ILDG Binary File Format
(Rev. 1.1)

ILDG Metadata Working Group

15 December 2005

This document specifies the file format which is used to exchange binary files within ILDG.

1 Overview

For sharing binary files which contain, e.g., gauge field configurations, ILDG adopted the strategy of defining a real file format. All providers of configurations have to provide their configurations using this file format. The goal was to to define a structured file format which allows to encapsulate, in an extensible way, binary data and meta data within a single file. While currently only exchange of binary files containing gauge field configurations is foreseen, in future also other kind of binary data, e.g. different kind of propagators, may be exchanged.

2 Specification

2.1 Packaging

The ILDG binary file format consists of several parts which are packaged using the LIME file format. LIME stands for "Lattice QCD Interchange Message Encapsulation" and has been developed by SciDAC [1]. It allows to encapsulate one or more *records* containing ASCII or binary data One or more records can be packaged into one *message*. A LIME file may consist of several messages.

2.2 File structure

The ILDG binary file has the following structure:[1]

[1]Numbering of messages and records is 1, 2, ...

message	record	LIME record type
#1	...	...
...	...	...
#n	...	...
	#i	ildg-format
	...	...
	#j	ildg-binary-data
	...	...
...	...	...
#m	#1	ildg-data-lfn
...	...	...

Where

- Groups are free to introduce their own messages/records and insert them in the locations indicated by '...'.
- The records/messages indicated by '...' may not exist.
- The first part of the LIME type strings, i.e. all characters until the first dash ('-'), identifies a namespace.
- Each group introducing their own records should have a unique namespace. These namespaces should be registered with ILDG to avoid name clashes.
- The order of the messages `#n` and `#m` <u>is not fixed</u>, i.e. both `#n<#m` and `#m<#n` are allowed.
- The record `ildg-format` has to precede the record `ildg-binary-data`, the order of the two records <u>is fixed</u> to `#i<#j`.
- Matching of LIME record times is performed case sensitive.

2.3 Record `ildg-format`

This record consists of a XML document which contains a minimal set of non-mutable parameters needed to read the binary data. The document has to conform to the schema provided in the appendix. Here an example document:

```
<?xml version="1.0" encoding="UTF-8"?>
<ildgFormat xmlns="http://www.lqcd.org/ildg"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://www.lqcd.org/ildg/filefmt.xsd">
  <version> 1.0 </version>
  <field> su3gauge </field>
  <precision> 32 </precision>
  <lx> 20 </lx> <ly> 20 </ly> <lz> 20 </lz> <lt> 64 </lt>
</ildgFormat>
```

2.4 Record `ildg-binary-data`

This record may contain different kind of fields or other data. Currently, only storing SU(3) gauge fields is foreseen. The record consists of a sequence of IEEE floating point numbers. The precision is defined in the `ildg-format` record. The endianess is fixed to big.

2.4.1 SU(3) gauge configurations

A SU(3) gauge configuration is a set of SU(3) matrices assigned to the links of a four dimensional hypercube. If $x(n)$ is a value of a quark field at site $n = (n_1, n_2, n_3, n_4)$,

$$x^\dagger(n)U_\mu(n)x(n+\hat{\mu}) \equiv \sum_{a,b=1}^{3} [x^\dagger(n)]_a [U_\mu(n)]_{ab} [x(n+\hat{\mu})]_b \tag{1}$$

is gauge invariant.

The configuration is stored as 8 (or 7) dimensional array of floating point numbers (or complex numbers). The dimensions ordered from slowest to fastest running index are (in brackets: C-style index):

1. Site index in time-direction t (`t=0,...,NT-1`).
2. Site index in space-direction z (`z=0,...,NZ-1`).
3. Site index in space-direction y (`y=0,...,NY-1`).
4. Site index in space-direction x (`x=0,...,NX-1`).
5. Direction index μ (`mu=0,...,3`, where `0`$\leftrightarrow x$, `1`$\leftrightarrow y$, `2`$\leftrightarrow z$, `3`$\leftrightarrow t$).
6. Colour index a (`a=0, 1, 2`).
7. Colour index b (`b=0, 1, 2`).
8. [Index referencing real (0)/imaginary (1)part (`ir=0, 1`).]

The corresponding definition of such an array in C (i.e. the rightmost index runs fastest) is:

```
double U[NT][NZ][NY][NX][NDIMENSION][NCOLOR][NCOLOR][2];
```

For FORTRAN (i.e. the leftmost index runs fastest) the equivalent statement is:

```
complex U(NCOLOR,NCOLOR,NDIMENSION,NX,NY,NZ,NT)
```

2.5 Record `ildg-data-lfn`

This record contains a single string with the logical filename (LFN) used for storing the binary file in the Grid. The content must be identical to the element `<dataLFN>` of the corresponding configuration metadata document.

Appendix

2.6 Schema for `ildg-format` document

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.lqcd.org/ildg"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://www.lqcd.org/ildg"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified">
  <xs:simpleType name="fieldType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="su3gauge"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="precisionType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="32"/>
      <xs:enumeration value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="ildgFormat">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="version" type="xs:string"/>
        <xs:element name="field" type="fieldType"/>
        <xs:element name="precision" type="precisionType"/>
        <xs:element name="lx" type="xs:integer"/>
        <xs:element name="ly" type="xs:integer"/>
        <xs:element name="lz" type="xs:integer"/>
        <xs:element name="lt" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

References

[1] SciDAC Software Coordinating Committee, “LIME (Version 1.1),” http://www.physics.utah.edu/~detar/scidac