NordiaSoft SCA Architect 2016

Product_code: NordiaSoft_SCAArchitect_2_3_0
Target Java Runtime Environment: Oracle JRE 1.8
Date: January 2016

SCA Architect

NordiaSoft SCA Architect is the modeling tool used by developers to compose and assemble software components into applications. Based on a Model-Driven Development (MDD) concept, SCA Architect models abstract all of the SCA programming specifics from the developer, enabling code generation of fully functional SCA components using POSIX C++. It's powerful code generation engine employs a Zero-Merge philosophy that allows developers to remodel existing components, and regenerate code without having to merge two versions of the source code.

With over 180 validation rules, SCA Architect models are thoroughly checked for compliance and any deviation is reported and suggestions provided. Using a Drag-and-Drop approach, the developer can reuse previously created models and adapt them to new situations, greatly speeding the development process.

new SCA Architect 2016 new features at a glance

SCA Architect 2015 contains a wide range of new features and enhancements, including deployment channels, automatic code file separation, user-defined library creation, automated creation of software dependencies, notification of property change, support for data files, support for file references, support for limited number of input arguments, build specification support for Core Framework components, enhanced support for heterogenous platforms, dynamic interface access and customizable IDL compiler invocation.

Rapid Application Development (RAD) Philosophy

SCA Architect was designed from the ground up to allow fast and short development cycles. Developers can quickly generate fully functional source code from models, launch generated components, and go back to modeling. Being able to run modeled components early in the process allows developers to quickly assess the footprint and boot-time requirements. It also encourages developers to develop iteratively with very small iterations that incrementally add behavior to the overall system under development.

Eclipse-based modeling tool

SCA Architect is a graphical modeling tool for the SCA. It allows developers to model applications and nodes made of several SCA components. It generates C++ source code for fully functional and SCA compliant components. SCA Architect is provided as a plug-in for the universally adopted, platform independent, Eclipse™ framework. Developers quickly become productive with SCA Architect thanks to the well-known interface of Eclipse.

Unambiguous graphical modeling language

SCA Architect provides a precise modeling language that can be used to express every SCA concept graphically. Developers don't have to rely on obscure property panels to model different types of connections, collocation requirements, or uses-device relationships. Models are used to generate Core Framework independent valid SCA XML profiles.

Real-Time Model Validation

The real-time model validation eliminates time-consuming retrofits to correct models after hours of modeling. Real-time validation detects errors early which greatly accelerates the creation of fully functional compliant SCA components. In-depth experience and expertise with the SCA has provided NordiaSoft SCA Architect with a very large set of validation rules.

Importing Models

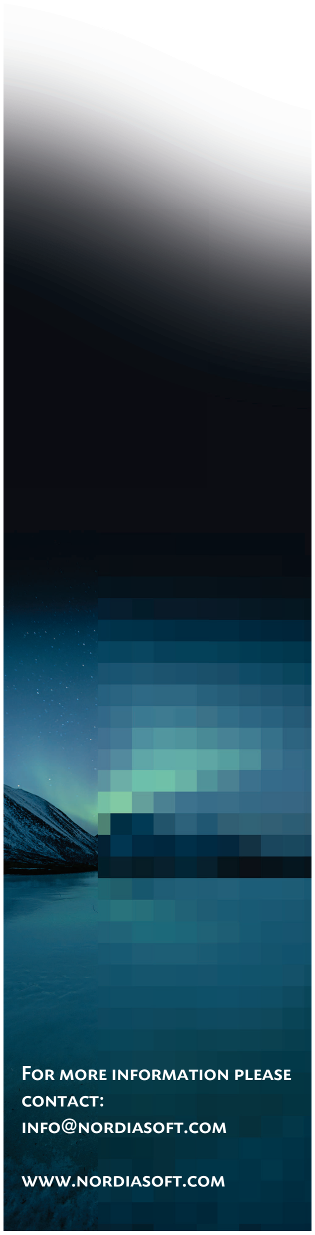
NordiaSoft SCA Architect can also be used to import and validate pre-existing SCA domain profiles (sometimes called reverse-engineering). It allows developers to import and repair legacy domain profiles. NordiaSoft SCA Architect's unique re-factoring features can also automatically correct errors from a number of suggested fixes.

Enhanced Support for Connections

SCA Architect offers a modeling language that already allows developers to graphically express every concept of the SCA, including concepts such as uses-device connections and device-that-loaded connections. It now provides quicker access to specific instances of Devices and services for connections with application components. This allows developers to make connections in one single step instead of several steps.

