

# DTAG100 BrightSign Integration Guide

## Plug-in V1.02

---

### Introduction

This document describes how to use the DTAG100 plug-in for BrightSign players, which provides a simple API for controlling the attached dynamic NFC tag and Bluetooth beacon functionality from BrightAuthor interactive scripts and/or embedded HTML pages.

The DTAG100 is a specialist USB device that provides real-time dynamic control of both an NFC tag touchpoint and a Bluetooth beacon transmitter. It has been designed specifically to enable digital signage systems to make use of these technologies, so that they can interact with consumer mobile devices in the vicinity.

This can range from a simple “tap to view” response to a digital poster advertisement, to more complex interactions and engagements.

In order to change the NFC or Bluetooth data that is transmitted, you simply copy various configuration files to the DTAG100, which appears to the host system as a mass storage device (ie like a USB memory stick) and it automatically reads these and updates its settings as needed.

Users will need to be familiar with the DTAG100 and how it operates, as well as the associated Bipzone mobile application, if that is being used also.

We recommend that you test all DTAG100 configuration files in isolation before integrating the desired functionality into your BrightSign presentation or HTML content. Please refer to the **DTAG100 User Guide** and/or the **Bipzone User Guide** for details of how the various DTAG100 configuration files are formatted and used.

### Plug-in operation basics

Version 1 of the DTAG100 plug-in provides a simple mechanism for copying pre-existing DTAG100 configuration files from the player SD card to the DTAG100 mass storage drive, as and when required.

It automates the process of locating the DTAG100 device, irrespective of which USB port it has been connected to, and also makes it easy to publish the additional DTAG100 configuration files using BrightAuthor.

Depending on the requirements, one or more configuration files on the DTAG100 will typically need to be updated at certain stages/events during an interactive script, either automatically (eg at the start or end of a video segment) or according to user actions (eg when a specific button is pressed

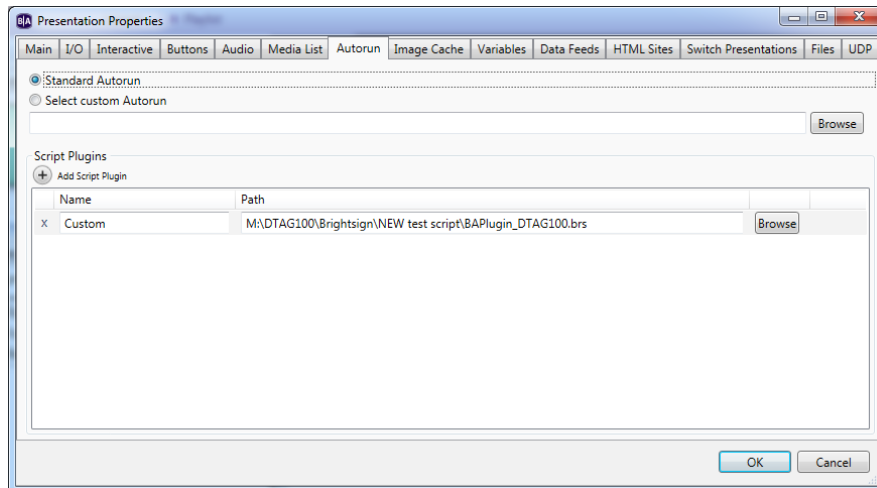
or certain HTML content is displayed). The plug-in currently provides the ability to update four specific DTAG100 configuration files, each of which has a specific purpose as follows:

<b>DTAG100 Configuration File</b>	<b>Function</b>
<i>config.txt</i>	<p>This file is used to set certain high-level DTAG100 configuration parameters, including the Bluetooth transmitted signal strength, logging/interface modes etc.. It can also be used to turn the Bluetooth and/or NFC functions on and off completely.</p> <p>Note that this file is <u>not required to be present</u> on a DTAG100 for it to operate normally, since all the config parameters have default settings that will work for many applications.</p>
<i>tagdata.txt</i>	<p>This file contains all of the information used to set the contents of the DTAG100 dynamic NFC tag. This data is transmitted to any compatible phone or other device capable of reading NFC tags, when in range of its RF field. For many applications the tag data will use the standard 'Smart Poster' format that includes a target URL that will launch when the phone is tapped onto the DTAG100 NFC hot spot.</p>
<i>bledata.txt</i>	<p>This file contains the information used to configure the DTAG100 dynamic Bluetooth beacon. This typically comprises the beacon type (such as iBeacon, Eddystone or Connectable/Bipzone) plus the transmitted unique ID and/or other appropriate beacon data (depending on the type).</p> <p>When used with the Bipzone app, the appropriate bledata file contents will be provided for each Bipzone brand and location being used. For other mobile applications/service providers you will need to obtain the appropriate information and create suitable bledata files accordingly.</p>
<i>bleattr.txt</i>	<p>This file is primarily used to set the message and target URL data for Bipzone beacons, in a similar way to the tagdata file for NFC tags.</p>

Later versions of the plug-in will provide the ability to write URLs and other content directly into the appropriate DTAG100 config files, without the need to pre-prepare and save these files in advance.

## Installing the DTAG100 plug-in script

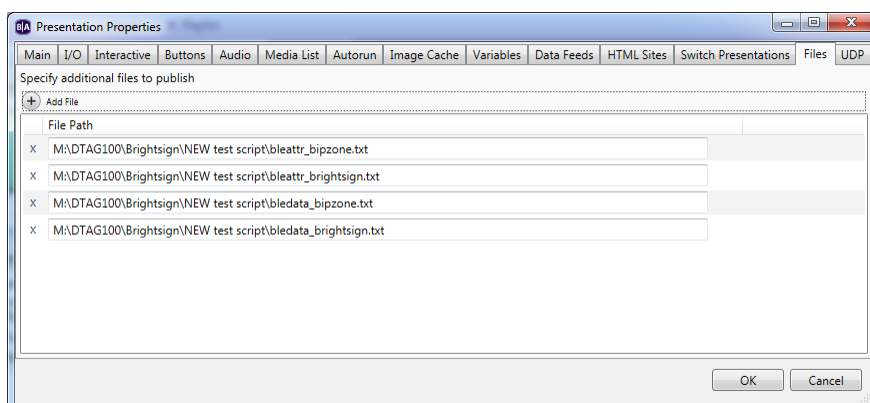
In order use the DTAG100 plug-in from within a BrightAuthor presentation, it must first be included as a new script plug-in. To do this, create or open the presentation, click File – Presentation Properties, then choose the Autorun tab and click on Add Script Plugin. Type ‘Custom’ into the name field and enter the path to the *BAPugin\_DTAG100.brs* script file provided (or use the Browse function to find the file):



## Including your DTAG100 config files

We recommend that you pre-prepare the various DTAG100 configuration files needed for your application, and give them suitable file names that indicate which type of file they are, and when they should be used (see the examples provided in Appendix A).

Once these have been tested in isolation, by copying them manually to your DTAG100 and checking that they operate as required, you can then include these files in the BrightAuthor presentation to enable automatic operation. To do this, click File – Presentation Properties again, then choose the Files tab and click on Add File to include one or more config files as needed:



