

# **Tealium Oracle eCommerce Module**

## **Developers Manual**

Version 1.0.1

## Table of Contents

Legal notes.....	3
Introduction.....	4
Who should read this document.....	5
Compatibility notes.....	6
Installation instructions.....	7
Module Contents.....	8
Integration guide.....	9
Configuration options.....	9
Tealium droplets.....	10
Using Tealium droplets.....	10
/tealium/droplet/UtagSyncDroplet.....	10
/tealium/droplet/GenericPageDroplet.....	10
/tealium/droplet/HomePageDroplet.....	11
/tealium/droplet/CustomerDetailDroplet.....	12
/tealium/droplet/ProductDetailDroplet.....	13
/tealium/droplet/CategoryPageDroplet.....	14
/tealium/droplet/ShoppingCartDroplet.....	15
/tealium/droplet/OrderConfirmationDroplet.....	16
/tealium/droplet/SearchResultsDroplet.....	18
Extending Tealium tags generation.....	19

## **Legal notes**

- TealiumIQ is registered trademark of Tealium, Inc
- Oracle e-Commerce, previously know as Oracle ATG is registered trade mark or Oracle Corporation
- Oracle Java is registered trademark of Oracle Corporation

All right reserved.

## **Introduction**

The ATG Tealium module is designed to help you integrate your Oracle ATG e-Commerce solution within Tealium tag management. Using this module you are able to put Tealium Data Layer and JavaScript tag on your JSP pages without writing your own solution from scratch.

## **Who should read this document**

This document is for developers who need to integrate an Oracle ATG web site with Tealium Tag Management. Please note that you should be familiar with ATG e-Commerce product core and eCommerce functionality. If you are not familiar with Oracle ATG please read the ATG Platform Developer guide and ATG Commerce Developer Guide documents provided with your ATG distribution package.

## **Compatibility notes**

ATG Tealium module is compatible with Oracle ATG 9 up to Oracle e-Commerce 11 including all minor releases.

# Installation Instructions

Tealium module is distributed as packed file in deflate ZIP format or tar.bz2 format. In order to install module, unpack content of **atg-tealium-module.zip** or **atg-tealium-module.tar.bz2** into ATG installation directory, (should point to DYNAMO\_ROOT environment variable). API documentation in javadoc format and this document provided separately.

Github: <https://github.com/Tealium/oracle-commerce-tealium-extension>

Please be aware that ATG is properly installed on your operating system and all environment variables are set. If not please follow ATG platform installation instructions, which is provided by Oracle Corporation.

For example:

- On Microsoft Windows open **atg-tealium-module.zip** in Explorer, copy Tealium folder and paste it into ATG installation directory i.e. **%DYNAMO\_ROOT%**
- On GNU/Linux you can use following command:

```
tar -xjf atg-tealium-module.tar.bz2 ${DYNAMO_ROOT}
```

# Module Contents

The distribution package adds the following module folders into your ATG installation directory

**Tealium\**

**Tealium\ eCommerce\config**

**Tealium\ eCommerce\config**

**Tealium\ eCommerce\config\config.jar**

**Tealium\ eCommerce\lib**

**Tealium\ eCommerce\lib\classes.jar**

**Tealium\ eCommerce\lib\commons-lang3-3.2.1.jar**

**Tealium\ eCommerce\lib\guava-16.0.1.jar**

**Tealium\ eCommerce\lib\tealium\_undo\_helper\_1\_2\_0.jar**

**Tealium\ eCommerce**

**Tealium\ eCommerce\META-INF**

**Tealium\ eCommerce\META-INF\MANIFEST.MF**

**Tealium\META-INF**

**Tealium\META-INF\MANIFEST.MF**



# Integration Guide

To be able to use Tealium ATG module with in your application, you will need to add a dependency to your application module. For example, if you'd like to add such functionality to the ATG Commerce Reference Store, then you need to edit

**`${DYNAMO_ROOT}/CommerceReferenceStore/Store/Storefront/META-INF/MANIFEST.MF`** with adding **Tealium.eCommerce** dependency into **ATG-Required** section.

