

# Never Null

# Table of Contents

## Never Null

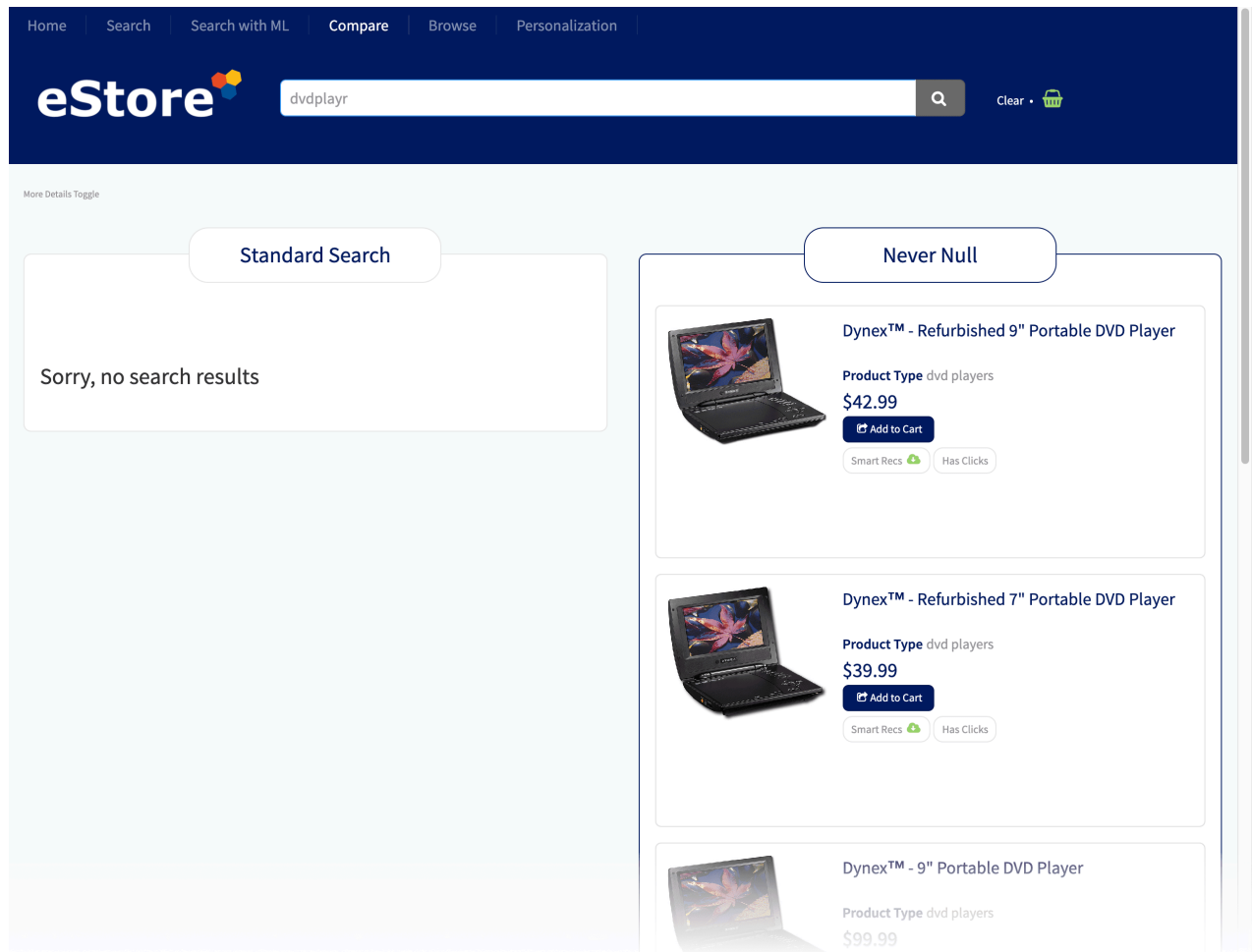
Getting Started .....	4
Product Data Requirements .....	5
Signals Data Requirements .....	6
Send Your Initial Data Stream ..	9
Fetch Results .....	10
Update Your Data .....	13
Get Support from Lucidworks .....	14
Semantic Vector Search .....	15