## Using the plug-in from BrightAuthor

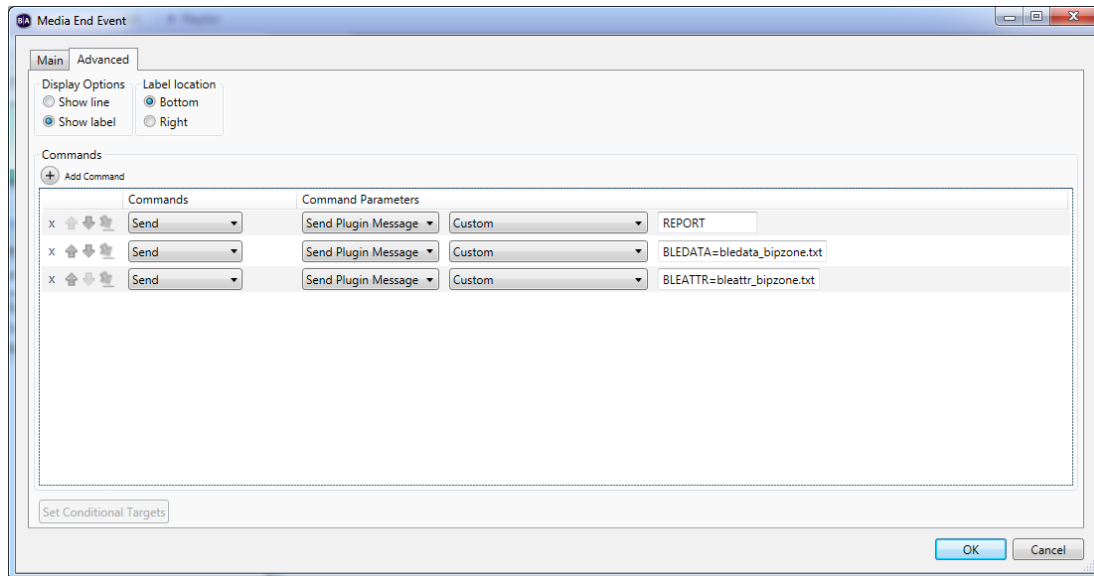
In order to control the DTAG100 and update its configuration from within an interactive BrightAuthor presentation, you must send specific messages to the plug-in, as follows:

Message string	Function
TAGDATA=<filename.txt>	This copies the named source file to /tagdata.txt on the DTAG100. If the file name starts with a '/' or '\' then it is assumed to exist in the root directory of the SD card, otherwise it is searched for automatically within the asset pool (which is where the files will be located if they are included in the presentation as described above).
BLEADATA=<filename.txt>	This copies the named source file to /bledata.txt on the DTAG100, according to the same naming rules as above.
BLEATTR=<filename.txt>	This copies the named source file to /bleattr.txt on the DTAG100, according to the same naming rules as above.
CONFIG=<filename.txt>	This copies the named source file to /config.txt on the DTAG100, according to the same naming rules as above.
REPORT	This is primarily used for set-up and debug purposes, and will cause the found DTAG100 device name (ie USBn) plus the contents of boot.txt & bleinfo.txt (if present) to be printed as standard serial or Telnet debug output. If no DTAG100 is found then this will be reported instead.

To do this from within BrightAuthor, you would normally use the Add Command function from within a specific interactive event, and then choose Send – Send Plugin Message – Custom, plus the required message string according to the format above.

Note that you may wish to update multiple config files on the DTAG100, for example to change the NFC tag, Bluetooth beacon ID and/or Bipzone contents all at once. To do this, you simply need to send multiple messages to the plug-in one after the other.

Here is an example of multiple messages being sent as part of a media end event:



## Using the plug-in from JavaScript

A very similar mechanism is used to send commands to the plug-in from within HTML content, using JavaScript. In this case, the command message is split into two values – firstly the operation string, which defines which file is to be updated or other function is to be performed (eg ‘TAGDATA’ or ‘REPORT’, as per the table above) and secondly the source filename, as before.

In order to send a message to the DTAG100, the following two lines of JavaScript should be used:

```
var bsMessage = new BSMessagePort();
bsMessage.PostBSMessage({dtag100op: "<operation>", dtag100file:
"<filename>"});
```

So for example, the following HTML fragment implements a JavaScript function to update the DTAG100 bleattr file from a specific file stored within the presentation:

```
<script>
function DTAG100_set_Bipzone_content() {
var bsMessage = new BSMessagePort();
bsMessage.PostBSMessage({dtag100op: "BLEATTR", dtag100file:
"bleattr_brightsign.txt"});
}
</script>
```

All five operation strings in the table above are supported under the JavaScript method (TAGDATA, BLEADATA, BLEATTR, CONFIG and REPORT).

Note that, in order to use the JavaScript plug-in functionality, it is important to tick the “Enable External Data” box in the individual HTML5 object within BrightAuthor, as per the example below:

