

# DOCsys - Direct Operating Costs of Aircraft Systems

DOCsys berechnet die direkten Betriebskosten von Flugzeugsystemen

Um die Verwendung von Hilfe zu erlernen, drücken Sie die Taste **F1**.

## Hilfen zur Berechnungsmethode

- Eine **Zusammenfassung der Berechnungsmethode** zu DOCsys enthält die Hilfe [DOCsys - Methode](#) (Datei: DOCMATH.HLP).
- **Veröffentlichungen** zu DOCsys sind im Literaturverzeichnis der Hilfe [DOCsys - Methode](#) aufgelistet.
- Eine etwas ausführlichere Darstellung der **Berechnungsmethode mit** Ergebnissen einer **Beispielanwendung** können über das Internet bezogen werden:  
<http://www.fh-hamburg.de/pers/Scholz/paper/DOCsysPaper.pdf>

## Hilfen zur Programmbedienung

### Fenster

[Das Hauptfenster](#)  
[Das Fenster zur Anzeige der numerischen Ergebnisse](#)  
[Graphische Darstellung der Ergebnisse: Balkendiagramm](#)  
[Graphische Darstellung der Ergebnisse: Kreisdiagramm](#)

### Menüpunkte

[Befehle des Datei-Menüs](#)  
[Befehle des Bearbeiten-Menüs](#)  
[Befehle des Berechnen-Menüs](#)

### Methoden

[Öffnen vorhandener Dateien](#)  
[Speichern von Daten in Dateien](#)  
[Kopieren, Ausschneiden und Einfügen von Text](#)  
[Löschen von Text](#)  
[Edit-Text-Tasten](#)

### Dateien

[Dateistruktur: Programm-, Daten- und Ergebnisdateien](#)  
[Arbeitsverzeichnis einstellen](#)  
[Datenbasis](#)  
[Datendatei: EngineData.txt](#)  
[Datendatei: DelayData.txt](#)  
[Datendatei: CostData.txt](#)  
[Datendatei: InflationCountry.txt](#)  
[Initialisierungsdatei: doc.ini](#) Details in: [preference file](#)  
[Initialisierungsdatei: gnuplotPath.ini](#)

### Was sonst noch interessiert

[Zur Geschichte von DOCsys](#)  
[Zur Programmierung von DOCsys](#)  
[Fehlermeldungen](#)  
[Lizenz](#)  
[Support](#)

## Das Hauptfenster

Das Hauptfenster dient zur Eingabe der Parameter. Optisch getrennt werden eingegeben:

Flugzeugdaten,  
allgemeine Systemdaten,  
ökonomische Daten,  
Flumissionsdaten.

In gesonderte Dialogboxen, die über Knöpfe einzeln aufgeblendet werden, können weitere Systemdaten zu den verschiedenen DOC-Bestandteilen eingegeben werden. Es handelt sich um Daten zur

Abschreibung,  
Wartung,  
Verspätung,  
Ersatzteilbevorratung,  
Kraftstoff für fixe Massen,  
Kraftstoff für variable Massen,  
Kraftstoff für Wellenleistung,  
Kraftstoff für Zapfluft,  
Kraftstoff für Stauluft,  
Kraftstoff für Widerstand.

Beim **Speichern** werden die Daten in der entsprechenden Datei als Datenbasis abgelegt.

### Das Menü im Hauptfenster"

**Datei** ist das Pull-Down Datei-Menü mit den Befehlen

Neu  
Öffnen...  
Speichern  
Speichern unter...  
Drucken...  
Beenden

**Bearbeiten** ist das Pull-Down Bearbeiten-Menü mit den Befehlen

Ausschneiden  
Kopieren  
Einfügen

**Berechnen** ist das Pull-Down Berechnen-Menü mit den Befehlen

DOCsys  
erweiterte DOCsys

**Hilfe** ist das Pull-Down Menü mit den Befehlen

Inhalt  
kontext-sensitive Hilfe...  
Hilfe benutzen  
Über...

## Das Fenster zur Anzeige der numerischen Ergebnisse

Das Fenster "Ergebnisse" dient zur **Anzeige der Berechnungsergebnisse** am Bildschirm.

Die Ergebnisse können in eine andere Windows-Anwendungen kopiert oder graphisch dargestellt werden.

Die Berechnungsergebnisse können in eine **Datei ERGEBNIS.TXT** geschrieben werden. Diese Textdatei kann über einen Editor oder ein Textverarbeitungssystem gedruckt werden. Bei einem weiteren Rechenlauf wird die Datei ERGEBNIS.TXT überschrieben. Soll der Inhalt der Datei erhalten bleiben, so muß die Datei vor einer neuen Berechnung umbenannt werden.

Der Text im Fenster "Ergebnisse" kann nicht editiert werden. Ein Editieren ist jedoch mit einem Editor in der Datei ERGEBNIS.TXT möglich. So können z.B. eigene Kommentare zu den Berechnungsergebnissen hinzugefügt werden.