Test Automation capabilities

SCA Architect provides a command line tools allowing the automation of model testing. Using these tools, source code and XML SCA Domain Profile files can be generated from models in order to package and deploy assemblies to be tested on an SCA platform.



For more information please contact:
info@nordiasoft.com

www.nordiasoft.com

NordiaSoft SCA Architect 2016

Product_code: NordiaSoft_SCAArchitect_2_3_0
Target Java Runtime Environment: Oracle JRE 1.8
Date: January 2016

Sophisticated Project Browser

The project browser has been designed to simplify the user experience for very large projects and for programs with several projects. It allows the developer to access components instantly using a Quick Search Bar. Developers can also drag-and-drop directly from the project browser to the canvas instead of using the canvas drawers. The browser also supports the creation of logical groups of components for a more organized view.

Shared Projects

Allow developers to use modeling elements (ports, properties, etc.) that come from a different SCA Architect project, without having to import and duplicate modeling elements. This feature helps developers create projects that contain common elements that are reused across several other projects.

Comprehensive Code generation

In addition to SCA wrapper C++ source code for any kind of IDL interface, SCA Architect also generates source code with behavior for every type of assembly component: Resource, Resource Factory, Assembly Controller, Device, Loadable Device, and Executable Device. The generated code implements the full behavior for many of the required SCA APIs and is Core Framework independent. Generating code from models drastically reduces the amount of lines of code a developer must write. Code generation generates source code that has been tested and therefore eliminate defects which accelerates development while reducing costs.

Executable Components out-of-the-box

The code generator produces components that can be executed without having to perform a single modification to the source code. The code generator does not only generate structural source code like most modeling tools do. It also generates very sophisticated behavioral source code that implements required SCA APIs and can be executed out-of-the-box.

- Automatically takes care of command-line arguments: SCA-specific and user defined
- Automatically takes care of POSIX signals as required
- Automatically takes care of component registration
- Automatically takes care of tracking connections
- Facilitates Integration with 3rd Party Software by facilitating calls to pre-existing business logic

Multiple Operating Environment Support

The generated source code is guaranteed to compile and run on any Operating Environment supported by the SCARI Core Framework. This characteristic is paramount as it helps developers to future-proof their investment in building SCA components. The SCARI Core Framework support a very large combination of operating systems (single or multiple address spaces, with/without shared libraries), processors (single or multiple cores, System-on-chip or standalone processors), and CORBA Object Request Brokers.

Enhanced Support for Multi-Platform Components

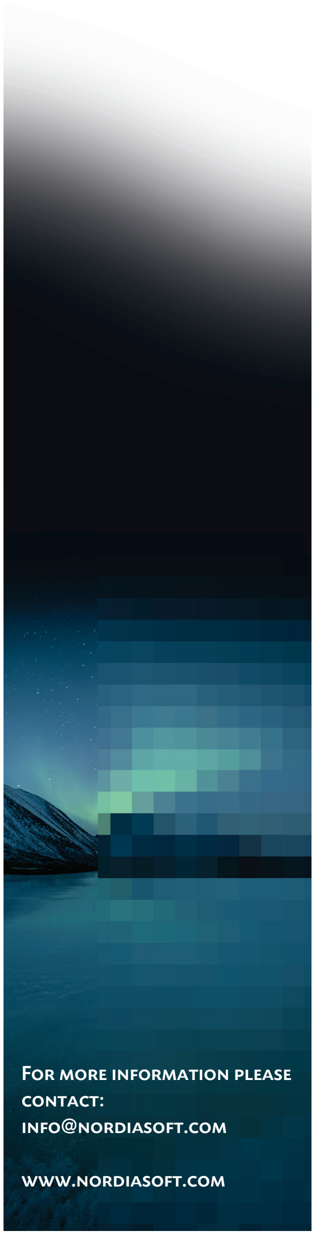
SCA Architect already generates code that support multiple operating environments by allowing components to have several implementations. This feature allows developers to use a pre-defined system variable to define the paths where the implementations of the components are generated. It allows developers to use the same name for each implementation file without conflicts. This makes it easier to not only produce, but package and deliver components which contain several implementation binaries.

Provides unique Zero-Merge Code Generation

SCA Architect breaks new ground with a Zero-Merge code generator. Using this code generator, developers can easily iterate through short development cycles in which models are iteratively improved without ever having to merge source code between each iteration. In short, models can always be modified and refined after source code has already been generated and specialized. Zero-Merge completely eliminates the need for the painful process of merging different versions of source code.

Model Documentation

Generates documentation describing each modeling elements used in a project. It provides detailed documentation about the properties of each model as well as graphical representations of the models. Links to external documentation can be added to SCA Architect models. All the documentation is generated based on templates that can be customized. This feature helps to quickly produce documentation about the models.



