White Paper
June 2018

DeltaV™ SIS Configuration Using exSILentia

Safety Instrumented Systems (SIS) have traditional been configured manually using different source documents such as the Safety Requirement Specification (SRS), Cause and Effect matrices (CEM), Safety Integrity Level (SIL) calculations, and I/O definition among other information.

Emerson partnered with exida to deliver a database-based solution that enables automatic configuration of safety logic based on information captured in exida's exSILentia software suite. One evident advantage of this approach is the reduced configuration effort. However, the real benefit is having a consistent configuration approach with less errors and less rework that is easily traceable back to the SRS.

The purpose of this whitepaper is to provide a high-level overview of the capabilities of the DeltaV™ SIS Configurator developed by exida. This document is not intended as a configuration guideline. For further details please refer to exida's documentation (exSILentia User Guide and DeltaV SIS Configurator User Guide).

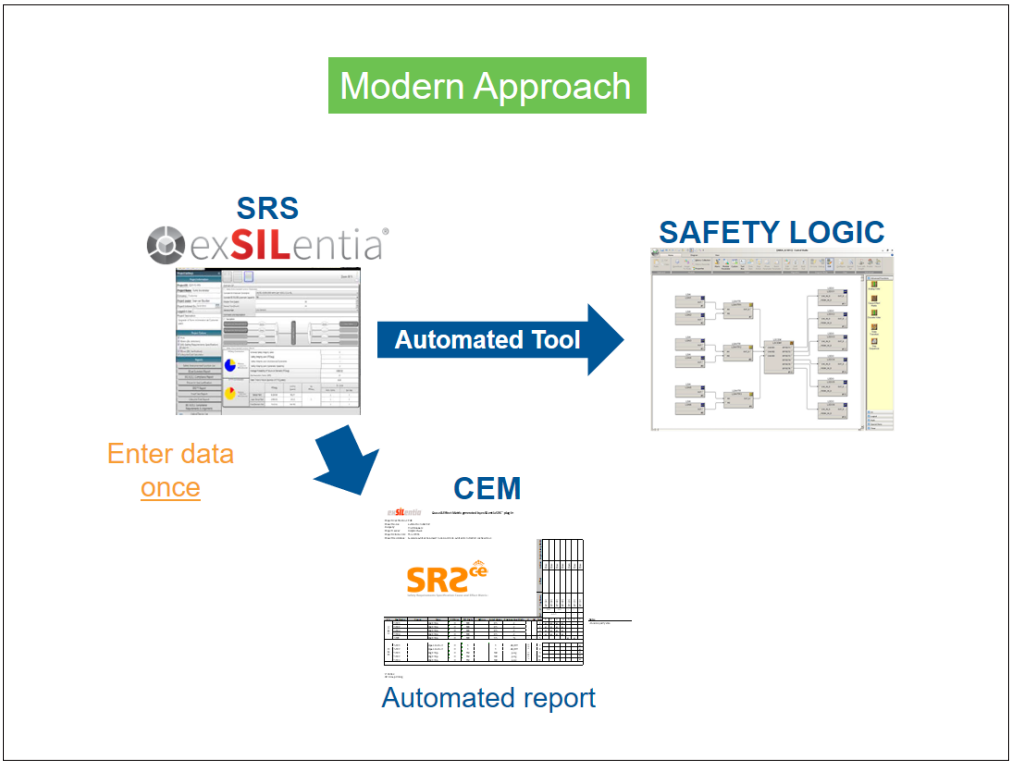

Table of Contents

DeltaV SIS Configuration Overview

The DeltaV SIS process safety system makes configuration of safety instrumented functions (SIF) very easy. The DeltaV SIS built-for-purpose function blocks can help to eliminate engineering hours required to implement safety applications. The TÜV-certified function blocks deliver powerful functionality out of the box, simplifying the implementation of complex SIS applications.

One of the advanced function blocks is the Analog Voter function block which provides advanced features to easily implement M out of N voter functions. That is, M inputs of the total N inputs must vote to trip. For example, the block can be configured as 2oo3 (two out of three) voter, where two of the three inputs must exceed the trip limit before the output is tripped. What used to take a fair amount of programming using AND and OR logical gates, it is now replaced by a standard function block configured using radio buttons and check-boxes. For example, if the application requires to prevent multiple maintenance bypasses at the same time, the user only needs to check one box. If the application requires a bypass timeout to either automatically remove the bypass after a pre-defined time or simplify provide an alert, the user again just need to select the proper options within the bypass option parameter.