Für einen **zukünftigen Ausbau des Programms** sind bereits Erweiterungsmöglichkeiten vorgesehen:

[[Berechnungsergebnisse können daher auch in die [Datenbasis](#) geschrieben werden. Die Berechnungsergebnisse könnten dann bei einer späteren Erweiterung des Programms in anderen Modulen dann als Eingabe- und Startwerte dienen. Das **Speichern** der ausgewählten Berechnungsergebnisse kann nach erfolgreicher Berechnung und Anzeige der Ergebnisse vorgenommen werden. Dazu wird **Speichern** im Pull-Down-Menü **Ergebnisse** gewählt. Wichtig ist dabei auf eine vollständige und konsistente Datenhaltung zu achten: Zu den Berechnungsergebnissen sollten stets auch die aktuellen Eingabewerte gespeichert werden. Der Anwender wird daher gefragt, ob auch die Eingabewerte zu den Berechnungsergebnissen gespeichert werden sollen. Die Eingabewerte können ferner auch über das Dateimenü des Hauptfensters gespeichert werden. (Siehe [Befehle des Datei-Menüs](#) ).]]

### Das Menü im Fenster zur Anzeige der Ergebnisse

|                   |  |
|-------------------|--|
| <b>Ergebnisse</b> | ist das Pull-Down <a href="#">Datei-Menü</a> mit den Befehlen            |
| Speichern         | Ergebnisse in Datenbasis speichern                                       |
| Drucken...        | Berechnungsergebnisse in Datei ERGEBNIS.TXT schreiben                    |
| Schließen         |  |
| <b>Bearbeiten</b> | ist das Pull-Down <a href="#">Bearbeiten-Menü</a> mit dem Befehl         |
| Kopieren          |  |
| <b>Plot</b>       | erstellt einen Plot aus den Ergebnisdaten (siehe: <a href="#">Plot</a> ) |
| <b>Hilfe</b>      | liefert dieses Help-Topic  |

## Graphische Darstellung der Ergebnisse: Balkendiagramm

Die Anwendung sieht eine graphische Darstellung der Ergebnisse als Säulendiagramme vor. Diese Graphik wird über den Menü-Eintrag **Plot** im Ergebnisfenster erzeugt.

Während der Berechnung der Ergebnisse wurden bereits folgende Schritte durchgeführt:

- 1.) Erstellung einer Datei mit den Daten für den Plot. Diese Datei erhält die Dateinamenserweiterung **.dat**.
- 2.) Erstellung einer Datei mit den Plot-Befehlen. Diese Datei erhält die Dateinamenserweiterung **.plt**.

Eine Datei **bars.ini** im Verzeichnis in dem auch die Datei **doc.exe** liegt, teilt dem Plot-Programm mit, welcher Plot ausgeführt werden soll.

Plot führt folgende Schritte durch:

- 1.) Aufruf des Plot-Programms **GNU PLOT für Windows** als Icon mit dem Namen "gnuplot".
- 2.) Öffnen eines Fensters zur Anzeige des Plots. Das Plot-Fenster trägt den Namen "gnuplot graph".

**Modifikationen am Plot** können sofort mit Hilfe des Fenster "gnuplot" durchgeführt werden.

**Ausdrucke des Plots** können entweder sofort direkt im Fenster "gnuplot graph" oder im Fenster "gnuplot" durchgeführt werden.

**Spätere Arbeiten an den Plots** sind mit GNU PLOT möglich, wenn die erstellten Dateien mit den Dateinamenserweiterungen **.dat** und **.plt** genutzt werden. Nach einer neuen Berechnung von Ergebnissen werden die Dateien mit der Dateinamenserweiterung **.dat** und **.plt** überschrieben. Soll mit den Dateien später noch gearbeitet werden (für weitere Plots), so müssen die Dateien rechtzeitig umbenannt werden.

**Hilfe zu GNU PLOT** erhalten Sie über die Windows-Hilfe in GNU PLOT, oder durch einen Ausdruck der Anleitung zu GNU PLOT. Die Anleitung liegt als Postscript-Datei und als PDF-Datei GNU PLOT-Verzeichnis.

**Zur Installation von GNU PLOT:** In der Initialisierungsdatei (preference file) **doc.ini** zur Anwendung wird angegeben in welchem Verzeichnis GNU PLOT für Windows installiert worden ist.

## Graphische Darstellung der Ergebnisse: Kreisdiagramm

Automatisch wird mit jeder Rechnung die Aufteilung der DOC-Kostenelemente in einem Kreisdiagramm angezeigt. Das Fenster in dem das Diagramm angezeigt wird trägt den Titel **Kreisdiagramm** ("Tortendiagramm"). Das Kreisdiagramm kann in andere Anwendungen übernommen werden. Dazu muß das Fenster mit dem Diagramm als aktives Fenster ausgewählt werden. Anschließend ist die Tastenkombination <Alt> <Druck> zu betätigen. Dadurch gelangt ein Screen Shot des Fensters in die Zwischenablage. Mit **Einfügen** kann das Bild dann in andere Anwendungen aufgenommen werden.

## Befehle des Datei-Menüs

Das Datei Pull-Down Menü enthält folgende Befehle:

### **Neu**

Löscht die Eingabefelder für die Erstellung einer neuen Datenbasis.

Wenn Sie den Befehl **Neu** wählen, können Sie die Änderungen an Ihrem zuletzt erarbeiteten Datenbasis speichern.

### **Öffnen**

Öffnet eine Datei.

Die Datei muß eine Datenbasis enthalten. Eine solche Datei entsteht durch Abspeichern der eingegebenen Werte mit **Speichern** oder **Speichern unter**.

Wenn Sie den Befehl **Öffnen** wählen, können Sie die gegebenenfalls vorgenommenen Änderungen an Ihrem zuletzt bearbeiteten Dokument speichern.

