

# DNN Liquid Content in Second Life

## Project: DNN Subscribo

### Background

Several years ago, [Second Life](#)<sup>®</sup> residents were limited to 25 groups that they can join. Joining groups were necessary for many things, including getting freebies from a vendor, getting access to a parcel or a region, getting permissions to terraform land or modify objects, getting notices about events or new products or store discounts, and many others. Eventually, Second Life increased the allowed number of groups to 42, which is still not enough for some residents.



Figure 1 - Freebie merchandise like these are usually given away to members of a store group to get customers to come back or to introduce newbies to the brand. Location: <http://maps.secondlife.com/secondlife/Maitreya%20Isle/122/190/23>

To capture users who don't want to use up their group slots, a few scripters have created "subscribos", which are objects that residents touch to be added to a mailing list, which is stored in some database off-world (i.e., Google Sheets).

Similar techniques have been used for scripted vendors, which are inworld objects that sell merchandise. Vendors can be created without scripts; however, scripts can track what customers have bought inworld, so that they could be automatically sent updates to the product.

This also allows merchants to review their sales without having to do much bookkeeping. Inworld transactions in Linden dollars (the inworld currency) are kept on the Second Life database for only about a month. Then they are deleted forever. So, scripted vendors eliminate the worry that the record of a sales transaction is lost, just because the merchant was stranded for a few months on a remote tropical island with no internet access.

The screenshot shows the Second Life Marketplace interface. At the top, there's a navigation bar with 'Items', 'Merchants/Stores', and a search bar. Below this, a search filter section shows 'Search in' set to 'All categories', 'Show maturity levels' set to 'General, Moderate', and a 'Keywords' search bar. The main content area features a large image of the 'subscribe-o-matic' product, which is a digital display showing various subscription options. To the right of the image, the price is listed as 'L\$0', and there are 'Add To Cart' and 'Buy Now' buttons. Below the image, a list of features is provided, including 'let your customers subscribe with one simple touch', 'easily send messages and items to thousands of users', 'mass distribute your magazine', 'tell your customers about new products', and 'without requiring them to join a group!'. A 'try it free!' badge is also present. The seller's name, 'MSo Lambert', is displayed, along with a 'Visit The Store' link. A 'Use It Now' section explains that the item will be delivered directly to the user or a friend in Second Life, unpacked and ready to use. The product has 953 reviews, indicated by a star rating. At the bottom, there are tabs for 'Details', 'Features', 'Contents', and 'Reviews (953)'. A notice states: 'NEW: Subscribe-O-Matic is now completely free for groups under 500 members - no monthly or set up costs!'. A link to the website 'http://www.subscribeomatic.com!' is provided. A 'What is Subscribe-O-Matic and how can it help me?' link is also visible.

SECOND LIFE MARKETPLACE

Items Merchants/Stores Search this store

Search in All categories Show maturity levels General, Moderate Keywords Search

General maturity level: Gadgets » Information Management

Subscribe-O-Matic - now free! Version 1.7.4

subscribe-o-matic

Are you still asking your customers to join your update group?

- › let your customers subscribe with one simple touch
- › easily send messages and items to thousands of users
- › mass distribute your magazine,
- › tell your customers about new products
- › without requiring them to join a group!

try it free!

synthetik www.synthetik-on.net

Zoom/View images (2)

L\$0

Add To Cart

Buy Now

MSo Lambert Visit The Store

Use It Now

This item will be delivered directly to you or a friend in Second Life, unpacked and ready to use. No land or sandbox required.

★★★★★ Reviews (953)

Permissions: ✓Copy ✓Modify Transfer

Prim count: 0

Add To Favorites

Like 1

Share this item | |

Flag this item

Details Features Contents Reviews (953)

NEW: Subscribe-O-Matic is now completely free for groups under 500 members - no monthly or set up costs!

Please check out our website for detailed information, subscription plans and current news - <http://www.subscribeomatic.com!>

What is Subscribe-O-Matic and how can it help me?

Figure 2 - One of several subscribo brands in the SL Marketplace.