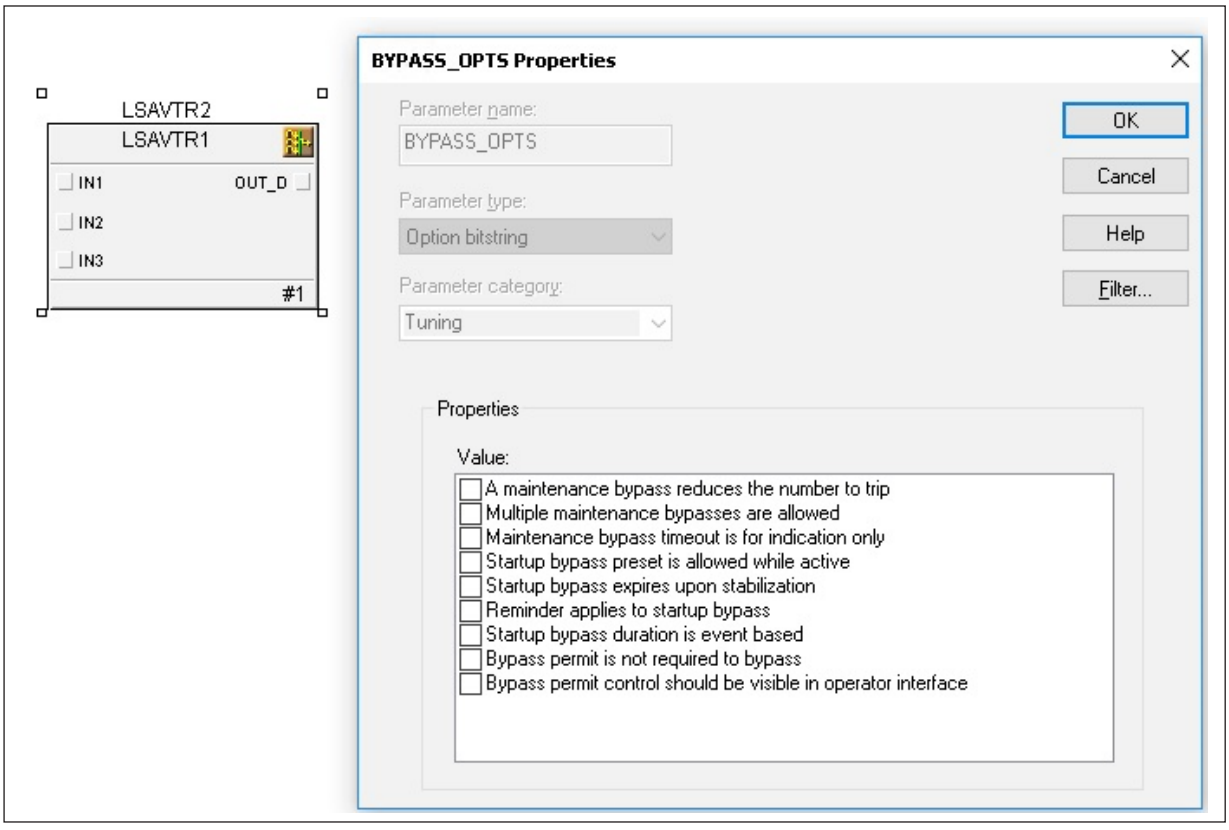

Figure 1 — *Bypass Options for Analog Voter Block.*

Trip limits, trip delays and detection types (high limit or low limit) are easily configured using parameters.

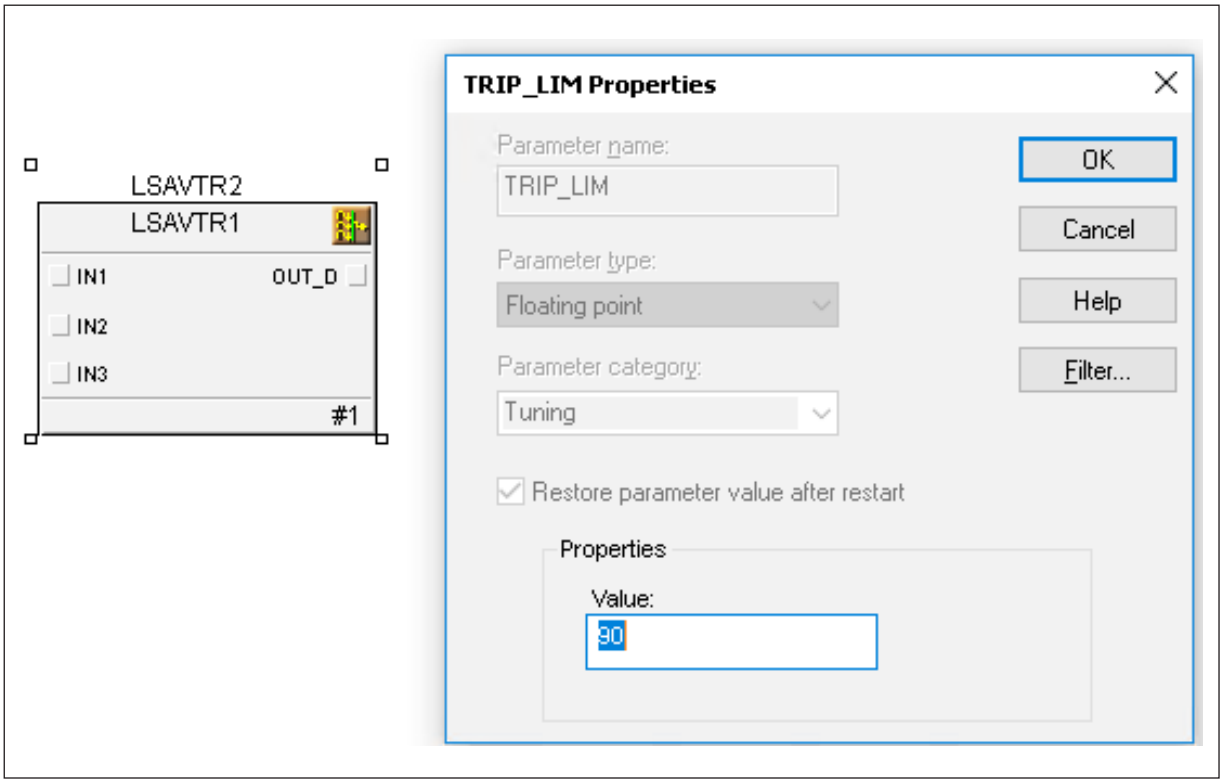

Figure 2 — *Analog Voter Trip Limit Parameter*

All the traditional programming required for implementing an analog voting function has been replaced by a few configuration settings. In the same way, there is a Discrete Voter function block with similar functionality.

Implementing a Cause and Effect relationship is done using another advanced function block. The Cause and Effect Matrix function block defines interlock and permissive logic that associates as many as 16 inputs (causes) and 16 outputs (effects). The block's MATRIX parameter defines the causes that produce each effect to trip. Figure 3 provides an example of how to configure an 8x3 CEM. Defining the trip logic is as simple as selecting the proper intersections in the MATRIX parameter.

