

# FOAF Vocabulary Specification

Namespace Document 19 May 2005 - ('Are Friends Downloadable?' Edition)

## Latest version:

<http://xmlns.com/foaf/0.1/>

## Previous version:

2005-04-03 ([html](#), [rdf](#))

## Last update:

\$Date: 2007-06-07 17:51:51 \$

## Revision:

\$Revision: 1.1 \$

## Authors:

[Dan Brickley](#), [Libby Miller](#)

## Contributors:

Members of the FOAF mailing list ([rdweb-dev@vapours.rdfweb.org](mailto:rdweb-dev@vapours.rdfweb.org)) and the wider [RDF developer community](#). See [acknowledgements](#).

Copyright © 2000-2005 Dan Brickley and Libby Miller

This work is licensed under a [Creative Commons License](#). This copyright applies to the *FOAF Vocabulary Specification* and accompanying documentation and does not apply to FOAF data formats, vocabulary terms, or technology. Regarding underlying technology, FOAF relies heavily on W3C's [RDF](#) technology, an open Web standard that can be freely used by anyone.



## Status of This Document

**You are reading a live-edit edition of FOAF.** This means that the spec will change a bit, up until the date given (19 May in this case; tomorrow as I write). This is an experiment introduced alongside the links to date-snapshotted previous versions.

**NOTE:** This specification is an evolving document. The bulk of this document is now generated by combining the [RDFS/OWL](#) machine-readable FOAF ontology with a set of [per-term](#) documents. The remaining sections require serious attention; we are well aware that better examples and tighter intro text are needed. Rather than try to perfect things offline, we are switching over to the new spec as it brings a lot of new explanatory text into view.

Our editorial TODO list includes: fix table of contents, reorg intro text, add examples, fix CSS, find a strategy to allow XHTML validation while embedding RDF (currently document won't validate). The spec-management scripts should also be extended to generate an index of FOAF terms organised by status (ie. 'unstable', 'testing' or 'stable') as well as to expose OWL information about FOAF classes and properties, such as 'mutually disjoint' classes, 'inverse' properties, etc.

*This section describes the status of this document at the time of its publication. Other documents may supersede this document.*

This document is produced as part of the [RDFWeb FOAF project](#), to provide authoritative documentation of the contents, status and purpose of the RDF/XML vocabulary and document formats known informally as 'FOAF'.

The authors welcome comments on this document. Please send comments on this document to the FOAF developers list [rdweb-dev@vapours.rdfweb.org](mailto:rdweb-dev@vapours.rdfweb.org); [public archives](#) are available. This document may be updated or added to based on implementation experience, but no commitment is made by the authors regarding future updates (*todo: link to stability info re FOAF properties, also need to explain extensibility model*).

This document contains embedded XML/RDF statements defining and describing the FOAF vocabulary.

## Recent Changes

- 2004-09-01: dropped the claim that foaf:homepage has an rdfs:domain of foaf:Agent, to allow other kinds of things to be said to have homepages (eg. Events)

## Table of Contents

- [Introduction](#)
  - [FOAF Project](#)
  - [FOAF files](#)
  - [FOAF vocabulary](#)
  - [Using this specification](#)
- [Introduction to RDF/XML](#)
  - RDF
  - RDFS/OWL
- [FOAF Syntax](#)
- [FOAF Vocabulary](#)
  - [Classes and Properties](#)
  - [FOAF Basics](#)
  - [Personal Info](#)
  - [Projects and Groups](#)
  - [Documents and Images](#)
  - [Messaging / Chat](#)
  - [Extensions and mixing](#)
    - [Dublin Core](#)
    - [Wordnet](#)
    - [RSS 1.0 and Atom feeds](#)
    - [Other extensions](#)
- [Security considerations](#)
- [Privacy considerations](#)
- [FOAF development process](#)
- [Vocabulary Cross-reference](#)
- [RDFS/OWL formal specification](#)
- [FOAF At A Glance](#)
- [FOAF Overview](#)
- [FOAF Basics](#)
- [Personal Info](#)
- [FOAF Classes \(types\)](#)
- [IM / Chat](#)
- [Projects and Groups](#)

- Documents and Images
- Introduction
- FOAF Basics
- Example
- Background
- FOAF and Standards
- The FOAF Vocabulary Description
- Evolution and Extension of FOAF
- FOAF Auto-Discovery: Publishing and Linking FOAF files

## FOAF At A Glance

An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | [Agent](#) | [Document](#) | [Group](#) | [Image](#) | [OnlineAccount](#) | [OnlineChatAccount](#) | [OnlineEcommerceAccount](#) | [OnlineGamingAccount](#) | [Organization](#) | [Person](#) | [PersonalProfileDocument](#) | [Project](#) |

Properties: | [accountName](#) | [accountServiceHomepage](#) | [aimChatID](#) | [based\\_near](#) | [birthday](#) | [currentProject](#) | [depiction](#) | [depicts](#) | [dnaChecksum](#) | [family\\_name](#) | [firstName](#) | [fundedBy](#) | [geekcode](#) | [gender](#) | [givenname](#) | [holdsAccount](#) | [homepage](#) | [icqChatID](#) | [img](#) | [interest](#) | [jabberID](#) | [knows](#) | [logo](#) | [made\\_maker](#) | [mbox](#) | [mbox\\_sha1sum](#) | [member](#) | [membershipClass](#) | [msnChatID](#) | [myersBriggs](#) | [name](#) | [page](#) | [pastProject](#) | [phone](#) | [plan](#) | [primaryTopic](#) | [publications](#) | [schoolHomepage](#) | [sha1](#) | [surname](#) | [theme](#) | [thumbnail](#) | [tipjar](#) | [title](#) | [topic](#) | [topic\\_interest](#) | [weblog](#) | [workInfoHomepage](#) | [workplaceHomepage](#) | [yahooChatID](#) |

## FOAF Overview

FOAF terms, grouped in broad categories.

### FOAF Basics

- [Agent](#)
- [Person](#)
- [name](#)
- [nick](#)
- [title](#)
- [homepage](#)
- [mbox](#)
- [mbox\\_sha1sum](#)
- [img](#)
- [depiction](#) ([depicts](#))
- [surname](#)
- [family\\_name](#)
- [givenname](#)
- [firstName](#)

### Personal Info

- [weblog](#)
- [knows](#)
- [interest](#)
- [currentProject](#)
- [pastProject](#)
- [plan](#)
- [based\\_near](#)
- [workplaceHomepage](#)
- [workInfoHomepage](#)
- [schoolHomepage](#)
- [topic\\_interest](#)
- [publications](#)
- [geekcode](#)
- [myersBriggs](#)
- [dnaChecksum](#)

### Online Accounts / IM

- [OnlineAccount](#)
- [OnlineChatAccount](#)
- [OnlineEcommerceAccount](#)
- [OnlineGamingAccount](#)
- [holdsAccount](#)
- [accountServiceHomepage](#)
- [accountName](#)
- [icqChatID](#)
- [msnChatID](#)
- [aimChatID](#)
- [jabberID](#)
- [yahooChatID](#)

### Projects and Groups

- [Project](#)
- [Organization](#)
- [Group](#)
- [member](#)
- [membershipClass](#)
- [fundedBy](#)
- [theme](#)

### Documents and Images

- [Document](#)
- [Image](#)
- [PersonalProfileDocument](#)
- [topic](#) ([page](#))
- [primaryTopic](#)
- [tipjar](#)
- [sha1](#)
- [made](#) ([maker](#))
- [thumbnail](#)
- [logo](#)

## 1 Introduction: FOAF Basics

The FOAF project is based around the use of *machine readable* Web homepages for people, groups, companies and other kinds of thing. To achieve this we use the "FOAF vocabulary" to provide a collection of basic terms that can be used in these Web pages. At the heart of the FOAF project is a set of definitions designed to serve as a dictionary of terms that can be used to express claims about the world. The initial focus of FOAF has been on the description of people, since people are the things that link together most of the other kinds of things we describe in the Web: they make documents, attend meetings, are depicted in photos, and so on.

The FOAF Vocabulary definitions presented here are written using a computer language (RDF/OWL) that makes it easy for software to process some basic facts about the terms in the FOAF vocabulary, and consequently about the things described in FOAF documents. A FOAF document, unlike a traditional Web page, can be combined with other FOAF documents to create a unified database of information.

## Example

Here is a very basic document describing a person:

```
<foaf:Person>
  <foaf:name>Dan Brickley</foaf:name>
  <foaf:mbox_sha1sum>241021fb0e6289f92815fc210f9e9137262c252e</foaf:mbox_sha1sum>
  <foaf:homepage rdf:resource="http://rdfweb.org/people/danbri/" />
  <foaf:img rdf:resource="http://rdfweb.org/people/danbri/mugshot/danbri-small.jpeg" />
</foaf:Person>
```

This brief example introduces the basics of FOAF. It basically says, "there is a [foaf:Person](#) with a [foaf:name](#) property of 'Dan Brickley' and a [foaf:mbox\\_sha1sum](#) property of 241021fb0e6289f92815fc210f9e9137262c252e; this person stands in a [foaf:homepage](#) relationship to a thing called [http://rdfweb.org/people/danbri/](#) and a [foaf:img](#) relationship to a thing called [http://rdfweb.org/people/danbri/mugshot/danbri-small.jpeg](#).

## The Basic Idea

The basic idea is pretty simple. If people publish information in the FOAF document format, machines will be able to make use of that information. If those files contain "see also" references to other such documents in the Web, we will have a machine-friendly version of today's hypertext Web. Computer programs will be able to scutter around a Web of documents designed for machines rather than humans, storing the information they find, keeping a list of "see also" pointers to other documents, checking digital signatures (for the security minded) and building Web pages and question-answering services based on the harvested documents.