# Never Null

## Concept

Never Null is a cloud-based service that complements your existing search platform to provide relevant results for queries that would otherwise produce zero results or under-performing results. It uses deep learning and semantic vector search to produce relevant search results that you can use to replace your "0 results found" page or augment the results from your own search platform. With Never Null, your customers can find relevant products even when they misspell their query terms or use terms that have no exact matches in your product catalog.

For example, imagine that a customer enters "dvdplayr" instead of "dvd player". A keyword search may return no results, but Never Null can still provide relevant results:



You can transparently replace the null results with Never Null results, or present the Never Null results as alternative "Did you mean..." results.

Never Null provides the most relevant results based on semantic meaning and historical user behavior, not just keywords. When you query Never Null, you are not querying your product catalog or your signals catalog; you're querying a data model that is trained to predict which results will engage your customers.

## Getting started

Your initial integration with Never Null entails formatting your data according to our data models, then sending the complete dataset for analysis.

1. *Format your product data.*
2. *Format your signals data.*
3. *Send your initial data stream.*


Using your data, Lucidworks trains a custom deep learning model, tailored to your unique site. Allow some time for the Lucidworks team and the machine learning service to perform this initial analysis, especially if your catalog is large. Lucidworks will notify you when your Never Null data is ready to query.

## **Using Never Null**

After the initial data ingest and analysis, you can:

- *Fetch relevant results* by performing a secondary query to Never Null.
- *Keep your Never Null data fresh* by sending incremental updates about your product catalog and signals.
- *Get support from Lucidworks* whenever you need it.

## **Additional resources**

- [How Zero Results Are Killing Ecommerce Conversions](#) 

## Getting Started

### Concept

Your initial integration with Never Null entails formatting your data according to our data models, then sending the complete dataset for analysis.

1. *Format your product data.*
2. *Format your signals data.*
3. *Send your initial data stream.*

Using your data, Lucidworks trains a custom deep learning model, tailored to your unique site. Allow some time for the Lucidworks team and the machine learning service to perform this initial analysis, especially if your catalog is large. Lucidworks will notify you when your Never Null data is ready to query.

After the initial analysis, you can *fetch query results*.

## Product Data Requirements

### Reference

Your product data is your main content catalog, whether it's your product catalog, your organization's knowledge base, or any other searchable body of content.

### Required fields

- Product ID
- A short product description

Additional fields containing descriptive attributes can also be included. Consult with Lucidworks for guidance.

### The product ID

Send the product ID field from your product catalog as-is. Never Null returns this value in search results for integration with your platform.

The product ID field should also be the same one tracked in your *signals*.


### Product description guidelines

The contents of your product description generally include your *searchable* fields. Usually, this means you need to combine multiple fields from your data into one aggregated description field for Never Null.

The exact fields depend on the market in which your organization operates, such as apparel, consumer electronics, and so on. For example, searchable fields for apparel may include size, color, activity, and fiber, whereas a consumer electronics catalog uses an entirely different set of fields to describe its products.

Generally, you should omit the long description field in order to reduce noise. Never Null provides the best results when the product description is short and contains only the essential terms describing the product, without extraneous verbiage. Consult with Lucidworks for guidance based on your unique product catalog.

### Supported formats

- Tab-separated values (TSV)  
See *Example TSV data* below.
- Compressed TSV
- Parquet  
See the [Apache Parquet format specifications](#) .

### Example TSV data

product_id_s	description_t
1379934	Lightweight Travel Twill cotton men's pant.
1575542	Lightweight Work A tough cotton women's pant that repels water.
1457777	Stretch T-Shirts Short Sleeve Moisture Wicking Casual v-neck with UPF 15 sun protection.

