Smart Marketing Catalyst

E-Commerce

Cart Tracking – Configuration Guide

5/20/2016

Canada • France • Russia

dialoginsight.com

Table of Contents

Introduction

The E-Commerce module of the Dialog Insight application offers the possibility to import, structure and organize data coming from your transactional site. Up to now, the E-Commerce module could only collect transactional information from one of the supported platforms, which are Google, Magento and Shopify.

But now, it is possible to use Dialog Insight's E-Commerce module to capture real-time information about transactions and carts in your transactional sites in order to associate this information to your contacts in Dialog Insight.

To take advantage of this feature, you simply need to deploy our Java scripts to your E-Commerce site, and then use our transaction and cart tracking options to send in real-time all client interactions with products or sales to Dialog Insight.

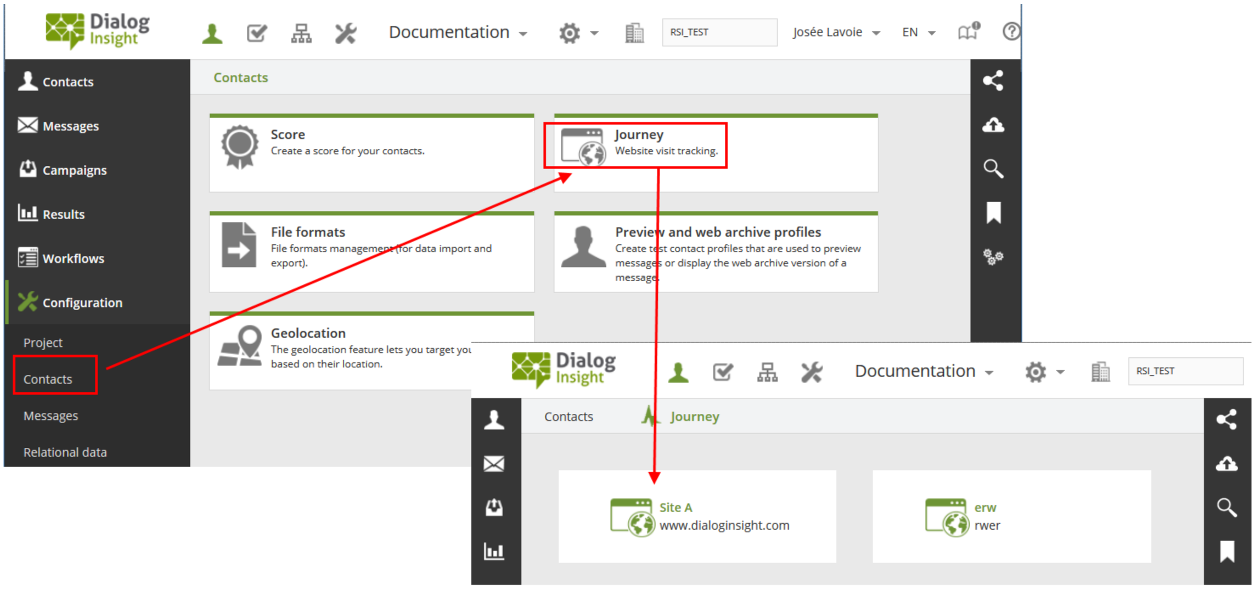
Terminology

To ensure this guide is understood as it is written, here are some of the most common terms used in this guide and their respective definition:

- ✓ shopping cart: The cart already in use in your site and edited by clients. It is used to place an order that will trigger the real purchase.
- ✓ tracking cart: The Javascript object that contains the same information as the shopping cart (products, prices, quantities, etc.) and that is sent to Dialog insight for real-time tracking of client transactions. This is the client information you will use to target contacts for future communications.
- ✓ transaction: The actual purchase that was done on your transactional site. This purchase can be associated to a shopping cart or to products without carts.

Activation

In the Journey module, after you have configured the site to track, go to the E-Commerce section in order to enable cart tracking.



Score – Create a score for your contacts.

Journey – Website visit tracking.

File formats – File formats management (for data import and export).

Preview and web archive profiles – Create test contact profiles that are used to preview messages or display the web archive version of a message.

Geolocation – The geolocation feature lets you target you based on their location.