Verwandtes Thema: [Öffnen vorhandener Dateien](#)

### **Speichern**

Speichert die Daten. Die Daten werden in der Datei gespeichert, die im Titelbalken des Fensters angegeben ist. Ist noch kein Dateiname vereinbart (Dateiname: "unbenannt"), wird automatisch der Befehl **Speichern unter** ausgeführt.

Wenn Sie den Befehl **Speichern** wählen, bleibt das Dokument geöffnet, damit Sie weiter daran arbeiten können.

Verwandtes Thema: [Speichern von Daten](#)

### **Speichern unter**

Speichert den Fensterinhalt in einer neuen oder vorhandenen Datei.

Sie können eine neue Datei benennen, oder Daten einer bestehenden Datei unter einem neuen Namen speichern. Das Original bleibt dabei unverändert.

Wenn Sie den Befehl **Speichern unter** wählen, bleibt das Dokument geöffnet, damit Sie weiter daran arbeiten können.

Verwandtes Thema: [Speichern von Daten](#)

### **Drucken**

Schreibt den Inhalt des Fensters in eine Textdatei **DRUCK.TXT**. Diese Datei kann in einen Editor oder ein Textverarbeitungssystem geladen werden und anschließend gedruckt werden.

### **Beenden**

Schließt alle Fenster und beendet die Anwendung.

Nur das Hauptfenster der Anwendung enthält den Befehl **Beenden** im Datei Pull-Down Menü, andere Fenster enthalten den Befehl **Schließen**. Dies schließt nur das jeweilige Fenster, die Anwendung wird NICHT beendet.

## Befehle des Bearbeiten-Menüs

### Ausschneiden

Löscht einen markierten Text und kopiert ihn in die Zwischenablage. Dabei wird der bisherige Inhalt der Zwischenablage überschrieben.

Verwenden Sie den Befehl **Einfügen**, um den ausgeschnittenen Text in das gleiche Fenster, ein anderes Fenster oder eine andere Anwendung einzufügen.

Verwandte Themen:

[Kopieren, Ausschneiden und Einfügen von Text](#)  
[Löschen von Text](#)

### Kopieren

Kopiert Text aus einem Fenster in die Zwischenablage. Das Original bleibt dabei unverändert erhalten. Der bisherige Inhalt der Zwischenablage wird überschrieben.

Verwenden Sie den Befehl **Einfügen**, um den ausgeschnittenen Text in das gleiche Fenster, ein anderes Fenster oder eine andere Anwendung einzufügen.

Verwandtes Thema:

[Kopieren, Ausschneiden und Einfügen von Text](#)

### Einfügen

Fügt eine Kopie des Inhalts der Zwischenablage an der Einfügemarke oder vor einem markiertem Text ein.

Verwandtes Thema:

[Kopieren, Ausschneiden und Einfügen von Text](#)

## Öffnen vorhandener Dateien

Die Anwendung arbeitet nur mit solchen Dateien, die Daten in einem bestimmten Textformat enthalten. Dies sind solche Dateien, die vorher mit dem Befehlen **Speichern** oder **Speichern unter** erstellt wurden. Die Anwendung speichert die Daten in einer ASCII-Datei als Datenbasis. Jede Datenbasis ist charakterisiert durch die erste Zeile der Datei. Diese hat das Aussehen

### Datenbasis ### .

Öffnen einer Datei:

1. Wählen Sie aus dem **Datei-Menü** den Befehl **Öffnen**.
2. Markieren Sie das Laufwerk bzw. das Verzeichnis, in dem sich die Datei befindet.
3. Geben Sie den Dateinamen ein oder wählen Sie ihn aus.
4. Wählen Sie die Schaltfläche **OK**.

Verwandtes Thema:

[Speichern von Daten](#)



## Speichern von Daten in Dateien

### A ) Speichern der Eingabedaten in der Datenbasis

1. Wählen Sie aus dem Datei-Menü den Befehl **Speichern unter**.
2. Markieren Sie das Laufwerk bzw. das Verzeichnis, in das die Datei gespeichert werden soll.
3. Geben Sie einen Dateinamen ein oder wählen Sie den Dateinamen aus.
4. Wählen Sie die Schaltfläche **OK**.

### B ) Speichern der Eingabedaten in einer Textdatei für den späteren Ausdruck

Mit dem Befehl **Drucken** aus dem Datei-Menü können die Daten des Hauptfensters in einer Textdatei **DRUCK.TXT** im aktuellen Verzeichnis gespeichert werden. Diese Datei kann mit Befehlen des Betriebssystems gedruckt werden.

### C ) Speichern von Berechnungsergebnissen (Option für spätere Programmerweiterungen, zur Zeit inaktiv)

Berechnungsergebnisse können in die Datenbasis geschrieben werden. Diese Werte könnten in zukünftigen Modulen als Eingabe- und Startwerte für weiterführende Berechnungen dienen. Das Speichern der ausgewählten Berechnungsergebnisse kann nach erfolgter Berechnung und Anzeige der Ergebnisse erfolgen:

1. Wählen Sie **Speichern** im Pull-Down-Menü des Fensters **Ergebnisse**.
- > Hinweis: Achten Sie auf eine vollständige und konsistente Datenhaltung. Zu den Berechnungsergebnissen sollten stets auch die aktuellen Eingabewerte gespeichert werden.
2. Sie werden gefragt: "Eingaben jetzt speichern?" Wählen Sie **Ja** um die Eingabedaten zu den Berechnungsergebnissen zu speichern.
- > Die Eingaben können ferner jederzeit über das Dateimenü des Hauptfensters gespeichert werden. (Siehe Befehle des Datei-Menüs).