Alternatively you can add **Tealium.eCommerce** dependency into **runAssembly** arguments used for constructing ATG enterprise archive EAR.

Tealium ATG module uses ATG droplets for rendering Tealium JavaScript tags on your site. Just input appropriate droplets on JSP pages where you need the tag management. Please be aware that Tealium utag.sync.js script is required in the `<head>` HTML section, the rest of Tealium scripts must be rendered in the `<body>` HTML section. The utag.sync.js script is option and only required for synchronous tags. A typical setup will only have utag.js running asynchronously in the web page.

To configure your Tealium credentials you need to redefine the **/tealium/config/TealiumConfiguration** ATG component configuration some were in you configuration path.

It is strongly recommended to hold this configuration at live configuration. i.e. for development environment it is:

**`${DYNAMO_HOME}/servers/<your server>/localconfig`** folder.

And for standalone retirements is your **ATG\_DATA** directory.

## Configuration options

Thallium configuration contains flowing settings:

- **enabled** – boolean, allows enabling or completely disabling generation of all Tealium tags, optional value default is true.
- **tealiumCDN** – a string value which points on Tealium URI, do not specify it until you don't need a special value, provided to you from TealiumIQ support
- **accountName** – a string value represent your Tealium account name, required
- **profileName** – a string value represent your Tealium profile name, required

- **environmentName** - a string value represent your Tealium environment name, required
- **utagSyncEnabled** - allows enabling or completely disabling generation of Tealium utag sync script, optional default is true
- **allowCustomUDO** - allows generation of custom UDO options, reserved for the future uses, not required

## Tealium droplets

### Using Tealium droplets

To be able to render Tealium tags on your web shop front end, you need to insert Tealium droplet to the front end JSP files. Those droplets, will generate the JavaScript tags, which are required by Tealium tag management back-end system. You can put those droplets code to the main application header block JSP, or directly to the page block, depending of your application. In case of Oracle eCommerce CRS 11, you can use the **%DYNAMO\_ROOT%\CommerceReferenceStore\Store\Storefront\j2ee-apps\Storefront\store.war\includes\pageStart.jsp** file for this purpose.

Tealium droplets specification follows

**/tealium/droplet/UtagSyncDroplet**

Renders the Tealium utag.sync.js tag into the page. Please be aware that this droplet must be used in <head> HTML section.

#### Example

```
<!-- Serve the Tealium eCommerce utag.sync.js script --%>
<dsp:droplet name="/tealium/droplet/UtagSyncDroplet">
</dsp:droplet>
```

Generates flowing script:

```
<script type='text/javascript'
src='//tags.tiqcdn.com/utag/account/profile/environment/utag.sync.js'>
</script>
```

## /tealium/droplet/GenericPageDroplet

Generates Tealium script for the generic (unknown) page type

### Parameters

Generic page droplet requires flowing parameters from page.

- **pageName** – string, name of the page, required
- **language** – string current site locale, required
- **currency** – string, current site currency, required

### Example

```
<dsp:droplet name="/tealium/droplet/GenericPageDroplet">
  <dsp:param name="pageName" value="testPage" />
  <dsp:param name="language" value="en_US" />
  <dsp:param name="currency" value="USD" />
</dsp:droplet>
```

### Generates flowing tags

```
<script type='text/javascript'>
  var utag_data =
  {
    "site_currency":"USD",
    "site_region":"en",
    "page_type":"generic",
    "page_name":"testPage"
  }
</script>
<script type='text/javascript'>
  (function(a,b,c,d){
    a="//tags.tiqcdn.com/utag/account/profile/environment//utag.js";
    b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
    a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
  })();
</script>
```

## /tealium/droplet/HomePageDroplet

Generates Tealium script for the home page type

### Parameters

- **pageName** – string, name of the page, required
- **language** – string current site locale, required

- **currency** - string, current site currency, required

## Example

```
<dsp:droplet name="/tealium/droplet/HomePageDroplet">
  <dsp:param name="pageName" value="Home Page" />
  <dsp:param name="language" value="en_US" />
  <dsp:param name="currency" value="USD" />
</dsp:droplet>
```

