The XIM Transport Specification

Revision 0.1

Takashi Fujiwara, FUJITSU LIMITED

The XIM Transport Specification: Revision 0.1

©

Abstract

Table of Contents

List of Tables

Chapter 1. X Transport Specification

Introduction

Initialization

Registering structure to initialize

```
typedef struct {
      char *transport_name;
      Bool (*config);
} TransportSW;
```

```
TransportSW _XimTransportRec[] = {
/*     char *:
 *     transport_name,     Bool (*config)()
 */
```

```
        "X",              _XimXConf,
        "tcp",            _XimTransConf,
        "local",          _XimTransConf,
        "decnet",         _XimTransConf,
        "streams",        _XimTransConf,
        (char *)NULL,     (Bool (*)())NULL,
};
```

Initialization function

```
Bool (*config)( im,  *transport_data);
```

_XimTransportRec

```
Bool (*connect)();                /* Open connection */
Bool (*shutdown)();               /* Close connection */
Bool (*write)();                  /* Write data */
Bool (*read)();                   /* Read data */
Bool (*flush)();                  /* Flush data buffer */
Bool (*register_dispatcher)();    /* Register asynchronous data handler */
Bool (*call_dispatcher)();        /* Call dispatcher */
```

The interface/transport layer functions

Opening connection

XOpenIM

```
Bool (*connect)( im);
```

```
Bool  _XimConnect( im);
```

Closing connection

XCloseIM

```
Bool  (*shutdown)( im);
```

```
Bool _XimShutdown( im);
```

Writing data

```
Bool  _XimWrite( im,  len,  data);
```

```
Bool _XimWrite( im,  len,  data);
```

Reading data

```
Bool  _XimRead( im,  read_buf,  buf_len,  *ret_len);
```

```
Bool  _XimRead( im,  *ret_len,  buf,  buf_len,  (*predicate)(),
predicate_arg);
```

```
void (*predicate)( im, len, data, predicate_arg);
```

Flushing buffer

```
void (*flush)( im);
```

```
void _XimFlush( im);
```

Registering asynchronous data handler

_XimCallDispatcher

```
Bool (*register_dispatcher)( im, (*dispatcher)(), call_data);
```

```
Bool (*dispatcher)( im, len, data, call_data);
```

```
Bool  _XimRegisterDispatcher( im,  (*dispatcher)(),  call_data);
```

Calling dispatcher

```
Bool (*call_dispatcher)( im,  len,  data);
```

```
Bool  _XimCallDispatcher( im,  len,  call_data);
```

Sample implementations for the Transport Layer

X Transport

Connection


```
only-CM            :    data is sent via a ClientMessage
multi-CM           :    data is sent via multiple ClientMessages
Property-with-CM   :    data is written in Property, and its Atom
                         is send via ClientMessage
PropertyNotify     :    data is written in Property, and its Atom
                         is send via PropertyNotify
```

read/write

Format for the data from the Client to the IM Server

XInternAtom

Format for the data from the IM Server to the Client

`XInternAtom`

Closing Connection

References