So, what is the 'FOAF document format'? FOAF files are just text documents (well, Unicode documents). They are written in XML syntax, and adopt the conventions of the Resource Description Framework (RDF). In addition, the FOAF vocabulary defines some useful constructs that can appear in FOAF files, alongside other RDF vocabularies defined elsewhere. For example, FOAF defines categories ('classes') such as `foaf:Person`, `foaf:Document`, `foaf:Image`, alongside some handy properties of those things, such as `foaf:name`, `foaf:mbox` (ie. an internet mailbox), `foaf:homepage` etc., as well as some useful kinds of relationship that hold between members of these categories. For example, one interesting relationship type is `foaf:depiction`. This relates something (eg. a `foaf:Person`) to a `foaf:Image`. The FOAF demos that feature photos and listings of 'who is in which picture' are based on software tools that parse RDF documents and make use of these properties.

The specific contents of the FOAF vocabulary are detailed in this [FOAF namespace document](#). In addition to the FOAF vocabulary, one of the most interesting features of a FOAF file is that it can contain "see Also" pointers to other FOAF files. This provides a basis for automatic harvesting tools to traverse a Web of interlinked files, and learn about new people, documents, services, data...

The remainder of this specification describes how to publish and interpret descriptions such as these on the Web, using RDF/XML for syntax (file format) and terms from FOAF. It introduces a number of categories (RDF classes such as 'Person') and properties (relationship and attribute types such as 'mbox' or 'workplaceHomepage'). Each term definition is provided in both human and machine-readable form, hyperlinked for quick reference.

## What's FOAF for?

For a good general introduction to FOAF, see Edd Dumbill's article, [XML Watch: Finding friends with XML and RDF](#) (June 2002, IBM developerWorks). Information about the use of FOAF [with image metadata](#) is also available.

The [co-depiction](#) experiment shows a fun use of the vocabulary. Jim Ley's [SVG image annotation tool](#) show the use of FOAF with detailed image metadata, and provide tools for labelling image regions within a Web browser. To create a FOAF document, you can use Leigh Dodd's [FOAF-a-matic](#) javascript tool. To query a FOAF dataset via IRC, you can use Edd Dumbill's [FOAFbot](#) tool, an IRC 'community support agent'. For more information on FOAF and related projects, see the [FOAF project home page](#) at [rdfweb.org](#).

## Background

FOAF is managed as a collaborative effort amongst Semantic Web developers on the FOAF ([rdfweb-dev@vapours.rdfweb.org](mailto:rdfweb-dev@vapours.rdfweb.org)) mailing list. The name 'FOAF' is derived from traditional internet usage, an acronym for 'Friend of a Friend'.

The name was chosen to reflect our concern with social networks and the Web, urban myths, trust and connections. Other uses of the name continue, notably in the documentation and investigation of Urban Legends (eg. see the [alt.folklore.urban archive](#) or [snopes.com](#)), and other FOAF stories. Our use of the name 'FOAF' for a Web vocabulary and document format is intended to complement, rather than replace, these prior uses. FOAF documents describe the characteristics and relationships amongst friends of friends, and their friends, and the stories they tell.

## FOAF and Standards

It is important to understand that the FOAF *vocabulary* as specified in this document is not a standard in the sense of [ISO Standardisation](#), or that associated with [W3C Process](#).

FOAF depends heavily on W3C's standards work, specifically on XML, XML Namespaces, RDF, and OWL. All FOAF *documents* must be well-formed RDF/XML documents. The FOAF vocabulary, by contrast, is managed more in the style of an [Open Source](#) or [Free Software](#) project than as an industry standardisation effort (eg. see [Jabber JEPs](#) or Python [\\_re](#) extensions).

This specification contributes a vocabulary, "FOAF", to the Semantic Web, specifying it using W3C's [Resource Description Framework](#) (RDF). As such, FOAF adopts by reference both a syntax (using XML) a data model (RDF graphs) and a mathematically grounded definition for the rules that underpin the FOAF design.

## The FOAF Vocabulary Description

This specification serves as the FOAF "namespace document". As such it describes the FOAF vocabulary and the terms (RDF classes and properties) that constitute it, so that [Semantic Web](#) applications can use those terms in a variety of RDF-compatible document formats and applications.

This document presents FOAF as a [Semantic Web](#) vocabulary or *Ontology*. The FOAF vocabulary is pretty simple, pragmatic and designed to allow simultaneous deployment and extension. FOAF is intended for widescale use, but its authors make no commitments regarding its suitability for any particular purpose.

## Evolution and Extension of FOAF

The FOAF vocabulary is identified by the namespace URI '<http://xmlns.com/foaf/0.1/>'. Revisions and extensions of FOAF are conducted through edits to this document, which by convention is published in the Web at the namespace URI. The evolution of FOAF is best considered in terms of the stability of individual vocabulary terms, rather than the specification as a whole. As terms stabilise in usage and documentation, they progress through the categories '**unstable**', '**testing**' and '**stable**'.

The properties and types defined here provide some basic useful concepts for use in FOAF descriptions. Other vocabulary (eg. the [Dublin Core](#) metadata elements for simple bibliographic description), RSS 1.0 etc can also be mixed in with FOAF terms, as can local extensions. FOAF is designed to be extended. The [FoafVocab](#) page in the FOAF wiki lists a number of extension vocabularies that are particularly applicable to use with FOAF.

## FOAF Auto-Discovery: Publishing and Linking FOAF files

If you publish a FOAF self-description (eg. using [foaf-a-matic](#)) you can make it easier for tools to find your FOAF by putting markup in the `head` of your HTML homepage. It doesn't really matter what filename you choose for your FOAF document, although `foaf.rdf` is a common choice. The linking markup is as follows:

```
<link rel="meta" type="application/rdf+xml" title="FOAF" href="http://example.com/people/~you/foaf.rdf" />
```

...although of course change the *URL* to point to your own FOAF document. See also: more on [FOAF autodiscovery](#) and services that make use of it.

## FOAF and RDF

Why does FOAF use [RDF](#)?

FOAF is an application of the Resource Description Framework (RDF) because the subject area we're describing -- people -- has so many competing requirements that a standalone format could not do them all justice. By using RDF, FOAF gains a powerful extensibility mechanism, allowing FOAF-based descriptions can be mixed with claims made in *any other RDF vocabulary*

People are the things that link together most of the other kinds of things we describe in the Web: they make documents, attend meetings, are depicted in

photos, and so on. Consequently, there are many many things that we might want to say about people, not to mention these related objects (ie. documents, photos, meetings etc).

FOAF as a vocabulary cannot incorporate everything we might want to talk about that is related to people, or it would be as large as a full dictionary. Instead of covering all topics within FOAF itself, we buy into a larger framework - RDF - that allows us to take advantage of work elsewhere on more specific description vocabularies (eg. for geographical / mapping data).

RDF provides FOAF with a way to mix together different descriptive vocabularies in a consistent way. Vocabularies can be created by different communities and groups as appropriate and mixed together as required, without needing any centralised agreement on how terms from different vocabularies can be written down in XML.

This mixing happens in two ways: firstly, RDF provides an underlying model of (typed) objects and their attributes or relationships. `foaf:Person` is an example of a type of object (a "class"), while `foaf:knows` and `foaf:name` are examples of a relationship and an attribute of an `foaf:Person`; in RDF we call these "properties". Any vocabulary described in RDF shares this basic model, which is discernable in the syntax for RDF, and which removes one level of confusion in *understanding* a given vocabulary, making it simpler to comprehend and therefore reuse a vocabulary that you have not written yourself. This is the minimal *self-documentation* that RDF gives you.

Secondly, there are mechanisms for saying which RDF properties are connected to which classes, and how different classes are related to each other, using RDF Syntax and OWL. These can be quite general (all RDF properties by default come from an `rdf:Resource` for example) or very specific and precise (for example by using [OWL](#) constructs, as in the `foaf:Group` example below. This is another form of self-documentation, which allows you to connect different vocabularies together as you please. An example of this is given below where the `foaf:based_near` property has a domain and range (types of class at each end of the property) from a different namespace altogether.

In summary then, RDF is self-documenting in ways which enable the creation and combination of vocabularies in a devolved manner. This is particularly important for a vocabulary which describes people, since people connect to many other domains of interest, which it would be impossible (as well as suboptimal) for a single group to describe adequately in non-geological time.

RDF is usually written using XML syntax, but behaves in rather different ways to 'vanilla' XML: the same RDF can be written in many different ways in XML. This means that SAX and DOM XML parsers are not adequate to deal with RDF/XML. If you want to process the data, you will need to use one of the many RDF toolkits available, such as Jena (Java) or Redland (C). [RDF Interest Group](#) members can help with issues which may arise; there is also the [rdfweb-dev@yapours.rdfweb.org](#) mailing list which is the main list for FOAF, and two active and friendly [IRC](#) channels: [#rdfig](#) and [#foaf](#) on [freenode](#).

## FOAF cross-reference: Listing FOAF Classes and Properties

FOAF introduces the following classes and properties. View this document's source markup to see the RDF/XML version.

See also: [rdf schema signature](#) and [html doc signature](#) (@@TODO).

### Classes and Properties (full detail)

#### Class: foaf:Agent

*Agent* - An agent (eg. person, group, software or physical artifact).

