

CFturbo2ICEM

Automatische Geometrie- und Gitter-Generierung
in ANSYS ICEM CFD
aus CFturbo-Geometrien

Version 4
Oktober 2010

CFturbo

© CFturbo GmbH
Dresden, Germany

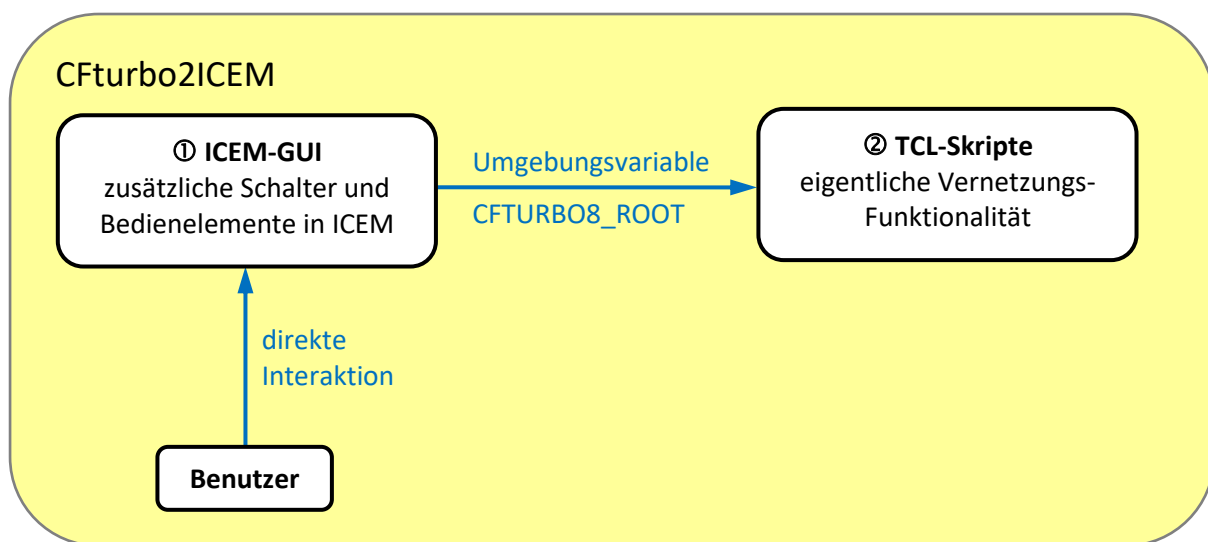
www.cfturbo.com

**Verwendete Dateien und
deren Position im Dateisystem
(manuelle Installation)**

Unter **Windows** erfolgt die vollständige Installation von CFturbo2ICEM mit dem Installationsprogramm „CFturbo2Icem*.exe“. Voraussetzung ist ein installiertes ICEM und ein installiertes CFturbo. In diesem Fall ist keinerlei manueller Aufwand notwendig.

Unter **Linux** ist CFturbo nicht lauffähig, CFturbo2ICEM kann jedoch genutzt werden. Derzeit existiert dafür jedoch kein Installationsprogramm, die einzelnen Dateien müssen manuell anhand dieser Anleitung installiert (kopiert) werden. Unter Linux muss generell auf exakte Groß- und Kleinschreibung geachtet werden.

Prinzipielles Schema für die Nutzung von CFturbo2ICEM



Es gibt zwei Speicherorte für Dateien:

- ① Dateien, die die Bedienung der Skripte über die ICEM-GUI ermöglichen
- ② TCL-Skripte selbst

Die Verbindung zwischen ICEM und den TCL-Skripten wird über die Umgebungsvariable CFTURBO8_ROOT gesteuert. Der Wert dieser Variable enthält den Pfad, in dem die Skripte gespeichert sind.

Betriebssystem	Windows	Linux
Umgebungsvariable	%CFTURBO8_ROOT%	\$CFTURBO8_ROOT
Wertzuweisung	automatisch durch Installations-Programm	je nach verwendeter Shell, z.B. export (bash), setenv (tcsh)

Auf Windows-Systemen, auf denen CFturbo installiert wurde, wird die Umgebungsvariable CFTURBO8_ROOT bereits bei der CFturbo-Installation angelegt und zeigt auf das CFturbo-Installationsverzeichnis, standardmäßig „C:\Programme\CFturbo 8“. Auf Linux-Systemen muss die Umgebungsvariable CFTURBO8_ROOT vor dem Start von ICEM gesetzt werden.

① ICEM-GUI

Diese Dateien sind einerseits graphische Dateien (Icons) und andererseits veränderte ICEM-Dateien zur Sichtbarmachung der CFturbo2ICEM-Buttons innerhalb der ICEM-GUI.

