

Build Your SCA Light Distribution

Unable to render
{include} page
included page could
not be found.

⚠ This page will move to the website when it is complete. Please help to complete it. Thanks.

What this is about

As of release 1.3 Tuscany SCA is distributed via one binary distribution that includes everything. Based on user feedback, choice of a smaller distribution packages is desirable. Tuscany community is in the process of addressing this requirement. Please see mailing list for that discussion. This is not the purpose of this page.

Tuscany users have asked for a write-up that highlights module dependencies. This enables them to build their own distribution which may have a lower or higher granularity than the pre-packaged binary distributions. Purpose of this page is to help users understand how to build their own tuscany SCA light distribution based on module dependency knowledge that is shared here.

Therefore:

This page is not about building Tuscany binary distributions to distribute

This page is about understanding dependencies so that users can build their own distribution

How to build your own distribution

Let's first understand the different categories of module types in Tuscany. This understanding will help us determine how to put the modules together to have a runnable and useful system.

Tuscany Modules Categorized

SCA programming Model: Assembly, Deployment, Management

SCA provides a programming model for Construction, Assembly, Deployment and Management of network of services. Modules under these categories support the SCA programming model.

Tuscany Container

Modules in this category are specific to running SCA in a Tuscany environment.

Tuscany container also provides plug points for different host types, bindings, implementations, policies as well as deployment and management (contributions and domain handling).

Policy

Module in this category include support for enabling policies as well as policy types.

Binding

Modules in this category include various bindings that support different protocols. Typically Bindings are used with SCA programming model, Tuscany container and some component implementation types.

Implementation

Modules in this category include support for different component types. Basic SCA environment is needed to use any of these modules.

Host Environment

Modules in this category include support for different host environments, such as Tomcat, Jetty, etc.

Module Category Dependency

This section explains the dependencies amongst various categories. This information can be used to define the scope of modules needed to perform different tasks using Tuscany.

Category	Dependency
SCA Programming Model	None. It runs in any container
Tuscany Container	SCA Programming Model. Note that Tuscany Container has plug points for various host environments, but is not dependent on any.
policy	Tuscany Container and SCA PM
Binding	Tuscany Container and some host environment depending on the protocol type
Implementation	Tuscany Container

Tuscany Modules Break Down by Category

Note All modules that represent models are highlighted in green.

Category	Description	SCA constructs	Tuscany Modules	Detail
SCA PM	SCA assembly	composite componentType	assembly assembly-xml assembly-xsd	Model interfaces for the SCA assembly XML Reader, Writer and Resolver XSDs for the assembly model
SCA PM	Java Interface	interface.java	interface interface-java interface-java-xml	Model for interface/operation across Java and WS Model for java interface Reader/Writer for interface.java
SCA PM	WSDL Interface	interface.wsdl	interface interface-java-jaxws interface-wsdl interface-wsdl-xml xsd xsd-xml	Interface processor for JAXWS annotation WSDL port type as the interface definition Reader/Writer for interface.wsdl Model for XSD artifacts Reader/Writer to load/save XSDs
Policy	Policy	intent policySet	policy policy-xml policy-xml-ws policy-logging policy-security policy-security-jsr250 policy-security-ws policy-transaction definitions definitions-xml	Model for policy Read/Writer for policy XML Reader/Writer for WS policy XML
Tuscany Container	Contribution Processing		contribution contribution-xml contribution-impl contribution-java contribution-namespace contribution-osgi contribution-resource	Process SCA contribution (archives) and build the model for containing artifacts
Tuscany Container	Extensibility		extensibility extensibility-equinox	Provide the extensibility to plugin tuscan extensions such as utilities, XML processors, binding providers, implementation providers and databindings
Tuscany Container	Runtime core and SPI for extensions		core core-spi core-databinding	Core activates SCA composite and build up the invocation chain between SCA services and references over the bindings
Tuscany Container	Databinding		databinding databinding-axiom databinding-fastinfoset databinding-jaxb databinding-jaxb-axiom databinding-json databinding-saxon databinding-sdo databinding-sdo-axiom databinding-xmlbeans	A framework to deal with data representation and transformation
SCA PM	SCA API		sca-api	SCA Java common APIs and Annotations
Tuscany Container	SCA Node API, Impl and Launchers		node-api node-dynamic node-impl node-launcher node-launcher-equinox thirdparty-library node-launcher-webapp node-manager implementation-node implementation-node- runtime host-embedded	Api used to Start/Stop SCA node which bootstraps SCA runtime to run the application.
Tuscany Container	SCA Domain Manager		domain-manager workspace workspace-impl workspace-xml	Define and manage domain
Tuscany Container	Serviceability: Monitoring, Logging and Tracing		monitor monitor-logging tracing-aspectj	
Host Types	Host environments		host-http host-jetty host-tomcat host-webapp host-webapp-junit	HTTP protocol support for all http-based bindings, more can be added through host SPIs.
Binding	Feed bindings: Atom RSS	binding.atom binding.rss	binding-atom binding-atom-abdera binding-rss binding-rss-rome	Feed support, including atom and rss protocols. Needs http type host environment.