**Status:** unstable

**in-range-of:** [foaf:maker](#) [foaf:member](#)

**in-domain-of:** [foaf:mbox](#) [foaf:mbox\\_sha1sum](#) [foaf:gender](#) [foaf:jabberID](#) [foaf:aimChatID](#) [foaf:icqChatID](#) [foaf:yahooChatID](#) [foaf:msnChatID](#) [foaf:weblog](#) [foaf:tipjar](#) [foaf:made](#) [foaf:holdsAccount](#) [foaf:birthday](#)

The [foaf:Agent](#) class is the class of agents; things that do stuff. A well known sub-class is [foaf:Person](#), representing people. Other kinds of agents include [foaf:Organization](#) and [foaf:Group](#).

The [foaf:Agent](#) class is useful in a few places in FOAF where [foaf:Person](#) would have been overly specific. For example, the IM chat ID properties such as `jabberID` are typically associated with people, but sometimes belong to software bots.

[\[back to top\]](#)

#### Class: foaf:Document

*Document* - A document.

**Status:** testing

**in-range-of:** [foaf:homepage](#) [foaf:weblog](#) [foaf:tipjar](#) [foaf:workplaceHomepage](#) [foaf:workInfoHomepage](#) [foaf:schoolHomepage](#) [foaf:interest](#) [foaf:publications](#) [foaf:page](#) [foaf:accountServiceHomepage](#)

**in-domain-of:** [foaf:sha1](#) [foaf:topic](#) [foaf:primaryTopic](#)

The [foaf:Document](#) class represents those things which are, broadly conceived, 'documents'.

The [foaf:Image](#) class is a sub-class of [foaf:Document](#), since all images are documents.

We do not (currently) distinguish precisely between physical and electronic documents, or between copies of a work and the abstraction those copies embody. The relationship between documents and their byte-stream representation needs clarification (see [foaf:sha1](#) for related issues).

[\[back to top\]](#)

#### Class: foaf:Group

*Group* - A class of Agents.

**Status:** unstable

**in-domain-of:** [foaf:member](#) [foaf:membershipClass](#)

The [foaf:Group](#) class represents a collection of individual agents (and may itself play the role of a [foaf:Agent](#), ie. something that can perform actions).

This concept is intentionally quite broad, covering informal and ad-hoc groups, long-lived communities, organizational groups within a workplace, etc. Some such groups may have associated characteristics which could be captured in RDF (perhaps a [foaf:homepage](#), [foaf:name](#), mailing list etc.).

While a [foaf:Group](#) has the characteristics of a [foaf:Agent](#), it is also associated with a number of other [foaf:Agents](#) (typically people) who constitute the [foaf:Group](#). FOAF provides a mechanism, the [foaf:membershipClass](#) property, which relates a [foaf:Group](#) to a sub-class of the class [foaf:Agent](#) who are members of the group. This is a little complicated, but allows us to make group membership rules explicit.

The markup (shown below) for defining a group is both complex and powerful. It allows group membership rules to match against any RDF-describable characteristics of the potential group members. As FOAF and similar vocabularies become more expressive in their ability to describe individuals, the [foaf:Group](#) mechanism for categorising them into groups also becomes more powerful.

While the formal description of membership criteria for a [foaf:Group](#) may be complex, the basic mechanism for saying that someone is in a [foaf:Group](#) is very simple. We simply use a [foaf:member](#) property of the [foaf:Group](#) to indicate the agents that are members of the group. For example:

```
<foaf:Group>
  <foaf:name>ILRT staff</foaf:name>
  <foaf:member>
    <foaf:Person>
      <foaf:name>Libby Miller</foaf:name>
      <foaf:homepage rdf:resource="http://ilrt.org/people/libby/" />
      <foaf:workplaceHomepage rdf:resource="http://www.ilrt.bris.ac.uk/" />
    </foaf:Person>
  </foaf:member>
</foaf:Group>
```

Behind the scenes, further RDF statements can be used to express the rules for being a member of this group. End-users of FOAF need not pay attention to these details.

Here is an example. We define a [foaf:Group](#) representing those people who are ILRT staff members. The [foaf:membershipClass](#) property connects the group (conceived of as a social entity and agent in its own right) with the class definition for those people who constitute it. In this case, the rule is that all group members are in the ILRTStaffPerson class, which is in turn populated by all those things that are a [foaf:Person](#) and which have a [foaf:workplaceHomepage](#) of <http://www.ilrt.bris.ac.uk/>. This is typical: FOAF groups are created by specifying a sub-class of [foaf:Agent](#) (in fact usually this will be a sub-class of [foaf:Person](#)), and giving criteria for which things fall in or out of the sub-class. For this, we use the `owl:onProperty` and `owl:hasValue` properties, indicating the property/value pairs which must be true of matching agents.

```
<!-- here we see a FOAF group described.
      each foaf group may be associated with an OWL definition
      specifying the class of agents that constitute the group's membership -->
<foaf:Group>
  <foaf:name>ILRT staff</foaf:name>
  <foaf:membershipClass>
    <owl:Class rdf:about="http://ilrt.example.com/groups#ILRTStaffPerson">
      <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://xmlns.com/foaf/0.1/workplaceHomepage"/>
          <owl:hasValue rdf:resource="http://www.ilrt.bris.ac.uk/" />
        </owl:Restriction>
      </rdfs:subClassOf>
    </owl:Class>
  </foaf:membershipClass>
</foaf:Group>
```

Note that while these example OWL rules for being in the eg:ILRTStaffPerson class are based on a [foaf:Person](#) having a particular [foaf:workplaceHomepage](#), this places no obligations on the authors of actual FOAF documents to include this information. If the information *is* included, then generic OWL tools may infer that some person is an eg:ILRTStaffPerson. To go the extra step and infer that some eg:ILRTStaffPerson is a [foaf:member](#) of the group whose [foaf:name](#) is "ILRT staff", tools will need some knowledge of the way FOAF deals with groups. In other words, generic OWL technology gets us most of the way, but the full [foaf:Group](#) machinery requires extra work for implementors.

The current design names the relationship as pointing *from* the group, to the member. This is convenient when writing XML/RDF that encloses the members within markup that describes the group. Alternate representations of the same content are allowed in RDF, so you can write claims about the Person and the Group without having to nest either description inside the other. For (brief) example:

```
<foaf:Group>
  <foaf:member rdf:nodeID="libby"/>
  <!-- more about the group here -->
</foaf:Group>
<foaf:Person rdf:nodeID="libby">
  <!-- more about libby here -->
</foaf:Person>
```

There is a FOAF [issue tracker](#) (8) associated with this FOAF term. A design goal is to make the most of W3C's [OWL](#) language for representing group-membership criteria, while also making it easy to leverage existing groups and datasets available online (eg. buddylists, mailing list membership lists etc). Feedback on the current design is solicited!

[\[back to top\]](#)

## Class: foaf:Image

*Image* - An image.

**Status:** testing

**in-range-of:** [foaf:img](#) [foaf:depiction](#) [foaf:thumbnail](#)

**in-domain-of:** [foaf:depicts](#) [foaf:thumbnail](#)

The class [foaf:Image](#) is a sub-class of [foaf:Document](#) corresponding to those documents which are images.

Digital images (such as JPEG, PNG, GIF bitmaps, SVG diagrams etc.) are examples of [foaf:Image](#).

[\[back to top\]](#)

## Class: foaf:OnlineAccount

*Online Account* - An online account.

**Status:** unstable

**in-range-of:** [foaf:holdsAccount](#)

**in-domain-of:** [foaf:accountServiceHomepage](#) [foaf:accountName](#)

A [foaf:OnlineAccount](#) represents the provision of some form of online service, by some party (indicated indirectly via a [foaf:accountServiceHomepage](#)) to some [foaf:Agent](#). The [foaf:holdsAccount](#) property of the agent is used to indicate accounts that are associated with the agent.

See [foaf:OnlineChatAccount](#) for an example. Other sub-classes include [foaf:OnlineEcommerceAccount](#) and [foaf:OnlineGamingAccount](#).

[\[back to top\]](#)

## Class: foaf:OnlineChatAccount

*Online Chat Account* - An online chat account.

**Status:** unstable

A [foaf:OnlineChatAccount](#) is a [foaf:OnlineAccount](#) devoted to chat / instant messaging.

This is a generalization of the FOAF Chat ID properties, [foaf:jabberID](#), [foaf:aimChatID](#), [foaf:msnChatID](#), [foaf:icqChatID](#) and [foaf:yahooChatID](#).

Unlike those simple properties, [foaf:OnlineAccount](#) and associated FOAF terms allows us to describe a great variety of online accounts, without having to anticipate them in the FOAF vocabulary.

For example, here is a description of an IRC chat account, specific to the Freenode IRC network:

```
<foaf:Person>
  <foaf:name>Dan Brickley</foaf:name>
  <foaf:holdsAccount>
    <foaf:OnlineAccount>
      <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/OnlineChatAccount"/>
      <foaf:accountServiceHomepage rdf:resource="http://www.freenode.net/irc_servers.shtml"/>
      <foaf:accountName>danbri</foaf:accountName>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
</foaf:Person>
```

Note that it may be impolite to carelessly reveal someone else's chat identifier (which might also serve as an indicate of email address) As with email, there are privacy and anti-SPAM considerations. FOAF does not currently provide a way to represent an obfuscated chat ID (ie. there is no parallel to the [foaf:mbox](#) / [foaf:mbox\\_sha1sum](#) mapping).

In addition to the generic [foaf:OnlineAccount](#) and [foaf:OnlineChatAccount](#) mechanisms, FOAF also provides several convenience chat ID properties ([foaf:jabberID](#), [foaf:aimChatID](#), [foaf:icqChatID](#), [foaf:msnChatID](#), [foaf:yahooChatID](#)). These serve as as a shorthand for some common cases; their use may not always be appropriate.

