

# IntelliJ Integration

QDL syntax highlighting can be integrated into your IntelliJ IDE and you can use the full power of your IDE's editor. This is *not* a full blown plugin, but a quick way to get you up to speed.

This is actually not hard to do at all (if you slog through enough documentation to find it!) and is mostly clicking a few boxes and pasting a whole bunch of stuff from below.

## 1. Setup the file type.

Open Settings (Ctrl + Alt + s) and navigate to

Editor → File Types

Under **Recognized File Types** click + (to add a new one).

Set the name and description to be QDL and the following (See the screen shot)

Then click on the File

Line comment: //

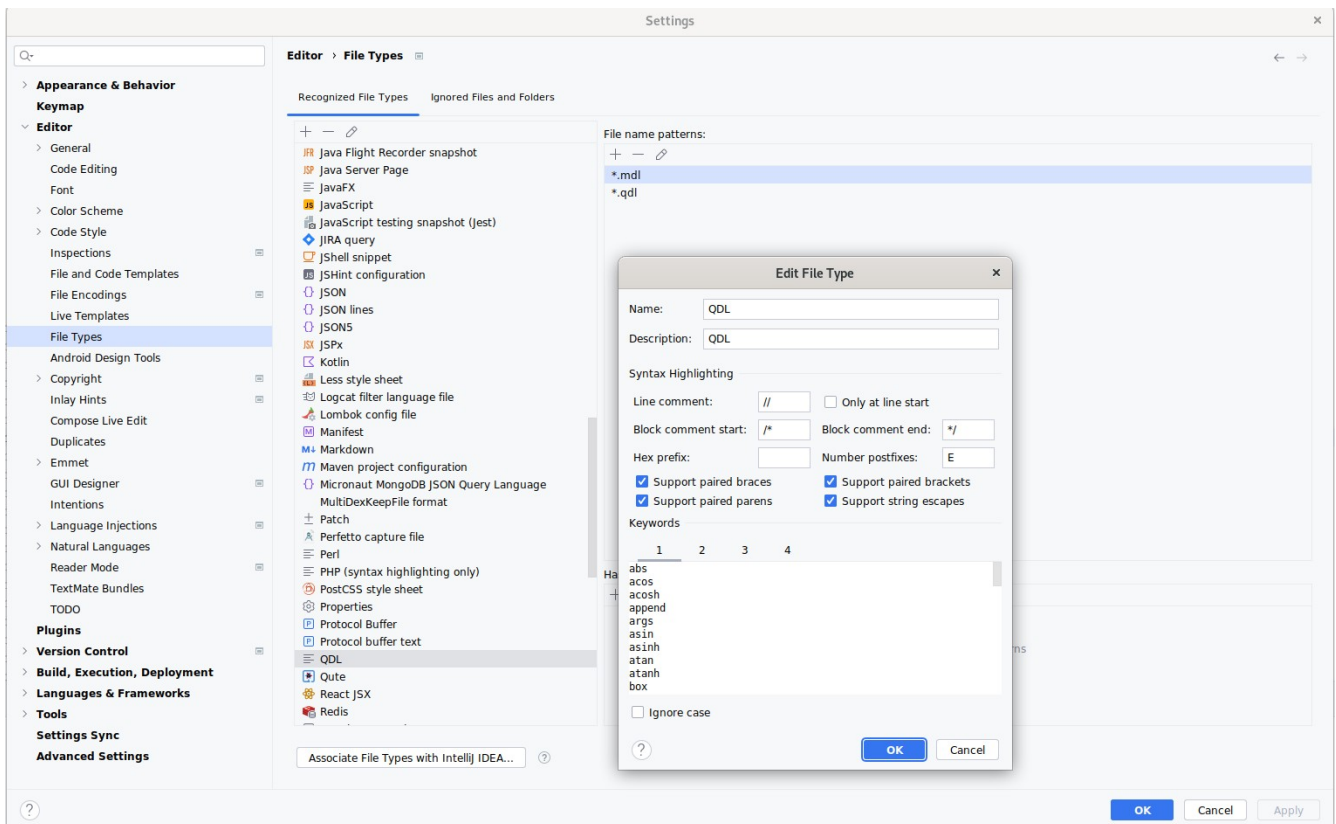
Block Comment Start: /\*

Block Comment End: \*/

Number postfixes: **E**

You want to select all of the next set of boxes for pairing brackets and parentheses.

