungeons type interiors.

Verify cell setup (lighting template, imagespace, external emittance, location, ownership, flags, acoustic space) to have a consistent settlement.

If there are too many issues, write them down and talk to the level design dept' for fixing.

Exterior Cells Setup

Cells should be **named** by the Landscaping department with the following convention:
<Prefix><SettlementName>0X

If one or more cells are missing names, assign one according to the above guidelines.

All the **exterior cells** of the settlement should have the **settlements location** assigned to them. **<Prefix><SettlementName>Location**.

Settlements should have one, and only one, **map marker** associated. Radius covering roughly the whole settlement area. Don't check any of the mapmarker specific checkboxes.

Map marker should have the **LocRefTypeMapMarker** assigned (used by quest aliases to auto reveal map markers).

Map marker should be **linked** to an **XMarkerHeading**. This XMarker is where the player should teleport after fast traveling here, by default its the map marker. (open map marker, linked ref and select xmarkerheading)

Settlements should have one **Location center marker**. It's an xmarker in the center of the settlement, with the LocationCenterMarker LocRefType, and named **<Prefix><SettlementName>CenterMarker**

Setup any **lights** in exteriors to have **FXLightInvertXXX external emittance** of some kind, so that they will fade and emit less light during the day.

Check that **items, containers and farm animals** in the exterior are **owned by the relevant faction**. Low value items and containers don't have to be owned. Exteriors cannot have an owner so this has to be done manually.

Pro tip: Use the reference batch action window to quickly do this on all selected refs (press “-” with the render window selected to open)


Add **guards** from the county **patrolling** the area if relevant.

Containers, valuable clutter and animals should be **owned** by the owning npc or the settlement faction. Crops should **not** be owned.

Make sure **navmesh** is **finalized**, shown by green markers along cell borders and under teleport markers.

Interior Cells Setup

For a more detailed guide, see this

 [Setting Up Interiors.docx](#)

Interior cells should be named **<Prefix><SettlementName><HouseName>**.

Example: CYRBrumaJerallViewInn, CYRBrumaRenodsHouse

Interior cells should have their **north marker** pointing to what would be north outside the cell. (The level design department should have added it. If it's missing, add it.)

Interior cells should have a **COCMarkerHeading**. This is the default point where you console teleport into the cell (using COC/COW) or when you fall off the map.

Interior cells should have their own **location** assigned, being a **child location of the settlement location**. If the interior has more than one cell (such as a basement), all cells use the same location.

Interior cells should also have their own **LocationCenterMarker** (used by quests to force npcs to go back to their cell to hide, for example).

Inns should have their own LocationCenterMarker, a **setup for bards** scene, and their **full Inn setup** (Innkeeper, Waiters / Innkeep backup, player's rentable bed, ...)

Create **locklists** for the interior. This is a formlist that contains all the NPCs living there who also own the **key** to the cell. Not required for permanently open buildings (with the public flag) like inns or chapels.

Ensure **ownership** of cells, as well as **locklists**, and everything related to trespassing. NPCs who live in the cell must be inside a locklist if the cell is ever locked.

Doors should all use the same **unique key** that all NPCs in the cell's locklist should carry.

Doors should be set as **locked ONLY from inside**. The inside door also has to be manually set as **owned** by the **faction** that owns the cell. Exterior side of the door should have no lock or faction assigned.

Cell owner NPCs should have the **Boss LocRefType**. When all NPCs marked as Boss are dead, location is marked as cleared and items are no longer owned by the cell owners.

Important containers (usually 1 or 2 chests) should have the **HouseContainerRefType** LocRefType (used for radiant quests where the player has to place or steal an item in/from the NPC's house).

The **main door** should have the **HouseMainDoorRefType** (used by the radiant system, usually to force the door to lock/unlock independently of NPC's AI).

If there is a **backdoor**, it should have the **HouseBackDoorRefType** (same as above, if need to force the player to go through maindoor only, or backdoor only).

Any **beds** should have the **HouseBedRefType**, and should be **owned** by the npc that sleeps in it. If the bed is a double bed, it should be owned by the interior faction.

A House location should have the **LocTypeHouse** and **LocTypeDwelling** keywords. A guild should have the guild keyword, etc

Merchant containers should have the **MerchantContainerRefType** LocRefType, mainly to easily identify them by the white marker(?). They must also always be out of bounds, not accessible by the player.

