

This is a preview of "PAS 212:2016". [Click here to purchase the full version from the ANSI store.](#)

## PAS 212:2016

# Automatic resource discovery for the Internet of Things – Specification



This is a preview of "PAS 212:2016". [Click here to purchase the full version from the ANSI store.](#)

### **Publishing and copyright information**

The BSI copyright notice displayed in this document indicates when the document was last issued.

© The British Standards Institution 2016. Published by BSI Standards Limited 2016.

**ISBN** 978 0 580 91026 5

**ICS** 35.080, 35.240.01

*No copying without BSI permission except as permitted by copyright law.*

### **Publication history**

First published May 2016

This is a preview of "PAS 212:2016". [Click here to purchase the full version from the ANSI store.](#)

# Contents

Foreword .....	iii
Introduction .....	iv
<b>1 Scope</b> .....	<b>1</b>
<b>2 Normative references</b> .....	<b>2</b>
<b>3 Terms, definitions and abbreviations</b> .....	<b>3</b>
<b>4 Catalogue format</b> .....	<b>4</b>
<b>5 Server API</b> .....	<b>7</b>
<b>6 Search extensions</b> .....	<b>9</b>
<b>7 Security extensions</b> .....	<b>13</b>
<b>8 Other extensions</b> .....	<b>15</b>
<b>9 Claims of conformity</b> .....	<b>17</b>
<b>Annexes</b>	
Annex A (informative) Adopting a security-minded approach .....	18
Annex B (informative) Minimum valid catalogue .....	20
Annex C (informative) Search queries and results .....	21
Annex D (informative) Multi-search queries .....	24
Annex E (informative) Signing catalogues .....	25
Annex F (informative) Providing an access hint .....	27
Annex G (informative) Catalogue subscription examples .....	28
Annex H (informative) Mechanisms for catalogue subscription .....	29
Annex I (informative) Declaring a licence .....	31
Bibliography .....	32
<b>List of figures</b>	
Figure 1 – The structure of the catalogue format .....	iv
Figure B.1 – Canonical minimum valid catalogue .....	20
Figure C.1 – Example catalogue used to illustrate simple search queries	21
Figure C.2 – Example catalogue resulting from simple search queries set out above .....	22
Figure C.3 – Example catalogue resulting from above further simple search queries .....	23
Figure D.1 – Example multi-search query containing a single query .....	24

This is a preview of "PAS 212:2016". [Click here to purchase the full version from the ANSI store.](#)

Figure D.2 – Example multi-search query for the intersection of two queries .....	24
Figure D.3 – Example multi-search query for the union of two queries .....	24
Figure D.4 – Example complex multi-search query .....	24
Figure E.1 – Example of signed catalogue .....	25
Figure E.2 – Example of signed item .....	26
Figure F.1 – Example access hint .....	27
Figure G.1 – Subscribing to a stream .....	28
Figure G.2 – Receiving an update .....	28
Figure G.3 – Receiving a delete event .....	28
Figure I.1 – Example of a catalogue declaring a licence .....	31

#### List of tables

Table 1 – Required catalogue properties .....	4
Table 2 – Required item properties .....	5
Table 3 – Required metadata properties .....	5
Table 4 – Required metadata relations .....	5
Table 5 – Required catalogue relations .....	5
Table 6 – Optional catalogue relations .....	6
Table 7 – Optional item relations .....	6
Table 8 – Status codes .....	8
Table 9 – Simple search parameters .....	9
Table 10 – Prefix search .....	9
Table 11 – Prefix search examples .....	10
Table 12 – Examples of item relations to enable lexicographic search ...	10
Table 13 – Lexicographic range search parameters .....	11
Table 14 – Item relations to allow geographic search .....	11
Table 15 – Geographic search query parameters .....	11
Table 16 – Multi-search parameters .....	12
Table 17 – JWS relations .....	14
Table 18 – Security access relation .....	14
Table 19 – Credential acquisition relation .....	14
Table 20 – Event relations .....	15
Table 21 – Event format .....	15
Table 22 – Linked data relations .....	16
Table 23 – Licence relations .....	16

This is a preview of "PAS 212:2016". [Click here to purchase the full version from the ANSI store.](#)

# Foreword

This PAS was sponsored by Flexeye with funding from Innovate UK. Its development was facilitated by BSI Standards Limited and it was published under licence from The British Standards Institution. It came into effect on 31 May 2016.

