

First, what is the primary difference between and TGF format and an Inspection Template. The answer is, almost nothing. We differentiate the two for one reason and one reason only. The Inspection Template has two types of fields that are critical to it's use and are unique to it. The first is the **PRIMARY KEY**. A primary key is the unique ID that every query able database must have so that you can easily find a record. In order for inspections to work, **THEY MUST ALWAYS HAVE A PRIMARY KEY**. Next is the **BARCODE FIELD**. Because MUS Inspection allows records to be queried with barcode scanners, you can make your primary key "Barcodable." Those are the only differences. The rest of what is described below works for both Inspection Templates and TGF files.

TGF and Inspection files can be created in TGT DataView with our creation engine. However, this tutorial explains how to create and edit these files in a generic text editing program like WordPad or NotePad....

STANDARD INSPECTION FORMAT:

```
<?xml version="1.0" standalone="yes"?>
<GenericInspection>
  <SubItem>
    <Name>AutoID</Name>
    <Label>Auto ID!</Label>
    <Info>This is the info button</Info>
    <Data>String</Data>
    <Control>AutoID</Control>
    <Required>No</Required>
    <Barcode>Yes</Barcode>
    <Tab>Tab1</Tab>
    <PrimaryKey>Yes</PrimaryKey>
  </SubItem>
</GenericInspection>
```

WHAT IT LOOKS LIKE IN INSPECTION:

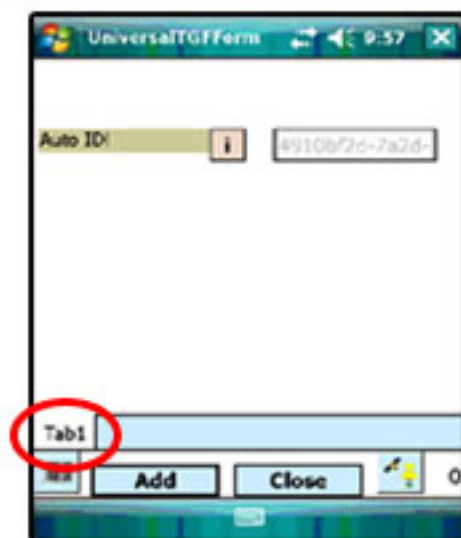


This is an inspection form in it's simplest state. Lets add a few items...

These fields are **REQUIRED**:

- **Header Field:** `<?xml version="1.0" standalone="yes"?>`
- **TGF/Inspection Name** (*name your form*)
- **A SubItem** (*Subitems correspond to the fields you wish to see.*)

*** DO NOT edit the Header Field****



In this inspection you see that our one field is applied to a Tab called "Tab1". It only has one field that we made "Barcodeable". It is also our Primary Key for the database.

The rest of this tutorial will be spent explaining the different types of "TAGS" that are supported.

- 1) Name:** Name is the actual field name that will match up with the field you wish to have in your database. Most databases are Case Sensitive and you will need to be specific about what you place in the name. Additionally, most databases, including Tri-Global's will not like special characters and even spaces in the field name.
- 2) Label:** Label is what you want your users to see in the field name of the form. Notice in the example above, that I placed a space and an exclamation mark, Neither of these would be acceptable in a field name, but work fine in a label field.
- 3) Info:** Info was created to apply additional information about field. In other words, You might have very specific instructions about how to record a value, but it is too wordy to put in the label. This is why you would use information field. When you put an <Info> tag in your xml file, it will show up as a little i next to the label. You simply click on the i to see the associated information.
- 4) Data:** This provides the format for your database field. MUS can currently only have three types of data fields:
 - a. String:** any text field
 - b. Int:** any integer number (i.e. 56870)
 - c. Double:** any decimal number (i.e. 6.789)

- 5) **Control:** Control represents what the field will do. We have a number of advanced control types listed below:
- a. **Text:** text is pretty self explanatory, it creates a text field in the inspection
 - b. **Combo:** Creates a picklist or combo box for the users.
 - c. **AutoID:** AutoID is very useful for a primary key. It automatically creates a unique ID number, as soon as the form is loaded.
 - d. **Camera:** Stores an image name and path for integrated cameras, such as the Nomad and Juno.
 - e. **Date:** Auto populates with today's date. It is editable and will pop up a calendar.
 - f. **Time:** Auto populates with the current time. It is an editable field.
 - g. **SpatialQuery:** A very useful field for filling in fields with values from a background layer. It currently only works with Polygons and Points. (Lines may work but as of 1/25/2010 are untested.) The field works by clicking on a button. It then fills in the field with data from the underlying shapefile.
 - h. **DKey:** Dicotomous Key, this is a very involved field that will not be described in detail in this tutorial. We recommend consulting with Tri-Global to set these up. However, it is a field that asks a series of questions. The questions will drill down to the answer for the field. Example DKey is available upon request.
 - i. **Search:** Search is essentially a Combo box, but is ideal when there are a huge number of options. Search allows you to type in the value you are looking for and will then narrow down the list of dropdown items.
 - j. **Alias:** Alias is sometimes called "coded" values. This is perfect for either abbreviations or numeric codes for an answer. (i.e. user might see GRN as a dropdown item, but Green is populated in the database.)
 - k. **Slave:** Slave is a field that can be configured when an answer to the field is based on another field. (i.e. if you have the fields City, County and State. County and State can be a slave of City. When you select Athens in City. Clarke and GA will populate in County and State

