

Whop Ads Tracking

Track conversions from your funnel pages to the Whop Pixel using the HandL JS library.

- [Base Installation](#)
- [Event Sending](#)
- [Advanced Matching](#)
- [Custom Events / Listeners](#)

Base Installation

Prerequisites

- HandL JS library must be loaded on your page
- You must have a Whop Account ID from your Whop Dashboard

Step 1: Get Your Whop Account ID

1. Log in to your [Whop Dashboard](#)
2. Look at the dashboard URL
3. Copy the portion after `dashboard/` that begins with `biz_` (format: `biz_XXXXXXXXXXXX`)

Step 2: Initialize Whop Pixel

Add the following script to your HTML page **before** the HandL script tag:

```
<script>
    // Set your Whop Account ID
    window.whop_account_id = 'YOUR_WHOP_ACCOUNT_ID_HERE';
</script>

<!-- Load HandL script --> Make sure to change your license key
<script>
var handl_custom_params = [];
var handl_js = document.createElement("script");
handl_js.setAttribute("src", "https://track.utmsimple.com/utm.js?license=<license code goes here>"), document.head.appendChild(handl_js),
handl_js.onload = function() {
    /*This is the UTM Simple ready event, you can put any code you'd like to trigger after the script fully loaded here*/
    console.log("I'm ready")
    console.log("prnting all the cookies")
    console.log(HandL.getAll())
};
</script>
```

Example:

```
<script>
  window.whop_account_id = 'biz_XXXXXXXXXXXX';
</script>
<script src="https://cdn.example.com/utm.js"></script>
```

Multiple Accounts

To track the same funnel across several Whop accounts, pass an array of Account IDs:

```
<script>
  window.whop_account_id = ['biz_XXXXXXXXXXXX', 'biz_yyyyyyyyyyyyyy'];
</script>
```

Note: Do not also paste Whop's own pixel snippet on the page. HandL loads it for you, and adding both will double-count your events.

Step 3: Verify Installation

After the page loads, the Whop Pixel will automatically:

- Load the Whop Pixel script
- Set your account scope
- Fire a `page` event

You can verify the installation by:

1. Opening browser DevTools (F12)
2. Going to the Network tab
3. Filtering for `t.whop.tw`
4. You should see requests to Whop's tracking endpoint

Event Sending

Basic Event Sending

Use the `HandL.sendWhopEvent()` function to send events to Whop:

```
HandL.sendWhopEvent(eventName, eventData);
```

Parameters:

- `eventName` (string, required): The name of the event
- `eventData` (object, optional): Event properties and customer information

Note: Do not track Whop's native checkout, subscriptions or trials. Whop records those itself. Use `sendWhopEvent` only for funnel steps on your own pages.

Standard Events

Whop supports the following standard events:

lead

Track contact info or opt-in form submissions:

```
HandL.sendWhopEvent('lead', {  
  event_id: 'lead_8f21c0'  
});
```

schedule

Track call or appointment bookings:

```
HandL.sendWhopEvent('schedule', {  
  value: 50,  
  currency: 'USD'  
});
```

submit_application

Track application submissions:

```
HandL.sendWhopEvent('submit_application');
```

contact

Track conversation or contact requests:

```
HandL.sendWhopEvent('contact');
```

complete_registration

Track signup or registration completion:

```
HandL.sendWhopEvent('complete_registration');
```

view_content

Track key page or content views:

```
HandL.sendWhopEvent('view_content');
```

add_to_cart

Track items added to cart:

```
HandL.sendWhopEvent('add_to_cart', {  
  value: 19.99,  
  currency: 'USD'  
});
```

purchase

Track completed sales on your own checkout only:

```
HandL.sendWhopEvent('purchase', {  
  value: 29.99,  
  currency: 'USD',  
  event_id: 'order_12345'  
});
```

Custom Events

You can also send custom events with any event name:

```
HandL.sendWhopEvent('CustomEventName', {  
  value: 10,  
  currency: 'USD'  
});
```

Event Data Parameters

Common event parameters you can include:

Parameter	Type	Description	Example
<code>value</code>	number	Monetary value of the event	<code>29.99</code>
<code>currency</code>	string	Currency code (ISO 4217), defaults to USD	<code>'USD'</code> , <code>'EUR'</code>
<code>event_id</code>	string	Unique identifier, prevents duplicate counting	<code>'order_12345'</code>

Customer information fields can also be passed here. See **Advanced Matching**.

