

# FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

## FIPA JXTA Discovery Middleware Specification

<b>Document title</b>	FIPA JXTA Discovery Middleware Specification		
<b>Document number</b>	Version 1.2	<b>Document source</b>	FIPA TC Ad Hoc
<b>Document status</b>	Preliminary	<b>Date of this status</b>	2003/10/20
<b>Supersedes</b>	None		
<b>Contact</b>	adhoc@fipa.org		

© 1996-2003 Foundation for Intelligent Physical Agents

<http://www.fipa.org/>

Geneva, Switzerland

### Notice

Use of the technologies described in this specification may infringe patents, copyrights or other intellectual property rights of FIPA Members and non-members. Nothing in this specification should be construed as granting permission to use any of the technologies described. Anyone planning to make use of technology covered by the intellectual property rights of others should first obtain permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of this specification to determine first whether part(s) sought to be implemented are covered by the intellectual property of others, and, if so, to obtain appropriate licenses or other permission from the holder(s) of such intellectual property prior to implementation. This specification is subject to change without notice. Neither FIPA nor any of its Members accept any responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this specification.

## 21 **Foreword**

22 The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to  
23 promoting the industry of intelligent agents by openly developing specifications supporting interoperability among  
24 agents and agent-based applications. This occurs through open collaboration among its member organizations,  
25 which are companies and universities that are active in the field of agents. FIPA makes the results of its activities  
26 available to all interested parties and intends to contribute its results to the appropriate formal standards bodies  
27 where appropriate.

28 The members of FIPA are individually and collectively committed to open competition in the development of  
29 agent-based applications, services and equipment. Membership in FIPA is open to any corporation and individual  
30 firm, partnership, governmental body or international organization without restriction. In particular, members are  
31 not bound to implement or use specific agent-based standards, recommendations and FIPA specifications by  
32 virtue of their participation in FIPA.

33 The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a  
34 specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the  
35 process of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications  
36 Policy [f-out-00003]. A complete overview of the FIPA specifications and their current status may be found on the  
37 FIPA Web site.

38 FIPA is a non-profit association registered in Geneva, Switzerland. As of June 2003, the 50 members of FIPA  
39 represented many countries worldwide. Further information about FIPA as an organization, membership  
40 information, FIPA specifications and upcoming meetings may be found on the FIPA Web site at  
41 <http://www.fipa.org/>.

## 42 Contents

43	1	Scope .....	1
44	2	JXTA Discovery Middleware .....	2
45	2.1	Discovery Middleware Component Name .....	2
46	2.2	New JXTA Components .....	2
47	2.2.1	Generic Discovery Service .....	2
48	2.2.2	Agent Peer Group .....	2
49	2.2.3	Generic Discovery Advertisements .....	3
50	2.3	Interface Definition .....	4
51	2.3.1	Generic Discovery Protocol .....	4
52	3	References .....	6
53	4	Informative Annex A – Generic Discovery Advertisement Example .....	7
54	5	Informative Annex B – Generic Discovery Protocol Example .....	8
55	6	Informative Annex C – Notes for Developers .....	10

## 1 Scope

This document deals with the application of JXTA [JXTA] as discovery middleware (DM) for the Agent Discovery Service (ADS). This specification forms part of the FIPA Agent Discovery Service Specification [<<Agent Discovery Service Specification>>] and contains specifications for:

- The necessary extensions to JXTA to be usable as DM within the ADS.
- The interface for the interaction of JXTA DMs.

## 2 JXTA Discovery Middleware

JXTA technology is a set of open protocols that allow any connected device on the network to communicate in a peer-to-peer manner. For details see [JXTA].

### 2.1 Discovery Middleware Component Name

The name assigned to this discovery middleware component is:

`fipa.ads.dm.jxta.std`

### 2.2 New JXTA Components

JXTA advertisements are used to describe any kind of resource within the network. A DF-like, template based discovery realized with the means of JXTA needs a mechanism to find arbitrary attribute/value pairs within JXTA advertisements. The new components, which are necessary to realize this, are a new:

- JXTA service (see section 2.2.1)
- JXTA peer group (see section 2.2.2)
- JXTA advertisement type (see section 2.2.3)

#### 2.2.1 Generic Discovery Service

This specification does not specify how the Generic Discovery Service (GDS) must look like, but the GDS must provide the DM functionality specified in [<<Agent Discovery Service Specification>>].