We should specify some mappings between the abbreviated and full representations of [Jabber](#), [AIM](#), [MSN](#), [ICQ](#), [Yahoo!](#) and [MSN](#) chat accounts. This requires us to identify an appropriate [foaf:accountServiceHomepage](#) for each. If we wanted to make the [foaf:OnlineAccount](#) mechanism even more generic, we could invent a relationship that holds between a [foaf:OnlineAccount](#) instance and a convenience property. To continue the example above, we could describe how [Freenode](#) could define a property 'fn:freenodeChatID' corresponding to Freenode online accounts.

[\[back to top\]](#)

## Class: foaf:OnlineEcommerceAccount

*Online E-commerce Account* - An online e-commerce account.

**Status:** unstable

A [foaf:OnlineEcommerceAccount](#) is a [foaf:OnlineAccount](#) devoted to buying and/or selling of goods, services etc. Examples include [Amazon](#), [eBay](#), [PayPal](#), [thinkgeek](#), etc.

[\[back to top\]](#)

## Class: foaf:OnlineGamingAccount

*Online Gaming Account* - An online gaming account.

**Status:** unstable

A [foaf:OnlineGamingAccount](#) is a [foaf:OnlineAccount](#) devoted to online gaming.

Examples might include [EverQuest](#), [Xbox live](#), [Neverwinter Nights](#), etc., as well as older text-based systems (MOOs, MUDs and suchlike).

[\[back to top\]](#)

## Class: foaf:Organization

*Organization* - An organization.

**Status:** unstable

The [foaf:Organization](#) class represents a kind of [foaf:Agent](#) corresponding to social institutions such as companies, societies etc.

This is a more 'solid' class than [foaf:Group](#), which allows for more ad-hoc collections of individuals. These terms are rather roughly defined, and further work is needed to clearly specify their inter-relationships.

[\[back to top\]](#)

## Class: foaf:Person

*Person* - A person.

**Status:** testing

**in-range-of:** [foaf:knows](#)

**in-domain-of:** [foaf:geekcode](#) [foaf:firstName](#) [foaf:surname](#) [foaf:family\\_name](#) [foaf:plan](#) [foaf:img](#) [foaf:myersBriggs](#) [foaf:workplaceHomepage](#) [foaf:workInfoHomepage](#) [foaf:schoolHomepage](#) [foaf:knows](#) [foaf:interest](#) [foaf:topic\\_interest](#) [foaf:publications](#) [foaf:currentProject](#) [foaf:pastProject](#)

The [foaf:Person](#) class represents people. Something is a [foaf:Person](#) if it is a person. We don't nitpic about whether they're alive, dead, real, or imaginary.



The [foaf:Person](#) class is a sub-class of the [foaf:Agent](#) class, since all people are considered 'agents' in FOAF.

[\[back to top\]](#)

#### Class: foaf:PersonalProfileDocument

*PersonalProfileDocument* - A personal profile RDF document.

**Status:** testing

No detailed documentation for this term.

[\[back to top\]](#)

#### Class: foaf:Project

*Project* - A project (a collective endeavour of some kind).

**Status:** unstable

The [foaf:Project](#) class represents the class of things that are 'projects'. These may be formal or informal, collective or individual. It is often useful to indicate the [foaf:homepage](#) of a [foaf:Project](#).

Further work is needed to specify the connections between this class and the FOAF properties [foaf:currentProject](#) and [foaf:pastProject](#).

[\[back to top\]](#)

#### Property: foaf:accountName

*account name* - Indicates the name (identifier) associated with this online account.

**Status:** unstable

**Domain:** [foaf:OnlineAccount](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:accountName](#) property of a [foaf:OnlineAccount](#) is a textual representation of the account name (unique ID) associated with that account.

[\[back to top\]](#)

#### Property: foaf:accountServiceHomepage

*account service homepage* - Indicates a homepage of the service provide for this online account.

**Status:** unstable

**Domain:** [foaf:OnlineAccount](#)

**Range:** [foaf:Document](#)

The [foaf:accountServiceHomepage](#) property indicates a relationship between a [foaf:OnlineAccount](#) and the homepage of the supporting service provider.

[\[back to top\]](#)

#### Property: foaf:aimChatID

*AIM chat ID* - An AIM chat ID

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:aimChatID](#) property relates a [foaf:Agent](#) to a textual identifier ('screenname') assigned to them in the AOL Instant Messenger (AIM) system. See AOL's [AIM](#) site for more details of AIM and AIM screennames. The [iChat](#) tools from [Apple](#) also make use of AIM identifiers.

See [foaf:OnlineChatAccount](#) (and [foaf:OnlineAccount](#)) for a more general (and verbose) mechanism for describing IM and chat accounts.

[\[back to top\]](#)

#### Property: foaf:based\_near

*based near* - A location that something is based near, for some broadly human notion of near.

**Status:** unstable

**Domain:** [http://www.w3.org/2003/01/geo/wgs84\\_pos#SpatialThing](http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing)

**Range:** [http://www.w3.org/2003/01/geo/wgs84\\_pos#SpatialThing](http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing)

The [foaf:based\\_near](#) relationship relates two "spatial things" (anything that can *be somewhere*), the latter typically described using the geo:lat / geo:long [geo-positioning vocabulary](#) (See [GeoInfo](#) in the W3C semweb wiki for details). This allows us to say describe the typical latitude and longitude of, say, a Person (people are spatial things - they can be places) without implying that a precise location has been given.

We do not say much about what 'near' means in this context; it is a 'rough and ready' concept. For a more precise treatment, see [GeoOnion vocab](#) design discussions, which are aiming to produce a more sophisticated vocabulary for such purposes.

FOAF files often make use of the `contact:nearestAirport` property. This illustrates the distinction between FOAF documents (which may make claims using *any* RDF vocabulary) and the core FOAF vocabulary defined by this specification. For further reading on the use of `nearestAirport` see [UsingContactNearestAirport](#) in the FOAF wiki.

[\[back to top\]](#)

## Property: foaf:birthday

*birthday* - The birthday of this Agent, represented in mm-dd string form, eg. '12-31'.

**Status:** unstable

**Domain:** [foaf:Agent](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:birthday](#) property is a relationship between a [foaf:Agent](#) and a string representing the month and day in which they were born (Gregorian calendar). See [BirthdayIssue](#) for details of related properties that can be used to describe such things in more flexible ways.

[\[back to top\]](#)

## Property: foaf:currentProject

*current project* - A current project this person works on.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

A [foaf:currentProject](#) relates a [foaf:Person](#) to a [foaf:Document](#) indicating some collaborative or individual undertaking. This relationship indicates that the [foaf:Person](#) has some active role in the project, such as development, coordination, or support.

When a [foaf:Person](#) is no longer involved with a project, or perhaps is inactive for some time, the relationship becomes a [foaf:pastProject](#).

If the [foaf:Person](#) has stopped working on a project because it has been completed (successfully or otherwise), [foaf:pastProject](#) is applicable. In general, [foaf:currentProject](#) is used to indicate someone's current efforts (and implied interests, concerns etc.), while [foaf:pastProject](#) describes what they've previously been doing.

Note that this property requires further work. There has been confusion about whether it points to a thing (eg. something you've made; a homepage for a project, ie. a [foaf:Document](#) or to instances of the class [foaf:Project](#), which might themselves have a [foaf:homepage](#). In practice, it seems to have been used in a similar way to [foaf:interest](#), referencing homepages of ongoing projects.

[\[back to top\]](#)

## Property: foaf:depiction

*depiction* - A depiction of some thing.

**Status:** testing

**Domain:** <http://www.w3.org/2000/01/rdf-schema#Resource>

**Range:** [foaf:Image](#)

The [foaf:depiction](#) property is a relationship between a thing and an [foaf:Image](#) that depicts it. As such it is an inverse of the [foaf:depicts](#) relationship.

A common use of [foaf:depiction](#) (and [foaf:depicts](#)) is to indicate the contents of a digital image, for example the people or objects represented in an online photo gallery.

Extensions to this basic idea include '[Co-Depiction](#)' (social networks as evidenced in photos), as well as richer photo metadata through the mechanism of using SVG paths to indicate the *regions* of an image which depict some particular thing. See '[Annotating Images With SVG](#)' for tools and details.

The basic notion of 'depiction' could also be extended to deal with multimedia content (video clips, audio), or refined to deal with corner cases, such as pictures of pictures etc.

The [foaf:depiction](#) property is a super-property of the more specific property [foaf:img](#), which is used more sparingly. You stand in a [foaf:depiction](#) relation to *any* [foaf:Image](#) that depicts you, whereas [foaf:img](#) is typically used to indicate a few images that are particularly representative.

[\[back to top\]](#)

## Property: foaf:depicts

*depicts* - A thing depicted in this representation.

**Status:** testing

**Domain:** [foaf:Image](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

The [foaf:depicts](#) property is a relationship between a [foaf:Image](#) and something that the image depicts. As such it is an inverse of the [foaf:depiction](#) relationship. See [foaf:depiction](#) for further notes.

[\[back to top\]](#)

## Property: foaf:dnaChecksum

*DNA checksum* - A checksum for the DNA of some thing. Joke.

**Status:** unstable

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:dnaChecksum](#) property is mostly a joke, but also a reminder that there will be lots of different identifying properties for people, some of which we might find disturbing.

[\[back to top\]](#)

## Property: foaf:family\_name

*family\_name* - The family\_name of some person.