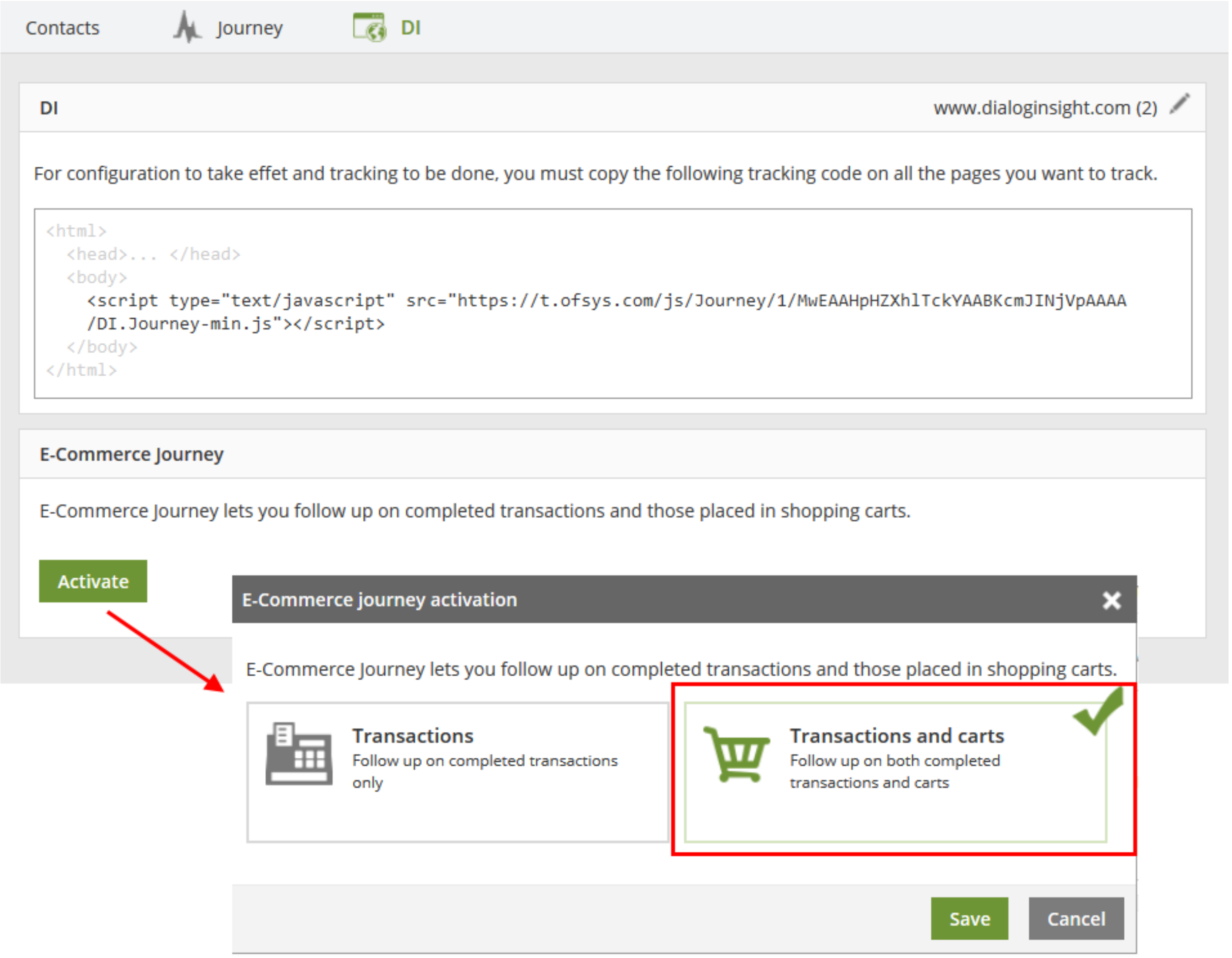
Site A – www.dialoginsight.com

erw – rwer

The first section describes the script to be used to track contacts on your site. By integrating the generated script, all click and tracking features will be managed (refer to the use of the Journey module).

The second section lets you enable transaction and cart tracking in real-time. To do this, you first need to specify what you want to track:

- purchases only (transactions)
- both purchases and carts



For configuration to take effet and tracking to be done, you must copy the following tracking code on all the pages you want to track.

```
<html>
  <head>... </head>
  <body>
    <script type="text/javascript" src="https://t.ofsys.com/js/Journey/1/MwEAAHpHZXhlTckYAABKcmJINjVpAAAA
    /DI.Journey-min.js"></script>
  </body>
</html>
```

E-Commerce Journey lets you follow up on completed transactions and those placed in shopping carts.