## Kopieren, Ausschneiden und Einfügen von Text

### Kopieren und Einfügen oder Ausschneiden und Einfügen von Text

1. Markieren Sie den Text.
2. Um Text zu kopieren, wählen Sie aus dem Bearbeiten-Menü den Befehl **Kopieren**. Um Text auszuschneiden, wählen Sie aus dem Bearbeiten-Menü den Befehl **Ausschneiden**.
3. Bewegen Sie die Einfügemarke an die Stelle, an der der Text eingefügt werden soll.
4. Wählen Sie aus dem Bearbeiten-Menü den Befehl **Einfügen**.

Verwandtes Thema:

[Löschen von Text](#)

## Löschen von Text

### Löschen eines einzelnen Zeichens

- Drücken Sie die RÜCKTASTE, um das Zeichen links neben der Einfügemarke zu löschen.
- Drücken Sie die ENTF-TASTE, um das Zeichen rechts neben der Einfügemarke zu löschen.

### Löschen mehrerer Zeichen

1. Markieren Sie den Text, den Sie löschen möchten.
2. Drücken Sie die RÜCKTASTE oder die ENTF-TASTE.  
Um den Text in die Zwischenablage zu kopieren, wählen Sie aus dem [Bearbeiten-Menü](#) den Befehl **Ausschneiden**.

Verwandtes Thema:

[Kopieren, Ausschneiden und Einfügen von Text](#)

## Edit-Text-Tasten

Im Edit-Text können Sie die folgenden Tasten benutzen.

### Bewegen der Einfügemarke

| Taste(n)         | Bewegt die Einfügemarke  |
|------------------|--------------------------|
| NACH-OBEN        | Eine Zeile nach oben.    |
| NACH-UNTEN       | Eine Zeile nach unten.   |
| NACH-RECHTS      | Ein Zeichen nach rechts. |
| NACH-LINKS       | Ein Zeichen nach links.  |
| STRG+NACH-RECHTS | Ein Wort nach rechts.    |
| STRG+NACH-LINKS  | Ein Wort nach links.     |
| POS1             | Zum Zeilenanfang.        |
| ENDE             | Zum Zeilenende.          |
| BILD-NACH-OBEN   | Eine Fenster nach oben.  |
| BILD-NACH-UNTEN  | Ein Fenster nach unten.  |
| STRG+POS1        | Zum Dokumentanfang.      |
| STRG+ENDE        | Zum Dokumentende.        |

### Markieren von Text

Die nachfolgenden Verfahren, um Text zu markieren, verwenden als Ausgangspunkt die Position der Einfügemarke. Wenn bereits Text markiert wurde, wird durch Drücken einer dieser Tasten die Markierung aufgehoben.

| Taste(n)                        | Markiert Text oder hebt Markierung auf |
|---------------------------------|--|
| UMSCHALT+NACH-RECHTS/NACH-LINKS | Zeichenweise nach links oder rechts.   |
| UMSCHALT+NACH-UNTEN/NACH-OBEN   | Zeilenweise nach oben oder nach unten. |
| UMSCHALT+BILD-NACH-OBEN         | Fensterweise nach oben.                |
| UMSCHALT+BILD-NACH-UNTEN        | Fensterweise nach unten.               |
| UMSCHALT+POS                    | Bis zum Zeilenanfang.                  |
| UMSCHALT+ENDE                   | Bis zum Zeilenende.                    |
| STRG+UMSCHALT+NACH-LINKS        | Vorhergehendes Wort.                   |
| STRG+UMSCHALT+NACH-RECHTS       | Nächstes Wort.                         |
| STRG+UMSCHALT+POS               | Bis zum Dokumentanfang.                |
| STRG+UMSCHALT+END               | Bis zum Dokumentende.                  |

## Initialisierungsdatei: **doc.ini** (preference file)

Aufgabe der Initialisierungsdatei **doc.ini** ist, die Anwendung mit Parametern zur Steuerung des Ablaufs zu versorgen. Insbesondere wird das Graphical User Interface (GUI) initialisiert und das Aussehen und Verhalten der Benutzeroberfläche definiert. Von der langen Definition des GUIs in **doc.ini** sind nur zwei **Preference Variables** von besonderer Bedeutung:

OUIDICTPATH legt fest, in welchen Verzeichnissen nach den **Dictionaries** mit der Endung \*.oui gesucht werden soll.

OUIHELPPATH legt fest, in welchen Verzeichnissen nach den **Hilfe-Dateien** mit der Endung \*.hlp gesucht werden soll.

Im einfachsten Fall, wenn die Dictionaries und die Hilfe-Dateien zusammen im gleichen Verzeichnis liegen wie die DOCsys-Programmdatei **doc.exe** ist zu setzen:

```
OUIDICTPATH=%A  
OUIHELPPATH=%A
```

Weitere Informationen zu [preference files](#) habe ich einer entsprechenden Hilfe-Datei der Firma "Open Software Associates" entnommen, mit deren Produkt - "OpenUI" - das GUI von DOCsys erstellt wurde.

## Initialisierungsdatei: **gnuplotPath.ini**

Die Initialisierungsdatei **gnuplotPath.ini** dient der Anbindung von **gnuplot** an DOCsys. Die Datei enthält eine Zeile, die z.B. so aussehen kann:

```
gnuplotExe c:\progra~1\gnuplot\wgnuplot.exe
```

DOCsys wird damit gesagt, daß die gnuplot-Programmdatei im Verzeichnis *c:\Programme\gnuplot* liegt und den Programmnamen *wgnuplot.exe* trägt.

An diesem Beispiel wird deutlich, daß der Pfad in DOS-Schreibweise ausgeführt sein muß. Wie der jeweilige Pfad unter DOS erscheint, wird in einem DOS-Fenster mit dem DIR-Befehl angezeigt.

## Datenbasis

Die Datenbasis wurde so aufgebaut, daß verschiedene **Module eines Programmpaketes über die Datenbasis kommunizieren können**. Mit DOCsys liegt nur ein einzelnes Programm vor, so daß die Datenbasis im wesentlichen nur als ein einfaches Format zur Speicherung der Eingabedaten dient. Prinzipiell kann die Datenbasis sowohl die Eingabedaten als auch die Ergebnisdaten aufnehmen. Die Ergebnisdaten eines vorgelagerten Moduls können dabei als Eingabedaten für ein nachgelagertes Modul dienen... Aber wie gesagt, eine derartige Anwendung bleibt zukünftigen Erweiterungen des Programms vorbehalten.

**Eine Datenbasis wird angelegt**, indem **Speichern** oder **Speichern Unter...** im Datei-Menü gewählt wird. Aus einer Datenbasis kann gelesen werden, indem **Öffnen** im Datei-Menü gewählt wird.

Für die Datenbasis ist **keine spezielle Form des Dateinamens** vorgesehen. Es ist jedoch zu beachten, daß nur Dateien gelesen werden können, die eine Datenbasis enthalten. Zum leichteren Auffinden sollte der Anwender daher seine Datenbasen leicht identifizierbar gestalten. Empfohlen wird ein Dateiname der Form **\*.db** .

Eine **Datenbasis ist eine Datei im ASCII-Format**. Der Anwender muß sich nicht mit dem Inhalt dieses ASCII-Textes auseinandersetzen. Dies geschieht standardmäßig über die Benutzeroberfläche. Sollte jedoch die Notwendigkeit entstehen Einblick in die Datenbasis zu nehmen, so ist dies sehr wohl möglich. (Siehe: [Arbeiten mit der Datenbasis](#))

## Arbeiten mit der Datenbasis

In der Regel wird der Anwender den ASCII-Text der Datenbasis nicht betrachten. Die Datenbasis wird über die Benutzeroberfläche der Anwendung manipuliert. (Siehe: [Datenbasis](#))

Eine **Datenbasis ist durch ihre erste Zeile gekennzeichnet**. Diese lautet `### Datenbasis ###`. Die Datenbasis enthält eine Datenliste. Die **Einträge in der Datenliste** sind formatiert gemäß:

Schlüsselwort <TAB> Bezeichnung <TAB> Einheit <TAB> Wert

Über das 'Schlüsselwort' werden die Einträge identifiziert. Die 'Bezeichnung' und die 'Einheit' sind Einträge, die rein zur Information dienen. Der 'Wert' enthält den eigentlichen Inhalt zum Schlüsselwort. Dabei kann es sich sowohl um eine Zahl, einen Text oder um ein Zahlenfeld handeln. Die Schlüsselworte sind hierarchisch gegliedert. Die Gliederungsebenen werden durch einen Punkt getrennt. Der letzte Anteil des Schlüsselwortes stimmt in der Regel mit der Bezeichnung der Variablen überein.

Der Inhalt der Datenbasis kann in Datenbanken oder Tabellenkalkulationsprogramme eingelesen werden. Dies wird durch die Struktur der Datenliste (Trennung der Einträge durch <TAB>) möglich.

Die Reihenfolge der Einträge in der Datenliste entspricht der Chronologie ihrer Entstehung. Möchte man einen inhaltlichen **Überblick über die Daten der Datenliste** erhalten, so können die Zeilen der Datenliste alphabetisch sortiert werden. Dazu bedient man sich eines Textverarbeitungsprogramms oder man sortiert einfach auf Betriebssystemebene: Unter DOS bzw. UNIX kann der Befehl **SORT** auf die Datenbasis angewandt werden.

Eine übersichtliche **Ausgabe der Datenliste** erfolgt am besten über ein Textverarbeitungsprogramm bei dem die Tabulatoren so gesetzt wurden, daß die durch <TAB> getrennten Einträge der Datenliste in getrennten Spalten angeordnet werden.



## Datendatei: EngineData.txt

Die **Datendateien** dienen der **Bereitstellung grundlegender Parameter** für das Programm **DOCsys**. Die Parameter der Datendateien wurden in der Regel über einen längeren Zeitraum sorgfältig recherchiert und können bei einer Reihe von Berechnungen unverändert verwendet werden. Trotzdem wird es erforderlich sein auch derartige grundsätzliche Parameter zu ergänzen, in ihrer Genauigkeit zu verbessern oder dem neuen Stand der Technik anzupassen.

