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: **//**

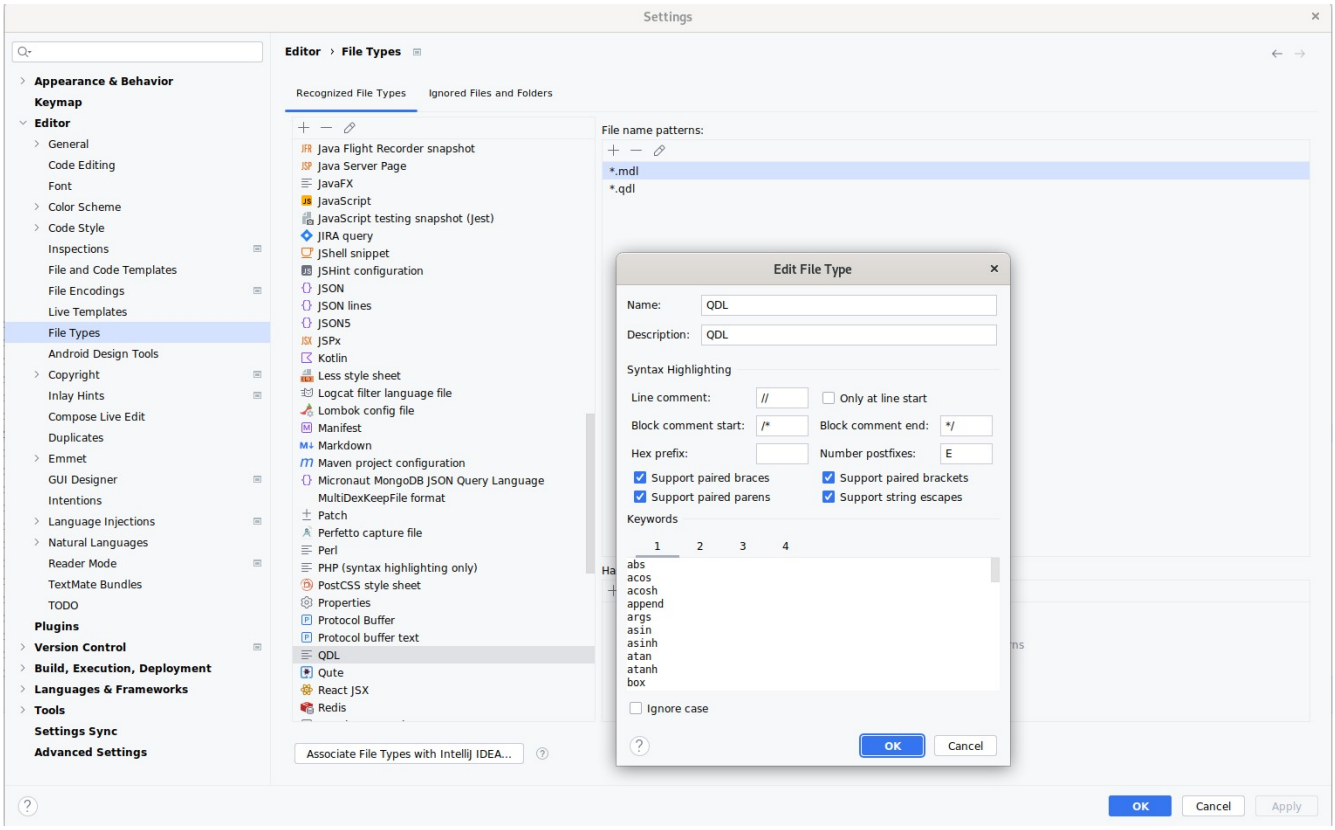
Block Comment Start: **/***

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

Box 1 (functions)

```
abs
acos
acosh
apply
arg_count
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

```
@
∋
∌
∀
∓
µ
⊙
⊕
∃
∄
∈
∉
<<
⊢
|^
⌊
⌈
∪
∩
^
~|
↛
*
×
/
÷
%
Δ
+
⁺
-
−

&&
∧
||
∨
≡
≠
≤
```

≥
!
⌐
≈
∂
[|
⟦
|]
⟧

Box 4. Standard Math operators (ASCII)

~
!
#
%
&
*
+
-
<
=
>
?
@
^
|
~
¬
¿
×
÷
/
\

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.