E-Commerce Journey lets you follow up on completed transactions and those placed in shopping carts.

Transactions
Follow up on completed transactions only

Transactions and carts
Follow up on both completed transactions and carts

Once the Journey module is enabled and configured, you will be presented with examples of use for each available feature.

Shopping Cart Tracking

The shopping cart is a collection of items that the visitor has put on hold for purchase. In order not to loose any information, it is necessary to accurately synchronize the information you wish to retrieve in Dialog Insight with the order the contact is about to place.

To do so, you must create a "tracking cart" when you create the "shopping cart" using the *AddCart* method (see Adding a Cart)

Once the tracking cart has been created, you need to update it the same way you would update the shopping cart. When a visitor adds an item in his shopping cart, you must also add this item to the tracking cart using the *AddCartItem* method (see Adding an Item in a Cart). Then, if an item is deleted from the shopping cart, it also needs to be removed from the tracking cart using the *RemoveCartItem* method (see Removing an Item from a Cart).

When an item is added or removed from the tracking cart using these 2 methods, only the list of items in the tracking cart is edited. To update the quantity and price of a cart for example, you also need to update this cart using the *UpdateCart* method (see Updating a Cart).

All these methods allow you to store, on the client side, a tracking cart that is synchronized with your site's shopping cart. But to have all this information transferred to Dialog Insight's E-Commerce tables, you must "push" the information using the *SubmitCart* method (see Submitting a Cart).

The objective is to push the collected cart information each time the user performs an action (after each "post" for example). So after each addition, deletion, quantity change, you must call the *SubmitCart* method. Obviously, when adding or deleting a batch of items, you must first process the items added and then complete by calling the *SubmitCart* method at the end.

Important Points to Consider

- Tracking cart information is stored in E-Commerce tables for 15 days.

- When a cart is used in a transaction, it is removed from our systems.

- The tracking cart has a “Status” field to hold information on the cart’s status. Although it is not required, it is important to add this information to better manage data for following up on carts.

  Example: If you want to target only some of the contacts who have abandoned their shopping cart, such as those who were put off by delivery fees and freight charges, you just need to search for abandoned carts whose “Status” field is equal to “*ValidateShippingStep*” for example. It is therefore important to specify the status of the cart each time the client evolves in the buying process (selecting a product, selecting the delivery address, selecting the delivery mode, validating the order, etc.)

1. Adding a Cart

This method lets you create a tracking cart on the "client's side".
Note: After creating or editing this cart, it is important to submit it using the SubmitCart method as no Javascript object will survive during the client's browsing.

Example of use:

```
DI.Journey.ECommerce.AddCart(
{
        idCart: '43234'
        status: 'ValidateAddressStep',//Not required
        TotalPrice: '50.00'
});
```

Parameters:

- idCart: NVarChar(500)
- status: NVarChar(50)
- TotalPrice: Money

Return value:

None

2. Updating a Cart

This method lets you update one or multiple fields in the tracking cart. It is used to update the cart's status each time an item is added, deleted or edited in order to update the "TotalPrice".

Example of use:

```
DI.Journey.ECommerce.UpdateCart(
{
        idCart: '43234'
        status: 'CheckOutStep',// Not required
        TotalPrice: '50.00'// Not required
});
```

Parameters:

- idCart: NVarChar(500)
- status: NVarChar(50)
- TotalPrice: Money

Return value:

None

3. Adding an Item in a Cart

This method is used to add an item to a tracking cart that was previously created.

Example of use:

```
DI.Journey.ECommerce.AddCartItem(
{
       idCart: '43234',
       idProduct: '54368',
       productName: 'M-260',
       category: 'PC',// Not required
       idCategory: '123',// Not required
       priceunit: '100',
       quantity: '1'
});
```

Parameters:

- idCart: NVarChar(500) – Required
- idProduct: NVarChar(500) – Required
- productName: NVarChar(500) – Required
- category: NVarChar(500) – Not required
- idCategory: NVarChar(500) – Not required
- priceunit: float– Required
- quantity: int– Required

Return value:

None

4. Updating an Item in a Cart

This method is used to update an item in the tracking cart in order, for instance, to change its price (after a rebate coupon has been applied for example), or to change the ordered quantity. For example, if a client adds the same item twice in his shopping cart, you don't want to add two AddCartItem, but only one AddCartItem when the first item is added and then use UpdateCartItem when the second item is added.