- 6) **ControlInput**: Control Input is not used on every Control type but various different Controls will require an input:
 - a. **Combo**: ControlInput will equal the values, separated by a semicolon (ex. 1;2;3)
 - b. **SpatialQuery**: Control Input for Spatial Query is a list of the layer, field, and buffer. Separate each item with a semicolon. (Example: layer;field;buffer)
 - c. **DKey**: The control Input is the name of the DKey.xml file.
 - d. **Search**: Looks just like the combo box control input.
 - e. **Alias**: Looks similar to the combo box, but Control Input for Alias is a list of the items and their aliases with an @ symbol between the item name and alias. Separate each item with a semicolon. (Example:
item1@alias1;item2@alias2)
 - f. **Slave**: Control Input for Slave is the name of the Master Control, and the list name .xml file, separated with a semicolon. (Example: City;Counties.xml)

- 7) **Required**: Selecting Yes on this tag will make the item required in the inspection. Use caution on this, because if you select yes, the value will need to be filled out every time.

- 8) **Tab**: An unlimited number of tabs can be set up in MUS. Simply place the tab name you wish the field to show up under in this tag.

- 9) **Master**: Master is much like barcode or primary key and is not necessary for every field. It is only necessary if you are identifying that a combo box field is also the Master of a slave field. You can have one master to multiple slaves, or multiple master and slave combinations. See the example below.

A link to the GenericInspection.xml is provided below. You may download this file and it is ready to be used in MUS Inspection. You can also download the GenericInspection.txt for editing and customization purposes. Open the .txt with a generic word editor like WordPad.

<http://www.triglobal.net/GenericInspection.xml>
(ready for use with MUS Inspection)

<http://www.triglobal.net/GenericInspection.txt>
(ready for editing and customization)

WARNING:

The [Counties.xml](#) prerequisite for the Slave field to work should look like this and be placed in the device's [Scripts directory](#):

```
<Counties>
  <Subitem>
    <Master>Athens</Master>
    <Slave>Clarke</Slave>
  </Subitem>

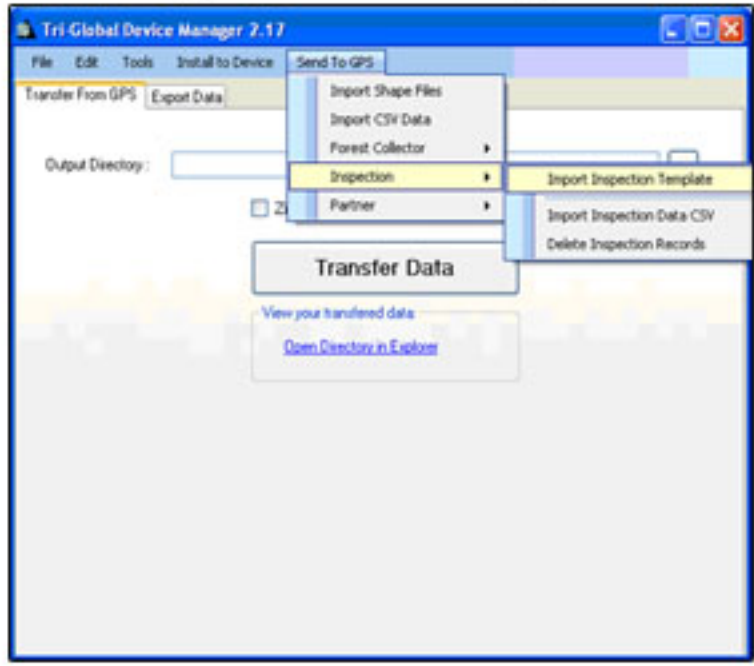
  <Subitem>
    <Master>Winder</Master>
    <Slave>Barrow</Slave>
  </Subitem>

  <Subitem>
    <Master>Atlanta</Master>
    <Slave>Fulton</Slave>
  </Subitem>

  <Subitem>
    <Master>Columbus</Master>
    <Slave>Muscogee</Slave>
  </Subitem>

  <Subitem>
    <Master>Savannah</Master>
    <Slave>Chatham</Slave>
  </Subitem>
</Counties>
```

After Creating the XML file, you will want to import the template into MUS Inspection using TGT Device Manager:



Your Inspection should look like this:

A screenshot of the UniversalTGFForm application. The form contains several input fields: 'AutoID' with value '1c48ae48-b11e', 'Meter Number' with an asterisk, 'City Name' with a dropdown arrow, 'County' with value 'Clarke', and 'Quality' with a checked checkbox. At the bottom, there are tabs for 'Tab1', 'Tab2', and 'Camera', and buttons for 'Add' and 'Close'.A screenshot of the UniversalTGFForm application. The form contains several input fields: 'Notes' with value 'example', 'Date' with value '1/25/10', 'Time' with value '11:41:14 AM', 'Search' with value 'a2' and an asterisk, and 'Alias_Combo' with value 'b' and a dropdown arrow. At the bottom, there are tabs for 'Tab1', 'Tab2', and 'Camera', and buttons for 'Add' and 'Close'.