Advanced Matching

Advanced matching allows Whop to match events to users more accurately by sending customer identifiers alongside the event.

Important: Unlike TikTok and Meta, Whop expects customer fields in **plain text**. HandL sends them unhashed by design. Do not hash emails or phone numbers before passing them.

How It Works

- Data Collection:** When users fill out forms, HandL automatically captures customer details. Data is stored in localStorage with keys such as `em` (email) and `ph` (phone).
- Automatic Merging:** Every call to `HandL.sendWhopEvent()` merges any stored customer values into the event payload. Fields you pass directly always override the stored values.
- External ID:** Uses the `first_handlID` cookie as `external_id`. This is automatically generated on first visit.
- Full Name:** When a first or last name is available, HandL composes `name` automatically.

Manual Data Collection

You can pass customer fields directly on any event:

```
HandL.sendWhopEvent('lead', {
  event_id: 'lead_8f21c0',
  email: 'visitor@example.com',
  first_name: 'Jane',
  last_name: 'Doe',
  phone: '+15551234567'
});
```

Advanced Matching Parameters

Whop supports the following identifiers (all sent as plain text):

Parameter	Description	Source
<code>email</code>	User's email address	<code>localStorage.getItem('em')</code>
<code>first_name</code>	User's first name	<code>localStorage.getItem('fn')</code>
<code>last_name</code>	User's last name	<code>localStorage.getItem('ln')</code>

Parameter	Description	Source
<code>name</code>	Full name	Composed from first and last name
<code>phone</code>	User's phone number	<code>localStorage.getItem('ph')</code>
<code>city</code>	User's city	<code>localStorage.getItem('ct')</code>
<code>state</code>	User's state	<code>localStorage.getItem('st')</code>
<code>postal_code</code>	User's postal code	<code>localStorage.getItem('zp')</code>
<code>country</code>	User's country	<code>localStorage.getItem('country')</code>
<code>external_id</code>	Unique identifier	<code>Cookies.get('first_handlID')</code>

Empty fields are dropped automatically and never sent.

Verification

To verify advanced matching is working:

1. Open browser DevTools (F12)
2. Go to the Network tab
3. Filter for `t.whop.tw`
4. Inspect the request payload and confirm your customer fields are present

Custom Events / Listeners

Listening to Whop Events

HandL fires custom events you can listen to:

```
// Listen for Whop initialization
document.body.addEventListener('UTMSimpleWhopInitCompleted', function(event) {
    console.log('Whop Pixel initialized');
});

// Listen for any Whop event completion
document.body.addEventListener('UTMSimpleWhopEventCompleted', function(event) {
    console.log('Whop event sent');
});

// Listen for specific event completion
document.body.addEventListener('UTMSimpleWhoppurchaseCompleted', function(event) {
    console.log('Purchase event sent to Whop');
});
```

Note: The event-specific listener name is `UTMSimpleWhop` + your event name + `Completed`, using the event name exactly as you passed it. Because Whop's standard events are lowercase, `sendWhopEvent('purchase')` fires `UTMSimpleWhoppurchaseCompleted`.

Example: Track Button Clicks

```
document.getElementById('book-call-button').addEventListener('click', function() {
    HandL.sendWhopEvent('schedule', {
        value: 50,
        currency: 'USD'
    });
});
```

Example: Track Form Submissions

```
<form id="contact-form" onsubmit="trackFormSubmission(event)">
    <!-- form fields -->
```

```
</form>
```

```
<script>
```

```
function trackFormSubmission(event) {  
  // Form submission logic here  
  
  HandL.sendWhopEvent('lead', {  
    event_id: 'lead_' + Date.now()  
  });  
}  
</script>
```