### Next steps

- If you also have signals data available, see *Signals Data Requirements*.
- Once you have planned how to format your data, you can *send your initial data stream*.
- After Never Null has performed its initial analysis of your data, you can *fetch query results*.

## Signals Data Requirements

### Reference

Your signals data is historical data describing the behavior of your users.

### Required signal/event types

At a minimum, Never Null requires these signal types:

Query response signals	A response from the search platform that includes search results, if any.
Add-to-cart signals	The user adds an item to the cart.

### Recommended signal/event types

The more signal types you can provide, the better your Never Null results. Some recommended signal types are described below. Plan on providing any that your search application supports.

Click signals	The user clicks a search result.
Add-to-favorites signals	The user adds an item to a list of favorites.
Hover/quick-view signals	The user hovers over a product, for example to display a short description.
Purchase signals	The user purchases a product.
Remove-from-cart signals	The user removed an item from the cart.
Query request signals	Events that generate a query, such as entering a query string or clicking a facet/filter. If you have response signals with the required fields, then request signals are not needed.

## Required fields

Field	Description
Timestamp	The date and time at which the event occurred, preferably in Solr UTC format, as in <code>YYYY-MM-DDThh:mm:ssZ</code> .
Signal ID	A unique identifier for this individual signal record.
Event type	The type of interaction recorded by this signal, such as click, add-to-cart, like, purchase, and so on.
Product ID	The product ID from your catalog, corresponding to the item with which the user interacted.
Query string	The user query terms associated with this event, if any.
Zero-results flag	<i>In query response signals only</i> , a flag indicating whether the user's query returned zero results, if available. This could be a value representing the number of search results, where the value "0" is the zero-results flag.
Search result ID(s)	<i>In query response signals only</i> , one or more product IDs that were returned in search results.
Session ID	A unique identifier shared by all signals from a specific user's session, regardless of event type.

## Additional guidelines

- Signals must be raw, not aggregated.
- The required number or time range of signals depends on the volume of activity on your site. One to three months of signals is generally sufficient. Consult with Lucidworks for a recommendation tailored to your site's activity level.
- Omit details that could identify a specific user.

## Supported formats

- Tab-separated values (TSV)  
See the *example TSV data* below.
- Parquet  
See the [Apache Parquet format specifications](#).



### Example TSV data

timestamp_tdt	signal_id	type	product_id	query_string	num_results	result_docs	session_id
2020-01-31 15:50:18	0c3b742c-77f2-4dc7-8f34-4809be0db103	click	518	wool			q6kowbxwp31hqhiqc5zbcke375kaltq9
	0c401791-d197-4570-a085-b5b4e90b2ced	session					0dkjtr6hz75z1eivvzt5eai7xts26w0y
2020-01-31 16:09:40	13WjqqrKcH	response		shorts mens white	0		da645378-cb14-422a-aecd-2b60f6cf83b6
2020-01-31 16:13:02	0c6ac8ea-8bd4-448d-a06a-ac73c7d0e2b7	request					60a3d62a-f5f9-493e-8d10-976f756ddcf0
2020-01-31 16:18:29	3akllcgJuT	response		red shirt	4	1,932,197,217,281,340	20a7fa35-8f4b-4519-b04e-1590f5dd7f7e
2020-01-31 16:18:33	0c76c62c-1820-498c-9b85-80d71365d461	cart-add	1992				wgvdx117osd9ic63wr0an0ssd3hlzayd
2020-01-31 16:20:38	0c826f75-4836-4812-a163-18ce93639e29	purchase	1165				gbrf1z44bewylhwr1271f3nylm25hqpt

### Next steps

- Once you have planned how to format your *product data* and your signals data, you can *send your initial data stream*.
- After Never Null has performed its initial analysis of your data, you can *fetch query results*.

## Send Your Initial Data Stream

### How-to

Once your data is formatted to conform to the requirements for *product data* and *signals data*, you need to upload it to a secure data bucket and provide access for Lucidworks.