Die Datei EngineData.txt hat folgenden **Aufbau**:

```
/* "Triebwerksdeck" */
/* Triebwerksname Tmax[N] SFC_CLB[kg/Ns] SFC_CR[kg/Ns] SFC_DES[kg/Ns] Hersteller Einsatz Refe
V2500-A5          133347  16.095e-6  16.68e-6  16.68e-6  IAE          A320  Copp
...
```

Darin bedeutet:

|         |   |
|---------|---|
| Tmax    | Standschub                                    |
| SFC_CLB | spezifischer Kraftstoffverbrauch im Steigflug |
| SFC_CR  | spezifischer Kraftstoffverbrauch im Reiseflug |
| SFC_DES | spezifischer Kraftstoffverbrauch im Sinkflug  |

Die erste Spalte der Datei **EngineData.txt** wird darüberhinaus genutzt, um im GUI von DOCsys die Namen aller Triebwerke anzuzeigen, für die Daten vorhanden sind und die daher für die DOC-Rechnung genutzt werden können.

## Datendatei: DelayData.txt

Die **Datendateien** dienen der **Bereitstellung grundlegender Parameter** für das Programm **DOCsys**. Die Parameter der Datendateien wurden in der Regel über einen längeren Zeitraum sorgfältig recherchiert und können bei einer Reihe von Berechnungen unverändert verwendet werden. Trotzdem wird es erforderlich sein auch derartige grundsätzliche Parameter zu ergänzen, in ihrer Genauigkeit zu verbessern oder dem neuen Stand der Technik anzupassen.

Die Datei DelayData.txt hat folgenden **Aufbau**:

```
/* Wahrscheinlichkeit von Verspaetungen und Absagen in % [Lufthansa 11/95 - 04/96] */
ATA-Kapitel  D_I    D_II   D_III  D_C
21           0.0191  0.0527  0.0000  0.0029
22           0.0000  0.0259  0.0105  0.0011
23           0.0042  0.0377  0.0000  0.0007
...
49           0.0389  0.0260  0.0000  0.0004
```

Darin bedeutet:

```
D_I    Wahrscheinlichkeit für eine Verspätung der Kategorie I (bis 29 min.) in %
D_II   Wahrscheinlichkeit für eine Verspätung der Kategorie II (30 min. bis 59 min.) in %
D_III  Wahrscheinlichkeit für eine Verspätung der Kategorie III (mehr als 60 min.) in %
D_C    Wahrscheinlichkeit für eine Flugabsage in %
```

## Datendatei: CostData.txt

Die **Datendateien** dienen der **Bereitstellung grundlegender Parameter** für das Programm **DOCsys**. Die Parameter der Datendateien wurden in der Regel über einen längeren Zeitraum sorgfältig recherchiert und können bei einer Reihe von Berechnungen unverändert verwendet werden. Trotzdem wird es erforderlich sein auch derartige grundsätzliche Parameter zu ergänzen, in ihrer Genauigkeit zu verbessern oder dem neuen Stand der Technik anzupassen.

Die Datei CostData.txt hat folgenden **Aufbau**:

```
Regressionsgrade (a*seats + b) zur Berechnung der Kostengruppen
C_I: 0-29min; C_II: 30-59min; C_III: >59min; C_C: Flugabsage
a      b      in 1976$
0.658   185.7
1.701   468.3
5.088  2544.4
6.556  3389.1
Inflationsdaten fuer Deutschland
Jahr    Inflationsrate
1976    0.044
...
```

Die Kosten, die durch Verspätungen und Flugabsagen entstehen sind von der Größe der Flugzeuge abhängig. Die Flugzeuggröße wird ausgedrückt durch die Anzahl der Sitze (Standard-Layout) des Flugzeugs. Die Kosten für eine Verspätung der Kategorie I (bis 29 min.) berechnet sich dann mit den Daten der ersten Datenzeile zu

$$C_I = a * \text{seats} + b$$

bzw.

$$C_I = 0.658 \text{ US\$/seat} * \text{seats} + 185.7 \text{ US\$}$$

Entsprechend geht das Programm mit den anderen Kostengruppen vor. Die Bedeutung der Kostengruppen ist zusammenfassend:

```
C_I    Kosten in US$ für eine Verspätung der Kategorie I (bis 29 min.)
C_II   Kosten in US$ für eine Verspätung der Kategorie II (30 min. bis 59 min.)
C_III  Kosten in US$ für eine Verspätung der Kategorie III (mehr als 60 min.)
C_C    Kosten in US$ für eine Flugabsage
```

Die so berechneten Kosten sind Kosten in US\$ des Jahres 1976. Mit den Inflationsraten der jeweiligen Jahre ab 1976 kann auf das Kosten-Level anderer Jahre umgerechnet werden. In diesem Beispiel wurden die Inflationsraten von Deutschland zugrunde gelegt, was in einer Kommentarzeile vermerkt wurde. Im Programm **DOCsys** in der Dialogbox "Verspätung ..." ist ebenfalls angegeben auf welches Land sich die Umrechnung der Inflationsraten bezieht. Der Eintrag wird gesetzt aufgrund eines Ländereintrags in der Datei **InflationCountry.txt**.

## Datendatei: InflationCountry.txt

Die Datei **InflationCountry.txt** enthält die Bezeichnung des Landes nach dessen Inflationsdaten die Kosten der Verspätungen und Flugabsagen korrigiert werden - entsprechend den Daten in der Datei CostData.txt. Der Name des Landes, der in der Datei **InflationCountry.txt** enthalten ist wird in der Dialogbox "Daten zu Verspätungen und Flugabsagen" angezeigt.

