

# SCARI Software Suite

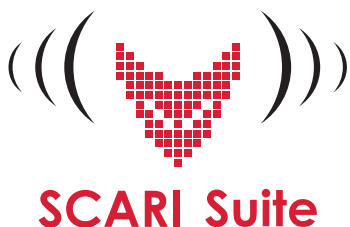


## THE SCARI SOFTWARE SUITE

The SCARI Software Suite is a comprehensive Integrated Development Environment (IDE) for heterogeneous embedded distributed systems. This VIAVI solution embraces the concepts of Component Based Development (CBD) and Model Driven Engineering (MDE).

The SCARI Software Suite allows developers to model and create software components that run in real-time and are independent from the underlying operating environment.

Used internationally by platform and radio manufacturers, as well as application providers, the VIAVI SCARI Software Suite reduces development risk and time-to-market, creating top quality software-defined products.



The SCARI Software Suite is the original integrated development environment for projects based on SCA version 2.2.2.



## SCARI GT Core Framework

The battle proven Core Framework (CF) of choice for COTS platform providers and radio manufacturers, supporting the largest combination of operating systems, object request brokers, and processors. Not being subject to ITAR restrictions, the SCARI CF runs in thousands of battlefield deployed radios from different international radio manufacturers.

Customers worldwide have benefited from the expertise of VIAVI engineers using the SCA. Used for research projects, for development purposes, and for actually fielded radios, SCARI GT is an SCA Core Framework based on the SCA 2.2.2 open standard which you can always rely on.

An SCA Core Framework allows developers to create software components that can be deployed in a large combination of operating environments.

Built from the ground up for embedded platforms, the SCARI GT Core Framework for SCA 2.2.2 is the third-generation core framework. It was designed for optimum boot time performances.

With the lessons learned from the JTRS-tested SCARI-Open core framework, SCARI GT is a commercial product available for many operating environments and has been used with several System-on-Chips solutions.

It is the most widely adopted Commercial SCA 2.2.2 Core Framework and is available pre-integrated with a number of generic SCA platform providers. SCARI GT is the cornerstone of the VIAVI SCA development solution and is the only battle proven commercial off-the-shelf (COTS) Core Framework. Not being subject to ITAR restrictions, the SCARI CF runs in thousands of radios deployed by different international radio manufacturers. The SCARI CF was developed hand-in-hand with the JTAP; the US DoD official compliance-testing tool.

### Features

- Boot time optimizations
- Small footprint
- Support for all major RTOS
- Support for ORBexpress RT<sup>®</sup>
- Used in thousands of radios currently deployed in the battlefield
- Not ITAR controlled

## **Operating Environments**

Designed with portability in mind, the SCARI GT Core Framework is used on a number of operating environments: INTEGRITY, VxWorks, different desktop Linux distributions, QNX, TimeSys, and MontaVista embedded Linux, and new portable platforms such as Android. SCARI GT can also be ported to operating systems with single or multiple address spaces, with or without a dynamic loader. It has also been used for a number of processors from the following families: XScale, x86, PPC, and ARM.

SCARI GT supports a number of different ORBs including ORBexpress RT, which is the most widely used real-time, secure, and high-performance ORB used in military radio platforms. The VIAVI team also has the required expertise to add support for any new operating environment.

## **High-Speed Core Framework**

The SCARI GT Core Framework provides a full implementation of all the SCA 2.2.2 framework interfaces. It implements exceptional real-time features to minimize the boot time of an SCA system. It transparently transforms indirect connections into faster direct connections. It uses a unique caching system to avoid long delays required for finding deployed components. The SCARI GT Device Manager even allows SCA Devices to be collocated into a single address space, which is significant for accelerating the boot sequence of a node. In fact, SCARI GT components can all be linked into a single address space to further accelerate the boot sequence.

## **Extended Introspection Features**

SCARI GT supports extended introspection features used by VIAVI Radio Manager monitoring tool. Using SCARI GT, developers can obtain detailed deployment information of the application which allows them to determine where components have been deployed and how they have been interconnected.

## **Debugging Support**

SCARI GT is provided in two binary forms: Release and Tracer. The Release version is compact and ready for embedded deployment. The Tracer version is instrumental in debugging code that produces several levels of tracing messages that can be selectively turned on or off. SCARI GT can also save the logging messages to a file which are produced before a Log service becomes available.





## SCA Architect

SCA Architect is an Eclipse-based IDE that allows you to create, validate, and debug SCA software components and applications. It provides a Zero-Merge source code generator and offers SCA behavioral source code generation. The generated source code builds for all supported operating environments without regenerating. It utilizes a visual modeling language that allows developers to unambiguously graphically express every concept of the SCA.

SCA Architect is the Integrated Development Environment (IDE) of the SCARI Software Suite. It covers the complete SDR development life cycle including modeling and validation of components, full behavioral generation of SCA compliant C++ source code, real-time model validation as well as creating component assembly into applications or nodes.