Binding	JSONRPC binding	binding.jsonrpc	binding-jsonrpc binding-jsonrpc-runtime	JSONRPC for AJAX. Needs http type host environment.
Binding	RMI binding	binding.rmi	binding-rmi host-rmi	RMI
Binding	JMS binding	binding.jms	binding-jms binding-jms-runtime	JMS
Binding	EJB binding	binding.ejb	binding-ejb binding-ejb-runtime host-ejb host-openejb	EJB
Binding	CORBA binding	binding.corba	binding-corba binding-corba-runtime host-corba host-corba-jee host-corba-jse host-corba-jse-tns	CORBA
Binding	GData binding	binding.gdata	binding-gdata binding-gdata-runtime binding-gdata-gsoc binding-gdata-runtime-gsoc	GData
Binding	HTTP binding	binding.http	binding-http binding-http-runtime	HTTP
Binding	Web Service binding	binding.ws	binding-ws binding-ws-axis2 binding-ws-axis2-policy binding-ws-wsdlgen binding-ws-xml	WebServices
Binding	SCA binding	binding.sca	binding-sca binding-sca-xml binding-sca-axis2 binding-sca-corba binding-sca-jms endpoint	One of the binding-sca-(types) is needed. The default is corba?
Binding	Java implementation	implementation.java	implementation-java implementation-java-xml implementation-java-runtime	Java component types
Implementation	BPEL implementation	implementation.bpel	implementation-bpel implementation-bpel-jbpm implementation-bpel-ode	Use BPEL components in a composite application. Includes support for Apache ODE.
Implementation	EJB implementation	implementation.ejb	implementation-ejb	EJBs as components in a composition
Implementation	Scripting language based implementation	implementation.script	implementation-script	enables using Groovy, JavaScript, Ruby, Python as components.
Implementation	Spring implementation	implementation.spring	implementation-spring	enables inclusion of Spring Application Context in a composite
Implementation	Widgets	implementation.widget	implementation-widget implementation-widget-runtime	Widget support for web20
Implementation	OSGi implementation	implementation.osgi	implementation-osgi	enables incusion of OSGI bundles in a composite
Implementation	HTTP Resource implementation	implementation.resource	implementation-resource implementation-resource-runtime	HTTP resource type component
Implementation	XQuery implementation	implementation.xquery	implementation-xquery	Xquery component
Implementation	Data Access Services		data-api data-engine-helper implementation-das implementation-data-xml	

Relationship of Modules and OSGI Bundles

There are various ways and views for creating OSGI bundles. This can be discussed under "OSGI bundles" topic which can define how modules in a bundle are versioned together, start together and share class loaders. For the sake of understanding structure of modules, we can say that each module is an OSGI bundle. These can be pulled together to create coarser grain bundles if needed.

Use case Examples for Tuscany usage

Use Tuscany as a runtime for SCA development

Example: Develop SCA applications in Tuscany based on POJOs

Basic Category of Modules Needed: SCA PM, Tuscany Container, implementation.java, SCA default binding

Adopt the programming model to create a component model for your own runtime

Example: Use SCA for configuration of and ESB engine

Basic Category of Modules Needed: SCA PM

Embed Tuscany within an application server

Example: Embed Tuscany with an application server such as Geronimo

Basic Category of Modules Needed: SCA PM, Pick the features that are needed to run on the given platform, binding.ejb, implementation.java, etc. In some cases you need to use host SPIs to provide the platform specific behavior. For examples check out host_* modules.

Create tools for SCA

Example: Write a development tool for SCA

Basic Category of Modules Needed: SCA PM

Other scenarios

Please help add other interesting scenarios

How to create your own distribution

TBD

Link to the page describing how to create a POM.xml and use maven to generate distribution.

Talk about how maven figures out third party dependencies.