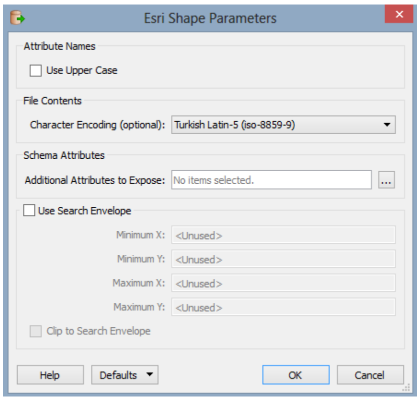
Character Encoding Case Study

CitySDK develops, together with its project partners, smart solutions for many countries. As these solutions become more common, some local issues may arise. It is important for us to share the problems we face and the solutions we find with our project partners and project implementers.

Being the first replication (endpoint) of CitySDK, one of the problems we, the Istanbul Metropolitan Municipality, faced was a basic and seemingly easy character-encoding problem. Given that other countries, which also have special characters in their national alphabets, may encounter similar problems when using ESRI products, we wanted to share our case and the solution we found.

Before inputting our spatial data into the CitySDK ecosystem, we exported data in shape format from our ESRI ArcSDE servers (one of our corporate solutions) and edited these data. We completed conversion to the WGS 1984 reference system, which is geographically projected,. CitySDK shape (.zip) supports data import in GeoJson and CSV formats. Therefore we tried using the ESRI shape format. However we noticed that the special characters in the Turkish alphabet, such as "İ", "Ş" and "Ğ", were not displayed correctly in the CitySDK platform which supports UTF-8. When we investigated the source of the problem with Waag, we found that the character encoding method was not logged in the shape format.

After some research on the web, we came across a few methods to force data into UTF-8 using advanced techniques. However, since we needed to apply this to many data sets we searched for a better optimized method with a work-flow process. We tried the Feature Manipulation Engine (FME) product of Safe Software. FME makes it possible to make changes and conversions to data formats. The solution we found was to import the shape data with ISO 8859-9 (Latin 5) Standard and convert them to GeoJson format using UTF-8 character encoding.


Esri Shape Parameters

Attribute Names

□ Use Upper Case

File Contents

Character Encoding (optional): Turkish Latin-5 (iso-8859-9)

Schema Attributes

Additional Attributes to Expose: No items selected.

□ Use Search Envelope

Minimum X: <Unused>

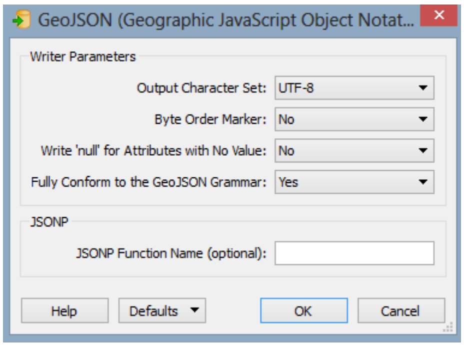
Minimum Y: <Unused>

Maximum X: <Unused>

Maximum Y: <Unused>

□ Clip to Search Envelope

Figure 1 Shape parameters


GeoJSON (Geographic JavaScript Object Notat...

Writer Parameters

Output Character Set: UTF-8

Byte Order Marker: No

Write 'null' for Attributes with No Value: No

Fully Conform to the GeoJSON Grammar: Yes

JSONP

JSONP Function Name (optional):

Figure 2 GeoJson Parameters

The basic problem we faced was that the systems could not make sense of the data, since the character encoding of the shape data was not identified. The solution offered by GeoJson satisfied out expectations.

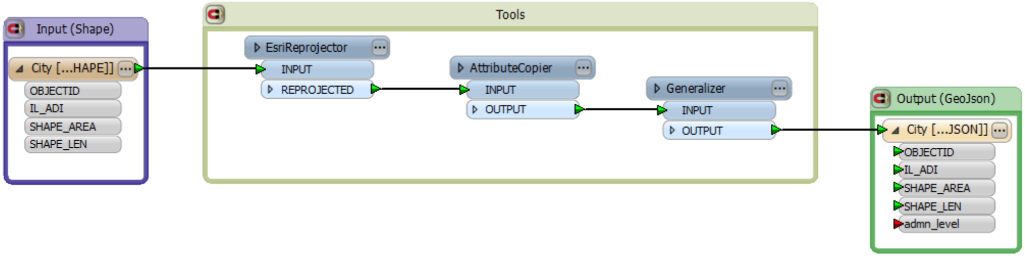


Figure 3 Shape2 GeoJson Conversion

We advise project partners who encounter similar character encoding problems to bear in mind the possible character encoding problems when working with shape data and to try working in CSV or GeoJson formats.