SCA Architect is provided as a plug-in for the universally adopted, platform independent, Eclipse framework. Embedded system developers will benefit from a well-known interface, making it easy to navigate between the different development phases of their projects.

### **Reusable Modeling Elements**

No need to repeat every step when creating components. VIAVI SCA Architect lets you create reusable component properties and ports, allowing you to assemble them into component types to create similar components much more quickly. This feature is also extremely useful when applying design modifications to many components. A modification in one place is automatically reflected in every component of a similar type – a precious time-saving feature.

### **Real-Time Model Validation**

The real-time model validation feature of VIAVI SCA Architect eliminates time-consuming retrofits to correct early errors, thereby greatly accelerating the creation of SCA components. In-depth experience and expertise with the SCA have provided VIAVI SCA Architect with the largest set of validation rules in the industry.

### **Features**

- Unambiguous graphical modeling
- Zero-merge SCA behavioral and structural code generation
- Real-time validation and re-factoring

### **Importing and Refactoring**

VIAVI SCA Architect can also be used to model and validate pre-existing SCA domain profiles. Its powerful import capabilities will import and repair legacy domain profiles. VIAVI SCA Architect's unique re-factoring feature can also automatically correct errors by offering developers a number of suggested fixes.

### **Unambiguous Graphical Modeling**

VIAVI SCA Architect's superior modeling capabilities allow unambiguous graphical representations of assemblies, capturing containment relationships between deployed components and their target a key concept to enable the graphical representation of all types of indirect connections and host collocation relationships. VIAVI SCA Architect's modeling capabilities provide a deterministic graphical representation of assemblies.

### **Zero-Merge Code Generation**

VIAVI SCA Architect generates fully functional SCA components using C++/POSIX/CORBA that can be built and used in applications without writing a single line of code. Being template-based, VIAVI SCA Architect could be tailored to support other programming languages or code conventions. SCA Architect breaks new ground by introducing "zero-merge" code generation capabilities. Developers specialize in the behavior of a component, instead of modifying it.

Developers can at last remodel existing components, and regenerate code without having to merge two versions of the source code.

### **Configuration Management**

VIAVI SCA Architect pioneers model-level configuration management. Developers no longer must manually track each individual artifact of a model element. It allows developers to save model elements directly to a repository.

Developers don't have to save incoherent versions of those models. This feature radically simplifies configuration management.

### **Shared Projects**

VIAVI SCA Architect provides a way to reuse common modeling elements without having to duplicate them. VIAVI SCA Architect supports the Eclipse concept of shared projects. Rather than duplicating a modeling element, developers can reference projects containing shared elements. After all, "Reuse" is one of the SCA core philosophies.

# Radio Manager

A run-time monitoring tool that allows installation and control of applications, as well as visualization of component deployment. The VIAVI Radio Manager is an essential tool for debugging and testing during SCA development.

The Radio Manager is an essential tool for platform integrators and application testers. Through a block diagram representation, the integrator and tester can quickly visualize the SCA platform composition and see where each application resource has been deployed. The Radio Manager offers a runtime view of how the connections between the SCA Resources and Devices have been established.

In real-time, the Radio Manager introspects the SCA platform and reports the status of the SCA Devices and Resources, refreshing the block diagram if needed. Any new waveform being added, device failing, or connection broken will be shown which provides valuable information to the integrators and testers to reduce debugging time.

The Radio Manager is to the SCA what a debugger is to source code. In fact, the Radio Manager can be connected to any embedded SCA platform, just like source code debuggers.

## Installing and Controlling Applications

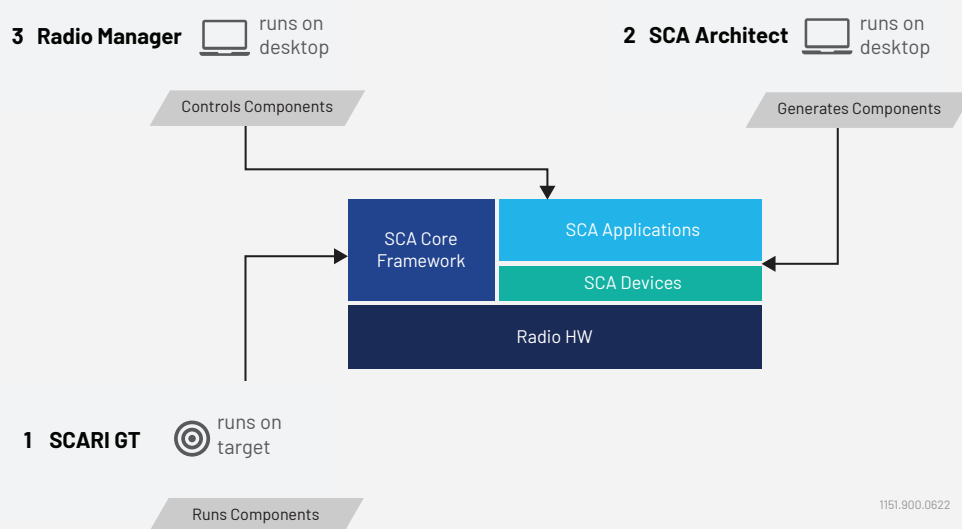
Built into the VIAVI Radio Manager is an application installer used to upload the required application artifacts onto the platform. Once done, the VIAVI Radio Manager can instantiate the application, start, configure, and stop it, and finally terminate and remove the application. The VIAVI Radio Manager provides a generic property browser that can render every type of SCA property and can change values dynamically or in batch mode.

