

# XEP-0096: SI File Transfer

**Abstract:** This specification defines a profile of the XMPP stream initiation extension for transferring files between two entities. The protocol provides a modular framework that enables the exchange of information about the file to be transferred as well as the negotiation of parameters such as the transport to be used.

**Authors:** Thomas Muldowney, Matthew Miller, Ryan Eatmon, Peter Saint-Andre

**Copyright:** © 1999 - 2016 XMPP Standards Foundation. [SEE LEGAL NOTICES](#).

**Status:** Draft

**Type:** Standards Track

**Version:** 1.2

**Last Updated:** 2004-04-13

---

NOTICE: The protocol defined herein is a **Draft Standard** of the XMPP Standards Foundation. Implementations are encouraged and the protocol is appropriate for deployment in production systems, but some changes to the protocol are possible before it becomes a Final Standard.

---

## Table of Contents

1. [Introduction](#)
  2. [Requirements](#)
  3. [Protocol](#)
    - 3.1. [Mandatory-to-Implement Technologies](#)
  4. [Examples](#)
  5. [IANA Considerations](#)
  6. [XMPP Registrar Considerations](#)
    - 6.1. [Stream Initiation Profiles](#)
    - 6.2. [URI Query Types](#)
      - 6.2.1. [sendfile](#)
      - 6.2.2. [recvfile](#)
  7. [XML Schema](#)
- [Appendices](#)
- [A: Document Information](#)
  - [B: Author Information](#)
  - [C: Legal Notices](#)
  - [D: Relation to XMPP](#)
  - [E: Discussion Venue](#)
  - [F: Requirements Conformance](#)
  - [G: Notes](#)
  - [H: Revision History](#)
- 

## 1. Introduction

The traditional mechanism for transferring files in the Jabber community is the [Out-of-Band Data \(XEP-0066\)](#) [1] protocol. That protocol has several drawbacks:

1. It is not reliable.
2. It does not work when one of the parties is behind a firewall.
3. It provides limited metadata about files to be exchanged.

The current document defines a profile of [Stream Initiation \(XEP-0095\)](#) [2] that solves the problems with out-of-band data, thus providing a robust, reliable mechanism for file transfers over the Jabber network. Implementors are referred to XEP-0095 regarding the underlying concepts of stream initiation.

## 2. Requirements

- Enable seamless file transfer, including fall-back mechanisms as appropriate.
- Ensure that the profile will work even when one or both parties are behind a firewall.
- Define a full-featured set of metadata for file transfers, including the following:
  - description
  - size
  - name
  - date
  - hash
- Optionally support ranged transfers.

## 3. Protocol

The file transfer profile is in the "http://jabber.org/protocol/si/profile/file-transfer" namespace. The profile is fairly simple: it consists of the root element with the possibility of one child describing the optional ranged transfers.

The root element is <file> and has four attributes. The attributes are used only during the offer stage of stream initiation:

- *size* - The size, in bytes, of the data to be sent.
- *name* - The name of the file that the Sender wishes to send.

- *date* - The last modification time of the file. This is specified using the DateTime profile as described in [XMPP Date and Time Profiles \(XEP-0082\)](#) [3].
- *hash* - The MD5 sum of the file contents.

The *size* and *name* attributes MUST be present in the profile. The other attributes MAY be present.

There are two possible child elements of the root: <desc> and <range>. Both are OPTIONAL.

<desc> is used to provide a sender-generated description of the file so the receiver can better understand what is being sent. It MUST NOT be sent in the result.

When <range> is sent in the offer, it should have no attributes. This signifies that the sender can do ranged transfers. When a Stream Initiation result is sent with the <range> element, it uses these attributes:

- *offset* - Specifies the position, in bytes, to start transferring the file data from. This defaults to zero (0) if not specified.
- *length* - Specifies the number of bytes to retrieve starting at offset. This defaults to the length of the file from offset to the end.

