SDTools
VIBRATION SOFTWARE & CONSULTING

44 rue Vergniaud, 75013 Paris FRANCE
Tel: (33) +1 44 24 63 71
info@sdtools.com

Licensing policy

24 February 2021

SDTools seeks to support industrial partners in solving test and analysis related problems. Our base libraries `SDT` and `FEMLink` are meant as of the shelf solutions for standard problems in vibration, but we also develop custom software solutions for use in industrial design processes. This document describes our licensing policies.

1 Off the shelf libraries, application modules, tokens for custom developments

Some problems are industry/company specific and in our consulting arrangements the customer fully owns all the software except for `SDT/base` functions.

Some problems, we identify as sufficiently generic that we invest time of our own to let us reuse groups of functions (called `tokens` in our license manager) in different ways for different customers. Identifying such functions and forcing our customers to acknowledge our prior work and full property on those elements by purchasing a license, is a necessity to let us work with a range of people while continuing to guarantee that we do not infringe on the intellectual property of others.

Pricing of those `tokens` follows two possible paths.

- Your application is something sufficiently **generic** that we have worked often enough on the subject to identify a set of tokens needed to fulfill a standard objective. We then define a `module` that will contain all tokens for the application. For example, the `SDT/squeal` module will contain the `conutil` and `nltgtmdl` token (to deal with the representation of contact), the `xfbuild` and `xfpole` token (to deal with parametric studies associated to tracking instabilities), … Modules are typically expensive since they reflect the knowhow of how SDTools deals with a class of consulting projects and we require maintenance since we are continuing development in the area and share this development through software updates and support.
- Your application is **really custom** and/or you are being very directive in what you want, but since we deal with lots of projects, we find that there is a strong overlap with an existing `token`. Adding purchase of this token to the consulting project you are having us do for you, gives access to our prior work but needs to contribute to our ability to continue investing and pay for the time spend supporting you. We sell access to tokens in various forms

- `License` access: you purchase a license to the token and can use the associated compiled MATLAB function (pcode) in any of your developments by checking out this license.
- `Deployed` access: as part of the project with you, SDTools delivers a MATLAB compiler executable that lets you run the token. The project typically addresses customization of the token to your own processes/projects. License access is still possible, but deployed access lets you use the token within the compiled application without any further check. Deployed access is cheaper than license access if not done too often. This is the option that makes sense for people who generate executables a few times a year or less or need to guarantee that a deployed application can continue being used without license restriction. Maintenance is then not included except by specific contract. Access to new releases typically requires a reinstatement fee to the current status of the token. SDTools archives the revision of SDT libraries at time of compilation to allow recompilation of an old release if you change your part of the code. This then does not require `token` upgrade.
- `Runtime` license is our most expensive option. This gives you the right to generate executable with deployed access. This only makes sense for design groups who routinely generate applications based on our libraries, typically to run parametric studies deployed on clusters.

2 Token/module map and pricing principles

Pricing of modules and tokens are similar but correspond to different needs: fully open access by a few, restricted access by many … The table below gives a map of the current tokens and how they are used in different modules.

Maintenance is at 20 % per year. Rental for one year is 40% of list price.

Tokens can be deployed in MATLAB compiled executables. If done by SDTools, this gives unlimited access to the token within the deployed application (we issue a permanent unlimited license for that specific executable). The price is the same than that of a license, but the restriction comes from the fact that you cannot use it outside a specific deployed application. Since you do not get support, no maintenance is due unless you want to reinstate your deployed license and ask us for a revision of your executable.

Tokens can also be purchased through a `Runtime` license. This gives you the ability to generate your own executables with the same conditions as deployed tokens. Pricing is then to be discussed but typically 5 times the price of the included tokens.