## DNN Subscribo

The **DNN Subscribo** is a basic subscribo, that uses **Liquid Content™** to store customer information as content items. This product can be expanded into an inworld vendor that is also a sales and customer tracking system.

### Components

On the DNN side:

- A content type to track subscribers:
  - The SL username can be used as the content item name. This will also be used to search for a specific subscriber (content item), if the content item ID is not known.
  - A single-line field to hold the SL UUID of the subscriber. This UUID is required by many Second Life APIs for private communications, product delivery, etc.
  - A multiple-choice field for mailing lists. Allow users to choose more than one, or set to **Checkboxes**.

The screenshot shows the configuration interface for a content type named "SL User". The interface is divided into two main sections: a left sidebar for settings and a main content area for the form.

**Left Sidebar (Settings):**

- Multiple Choice:** A checkbox is checked, indicating the form type is multiple choice.
- Form Type:** A dropdown menu is open, showing three options: "Dropdown", "Checkboxes" (which is selected), and "Radio Buttons".
- Choices:** A list of choices is displayed, each with a trash icon for deletion:
  - inactive
  - customer
  - creator
  - gifts
  - announcements
- Add Choice:** A button to add a new choice.
- List Orientation:** Two buttons, "Horizontal" and "Vertical", with "Vertical" selected.
- Other As An Option:** A toggle switch is turned "On".
- GENERAL SETTINGS:** A section header at the bottom of the sidebar.

**Main Content Area:**

- Content Properties:** A tabbed interface with "Content" and "Properties" tabs. The "Content" tab is active.
- SL UUID:** A single-line text input field.
- Subscriptions:** A section with a list of checkboxes for mailing lists:
  - ☒ inactive
  - ☐ customer
  - ☐ creator
  - ☐ gifts
  - ☐ announcements
  - ☐ Other

Figure 3 - Content Type for the subscriber information.

On the Second Life side:

- A textured cube, that is configured so that:
  - Face #0 serves as the SUBSCRIBE button,
  - Face #1 serves as the UNSUBSCRIBE button, and
  - Face #3 displays the company logo.



Figure 4 - Settings for our cube as described.

- A [Linden Scripting Language](#) (LSL) script. LSL is the programming language used in Second Life and other OpenSim worlds. (OpenSim is a technology based on SL.) The LSL script has dependencies on the way the object is configured. You can create the same interactive object many ways:
  - You can use multiple prims that are combined into one object, and **a different action is performed depending on which prim is touched**. Con: Parcels and regions have limitations on land impact. A basic prim equals one land impact (LI). The LI of “mesh” objects vary, depending on the number of tris on the mesh. So users prefer objects that use as few prims as possible. (Objects attached to avatar do not have LI limitations, although complex attachments can cause lag.)
  - You can also use just one face of the cube, and **a different action is performed depending on what part of the face is touched**. Con: This requires a more complex script that calculates the point that was touched and then matching it with some data structure that defines which points should do which action.
  - Our script (DNNSubscribeForSL.txt) assigns **different faces of a single prim (a cube) to do different actions**, which uses only one LI and less complex code.

## APIs Used

The core functionality of the LSL script uses the following APIs:

- In Liquid Content:
  - GET /api/ContentItem?name=...
  - GET /api/ContentItem/{id}
  - PUT /api/ContentItem/{id}
  - POST /api/ContentItem
- In LSL:
  - [llHTTPRequest\(\) function](#)
  - [http response event](#)
  - Various [JSON-related APIs](#) to parse responses and modify them.

## LSL Script



DNNSubscriboForS  
L.txt

The following are the DNN-related variables used by the script:

```
string URLDNNSOFTWARE = "http://www.dnnsoftware.com/";
string TGTDOMAIN = "https://dnnapi.com/content/api";
string DNNCONTENTTYPEID = "00000000-0000-0000-0000-000000000000";
string DNNCONTENTTYPENAME = "SL User";

list BASIC_HTTP_PARAMETERS = [
    HTTP_MIMETYPE, "application/json",
    HTTP_BODY_MAXLENGTH, 16384, // max length for Mono
    HTTP_PRAGMA_NO_CACHE, TRUE,
    HTTP_CUSTOM_HEADER, "Authorization", "Bearer 00000000000000000000000000000000"
];
```

NOTE: You must replace the `DNNCONTENTTYPEID` with the UUID of the content type you create in your DNN site. You must also replace the "Bearer" UUID with [your own API key](#).