**Status:** testing  
**Domain:** [foaf:Person](#)  
**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

A number of naming constructs are under development to provide naming substructure; draft properties include [foaf:firstName](#), [foaf:givenname](#), and [foaf:surname](#). These are not currently stable or consistent; see the [issue tracker](#) for design discussions, status and ongoing work on rationalising the FOAF naming machinery.

There is also a simple [foaf:name](#) property.

[\[back to top\]](#)

## Property: foaf:firstName

*firstName* - The first name of a person.

**Status:** testing  
**Domain:** [foaf:Person](#)  
**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

A number of naming constructs are under development to provide naming substructure; draft properties include [foaf:firstName](#), [foaf:givenname](#), and [foaf:surname](#). These are not currently stable or consistent; see the [issue tracker](#) for design discussions, status and ongoing work on rationalising the FOAF naming machinery.

There is also a simple [foaf:name](#) property.

[\[back to top\]](#)

## Property: foaf:fundedBy

*funded by* - An organization funding a project or person.

**Status:** unstable  
**Domain:** <http://www.w3.org/2000/01/rdf-schema#Resource>  
**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

The [foaf:fundedBy](#) property relates something to something else that has provided funding for it.

This property is under-specified, experimental, and should be considered liable to change.

[\[back to top\]](#)

## Property: foaf:geekcode

*geekcode* - A textual geekcode for this person, see <http://www.geekcode.com/geek.html>

**Status:** testing  
**Domain:** [foaf:Person](#)  
**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:geekCode](#) property is used to represent a 'Geek Code' for some [foaf:Person](#).

See the [Code of the Geeks](#) specification for details of the code, which provides a somewhat frivolous and willfully obscure mechanism for characterising technical expertise, interests and habits. The [foaf:geekCode](#) property is not bound to any particular version of the code. The last published version of the code was v3.12 in March 1996.

As the [Geek Code](#) website notes, the code played a small (but amusing) part in the history of the Internet. The [foaf:geekCode](#) property exists in acknowledgement of this history. It'll never be 1996 again.

Note that the Geek Code is a densely packed collections of claims about the person it applies to; to express these claims explicitly in RDF/XML would be incredibly verbose. The syntax of the Geek Code allows for '<' and '>' characters, which have special meaning in RDF/XML. Consequently these should be carefully escaped in markup.

An example Geek Code:

```
GED/J d-- s:+++; a-- C++(++++ ) ULU++ P+ L++ E---- W+(-) N+++ o+ K+++ w--- O- M+ V-- PS++>$ PE++>$ Y++ PGP++ t- 5+++ X++ R+++>$ tv+ b+
DI+++ D+++ G++++ e++ h r-- y+++*
```

...would be written in FOAF RDF/XML as follows:

```
<foaf:geekCode> GED/J d-- s:+++&gt;; a-- C++(++++ ) ULU++ P+ L++ E---- W+(-) N+++ o+ K+++ w--- O- M+ V-- PS++&gt;$ PE++&gt;$ Y++ PGP++ t-
5+++ X++ R+++&gt;$ tv+ b+ DI+++ D+++ G++++ e++ h r-- y+++* </foaf:geekCode>
```

See also the [geek code](#) entry in [everything2](#), which tells us that *the geek code originated in 1993; it was inspired (according to the inventor) by previous "bear", "smurf" and "twink" style-and-sexual-preference codes from lesbian and gay newsgroups*. There is also a [Geek Code Decoder Page](#) and a form-based [generator](#).

[\[back to top\]](#)

## Property: foaf:gender

*gender* - The gender of this Agent (typically but not necessarily 'male' or 'female').

**Status:** testing  
**Domain:** [foaf:Agent](#)  
**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:gender](#) property relates a [foaf:Agent](#) (typically a [foaf:Person](#)) to a string representing its gender. In most cases the value will be the string 'female' or 'male' (in lowercase without surrounding quotes or spaces). Like all FOAF properties, there is in general no requirement to use [foaf:gender](#) in any particular document or description. Values other than 'male' and 'female' may be used, but are not enumerated here. The [foaf:gender](#) mechanism is

not intended to capture the full variety of biological, social and sexual concepts associated with the word 'gender'.

Anything that has a [foaf:gender](#) property will be some kind of [foaf:Agent](#). However there are kinds of [foaf:Agent](#) to which the concept of gender isn't applicable (eg. a [foaf:Group](#)). FOAF does not currently include a class corresponding directly to "the type of thing that has a gender". At any point in time, a [foaf:Agent](#) has at most one value for [foaf:gender](#). FOAF does not treat [foaf:gender](#) as a *static* property; the same individual may have different values for this property at different times.

Note that FOAF's notion of gender isn't defined biologically or anatomically - this would be tricky since we have a broad notion that applies to all [foaf:Agent](#)s (including robots - eg. Bender from Futurama is 'male'). As stressed above, FOAF's notion of gender doesn't attempt to encompass the full range of concepts associated with human gender, biology and sexuality. As such it is a (perhaps awkward) compromise between the clinical and the social/psychological. In general, a person will be the best authority on their [foaf:gender](#). Feedback on this design is particularly welcome (via the FOAF mailing list, [rdfweb-dev](#)). We have tried to be respectful of diversity without attempting to catalogue or enumerate that diversity.

This may also be a good point for a periodic reminder: as with all FOAF properties, documents that use '[foaf:gender](#)' will on occasion be inaccurate, misleading or outright false. FOAF, like all open means of communication, supports *lying*. Application authors using FOAF data should always be cautious in their presentation of unverified information, but be particularly sensitive to issues and risks surrounding sex and gender (including privacy and personal safety concerns). Designers of FOAF-based user interfaces should be careful to allow users to omit [foaf:gender](#) when describing themselves and others, and to allow at least for values other than 'male' and 'female' as options. Users of information conveyed via FOAF (as via information conveyed through mobile phone text messages, email, Internet chat, HTML pages etc.) should be skeptical of unverified information.

[\[back to top\]](#)

#### Property: foaf:givenname

*Given name* - The given name of some person.

**Status:** testing

A number of naming constructs are under development to provide naming substructure; draft properties include [foaf:firstName](#), [foaf:givenname](#), and [foaf:surname](#). These are not currently stable or consistent; see the [issue tracker](#) for design discussions, status and ongoing work on rationalising the FOAF naming machinery.

There is also a simple [foaf:name](#) property.

[\[back to top\]](#)

#### Property: foaf:holdsAccount

*holds account* - Indicates an account held by this agent.

**Status:** unstable

**Domain:** [foaf:Agent](#)

**Range:** [foaf:OnlineAccount](#)

The [foaf:holdsAccount](#) property relates a [foaf:Agent](#) to an [foaf:OnlineAccount](#) for which they are the sole account holder. See [foaf:OnlineAccount](#) for usage details.

[\[back to top\]](#)

#### Property: foaf:homepage

*homepage* - A homepage for some thing.

**Status:** stable

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** <http://www.w3.org/2000/01/rdf-schema#Resource>

**Range:** [foaf:Document](#)

The [foaf:homepage](#) property relates something to a homepage about it.

Many kinds of things have homepages. FOAF allows a thing to have multiple homepages, but constrains [foaf:homepage](#) so that there can be only one thing that has any particular homepage.

A 'homepage' in this sense is a public Web document, typically but not necessarily available in HTML format. The page has as a [foaf:topic](#) the thing whose homepage it is. The homepage is usually controlled, edited or published by the thing whose homepage it is; as such one might look to a homepage for information on its owner from its owner. This works for people, companies, organisations etc.

The [foaf:homepage](#) property is a sub-property of the more general [foaf:page](#) property for relating a thing to a page about that thing. See also [foaf:topic](#), the inverse of the [foaf:page](#) property.

[\[back to top\]](#)

#### Property: foaf:icqChatID

*ICQ chat ID* - An ICQ chat ID

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:icqChatID](#) property relates a [foaf:Agent](#) to a textual identifier assigned to them in the ICQ Chat system. See the [icq chat](#) site for more details of the 'icq' service. Their "[What is ICQ?](#)" document provides a basic overview, while their "[About Us](#)" page notes that ICQ has been acquired by AOL. Despite the relationship with AOL, ICQ is at the time of writing maintained as a separate identity from the AIM brand (see [foaf:aimChatID](#)).

See [foaf:OnlineChatAccount](#) (and [foaf:OnlineAccount](#)) for a more general (and verbose) mechanism for describing IM and chat accounts.

[\[back to top\]](#)

#### Property: foaf:img

*image* - An image that can be used to represent some thing (ie. those depictions which are particularly representative of something, eg. one's photo on a homepage).

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** [foaf:Image](#)

The [foaf:img](#) property relates a [foaf:Person](#) to a [foaf:Image](#) that represents them. Unlike its super-property [foaf:depiction](#), we only use [foaf:img](#) when an image is particularly representative of some person. The analogy is with the image(s) that might appear on someone's homepage, rather than happen to appear somewhere in their photo album.

Unlike the more general [foaf:depiction](#) property (and its inverse, [foaf:depicts](#)), the [foaf:img](#) property is only used with representations of people (ie. instances of [foaf:Person](#)). So you can't use it to find pictures of cats, dogs etc. The basic idea is to have a term whose use is more restricted than [foaf:depiction](#) so we can have a useful way of picking out a reasonable image to represent someone. FOAF defines [foaf:img](#) as a sub-property of [foaf:depiction](#), which means that the latter relationship is implied whenever two things are related by the former.

Note that [foaf:img](#) does not have any restrictions on the dimensions, colour depth, format etc of the [foaf:Image](#) it references.