Both attributes are OPTIONAL on the <range> element. Sending no attributes is synonymous with not sending the <range> element. When no <range> element is sent in the Stream Initiation result, the Sender MUST send the complete file starting at offset 0. More generally, data is sent over the stream byte for byte starting at the offset position for the length specified.

### 3.1 Mandatory-to-Implement Technologies

In order to enable seamless file transfer and appropriate fall-back mechanisms, implementations of this profile MUST support both [SOCKS5 Bytestreams \(XEP-0065\)](#) [4] and [In-Band Bytestreams \(XEP-0047\)](#) [5], to be preferred in that order. The associated namespaces are to be included as option values for the "stream-method" variable as shown in the examples below.

Additionally, implementations MAY support other mechanisms.

## 4. Examples

### Example 1. Simple Profile Usage in Stream Initiation Offer

```
<iq type='set' id='offer1' to='receiver@jabber.org/resource'>
  <si xmlns='http://jabber.org/protocol/si'
    id='a0'
    mime-type='text/plain'
    profile='http://jabber.org/protocol/si/profile/file-transfer'>
    <file xmlns='http://jabber.org/protocol/si/profile/file-transfer'
      name='test.txt'
      size='1022' />
    <feature xmlns='http://jabber.org/protocol/feature-neg'>
      <x xmlns='jabber:x:data' type='form'>
        <field var='stream-method' type='list-single'>
          <option><value>http://jabber.org/protocol/bytestreams</value></option>
          <option><value>http://jabber.org/protocol/ibb</value></option>
        </field>
      </x>
    </feature>
  </si>
</iq>
```

### Example 2. Simple Profile Usage in Stream Initiation Result

```
<iq type='result' to='sender@jabber.org/resource' id='offer1'>
  <si xmlns='http://jabber.org/protocol/si'>
    <feature xmlns='http://jabber.org/protocol/feature-neg'>
      <x xmlns='jabber:x:data' type='submit'>
        <field var='stream-method'>
          <value>http://jabber.org/protocol/bytestreams</value>
        </field>
      </x>
    </feature>
  </si>
</iq>
```

### Example 3. Complete Profile Usage in Stream Initiation Offer

```
<iq type='set' id='offer1' to='receiver@jabber.org/resource'>
  <si xmlns='http://jabber.org/protocol/si'
    id='a0'
    mime-type='text/plain'
    profile='http://jabber.org/protocol/si/profile/file-transfer'>
    <file xmlns='http://jabber.org/protocol/si/profile/file-transfer'
```

```

        name='test.txt'
        size='1022'
        hash='552da749930852c69ae5d2141d3766b1'
        date='1969-07-21T02:56:15Z'>
    <desc>This is a test. If this were a real file...</desc>
</file>
<feature xmlns='http://jabber.org/protocol/feature-neg'>
  <x xmlns='jabber:x:data' type='form'>
    <field var='stream-method' type='list-single'>
      <option><value>http://jabber.org/protocol/bytestreams</value></option>
      <option><value>http://jabber.org/protocol/ibb</value></option>
    </field>
  </x>
</feature>
</si>
</iq>

```

#### Example 4. Complete Profile Usage in Stream Initiation Result

```

<iq type='result' to='sender@jabber.org/resource' id='offer1'>
  <si xmlns='http://jabber.org/protocol/si'>
    <file xmlns='http://jabber.org/protocol/si/profile/file-transfer'/>
    <feature xmlns='http://jabber.org/protocol/feature-neg'>
      <x xmlns='jabber:x:data' type='submit'>
        <field var='stream-method'>
          <value>http://jabber.org/protocol/bytestreams</value>
        </field>
      </x>
    </feature>
  </si>
</iq>

```

This range should retrieve 256 bytes from the beginning of the file:

#### Example 5.

```

<range length='256' />

```

This range should retrieve 256 bytes starting from the 128th byte in the file:

#### Example 6.

```

<range offset='128' length='256' />

```

This range should retrieve the remainder of the file starting at the 128th byte in the file:

#### Example 7.

```

<range offset='128' />

```

This range is the same as having not sent the range request and the entire file is sent:

#### Example 8.

```

<range />

```

## 5. IANA Considerations

No interaction with the [Internet Assigned Numbers Authority \(IANA\)](#) [6] is required as a result of this document.

