



# XMPP

## XEP-0498: Pubsub File Sharing

Jérôme Poisson

<mailto:goffi@goffi.org>

<xmpp:goffi@jabber.fr>

2024-11-20

Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	pubsub-relationships

This specification explains how to share files and optionally include directory structures similar to filesystems over XMPP Pubsub. It leverages XMPP Pubsub to enable notifications about file changes and manage permissions, providing users with real-time updates and control mechanisms. An optional mechanism is also specified for managing uploaded files.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 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.

## 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. ##

## 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.

## 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 <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>1</b>
<b>3</b>	<b>Glossary</b>	<b>2</b>
<b>4</b>	<b>General Structure and Payload</b>	<b>2</b>
<b>5</b>	<b>Node Structure and Naming</b>	<b>2</b>
5.1	Well-Known Node . . . . .	3
5.2	Node and Item Naming Convention . . . . .	3
<b>6</b>	<b>Handling Metadata and File Changes</b>	<b>4</b>
6.1	Metadata and File Association . . . . .	4
6.2	Mapping of Files and Metadata . . . . .	5
<b>7</b>	<b>End to End Encryption</b>	<b>5</b>
<b>8</b>	<b>Well-Known Directories</b>	<b>6</b>
<b>9</b>	<b>Business Rules</b>	<b>6</b>
<b>10</b>	<b>Discovering Support</b>	<b>7</b>
<b>11</b>	<b>Security Considerations</b>	<b>8</b>
<b>12</b>	<b>IANA Considerations</b>	<b>8</b>
<b>13</b>	<b>XMPP Registrar Considerations</b>	<b>8</b>
<b>14</b>	<b>Acknowledgements</b>	<b>8</b>

## 1 Introduction

File sharing is a foundational use case in XMPP, with numerous attempts to facilitate the sharing of file hierarchies. Notable efforts include [Tree Transfer Stream Initiation Profile \(XEP-0105\)](https://xmpp.org/extensions/xep-0105.html)<sup>1</sup>, [File Sharing \(XEP-0135\)](https://xmpp.org/extensions/xep-0135.html)<sup>2</sup>, [File Repository and Sharing \(XEP-0214\)](https://xmpp.org/extensions/xep-0214.html)<sup>3</sup>, and [File Information Sharing \(XEP-0329\)](https://xmpp.org/extensions/xep-0329.html)<sup>4</sup>. However, these previous approaches often fall short in key areas, particularly in the realm of notifications, and/or use now-deprecated specifications. While [File Repository and Sharing \(XEP-0214\)](https://xmpp.org/extensions/xep-0214.html)<sup>5</sup> is the only specification that supports file hierarchies with notifications, it has seen limited adoption and suffers from its reliance on an outdated stack and issues with [PubSub Collection Nodes \(XEP-0248\)](https://xmpp.org/extensions/xep-0248.html)<sup>6</sup>, especially concerning permission management between collection and leaf nodes.

The current specification addresses these shortcomings by leveraging a more modern stack. It adapts [Stateless file sharing \(XEP-0447\)](https://xmpp.org/extensions/xep-0447.html)<sup>7</sup> to Pubsub for robust access management and notifications, and incorporates [Pubsub Node Relationships \(XEP-0496\)](https://xmpp.org/extensions/xep-0496.html)<sup>8</sup> to handle hierarchical structures. This results in a flexible and easy-to-implement solution that supports a wide range of use cases, including server-based file hosting, ad-hoc or permanent per-device directory sharing, and gateways to other file-sharing protocols.

## 2 Requirements

The design goals of this XEP are:

- To use as many existing mechanisms as possible.
- To be as easy as possible to implement for both clients and services, once dependencies are in place.
- To support mapping a traditional filesystem directory structure.
- To support notifications for changes, such as new or deleted files and directories.
- To allow for server file hosting use cases.
- To enable device file directory structure sharing use cases.
- To facilitate gateways to other file hosting protocols or mechanisms, such as FTP or "cloud" hosting.

---

<sup>1</sup>XEP-0105: Tree Transfer Stream Initiation Profile <<https://xmpp.org/extensions/xep-0105.html>>.

<sup>2</sup>XEP-0135: File Sharing <<https://xmpp.org/extensions/xep-0135.html>>.

<sup>3</sup>XEP-0214: File Repository and Sharing <<https://xmpp.org/extensions/xep-0214.html>>.

<sup>4</sup>XEP-0329: File Information Sharing <<https://xmpp.org/extensions/xep-0329.html>>.