For more information please contact:
info@nordiasoft.com

www.nordiasoft.com

NordiaSoft SCA Architect 2016

Product_code: NordiaSoft_SCAArchitect_2_3_0
Target Java Runtime Environment: Oracle JRE 1.8
Date: January 2016

Configuration Management

Provides integration with the Eclipse configuration management plugin. SCA Architect helps the developer by automatically indicating which generated files are associated with a SCA Component which ensures all the required files get tracked.

Sophisticated Design Patterns

SCA Architect can be used to generate source code for a number of different use cases. For instance, it can generate a threaded data pump that routes data to application components connected via ports. It can also generate source code to route input data through signal processing functions and route the processed data to the appropriate out port and broadcast the data to all of components connected to an output port. Of course, all of the generated source code is thread safe and synchronized with the life-cycle of each component. The latest pattern made available allows developers to request input data to be queued inside a processing component. This is especially useful to avoid creating an empty pipeline of components when communication is done via two-way IDL interfaces.

Deployment Flexibility for address spaces

Communication performance is at the heart of the SCARI products. This feature allows a developer to specify co-location restrictions that would place several SCA components in a same address space without the use of a component factory. This feature even allows application components to be deployed in the same address space as a device component by deploying all the components in a same ExecutableDevice. This deployment ability can easily provide one order of magnitude faster communications between components. The feature relies on the support for shared libraries from the operating systems and on ORBexpress, the Object Request Broker (ORB) from Objective Interface Systems (OIS).

Internal Port Routing

Allows developers to specify the data-processing path that goes from an input port to an output port of a component that transforms the incoming data. This frees the developer from having to manually write the code to perform this very common task for signal processing applications such as Software Defined Radio waveform applications. Of course, starting and stopping a component as well as connecting or disconnecting ports of a component has an impact on the data being processed. The generated code is synchronized with the life cycle of SCA components which uses multiple threads of execution.

Threads on Ports

Allows developers to specify that a Device component needs a thread to acquire data from a hardware device before it broadcasts the new data to other components with which it is connected. Of course, starting and stopping a component as well as connecting or disconnecting ports of a component has an impact on the data being processed. The generated code is synchronized with the life cycle of SCA components which uses multiple threads of execution.

Proxy ports and properties

Allows a developer to define that a component has ports/properties that act as proxies to ports/properties of other components. This is a very common design pattern used in component-based development and SCA Architect generates code that completely handles the proxying functionality.

Component Unit Tests

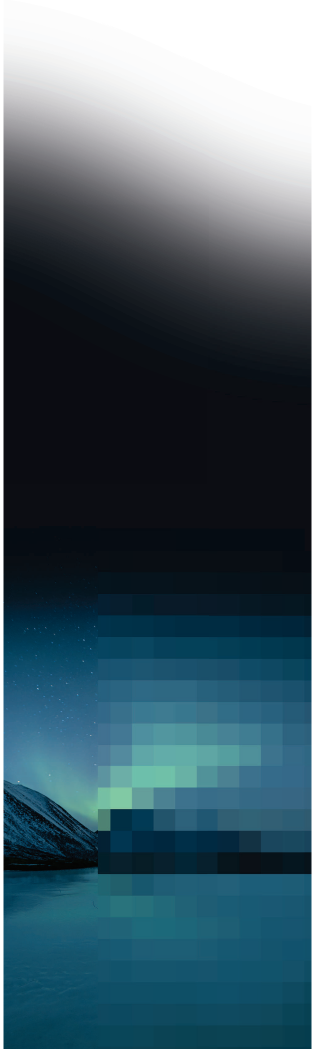
SCA Architect generates unit tests directly from the models. Unit tests serve several purposes. It allows developers to ensure the generate source code is free of defects. It also enables the developers to perform regression testing once domain-specific business logic is added to the generated code. Finally, it can be used to more efficiently work with debugging and memory analysis tools, and performance profilers to chase bugs introduced by business logic.

Generated Source Code Specialization

Developers can specify which modeling elements for a component will not need to be specialized. In such a case, the code generator will only produce a base source file without producing a specialization file. This allows the developer to keep control over the number of source files produced by the code generator.

SCA Compliance

SCA Architect comes with support for a plugin that provides a seamless integration with Reservoir Lab's SCA compliance analysis tool called R-Check SCA™. Source code generated by SCA Architect is automatically validated by R-Check SCA which performs a static analysis of the source code. R-Check SCA is used by the US DoD certification testing lab for SCA compliance. The tool accelerates SCA development by supporting integrated testing across file types, ensuring completeness and consistency.



