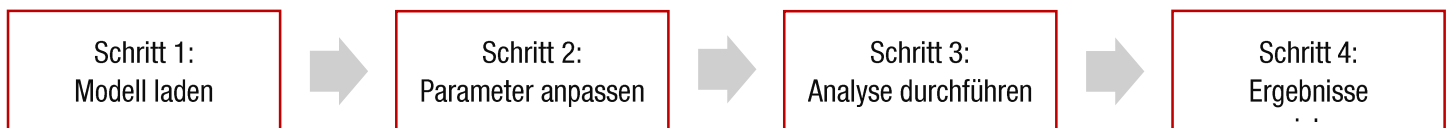


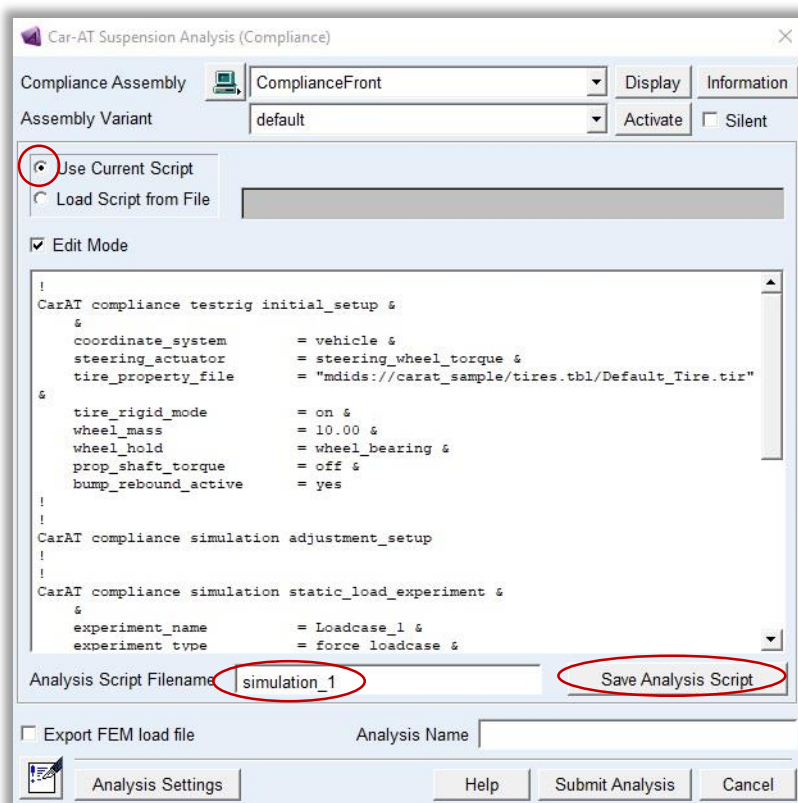
Adams/Car: Kleine Automatisierungen mittels Commands

Es muss nicht immer gleich eine DoE-Analyse sein, um die Vorzüge von Adam's Batch-Modus genießen zu können. Viele Anwender haben mit wiederkehrenden Simulationsaufgaben zu tun, die sich von Fall zu Fall nur leicht voneinander unterscheiden. Durch kleine Automatisierungen können wir uns die Arbeit erleichtern, sparen Zeit und stellen zudem sicher, stets demselben Berechnungsstandard zu folgen. In diesem Beispiel werden wir daher einen einfachen Workflow automatisieren, welcher aus folgenden 4 Schritten besteht:



Vorbereitung:

Für unsere Automatisierung mittels Commands benötigen wir neben einem geladenen Modell ein hierzu passendes Simulations-Skript. In diesem Beispiel wurde aus der CarAT-Beispiel-Datenbank die Assembly „ComplianceFront.asy“ geladen und anschließend eine „Static Loadcase“-Analyse durchgeführt, welche als Vorlage für unser Simulations-Skript dienen soll.



Über die Schaltflächen „Simulate → AT Suspension Analysis (Compliance) → Scripted Analysis“ kann durch Auswahl von „Use Current Script“ die gerade durchgeführte Analyse aus der Session geladen und mithilfe des Hakens „Edit Mode“ als Skript angezeigt werden.

Wir benennen dieses Skript als `simulation_1` und speichern es über „Save Analysis Script“ in unserer Datenbank unter `suspension_analysis_scripts.tbl` ab.

Das soeben generierte Simulations-Skript soll hinterher über ein Command-File ausgeführt werden. Hierfür erstellen wir uns zunächst ein Arbeitsverzeichnis und legen z.B. mit einem Text-Editor die Datei `run_1.cmd` an. Hierin werden die einzelnen Adams-Befehle unserer 4 Schritte definiert.

Schritt 1: Modell laden

Direkt nach dem Starten der Session soll die gewünschte Assembly eingelesen werden. Hierfür ist der entsprechende Pfad in Zeile 02 einzutragen.

```
01 acar files assembly open &
02   assembly_name = 'mdids://carat_sample/assemblies.tbl/ComplianceFront.asy' &
03   variant = default &
04   open_all = no
05
```