<sup>5</sup>XEP-0214: File Repository and Sharing <<https://xmpp.org/extensions/xep-0214.html>>.

<sup>6</sup>XEP-0248: PubSub Collection Nodes <<https://xmpp.org/extensions/xep-0248.html>>.

<sup>7</sup>XEP-0447: Stateless file sharing <<https://xmpp.org/extensions/xep-0447.html>>.

<sup>8</sup>XEP-0496: Pubsub Node Relationships <<https://xmpp.org/extensions/xep-0496.html>>.

- To permit access control at the node and descendant levels, without parent nodes overriding descendant node permissions and vice versa.
- To provide a quick file listing soon after a file structure is shared.
- To ensure compatibility with end-to-end encryption for both file metadata and transfers.

### 3 Glossary

This section defines key terms used throughout this specification.

- **Root Node:** The well-known node "urn:xmpp:pubsub-file-sharing:0" that serves as the top-level container for shared files and directories. It is the entry point for discovering all shared resources.
- **Directory:** A logical grouping of files and/or other directories, represented by a Pubsub node. They can map to usual filesystem directories.
- **Well-Known Node/Directory:** A predefined node/directory with a specific purpose, such as "urn:xmpp:pubsub-file-sharing:0" (root node) or "urn:xmpp:pubsub-file-sharing:0:/uploaded" (Uploaded directory).

### 4 General Structure and Payload

Pubsub nodes represent directories, and files data are put in their items. The payload of the file items MUST be a <file-sharing> element, including metadata and sources, as specified in [Stateless file sharing \(XEP-0447\)](#)<sup>9</sup>. The <file-sharing> element MUST NOT have a 'disposition' attribute as it doesn't make sense in the context of pubsub file sharing.

### 5 Node Structure and Naming

To facilitate a clear and consistent approach to file sharing, this specification defines a well-known node for discovering shared files and a node/item naming convention. This ensures that files and directories can be easily identified and managed.

---

<sup>9</sup>XEP-0447: Stateless file sharing <<https://xmpp.org/extensions/xep-0447.html>>.

## 5.1 Well-Known Node

Entities that support file sharing through this protocol MUST have a well-known node for discovering shared files. This node is defined as "urn:xmpp:pubsub-file-sharing:0".

When a user subscribes to this node using XEP-XXXX: Pubsub Extended Subscription, they will receive notifications for any new or deleted files and directories up to the requested depth. The node's items represent the shared files.

To discover file hierarchy, use XEP-XXXX: Pubsub Extended Discovery.

Listing 1: Discovering Directories From Well-known Node

```
<iq type='get'
  from='juliet@example.org/balcony'
  to='files.example.org'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='urn:xmpp:pubsub-file-sharing:0'>
    <x xmlns='jabber:x:data' type='submit'>
      <field var='FORM_TYPE' type='hidden'>
        <value>urn:xmpp:pubsub-ext-disco:0</value>
      </field>
      <field var='type'>
        <value>nodes</value>
      </field>
      <field var='depth'>
        <value>1</value>
      </field>
    </x>
  </query>
</iq>
```

## 5.2 Node and Item Naming Convention

To ensure a consistent and intuitive structure while maintaining unique node names across the pubsub service, the following naming convention is used for nodes and items:

- **Root Node:** The well-known node "urn:xmpp:pubsub-file-sharing:0" serves as the root of the file sharing structure.
- **Directory Nodes:** Each directory is represented by a node with a name that reflects its path relative to the root. This node name follows the template "urn:xmpp:pubsub-file-sharing:0:<unique ID>/<directory name>", where <unique ID> is a unique identifier across the pubsub service and MUST NOT contain the "/" character. For example, a directory named "Documents" with a unique ID of "abc123" would be represented by the node "urn:xmpp:pubsub-file-sharing:0:abc123/Documents".

- **File Items:** Each file is represented as an item within its corresponding directory node. The item's ID is the name of the file, and its payload contains metadata and source information for the file. For example, a file named "report.pdf" in the "Documents" directory (with unique ID "abc123") would be represented by the item "report.pdf", which is within the node "urn:xmpp:pubsub-file-sharing:0:abc123/Documents".

This convention allows for an easy mapping of file system structure to a pubsub node hierarchy, making it simple to navigate and manage shared files without encountering conflicts due to non-unique node names.

## 6 Handling Metadata and File Changes

This section specifies how metadata and file deletions are handled in the context of shared files and directories.