## Generates flowing tags

```
<script type='text/javascript'>
  var utag_data =
{
  "site_currency":"USD",
  "site_region":"en_US",
  "page_type":"home",
  "page_name":"Home Page"
}
</script>
<script type='text/javascript'>
  (function(a,b,c,d){
    a="//tags.tiqcdn.com/utag/account/profile/environment/utag.js";
    b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
    a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
  })();
</script>
```

## /tealium/droplet/CustomerDetailDroplet

Generates Tealium tags for the account page type

## Parameters

- **pageName** - string, name of the page, required
- **language** - string current site locale, required
- **currency** - string, current site currency, required
- **profile** - atg.userprofiling.Profile, required

## Example

```
<dsp:droplet name="/tealium/droplet/CustomerDetailDroplet">
  <dsp:param name="pageName"
    value="myAccountPage" />
  <dsp:param name="language" value="en_US" />
  <dsp:param name="currency" value="USD" />
  <dsp:param name="profile" bean="/atg/userprofiling/Profile"/>
</dsp:droplet>
```

## Generates flowing tags

```

<script type='text/javascript'>
    var utag_data =
    {
        "site_currency":"USD",
        "customer_email":"test@example.com",
        "customer_name":"Test",
        "site_region":"en_US",
        "gender":"male",
        "page_name":"myAccountPage",
        "customer_type":"",
        "customer_id":"tstUsr"
    }
</script>
<script type='text/javascript'>
    (function(a,b,c,d){
        a='//tags.tiqcdn.com/utag/account/profile/environment//utag.js';
        b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
        a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
    })();
</script>

```

## /tealium/droplet/ProductDetailDroplet

Generates Tealium tags for the product details page

### Parameters

- **pageName** – *string, name of the page, required*
- **language** – *string current site locale, required*
- **currency** – *string, current site currency, required*
- **product** – *atg.repository.RepositoryItem representing the ATG catalog product, required*

### Example

```

<dsp:importbean bean="/tealium/droplet/ProductDetailDroplet" />
<dsp:droplet name="/atg/commerce/catalog/ProductLookup">
    <dsp:param name="id" param="productId" />
    <dsp:oparam name="output">
        <dsp:droplet name="ProductDetailDroplet">
            <dsp:param name="pageName"
                value="Product Page" />
            <dsp:param name="language" value="en_US" />
            <dsp:param name="currency" value="USD" />
            <dsp:param name="product" param="element" />
        </dsp:droplet>
    </dsp:oparam>
</dsp:droplet>

```

## Generates flowing tags

```
<script type='text/javascript'>
    var utag_data =
{
    "page_category_name": "",
    "page_subcategory_name": "",
    "product_id": ["TP0"],
    "product_name": ["Test1Product"],
    "site_region": "en_US",
    "page_section_name": "",
    "product_unit_price": ["10.0"],
    "product_category": ["Test0"],
    "site_currency": "USD",
    "product_sku": ["TSKU0"],
    "product_brand": ["TestBrand0"],
    "page_type": "product",
    "page_name": "Test Product",
    "product_list_price": ["10.0"]
}
</script>
<script type='text/javascript'>
    (function(a,b,c,d) {
        a="//tags.tiqcdn.com/utag/account/profile/environment/utag.js";
        b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
        a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
    })();
</script>
```

## /tealium/droplet/CategoryPageDroplet

Generates Tealium tags for the browse category content page

### Parameters

- **pageName** – string, name of the page, required
- **language** – string current site locale, required
- **currency** – string, current site currency, required
- **category** – atg.repository.RepositoryItem representing the ATG catalog category, required

### Example

```
<dsp:droplet name="/atg/commerce/catalog/CategoryLookup">
    <dsp:param name="id" param="categoryId" />
    <dsp:oparam name="output">
        <dsp:droplet name="/tealium/droplet/CategoryPageDroplet">
            <dsp:param name="category" param="element" />
            <dsp:param name="pageName" value="categoryPage" />
            <dsp:param name="language" value="en_US" />
            <dsp:param name="currency" value="USD" />
        </dsp:droplet>
    </dsp:oparam>
</dsp:droplet>
```

```

        </dsp:oparam>
    </dsp:droplet>