## 6. XMPP Registrar Considerations

### 6.1 Stream Initiation Profiles

The profile described in this document is included in the stream initiation profiles registry maintained by the [XMPP](#)

**Registrar** [7] (see <<http://xmpp.org/registrar/si-profiles.html>>). The registry submission is as follows:

```
<profile>
  <name>http://jabber.org/protocol/si/profile/file-transfer</name>
  <doc>XEP-0096</doc>
  <desc>A profile for file transfer between any two entities.</desc>
</profile>
```

## 6.2 URI Query Types

As authorized by **XMPP URI Query Components (XEP-0147)** [8], the XMPP Registrar maintains a registry of queries and key-value pairs for use in XMPP URIs (see <<http://xmpp.org/registrar/querytypes.html>>).

As described below, the registered querytypes for file transfer actions are "sendfile" and "recvfile". Note well that "sendfile" means a second entity will send a file to the XMPP entity that controls the IRI/URI and that "recvfile" means a second entity will receive a file from the XMPP entity that controls the IRI/URI.

### 6.2.1 sendfile

To enable a second entity to send a file, the IRI/URI is of the following form:

#### Example 9. Sending a File: IRI/URI

```
xmpp:romeo@montague.net/orchard?sendfile
```

The application SHOULD then present an interface enabling the user to provide information about the file to be sent (e.g., by "browsing" the file system of the user's computer in order to choose a file). As a result, the application SHOULD then send a **Publishing Stream Initiation Requests (XEP-0137)** [9] message to the XMPP address encapsulated in the IRI/URI:

#### Example 10. Sending a File: Resulting Stanza

```
<message from='juliet@capulet.com/balcony' to='romeo@montague.net'>
  <si:pub xmlns='http://jabber.org/protocol/si-pub'
    id='publish-0123'
    mime-type='text/plain'
    profile='http://jabber.org/protocol/si/profile/file-transfer'>
    <file xmlns='http://jabber.org/protocol/si/profile/file-transfer'
      name='missive.txt'
      size='1024'
      date='2005-11-29T11:21Z' />
    </si:pub>
  </message>
```

The following submission registers the "sendfile" querytype.

```
<querytype>
  <name>sendfile</name>
  <proto>http://jabber.org/protocol/si/profile/file-transfer</proto>
  <desc>enables initiation of an inbound file transfer to XMPP entity</desc>
  <doc>XEP-0096</doc>
</querytype>
```

### 6.2.2 recvfile

To enable a second entity to receive a file, the IRI/URI is of the following form:

#### Example 11. Receiving a File: IRI/URI

```
xmpp:romeo@montague.net/orchard?recvfile;sid=pub234;mime-type=text%2Fplain;name=reply.txt;size=2048
```

That IRI/URI is equivalent to the following XML stanza:

#### Example 12. Receiving a File: Equivalent Stanza

```
<message from='romeo@montague.net' to='juliet@capulet.com/balcony'>
  <si:pub xmlns='http://jabber.org/protocol/si-pub'
    id='pub234'
    mime-type='text/plain'
    profile='http://jabber.org/protocol/si/profile/file-transfer'>
    <file xmlns='http://jabber.org/protocol/si/profile/file-transfer'
      name='reply.txt'
      size='2048' />
    </si:pub>
  </message>
```

In accordance with **XEP-0137**, the application SHOULD then initiate a file transfer exchange with by

in accordance with **XEP-0157**, the application SHOULD then initiate a file transfer exchange with by sending a stanza of the following form:

### Example 13. Receiving a File: Resulting Stanza

```
<iq from='juliet@capulet.com/balcony' to='romeo@montague.net/orchard'>
  <start xmlns='http://jabber.org/protocol/si-pub' id='pub234' />
</iq>
```

Note well that the request to begin the stream is sent to the full JID (user@host/resource) of the XMPP entity identified by the XMPP IRI/URI. Therefore, the IRI/URI SHOULD include a full JID. If it does not, the receiver MUST discover a full JID via presence or service discovery. If the receiver cannot discover a full JID for the sender (e.g., in the last resort through sending a presence subscription request to the sender and receiving presence from the sender's resources), then it SHOULD abort the file transfer exchange.

The following submission registers the "recvfile" querytype.

```
<querytype>
  <name>recvfile</name>
  <proto>http://jabber.org/protocol/si/profile/file-transfer</proto>
  <desc>enables initiation of an outbound file transfer from XMPP entity</desc>
  <doc>XEP-0096</doc>
  <keys>
    <key>
      <name>algo</name>
      <desc>the hash algorithm used to generate the checksum</desc>
    </key>
    <key>
      <name>hash</name>
      <desc>a checksum of the file contents</desc>
    </key>
    <key>
      <name>mime-type</name>
      <desc>the MIME type of the file being offered</desc>
    </key>
    <key>
      <name>name</name>
      <desc>the name of the file being offered</desc>
    </key>
    <key>
      <name>sid</name>
      <desc>the session ID associated with the file being offered</desc>
    </key>
    <key>
      <name>size</name>
      <desc>the size in bytes of the file being offered</desc>
    </key>
  </keys>
</querytype>
```

## 7. XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='http://jabber.org/protocol/si/profile/file-transfer'
  xmlns='http://jabber.org/protocol/si/profile/file-transfer'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0096: http://www.xmpp.org/extensions/xep-0096.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='file'>
    <xs:complexType>
      <xs:sequence minOccurs='0'>
        <xs:element name='desc' type='xs:string' />
        <xs:element ref='range' />
      </xs:sequence>
      <xs:attribute name='date' type='xs:string' use='optional' />
      <xs:attribute name='hash' type='xs:string' use='optional' />
      <xs:attribute name='name' type='xs:string' use='required' />
      <xs:attribute name='size' type='xs:integer' use='required' />
    </xs:complexType>
  </xs:element>
```

```

</xs:element>

<xs:element name='range'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='length' type='xs:integer' use='optional' />
        <xs:attribute name='offset' type='xs:integer' use='optional' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:simpleType name='empty'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='' />
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

---

## Appendices

### Appendix A: Document Information

Series: [XEP](#)  
 Number: 0096  
 Publisher: [XMPP Standards Foundation](#)  
 Status: [Draft](#)  
 Type: [Standards Track](#)  
 Version: 1.2  
 Last Updated: 2004-04-13  
 Approving Body: [XMPP Council](#)  
 Dependencies: XMPP Core, XEP-0095  
 Supersedes: None  
 Superseded By: None  
 Short Name: si-filetransfer  
 Schema: <<http://www.xmpp.org/schemas/file-transfer.xsd>>  
 Source Control: [HTML](#)  
 This document in other formats: [XML](#) [PDF](#)

---

### Appendix B: Author Information

#### Thomas Muldowney

Email: [temas@jabber.org](mailto:temas@jabber.org)  
 JabberID: [temas@jabber.org](mailto:temas@jabber.org)

#### Matthew Miller

Email: [linuxwolf@outer-planes.net](mailto:linuxwolf@outer-planes.net)  
 JabberID: [linuxwolf@outer-planes.net](mailto:linuxwolf@outer-planes.net)

#### Ryan Eatmon

Email: [reatmon@jabber.org](mailto:reatmon@jabber.org)  
 JabberID: [reatmon@jabber.org](mailto:reatmon@jabber.org)

#### Peter Saint-Andre

Email: [peter@andyet.net](mailto:peter@andyet.net)  
 JabberID: [stpeter@stpeter.im](mailto:stpeter@stpeter.im)  
 URI: <https://stpeter.im/>

---

### Appendix C: Legal Notices

#### Copyright

This XMPP Extension Protocol is copyright © 1999 - 2016 by the [XMPP Standards Foundation](#) (XSF).

#### Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

### Disclaimer of Warranty

**## NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. ##**

### Limitation of Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

### IPR Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <<http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/>> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

---

## Appendix D: Relation to XMPP

The Extensible Messaging and Presence Protocol (XMPP) is defined in the XMPP Core (RFC 6120) and XMPP IM (RFC 6121) specifications contributed by the XMPP Standards Foundation to the Internet Standards Process, which is managed by the Internet Engineering Task Force in accordance with RFC 2026. Any protocol defined in this document has been developed outside the Internet Standards Process and is to be understood as an extension to XMPP rather than as an evolution, development, or modification of XMPP itself.

---

## Appendix E: Discussion Venue

The primary venue for discussion of XMPP Extension Protocols is the <[standards@xmpp.org](mailto:standards@xmpp.org)> discussion list.

Discussion on other xmpp.org discussion lists might also be appropriate; see <<http://xmpp.org/about/discuss.shtml>> for a complete list.

Errata can be sent to <[editor@xmpp.org](mailto:editor@xmpp.org)>.

---

## Appendix F: Requirements Conformance

The following requirements keywords as used in this document are to be interpreted as described in [RFC 2119](#): "MUST", "SHALL", "REQUIRED"; "MUST NOT", "SHALL NOT"; "SHOULD", "RECOMMENDED"; "SHOULD NOT", "NOT RECOMMENDED"; "MAY", "OPTIONAL".

---

## Appendix G: Notes

1. XEP-0066: Out of Band Data <<http://xmpp.org/extensions/xep-0066.html>>.
2. XEP-0095: Stream Initiation <<http://xmpp.org/extensions/xep-0095.html>>.
3. XEP-0082: XMPP Date and Time Profiles <<http://xmpp.org/extensions/xep-0082.html>>.
4. XEP-0065: SOCKS5 Bytestreams <<http://xmpp.org/extensions/xep-0065.html>>.
5. XEP-0047: In-Band Bytestreams <<http://xmpp.org/extensions/xep-0047.html>>.
6. The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <<http://www.iana.org/>>.
7. The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <<http://xmpp.org/registrar/>>.
8. XEP-0147: XMPP URI Query Components <<http://xmpp.org/extensions/xep-0147.html>>.
9. XEP-0137: Publishing Stream Initiation Requests <<http://xmpp.org/extensions/xep-0137.html>>.

---

## Appendix H: Revision History

Note: Older versions of this specification might be available at <http://xmpp.org/extensions/attic/>

### Version 1.2 (2004-04-13)

More fully defined the XMPP Registrar considerations. (psa)

### Version 1.1 (2003-12-30)

Improved explanatory text; fixed several errors in the schema. (psa)

**Version 1.0 (2003-10-17)**

Per a vote of the Jabber Council, advanced status to Draft. (psa)

**Version 0.7 (2003-10-07)**

Added IBB as a MUST requirement. (tjm)

**Version 0.6 (2003-08-18)**

Cleaned up some namespace inconsistencies, added the <desc> element for file descriptions. (tjm)

**Version 0.5 (2003-07-15)**

Stream ids not needed on return results. Moved s5b, ibb, and url-data to the actual namespaces of the stream protocols. (rwe)

**Version 0.4 (2003-06-30)**

Fixed various typos and inconsistencies (lw)

**Version 0.3 (2003-06-30)**

Added XML Schema (lw)

**Version 0.2 (2003-06-24)**

Clarified many examples, added linuxwolf as an author (again, my bad, should have been there), clarified the allowed streams and how data is sent over it. (tjm)

**Version 0.1 (2003-06-10)**

Initial version. (tjm)

---

END