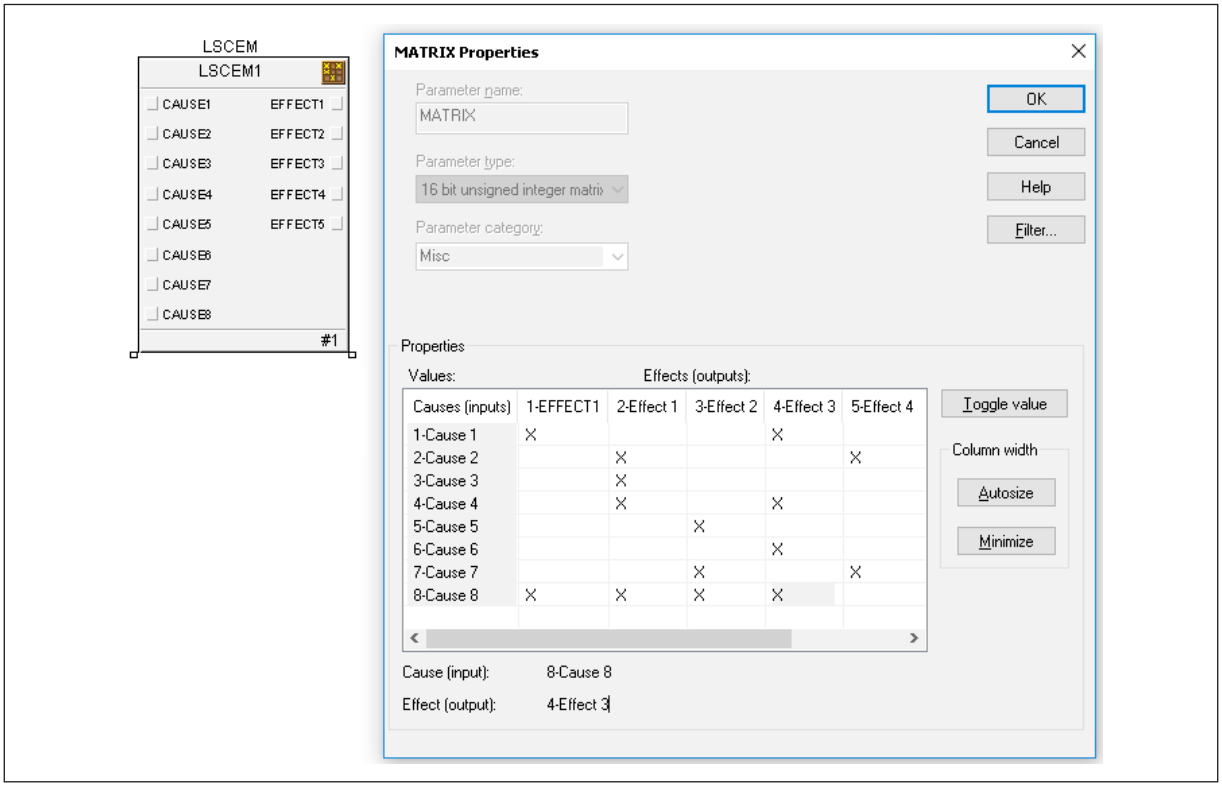

Figure 3 — *CEM Block Matrix Parameter*

While a 16x16 matrix might seem relatively small, the reality is that DeltaV SIS breaks the configuration in SIFs and the need for large matrices is greatly reduced. While the overall project CEM might include hundreds of causes and hundreds of effects, individual SIFs typically does not have more than 16 causes or 16 effects. Most of the SIFs can be implemented using the CEM function block. For the few SIFs requiring larger matrices, DeltaV v14 introduced two new function blocks (MONITOR function block and EFFECT function block). There is no set limit for the number of for causes or effects that can be implemented combining the new MONITOR and EFFECTs blocks.

SIF Configuration

Implementing a SIF in DeltaV SIS is quite simple:

1. Drag and drop the proper function blocks
2. Wired the function blocks as appropriate
3. Configure the proper parameters