Example of use:

```
DI.Journey.ECommerce.UpdateCartItem(
{
        idCart: '43234',
        idProduct: '54368',
        productName: 'M-260', // Not required
        category: 'PC', // Not required
        idCategory: '123', // Not required
        priceunit: '100', // Not required
        quantity: '1' // Not required
});
```

Parameters:

- idCart: NVarChar(500) – Required
- idProduct: NVarChar(500) – Required
- productName: NVarChar(500) – Required
- category: NVarChar(500) – Not required
- idCategory: NVarChar(500) – Not required
- priceunit: float– Required
- quantity: int– Required

Return value:

None

5. Removing an Item from a Cart

This method lets you remove an item from a tracking cart. Note that this method does not update information in the tracking cart. It is therefore recommended to use the UpdateCart method once an item is deleted from the cart so that the price is updated correctly.

Example of use:

```
DI.Journey.ECommerce.RemoveCartItem(
{
       idCart: '43234',
       idProduct: '54368'
});
```

Parameters:

- idCart: string(500) – Required
- idProduct: string(500) – Required

Return value:

None

6. Deleting a Cart

This method is used to delete all previously added items in a tracking cart. This method replaces the use of RemoveCartItem method for each cart item.

Example of use:

```
DI.Journey.ECommerce.ClearCart();
```

Parameters:

None

Return value:

None

7. Submitting a Cart

This method is used to send to Dialog Insight all the tracking cart information, and the items in this cart. Without this method, no information will be transferred in Dialog Insight's E-Commerce custom tables.

Example of use:

```
DI.Journey.ECommerce.SubmitCart();
```

Parameters:

None

Return value:

None

Examples of Use

1. New Cart

Let’s say a contact browses your website and decides to add an item in their shopping cart. First, you need to create the cart before you can add the first item. Then, this cart must be submitted to Dialog Insight so that the information is passed on our E-Commerce tables.

```
<html>
<head><title>Item added!</title></head>
<body>
This item has just been added to your cart.
Do you wish to continue browsing or complete your order?

<script type="text/javascript"
src="https://t.ofsys.com/js/ABCDEF.../DI.Journey-min.js"></script>

<script type="text/javascript">
DI.Journey.ECommerce.AddCart(
{
       idCart: 'C001'
       status: 'SelectProducts',// Not required
       TotalPrice: '870'
});

DI.Journey.ECommerce.AddCartItem(
{
       idCart: 'C001',
       idProduct: '1234',
       productName: 'M-260',
       category: 'PC',// Not required
       idCategory: '123',// Not required
       priceunit: '870',
       quantity: '1'
});

DI.Journey.ECommerce.SubmitCart();
</script>

</body>
</html>
```

2. Deleting and Editing Items

Let’s say that after putting items in his shopping cart, the user decides to delete an item and change the quantity of another one. Following is an HTML example of the actions to be duplicated in the tracking cart.

```
<html>
<head><title>Edit my shopping cart</title></head>
<body>

The selected items have been deleted from your cart.

<script type="text/javascript"
src="https://t.ofsys.com/js/ABCDEF.../DI.Journey-min.js"></script>

<script type="text/javascript">

DI.Journey.ECommerce.RemoveCartItem(
{
       idCart: 'C012',
       idProduct: '222'
});
 DI.Journey.ECommerce.RemoveCartItem(
{
       idCart: 'C012',
       idProduct: '333'
});

DI.Journey.ECommerce.UpdateCartItem(
{
       idCart: 'C012',
       idProduct: '111',
       productName: 'Cable HDMI',
       priceunit: '19.99',
       quantity: '1'
});

DI.Journey.ECommerce.SubmitCart();

</script>
</body>
</html>
```

Managing Transactions

A transaction is a purchase that has been done on the client’s E-Commerce site. No transaction should be created and pushed to Dialog Insight before the end page of the order.