Terminology: note that [foaf:img](#) is a property (ie. relationship), and that `code:Image` is a similarly named class (ie. category, a type of thing). It might have been more helpful to call [foaf:img](#) 'mugshot' or similar; instead it is named by analogy to the HTML IMG element.

[\[back to top\]](#)

## Property: foaf:interest

*interest* - A page about a topic of interest to this person.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** [foaf:Document](#)

The [foaf:interest](#) property represents an interest of a [foaf:Agent](#), through indicating a [foaf:Document](#) whose [foaf:topic](#)(s) broadly characterises that interest.

For example, we might claim that a person or group has an interest in RDF by saying they stand in a [foaf:interest](#) relationship to the [RDF](#) home page. Loosely, such RDF would be saying "*this agent is interested in the topic of this page*".

Uses of [foaf:interest](#) include a variety of filtering and resource discovery applications. It could be used, for example, to help find answers to questions such as "Find me members of this organisation with an interest in XML who have also contributed to [CPAN](#)".

This approach to characterising interests is intended to compliment other mechanisms (such as the use of controlled vocabulary). It allows us to use a widely known set of unique identifiers (Web page URIs) with minimal pre-coordination. Since URIs have a controlled syntax, this makes data merging much easier than the use of free-text characterisations of interest.

Note that interest does not imply expertise, and that this FOAF term provides no support for characterising levels of interest: passing fads and lifelong quests are both examples of someone's [foaf:interest](#). Describing interests in full is a complex undertaking; [foaf:interest](#) provides one basic component of FOAF's approach to these problems.

[\[back to top\]](#)

## Property: foaf:jabberID

*jabber ID* - A jabber ID for something.

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:jabberID](#) property relates a [foaf:Agent](#) to a textual identifier assigned to them in the [Jabber](#) messaging system. See the [Jabber](#) site for more information about the Jabber protocols and tools.

Jabber, unlike several other online messaging systems, is based on an open, publically documented protocol specification, and has a variety of open source implementations. Jabber IDs can be assigned to a variety of kinds of thing, including software 'bots', chat rooms etc. For the purposes of FOAF, these are all considered to be kinds of [foaf:Agent](#) (ie. things that *do* stuff). The uses of Jabber go beyond simple IM chat applications. The [foaf:jabberID](#) property is provided as a basic hook to help support RDF description of Jabber users and services.

See [foaf:OnlineChatAccount](#) (and [foaf:OnlineAccount](#)) for a more general (and verbose) mechanism for describing IM and chat accounts.

[\[back to top\]](#)

## Property: foaf:knows

*knows* - A person known by this person (indicating some level of reciprocated interaction between the parties).

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** [foaf:Person](#)

The [foaf:knows](#) property relates a [foaf:Person](#) to another [foaf:Person](#) that he or she knows.

We take a broad view of 'knows', but do require some form of reciprocated interaction (ie. stalkers need not apply). Since social attitudes and conventions on this topic vary greatly between communities, counties and cultures, it is not appropriate for FOAF to be overly-specific here.

If someone [foaf:knows](#) a person, it would be usual for the relation to be reciprocated. However this doesn't mean that there is any obligation for either party to publish FOAF describing this relationship. A [foaf:knows](#) relationship does not imply friendship, endorsement, or that a face-to-face meeting has taken place: phone, fax, email, and smoke signals are all perfectly acceptable ways of communicating with people you know.

You probably know hundreds of people, yet might only list a few in your public FOAF file. That's OK. Or you might list them all. It is perfectly fine to have a FOAF file and not list anyone else in it at all. This illustrates the Semantic Web principle of partial description: RDF documents rarely describe the entire picture. There is always more to be said, more information living elsewhere in the Web (or in our heads...).

Since [foaf:knows](#) is vague by design, it may be suprising that it has uses. Typically these involve combining other RDF properties. For example, an

application might look at properties of each [foaf:weblog](#) that was [foaf:made](#) by someone you "[foaf:knows](#)". Or check the newsfeed of the online photo archive for each of these people, to show you recent photos taken by people you know.

To provide additional levels of representation beyond mere 'knows', FOAF applications can do several things.

They can use more precise relationships than [foaf:knows](#) to relate people to people. The original FOAF design included two of these ('knowsWell', 'friend') which we removed because they were somewhat *awkward* to actually use, bringing an inappropriate air of precision to an intrinsically vague concept. Other extensions have been proposed, including Eric Vitiello's [Relationship module](#) for FOAF.

In addition to using more specialised inter-personal relationship types (eg [rel:acquaintanceOf](#) etc) it is often just as good to use RDF descriptions of the states of affairs which imply particular kinds of relationship. So for example, two people who have the same value for their [foaf:workplaceHomepage](#) property are typically colleagues. We don't (currently) clutter FOAF up with these extra relationships, but the facts can be written in FOAF nevertheless. Similarly, if there exists a [foaf:Document](#) that has two people listed as its [foaf:makes](#), then they are probably collaborators of some kind. Or if two people appear in 100s of digital photos together, there's a good chance they're friends and/or colleagues.

So FOAF is quite pluralistic in its approach to representing relationships between people. FOAF is built on top of a general purpose machine language for representing relationships (ie. RDF), so is quite capable of representing any kinds of relationship we care to add. The problems are generally social rather than technical; deciding on appropriate ways of describing these interconnections is a subtle art.

Perhaps the most important use of [foaf:knows](#) is, alongside the [rdfs:seeAlso](#) property, to connect FOAF files together. Taken alone, a FOAF file is somewhat dull. But linked in with 1000s of other FOAF files it becomes more interesting, with each FOAF file saying a little more about people, places, documents, things... By mentioning other people (via [foaf:knows](#) or other relationships), and by providing an [rdfs:seeAlso](#) link to their FOAF file, you can make it easy for FOAF indexing tools ('[scutters](#)') to find your FOAF and the FOAF of the people you've mentioned. And the FOAF of the people they mention, and so on. This makes it possible to build FOAF aggregators without the need for a centrally managed directory of FOAF files...

[\[back to top\]](#)

## Property: foaf:logo

*logo* - A logo representing some thing.

**Status:** testing

**Domain:** [http://www.w3.org/2000/01/rdf-schema#Resource](#)

**Range:** [http://www.w3.org/2000/01/rdf-schema#Resource](#)

The [foaf:logo](#) property is used to indicate a graphical logo of some kind. *It is probably underspecified...*

[\[back to top\]](#)

## Property: foaf:made

*made* - Something that was made by this agent.

**Status:** testing

**Domain:** [foaf:Agent](#)

**Range:** [http://www.w3.org/2000/01/rdf-schema#Resource](#)

The [foaf:made](#) property relates a [foaf:Agent](#) to something [foaf:made](#) by it. As such it is an inverse of the [foaf:maker](#) property, which relates a thing to something that made it. See [foaf:made](#) for more details on the relationship between these FOAF terms and related Dublin Core vocabulary.

[\[back to top\]](#)

## Property: foaf:maker

*maker* - An agent that made this thing.

**Status:** testing

**Domain:** [http://www.w3.org/2000/01/rdf-schema#Resource](#)

**Range:** [foaf:Agent](#)

The [foaf:maker](#) property relates something to a [foaf:Agent](#) that [foaf:made](#) it. As such it is an inverse of the [foaf:made](#) property.

The [foaf:name](#) (or other [rdfs:label](#)) of the [foaf:maker](#) of something can be described as the [dc:creator](#) of that thing.

For example, if the thing named by the URI [http://rdfweb.org/people/danbri/](#) has a [foaf:maker](#) that is a [foaf:Person](#) whose [foaf:name](#) is 'Dan Brickley', we can conclude that [http://rdfweb.org/people/danbri/](#) has a [dc:creator](#) of 'Dan Brickley'.

FOAF descriptions are encouraged to use [dc:creator](#) only for simple textual names, and to use [foaf:maker](#) to indicate creators, rather than risk confusing creators with their names. This follows most Dublin Core usage. See [UsingDublinCoreCreator](#) for details.

[\[back to top\]](#)

## Property: foaf:mailbox

*personal mailbox* - A personal mailbox, ie. an Internet mailbox associated with exactly one owner, the first owner of this mailbox. This is a 'static inverse functional property', in that there is (across time and change) at most one individual that ever has any particular value for [foaf:mailbox](#).

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** [http://www.w3.org/2000/01/rdf-schema#Resource](#)

The [foaf:mailbox](#) property is a relationship between the owner of a mailbox and a mailbox. These are typically identified using the [mailto:](#) URI scheme (see [RFC 2368](#)).

Note that there are many mailboxes (eg. shared ones) which are not the [foaf:mailbox](#) of anyone. Furthermore, a person can have multiple [foaf:mailbox](#) properties.

In FOAF, we often see [foaf:mailbox](#) used as an indirect way of identifying its owner. This works even if the mailbox is itself out of service (eg. 10 years old), since the property is defined in terms of its primary owner, and doesn't require the mailbox to actually be being used for anything.

Many people are wary of sharing information about their mailbox addresses in public. To address such concerns whilst continuing the FOAF convention of

indirectly identifying people by referring to widely known properties, FOAF also provides the [foaf:mbox\\_sha1sum](#) mechanism, which is a relationship between a person and the value you get from passing a mailbox URI to the SHA1 mathematical function.

[\[back to top\]](#)

#### Property: foaf:mbox\_sha1sum

*sha1sum of a personal mailbox URI name* - The sha1sum of the URI of an Internet mailbox associated with exactly one owner, the first owner of the mailbox.

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

A [foaf:mbox\\_sha1sum](#) of a [foaf:Person](#) is a textual representation of the result of applying the SHA1 mathematical functional to a 'mailto:' identifier (URI) for an Internet mailbox that they stand in a [foaf:mbox](#) relationship to.

