

You are in: [Distributed State Machine](#) > [Distributed State Machines](#) > [Dev Guide](#) > Configuration

Configuration

☐ Like | Updated September 21, 2011 by [David A. Wood](#) | Tags: *None*

DSM uses properties to configure its operation. The properties can have defaults defined in dsm.properties and wpml.properties files. They can be overridden with Java system properties on the command line as in ... *-Dsome.property=some.value...*

dsm.properties

Logging

Logging is done using java.util.logging and mechanisms similar to that used in WPML.

Property Name	Description	Default (defined in code)	Examples
logging.enabled	Turns logging on or off.	true	logging.enabled=true
com.ibm.watson.dsm.level	Sets the level of logging to one of SEVERE, WARNING, INFO, FINE, FINER, FINEST.	INFO	com.ibm.watson.dsm.level=FINE

Rule Engine

These configure the engine, which supports SolidDB in-memory DB or an arbitrary JDBC server, including Derby in-memory. The JDBC properties have fixed values, which must be set as defined below.

Property Name	Description	Default (defined in code)	Examples
engine.storage	Determines what type of JDBC server to use. One of 'jdbc' or 'soliddb'. For 'jdbc' generally any jdbc server can be used, although we have only tested with Derby. For 'soliddb', SolidDB in-memory database will be used. Defaults to 'jdbc'	jdbc	engine.storage=jdbc engine.storage=soliddb
engine.jdbc.drivers	Set the JDBC driver to be used by the engine. The value must be set to solid.jdbc.SolidDriver if engine storage is set ot 'memory'. For jdbc, the URL must contain a %app% field that will name the database with the name of the application.	undefined	engine.jdbc.drivers=org.apache.derby.jdbc.ClientDriver engine.jdbc.drivers=solid.jdbc.SolidDriver
engine.jdbc.url	Set the JDBC database URL. The value must be set to	undefined	engine.jdbc.url=jdbc:derby://localhost:1527/%app%; create=true

	solid.jdbc.SolidDriver if engine storage is set ot 'memory'. For jdbc, the URL must contain a %app% field that will name the database with the name of the application.		engine.jdbc.url=jdbc:derby:memory:%app%; create=true engine.jdbc.url=jdbc:solid://localhost:0
engine.jdbc.user	Must be set to 'root'	undefined	engine.jdbc.user=root
engine.jdbc.password	Must be set to 'dsm'	undefined	engine.jdbc.password=dsm
storage.memory.soliddb.port	Port used by soliddb. Must be set if running multiple engines on the same machine.	1934	storage.memory.soliddb.port=1935

Application Registry and Discovery

Application discovery can be done in one of two ways: database or Apache JCS. To configure which one use the following in dsm.properties:

Property Name	Description	Default (defined in code)	Examples
application.registry	Defines the type of registry implementation. One of <i>jcs</i> , <i>internal</i> or <i>wpml</i> . <i>jcs</i> indicates the use of Apache's distributed hash map. <i>wpml</i> indicates the use of one of WPML's repository implementations. Internal is a registry that does not work across JVMs.	jcs	application.registry=wpml

Apache JCS registry

When using JCS, you may use the following specified on the command line (as in java -Djcs....=...).

Property Name	Description	Default (defined in code)	Examples
jcs.auxiliary.LTCP.attributes.TcpListenerPort	Tells JCS which port to listen on for data exchange. This should be set when running more than one application on the same host that uses the JCS registry.	1110	jcs.auxiliary.LTCP.attributes.TcpListenerPort=1111

WPML-base registry

When using the WPMLTupleAppRegistry you may use the following in wpml.properties

Property Name	Description	Default (defined in	Examples
---------------	-------------	------------------------	----------

		code)	
singleton.pdp.repository	Defines the type of repository implementation (jdbc, file, grid, etc) to be used.	undefined	singleton.pdp.repository=jdbc
jdbc.drivers		undefined	
jdbc.url		undefined	
jdbc.username		undefined	root
jdbc.password		undefined	wpml

Application Discovery and Tuple Exchange

If an application is using a NetworkTupleReceiver to receive tuples over a network connection, then the propertie(s) below are relevant.

Property Name	Description	Default (defined in code)	Examples
receiver.network.port	Port on which the P2PTupleReceiver will listen for tuples from other local or remote applications.	3022	receiver.network.port=3022
receiver.network.address	This is only needed in environments where InetAddress.getLocalHost() fails. This includes the Core network emulation environment and perhaps environments that have more than 1 network interface.	null	receiver.network.address=9.2.234.23

Comments

There are no comments.