For more information please contact:
info@nordiasoft.com

www.nordiasoft.com

NordiaSoft

NordiaSoft SCA Architect 2016

Product_code: NordiaSoft_SCAArchitect_2_3_0
Target Java Runtime Environment: Oracle JRE 1.8
Date: January 2016

Build specifications

The code generator generates build files based on build specifications modeled in SCA Architect and applied to different application and node components. Build specification support system and environment variables to specify paths and selections. They can also be shared between SCA projects which helps to ensure a consistent build environment across developers, departments, and companies. The code generator can create Makefiles and project files for the INTEGRITY Multi IDE from Green Hills Software.

Source code header/footer dynamic content

SCA Architect allows adding header and footer content to all source code generated by the tool. In addition to static content, header and footer can have dynamic content that is replaced when the source code is generated. This dynamic content can be defined using pre-defined template variables and system variables. This feature helps to add information such as filename, date, author, company, etc.

Automatic code file separation

Facilitates the export of components that support multiple operating environments. Developer can specify that all the code files need to be exported in directory that bares the implementation name. This avoids the manual step to define a specific directory for each code file.

User-defined library creation

Offers support for the creation of user-defined libraries for user-defined source code. This is supported via Makefiles that can be specialized.

CORBA IDL profiles

Developers can easily generate source code that uses different Object Request Brokers (ORBs) for different implementation of a component. This is done by associating component implementations with different sets of IDL interfaces called IDL profiles. Profiles can also be defined as managed or unmanaged. Using a managed profile means the code generator will automatically generate the stubs and skeletons required by a component. Developers can also use an unmanaged profile if they don't want the code generator to produce stubs and skeletons. In that case, the developer must specify where the stubs and skeletons are located.

Refactoring capabilities

SCA Architect offers refactoring capabilities to rename, remove, move, or replace model elements, and to apply a suggested solution in order to fix a validation problem with one or many components. SCA Architect also offers views to let the developer see where modeling elements are being used and vice-versa. These refactoring capabilities help to quickly fix or properly modify models for a new development environment or a project change.

Saving Pre-Connection Log Messages

When a component gets started, if it sends log messages to a Log service that has not been launched yet, all the log messages are saved in a file in case the component crashes before it gets connected to a Log service. These messages can be very useful to a developer for debugging purposes.

Automatic generation of an Assembly Controller

Every SCA application is required to implement an Assembly Controller component that is responsible for coordinating the assembly of components that makes an application. SCA Architect can easily generate the full source code for a generic Assembly Controller component even when such a component must have proxy ports and proxy properties. This feature helps the developers accelerate the development by reducing the amount of source code that needs to be implemented every time the application start using new properties, new ports, or even new components.

Automatic generation of a ResourceFactory

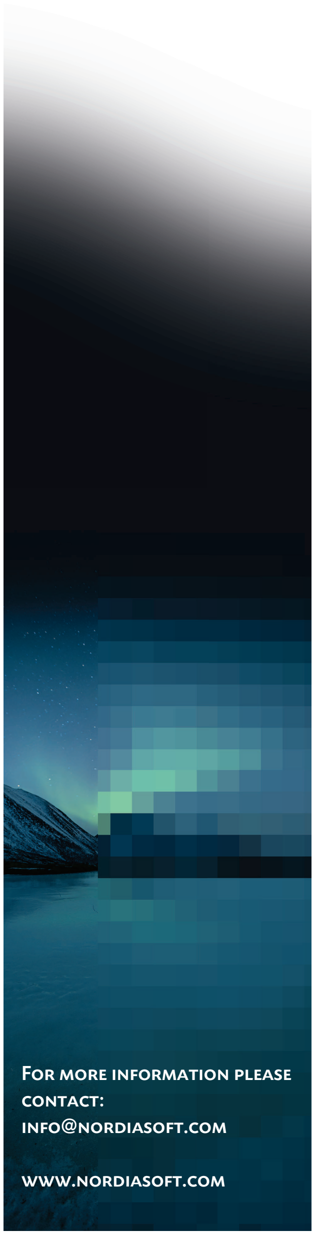
Code generation is such that components are broken in two pieces: a launcher and a servant. Thanks to this design pattern, SCA Architect provides a wizard that can easily recombine previously generated components into one-to-many ResourceFactory components, without having to change any source code from the existing components. This feature enables the developers to make late decisions in terms of how many address spaces should be used by an application. This can have a huge impact on the speed of communications between components of an application.