```

## Generates flowing tags

```

<script type='text/javascript'>
    var utag_data =
    {
        "page_category_name":"TestCat0",
        "page_subcategory_name": "",
        "site_currency": "USD",
        "site_region": "en_US",
        "page_type": "product",
        "page_section_name": "",
        "page_name": "Category Page"
    }
</script>
<script type='text/javascript'>
    (function(a,b,c,d){
        a='//tags.tiqcdn.com/utag/account/profile/environment/utag.js';
        b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
        a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
    })();
</script>

```

## /tealium/droplet/ShoppingCartDroplet

Generates Tealium tags for the basket page

### Parameters

- **pageName** - string, name of the page, required
- **language** - string current site locale, required
- **currency** - string, current site currency, required
- **shoppingCart** - atg.commerce.order.OrderHolder ATG shopping cart component, required

### Example

```

<dsp:importbean bean="/tealium/droplet/ShoppingCartDroplet" />
<dsp:droplet name="ShoppingCartDroplet">
    <dsp:param name="pageName" value="basket" />
    <dsp:param name="language" value="en_US" />
    <dsp:param name="currency" value="USD" />
    <dsp:param name="shoppingCart" bean="/atg/commerce/ShoppingCart" />
</dsp:droplet>

```

## Generates flowing tags

```

<script type='text/javascript'>
    var utag_data =
    {
        "product_category":["TestCat","TestCat"],
        "product_id":["TP0","TP0"],
        "product_unit_price":["100.0","200.0"],
        "product_sku":["TSKU0","TSKU1"],
        "site_currency":"USD",
        "product_name":["TestSKU0","TestSKU1"],
        "product_quantity":["1","2"],
        "product_brand":["TestBrand","TestBrand"],
        "site_region":"en",
        "cart_total":"560.0",
        "page_type":"checkout",
        "page_name":"ShoppingCard",
        "product_list_price":["100.0","400.0"]
    }
</script>
<script type='text/javascript'>
    (function(a,b,c,d){
        a='//tags.tiqcdn.com/utag/account/profile/environment/utag.js';
        b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
        a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
    })();
</script>

```

## /tealium/droplet/OrderConfirmationDroplet

Generates Tealium tags for the order confirmation page

### Parameters

- **pageName** - string, name of the page, required
- **language** - string current site locale, required
- **currency** - string, current site currency, required
- **order** - atg.commerce.order.Order, ATG order bean, required
- **userEmail** - string, customer email, optional

### Example

```

<dsp:getvalueof var="currentProfile" bean="/atg/userprofiling/Profile"/>
<%-- Handle the guest checkout without email --%>
<dsp:getvalueof var="userEmail" value="unknown"/>
<c:if test="${not currentProfile.transient}">
    <dsp:getvalueof var="userEmail" value="${currentProfile.email}" />
</c:if>
<dsp:droplet name="/tealium/droplet/OrderConfirmationDroplet">
    <dsp:param name="pageName" value="orderConfirmation" />
    <dsp:param name="language" value="en_US" />
    <dsp:param name="currency" value="USD" />
    <dsp:param name="order" bean="/atg/commerce/ShoppingCart.current" />
    <dsp:param name="userEmail" value="${userEmail}"/>
</dsp:droplet>

```



## Generates flowing tags

```
<script type='text/javascript'>
    var utag_data =
{
    "order_total":"560.0",
    "order_currency":"USD",
    "order_shipping":"10.0",
    "product_id":["TP0","TP0"],
    "order_payment_type":"Debit card",
    "product_name":["TestSKU0","TestSKU1"],
    "product_quantity":["1","2"],
    "customer_email":"test@example.com",
    "site_region":"en",
    "order_id":"TSTORD1",
    "order_subtotal":"610.0",
    "order_tax":"100.0",
    "product_category":["TestCat","TestCat"],
    "product_unit_price":["100.0","200.0"],
    "product_sku":["TSKU0","TSKU1"],
    "site_currency":"USD",
    "product_brand":["TestBrand","TestBrand"],
    "page_type":"checkout",
    "page_name":"Order Confirmation",
    "product_list_price":["100.0","400.0"],
    "order_discount":"100.0",
    "product_discount":["0.0","50.0"],
    "customer_id":"testuser"
}
</script>
<script type='text/javascript'>
    (function(a,b,c,d){
        a="//tags.tiqcdn.com/utag/account/profile/environment/utag.js";
        b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
        a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
    })();
</script>
```

