

# Java SCA Roadmap

This page contains a list of features that our community thinks is important for Tuscany Java SCA. The page is split into two parts. 1.x and 2.x. The 1.x code stream is our SCA 1.0 OSOA spec based code stream. 2.x is our SCA 1.1 OASIS spec based code stream (which is just starting up as of November 2008). Hence roadmap items may differ between the two.

You may be wondering where these roadmap items come from. Usually they are the things that individuals think need to be added or be fixed in the code base. If you are looking for something to work on and no roadmap items take your fancy the other useful places to look for potential new roadmap items are;

[Tuscany Survey Responses](#)  
[Outstanding Java SCA JIRA](#)

The [old roadmap page](#) (from when we were just doing SCA 1.0 development) is also a useful reference. But this is a clean page to allow us to separate 1.x and 2.x activities. I've put the items in I know about and which I think are imminent. I've resisted the temptation to put on those items that we never seem to get round to doing. I've included a 2 release horizon to try and excluded all of the fluff we have collected on roadmap pages before. Items can be linked to a more detailed roadmap page as they are being worked on.

**Please help to make this a live document. Thanks**

**Interested in helping with any of the items?** Great! We look forward to your participation. Start by posting your thoughts on the developer mailing list. You can subscribe to the developer mailing list [here](#)

## 1.x

### 1.4

<http://svn.apache.org/repos/asf/tuscany/branches/sca-java-1.4/distribution/src/main/release/CHANGES>

### 1.5

- Support JAX WS holder pattern and inout parameters
- Support rpc/literal wsdl
- Review RRB implementation and roll out to other bindings where appropriate
- Move policy over to binding wires away from handlers
- JMS binding implementation flexibility
- Geronimo integration
- JEE spec support
- Finish of Sprng spec support - [TODOs](#)

## 2.x

### 2.0 - M1 - OSGi Runtime

- Define and package a clean SPI
  - [Identify the SPIs](#)
- Tuscany delivery story (end to end)
  - developing core
  - developing extension
  - developing samples/demos
  - [building and releasing a distribution, how do we do this much more quickly than we do now TUSCANY-2813](#)
  - using a distribution, ease of use of samples, tooling integration etc.
- How to adopt OASIS spec features (2 functional)
  - [what features have changed](#)
  - [how to do changed features map to compliance testing](#)
  - how to extend the infrastructure, XSD, processors, runtime, tests
  - how to do compliance testing
  - how to improve our documentation
- Extensions
  - implementation.java
  - binding.sca
  - binding.ws

- Scenarios
  - samples/calculator
  - samples/calculator-webapp
  - samples/calculator-equinox

## 2.0 - M2 - Assembly compliance

- XSD/API
- Processors
- Assembly compliance tests
- Policy
- [Endpoint TUSCANY-2809](#)
- [launching the tuscan runtime TUSCANY-2810](#)
- Turn callback support into interceptors [TUSCANY-2808](#)
- Domain/Node [TUSCANY-2811](#)
  - Use [scenarios](#) to better understand how domain/nodes are used
    - multiple contributions installed to a SCA domain
    - cross-contribution artifact reference resolution
    - node configuration to represent runtime capabilities
    - nodes running SCA composite applications.
  - Decompose the basic services related to SCA domain management and provide SPIs for these atomic operations.
- Any other infrastructure fixes we come across required for OASIS compliance

## 2.0 - M3 - Policy/Extension compliance

- Ws compliance tests
- Java compliance tests
- More policy work + Policy compliance tests
- More Domain/node work + Assembly compliance tests relating to domain
- Bring in remaining OASIS specific extension on upgraded infrastructure, e.g.
  - JMS, JEE, BPEL, EJB binding, Spring (is there anything else) + compliance tests

## 2.0 - M4 - Tuscany extensions

- On top of OASIS compliance bring in any Tuscany specific extensions we think we need?

## 2.0 - Beta

- Lock down and prep for branch