Die Zielverzeichnisse <ICEM-Root> und <ICEM-Sys> können folgendermaßen lokalisiert werden:

Betriebssystem	Windows	Linux
ICEM-Installationsverzeichnis <ICEM-Root>	%ICEMCFD_ROOT120% z.B. "C:\Programme\ANSYS Inc\v120\icemcfd"	Verzeichnis suchen mit „which icemcfd“ z.B. /usr/ansys_inc/<version>/icemcfd
ICEM-Systemverzeichnis <ICEM-Sys>	%ICEMCFD_SYSDIR% z.B. „win“	je nach Prozessortyp z.B. linux64_amd

② TCL-Skripte

Die eigentlichen Skriptdateien liegen unterhalb eines Verzeichnisses, das durch die Umgebungsvariable <CFTURBO8_ROOT> gekennzeichnet ist. In diesem Verzeichnis muss ein Ordner mit dem Namen „script“ angelegt werden, in den die eigentlichen Skriptdateien kopiert werden.

Beispiel:

Betriebssystem	Windows	Linux
Umgebungsvariable <CFTURBO8_ROOT>	C:\Daten\CFturbo2ICEM	/usr/cfturbo8
Ordner „script“	C:\Daten\CFturbo2ICEM\script	/usr/cfturbo8/script

Liste aller Dateien

	Quelle CFturbo2ICEM/	Dateiname	Ziel
ICEM-GUI	icons/	CFturbo.gif ImpGeo35.gif ImpMesh35.gif VolGeo35.gif VolMesh35.gif	<ICEM-Root>/<ICEM-Sys>/ lib/ai_env/icons
	resource <ICEM-Version>/	geometryDEZs.ttkrf GeometryTab.ttkrf meshDEZs.ttkrf MeshTab.ttkrf	<ICEM-Root>/<ICEM-Sys>/ lib/ai_env/resource
		cfturbo.tcl	<ICEM-Root>/<ICEM-Sys>/ lib/ai_env/scripts
TCL-Skripte		version.dat	<CFTURBO8_ROOT>/ script
	udf/	cft_arc.tcl cft_calculate_nlayers.tcl cft_coeff.tcl cft_constants.tcl cft_cre_n_pts_on_crv.tcl cft_cre_pnt_cyl_coord.tcl cft_delete_families.tcl cft_delta_time.tcl cft_list.tcl cft_list_create_from_polynomial.tcl cft_match_cyl_surface.tcl cft_obtain_cyl_coord.tcl cft_obtain_pos_on_crv_at_val.tcl cft_smpl_crv.tcl cft_trim_surface.tcl cft_var.tcl cft_vector.tcl cft_write_body.tcl cft_write_thincuts.tcl load_udf.tcl	<CFTURBO8_ROOT>/ script/udf
	impeller/geometry/	create_bodies.tcl create_families.tcl createCFTGeolmp_0.tcl createCFTGeolmp_1.tcl createCFTGeolmp_2.tcl define_variables.tcl finalize_geometry.tcl full_impeller.tcl generator_geo.tcl impeller_construction.tcl interpolate.tcl ogrid.tcl repair.tcl set_params.tcl set_thincuts.tcl write_geo_info.tcl	<CFTURBO8_ROOT>/ script/impeller/geometry

TCL-Skripte	impeller/meshing/	createCFtMeshImp_0.tcl createCFtMeshImp_1.tcl createCFtMeshImp_2.tcl define_variables.tcl export.tcl full_impeller.tcl group_elements.tcl reload_proj.tcl reorient_mesh.tcl set_color.tcl write_mesh_info.tcl	<CFTURBO8_ROOT>/ script/impeller/meshing
	impeller/meshing/ tetraprism/delaunay/	build_topo.tcl check_mesh.tcl main_delaunay.tcl prism.tcl smooth.tcl tetra.tcl	<CFTURBO8_ROOT>/ script/impeller/meshing/ tetraprism/delaunay
	impeller/meshing/ tetraprism/octree/	build_topo.tcl check_mesh.tcl main_octree.tcl prism.tcl smooth.tcl tetra.tcl	<CFTURBO8_ROOT>/ script/impeller/meshing/ tetraprism/octree
	volute/geometry/	create_bodies.tcl create_families.tcl define_variables.tcl density.tcl diffuser1.tcl finalize_geometry.tcl generator_geo.tcl main_geometry.tcl pipe_construction.tcl repair.tcl round_cutwater.tcl set_params.tcl set_thincuts.tcl trim_cutwater.tcl volute_construction.tcl write_geo_info.tcl	<CFTURBO8_ROOT>/ script/volute/geometry
	volute/meshing/	createCFtMeshVol_0.tcl define_variables.tcl export.tcl group_elements.tcl reload_proj.tcl reorient_mesh.tcl set_color.tcl write_mesh_info.tcl	<CFTURBO8_ROOT>/ script/volute/meshing
	volute/meshing/ tetraprism/delaunay/	build_topo.tcl check_mesh.tcl main_delaunay.tcl prism.tcl smooth.tcl tetra.tcl	<CFTURBO8_ROOT>/ script/volute/meshing/ tetraprism/delaunay
	volute/meshing/ tetraprism/octree/	build_topo.tcl check_mesh.tcl main_octree.tcl prism.tcl smooth.tcl tetra.tcl	<CFTURBO8_ROOT>/ script/volute/meshing/ tetraprism/octree