

RADIOSS User's Code Interface

2017 version – January 2017

Introduction



Altair Engineering, Inc., World Headquarters: 1820 E. Big Beaver Rd., Troy MI 48083-2031 USA
Phone: +1.248.614.2400 • Fax: +1.248.614.2411 • www.altair.com • info@altair.com

INTRODUCTION

This manual describes the interface between RADIOSS and a user's subroutines.

You can define private code of various categories, including:

- User window (general interface for external code)
- User material laws for solids and shells
- User spring elements
- User solid elements (may be degenerated to shell or beam elements)
- User sensors
- User failure models

To link RADIOSS to a user module of each category, two subroutines must be provided: one linked with RADIOSS Starter and one with RADIOSS Engine. The Starter subroutines are only used to read user data and initialize specified parameters. The Engine subroutines define and execute user programs. Storage arrays are provided for each type of application. The communication between RADIOSS and the User's subroutines is done by passing function arguments by specialized functions. The function arguments are used for general purpose user window interface, user materials and elements. Only integer and float variables or arrays are used. The float arguments are Double Precision.

Otherwise, all user subroutines use special access functions to retrieve information from the RADIOSS database.

The next chapters will describe the standardized headers for each user subroutine, the access functions, and link procedures. Example program codes for each category are also provided.

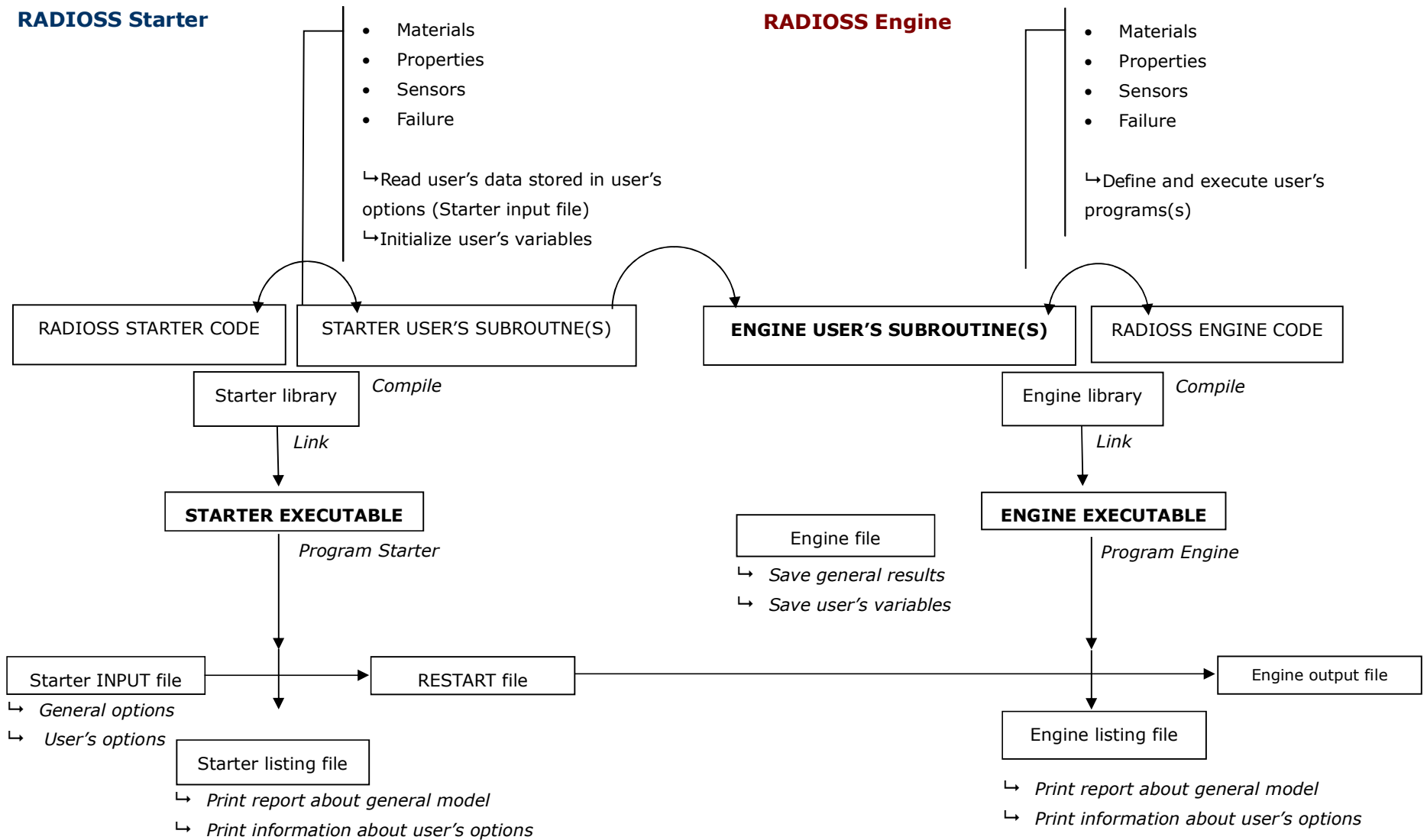
Table 1: RADIOSS Starter and Engine Subroutines and Executables

($nn = 29, 30$ or 31 ; $n = 1, 2$ or 3 ; $m = 1, 2, 3, \dots$; $i = 1$ to 18 ; $j = 1$ to 99 ; $k = 1$ to 99), $ll = 04, 05$ or 06

		RADIOSS Starter	RADIOSS Engine
		User's Input Options	Engine File Options
User material		/MAT/USER n	/ANIM/keyword/USR i /ANIM/keyword/USR j / k
User property		/PROP/USER n	
User sensor		/SENSOR/USER m	
User failure		/FAIL/USER n	
User's Subroutines			
User window		USERWIS.f	USERWI.f
User material laws 29, 30, 31	Shell	LECM nn .f	SIGEPS nn C.f
	Solid	LECM nn .f	SIGEPS nn .f
User property	Spring	LECG nn .f and RINI nn .f	RUSER nn .f
	Solid	LECG nn .f and SINI nn .f	SUSER nn .f
User failure 01, 02, 04	Shell	LECRII.f	fillawC.f
	Solid	LECRII.f	fillaw.f
User sensor		LECSEN_USER m .f	USER_SEN m .f
Object Files			
User window		USERWIS.o	USERWI.o
User material laws 29, 30, 31	Shell		LECM nn .o
	Solid		LECM nn .o
User spring	Spring		LECG nn .o and RINI nn .o
	Solid		LECG nn .o and SINI nn .o
User failure 01, 02, 04	Shell		LECRII.o
	Solid		LECRII.o
User sensor		LECSEN_USER m .o	USER_SEN m .o
Object library			
		"RADIOSS_engine_library".a	"RADIOSS_starter_library".a
User's Executables			
		"user_engine_executable"	"user_starter_executable"

RADIOSS Starter

RADIOSS Engine



Saving User's Variables in Animation File

The user's variables UVAR (*element_number,i*) defined in the user's material law can be saved in an animation file using the output USERi as keyword3 in the Engine option /ANIM/keyword2/keyword3.

The number of saved variables is limited to 18 ($1 \leq i \leq 18$). Results are saved for each integration point.

Example: The user variable number 1 UVAR (*element number,1*) is saved in animations by the Engine option /ANIM/ELEM/USER1.

The number of user's variables is set in NUVAR (Starter argument.)