## Dateistruktur: Programm-, Daten- und Ergebnisdateien

- Im **Programmverzeichnis** liegen die **Programmdateien**. Programmdateien sind die Dateien, die zusammen das Programm DOCsys bilden. Zum Programmverzeichnis gehören folgende Dateien:

Programmdatei:  
doc.exe

Initialisierungsdatei:  
doc.ini

Hilfedateien:  
DOC.HLP  
DOCMATH.HLP

Dictionaries zur Definition des Graphical User Interfaces (GUI):  
doc.oui  
basis.oui  
stddict.oui

Direct Link Library (für GUI):  
openui40.dll

- Im **Arbeitsverzeichnis** liegen die **Datenbasen und Datendateien** sowie eine weitere Initialisierungsdatei. Es können mehrere Arbeitsverzeichnisse angelegt werden, die dann alle die folgenden Dateien enthalten:

Datenbasen:  
\*.db (Verabredungsgemäß wird die Endung 'db' benutzt. Dies ist aber nicht zwingend.)

Datendateien:  
CostData.txt  
DelayData.txt  
EngineData.txt  
InflationCountry.txt

Eine Initialisierungsdatei zur Anbindung von gnuplot an DOCsys:  
gnuplotPath.ini

- Im **Arbeitsverzeichnis** werden vom Programm DOCsys ebenfalls die **Ergebnisdateien** abgelegt:

Dateien, die automatisch bei jedem Programmlauf entstehen:  
gnuplot.ini (wird nach der Beendigung des Programms automatisch wieder gelöscht)  
bars.dat  
bars.plt

Dateien für Ausdrücke:  
ergebnis.txt (ASCII-Datei mit Kopie des Ergebnisfensters.  
Wird erzeugt nach anklicken von "Ergebnisse -> Drucken...")  
druck.txt (ASCII-Datei mit allen Eingabedaten in übersichtlicher Form.  
Wird erzeugt nach anklicken von "Datei -> Drucken...")

## Arbeitsverzeichnis einstellen

Sinnvoll ist, auf dem Desktop eine **Verknüpfung** mit doc.exe abzulegen. Von diesem Programm-Icon aus kann DOCsys dann immer aufgerufen werden. Klickt man das Icon auf dem Desktop mit der rechten Maustaste an, so kann man die **Eigenschaften** bearbeiten:



Insbesondere kann hier das **Arbeitsverzeichnis** eingestellt werden. Das Arbeitsverzeichnis ist das Verzeichnis, in dem DOCsys zum gegenwärtigen Zeitpunkt die Berechnungen durchführen soll. In dieses Verzeichnis müssen folgende Dateien kopiert werden: [EngineData.txt](#), [CostData.txt](#), [DelayData.txt](#), [InflationCountry.txt](#) und [gnuplotPath.ini](#). Siehe dazu auch [Dateistruktur](#).

Es ist darauf zu achten, daß die Datendateien die für die Rechnungen passenden Daten enthalten.

## Befehle des Berechnen-Menüs

Eine **Zusammenfassung der Berechnungsmethode** zu DOCsys enthält [docmath.hlp](#).

Zwei verschiedene Berechnungsarten können nach Dateneingabe gestartet werden.

### **DOCsys**

Berechnung der DOC in Anlehnung an die Definition nach [ATA 1967](#)

DOCsys =

- Abschreibung
- + Wartungskosten
- + Kraftstoffkosten

### **erweiterte DOCsys**

Berechnung von DOC, die gegenüber [ATA 1967](#) erweitert wurden um Anteile, die eigentlich zu den Indirect Operating Costs (IOC) gezählt werden, jedoch eindeutig durch die Flugzeugsysteme beeinflusst sind:

erweiterte DOCsys =

- Abschreibung
- + Kraftstoffkosten
- + Wartungskosten
- + Kosten durch Verspätungen und Flugabsagen
- + Kosten durch das gebundene Kapital der Ersatzteile im Lager

## Eingabe der Parameter

Die erforderlichen Eingabeparameter sind in den Eingabemasken beschrieben.

Die Eingabefelder für numerische Werte erlauben nur **zulässige Zeichenfolgen** einzugeben. Eine zulässige Zeichenfolge kann maximal so aussehen:

Vorzeichen, Ziffernfolge, . , Ziffernfolge, E, Vorzeichen, Ziffernfolge

Zulässig wäre z.B. die Eingaben

23    oder    23.5    oder    .5    oder    +23.5    oder    -23.5    oder    -23.5E5    oder  
-23.5e5    oder    -23.5e-5    oder    23.5e-112

**Nicht** zulässig wäre jedoch die Eingaben

23,5    ( , statt . )  
abc    (keine Zahlendarstellung)  
23.3.4    (zwei Dezimalpunkte)

Die **Statuszeile** am unteren Ende des Hauptfensters gibt den zulässigen **Wertebereich der Eingabeparameter** an. Der Rechner prüft vor jedem Rechengang, ob die Eingabeparameter den zulässigen Wertebereich überschritten haben. Ist dies der Fall, so wird der Benutzer darauf aufmerksam gemacht.



## Zur Geschichte von DOCsys

Das Programm DOCsys und die Methode zur Berechnung von direkten Betriebskosten von Flugzeugsystemen nahm seinen ersten Anfang während meiner Arbeit an der Technischen Universität Hamburg Harburg (TUHH) am Arbeitsbereich Flugzeug-Systemtechnik. Zwei engagierte Studenten waren daran beteiligt: Heiko Zimmermann untersuchte im Rahmen seiner Diplomarbeit die "herkömmlichen" Direct Operating Costs (DOC) und stellte entsprechend erste Gleichungen für Flugzeugsysteme auf. Rouven Westphal arbeitete im Rahmen seiner Studienarbeit an den Methoden weiter und half bei der Programmierung von DOCsys.

