Subject: Re: [DISCUSSION] Add 'complex' datatype, to Value too
Posted by kohait00 on Mon, 20 Jun 2011 12:24:06 GMT
View Forum Message <> Reply to Message

as a starting point:

```
#include <complex>

NAMESPACE_UPP

typedef std::complex<double> cdouble;

template<> inline bool IsNull(const cdouble& r) { return r.real() < DOUBLE_NULL_LIM || r.imag() < DOUBLE_NULL_LIM; }
template<> inline void SetNull(cdouble& x) { x = cdouble(DOUBLE_NULL, DOUBLE_NULL); }
inline const cdouble& Nvl(const cdouble& a, const cdouble& b)  { return IsNull(a) ? b : a; }

const dword COMPLEX_V   = 20;
template<> inline dword ValueTypeNo(const cdouble*)   { return COMPLEX_V; }

//VALUE_COMPARE(cdouble), doesnt work since Value has no native cdouble conversion support, TODO
inline bool operator==(const Value& v, cdouble x)   { return RichValue<cdouble>::Extract(v) == x; }
inline bool operator==(cdouble x, const Value& v)   { return RichValue<cdouble>::Extract(v) == x; }
inline bool operator!=(const Value& v, cdouble x)   { return RichValue<cdouble>::Extract(v) != x; }
inline bool operator!=(cdouble x, const Value& v)   { return RichValue<cdouble>::Extract(v) != x; }

template<> inline unsigned GetHashValue(const cdouble& x) { return CombineHash(x.real(), x.imag()); }
template<> inline String AsString(const cdouble& x) { return String().Cat() << "C(" << x.real() << "," << x.imag() << ")"; }
template<> inline Stream& operator%(Stream& s, cdouble& x) {
 double r,i;
 if(s.IsStoring()) { r = x.real(); i = x.imag(); }
 s % r % i;
 if(s.IsLoading()) { x = cdouble(r,i); }
 return s;
}

END_UPP_NAMESPACE
```