Set up any **natural lights** near windows etc. to have a **FXLightRegionXXX** (not inverted) in **external emittance** so they will become less bright during the night. Same for any dust beam FX.

Check that the windows of exterior architecture assets have windows correctly glow during the night. If they don't, complain to the art team.

Check that interactable furniture (chairs, benches, crafting tables, idle markers) are correctly placed and don't clip NPCs or the player into the ground or get them stuck inside objects when used.

Make sure **navmesh** is **finalized**, shown by green markers along cell borders and under teleport markers.

NPCs

See

<https://docs.google.com/document/d/1R1bobBdg44Vf3Rk6U7aQC4PmV-GPF1AfhkDGelooG8k/edit>

for more detailed instructions when creating new NPCs

NPCs should be **placed in the interior cells they live in**, next to their bed. This way they can be assigned generic editor location packages for common life packages such as sleeping and eating.

NPC references should have a **Persist Location**. Set this as the **settlement location** (example, all NPCs in Solitude have SolitudeLocation).

Ensure they got their counties' **crime faction** assigned, as well as the **settlement's faction** and their own **house faction**.

Add the NPC to their house's **locklist** and give them the **key** to the cell if any of those exist.

Balance the amount of **idle markers**, depending on what's been done by level design. If there are none / really few, add more. If there are too many of them, remove some. Amount should be roughly the same as for Riverwood (best example of Settlement implementation in Vanilla).

Idle markers and furniture that are exclusively used by merchants/bar keepers etc.(lean on counter, sweep floor, cook) should be **owned by a faction**. This stops random NPCs from using them. Make them **owned by the faction** that should use them (usually the cell's owner faction).

Ensure they have eat/sleep package, as well as daily occupation(s).
Select the DefaultHomeOwnerPackageList as Default Package List.

If they should eat outside, the package should point to the location (xMarkers + radius), not a specific furniture.

Check that NPCs buying resources such as produce from dialog options do so properly.

Check that any dialog actions that require gold (buying horses, giving beggars gold etc.) actually check that the player has gold, remove the gold properly and either have dialog for not having gold or hide dialog options in such a case.

Setup **relationships** for NPCs, mainly used in radiant dialogue. Relationship naming conventions are usually [Prefix][Parent][Child]. Always put the superior NPC first.

Unique NPCs need to have **WIDeadBodyCleanUpScript** attached to the actorbase. This is so their body doesn't linger in the area for the full duration until cell reset.

Autofill it, and fill the **death container property** with their death container:

There are three ways to setup the death container, depending on the kind of npc:

1. Unique NPCs **living in cities** should have a coffin as a **death container**. Coffins have to be pre-placed in the city's crypts (should be done by level design), initially disabled. Owner's name will be set by script (container being "show owner" flagged). Do ensure congruent EditorIDs though.
2. Unique NPCs **not living in cities** use strongboxes as **death containers**. You can place or reuse strongboxes that are already there. Use empty strongboxes, such as CYRTreasStrongBoxLowerEmpty or CYRTreasStrongboxEmpty.
3. **If it doesn't make sense** to have a **grave for an npc** (such as a crime boss dying in a farming settlement) their grave container should be set to an inaccessible location (vanilla uses a container in **WIDeadBodyCleanupCell**).

Dialogue

- Each settlement has one and only one unique **Dialogue quest** named **<Prefix>Dialogue<SettlementName>**
- This dialogue quest contains dialogues for **all NPCs**, including their specific Hellos, Goodbyes, etc...
- It has to be **start game enabled** flagged..
- This quest is also responsible for handling the **aliases** for the innkeeper, stablemaster and any NPC replacements if there are any.
- Dialogue can be imported from the docs via [this](#)
- Any "entry barks" for merchants must be manually added to the CYRWIMerchant quest scene
- Any other "entry barks" for regular people must be manually added to the CYRWIHouse quest scene

Special cases

Female NPCs who are warrior-types (guards, Fighters guild...) should use opposite gender animations.

Horses the player can buy should have **PersistAll** as their Persist Location. This is also for NPCs who move around a lot (Cicero, Astrid, Galmar, Khajiit caravans in vanilla game).

No underscores in cell names! There are reports of this corrupting save files.