## /tealium/droplet/SearchResultsDroplet

Generates Tealium tags for the search results page

### Parameters

- **pageName** - *string, name of the page, required*
- **language** - *string current site locale, required*
- **currency** - *string, current site currency, required*
- **searchKeyword** - *string, keyword used for searching*
- **totalResultsNumber** - *string, with digital value - total number of search results*

### Example

```
<dsp:droplet name="/tealium/droplet/SearchResultsDroplet">
  <dsp:param name="pageName" value="Search Results" />
  <dsp:param name="language" value="en_US" />
  <dsp:param name="currency" value="USD" />
  <dsp:param name="searchKeyword" param="question"/>
  <dsp:param name="totalResultsNumber" param="resultsCount"/>
</dsp:droplet>
```

### Generates flowing tags

```
<script type='text/javascript'>
  var utag_data =
  {
    "site_currency":"USD",
    "site_region":"en",
    "page_type":"search",
    "page_name":"Search Results",
    "search_results":"2",
    "search_keyword":"test"
  }
</script>
<script type='text/javascript'>
  (function(a,b,c,d){
    a="//tags.tiqcdn.com/utag/account/profile/environment/utag.js";
    b=document;c='script';d=b.createElement(c);d.src=a;d.type='text/java'+c;d.async=true;
    a=b.getElementsByTagName(c)[0];a.parentNode.insertBefore(d,a);
  })();
</script>
```

## Extending Tealium tags generation

If you need to extend the Tealium scripts generation, according your application specific, for example in case when your have customized catalog structure you can follow next instruction.

The Tealium droplets not generates scripts them self. Generation is getting proxy to the `/tealium/connector/DataConverter` component.

If you need to change some Tealium script, you should extend `com.tealium.connector.DataConverter` class, and fulfill the `com.tealium.util.udohelpers.UDO` class, and then call `getTealiumHelper().outputFullHtml`

Then redefine the `/tealium/connector/DataConverter` with new class, as any any another ATG component.

For example if I what to generate the default site currency and default site language instead of transferring it from JSP page each time, I need to write next code.

```

package com.tealium.connector;

import com.tealium.util.udohelpers.TealiumHelper;
import com.tealium.util.udohelpers.TealiumHelper.PrebuiltUDOPageTypes;
import com.tealium.util.udohelpers.UDO;

public class MyTealiumDataConverter extends DataConverter {

    private String defaultCurrency;

    private String defaultLanguage;

    @Override
    public String getGenericPageScript(String pageName, String currency,
                                      String language) {
        if (getConfiguration().isEnabled()) {
            try {
                UDO udo = setupUDO(PrebuiltUDOPageTypes.HOME, pageName,
                                   getDefaultCurrency(), getDefaultLanguage());
                udo.setValue(
                    TealiumHelper.HomePageUDO.PredefinedUDOFIELDS.PAGE_TYPE,
                    "generic");
                return getTealiumHelper().outputFullHtml(udo);
            } catch (Exception exc) {
                return getExceptionString(exc);
            }
        } else {
            return "";
        }
    }

    public String getDefaultCurrency() {
        return defaultCurrency;
    }

    public void setDefaultCurrency(String defaultCurrency) {
        this.defaultCurrency = defaultCurrency;
    }

    public String getDefaultLanguage() {
        return defaultLanguage;
    }

    public void setDefaultLanguage(String defaultLanguage) {
        this.defaultLanguage = defaultLanguage;
    }
}

```

And bean configuration property file property will look like this:

```

# /tealium/connector/DataConverter
$class=com.tealium.connector.MyTealiumDataConverter
defaultCurrency=USD
defaultLanguage=en_US

```