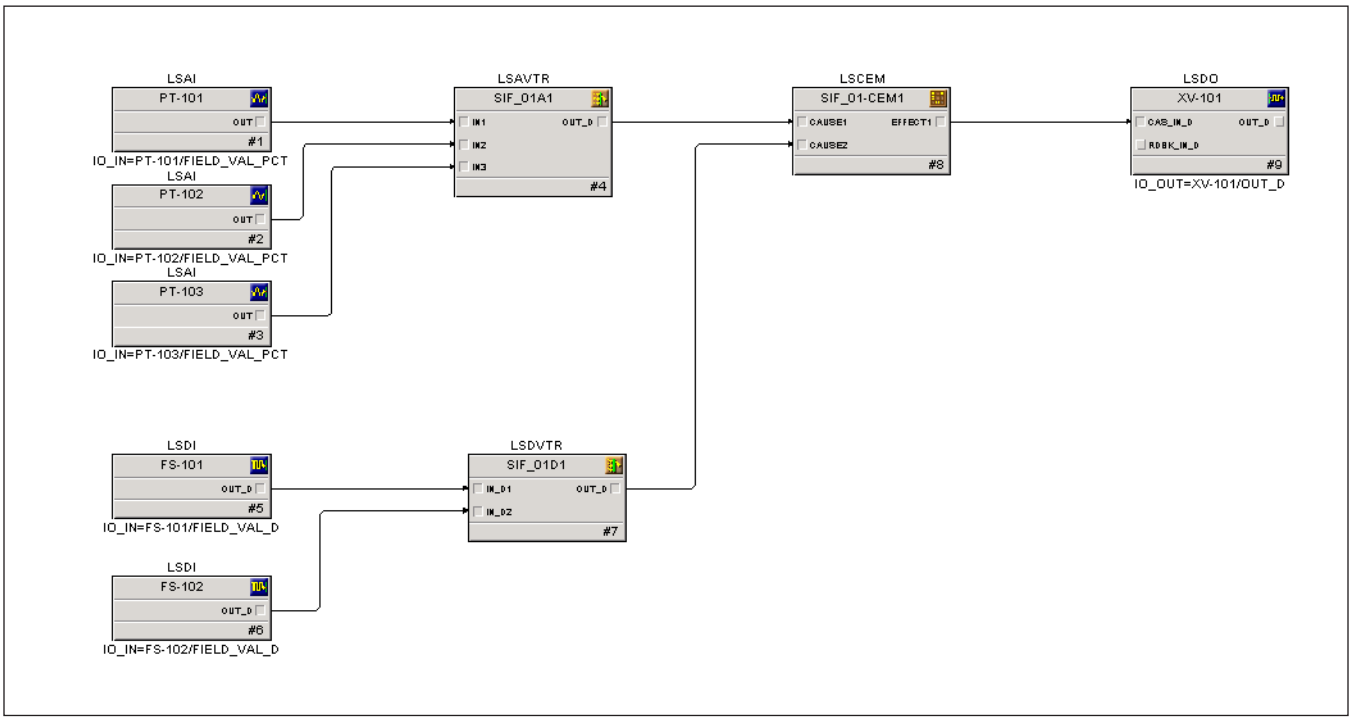

Figure 4 — *Example of a SIF in DeltaV*

exSILentia Overview

exSILentia is an integrated suite of engineering software tools designed to support the Process Safety Management (PSM) work process and the SIS Functional Safety Lifecycle. Data is seamlessly exchanged between the different phases of the safety lifecycle ensuring efficiency and consistency. Information from the PHA, LOPA and SIL target selection are feed directly into the SRS. Once the SIFs and associated risk reduction requirements are defined, exSILentia SILver™ facilitates the calculation of the achieved risk reduction for each SIF. Then, exSILentia enables the creation of a SRS that incorporate all the analysis done in the risk assessment.

The IEC 61511 Standard requires the creation of a SRS and defines what the SRS should contain. exSILentia facilitates the compliance to IEC61511 requirements. A proper SRS must contain all the requirements for the SIS and its associated SIFs. For each SIF, the SRS should include safe state, required SIL, maintenance and a startup overrides, architecture requirements, voting arrangements, trip delays, among other SIF requirements.

Emerson and exida Collaboration

Emerson collaborated with exida to create a new approach for SIS configuration. By pairing built-in DeltaV SIS functionality with exida's comprehensive software tools, user can develop safety logic configuration much faster and in fewer steps.

In a traditional SIS configuration approach, the project team use the SRS, along with a custom-built CEM as the basis for the safety logic configuration. The SRS and CEM are manually interpreted and translated into the safety logic. This configuration model requires multiple stages of data entry and present opportunities for human error. The new configuration approach powered by exSILentia leverages data structures created during the conceptual design to automatically generate safety logic.

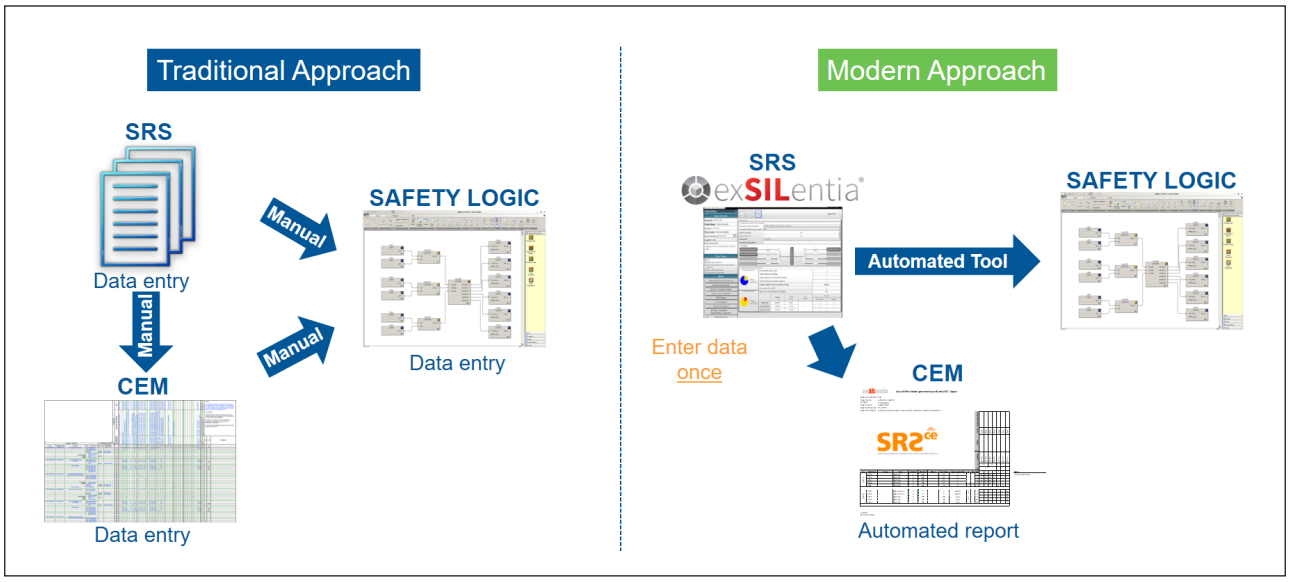

Figure 5 — *SIS configuration with exSILentia*

DeltaV SIS configuration using exSILentia

The DeltaV SIS Configurator created by exSILentia converts the exSILentia data into a DeltaV SIS configuration file (fhx file) that can be imported to create safety logic. The exSILentia approach for DeltaV SIS configuration leverages the SIL calculations and SRS captured in exSILentia, as well as the parallel structures between exSILentia and DeltaV SIS modules. Both exSILentia as DeltaV SIS follows a SIF approach that enables the overall SIS configuration to be divided in modular elements where a DeltaV SIS module contains one or more SIFs.

Each SIF contains a combination of sensors, voting arrangements, logic solver and final elements. Those elements defined in exSILentia are mapped to DeltaV SIS function blocks.

