

Define the fictional language txt

 Find the root directory of ProjectGenerator

open the map templates, where you can see several xml files defining program languages.

create a new file: txt.xml

and copy this content

```
<language>
  <name>txt</name>
  <tablenamefilter>
  </tablenamefilter>
  <dbfunctions>
    <java.lang.String>STRING(":column:")</java.lang.String>
    <long>LONG(":column:")</long>
    <boolean>BOOLEAN(":column:")</boolean>
    <java.sql.Date>DATE(":column:")</java.sql.Date>
    <float>FLOAT(":column:")</float>
    <double>DOUBLE(":column:")</double>
    <int>INTEGER(":column:")</int>
    <java.sql.Time>TIME(":column:")</java.sql.Time>
    <java.sql.Timestamp>TIMESTAMP(":column:")</java.sql.Timestamp>
    <byte>BYTE(":column:")</byte>
    <java.sql.Array>ARRAY(":column:")</java.sql.Array>
    <java.lang.Object>OBJECT(":column:")</java.lang.Object>
    <UNKNOWN>OBJECT(":column:")</UNKNOWN>
  </dbfunctions>
  <Objectcasting>
    <sqlconverter></sqlconverter>
    <datatypes>
    </datatypes>
  </Objectcasting>
</language>
```

This defines the txt language with metacode dbfunctions and no Objectcasting.

Note that tablenamefilter is empty, which indicates that all characters are allowed in the fictional txt language.

This file will be picked up by Metacoder when it starts.