## Features

- Install, control, and debug applications
- Introspect SCA radios
- Graphical representation of deployed SCA components
- Launch specialized Human Control Interfaces (HCI)
- Full control over deployed components

## Debugging Features

The VIAVI Radio Manager is a very useful tool to test the performance of the platform and applications. It offers full control over the deployed components. It can therefore be used to modify the values of the application parameters when testing the application under different conditions; it can shut down a complete node while applications are running to analyze how the platform reacts in specific scenarios.



## Extended Introspection

The VIAVI Radio Manager provides two different views for displaying the deployed software components. The hierarchical view uses a tree-like structure, where each node represents a deployed component. The block diagram view uses a block for each deployed component. When connected to the VIAVI core framework, the VIAVI Radio Manager can even show which components have been deployed onto other components and graphically displays how components have been interconnected.

## Specialized HCIs

The VIAVI Radio Manager allows the system integrator to launch its own specialized Human Control Interface (HCI) to control the application. This becomes particularly interesting when specific APIs are required to control the radio. Using VIAVI SCA Architect, an SCA application can be packaged with a specialized HCI that will be installed with the application on the target SCA platform. After instantiation of the application, the VIAVI Radio Manager searches for the existence of the specialized HCI, downloads it, and launches it.





## Component Development Libraries

The VIAVI Component Development Libraries (CDL) accelerate the creation of SCA compliant components. The CDL lowers the learning curve of the SCA (and speeds up development) by providing generic SCA components, designed to shield developers from the intricacies of the SCA and CORBA.

Using the CDL generic components, drastically reduces the number of lines of code required to create SCA compliant components, simplifying the generated code and associated testing, thereby reducing the development cycles. Also available as shared libraries, the CDL enables the creation of very small footprint SCA components.

The CDL is core framework independent and is available for the most popular operating environments.

### Generic Property Set

The CDL generic components provide a framework for implementing properties, which shield developers from the intricacies of the SCA and CORBA. Using the CDL Generic PropertySet, configuration requests are transformed into simple invocations to C++ member functions. Component property values are kept as simple C/C++ native data types. The CDL Generic PropertySet implements all the SCA required behavior for configuration requests, increasing SCA compliance.

### Debugging Support

The CDL is provided in two binary forms: Release and Tracer. The Release version is compact and ready for embedded deployment. The Tracer version is instrumental in debugging code that produces tracing messages of varying levels that can be selectively turned on or off.

### Features

- Generic component libraries
- Reduces lines of code and memory footprint
- Simplifies and speeds up development
- Core framework independent

## Generic Log and Event Ports

The CDL provides generic implementations of a Log port and an Event Channel port. These are used to report logging messages or to produce events that may be consumed by other components. They are implemented as generic ports that transparently handle all the requirements of the SCA. Developers only need to worry about producing messages; they don't need to worry about levels or connections. In fact, the CDL offers much more than the SCA required behavior, allowing components to save the logging messages produced before a Log service becomes available. This unique feature provides precious debugging information that is otherwise lost.

## Generic Components

The CDL provides a generic implementation of the two types of components that SCA developers must create: the Resource and the Device. CDL components will automatically generate the SCA required log messages and events preventing developers from making costly compliance mistakes. They also implement the SCA and CORBA life cycle, avoiding other potential compliance issues. The implementation of the state behavior for an SCA Device is notoriously difficult, being one of the major sources of compliance issues. The CDL generic device takes that burden off the developer's shoulders by handling capacity allocation requests. This further contributes to the reduction in quantity of source code required for implementing a new SCA Device. The CDL Device automatically handles the three state machines required for all SCA Devices, which, when combined, lead to 25 states and close to 70 transitions.





[viavisolutions.com](https://viavisolutions.com)

Contact Us    +1 800 835 2352  
                    avcomm.sales@viavisolutions.com

To reach the VI.AVI office nearest you, visit [viavisolutions.com/contact](https://viavisolutions.com/contact)

© 2024 VI.AVI Solutions Inc.  
Product specifications and descriptions in this  
document are subject to change without notice.  
Patented as described at [viavisolutions.com/patents](https://viavisolutions.com/patents)

scari-br-sca-nse-ae  
30194264 900 112