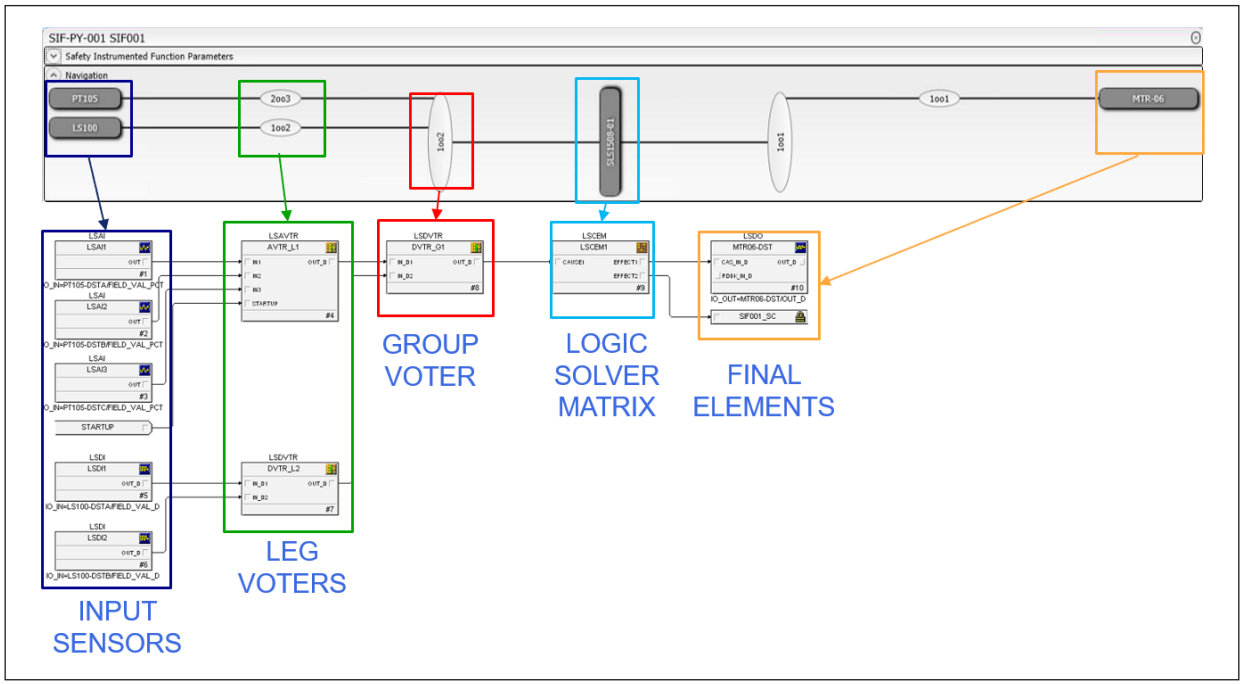

Figure 6 — *Parallel Structures between exida and DeltaV SIS*

Creating SIS Modules from exSILentia

exSILentia defines DeltaV SIS function blocks and the appropriate connections based on the SIL calculation diagram (in SILVer). exSILentia also parameterizes the DeltaV SIS function blocks based on the exSILentia data. Parameters such as I/O tag, trip limits, ranges, engineering units, and trip direction are defined as part of SIF definition within exSILentia. All those exSILentia settings are properly mapped to parameters within the DeltaV SIS function block.

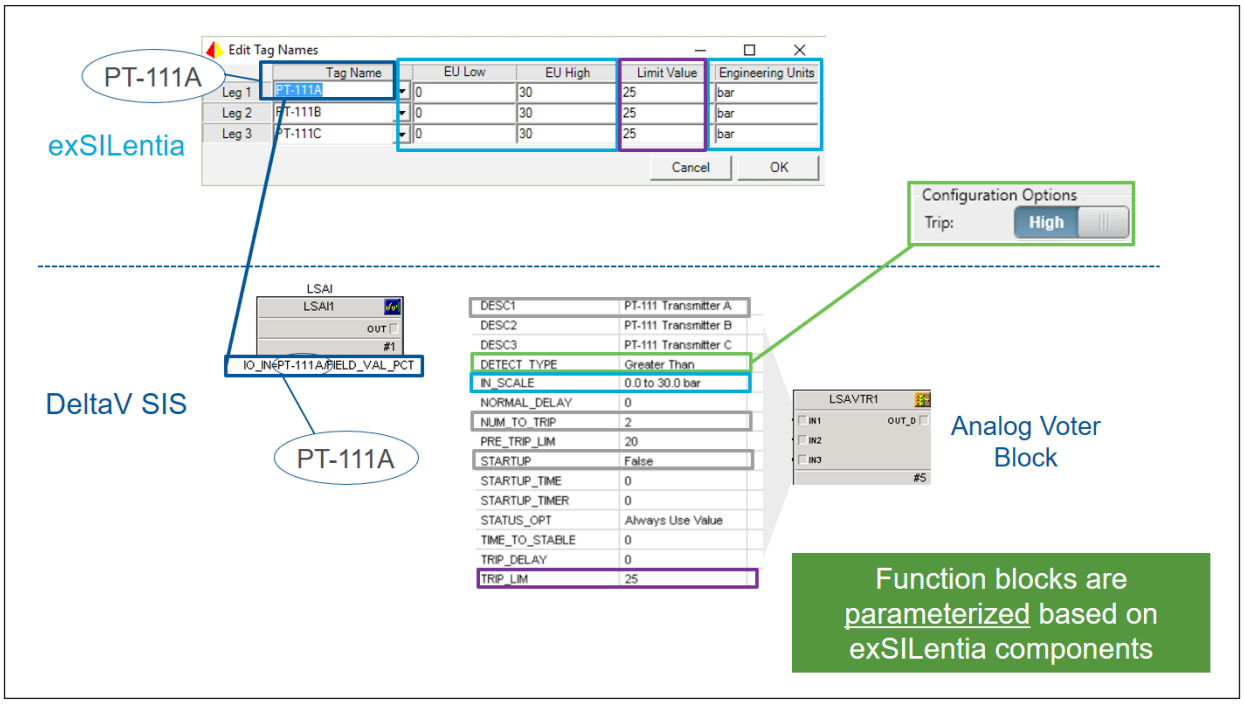

Figure 7 — *Function Block Parameterization*

There are also parallel structures for maintenance overrides. The maintenance overrides requirements in exSILentia are mapped to the DeltaV SIS bypass option parameter.