In other words, if you have a mailbox ([foaf:mbox](#)) but don't want to reveal its address, you can take that address and generate a [foaf:mbox\\_sha1sum](#) representation of it. Just as a [foaf:mbox](#) can be used as an indirect identifier for its owner, we can do the same with [foaf:mbox\\_sha1sum](#) since there is only one [foaf:Person](#) with any particular value for that property.

Many FOAF tools use [foaf:mbox\\_sha1sum](#) in preference to exposing mailbox information. This is usually for privacy and SPAM-avoidance reasons. Other relevant techniques include the use of PGP encryption (see [Edd Dumbill's documentation](#)) and the use of [FOAF-based whitelists](#) for mail filtering.

Code examples for SHA1 in C#, Java, PHP, Perl and Python can be found [in Sam Ruby's weblog entry](#). Remember to include the 'mailto:' prefix, but no trailing whitespace, when computing a [foaf:mbox\\_sha1sum](#) property.

[\[back to top\]](#)

#### Property: foaf:member

*member* - Indicates a member of a Group

**Status:** unstable

**Domain:** [foaf:Group](#)

**Range:** [foaf:Agent](#)

The [foaf:member](#) property relates a [foaf:Group](#) to a [foaf:Agent](#) that is a member of that group.

See [foaf:Group](#) for details and examples.

[\[back to top\]](#)

#### Property: foaf:membershipClass

*membershipClass* - Indicates the class of individuals that are a member of a Group

**Status:** unstable

**Domain:** [foaf:Group](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Class>

The [foaf:membershipClass](#) property relates a [foaf:Group](#) to an RDF class representing a sub-class of [foaf:Agent](#) whose instances are all the agents that are a [foaf:member](#) of the [foaf:Group](#).

See [foaf:Group](#) for details and examples.

[\[back to top\]](#)

#### Property: foaf:msnChatID

*MSN chat ID* - An MSN chat ID

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:msnChatID](#) property relates a [foaf:Agent](#) to a textual identifier assigned to them in the MSN online Chat system. See Microsoft's the [MSN chat](#) site for more details (or for a message saying "MSN Chat is not currently compatible with your Internet browser and/or computer operating system" if your computing platform is deemed unsuitable).

It is not currently clear how MSN chat IDs relate to the more general Microsoft Passport identifiers.

See [foaf:OnlineChatAccount](#) (and [foaf:OnlineAccount](#)) for a more general (and verbose) mechanism for describing IM and chat accounts.

[\[back to top\]](#)

#### Property: foaf:myersBriggs

*myersBriggs* - A Myers Briggs (MBTI) personality classification.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:myersBriggs](#) property represents the Myers Briggs (MBTI) approach to personality taxonomy. It is included in FOAF as an example of a property that takes certain constrained values, and to give some additional detail to the FOAF files of those who choose to include it. The [foaf:myersBriggs](#) property applies only to the [foaf:Person](#) class; wherever you see it, you can infer it is being applied to a person.

The [foaf:myersBriggs](#) property is interesting in that it illustrates how FOAF can serve as a carrier for various kinds of information, without necessarily being committed to any associated worldview. Not everyone will find myersBriggs (or star signs, or blood types, or the four humours) a useful perspective on human behaviour and personality. The inclusion of a Myers Briggs property doesn't indicate that FOAF endorses the underlying theory, any more than the existence of [foaf:weblog](#) is an endorsement of soapboxes.

The values for [foaf:myersBriggs](#) are the following 16 4-letter textual codes: ESTJ, INFP, ESFP, INTJ, ESFJ, INTP, ENFP, ISTJ, ESTP, INFJ, ENFJ, ISTP, ENTJ, ISFP, ENTP, ISFJ. If multiple of these properties are applicable, they are represented by applying multiple properties to a person.

For further reading on MBTI, see various online sources (eg. [this article](#)). There are various online sites which offer quiz-based tools for determining a person's MBTI classification. The owners of the MBTI trademark have probably not approved of these.

This FOAF property suggests some interesting uses, some of which could perhaps be used to test the claims made by proponents of the MBTI (eg. an analysis of weblog postings filtered by MBTI type). However it should be noted that MBTI FOAF descriptions are self-selecting; MBTI categories may not be uniformly appealing to the people they describe. Further, there is probably a degree of cultural specificity implicit in the assumptions made by many questionnaire-based MBTI tools; the MBTI system may not make sense in cultural settings beyond those it was created for.

See also [Cory Caplinger's summary table](#) or the RDFWeb article, [FOAF Myers Briggs addition](#) for further background and examples.

Note: Myers Briggs Type Indicator and MBTI are registered trademarks of Consulting Psychologists Press Inc. Oxford Psychologists Press Ltd has exclusive rights to the trademark in the UK.

[\[back to top\]](#)

## Property: foaf:name

*name* - A name for some thing.

**Status:** testing

**Domain:** <http://www.w3.org/2000/01/rdf-schema#Resource>

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:name](#) of something is a simple textual string.

XML language tagging may be used to indicate the language of the name. For example:

```
<foaf:name xml:lang="en">Dan Brickley</foaf:name>
```

FOAF provides some other naming constructs. These are under development. While foaf:name does not explicitly represent name substructure (family vs given etc.) it does provide a basic level of interoperability. See the [RDFWeb/FOAF issue tracker](#) for status of work on this issue.

[\[back to top\]](#)

## Property: foaf:nick

*nickname* - A short informal nickname characterising an agent (includes login identifiers, IRC and other chat nicknames).

**Status:** testing

The [foaf:nick](#) property relates a [foaf:Person](#) to a short (often abbreviated) nickname, such as those use in IRC chat, online accounts, and computer logins.

This property is necessarily vague, because it does not indicate any particular naming control authority, and so cannot distinguish a person's login from their (possibly various) IRC nicknames or other similar identifiers. However it has some utility, since many people use the same string (or slight variants) across a variety of such environments.

For specific controlled sets of names (relating primarily to Instant Messenger accounts), FOAF provides some convenience properties: [foaf:jabberID](#), [foaf:aimChatID](#), [foaf:msnChatID](#) and [foaf:icqChatID](#). Beyond this, the problem of representing such accounts is not peculiar to Instant Messaging, and it is not scalable to attempt to enumerate each naming database as a distinct FOAF property. The [foaf:OnlineAccount](#) term (and supporting vocabulary) are provided as a more verbose and more expressive generalisation of these properties.

[\[back to top\]](#)

## Property: foaf:page

*page* - A page or document about this thing.

**Status:** testing

**Domain:** <http://www.w3.org/2000/01/rdf-schema#Resource>

**Range:** [foaf:Document](#)

The [foaf:page](#) property relates a thing to a document about that thing.

As such it is an inverse of the [foaf:topic](#) property, which relates a document to a thing that the document is about.

[\[back to top\]](#)

## Property: foaf:pastProject

*past project* - A project this person has previously worked on.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

After a [foaf:Person](#) is no longer involved with a [foaf:currentProject](#), or has been inactive for some time, a [foaf:pastProject](#) relationship can be used. This indicates that the [foaf:Person](#) was involved with the described project at one point.

If the [foaf:Person](#) has stopped working on a project because it has been completed (successfully or otherwise), [foaf:pastProject](#) is applicable. In general, [foaf:currentProject](#) is used to indicate someone's current efforts (and implied interests, concerns etc.), while [foaf:pastProject](#) describes what they've previously been doing.

[\[back to top\]](#)

**Property: foaf:phone**

*phone* - A phone, specified using fully qualified tel: URI scheme (refs: <http://www.w3.org/Addressing/schemes.html#tel>).

**Status:** testing

The [foaf:phone](#) of something is a phone, typically identified using the tel: URI scheme.

[\[back to top\]](#)

**Property: foaf:plan**

*plan* - A .plan comment, in the tradition of finger and '.plan' files.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:plan](#) property provides a space for a [foaf:Person](#) to hold some arbitrary content that would appear in a traditional '.plan' file. The plan file was stored in a user's home directory on a UNIX machine, and displayed to people when the user was queried with the finger utility.

A plan file could contain anything. Typical uses included brief comments, thoughts, or remarks on what a person had been doing lately. Plan files were also prone to being witty or simply obscure. Others may be more creative, writing any number of seemingly random compositions in their plan file for people to stumble upon.

See [History of the Finger Protocol](#) by Rajiv Shah for more on this piece of Internet history. The [foaf:geekCode](#) property may also be of interest.

[\[back to top\]](#)

**Property: foaf:primaryTopic**

*topic* - The primary topic of some page or document.

**Status:** testing

**Domain:** [foaf:Document](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

No detailed documentation for this term.

[\[back to top\]](#)

**Property: foaf:publications**

*publications* - A link to the publications of this person.

**Status:** unstable

**Domain:** [foaf:Person](#)

**Range:** [foaf:Document](#)

The [foaf:publications](#) property indicates a [foaf:Document](#) listing (primarily in human-readable form) some publications associated with the [foaf:Person](#). Such documents are typically published alongside one's [foaf:homepage](#).

[\[back to top\]](#)

**Property: foaf:schoolHomepage**

*schoolHomepage* - A homepage of a school attended by the person.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** [foaf:Document](#)

The [schoolHomepage](#) property relates a [foaf:Person](#) to a [foaf:Document](#) that is the [foaf:homepage](#) of a School that the person attended.

FOAF does not (currently) define a class for 'School' (if it did, it would probably be as a sub-class of [foaf:Organization](#)). The original application area for [foaf:schoolHomepage](#) was for 'schools' in the British-English sense; however American-English usage has dominated, and it is now perfectly reasonable to describe Universities, Colleges and post-graduate study using [foaf:schoolHomepage](#).

