

# Cougaar

Cougaar Agent Architecture Open-Source site

## Templates Support

The Cougaar **template** property controls the standard set of infrastructure components that are loaded into the nodes and agents.

For example, setting the template to **embedded** tells the infrastructure to load a local-only (trivial) message transport and naming service. If the template is set to **lan** then the distributed message transport and naming service are loaded.

The System Property is:

```
-Dorg.cougaar.society.xml.param.template=$value
```

The supported values are:

<code>*embedded*</code>	= loopback mts/wp, no servlets, designed for applets and other embedded environments.
<code>*single_node*</code>	= loopback mts/wp, no metrics, adds standard plugins and servlets (e.g. "/tasks" servlet)
<code>*single_debug*</code>	= full mts, loopback wp, full metrics and aspects
<code>*legacy*</code>	= <u>default</u> backwards-compatible with prior Cougaar releases. distributed mts/wp, adds planning and communities.
<code>*lan*</code>	= distributed mts/wp, added metrics aspects, adds standard plugins and servlets (e.g. "/wp" servlet) adds servlet tunneling through mts (but will favor http-redirec
<code>*wan*</code>	= currently same as lan. eventually we'll add mts sequence checking here, etc.

It is defined in:

```
configs/common/SimpleAgent.xml
```

This parameter simply enables/disables other parameters, and can be overwritten in the society XML file. For example, an ‘embedded’ configuration can add servlets by setting:

```
-Dorg.cougaar.society.xml.param.servlets=true
```

### Matrix of enabled/disabled features

PARAMETER	TEMPLATE VALUE					
	EMBEDDED	SINGLE_NODE	SINGLE_DEBUG	LEGACY	LAN	WAN
<b>threadService</b>	trivial	full	full	full	full	full
<b>pluginThreadPool</b> <i>note</i>	N/A	30	30	30	30	30
<b>mts</b>	singlenode	singlenode	full	full	full	full
<b>wpserver</b>	singlenode	singlenode	singlenode	true <i>note</i>	full	full
<b>socketFactory</b> required by full wp and mts	false	false	true	true	true	true
<b>metrics</b>	trivial	trivial	full	full	full	full
<b>standard_aspects</b>	false	false	true	false	true	true
<b>sensors</b> e.g. agent load	false	false	true	false	true	true
<b>mobility</b>	false	false	false	false	true	true
<b>servlets</b> engine	false	true	true	true	true	true
<b>standard_node_servlets</b> e.g. “/wp”	false	true	true	false	true	true
<b>standard_agent_servlets</b> e.g. “/tasks”	false	true	true	false	true	true
<b>servlet_engine.tomcat</b>	false	true	true	true	true	true
<b>servlet_engine.micro</b>	false	false	false	false	false	false
<b>servlet_engine.mts</b>	false	false	false	false	true	true
<b>servlet_redirector.http_redirect</b>	false	true	true	true	true	true
<b>servlet_redirector.http_tunnel</b>	false	false	false	false	false	false
<b>servlet_redirector.mts_tunnel</b>	false	false	false	false	true	true
<b>communities</b>	false	false	false	legacy <i>note</i>	false	true
assume <b>planning</b> domain should be	false	false	false	true <i>note</i>	false	false

loaded

read <b>domain_ini</b> file	false	false	false	true <i>note</i>	false	false
<b>link_protocol.loopback</b>	default	default	true <i>note</i>	default	default	default
<b>link_protocol.rmi</b>	default	default	default	default	default	default
<b>link_protocol.jms</b>	default	default	default	default	default	default

## Notes

The full **threadService** supports a plugin thread limit. The trivial thread service implementation has no limit.

The **wpserver** supports a default “true” value, which tells the node to load the WPServer component into every node. This is for backwards compatibility.

The **communities** option supports a default “false” value, a “legacy” value for the old lib/community.jar implementation, and a “true” value for the new core implementation in `org/cougaar/core/agent/service/community`.

For backwards compatibility, the legacy configuration loads the **planning** domain. The preferred approach is to explicitly list domains in each agent, just like plugins.

Similar to the above planning issue, the legacy configuration reads a “LDMDomains.ini” domain configuration file.

By default, the RMI and Loopback link protocols are loaded. In the ‘single\_debug’ configuration we only want loopback.

---