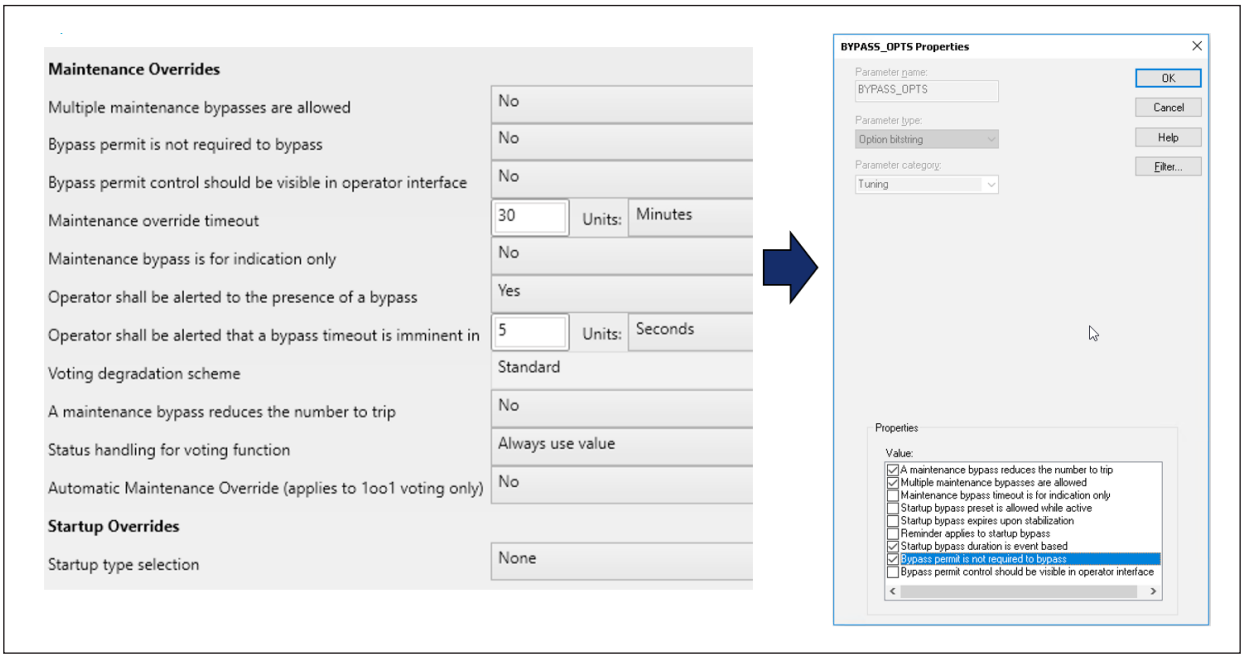

Figure 8 — *Maintenance Override in exSILentia and DeltaV SIS*

SIFs sharing final elements are automatically combined into the same DeltaV SIS module but user can also manually group SIFs into DeltaV SIS modules even if those SIFs are not sharing a final element.

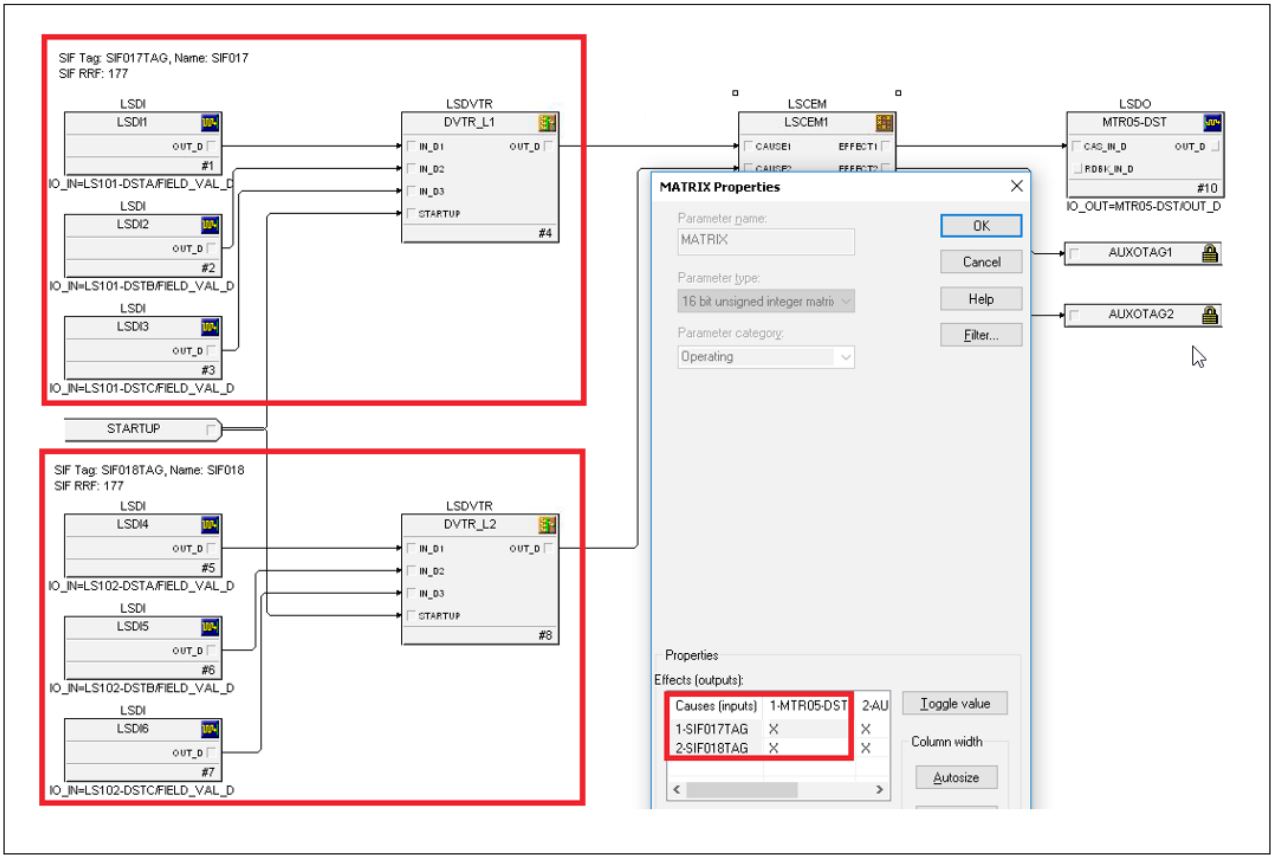

Figure 9 — *Automatically Grouping of SIF based on Final Elements.*

Annotations within SIS Modules

One key feature is related to the ability to add proper annotations within the safety logic. exSILentia automatically add relevant information that facilitate traceability to the SRS as well as increasing the readability of the logic.