The following are the HTTP Requests:

### GET using a query

```
t_kHttpReqGet = llHTTPRequest(
    TGTDOMAIN + "/ContentItems?name=" + (string) t_sUsername,
    BASIC_HTTP_PARAMETERS + [ HTTP_METHOD, "GET" ],
```

```
"" );
```

### GET using the content item UUID of the specific "SL User"

```
t_kHttpReqGet = llHTTPRequest(  
    TGTDOMAIN + "/ContentItems/" + t_kDNNID,  
    BASIC_HTTP_PARAMETERS + [ HTTP_METHOD, "GET" ],  
    "" );
```

### POST

```
t_kHttpReqPost = llHTTPRequest(  
    TGTDOMAIN + "/ContentItems?publish=true",  
    BASIC_HTTP_PARAMETERS + [ HTTP_METHOD, "POST" ],  
    sPostStr );
```

### PUT

```
t_kHttpReqPut = llHTTPRequest(  
    TGTDOMAIN + "/ContentItems/" + t_kDNNID + "?publish=true",  
    BASIC_HTTP_PARAMETERS + [ HTTP_METHOD, "PUT" ],  
    sPostStr );
```

To parse the JSON structure returned in the HTTP Response, insert the parsing code (using `llJsonGetValue()`) in the `http_response` event handler.

