

# Display Implementation

## Colruyt Group — Adhese Tag Implementation

### Overview

The `colruytgrouptest` customer tag integrates the Adhese TypeScript SDK into Colruyt Group web pages. Each HTML page configures and initialises an Adhese instance via a `window.AdheseConfig` global, which is consumed by a shared TypeScript entry point (`src/tag.ts`).

### How It Works

#### 1. Page-level configuration (`window.AdheseConfig`)

Each HTML page defines a `window.AdheseConfig` object in an inline `<script>` block **before** loading the tag script. This configuration is page-specific and declares:

Property	Description
account	The Adhese account identifier <ul style="list-style-type: none"><li>test account: <code>"colruytgrouptest"</code></li><li>productie account: <code>"colruytgroup"</code></li></ul>
location	The ad location/slot context for this page (e.g. <code>"clp_web_home"</code> , <code>"clp_web_acties"</code> ) Correct information for each page can be found in <a href="#">the Inventory excel</a> .
consent	Boolean — whether user has given personalized ads consent.
data	Targeting parameters (user ID, device ID, language, categories, etc.)
slots	Array of ad slots to render on this page

Other settings can be used if the implementation requires it. More information about any specific usecase can be found on the documentation pages of the SDK:

## 1.1 Data object

The data object can handle the following inputs:

Field	Adheses param	Type	Description	Pages
userID	mi	<code>string</code>	User identifier	All
deviceID	di	<code>string</code>	Device identifier	All
language	ln	<code>string</code>	Content/UI language (e.g. <code>"en"</code> , <code>"nl"</code> )	All
categories	ct	<code>string[]</code>	List of content categories (e.g. <code>["food", "beverages"]</code> )	Product
currentCategory	cc	<code>string</code>	The active/current category	Product
searchKeyword	kw	<code>string</code>	The keyword used in the sites search.	Product Search
appVersion	ve	<code>string</code>	App version string (e.g. <code>"1.0.0"</code> )	All

Each slot entry specifies:

- `format` — the ad format name (e.g. `"herobanner"`, `"actiebanner"`, `"contenttile"`)
  - Correct information for each slot can be found in [the Inventory excel](#).
- `containingElement` — the `id` of the DOM element that will hold the ad
- `slot` (*optional*) — a positional suffix (e.g. `"_top_1"`, `"_tile_2"`)
  - Correct information for each slot can be found in [the Inventory excel](#).
- `height` / `width` — dimensions passed to the SDK
  - If no fixed dimensions are known, set them to 100% to make them responsive and let the dimensions be dictated by the front end.

More information about any specific usecase can be found on the documentation pages of the SDK: [https://adheses.github.io/sdk\\_typescript/slots.html](https://adheses.github.io/sdk_typescript/slots.html)

## 2. Tag entry point (`tag.js`)

The single `tag.js` file is shared across all pages for this customer. It:

1. **Reads** window.AdheseConfig
2. **Maps** the data fields to Adhese parameter prefixes.
3. **Initialises** the Adhese SDK, passing the mapped parameters and the slot definitions in ad requests.
4. **Exposes** the resulting adhese instance on window.adhese for external access
5. Handles all **ad response**, rendering ads when needed.
6. Handles all **tracking** of the ads.

The use the following URL for the script:

- Test : <https://pool-colruytgrouptest.adhese.com/tag/tag.js>
- poroduction: <https://pool-colruytgroup.adhese.com/tag/tag.js>

### 3. Ad unit containers (HTML)

Each page contains `<div>` elements with matching `id` attributes that correspond to the containingElement values in the slot config. The SDK injects the rendered ads into these containers.

---

Revision #2

Created 20 April 2026 11:23:36 by Kevin Voortmans

Updated 20 April 2026 14:53:56 by Kevin Voortmans