Schritt 2: Parameter anpassen

Um das Verhalten einer Assembly zu untersuchen, sollen meist bestimmte Parameter variiert und untersucht werden. Beispielhaft ist hier die Änderung einer Hardpoint-Koordinate gezeigt. In Zeile 07 muss hierfür die korrekte Parameter-Bezeichnung angegeben werden, welche sich aus Assembly-Namen, Subsystem-Namen, dem „ground“ und schließlich dem Hardpoint-Namen zusammensetzt. In diesem Beispiel wurde die X-Koordinate des linken Pendelstützen-Anbindungspunkts um von 30mm um 2mm nach vorne verschoben. Über die Option in Zeile 09 wurde zudem festgelegt, dass diese Änderung auch auf der rechten Seite vorgenommen wird.

```
06 car standard_interface hardpoint &
07   hardpoint = '.ComplianceFront.SuspensionFront.ground.hpl_Anbindung_Pendelstuetze
08 &
09   location = 28.0,-530.0,160.0 &
10   symmetric = yes
```

Schritt 3: Analyse durchführen

Hier findet nun das in der Vorbereitung erstellte Simulations-Skript Anwendung. Nachdem in Zeile 12 ein beliebiger Analyse-Name vergeben worden ist, muss in Zeile 14 der Modell-Name der geladenen Assembly eingetragen werden. Schließlich wird in Zeile 15 der Pfad zum abgespeicherten Simulations-Skript referenziert. Das *.sas-Format entspricht hier dem Standard von Adams/CarAT. Unter Adams/Car hingegen wird das *.lcf-Format verwendet.

```
11 carat compliance simulation submit &
12   analysis_name = 'variant_1' &
13   analysis_type = scripted &
14   model = ComplianceFront &
15   analysis_script =
16   'C:\private.cdb\suspension_analysis_scripts.tbl\simulation_1.sas'
```

Schritt 4: Ergebnisse speichern

Zum Abschluss sollen die Analyse-Ergebnisse im *.res-Format exportiert werden. In Zeile 18 ist hierzu zunächst der Analyse-Name anzugeben, welcher innerhalb des Simulations-Skripts als „experiment_name“ angegeben ist. In Zeile 19 kann wiederum ein beliebiger Dateiname für die Output-Datei vergeben werden. Diese soll später in das angelegte Arbeitsverzeichnis geschrieben werden.

```
17 file results write &
18   analysis_name = loadcase_1 &
19   file_name = 'variant_1'
20
```

Da nach dem Durchlauf der 4 beschriebenen Schritte die Adams-Session nicht geöffnet bleiben soll, muss für ein automatisches Schließen am Ende des Command-Files noch die folgende Zeile ergänzt werden:

```
21 Quit confirmation = yes
22
```

Ausführung:

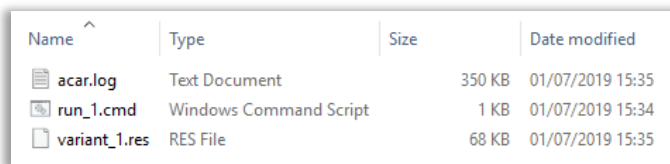
Das Command-File `run_1.cmd` ist somit fertiggestellt und kann von Adams ausgeführt werden. Durch die Automatisierung sind keinerlei Eingaben innerhalb der Programmoberfläche mehr notwendig, sodass wir diese zur Zeitersparnis unterdrücken können. Dazu öffnen wir eine DOS-Shell und wechseln in unser zuvor angelegtes Arbeitsverzeichnis, in welchem auch unser Command-File liegt:

```
cd C:\MeinArbeitsverzeichnis
```

Anschließend rufen wir Adams/Car über den untenstehenden Befehl auf. Der Pfad des `mdi.bat` kann je nach Installation leicht variieren. Entscheidend für die Unterdrückung der Programmoberfläche ist das Argument `b`, welches im Gegensatz zum standardmäßigen `i` für „interactive“ den sogenannten „batch“-Modus festlegt. Als letztes Argument wird mit `run_1.cmd` unser Command-File angegeben.

```
C:\MSC.Software\Adams\2018_1\common\mdi.bat acar ru-st b run_1.cmd
```

Nachdem Adams das Command-File im Hintergrund ausgeführt hat, befinden sich im Arbeitsverzeichnis neben dem `run_1.cmd` zusätzlich das Log-File der Session `acar.log` sowie das Ergebnis-File `variant_1.res`, welches in einem entsprechenden Postprocessing weiterverarbeitet werden kann.



Name ^	Type	Size	Date modified
acar.log	Text Document	350 KB	01/07/2019 15:35
run_1.cmd	Windows Command Script	1 KB	01/07/2019 15:34
variant_1.res	RES File	68 KB	01/07/2019 15:35

Für weitergehende Untersuchungen können nun sowohl im Simulations-Skript als auch im Command-File beliebig Änderungen vorgenommen werden. So können z.B. im Rahmen von Parameter-Studien oder automatisierten Co-Simulationen diese beiden Files als Schnittstelle zu anderen Tools und Programmen dienen.

Komplettes Command-File des gezeigten Beispiels:

```
acar files assembly open &
assembly_name = 'mdids://carat_sample/assemblies.tbl/ComplianceFront.asy' &
variant = default &
open_all = no

car standard_interface hardpoint &
hardpoint = .ComplianceFront.SuspensionFront.ground.hpl_Anbindung_Pendelstuetze &
location = 28.0,-530.0,160.0 &
symmetric = yes

carat compliance simulation submit &
analysis_name = 'variant_1' &
analysis_type = scripted &
model = ComplianceFront &
analysis_script = 'C:\private.cdb\suspension_analysis_scripts.tbl\simulation_1.sas'

file results write &
analysis_name = loadcase_1 &
file_name = 'variant_1'

Quit confirmation = yes
```

Mehr Informationen über Adams: www.mscsoftware.com/de/product/adams