

Chinook User Documentation (CUD)

Chinook Advertisements

Stephen Montgomery Oct. 27th, 2003

Chinook servers publish advertisements using the JXTA protocol. The client peer intercepts these advertisements and displays services (for which jobs can subsequently be run). This document describes how Chinook advertisements are made and how you can edit them for your services.

Advertisements in JXTA

An advertisement is an XML document that describes a particular JXTA message, whether that is a peer, peer group or service. These messages are discovered then cached locally. (To see your cache go to your own .jxta/ directory - created when you first run Chinook). As a service provider, you are interested in only two types of messages, the `ModuleSpecAdvertisement` (MSA) and the `ModuleImplAdvertisement` (MIA). These are located in your `advertisements/` directory. (Chinook handles peer group and peer advertising by itself)

The Chinook ModuleSpecAdvertisement

Chinook has two `ModuleSpecAdvertisement`'s in its `advertisements/` folder. One for the RMI protocol and another for the web services protocol. Depending on how you want to run your server, you will need to modify one of these advertisements. For the purpose of this example, we will look at the RMI advertisement.

chinookSpecAdvRMI.xml

```
<?xml version="1.0"?>
<!DOCTYPE jxta:MSA>
<jxta:MSA xmlns:jxta="http://jxta.org">
  <MSID>
    urn:jxta:uuid-
72CE4F415C994ADBB5BCB897E6BBB3D0EB39B9952C0D4D79BAD5BDE678877F4D06
  </MSID>
  <Name>
    JXTASPEC:Chinook-RMI
  </Name>
  <Crtr>
    smontgom@bcgsc.bc.ca
  </Crtr>
  <SURI>
    http://smweb.bcgsc.bc.ca
  </SURI>
  <Vers>
    1.0
  </Vers>
  <Desc>
    A Chinook DBAS RMI Server
  </Desc>
</jxta:MSA>
```

The `ModuleSpecAdvertisement` is simple in nature. The `<MSID>` tag holds a unique id that identifies this service, for your purposes it can be any valid JXTA id (see <http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.html> JXTA Protocol Specification). The `<Name>` tag shouldn't be changed; Chinook searches for Spec advertisements based on this. If the `<Name>` tag is changed, your advertisement won't be discovered. The `<Crtr>` tag specifies who the publisher of this service is (I always use my own name, this will show up for the Chinook client and they will be able to contact me based on this info). The `<SURI>` tag points to the documentation for Chinook; but this can be changed to be any service providers relevant service documents. The `<Vers>` tag specifies what version of the Chinook is being used. If there is ever a release that changes so much in functionality, this may be upgraded. Finally, the `<Desc>` tag describes the service.

The Chinook ModuleImplAdvertisement

Chinook has two `ModuleImplAdvertisement`'s in its `advertisements/` folder. These correspond to specific implementations of the `ModuleSpecAdvertisement`'s. Any service provider MUST change these advertisements to reflect the appropriate server information (the URI). We will look at the RMI example of the `ModuleImplAdvertisement`.

chinookImplAdvRMI.xml

```
<?xml version="1.0"?>
<!DOCTYPE jxta:MIA>
<jxta:MIA xmlns:jxta="http://jxta.org">
  <MSID>
    urn:jxta:uuid-
72CE4F415C994ADBB5BCB897E6BBB3D0EB39B9952C0D4D79BAD5BDE678877F4D06
  </MSID>
  <Comp>
    <Efmt> JDK1.4 </Efmt> <ChinookImpl> 1.0 </ChinookImpl>
  </Comp>
  <Code>
    //localhost:1099/ApplicationServerImpl
  </Code>
  <PURI>
    Not yet available
  </PURI>
  <Prov>
    smontgom@bcgsc.bc.ca
  </Prov>
  <Desc>
    RMI Chinook Implementation, Parm is generated with service names
  </Desc>
</jxta:MIA>
```

The `ModuleImplAdvertisement` here has only a few things that must be noted. The `<MSID>` tag must match that of the corresponding `ModuleSpecAdvertisement`. The `<Comp>` tag specifies compatibility information. Here it states that users must have at least JDK1.4 and the 1.0 implementation of the Chinook interface (See CUD “The Chinook Remote Interface”). The `<Code>` tag points to the

relevant URI to run the Chinook service. The <PURI> tag specifies a location to download the appropriate classes; this is not yet implemented. The <Prov> tag specifies who is providing this implementation (here the provider is myself; you will want to change this to yourself). Finally, the <Desc> tag describes the implementation.

Why are discovered advertisements different from these?

A discovered Chinook ModuleImplAdvertisement will have automatically filled in <service_type> and <service_name> elements as part of the <Param> tag. The <Param> tag specifies additional Chinook metadata. Chinook uses this to ontologically classify services. See “CUD- Adding New Services”. This allows versions of Chinook to discover advertisements of only the type they are interested in.

For questions or updates to this document, e-mail chinook@bcgsc.bc.ca