A JXTA Module Class ID uniquely identifies a class of JXTA modules, for instance JXTA services, that provide a certain functionality. The JXTA Module Class ID for GDS services offering the DM functionality specified in [<<Agent Discovery Service Specification>>] is:

`urn:jxta:uuid-A4182DD0DC504CD48CC4A441E56627B405`

A JXTA Module Specification ID uniquely identifies a set of protocol compatible modules. The JXTA Module Specification ID for GDS services offering the DM functionality specified in [<<Agent Discovery Service Specification>>] and implementing the Generic Discovery Protocol as specified in 2.3.1 is:

`urn:jxta:uuid-A4182DD0DC504CD48CC4A441E56627B4F15B15DB91E74BC28945908432FC790406`

The version of the JXTA Module Specification belonging to the GDS services offering the DM functionality specified in [<<Agent Discovery Service Specification>>] and implementing the Generic Discovery Protocol is: 1.0

#### 2.2.2 Agent Peer Group

The Agent Peer Group (APG) is a child of the JXTA Net Peer Group. The APG must be joined during the startup of the DM, because the APG hosts the GDS and is the scope of the GDS. Other services that may also be hosted by the APG are outside the scope of this specification.

A JXTA Peer Group ID uniquely identifies a JXTA peer group. The JXTA Peer Group ID assigned to the APG is:

`urn:jxta:uuid-A19999539B18489C8E47860E6C89549E02`

The JXTA Module Specification ID for the APG uniquely identifies the module providing the implementation of the peer group module on the JXTA peer. The JXTA Module Specification ID for the APG is:

`urn:jxta:uuid-DEADBEEFDEAFBABAEEEDBABE00000001994191D8846242F8AF4FBB14AF2CDFAF06`

## 2.2.3 Generic Discovery Advertisements

A Generic Discovery Advertisement (GDA) is used to handle agent or service descriptions, for example FIPA `df-agent-descriptions` [FIPA00023], at the level of JXTA. A GDA consists of meta information elements and of elements describing agents or services

The `PublishingTime` meta information element contains the time in milliseconds since 1970, when the GDA was published, i.e. made available publicly in the network.

The `Id` meta information element is used to specify the content type of the GDA, i.e. it defines how the elements of the GDA, except the meta information elements, must be interpreted. Table 1 summarizes the currently available GDA content types.

GDA Content Type	Interpret Content as	Note
Df-agent-description	FIPA <code>df-agent-description</code>	see [FIPA00023] and 2.2.3.1

Table 1: Possible Content Types of a GDA

### 2.2.3.1 Generic Discovery Advertisements with `df-agent-description` Content

In a GDA with `df-agent-description` content, value sets and value sequences of the FIPA `df-agent-description` parameters are broken up to single advertisement elements. For value sequences the order of the single elements in the advertisement is according to the order of values within a sequence of a FIPA `df-agent-description`.

The XML Schema<sup>1</sup> [W3C] of a GDA with `DFAD` content is as follows:

```
<xs:element name="GenericDiscoveryAdv" type="fipa:GenericDiscoveryAdv"/>
<xs:complexType name="GenericDiscoveryAdv">
  <xs:sequence>
    <xs:element name="Id" type="DFAD"/>
    <xs:element name="PublishingTime" type="xs:unsignedInt"/>

    <xs:complexType name="agentID" minOccurs="0"2>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="address" type="xs:string"
          maxOccurs="unbounded" minOccurs="0"/>
        <xs:element name="resolver" type="agentID"
          maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="service" maxOccurs="unbounded" minOccurs="0">
      <xs:sequence>
        <xs:element name="name" type="xs:string" minOccurs="0"/>
        <xs:element name="type" type="xs:string" minOccurs="0"/>
        <xs:element name="protocol" type="xs:string"
          maxOccurs="unbounded" minOccurs="0"/>
        <xs:element name="ontology" type="xs:string"
          maxOccurs="unbounded" minOccurs="0"/>
        <xs:element name="language" type="xs:string"
          maxOccurs="unbounded" minOccurs="0"/>
        <xs:element name="ownership" type="xs:string" minOccurs="0"/>
        <xs:complexType name="property" maxOccurs="unbounded" minOccurs="0">
          <xs:sequence>
```

<sup>1</sup> As stated in [Sun], service and protocol authors are recommended to specify advertisements and messages by using the XML Schema language.