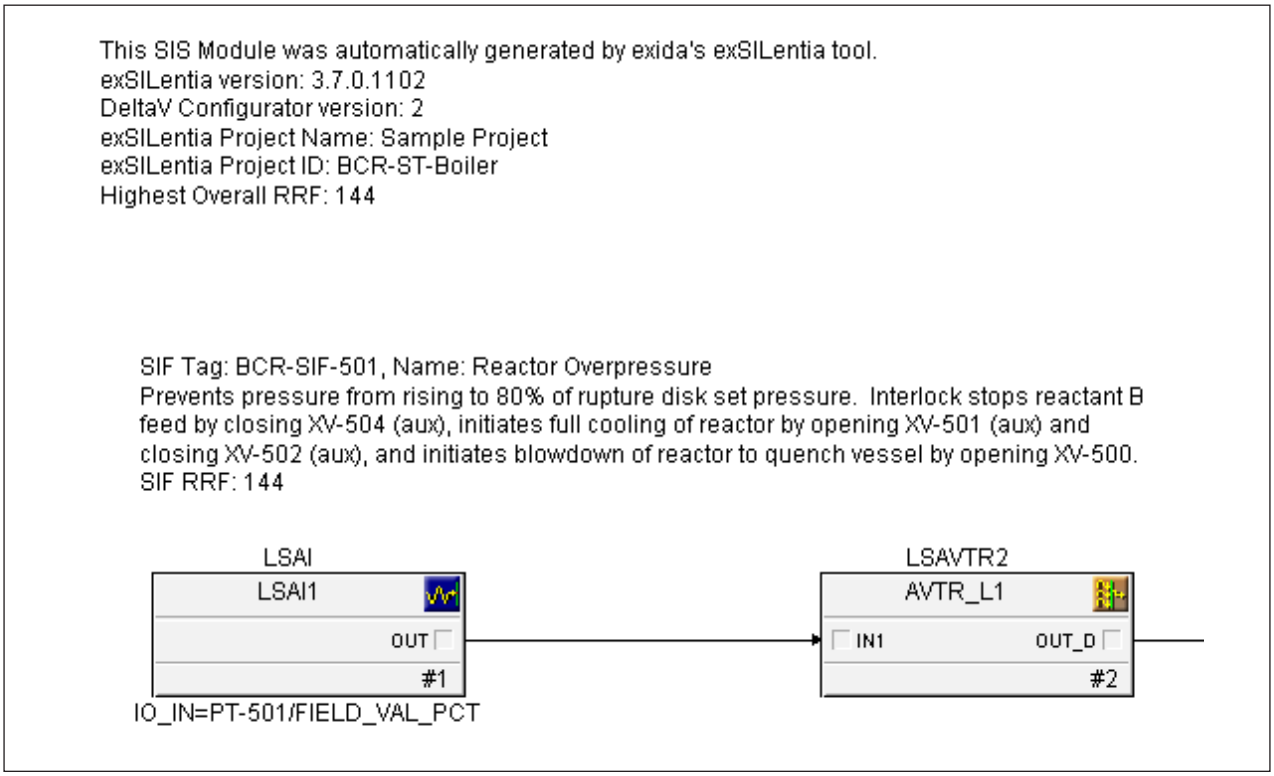

Figure 10 — *Annotation within DeltaV SIS modules created by exSILentia.*

Generation of DeltaV SIS Configuration

Once the SIF definition and the design SRS are completed, the DeltaV SIS configuration can be generated using the SRS C&E menu in exSILentia.

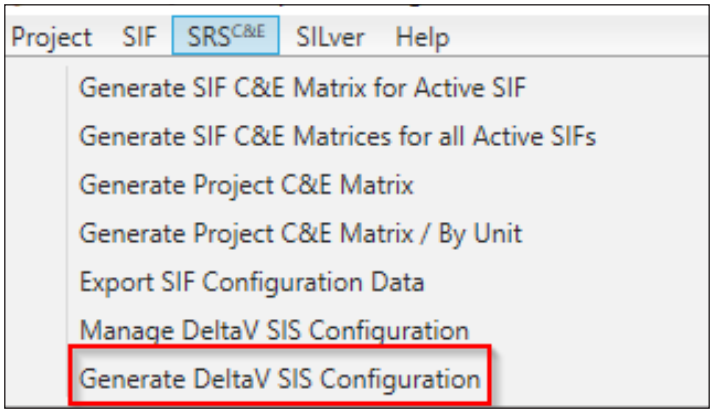

Figure 11 – Generation of DeltaV SIS Configuration

The user can choose to either generate the configuration for all SIFs or only selected SIFs. The option for selected SIFs is useful for update a SIF after late changes in the conceptual design.

