	Туре	Name	Value range	Default	Brief description	Long description
onfig						
_	AeroModel::_ (enum)	aeromodel	See btSoftBody::eAeroModel	eAeroModel::V_Point	Aerodynamic model	Define what kind of feature is used to compute aerodynamic forces (specifies orientation of vertex or face normals for aeurodynamics).
		kVCF	2		Velocities correction factor (Baumgarte)	Define the amount of correction per time step for drift solver (sometimes referred to as ERP in rigid bodies solvers).
	tScalar	kDP	[0, 1]		Damping coefficient	(Velocity?) damping.
	tScalar	kDG	[0, +inf]		0 Drag coefficient	For aerodynamics computations. See wikipedia "Drag coefficient". 0 = no drag.
	tScalar	kLF	[0, +inf]		0 Lift coefficient	For aerodynamics computations. See wikipedia "Lift (force)". 0 = no lift.
	tScalar	kPR	[-inf, +inf]		Pressure coefficient	Affects aerodynamics computations. Also, when setPose(true,)*** has been called, defines pressure used to conserve volume. (?)
	tScalar	kVC	[0, +inf]		Volume conservation coefficient	Volume conservation. Also, when setPose(true,)*** has been called, defines magnitude of the force used to conserve volume. (?)
	tScalar	kDF	[0, 1]		0.2 Dynamic friction coefficient	Same as rigid body friction. 0 = slides, 1 = sticks.
	tScalar	kMT	[0, 1]		Pose matching coefficient	When setPose(, true)*** has been called, defines the factor used for pose matching. (enforcing relative vertex positions)
		kCHR	[0, 1]		1 Rigid contacts hardness	Defines how "soft" contact with rigid bodies are. 0 = no penetration correction, 1 = full correction.
		kKHR	[0, 1]		0.1 Kinetic contacts hardness	Defines how "soft" contact with kinetic/static bodies are. 0 = no penetration correction, 1 = full correction.
		kSHR	[0, 1]		1 Soft contacts hardness	Defines how "soft" contact with other soft bodies are. 0 = no penetration correction, 1 = full correction.
		kAHR	[0, 1]		0.7 Anchors hardness	Defines how "soft" anchor constraints (joints) are. 0 = no drift correction, 1 = full correction.
		kSRHR CL	[0, 1]	_	0.1 Soft vs rigid hardness	Used with clusters only. Presumably similar function as kCHR.
		kSKHR_CL	[0, 1]		1 Soft vs kinetic hardness	Used with clusters only. Presumably similar function as kKHR.
		kSSHR_CL	[0, 1]		0.5 Soft vs soft hardness	Used with clusters only. Presumably similar function as kSHR.
		kSR_SPLT_CL			0.5 Soft vs rigid impulse split	Used with clusters only. What proportion to split impulse with a rigid body after collision.
		kSK_SPLT_CL			0.5 Soft vs kinetic impulse split	Used with clusters only. What proportion to split impulse with a kinetic/static body after collision.
		kSS_SPLT_CL			0.5 Soft vs soft impulse split	Used with clusters only. What proportion to split impulse with another soft body after collision.
		maxvolume	[0, +inf] (?)		1 Maximum volume ratio for pose	?
	tScalar	timescale	[0, +inf] (?)		1 Time scale	Factor of time step. Can be used to speed up or slow down simulation of a specific soft body
in	nt	viterations	[0, +inf] (?)		0 Velocities solver iterations	Number of iterations for velocities solvers (if any).
in	nt	piterations	[0, +inf] (?)		1 Positions solver iterations	Number of iterations for position solvers (if any).
in	nt	diterations	[0, +inf] (?)		0 Drift solver iterations	Number of iterations for drift solvers (if any).
in	nt	citerations	[0, +inf] (?)		4 Cluster solver iterations	Number of iterations for cluster solvers (if any).
in	nt	collisions	See btSoftBody::fCollision	fCollision::Default	Collisions flags	See btSoftBody::fCollision
t٧	/SolverArray	m_vsequence	n/a	-	Velocity solvers sequence	Defines order and type of solvers to apply for velocities. See btSoftBody::eVSolver.
tF	PSolverArray	m_psequence	n/a	-	Position solvers sequence	Defines order and type of solvers to apply for positions. See btSoftBody::ePSolver.
tF	PSolverArray	m_dsequence	n/a	-	Drift solvers sequence	Defines order and type of solvers to apply for drift. See btSoftBody::ePSolver.
	·	·				
aterial						
		m_kLST	[0, 1]		1 Linear stiffness coefficient	
		m_kAST	[0, 1]		1 Angular stiffness coefficient	
b	tScalar	m_kVST	[0, 1]		1 Volume stiffness coefficient	
in	nt	m_flags	See btSoftBody::fMaterial	fMaterial::Default		
						*** at Daga (by all mag)
						***setPose(bvolume,bframe):
						- setPose take the current state of the body (usually after creation, scaling, setting mass, etc), and do two things depending of the value (true/false) of 'bvolume' and 'bframe'. - bvolume set at true mean that you enable kVC and/or kPR based on the current volume of the body. keep in mind that volume mean closed, so check your meshes.
						- bframe set at true mean the you enable kMT (shape matching), based on the current body shape (current orientation/translation doesn't matter, just node positions relative to each others).
						- you need to call setPose(,true) only when you want to use shape matching (kMT), and setPose(true,) when you want to use pressure forces (kPR) and/or volume conservation forces (kVC).