### 6.1 Metadata and File Association

Each file item includes metadata and source information. The metadata is provided using the file element as described in [File metadata element \(XEP-0446\)](#)<sup>10</sup>. The source information is provided using the sources element, which lists the available sources for the file.

Listing 2: Example of a File Item With Metadata and Sources

```
<iq type='result'
  from='files.example.org'
  to='juliet@example.org/balcony'
  id='items1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='urn:xmpp:pubsub-file-sharing:0:EASD1546/Images'>
      <item id='beach-sunset.jpg'>
        <file-sharing xmlns='urn:xmpp:sfs:0'>
          <file xmlns='urn:xmpp:file:metadata:0'>
            <media-type>image/jpeg</media-type>
            <name>beach-sunset.jpg</name>
            <size>8765432</size>
            <hash xmlns='urn:xmpp:hashes:2' algo='sha3-256'>
              GZTbL1pJmX9jzVgQwR0HfI+DyKuPcWvE2nUxYiMkS78=</hash>
            <desc>A stunning sunset on the beach.</desc>
          </file>
          <sources>
            <url-data xmlns='http://jabber.org/protocol/url-data'
              target='https://photos.example.org/gallery/beach-
                sunset.jpg' />
          </sources>
        </file-sharing>
      </item>
    </items>
  </pubsub>
</iq>
```

<sup>10</sup>XEP-0446: File metadata element <<https://xmpp.org/extensions/xep-0446.html>>.

```

        <jinglepub xmlns='urn:xmpp:jinglepub:1' from='files.
            example.org' id='9876543B-2CFB-AE7E-B45Z-3DAA225972BB'
            >
            <description xmlns='urn:xmpp:jingle:apps:file-transfer:5
                ' />
            </jinglepub>
        </sources>
    </file-sharing>
</item>
</items>
</pubsub>
</iq>

```

## 6.2 Mapping of Files and Metadata

For services declaring support for this specification (as explained in the [Discovering Support](#) section), the following rules apply:

- When a mapped file is deleted or updated, the corresponding items SHOULD be deleted or updated accordingly.
- Conversely, when an item in pubsub corresponding to a file is updated or retracted, the corresponding file SHOULD be updated or deleted accordingly.
- When a directory is deleted, the mapped node SHOULD also be deleted accordingly.
- When a directory is renamed, the mapped node SHOULD first be deleted and another one SHOULD be created with the new name. Additionally, the `<redirect/>` element SHOULD be used as specified in the [XEP-0060: Delete a Node/Success Case](#).

A service MAY prohibit deletion or update of items based on its internal policy. In this case, the service MUST return a `<forbidden>` error and SHOULD use a human-readable explanation of the error.

If support for this specification is not advertised, it means that file sharing metadata are manually set by an XMPP client on a generic pubsub service. In such cases, the mapping cannot be performed by the pubsub service and there is no guarantee of synchronization between files and metadata.

## 7 End to End Encryption

File sharing can optionally use end-to-end encryption. For file metadata, this can be achieved at the pubsub level using specifications such as [OpenPGP for XMPP Pubsub \(XEP-0473\)](#)<sup>11</sup> or

<sup>11</sup>XEP-0473: OpenPGP for XMPP Pubsub <<https://xmpp.org/extensions/xep-0473.html>>.



[Pubsub Targeted Encryption \(XEP-0477\)](#)<sup>12</sup> or any relevant mechanism. For file transfers, this is dependent on the chosen source, and specifications such as [Jingle Encrypted Transports \(XEP-0391\)](#)<sup>13</sup> with [Jingle Encrypted Transports - OMEMO \(XEP-0396\)](#)<sup>14</sup> could be used.

## 8 Well-Known Directories

This specification introduces predefined nodes for special directories that MAY be used by pubsub services. Other specifications MAY define additional well-known nodes as needed. If these directories are implemented, they SHOULD be attached to the root node "urn:xmpp:pubsub-file-sharing:0".

The only explicitly defined directory in this specification is the 'Uploaded' directory. This directory MUST use the node named "urn:xmpp:pubsub-file-sharing:0:/uploaded", where the unique ID field is intentionally left blank. The purpose of the 'Uploaded' directory is to track files uploaded via specifications such as [HTTP File Upload \(XEP-0363\)](#)<sup>15</sup>. It enables end-users to view and manage their uploaded files (e.g., for deletion purposes), typically through the PEP service associated with each user.