### How to get your data into Never Null

1. Confirm that your product data is formatted according to the *product data requirements*.
2. Confirm that your signals data is formatted according to the *signals data requirements*.
3. Upload it to a secure data bucket, such as GCS, S3, or Azure Blob Storage.
4. Provide Lucidworks with access to the data bucket.

Lucidworks will inform you when the initial model training is complete and your Never Null data is ready to *query*.

Plan on periodically *updating your Never Null data* to incorporate your latest product and signals data using a similar method.

## Fetch Results

### How-to

For any primary query to your search platform, you can perform a secondary query to Never Null and get results that enhance your own. This topic shows you how to construct and send a secondary query to Never Null's REST API using the Solr query language.

To use the API, you need these items from Lucidworks:

- Your short customer name
- Your query profile name
- Your credentials

#### Note

If your content was recently ingested for the first time, Lucidworks will inform you when the initial model training is complete and your Never Null data is ready to query.

### Authentication

Before you can begin fetching Never Null results, you must log in to the service as shown below:

```
curl -i -H "content-type:application/json" -X POST -d  
'{"username":"username","password":"password"}'  
'https://name.b.lucidworks.com/api/session?realmName=native'
```

Never Null returns a cookie, as in the example below:

## Query URL format

`https://name.b.lucidworks.com/api/query/profileName?q=QUERY-STRING...`

The response is a JSON response where each item includes these fields:

- Product ID
- Short product description
- Similarity score
- Additional objects that reflect the input parameters

### Example request and response

As "bigbox", query a profile named 'bigbox\_en' for the misspelled term "dvdplayr".

## Request

```
curl -u USERNAME:PASSWORD
```

```
https://bigbox.b.lucidworks.cloud/api/query/bigbox_en?q=dvdplayr
```

## Response

```
{ "response" : { "docs" : [ { "description_s" : "Dynex™ – Refurbished  
9\" Portable DVD Player", "id" : "1559547", "score" : 0.685536 }, {  
"description_s" : "Dynex™ – Refurbished 7\" Portable DVD Player", "id"  
: "1559538", "score" : 0.659592 }, { ... } ] }
```

## Update Your Data

### How-to

Once your *initial data upload and analysis* are complete, you need to keep your Never Null up to date by periodically sending incremental updates. Like the initial data upload, you can upload incremental updates to a secure data bucket and provide access for Lucidworks.

### How to update your Never Null data


1. Confirm that your product data is formatted according to the *product data requirements*.
2. Confirm that your signals data is formatted according to the *signals data requirements*.
3. Upload it to a secure data bucket, such as GCS or S3.
4. Provide Lucidworks with access to the data bucket.

## Get Support from Lucidworks

### How-to

Lucidworks provides technical support to customers 24 hours a day, 7 days a week, with unlimited incidents.

[Submit a request for support](#)  .

For detailed information about opening a support case, see [Lucidworks Customer Center Training](#)  .

# Semantic Vector Search

## Concept

Never Null delivers semantic vector search as a standalone managed service.



SVS features are available in Fusion 5.3+.

## What is semantic vector search?

Semantic vector search (SVS) is the process of interpreting the intent and contextual meaning of queries and content. This is unique from the lexical approach to search, which is driven by keyword matching.

SVS uses deep metric learning to train a semantic model. The model learns to organize products and queries in a vector space such that similar things are placed near each other and dissimilar things are pushed far apart. At query time, SVS uses the semantic model to embed the query in the vector space and then find nearest neighbor products. This is *searching by meaning*.

### Semantic Vector Search advantages:

1. Rich semantic capabilities for better relevancy than lexical search
2. Fixes problems:
  - a. Misspelled queries, synonyms, and phrases
  - b. Underperforming and zero-result queries
3. Doesn't require signals to start
  - a. Uses general pre-trained models for cold-start solution
4. Collected signals are used to improve relevancy
  - a. Models are directly trained to rank items higher when users are more likely to interact with them

For more information, see *How to Set Up SVS*