In concrete terms, a transaction is used to buy one or more items or to transform a cart into a purchase. If there are not real payment transaction, then you need to use the methods described above for tracking carts.

idCart of Tracking Cart

If there is a tracking cart, it is important to include its **idCart** when creating the transaction. This allows us to know that the cart created by the client has been converted into a purchase and has not be abandoned. Therefore, creating a transaction will result in deleting the related tracking cart. This information is very important as if you want to send reminders for abandoned carts, you will want to exclude the carts that became purchases.

1. Adding a Transaction

This method is used to create a new transaction on the “client side”. This transaction lets you connect the transaction to a previously used shopping cart or to add items without having a cart. In both cases, it is necessary to add all the items that have been purchased.

Example of use:

```
DI.Journey.ECommerce.AddTransaction(
{
        idTransaction: '1234',
        idCart: '1234', // Not required
        affiliation: 'MyStore', // Not required
        revenue: '0',
        tax: '0',
        shipping: '0'
});
```

Parameters:

- idTransaction: string(500) – Required
- idCart: string(500) – Not required
- affiliation: string(500) – Not required
- revenue: float – Required
- tax: float – Required
- shipping: float - Required

Return value:

None

2. Adding an Item in a Transaction

This method is used to add an item to an existing transaction. All items must be added to the transaction before being submitted to Dialog Insight.

Example of use:

```
DI.Journey.ECommerce.AddItem(
{
       idTransaction: '1234',
       productName: 'M-260',
       idProduct: '54368',
       category: 'PC',// Not required
       idCategory: '123',// Not required
       Price_unit: '100',
       Price_total: '100',
       quantity: '1'
});
```

Parameters:

- idTransaction: string(500) – Required
- productName: string(500) – Required
- idProduct: string(500) – Required
- category: string(500) – Not required
- idCategory: string(500) – Not required
- Price_unit: float – Required
- Price_total: float – Required
- quantity: int – Required

Return value:

None

3. Submitting a Transaction

This method submits to Dialog Insight all the information related to the existing transaction and all its related items. Without this method, no information is pushed in Dialog Insight's E-Commerce custom tables.

Example of use:

```
DI.Journey.ECommerce.SubmitTransaction();
```

Parameters:

None

Return value:

None

Example of Use

After having entered its payment information, the client is redirected to a page that confirms the placed order. It's at this moment, when the transaction is final, that the information is sent to Dialog Insight.

To do so, in the order's confirmation page, you need to create a transaction using the *AddTransaction* method and then add all the items in the cart when the purchase is done by calling the *AddItem* method for each item. At the end, using the *SubmitTransaction* method will add all this information in our database.

Be certain to pass the **idCart** in the transaction if you want to know which carts have been converted into purchases and have not been abandoned.

Example of use:

```
<html>
<head><title>Your order has been placed!</title></head>

<body>

Thank you for doing business with us!
The 3 items your ordered will be delivered soon.

<script type="text/javascript"
src="https://t.ofsys.com/js/ABCDEFGH.../DI.Journey-
min.js"></script>

<script type="text/javascript">

// Create the transaction
DI.Journey.ECommerce.AddTransaction(
{
        idTransaction: '1234',
        idCart: '7890', // Not required
        affiliation: 'MyStore', // Not required
        revenue: '155',
        tax: '23.25',
        shipping: '7.99'
});
// Add all the items in the previously used cart
DI.Journey.ECommerce.AddItem(
{
        idTransaction: '1234',
        productName: 'HUB-USB 48 ports',
        idProduct: '4976',
        category: 'PC Accessories',// Not required
        idCategory: '79',// Not required
        Price_unit: '80',
        Price_total: '80',
        quantity: '1'
});

DI.Journey.ECommerce.AddItem(
{
        idTransaction: '1234',
        productName: 'HDMI cable',
        idProduct: '13745',
        category: 'Cables',// Not required
        idCategory: '17',// Not required
        Price_unit: '37.50',
        Price_total: '75',
        quantity: '2'
});

// Send transactional information to Dialog Insight
DI.Journey.ECommerce.SubmitTransaction();

</script>

</body>
</html>
```

Contact

Canada: 1 866 529-6214
France: 01 84 88 40 66
Russia: +7 (495) 226-04-11

Email: info@dialoginsight.com
Website : www.dialoginsight.com
Blog: www.dialoginsight.com/en/resources/academy

@DialogInsight
Dialog Insight