## 9 Business Rules

The following business rules apply to the file sharing protocol:

- Support for XEP-XXXX Pubsub Node Relationships is optional. If not supported, the file sharing protocol cannot handle file hierarchies and can only manage flat files. Well-known directories cannot be attached to the root node, and clients must directly search for them. Therefore, support for file hierarchies is tied to support for XEP-XXXX: Pubsub Node Relationships and is detected as explained in its Discovering Support section.
- If a service advertises support for this specification (as explained in [Discovering Support](#)) but cannot monitor file changes (e.g., because it does not support [inotify](#) or a similar mechanism), it MUST refuse any subscription attempts by sending a `<feature-not-implemented/>` error with a pubsub-specific error condition of `<unsupported/>`, as explained in [XEP-0060: §6.1.3.10 Subscriptions Not Supported](#).
- Entities MUST ensure that access control is correctly managed at the node and descendant levels, without parent nodes overriding descendant node permissions and vice versa.

<sup>12</sup>XEP-0477: Pubsub Targeted Encryption [-<https://xmpp.org/extensions/xep-0477.html>](https://xmpp.org/extensions/xep-0477.html).

<sup>13</sup>XEP-0391: Jingle Encrypted Transports [-<https://xmpp.org/extensions/xep-0391.html>](https://xmpp.org/extensions/xep-0391.html).

<sup>14</sup>XEP-0396: Jingle Encrypted Transports - OMEMO [-<https://xmpp.org/extensions/xep-0396.html>](https://xmpp.org/extensions/xep-0396.html).

<sup>15</sup>XEP-0363: HTTP File Upload [-<https://xmpp.org/extensions/xep-0363.html>](https://xmpp.org/extensions/xep-0363.html).

- A pubsub service MAY propose different hierarchies (i.e., different child nodes or items) from the root node according to the requesting entities.
- As the number of items may be high, a supporting service SHOULD use [Result Set Management \(XEP-0059\)](#)<sup>16</sup>.

## 10 Discovering Support

If an entity supports sharing files through the protocol specified in this XEP, it MUST advertise it by including the "urn:xmpp:pubsub-file-sharing:0" discovery feature in response to a [Service Discovery \(XEP-0030\)](#)<sup>17</sup> information request.

Listing 3: Service Discovery Information Request

```
<iq type='get'
  from='juliet@example.org/balcony'
  to='files.example.org'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 4: Service Discovery Information Response

```
<iq type='result'
  from='files.example.org'
  to='juliet@example.org/balcony'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:pubsub-file-sharing:0' />
    ...
  </query>
</iq>
```

When declaring support for this protocol, a pubsub service MUST manage a mapping between files and pubsub nodes.

Alternatively, XMPP clients can use this specification by filling in metadata related to files on any generic pubsub service. In such cases, there is no need to advertise support at the discovery level. File sharing nodes are identified either by searching for the well-known node 'urn:xmpp:pubsub-file-sharing:0' or any other node that starts with a similar prefix.

<sup>16</sup>XEP-0059: Result Set Management <<https://xmpp.org/extensions/xep-0059.html>>.

<sup>17</sup>XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

## 11 Security Considerations

The following security considerations apply to the file sharing protocol:

- Entities **MUST** manage access control at both the node and descendant levels, ensuring that permissions are preserved for all child and parent nodes. This prevents unauthorized access to files and directories.
- Entities **SHOULD** use secure protocols (e.g., HTTPS) for transferring files to prevent eavesdropping and tampering with the data.
- Entities **SHOULD** implement rate limiting and other security measures to prevent abuse of the file sharing service.
- File sharing involves numerous security risks. Implementations must take care to prevent the diffusion of sensitive files (e.g., "/etc/password"), denial-of-service (DoS) techniques such as never completing a file transfer or repeatedly requesting files, improper permission mapping, and accidental sharing (e.g., by following symbolic links).
- The security considerations of dependencies, such as [Stateless file sharing \(XEP-0447\)](#)<sup>18</sup>, also apply.
- Supporting services **SHOULD** hide nodes and items from entities that are not authorized to access the corresponding files or directories.

## 12 IANA Considerations

This document does not require interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>19</sup>.

## 13 XMPP Registrar Considerations

TODO

## 14 Acknowledgements

Thanks to NLNet foundation/NGI Zero Core for funding the work on this specification.

---

<sup>18</sup>XEP-0447: Stateless file sharing <<https://xmpp.org/extensions/xep-0447.html>>.

<sup>19</sup>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/>>.