Subject: To advance using int64 in ESC
Posted by hans on Fri, 19 Sep 2014 18:24:49 GMT
View Forum Message <> Reply to Message

Hi,
as recently the internal representation of number was optimized for int64,
I have taken the ball and added optimized parsing and done a bit more of internal optimization.

First the bugfixes in EscValue::ToString() function.
If the number has the value of INT64_MIN it is not printed because FormatInt64() treats as Null.
But it should, as number has other value for Null.

Change

```
   if((int64)number == number)
    s << FormatInt64((int64)number);
```
to
```
   int64 in= (int64)number;
   if(in == number)
    s << String(in < 0 ? "-" + Format64(-in) : Format64(in));
```

and below

```
   s << FormatInt64(i64);
```
to
```
   s << String(i64 < 0 ? "-" + Format64(-i64) : Format64(i64));
```

Now my changes.

The optimized parser will first detect if it is a double (length and . or exponent) and if not to parse the string with the uint64 parser.

This works also if it really is a double (ie. string length == 18 but outside int64 range). Caveat is it will get slightly different double values than with the double parser.

The EscValue::IsInt[64]() / EscValue::GetInt[64] sequences are replaced with one function, named GetInt[64]Only.

If some dev will take my changes as inspiration to get them into U++ it would be great.

I have attached the zip file of my sources.
When diffing please ignore String/WString differences.

If you have questions about my code please ask.

Thanks.

File Attachments

1) toUpp.zip, downloaded 519 times