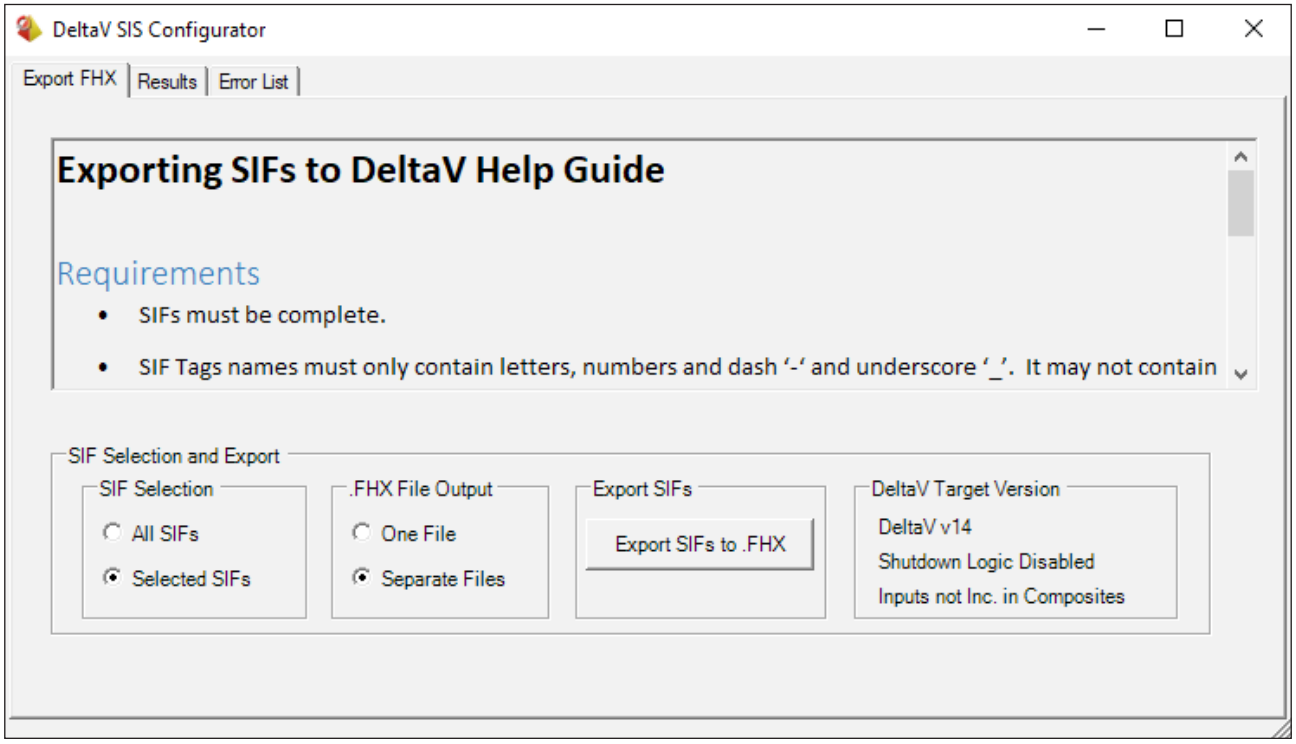

Figure 12 — *Generation of DeltaV SIS Configuration*

Workflow for using exSILentia to generate DeltaV Configuration

For End Users and EPCs that already use or are intending to use the exSILentia tool, the DeltaV SIS Configurator plug-in enables the ability to automatically generate safety logic to be imported into the DeltaV SIS database.

The use of exSILentia within a DeltaV SIS project is described by the following short list of activities.

1. SIF modelling. Once the SIFs are deemed required per the Analysis Phase, the SIF architecture is defined to meet the SIL requirement. exSILentia is used to specify and model the SIFs.
2. SIF Detailing. After the SIF architecture is defined, the user need to detail the SIF within exSILentia. At this point, the user sets variable ranges, trip limits, etc.
3. Data Transfer. The exSILentia configuration file is now sent to the project team performing the DeltaV SIS configuration. In this approach, the exSILentia configuration file replaces the Cause and Effect diagram and other information that is typically sent to project teams.
4. SIF Detailing for DeltaV SIS. The project team will load the exSILentia database and will work on design details the End User or EPC not necessarily needs to care about during the Analysis Phase, but are important for the configuration. This includes, for example, logic solver names associated with each SIF and defining the grouping of multiple SIFs into one single SIS module. The information is limited to safety logic and excludes graphics and I/O configuration beyond I/O references within I/O function blocks.

5. Configuration Generation. The project team will use exSILentia DeltaV Configurator to generate DeltaV SIS Logic (FHX file(s))
6. DeltaV Import. The project team will import the generated FHX file and finalize all the DeltaV configuration not supported by exSILentia (i.e. alarms, IO allocation and IO binding, auxiliary actions implemented in BPCS, CHARM configuration, etc.)
7. Finalizing DeltaV Configuration. Project team will configure HMI graphics and verify logic implementation together with HMI
8. Validation of SIF logic. User will validate the SIF logic to verify proper connections and safety functionality for each SIF

Applicability of DeltaV SIS Configurator

The ability to automatically generate DeltaV SIS safety logic is based on a sound conceptual design using exSILentia. An incomplete SRS will not generate the expected results. DeltaV relevant information (e.g. signal tags) needs to conform with DeltaV syntax. The DeltaV SIS Configurator tool generate proper warning messages when the DeltaV SIS syntax rules are not being followed and in most cases the log file provides sufficient indication for resolving the issue. The use of the DeltaV SIS Configurator does not eliminate proper engineering practices, in fact, a more structured approach is needed. Configuration implementation should not start until the design SRS is finalized and all relevant information for DeltaV SIS is properly captured. Only users that have used exSILentia as both a SRS compilation tool and SIL calculation tool will benefit of this solution. There are no migration tools to take a SRS or SIL calculations developed in other software/tools and convert it into exSILentia. The exSILentia approach requires early engagement during the conceptual design.

The use of the DeltaV SIS Configurator is mainly targeted to ESD applications. It is estimated that the tool would be able to create up to 90% of the safety logic but it would depend of the complexity of the application. Currently there is no support for sequential type of safety logic (e.g. BMS) and applicability to Fire and Gas is greatly impacted by the lack of SIL calculations in these applications.

The exSILentia tool focuses on the generation of tag-based safety logic. The I/O configuration including CSLS configuration, CHARM configuration and device allocation is not part of the scope of the tool. The user can use DeltaV SIS late binding capability to easily bind the tag-based configuration with the I/O design developed independently from SIF design. DeltaV Smart Commissioning is supported on configuration safety logic created by exSILentia. Graphics are not automatically generated either. Only a few alarms are configured by the tool, most SIS alarms must be configured manually or generated from an alarm rationalization software.

Conclusions

The DeltaV SIS Configurator tools provide great benefits for the implementation of ESD projects within DeltaV SIS. While the exSILentia tool does not generate the full DeltaV SIS configuration, it can potentially generate up to 90% of the safety logic. The user still need to create I/O configuration, graphics and alarms. In addition of the time savings, another benefit is a consistent approach with less error and that is easily traceable back to the SRS.

DeltaV SIS Configuration Using exSILentia

White Paper
June 2018

Emerson
North America, Latin America:
+1 800 833 8314 or
+1 512 832 3774

Asia Pacific:
+65 6777 8211

Europe, Middle East:
+41 41 768 6111

www.emerson.com/deltav