<sup>2</sup> The default value for the `minOccurs` and `maxOccurs` attributes is 1.

```

161         <xs:element name="name" type="xs:string"/>
162         <xs:element name="type" type="xs:string"/>
163     </xs:sequence>
164 </xs:complexType>
165 </xs:sequence>
166 </xs:complexType>
167
168 <xs:element name="protocol" type="xs:string"
169     maxOccurs="unbounded" minOccurs="0"/>
170 <xs:element name="ontology" type="xs:string"
171     maxOccurs="unbounded" minOccurs="0"/>
172 <xs:element name="language" type="xs:string"
173     maxOccurs="unbounded" minOccurs="0"/>
174 <xs:element name="lease-time" type="xs:unsignedInt" minOccurs="0"/>
175 </xs:sequence>
176 </xs:complexType>
177
178 <xs:simpleType name="DFAD">
179     <xs:restriction base="xs:string">
180         <xs:pattern value="df\ -agent\ -description"/>
181     </xs:restriction>
182 </xs:simpleType>
183

```

## 2.3 Interface Definition

To enable the interaction of JXTA DMs on different agent platforms (AP), the Generic Discovery Protocol (GDP) is used to exchange GDAs. The GDP is implemented by the GDS, because it is a part of the GDS. All GDS implementations, which use the JXTA Module Specification ID defined in 2.2.1, must implement the GDP as specified in 2.3.1.

### 2.3.1 Generic Discovery Protocol

The GDP is a request/response protocol to discover GDAs. The GDP comprises two messages, the `GenericDiscoveryQuery` message and the `GenericDiscoveryResponse` message.

The `GenericDiscoveryQuery` is used for emitting discovery queries. The `Pattern` attribute of the `GenericDiscoveryQuery` is a required element and must contain a GDA that acts as the search pattern. The `Threshold` attribute is an optional element defining the maximum number of advertisements that should be sent by a peer responding to this query.

The XML Schema of a `GenericDiscoveryQuery` is as follows:

```

200
201 <xs:element name="GenericDiscoveryQuery" type="fipa:GenericDiscoveryQuery"/>
202 <xs:complexType name="GenericDiscoveryQuery">
203     <xs:sequence>
204         <xs:element name="Pattern" type="fipa:GenericDiscoveryAdv"/>
205         <xs:element name="Threshold" type="xs:unsignedInt" minOccurs="0"/>
206     </xs:sequence>
207 </xs:complexType>
208

```

The `GenericDiscoveryResponse` message is used for returning matching GDAs in response to a `GenericDiscoveryQuery`. The `Count` attribute of the `GenericDiscoveryResponse` message is a required element and contains the number of returned results, i.e. the number of `Response` elements. A `GenericDiscoveryResponse` message contains one or more `Response` elements, each one wrapping a GDA that matches the previously received GDA search pattern.

The XML Schema of a `GenericDiscoveryResponse` is as follows:

```

215 <xs:element name="GenericDiscoveryResponse" type="fipa:GenericDiscoveryResponse"/>
216
217

```

```
218 <xs:complexType name="GenericDiscoveryResponse">
219   <xs:sequence>
220     <xs:element name="Count" type="xs:unsignedInt"/>
221     <xs:element name="Response" type="fipa:GenericDiscoveryAdv"
222               maxOccurs="unbounded"/>
223   </xs:sequence>
224 </xs:complexType>
225
```



### 3 References

- [FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2002.  
<http://www.fipa.org/specs/fipa00023/>
- [JXTA] Project JXTA.  
<http://www.jxta.org/>
- [W3C] XML Schema Part 1: Structures. World Wide Web Consortium (W3C), May 2001.  
<http://www.w3.org/TR/xmlschema-1/>
- [Sun] JXTA v2.0 Protocols Specification. Sun Microsystems, March 2003.  
<http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.html>

## 4 Informative Annex A – Generic Discovery Advertisement Example

As an example the GDA of the job-agent is shown below. The GDA is of content type df-agent-description and describes the agent itself, two of its supported services, one supported ontology and the duration how long the df-agent-description is valid.

```
<?xml version="1.0"?>
<!DOCTYPE fipa:GenericDiscoveryAdv>

<fipa:GenericDiscoveryAdv xmlns:fipa="http://www.fipa.org">
  <Id>df-agent-description</Id>
  <PublishingTime>2753606022</PublishingTime>

  <agentID>
    <name>job-agent@foo.com </name>
    <address>iiop://foo.com/acc</address>
  </agentID>

  <service>
    <name>job-application</name>
    <protocol>application-proto</protocol>
  </service>

  <service>
    <name>job-offer</name>
    <protocol>offer-proto</protocol>
  </service>

  <ontology>job-onto</ontology>
  <lease-time>3600000</lease-time>
</fipa:GenericDiscoveryAdv>
```