IDL integration

IDL file for the IDL interface implemented by a modeling element can be accessed directly from the models. This avoid spending time for searching IDL files.

Supports user-defined IDL

Developers can import user-defined IDL files into IDL Profiles and use the new interfaces to define ports and generate the associated source code. This provides the flexibility designers need to innovate with new domain-specific interfaces.



For more information please contact:
info@nordiasoft.com

www.nordiasoft.com

NordiaSoft SCA Architect 2016

Product_code: NordiaSoft_SCAArchitect_2_3_0
Target Java Runtime Environment: Oracle JRE 1.8
Date: January 2016

Easily create application and node packages

SCA Architect can be used to package all of the required SCA artefacts into a single ZIP archive file. This makes it easy to transport all the pieces of an SCA application or node to an SCA radio for installation. The NordiaSoft Radio Manager also supports the installation of SCA applications packaged as an archive file. The new version of this feature also supports the TAR format.

new Deployment channels

Supports standard SCA channels using the Domain Manager Configuration Editor. The export feature now generates the Deployment Platform Descriptor (PDD) domain profile file. The channel feature provides the ability to restrict the deployment of an application to Devices that are part of a specific channel.

new Automated creation of software dependencies

Automates the creation of SCA software dependencies for user-specific libraries that need to be deployed with their components. Instead of doing all the steps to create the software dependency and add it to the component by himself, the developer simply select which user libraries the tool must create software dependencies for.

new Notification of property change

Automates the generation of an event when a property value is changed. This is used to notify other components of changes. The Developer can choose for which properties an event must be sent to an Event Channel.

new Support for data files

Supports a new code file type "Other" for data files that simply need to be loaded on the target alongside the main component binary. This code file type allows the deployment engine to avoid special handling associated with SCAv2.2.2 code file types (executable, shared library, kernel module, driver). The new code file type can be used for data input files, configuration files, etc.

new Support for file references

Generates the source code to be able to read the input arguments to obtain a reference to a file that was specified as a software dependency of the component. At execution, the deployment engine provides the file a references. The file reference can take the form of a fully qualified SCA path or an IOR to an opened SCA File.

new Support for limited number of input arguments

Generates code that can receive its input arguments via an input file as well as via the command line. This is useful when the operating environment is limited to a small number of input arguments (e.g. VxWorks). When the input arguments are provided via a file, the filename is specified as an executable parameter.

new Build specification support for Core Framework components

Generates build files for Domain Manager, Device Manager and service components in order to allow them to be built with the same build specification than the other components generated by SCA Architect. This allows the developer to build all the components using the same customized build recipe.

new Improved integration with CDT

The generated code is aligned with how the CDT handles some pre-compiler directives. This eliminates warning and errors in the CDT editor.

new Enhanced support for heterogenous platforms

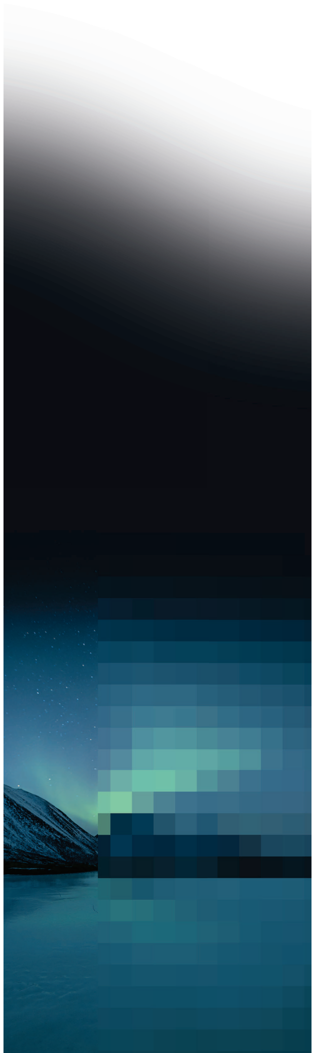
Developers can choose to disable the automatic build for implementations that cannot be built in a specific development environment. This allows developers to not have all the cross-compiling tool chains installed in a single development environment. SCA Architect does not attempt to generate the source code or export implementation associated with the disabled build specification.

new Dynamic interface access

Generates code that provides runtime access to the various interfaces supported by a component. Access to the various interfaces is implemented via special port. This is useful for cases where a component does not always implement all interfaces depending upon runtime conditions. It can also be used to provide more/less specialized versions of an interface at runtime.

new Customizable IDL compiler invocation

Enhanced support for automatic generation IDL stubs/skeletons. The developer can now choose different flags for the IDL compiler.



For more information please contact:
info@nordiasoft.com

www.nordiasoft.com