

[Tag Data Translation](#)

Tag Data Translation

- [Introduction](#)
 - [Users Guide](#)
-

Introduction

Tag data translation or Tag Virtualization is the process of translating the information stored on a tag from a specific encoding to another one, e.g from binary format to [GID-96](#). This process is very important especially for open loop systems. EPCglobal Inc. has defined a standard which clarifies this process for EPC based systems.

Tag virtualization capitalises on a machine-readable version of the [EPC Tag Data Standards specification](#). The actual version of TDS standard implemented in TDT is [TDT Standard v1.6](#). It was updated by using the library [tdt-1.0.0-with-dependencies](#) released by [Fosstrak](#).

This machine-readable version can be used for abstracting underlying RFID tags through bridging and mapping different representations. Hence, a tag translation module is a (standalone or embedded) middleware component that enables the interpretation of machine readable version of the tag.

The interpretation can be used in automated fashion. In addition the tag translation specification can be used for validating machine readable formats of the EPC tags. The TDT engine built in the scope of the AspireRfid project currently supports the [ISO15693](#), [ISO14443](#), [NFC](#), [MAC address](#), [phone numbers](#) (with company prefixes), various [GS1 formats](#) ([EAN/UPC](#), [GS1 DataBar](#), [GS1-128](#), [ITF-14](#), [GS1 DataMatrix](#), and [Composite Component](#)), as well as [Bar Codes 1D and 2D](#): Note that barcode support is deemed particularly important given the vast number of legacy barcode applications, which need to be interoperable with emerging RFID applications. The AspireRFID TDT supports EPC tags by using an implementation of the EPCglobal standard provided by [Fosstrak](#) project.

More information regarding tag virtualization in AspireRFID can be acquired through the public deliverable [D3.2 Readers and Tags Virtualization](#) of the Aspire FP7 project

Users Guide

This component is provided as a stand alone tool on the project forge [here](#). It depends on the Fosstrak TDT library available [here](#) (download the JAR with Dependencies file).

The file `TDTFrontEnd.java` shows how to use the AspireRfid TDT. The following source code is a quote from this file, explaining the use of the engine parameters:

```
AspireTDTEngine engine = new AspireTDTEngine();HashMap<String,String> extraparams = new
HashMap<String, String>(); // a HashMap providing extra parameters needed in addition to the
input value/** MANDATORY PARAMETERS FOR TDT */ extraparams.put("dataType","gs1"); // permitted
values are 'gs1', 'ISO' or 'phone' (not case sensitive)
extraparams.put("outputFormat","LEGACY"); // permitted values are : // for EPC Tags and GS1 Bar
Code (EPC TDS Compliant)* : 'LEGACY', 'BINARY', 'TAG_ENCODING', 'PURE_IDENTITY', 'ONS_HOSTNAME'
and 'GS1_AI_ENCODING' // for GS1 Bar Code (not EPC TDS Compliant) : 'LEGACY' and
'GS1_AI_ENCODING' // for ISO Tags : 'HEXA', 'LEGACY', 'BINARY', 'TAG_ENCODING', 'PURE_IDENTITY',
'ONS_HOSTNAME' /** PARAMETERS FOR EPC Tags TDT */ /* These parameters are mandatory for inbound
levels 'PURE_IDENTITY', 'LEGACY' or 'GS1_AI_ENCODING', */ /* and if outputFormat=GS1_AI_ENCODING
*/ extraparams.put("taglength","96"); // the taglength must be specified as "64" or "96"
extraparams.put("filter","1"); // the filter value must be specified - range depends on coding
scheme. extraparams.put("companyprefixlength","6"); // the companyprefixlength (length of the
```

Aspire Wiki - ObjectWeb - TDT

```
EAN.UCC Company Prefix) must be specified for GS1 coding schemes /** PARAMETERS FOR GS1 Bar Code
TDT **/ /* These parameters are mandatory if outputFormat=GS1_AI_ENCODING */
extraparams.put("gs1symbol","1e0"); // GS1 Bar Code Type (permitted values are '1E0', '1E1',
'1E2', '1E3', '1E4', '1I1', '1C1', '1e0' or '1d2') extraparams.put("codelength","12"); // size
of the bar code (mandatory with gs1symbol values 1E0 an 1E3), permitted values are 12 (for
UPC-A), 13 (for EAN-13) or 8 (for UPC-E)** PARAMETERS FOR PHONE NUMBER TDT **/
extraparams.put("countryprefixlength","96"); // the countryprefixlength must be specified as
"1", "2" or "3"
```

The following method provided by the AspireTDTEngine object translates the input string of a specified input level to a specified outbound level of the same coding scheme.

```
public String convert(String input, Map<String,String> inputParameters) throws TDTEException
```

For example, the input string value may be a tag-encoding URI and the outbound level specified by string `outboundlevel` may be `BINARY`, in which case the return value is a binary representation expressed as a string. The different available levels depend on the input string and are:

- `BINARY`
- `TAG_ENCODING`
- `PURE_IDENTITY`
- `LEGACY`
- `ONS_HOSTNAME`
- `GS1_AI_ENCODING`
- `HEXA`

[Tag Data Translation](#) (en)

Creator: xwiki:XWiki.nkef Date: 2009/01/29 17:54

Last Author: xwiki:XWiki.driss_r Date: 2013/08/20 15:04

Copyright (c) 2008-2010, [Aspire](#)