

---

STRUCTUR

# Anti-Griefing Protocol Standard

Version 1.0

---

Published by Dickie Tower / Structur

[marketplace.secondlife.com/stores/11648](https://marketplace.secondlife.com/stores/11648)

## PREAMBLE

Second Life is a shared environment. Every scripted object deployed in a shared space is a guest in someone else's world. The Structur Anti-Griefing Protocol Standard exists to define what it means to be a responsible guest -- to build things that do what they say, behave predictably, respect the spaces they inhabit, and cannot be weaponized against the people around them.

Meeting this standard is not difficult for a creator who already builds with care. It is a codification of good practice, not an imposition of new constraints.

---

## CATEGORY 1 -- ACCESS CONTROL

### 1.1

All owner-level commands and configuration controls must respond exclusively to the object owner or verified same-group members. No access control mechanism may respond to arbitrary third party chat or touch input for privileged functions.

### 1.2

Public interaction -- where intentionally permitted -- must be explicitly designed and documented. Default behavior must always be restrictive rather than permissive. Public interaction must never include the ability to modify object behavior, configuration or state.

### 1.3

Objects that accept chat commands must use non-default channels. Channel 0 open chat commands are prohibited for any privileged function.

### 1.4

Listen handlers must always verify the identity of the sender before acting on any command regardless of the channel used.

---

## CATEGORY 2 -- SOUND AND MEDIA

### 2.1

Objects that play sounds autonomously must implement a minimum cooldown period between activations to prevent sound spam.

### 2.2

Sound volume must be set appropriately for the product's intended purpose. Maximum volume must not be used as a default where lower volumes would serve equally well.

### 2.3

Objects must not be designed to play sounds continuously on channel 0 in a manner that disrupts normal conversation in shared spaces.

### 2.4

Objects must not use media or sound functions in a manner intended to harass specific avatars.

### 2.5

Looping sounds must provide an owner-accessible mechanism to stop playback.

---

## CATEGORY 3 -- OBJECT REZZING

### 3.1

Objects that rez child objects must verify parcel permissions before attempting to rez. Failure to obtain rez permissions must be handled gracefully without error spam.

### 3.2

All rezzed child objects must carry either the TEMP\_ON\_REZ flag or an explicit self-destruct timer, or both. Orphaned objects that persist indefinitely on third party land are prohibited.

### 3.3

Rezzed child objects must not themselves rez further objects. Cascading rez chains are prohibited.

### **3.4**

Rezzed physical objects must be designed with appropriate mass and velocity limits to prevent griefing by impact.

### **3.5**

The number of simultaneously rezzed child objects must be proportionate to the product's stated purpose. Excessive simultaneous rezzing that could constitute a lag attack is prohibited.

---

## **CATEGORY 4 -- PHYSICS AND MOVEMENT**

### **4.1**

Objects must not use `lIPushObject` or equivalent functions in a manner designed to move, displace or harass avatars without explicit consensual interaction design.

### **4.2**

Physical objects must not be designed to persistently follow, track or interfere with avatars who have not consented to interaction.

### **4.3**

Objects that become physical must do so only for their intended purpose and must return to non-physical state when that purpose is complete.

---

## **CATEGORY 5 -- LAND AND PARCEL RESPECT**

### **5.1**

Objects must check and respect parcel flags before performing operations that affect the parcel, including object rezzing, media setting and environment changes.

### **5.2**

Objects must not be designed to function as land impact griefing tools -- deliberately consuming land impact on parcels the owner does not control.

### **5.3**

Objects must include a reasonable mechanism for the owner to cleanly remove all components from a parcel, leaving no persistent residue.

## 5.4

Objects must not attempt to modify parcel settings, environment settings or access lists without explicit owner authorization.

---

## CATEGORY 6 -- LISTENER HYGIENE

### 6.1

All IListen calls must be paired with corresponding IListenRemove calls when the listener is no longer needed.

### 6.2

Listen channels must be non-trivial. Where possible, randomly generated per-session channels should be used to prevent channel sniffing and command injection by third parties.

### 6.3

Listeners must always verify the identity of the sender before acting on any command.

### 6.4

Listeners must not persist indefinitely without a timeout mechanism. Open listeners that are no longer needed represent unnecessary script load and a potential security surface.

---

## CATEGORY 7 -- PERFORMANCE AND LAG

### 7.1

Scripts must not use ISleep in event handlers except where no timer-based alternative exists and the sleep duration is genuinely trivial.

### 7.2

Scripts must not use busy-wait loops or rapid timer events for functions that can be achieved through event-driven architecture.

### 7.3

Scripts must be optimized to minimize memory footprint and unnecessary computation in frequently called functions.

## 7.4

Objects must not use sensor sweeps at intervals shorter than necessary for their intended function.

## 7.5

Scripts must not make unnecessary repeated HTTP requests or external calls that could constitute abuse of external services.

---

# CATEGORY 8 -- TRANSPARENCY

## 8.1

Every product built to this standard must include a notecard or equivalent documentation clearly describing its behavior, permissions and any owner commands.

## 8.2

Products must clearly identify the creator and provide a support contact mechanism.

## 8.3

Products that interact with avatars in any way must make that interaction apparent to the owner through documentation even if the interaction itself is invisible to third parties.

## 8.4

Permissions must be set consistently and intentionally. The relationship between object permissions, script permissions and asset permissions must be coherent and documented.

---

# APPLICABILITY

Not every category applies to every product. A simple static decorative object with no scripts does not need to address object rezzing or listener hygiene. The standard is applied to applicable categories only during certification review. Where a category is not applicable to a specific product this will be noted in the certification record.

---

# VERSIONING

This standard is version 1.0. It may be updated as Second Life's scripting capabilities evolve or as new grieving vectors emerge. Version numbers will increment with any substantive change. Products

certified under a specific version are grandfathered under that version unless re-submission is requested.

---

## CERTIFICATION

Products may be submitted for Structur AGP Certification review. Certified products are listed in the public Structur AGP Certified Product Registry and receive the AGP Certified display prim for inclusion in their package. Certification confirms that a product has been reviewed against this standard and found to meet all applicable requirements at the time of review.

For certification application instructions contact Dickie Tower inworld or visit the Structur Marketplace store.

---

## PUBLIC REGISTRY

The Structur AGP Certified Product Registry is publicly available at:

[https://docs.google.com/spreadsheets/d/1ICUYcJfwoFs0pEonY9FO3jE\\_caHWQo9jp-QjWOfVh\\_M/pubhtml](https://docs.google.com/spreadsheets/d/1ICUYcJfwoFs0pEonY9FO3jE_caHWQo9jp-QjWOfVh_M/pubhtml)

The registry lists all currently certified products, their creators, certification dates and current certification status. Certification status is verified in real time by the AGP Certification Prim included with every certified product.

---

Dickie Tower / Structur

[marketplace.secondlife.com/stores/11648](https://marketplace.secondlife.com/stores/11648)

Structur -- Spaces. Stages. Systems.

Because responsible design is good design.