Token/ Module	k€	FEMLink	Contact	Nlsim	CMT	ZParam	Visc	Squeal	HBM	Rotor	JobH
fem	1.5	x						x			
conutil	2		x	x				x			
nltgtmdl	2		x	x				x	x		
nlknkt	1			x				x	x		
nlmodal	5			x							
nlstatic	2			x							
nltraj	1			x				x	x		
femat	5						x				
viscrange	5						x				
viscmat	5						x				
xfbuild	2			x		x	x	x			
xfpole	1				x	x	x	x			
xfplot	1				x	x	x	x			
xffrfzr	1					x	x				
cmtlist	10				x						
cmtbuild	10				X						
cmtpar	3				X						
hbmui	10								x		
hbmcore	10								x		
rotor	10									X	
sdtjob	3										X
batch	2										X
Price k€		1.5	4	15	25	5	20	11.5	24	10	5

3 Token description

BASE

The base token provides access to main SDT library. This supports FEM modeling (OpenFEM library), model reduction, experimental modal analysis, all the tools for the GUI API of SDT, the DOE manager, … www.sdtools.com/pdf/sdt.pdf Piezoelectric modeling capabilities are described in a specific manual www.sdtools.com/pdf/piezo.pdf

FEM

The fem token provides access to the FEMLink library. It gathers input/output functions to other commercial codes as well as specific handling. It supports NASTRAN, ABAQUS, ANSYS. As well as an always evolving partial support of SAMCEF, MSC.Marc, PERMAS, Universal files, Excite, … These are documented in the main SDT manual.

NLTGTMDL

The nltgtmdl token provides model linearization and associated assembly capabilities. Non-linearities are transformed into superelements that can be added in the base model in various ways. Parametrization is handled.

NLKNKT

The `nlknkt` token implements pre- and post- for surface based elastic connections analysis. This methodology estimates frequency/damping sensitivity to normal and tangential coupling stiffness densities.

NLMODAL

The `nlmodal` token handles non-linear time domain simulations for reduced models. In particular it implements non-linearity reduction and the fully compiled `modalNewmark` and explicit `expNewmark` time integration solver. It includes restart and stepped experiment capabilities.

Key functions are `nl_solve`, `mkl_utils`, documentation www.sdtools.com/helphbm (also contains documentation of the `hbmcore` token).

NLSTATIC

The `nlstatic` token provides methods to evaluate and analyze static states of models with non-linearities.

NLTRAJ

The `nltraj` token provides methods to analyze deformation trajectories applied to a model with non-linearities. This includes post-treatment handling of non-linear transient simulations.

FEMAT

The `femat` token implements advanced material handling in SDT. The `matsplit` functionality that allows several strategies to decompose a material into spatial contributions for further studies constitutes the core of this token. Functionalities associated to the post-treatment of energy distribution by constitutive law coefficients is also included.

VISCRANGE

The `viscrange` token implements parametric studies with procedures adapted to viscoelasticity. Performance maps, temperature influence, material selection GUI, …

VISCMAT

The `viscmat` token implements viscoelastic material modelling in SDT. This includes advanced meshing utilities to implement damping devices solutions in a model and material implementation. Implementation includes nomogram handling, visualization, use in frequency domain parametric solvers, exports to NASTRAN.

XFBUILD

The `xfbuild` token implements multi-model reduction strategies in SDT. From a parametered model the objective is to provide either a reduction basis or a fully reduced model that accounts for declared parameters variations and structural effects.

XFPOLE

The xfpole token handles modal analyses parametric studies. It comprises the optimized generation of the DoE results and a series of post-treatments oriented towards modeshape variations, including pole tracking and pole clustering techniques. Pole tracking aims at globally identifying a series of parametered modes by clustering by distance in an adapted metric based on the modeshape from a nominal point. A second local clustering technique aims at segregating modes by modeshape proximity in a given frequency/damping domain. Parametric regression/distribution analysis is possible per cluster.

XFPLOT

The xfplot token implements interactive displays associated to the results of a parametric study. This includes a generation GUI, customizable labels and links to modeshape restitution.

XFFRFZR

The xffrfzr token handles direct response analyses parametric studies.