## Flowchart of the LSL Script

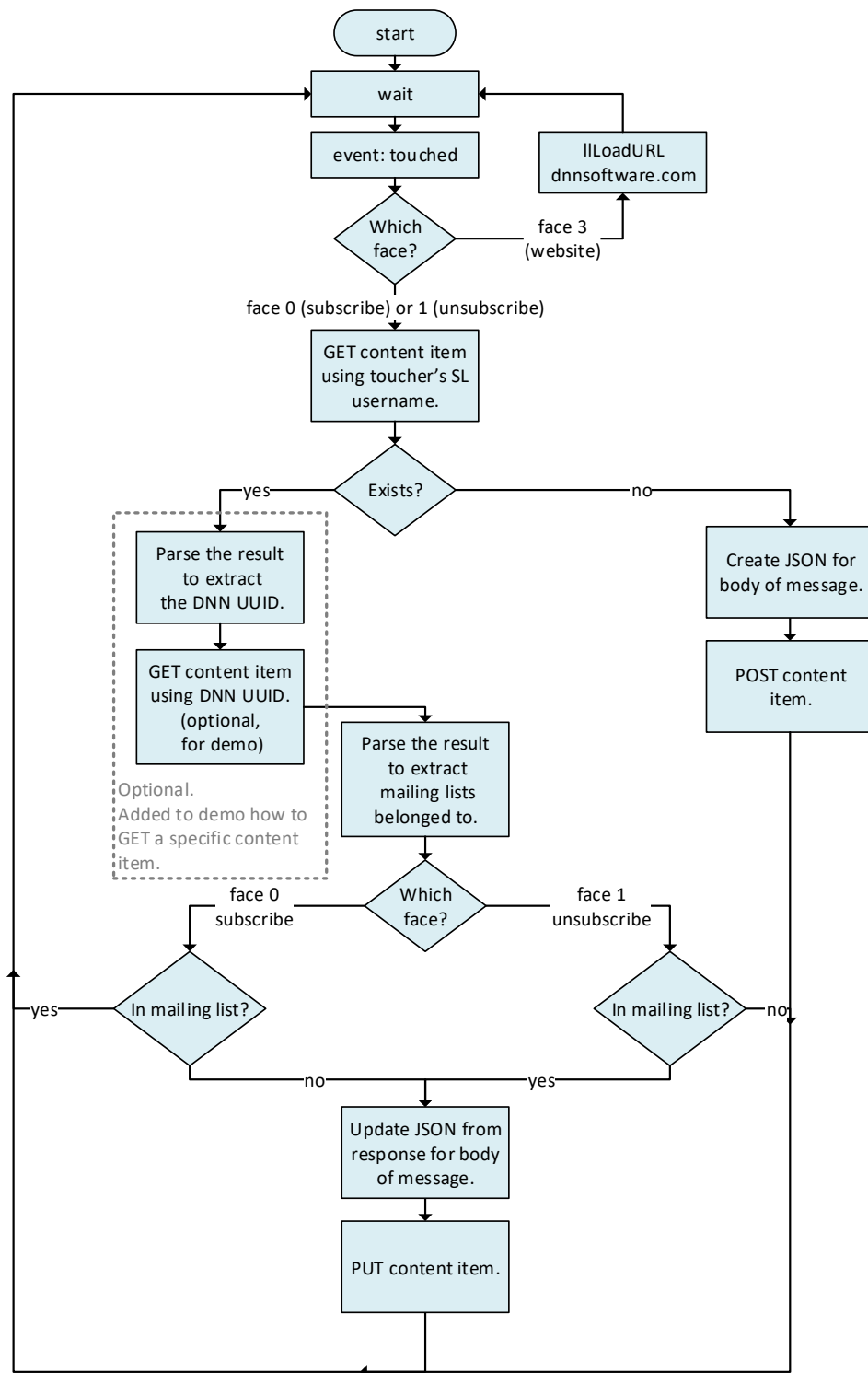


Figure 5 - Flowchart of the LSL script.

## Prerequisites

- An Evoq site.
- Superuser/Host access to your site.
- A Second Life account.
- Some knowledge of how to create and manipulate prims.
- Some knowledge of how to script in the Linden Scripting Language.
- 10L\$ (10 Linden dollars = approximately US\$0.04) to upload a texture.

## Steps

### 1) In Evoq

#### a) [Create a content type](#). See *Figure 3*.

- i) Set the content type name to "SL User".
- ii) Create the following fields:

Single-line	Name: "SL UUID"
Multiple Choice	Name: "Subscriptions" Appearance: Checkboxes Choices: "inactive", "customer", "creator", "announcements", "gifts", etc. (The script uses "announcements".)

#### b) [Create an API key](#).

### 2) In a graphics program

Create a square texture as follows:

- i) The bottom 1/8 as a red stripe for unsubscribing.
- ii) The next higher 1/8 as a green stripe for subscribing.
- iii) The remaining space to indicate what mailing list people would be subscribing to.

Sample texture:

(You can request a full-perms copy of this texture inside Second Life from **DNNDiaNNa**.)





*Figure 6 - Texture applied on the prim for the subscribo.*

### 3) Inside Second Life

- a) Upload the texture.
- b) Create the object.
  - i) Rez a cube.
  - ii) Torture the cube.
    - (1) Size: <0.5, 0.60396, 0.01>
    - (2) Taper: x = 0.0, y = 0.8
    - (3) Top Shear: x = 0.0, y = -0.2
  - iii) Apply the texture. Adjust the texture scale and offset for each of the active faces to show only the relevant part of your texture.
- c) Create a new script.
  - i) Create a new script in the Contents tab of the object or in your inventory.
  - ii) Copy and paste the LSL code to the script you created (replacing the default code) and save.
  - iii) In the script, replace:
    - (1) The Content Type ID (DNNCONTENTTYPEID variable in Line 17/16).
    - (2) The API key (HTTP\_CUSTOM\_HEADER number at the end of Line 24/23).

(Note that the LSL editor inworld starts with Line 0 as the first line.)  
Both of these numbers are linked to your Evoq site.  
iv) You can make additional changes as you wish.

d) Set the permissions of the script and the object itself.

## Related links

- DNN Documentation Center: [Creating Apps That Use Microservices](#)
- [SecondLife.com](#)
- Second Life Wiki: [LSL Portal](#)



*Figure 7 - DiaNNa (username:dhndianna) inside Second Life with the completed subscribo.*