DD4T 2 Release notes

Date: XXX

# New functionality in 2.0

## Templates and data model

### Switch from XML to JSON

The DD4T templates now generate JSON instead of XML. JSON is smaller and it is faster to deserialize, making your web application faster.

Note that the web application is still able to consume XML (DD4T 1 style). See migration scenarios.

As a consequence, the template building blocks Minimize XML and Convert XML to Java have been removed.

### Compression

It is now possible to compress your data. Compression reduces the size of the published data in the broker database, and it is faster to transport from the database to your web application.

This is another way of making your web application faster.

To enable, set the template parameter 'Compression' to 'yes'.

### Publishing binaries is now the default

In previous versions, you had to add the Template Building Block 'Publish binaries for component/page' to your templates in order to publish multimedia components (binaries, like images and documents). This is no longer required. You can simply use 'Generate dynamic component presentation/page'.

### Support for Media Manager / ECL

If you use the integration with SDL Media Manager or any other External Content Library (ECL), links to these external assets are now correctly resolved during publishing. Note that the binary data itself is NOT included in the published data, only a link to the public endpoint of the asset.

To enable, set the template parameter 'ECLEnabled' to 'yes'.

### Omitting unessential data

By default, the DD4T templates used to generate many data elements which were not normally used in the web application. These (normally unnecessary) data elements can now be toggled using template parameters, as follows:

* For fields of a non-textual type, a textual representation of the values was included in the Values property (e.g. for component links, the Values property contained a list of the URIs). It is now possible to omit these textual representations by setting the template parameter OmitValueLists to ‘yes’.
* All category and keyword information was included twice: once in the context of the field itself, and once in a separate list of categories as a property of the page or component. It is now possible to omit this separate list of categories by setting the template parameter OmitCategories to ‘yes’.
* All objects in the package contained a context publication, owning publication and folder property. In reality all these are very rarely used. It is now possible to omit them by setting the template parameters OmitContextPublications, OmitOwningPublications and/or OmitFolders to ‘yes’.

**Note that the context publication, owning publication and folder/structure group of the 'root object' (the page or component being published) cannot be omitted!**

### Root object for dynamic component presentations

In previous versions, if you published a dynamic component presentation (DCP), the data would contain a Component as its root object. It is now possible to use the ComponentPresentation as root object instead. This means that the data would look like this:

{

"Component": {

"Id": "tcm:4-305",

"Title": "Andean condor"

},

"ComponentTemplate": {

"Id": "tcm:5-1710-32",

"Title": "Animal Details"

},

"IsDynamic": true,

}

... and not like this:

{

"Id": "tcm:4-305",

"Title": "Andean condor",

"Fields": [

...

]

}

This is used by the new ComponentPresentationFactory which is part of the presentation side logic of DD4T 2.0.

The benefit of this new approach is that you have access to the complete component presentation. This is especially useful if the component presentation is returned by some dynamic process, such as the SDL Content Delivery API or Smart Target. In that case, you can render the component presentations with the correct views, for example.

To enable this feature, use the TBB 'Generate dynamic component presentation' instead of 'Generate dynamic component'. We recommend to do this as soon as possible. The old TBB is still included for backwards compatibility reasons, but it is marked as obsolete. It will probably be removed in the next major release.

## .NET presentation logic

### ViewModels

TODO: describe

### Dependency injection

TODO: describe

### JMS cache invalidation

TODO: describe

### REST service

TODO: describe

## Distribution

The primary method of distribution of DD4T.NET is NuGET. We offer the following packages:

* The essentials:
  + DD4T.Model (contains the data model, which is also used by the templates)
  + DD4T.Core (the .NET core, consisting of factories, utils and contracts)
* The MVC integrations (plus binary distribution and resource management):
  + DD4T.MVC4
  + DD4T.MVC5
* The Tridion integrations:
  + DD4T.Providers.SDLTridion2011sp1
  + DD4T.Providers.SDLTridion2013
  + DD4T.Providers.SDLTridion2013sp1
  + DD4T.Providers.REST (connects to the new DD4T REST service)
* The dependency injection stuff:
  + DD4T.DI.Autofac
  + DD4T.DI.Ninject
  + DD4T.DI.Unity
* The rest:
  + DD4T.ViewModels (a ‘strongly typed model’ implementation)

To get started, you need to pick 1 MVC package, 1 Tridion package and 1 dependency injection package. You also need to have the corresponding MVC and DI frameworks installed, as well as the correct Tridion DLLs. Since Tridion does not offer its software through NuGET, these DLLs must be referenced directly.

# Bug fixes in 2.0

## Links to multimedia components in rich text

If you include a link to a multimedia component (e.g. a PDF) in your rich text field, it will now be published correctly.

## Full support for Keywords

In previous versions, there was some support for keyword fields but it was not complete. For example, if the keyword's metadata included a multimedia link, the binary was not published. Also, the metadata schema was lacking. This has been corrected in DD4T 2.0.

## Duplicate keywords

If you had more than one DD4T template building block in your templates, the keyword values were duplicated. This is fixed in DD4T 2.0.

TODO: add more bug fixes

# Changes in compatibility in 2.0

* Dropped support for ASP.NET MVC 3
* Dropped support for SDL Tridion 2009 and SDL Tridion 2011 GA (2011SP1, 2013GA and 2013SP1 are still supported)

# Changes in behaviour since version 1.31

* ResolveWidthAndHeight is now true unless explicitly set to false in a parameter (used to be false by default)
* Changed type of Multimedia.Size from int to long (because Tridion has also changed the type of the corresponding property)
* Page templates can now follow links to components and resolve binary width/height if you set the correct parameters (similarly to component templates)

# Migration scenarios

## DD4T 1.\* -> 2.0

The new DD4T .NET core can handle DD4T 1.\* style data (XML) as well as DD4T 2 style JSON. However, the new templates can only generate JSON. As a result, you need to upgrade the web application first.

After you have upgraded the web application, you can upgrade the templates. There is no need to do a mass republishing, since both formats are supported.

There is one exception: if you need one of the new features in the templates, like support for External Content Libraries like SDL Media Manager, you must republish the content with the new templates.