## 5 Informative Annex B – Generic Discovery Protocol Example

A `GenericDiscoveryQuery`, which can be emitted to discover the job agent, is shown below. This kind of message is used as part of the GDP in the discovery process of the GDS. The `GenericDiscoveryQuery` wraps within the `Pattern` element a GDA with content type `df-agent-description`. The GDA is used as `df-agent-description` search template. The `PublishingTime` element of the `df-agent-description` advertisement is zero because a search template is never published.

The `<` characters of the wrapped GDA are encoded by the escape sequence `&lt;`. This is a means of JXTA to guarantee the appropriate processing of nested advertisements.

```
<?xml version="1.0"?>
<!DOCTYPE fipa:GenericDisocveryQuery>

<fipa:GenericDiscoveryQuery xmlns:fipa="http://www.fipa.org">
  <Pattern>
    &lt;?xml version="1.0"?>
    &lt;!DOCTYPE fipa:GenericDiscoveryAdv>

    &lt;fipa:GenericDiscoveryAdv xmlns:fipa="http://www.fipa.org">
      &lt;Id>df-agent-description&lt;/Id>
      &lt;PublishingTime>0&lt;/PublishingTime>

      &lt;service>
        &lt;name>job-application&lt;/name>
        &lt;protocol>application-proto&lt;/protocol>
      &lt;/service>

      &lt;ontology>job-onto&lt;/ontology>
    &lt;/fipa:GenericDiscoveryAdv>
  </Pattern>

  <Threshold>5</Threshold>
</fipa:GenericDiscoveryQuery>
```

A `GenericDiscoveryResponse` belonging to the previous query is shown below. One matching result is returned within the `Response` element, namely the job-agent.

```
<?xml version="1.0"?>
<!DOCTYPE fipa:GenericDisocveryResponse>

<fipa:GenericDiscoveryResponse xmlns:fipa="http://www.fipa.org">
  <Count>1</Count>

  <Response>
    &lt;?xml version="1.0"?>
    &lt;!DOCTYPE fipa:GenericDiscoveryAdv>

    &lt;fipa:GenericDiscoveryAdv xmlns:fipa="http://www.fipa.org">
      &lt;Id>df-agent-description&lt;/Id>
      &lt;PublishingTime>2753606022&lt;/PublishingTime>

      &lt;agentID>
        &lt;name>job-agent@foo.com&lt;/name>
        &lt;address>iiop://foo.com/acc&lt;/address>
```

```
326      </agentID>
327
328      <service>
329        <name>job-application</name>
330        <protocol>application-proto</protocol>
331      </service>
332
333      <service>
334        <name>job-offer</name>
335        <protocol>offer-proto</protocol>
336      </service>
337
338      <ontology>job-onto</ontology>
339      <lease-time>3600000</lease-time>
340
341    </fipa:GenericDiscoveryAdv>
342  </Response>
343
344 </fipa:GenericDiscoveryResponse>
```

## 6 Informative Annex C – Notes for Developers

1. A straight forward way to implement the GDS is to use JXTA's Resolver Service with its Peer Resolver Protocol and to design the GDS similar as JXTA's Discovery Service with its Peer Discovery Protocol. The Peer Resolver Protocol could be used to transport GDP messages. The functionality of JXTA's Discovery Service could be extended to find arbitrary attribute/value pairs within GDAs.
2. The `Id` meta information element in the GDA can be used efficiently to access a certain type of advertisement within JXTA's local cache. This cache is accessed via the JXTA Discovery Service. Therefore for example all GDAs, which contain `df-agent-description` information and have an `Id` meta information element stating to be a `df-agent-description`, can be retrieved from the JXTA cache via the JXTA Discovery Service.
3. The GDA's XML Schema definition uses the compositor `sequence`, which is a property of the model group schema component described in [W3C]. Note that despite its use, the order of the elements contained within these compositors is assumed to be arbitrary.  
This modulus operandi complies to the one applied in JXTA source code and results from the fact that the current version of the XML Schema language offers no better suited compositor. The `all` compositor would allow for an arbitrary order of contained elements, but limits the occurrence of a certain element to zero or one. This would be an even bigger limitation in the context of a GDA that should contain for example several service descriptions. For details see [W3C] section 3.8.