Mod Events (Vanilla; requires default vanilla parameter setup)

ReporterLOSGain

This mod event is sent whenever an LPO Reporter—any proximity NPC tagged for LOS—sees the player. LPO catches OnGainLOS for SingleUpdate, but this event may nonetheless fire in quick succession.

```
RegisterForModEvent("BM-LPO_ReporterLOSGain",
"LPO_OnReporterLOSGain")
Event LPO_OnReporterLOSGain(string eventName, string strArg,
float numArg, Form sender)
```

ViolationCheck

This mod event is sent when LPO conducts a violation check.

```
RegisterForModEvent("BM-LPO_ViolationCheck",
"LPO_OnViolationCheck")
Event LPO_OnViolationCheck(string eventName, string strArg,
float numArg, Form sender)
```

ViolationFound

This mod event is sent when LPO finds a license violation and attempts to start the bounty quest. This event only fires after detected violations are collected for processing, not immediately when any one violation is detected.

```
RegisterForModEvent("BM-LPO_ViolationFound",
"LPO_OnViolationFound")
Event LPO_OnViolationFound(string eventName, string strArg,
float numArg, Form sender)
```

BountyStart

This mod event is sent when LPO's bounty quest successfully starts with LPO Enforcers filling its aliases.

```
RegisterForModEvent("BM-LPO_BountyStart", "LPO_OnBountyStart")
```

```
Event LPO_OnBountyStart(string eventName, string strArg, float
numArg, Form sender)
```

BountyEnd

This mod event is sent under two scenarios: when the quest loses all valid aliases, and after LPO processes its post-confrontation sequence. This event always fires immediately before a quest Stop() call and only ever after LPO_ConfrontationEnd if Enforcer engaged dialogue with the player.

```
RegisterForModEvent("BM-LPO_BountyEnd", "LPO_OnBountyEnd")
Event LPO_OnBountyEnd(string eventName, string strArg, float
numArg, Form sender)
```

Mod Events (Custom)

BM-LPO_ConfrontationEnd

This mod event is sent when LPO finishes its enforcer-player confrontation scene. End types: walkaway = -1, arrest = 0, normal = 1, lenient = 2

```
RegisterForModEvent("BM-LPO_ConfrontationEnd",
"LPO_OnConfrontationEnd")
Event LPO_OnConfrontationEnd(int aiArg1)
```

BM-LPO_LicenseAdded

This mod event is sent when the player is given a license that passes an inventory event filter for OnItemAdded.

```
RegisterForModEvent("BM-LPO_LicenseAdded", "LPO_OnLicenseAdded")
Event LPO_ OnLicenseAdded(int aiArg1)
```

BM-LPO_LicenseRemoved

This mod event is sent when the player loses a license that passes an inventory event filter for OnItemRemoved.

```
RegisterForModEvent("BM-LPO_ LicenseRemoved",
"LPO_OnLicenseRemoved")
Event LPO_ OnLicenseRemoved(int aiArg1)
```

BM-LPO_LicensePurchased

This mod event is sent when the player purchases a license and adds the corresponding book item to the moderator's inventory event filter.

```
RegisterForModEvent("BM-LPO_ LicenseRemoved",
"LPO_OnLicenseRemoved")
Event LPO_ OnLicenseRemoved(int aiArg1)
```

BM-LPO_LicenseExpired

This mod event is sent when a license expires and removes its corresponding book item from the moderator's inventory event filter.

```
RegisterForModEvent("BM-LPO_ LicenseExpired", "LPO_On
LicenseExpired")
Event LPO_ On LicenseExpired(int aiArg1)
```

Functions

GetModVersion - String

Returns a string in a semantic (x.x.x) format.

```
Function GetModVersion()
```

GetLicenseID - Int

Returns an integer corresponding to a given license prefix.

```
Function GetLicenseID(string LicensePrefix)
```

GetLicenseTimeLeft - Float

Returns a remaining time per in-game hours.

```
Function GetLicenseTimeLeft(int LicenseType)
```

FlagViolation - Bool

Marks a corresponding violation boolean as TRUE, and then pushes to assess active violations.

```
Function FlagViolation(int ViolationType, bool Push = true, bool
CheckSafety = true)
```

ClearViolations - Bool

Removes any active and valid violations, optionally removing persistent violations as well.

```
Function ClearViolations(bool ClearPersistent = false, bool
CheckSafety = true)
```

PurchaseLicense - Bool

Activates a corresponding license cycle.

```
Function PurchaseLicense(int LicenseType, bool SubtractGold =
true, bool CheckSafety = true)
```

ExpireLicense - Bool

Terminates a corresponding license cycle.

```
Function ExpireLicense(int LicenseType, bool Push = true)
```

RemoveLicense - Bool

Removes a corresponding license Book Item from the player's inventory, optionally moving it to another container.

```
Function RemoveLicense(int LicenseType, int LicenseCount = 0,
ObjectReference DestinationContainer = None, bool CheckSafety =
true)
```

License/Violation Type Integer Map

Integer Map	License Prefix	License Suffix	Violation Prefix
1	Armor	License	Armor
2	Bikini	License	Armor
3	Clothing	License	Armor
4	Magic	License	Magic
5	Weapon	License	Weapon
6	Crafting	License	Crafting
7	Travel	Permit	Travel
8	Collar	Exemption	Collar
9	Insurance	N/A*	Uninsured
10	Curfew	Exemption	Curfew
11	Trading	License	Trading
12	Whore	License	Whore

*The Life Insurance feature uses "Life" as an aesthetic prefix and the `IsInsured` boolean as an intermediary value between related License and Violation script properties.

Definitions

Reporter

- Any actor tagged by the quest BM_Licenses_Detection.

Enforcer

- Any actor tagged by the quest BM_Licenses_Bounty. These actors are typically classified as "Guard" by vanilla faction and keyword conditions.

License

- A basic type of permissive document that emphasizes the primacy of a *use* of a privilege.

Permit

- A type of license that emphasizes *ephemerality* over *use*. Typically includes additional functionality to support the temporary nature of the related license feature.

Exemption

- A type of license that emphasizes *exclusion* over *use*.