It should look a lot like the next screen shot:



## 2. Set the syntax highlighting.

Under the boxes you will find a box for **keywords** with 4 tabs. Each tab corresponds to a different highlighting color. The following lists go into the boxes. Remember that each line is a token so line breaks are essential.

### Box 1 (functions)

```
abs
acos
acosh
append
args
asin
asinh
atan
atanh
box
break
cb_exists
cb_read
cb_write
ceiling
```

check\_after  
check\_syntax  
common\_keys  
constants  
contains  
continue  
cos  
cosh  
date\_iso  
date\_ms  
debugger  
decode  
detokenize  
diff  
differ\_at  
dim  
dir  
docs  
encode  
excise  
exclude\_keys  
exp  
expand  
file\_read  
file\_write  
floor  
for\_each  
for\_keys  
for\_lines  
for\_next  
fork  
from\_json  
from\_uri  
funcs  
gcd  
halt  
has\_key  
has\_keys  
has\_value  
hash  
head  
identity  
import  
include\_keys  
index\_of  
indices  
info  
input\_form  
insert  
insert\_at

interpret  
is\_defined  
is\_function  
is\_list  
is\_null  
j\_load  
j\_use  
join  
keys  
kill  
lcm  
lib\_entries  
list\_copy  
list\_keys  
ln  
load  
loaded  
log  
logger  
mask  
max  
min  
mod  
module\_import  
module\_load  
module\_path  
module\_remove  
names  
nroot  
numeric\_digits  
os\_env  
pi  
pick  
print  
query  
raise\_error  
random  
random\_string  
rank  
reduce  
remap  
remove  
rename  
rename\_keys  
replace  
return  
reverse  
say  
scan  
script\_args

script\_load  
script\_name  
script\_path  
script\_run  
set\_default  
shuffle  
sin  
sinh  
size  
sleep  
sort  
star  
starts\_with  
sublist  
substring  
tail  
tan  
tanh  
to\_boolean  
to\_json  
to\_lower  
to\_number  
to\_string  
to\_upper  
to\_uri  
tokenize  
transpose  
trim  
unbox  
union  
unique  
unload  
use  
values  
var\_type  
vars  
vfs\_mount  
vfs\_unmount  
ws\_macro  
n

## **Box 2. Reserved QDL keywords.**

assert  
block  
body  
define  
do

```

else
if
module
switch
then
while
∅
⊢

```

### Box 3. Unicode operators

@  
É  
Ð  
À  
Ä  
µ  
©  
⊕  
∃  
∂  
∈  
∉  
<<  
┐  
└<sup>^</sup>  
└  
└  
└  
└  
∩  
∧  
~  
|  
+  
\*  
×  
/  
÷  
%  
Δ  
+  
+  
-  
-  
  
&&  
∧  
||  
v  
≡  
≠  
≤

≥  
!  
┐  
≈  
∂  
[ |  
[[  
| ]  
]]

#### Box 4. Standard Math operators (ASCII)

~  
!  
#  
%  
&  
\*  
+  
-  
<  
=  
>  
?  
@  
^  
|  
~  
┐  
┌  
  
ˆ  
×  
÷  
/  
\

### 3. Set the file pattern

Close the syntax box and add two file patterns, one for QDL files and one for modules files:

\*.qdl

\*.mdl

You should now be able to create a file with the .qdl extension and all the syntax is highlighted, brackets, braces matched etc.