How to use Msgpack 😱

Msgpack Reading

- Read all the content of input file in binary mode
  - char[] , stringstream
- Using **unpack** to get the object handler
  - client controls a buffer
  - unpacker control
- Using **get** & **convert** to get the standard data type (Ex: int , vector<int>)
- unpack(const char* data, size_t len, size_t offset)
- cpp_unpacker

Msgpack Writing

- Using **pack** to serialize data type into buffer (sbuffer or stringstream)
- Write the sbuffer to the output file

Msgpack official example

```
// deserialize the buffer into msgpack::object instance.
std::string str(buffer.str());  //buffer is sstream
msgpack::object_handle oh =
msgpack::unpack(str.data(), str.size());

// deserialized object is valid during the msgpack::object_
msgpack::object deserialized = oh.get();

// convert msgpack::object instance into the original type.
msgpack::type::tuple<int, bool, std::string> dst;
deserialized.convert(dst);
```

Msgpack example

```
vector<int> arr;
//reading data
size_t off = 0;
oh = msgpack::unpack(buf,sizebuf,off);
oh.get().convert(arr);
cout<<arr.size(); //array size
//optional
int *a = &arr[0];
int len = arr.size();

//ouput data
fstream fout;
fout.open("test.txt" , ios::binary | ios::out);
msgpack::sbuffer sbuf;
msgpack::pack(sbuf, num);
fout.write(sbuf.data() , sbuf.size() );
```

Msgpack type

C++ Type	msgpack::object type
bool	bool
int	integer
std::vector	array
std::string	str
std::list	array

https://github.com/msgpack/msgpack-c/wiki/v2_0_cpp_adaptor

Python msgpack

```
with open('input.txt', 'rb') as fin:
        unp = msgpack.Unpacker(fin)
        arr = unp.unpack();
with open('output.txt', 'rb') as fout:
        fout.write(msgpackb(arr))
```

😮

Q&A ❓

https://github.com/msgpack/msgpack-c/wiki/