Misc implementation

Setting up **merchants**:

- create **merchant chest** (duplicate already existing one but remove any unique items like investor lvl lists from it),
 - move it outside the map in the shop cell
 - add MerchantContainerLocRefType
- give NPC **<Prefix>JobMerchantFaction** and **<Prefix>Job<TypeOfMerchant>Faction** (controls their vendor dialog),
- create **new faction <Prefix>Services<SettlementName><ShopName>**
 - add NPC to faction,
 - **set vendor hours** in vendor tab,
 - **link the merch chest** to this faction,
 - set cell/area where shop will be,
 - select **vendorList** (what types of keywords merchant buys)
 - and optionally add conditions when the merchant should be available (like quest stages)
- Merchants usually have package with the template **SandboxAndKeepEyeOn** to keep watch on the player when he's running around the shop (they follow when the player is out of sight)
- make shop clutter be owned by the service faction, this will make the items available for sale in the vendor's inventory (press - in the render window to open the batch window)
- add new AI packages so NPC hangs around their shop area
- (optional) create new lvl list for **investor perk** effect (has to be done individually per vendor, see vanilla for examples) and add it to chest if investing should be option (can be done in batch later in dev)
- if NPC thinks you're trespassing or doesn't unlock cell doors, set interior door as owned by the NPCs faction and check that locklist exists
- Note that NPC will also sell items from their inventory (not equipped) with correct keywords, so don't put anything weird in there
- If there is no merchant chest, NPC will only sell things from their inventory

Setting up **innkeepers**:

- Innkeeper factions: **<Prefix>JobMerchantFaction**, **<Prefix>JobInnkeeperFaction** and **<Prefix>JobRentRoomFaction**

- Optionally CYRJobRentRoomFactionPrice[MiddleClass/UpperClass/Majestic] factions for higher quality inns
- Setup service faction, merchant chest and everything as you do for any merchant
- Rumors: **<Prefix>JobInnkeeper[County]RumorFaction** and if they're in a city **<Prefix>JobInnkeeper[City]RumorFaction**
- and add the **BSKRentRoomScript**.
 - Autofill **WI** property
 - and set **Bed** to the bed that the player should get when they rent a room

Setting up **gather/harvest services**:

- Faction: **<Prefix>FavorJobGatherResourcesFaction** (Usually BSK, unless for specialized services)
- Add profession keywords: BSKFarmerX BSKMinerX, etc.

Setting up **trainers**:

- NPC has to have a trainer class **Trainer[Skill]**,
- trainer factions: **<Prefix>JobTrainerFaction** and **<Prefix>JobTrainer[Skill]Faction**,
- and add the **TrainerGoldScript**.
 - Autofill **Gold001** property
 - and set **TrainerType** to 1=journeyman 2=expert or 3=master. (controls how much gold they keep from training or whatever, so they dont carry 10000's of gold)

Setting up **bards**:

- select the bard NPC, linked ref, choose the **xMarkerHeading** where he will play, choose the keyword **BardPlaySpot**.
 - xMarker should be named **<Prefix><SettlementName><TavernName>BardSpot** for clarity.
- NPC has to be in factions **<Prefix>JobBardFaction** and **<Prefix>JobBardSingerFaction**.
- add audience markers, NPCs who auto join scene use them (BardAudience1-6 locRefType) Add at least 4 of them with different numbered locRefType

Setting up **hostlers/stables**:

- See CYR stables quest, be careful of conditions when copy pasting

Setting up **NPC replacements** with aliases:

- Create your NPCs
- Add 2 aliases to **<Prefix>Dialogue<SettlementName>** quest for each pair - mark both as **Optional** (e.g. **<ShopOwner>** and **<ShopOwner>Backup**)
- Attach "**SwapAliasOnDeath**" script and define **<ShopOwner>Backup** as the script property.
- Ensure **<ShopOwner>** alias has all **factions and AI packs** required to perform the job (and items like keys if needed).

- Put whatever is needed for the merchant setup like the rent room script + service/job/rumor factions on the backup with factions being set to rank -1 (factions are used for the voice export)
- Backup alias only needs unique actor set
- When the primary NPC dies, the backup will take over their factions and schedules.
- Fix any dialog conditions where backup NPCs dialog doesnt match their new job

Setting up **patrols**:

- TBD

Setting up county **jail**:

- Place two PrisonMarker objects (this is a Door). One should be placed outside the building where the jail is located. This is where the player ends up after paying the fine or being released from jail. The other is inside the prison cell, and marks the location the player will be sent to if he submits. These two markers must link to each other in their teleport data.
- Also set up a xMarkerHeading as follower wait marker next to the exterior PrisonMarker.
- Make sure there is a locked jail door (the lock level can be chosen freely) with the owner set to the cell faction, and at least one bedroll inside the prison cell.
- There also need to be two chests, one for the player inventory and one for stolen goods nearby. Link those in the crime tab of the crime faction.
- Check that the player inventory evidence chest is added to the <prefix>JailQuest so that the misc task to retrieve your gear on a breakout works. If the quest doesn't exist yet, set them up along the lines of JailQuest and EscapeJailQuest from Vanilla

Setting up **carry packages**

- Set up two furniture refs in the world (without load doors in between(!)) which are owned by the actor
- Add linked references to the actor reference
- Add a BSKDefaultCarry package to the actor
- Add CarryActorScript to the actor and fill the properties. Count is usually 1
- Crate setups need to be a bit different
 - In the start area place pick up variants of the crates
 - In the end area place an equivalent amount of put down crates
 - Link the actor to a crate or some marker in the start and end area
 - The actor needs the BSKCarryCrateActorScript instead of CarryActorScript

	LinkCarryStart	LinkCarryEnd
Wood	WoodPile	WoodPile
Bucket	BucketCarryFillMarker	BucketCarryPourMarker

Basket		
Drink Tray		
Crate	Crate (Pick Up)	Crate (Put Down)

Dialogue Skill Checks

How to set up various skill checks dialog options.

For all skill checks add this property:

BSKFavorDialogueScript Property pFDS Auto
! AU students use the vanilla FavorDialogueScript instead !

Includes: script fragment, property to copy paste into the script fragment file, and dialog conditions

Persuade

`pFDS.Persuade(akSpeaker)`

Condition: player GetActorValue Speechcraft >= Speech<Level> (Global variable)

OR

Condition: player has equipped TGAmetOfArticulationList

Intimidate

`pFDS.Intimidate(akSpeaker)`

Condition: GetIntimidateSuccess = 1

The **success** is determined by a [formula](#), and is mostly affected by

- Player: Level & Speech
- Target: Level & Speech & Confidence

Make sure the npc doesn't have the confidence set to "foolhardy", otherwise it is impossible to intimidate them.

Bribe

`pFDS.Bribe(akSpeaker)`

Condition: GetBribeSuccess = 1

Condition: Player GetItemCount Gold001 >= BribeCost (Global variable)

[Formula](#)

This will cost the player a leveled amount of gold, the exact value will be seen in the place of <BribeCost> in the player dialogue.

Custom bribes can be set up using a different function of it by passing the amount as the second parameter.

```
pFDS.BribeCustom(akSpeaker, 1000)
```

Illusion (Beyond Skyrim only)

```
pFDS.Charm(akSpeaker)
```

Condition: player actor value illusion >= BSKSpeechIllusion (Global variable)

Brawl

```
pFDS.Brawl(akSpeaker)
```

Victory or defeat dialogue goes into <Prefix>DGIntimidateQuest.

The victory lines can be added in the player dialogue tab, the defeat lines are in an action of the brawl scene which is also part of the same quest.

Default values for level check globals:

VeryEasy = 10

Easy = 25

Average = 50

Hard = 75

VeryHard = 100

Frequently asked questions

Why don't NPCs open their doors/why do they think I'm trespassing?

The interior needs to have its interior load door(s) set as owned by the owner NPCs faction. An actor needs the "unlock on arrival" flag set in their package template when they are inside the building to allow the player to enter "legally".

If an NPC's package changes to one that doesn't unlock the cell, then the player will be trespassing.

Why won't my quest work??

Most likely the alias(es) fail(s) to fill. Test by setting all/certain aliases as optional.

Remember that script properties can't be changed in game after they have loaded. A new game/old save without the script active is required after changing properties.

Why does my dialog not work??

Many possible causes:

1. conditions on quest

2. conditions on dialog
3. non-optional alias failed to fill
4. wrong voicetype
5. wrong factions
6. quest not marked as start game enabled/not started at all
7. no SEQ file (need to quicksave and quickload if SEQ is out of date to get dialog working without it)

My weapon display racks are broken!!

Calling Enable/Disable on weapon racks and their activators breaks their functionality, don't do it. Often seen in player houses.

If that's not the case, blame Bethesda and try re-linking the weapon to the display.

My player house bookshelf crashes the editor!!!

You are trying to link too many things at once, CK fixes should have a workaround for this.