This very basic facility provides a basis for a low-cost, decentralised approach to classmate-reunion and suchlike. Instead of requiring a central database, we can use FOAF to express claims such as 'I studied *here*' simply by mentioning a school's homepage within FOAF files. Given the homepage of a school, it is easy for FOAF aggregators to lookup this property in search of people who attended that school.

[\[back to top\]](#)

**Property: foaf:sha1**

*sha1sum (hex)* - A sha1sum hash, in hex.

**Status:** unstable

**Domain:** [foaf:Document](#)

The [foaf:sha1](#) property relates a [foaf:Document](#) to the textual form of a SHA1 hash of (some representation of) its contents.

The design for this property is neither complete nor coherent. The [foaf:Document](#) class is currently used in a way that allows multiple instances at different URIs to have the 'same' contents (and hence hash). If [foaf:sha1](#) is an owl:InverseFunctionalProperty, we could deduce that several such documents were the self-same thing. A more careful design is needed, which distinguishes documents in a broad sense from byte sequences.

[\[back to top\]](#)



## Property: foaf:surname

*Surname* - The surname of some person.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

A number of naming constructs are under development to provide naming substructure; draft properties include [foaf:firstName](#), [foaf:givenname](#), and [foaf:surname](#). These are not currently stable or consistent; see the [issue tracker](#) for design discussions, status and ongoing work on rationalising the FOAF naming machinery.

There is also a simple [foaf:name](#) property.

[\[back to top\]](#)

## Property: foaf:theme

*theme* - A theme.

**Status:** unstable

**Domain:** <http://www.w3.org/2000/01/rdf-schema#Resource>

**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

The [foaf:theme](#) property is rarely used and under-specified. The intention was to use it to characterise interest / themes associated with projects and groups. Further work is needed to meet these goals.

[\[back to top\]](#)

## Property: foaf:thumbnail

*thumbnail* - A derived thumbnail image.

**Status:** testing

**Domain:** [foaf:Image](#)

**Range:** [foaf:Image](#)

The [foaf:thumbnail](#) property is a relationship between a full-size [foaf:Image](#) and a smaller, representative [foaf:Image](#) that has been derived from it.

It is typical in FOAF to express [foaf:img](#) and [foaf:depiction](#) relationships in terms of the larger, 'main' (in some sense) image, rather than its thumbnail(s). A [foaf:thumbnail](#) might be clipped or otherwise reduced such that it does not depict everything that the full image depicts. Therefore FOAF does not specify that a thumbnail [foaf:depicts](#) everything that the image it is derived from depicts. However, FOAF does expect that anything depicted in the thumbnail will also be depicted in the source image.

A [foaf:thumbnail](#) is typically small enough that it can be loaded and viewed quickly before a viewer decides to download the larger version. They are often used in online photo gallery applications.

[\[back to top\]](#)

## Property: foaf:tipjar

*tipjar* - A tipjar document for this agent, describing means for payment and reward.

**Status:** testing

**Domain:** [foaf:Agent](#)

**Range:** [foaf:Document](#)

No detailed documentation for this term.

[\[back to top\]](#)

## Property: foaf:title

*title* - Title (Mr, Mrs, Ms, Dr. etc)

**Status:** testing

The appropriate values for [foaf:title](#) are not formally constrained, and will vary across community and context. Values such as 'Mr', 'Mrs', 'Ms', 'Dr' etc. are expected.

[\[back to top\]](#)

## Property: foaf:topic

*topic* - A topic of some page or document.

**Status:** testing

**Domain:** [foaf:Document](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

The [foaf:topic](#) property relates a document to a thing that the document is about.

As such it is an inverse of the [foaf:page](#) property, which relates a thing to a document about that thing.

[\[back to top\]](#)

## Property: foaf:topic\_interest

*interest\_topic* - A thing of interest to this person.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Resource>

The [foaf:topic\\_interest](#) property is generally found to be confusing and ill-defined and is a candidate for removal. The goal was to be link a person to some thing that is a topic of their interests (rather than, per [foaf:interest](#) to a page that is about such a topic).

[\[back to top\]](#)

## Property: foaf:weblog

*weblog* - A weblog of some thing (whether person, group, company etc.).

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** [foaf:Document](#)

The [foaf:weblog](#) property relates a [foaf:Agent](#) to a weblog of that agent.

[\[back to top\]](#)

## Property: foaf:workInfoHomepage

*work info homepage* - A work info homepage of some person; a page about their work for some organization.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** [foaf:Document](#)

The [foaf:workInfoHomepage](#) of a [foaf:Person](#) is a [foaf:Document](#) that describes their work. It is generally (but not necessarily) a different document from their [foaf:homepage](#), and from any [foaf:workplaceHomepage](#)(s) they may have.

The purpose of this property is to distinguish those pages you often see, which describe someone's professional role within an organisation or project. These aren't really homepages, although they share some characteristics.

[\[back to top\]](#)

## Property: foaf:workplaceHomepage

*workplace homepage* - A workplace homepage of some person; the homepage of an organization they work for.

**Status:** testing

**Domain:** [foaf:Person](#)

**Range:** [foaf:Document](#)

The [foaf:workplaceHomepage](#) of a [foaf:Person](#) is a [foaf:Document](#) that is the [foaf:homepage](#) of a [foaf:Organization](#) that they work for.

By directly relating people to the homepages of their workplace, we have a simple convention that takes advantage of a set of widely known identifiers, while taking care not to confuse the things those identifiers identify (ie. organizational homepages) with the actual organizations those homepages describe.

For example, Dan Brickley works at W3C. Dan is a [foaf:Person](#) with a [foaf:homepage](#) of <http://rdfweb.org/people/danbri/>; W3C is a [foaf:Organization](#) with a [foaf:homepage](#) of <http://www.w3.org/>. This allows us to say that Dan has a [foaf:workplaceHomepage](#) of <http://www.w3.org/>.

```
<foaf:Person>
  <foaf:name>Dan Brickley</foaf:name>
  <foaf:workplaceHomepage rdf:resource="http://www.w3.org/">
</foaf:Person>
```

Note that several other FOAF properties work this way; [foaf:schoolHomepage](#) is the most similar. In general, FOAF often indirectly identifies things via Web page identifiers where possible, since these identifiers are widely used and known. FOAF does not currently have a term for the name of the relation (eg. "workplace") that holds between a [foaf:Person](#) and an [foaf:Organization](#) that they work for.

[\[back to top\]](#)

## Property: foaf:yahooChatID

*Yahoo chat ID* - A Yahoo chat ID

**Status:** testing

**OWL Type:** An InverseFunctionalProperty (uniquely identifying property)

**Domain:** [foaf:Agent](#)

**Range:** <http://www.w3.org/2000/01/rdf-schema#Literal>

The [foaf:yahooChatID](#) property relates a [foaf:Agent](#) to a textual identifier assigned to them in the Yahoo online Chat system. See Yahoo's the [Yahoo! Chat](#) site for more details of their service. Yahoo chat IDs are also used across several other Yahoo services, including email and [Yahoo! Groups](#).

See [foaf:OnlineChatAccount](#) (and [foaf:OnlineAccount](#)) for a more general (and verbose) mechanism for describing IM and chat accounts.

[\[back to top\]](#)

[foaf-contact@rdfweb.org](mailto:foaf-contact@rdfweb.org)

## Acknowledgments

There are far too many people who have contributed to the FOAF project to name everyone in this early-release of the new improved spec. FOAF wouldn't be such a fun project or be as widely known as it is today without the efforts, enthusiasm and intelligence of the folks who have contributed via the [rdfweb-dev](#) list, [#foaf](#) IRC channel, and [FoafProject](#) wiki site.

That said, a few milestones in FOAF's recent history should be mentioned... We owe particular thanks to Edd Dumbill for his IBM developerWorks articles (which attracted the affections of the Weblogging crowd) and for his Foafbot application whose evolution those articles have tracked. Also Morten Frederiksen's [FoafExplorer](#), Daniel Krech's [Web View](#) aggregator, Jim Ley and Liz Turner's work on [FOAFNaut](#), which alongside FOAFbot, have been instrumental in showing how FOAF data can be collected and used. Meanwhile Leigh Dodd's [foaf-a-matic](#) has been the data creation tool that has been most people's gateway to FOAFdom. FOAF also owes a lot to the folks at [Ecademy](#), [TypePad](#) and elsewhere for showing how end users can share FOAF self-descriptions on the Web without ever seeing a line of XML syntax. Jo Walsh has [enthused many](#) about hooking FOAF up to Geo and mapping data, as has Matt Biddulph by [explaining](#) the workings of his FOAF harvesting and image metadata tools. FOAF has also benefited greatly from documentation contributed in non-English languages, many thanks to all contributors of translations (foaf-a-matic and other docs). FOAF is now arguably better documented [in Japanese](#) and [Spanish](#) than in English, thanks to Masahide Kanzaki and Leandro Mariano López respectively. Thanks also to [Chris Schmidt](#) for fixing up the [spec generation](#) tool (now a Python/Redland script), as well as for contributing numerous [cool hacks](#) to the FOAF community.

This brief survey only scratches the surface of a growing body of work. Sincere thanks to all who have contributed tools, documentation, brain cells and enthusiasm to this project. We should also mention that FOAF would not be possible without the collaborative and opensource efforts of the RDF/XML developer community, both in terms of idea sharing (#rdfg etc) and freely available tools (Jena, Redland, RDFlib, Cwm, Sesame, 3store etc).