Acknowledgement is given to Pilgrim Beart of DevicePilot as the technical author of this PAS and to Toby Jaffey and John Davies as authors of the original specification on which this PAS is based. Acknowledgement is also given to the following organizations that were involved in the development of this PAS as members of the steering group:

- British Telecommunications plc
- BSI Consumer & Public Interest Network
- Centre for the Protection of National Infrastructure (CPNI)
- DevicePilot
- Flexeye
- Greater London Authority
- Hypercat
- IBM UK Ltd
- In Touch Ltd
- King's College London
- Knowledge Transfer Network
- RedBite Solutions Ltd
- Thingful
- Co-opted member

Acknowledgement is also given to the members of a wider review panel who were consulted in the development of this PAS.

The British Standards Institution retains ownership and copyright of this PAS. BSI Standards Limited as the publisher of this PAS reserves the right to withdraw or amend this PAS on receipt of authoritative advice that it is appropriate to do so. This PAS will be reviewed at intervals not exceeding two years, and any amendments arising from the review will be published as an amended PAS and publicized in *Update Standards*.

This PAS is not to be regarded as a British Standard. It will be withdrawn upon publication of its content in, or as, a British Standard.

The PAS process enables a specification to be rapidly developed in order to fulfil an immediate need in industry. A PAS can be considered for further development as a British Standard, or constitute part of the UK input into the development of a European or International Standard.

## Relationship with other publications

This PAS is based on a specification, known as Hypercat, which was originally developed by the Hypercat working group.

## Use of this document

It has been assumed in the preparation of this PAS that the execution of its provisions will be entrusted to appropriately qualified and experienced people, for whose use it has been produced.

## Presentational conventions

The provisions of this PAS are presented in roman (i.e. upright) type. Its requirements are expressed in sentences in which the principal auxiliary verb is "shall". The word "should" is used to express recommendations, the word "may" is used to express permissibility and the word "can" is used to express possibility, e.g. the consequence of an action of an event.

*Commentary, explanation and general informative material is presented in italic type, and does not constitute a normative element.*

Spelling conforms to The Shorter Oxford English Dictionary. If a word has more than one spelling, the first spelling in the dictionary is used.

## Contractual and legal considerations

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

**Compliance with a PAS cannot confer immunity from legal obligations.**

This is a preview of "PAS 212:2016". [Click here to purchase the full version from the ANSI store.](#)

# 0 Introduction

## 0.1 Motivation

There is an increasing need to share data easily, driven by innovations such as the Internet of Things (IoT), Smart Cities and big data, within specific sectors of the economy such as healthcare, automotive and energy, and across the World Wide Web in general. Some of this data can be made fully public, whilst access to other data needs to be controlled.

If there is a need for data-consuming clients to be hand-coded to access each specific data server, then software engineering becomes a resource constraint which inhibits exponential growth in the number and combination of such clients and servers. PAS 212 aims to address this issue by specifying a means to automate the discovery of such data resources, without either the client or server having to be written to be explicitly compatible with each other.

PAS 212 specifies a common catalogue format that clients can use to discover data in servers that they can use. It describes an open, lightweight JSON-based hypermedia catalogue format for exposing collections of uniform resource identifiers (URIs). Each catalogue may expose any number of URIs, each with any number of resource description framework (RDF)-like triple statements about it. This allows developers to publish linked-data descriptions of resources.

This PAS therefore allows a server to provide a set of resources to a client, each with a set of semantic annotations (metadata). Implementers are free to choose or invent any set of annotations to suit their needs. Where implementers choose similar or overlapping semantics, the possibilities for interoperability are increased.

This PAS intentionally does not set out to solve all the challenges of data interoperability, but only to address the problem of resource discovery and provide a framework in which other interoperability challenges, such as ontologies, monetization and privacy, may be worked out.

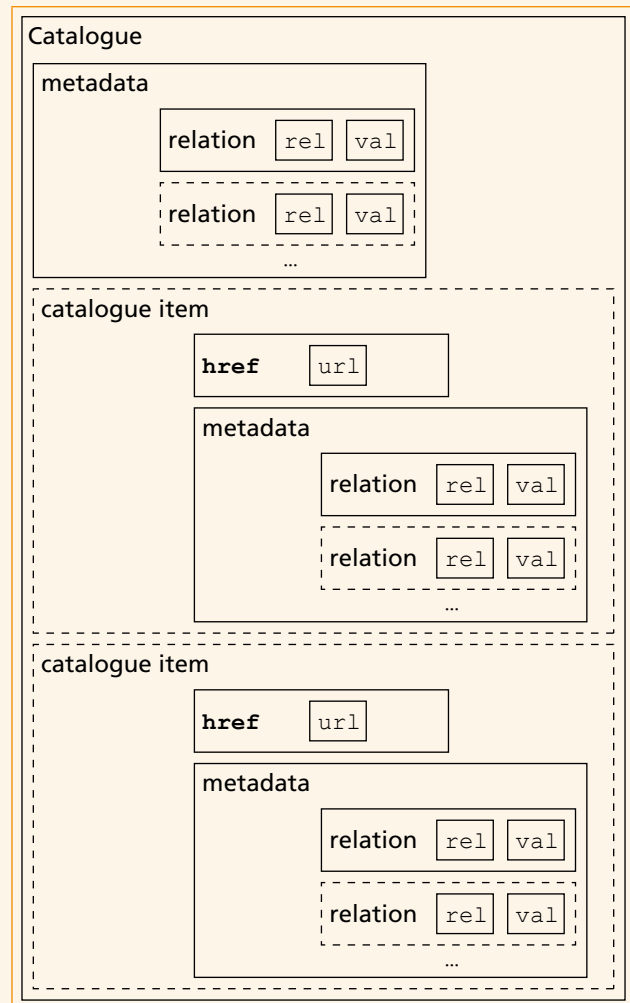
## 0.2 Structure of a catalogue

Figure 1 shows the structure of the catalogue format described in this PAS. Dotted lines indicate zero or more optional items. Ellipses (...) indicate optional repetition.

A catalogue contains its metadata, and also zero or more catalogue items. Each catalogue item consists of an href pointer to the item and metadata about the item.

Metadata consists of a list of relations. Each relation is a rel-val pair.

**Figure 1** – The structure of the catalogue format



This is a preview of "PAS 212:2016". [Click here to purchase the full version from the ANSI store.](#)

## 1 Scope

This PAS specifies a protocol whereby any compliant software client can automatically discover data that is stored within any compliant software server, without either the client or server having to be written to have been explicitly compatible with each other.

It applies to the design of services for IoT and the World Wide Web in general, and in particular to the design of applications intended to operate within broad ecosystems such as smart cities, as well as specific industry sectors. It aims to break down the vertically-integrated software silos that have previously existed within the IoT industry.

More specifically it covers the format for representing a catalogue of linked-data resources, annotated with metadata.

It also provides conditional requirements for catalogue access in the following areas:

- catalogue transport;
- security mechanisms to protect access and to prove provenance;
- search functions;
- subscription mechanisms;
- well-known entry-points;
- machine-readable hints to ease usability.

**NOTE 1** *Clauses 5 to 8 provide conditional requirements, which means that they might not be relevant, but if they are, then they need to be implemented as specified in these clauses.*

It does not cover implementation of the linked-data resource services themselves.

This PAS is for use by software engineers for IoT (or web services more generally), who are seeking to:

- write a software interface for a client that does not need to be re-written every time it is used with a new server; and
- write a software interface for a server that does not need to be re-written every time it is used with a new client.

Use of this PAS therefore seeks to solve the current problem that lack of interoperability is preventing the exponential growth in the number and combination of such clients and servers.

This PAS is also for use by commissioners of software projects, who, by recommending compliance to this specification, can promote open interoperability between the project parts, and thus avoid vendor lock-in.

This PAS allows and encourages the use of existing ontologies, schemas, etc. by reference.

**NOTE 2** *Attention is drawn to the importance of adopting a security-minded approach, further details of which can be found in Annex A.*