CMTLIST

The cmtlist token implements pre-processing for the CMT method and utilities to handle non-reduced mechanical assemblies. Tools to generate and control component listing, regularize constraints, assess coupling connections and potential kinematic chains are provided and interfaced. Utilities to remove, add or replace components are also present.

CMTBUILD

The cmtbuild token implements the CMT reduction method and provides interfaced post-treatment for component contribution analysis to assembled deformation shapes.

CMTPAR

The cmtpar token implements pre and post for parametric studies associated to a CMT model.

HBMCORE

The hbmcore token implements harmonic balance resolution. It handles optimized non-linearity representation through AFT. All script/function based pre and post-processing steps are included but not any visual aspect which is included in hbmui.

HBMUI

The hbmui token provides the integration of hbmcore utilities into a series of integrated simulation and analyses procedures.

ROTOR

The rotor module implements methods associated with rotating machinery modeling using full beam/volume models (gyroscopic effects, Campbell, transients with bearing non-linearities) and cyclic symmetry support (including multi-stage modeling, model reduction allowing mistuning studies, damping, integration with visc capabilities).

SDTJOB

The sdtjob token implements external job handling in SDT. It includes the generation of calls and monitoring for external software, local or remote, safe copy. In the latter case file transferring is also handled. An integrated job handler allows a study batch mode that can run as a server to consume jobs saved to a monitored directory.

BATCH

The batch token provides job execution deployment in runtime based batch mode for SDT scripts.

4 SDT modules

This section aims at presenting in details the aforementioned modules.

4.1 SDT / CMT

This module provides a framework to efficiently handle mechanical assemblies in SDT. A mechanical assembly is here understood as a series of components connected by linear constraints (TIE, …) or possibly non-linear laws (contact, …). Its base application is the integration of the Component Mode Tuning method, but with large pre and post processing capabilities to perform complete studies from a generic FE model.

- Pre-processing
  - Model linearization (boundary conditions and tangent states from non-linearities)
  - Identification of kinematic coupling chains (series of 1D elements at component interfaces)
  - Component list generation strategies through selection, recursive selection, completion, integration of unselected areas.
  - Component-wise reduction framework accepting ad hoc strategies (multi-modelling, contact based enhancement like KnKt, static capability, …)
  - Component verification tools, independence and rigid body mode consistency checks.
  - Component interaction verification tool, localization kinematics and flexibility evaluation.
  - Component replacement integration, with interface coupling regeneration, support of reduced or full models.
  - Integration of component connection studies, connection parametrization and pre-processing for KnKt studies.
  - GUI implementation for study setup, and interactive component selection
- Solver
  - Generation of a self-contained reduced SDT model under CMT framework. Integration of HDF based saving and handles (Out-Of-Core data) to optimize memory.
  - Parametric studies framework, design of experiment, second layer reduction strategies (when reduced component models account for many parameters), batch execution.
  - Generation of KnKt studies for advanced cases

- Post-processing
  - Model visualization with deformation restitution in `feplot`. Component based display selection possibly through GUI
  - Component mode and coupling sensitivity post-treatment with docked interactivity and automated report generation capability (selection possible through GUI).
  - Energy distribution visualization and compatible with different selections.

4.2 SDT/ZPARAM

This module implements the generation, simulation and post-treatment of reduced parametered multi-models. It is currently integrated in `SDT/Visco` and `SDT/CMT`, but identified separately at SDTools because it may be useful for other applications.

- Pre-processing:
  - Reduction basis design
  - Parameter and design of experiment definition
  - Latin Hypercube Sampling strategy
- Solver:
  - Reduction basis generation with Out-Of-Core support
  - Model projection
  - Optimized reduced reanalysis for experiments
  - Genetic algorithm implementation
- Post-processing:
  - Interactive results visualization
  - Mode tracking/clustering
  - Data based analysis and parametric regressions

4.3 SDT/Contact

This module provides support for contact handling in SDT. Its main objectives are

- `FEMLink` support to contact formulations
- The generation of contact observation matrices for customized exploitation
- The generation of tangent contact and friction coupling
- Contact data visualization