Bei einer Reihe von Aufträgen die "Applied Science" im Rahmen des Luftfahrtforschungsprogramms bearbeitete, nutzte ich DOCsys. Dabei verfeinerte ich die Methode und verbesserte das Programm kontinuierlich.

Das Programmkonzept welches DOCsys zugrunde liegt, ist erheblich umfassender als es hier zur Anwendung kommt. Im Rahmen meiner Promotion an der TUHH arbeitete ich daran, eine Reihe von Programmmodulen zum Entwurf von Flugzeugsystemen miteinander zu einem integrierten Tool zusammenzuschalten. Dabei hatte ich mir zum Ziel gesetzt, die Programmierung hardware-unabhängig zu gestalten. Der Programmcode sollte sowohl auf PCs mit Windows laufen als auch auf Workstations mit UNIX/Motif. Mehrere Module wurden erstellt und die Realisierbarkeit der gesteckten Ziele nachgewiesen. Zur allgemeinen Reife kam dann aber nur DOCsys. An der Uni war eine ausgereifte Programmentwicklung nicht Ziel der Arbeit. DOCsys entwickelte ich "nebenbei" weiter, was an der großen Nachfrage nach Betriebskostenrechnungen bei Applied Science lag. Andere Module wurden aus Zeitmangel nicht weitergeführt - obwohl in einem Fall bereits ein Auftrag vorlag.

## Zur Programmierung von DOCsys

DOCsys ist in ANSI-C geschrieben. Die Oberfläche (Graphical User Interface, GUI) wurden mit *OpenUI* einem objektorientierten *User Interface Management System* (UIMS) der Firma *Open Software Associates* (OSA) erstellt. Das GUI wird von C aus in's Leben gerufen. Danach kommunizieren C-Programm und GUI über "Messages". Daten werden teilweise direkt von C aus in's GUI geschrieben. Das Kreisdiagramm wurde mit *OpenUI* und C programmiert und ist Teil des GUI.

Die Hilfe wurde mit dem Programm *Help Magician* von *Zoschke Data* erstellt. DOCsys enthält eine kontext-sensitive Hilfe. GUI und Hilfesystem verständigen sich über "Context Numbers".

Das Balkendiagramm wird mit *gnuplot* erstellt, einem "public domain" Programm von Thomas Williams und Colin Kelley. *gnuplot* ist ein interaktives Programm - es kann aber auch von Dateien gesteuert werden. Angebunden an DOCsys werden die Dateien mit den aktuellen Daten von DOCsys erstellt. Anschließend wird *gnuplot* von DOCsys aufgerufen und arbeitet (von *gnuplot.ini* veranlaßt) die Anweisungen in der Datei *bars.plt* ab, wobei die Daten in *bars.dat* verwendet werden.

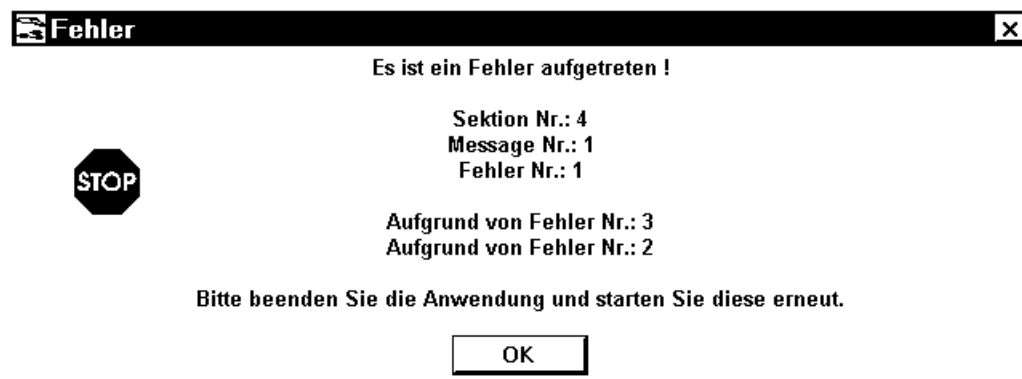
Über die Zwischenablage und die Datei *ergebnis.txt* können Daten zur Nachbearbeitung in andere Programme übertragen werden. So können beliebige andere graphische Darstellungen mit einem Tabellenkalkulationsprogramm erstellt werden. Das Kreisdiagramm kann mit der Tastenkombination <Alt> <Druck> in die Zwischenablage übertragen werden. Anschließend kann das Bild in einem Graphikprogramm weiter verarbeitet werden.

## Fehlermeldungen

Fehler lassen sich leider nicht ganz vermeiden. Zum einen kann (ausnahmsweise) der Anwender an einem "Fehler" schuld sein. In diesem Fall versucht DOCsys eine verständliche Meldung abzugeben. Die Meldung besagt 1.) was passiert ist und 2.) was zu tun ist. Ein Beispiel ist diese Fehlermeldung:



Es können jedoch auch andere Ereignisse auftreten, die eine Situation hervorrufen, mit denen das Programm nicht umgehen kann. Ich habe versucht, alle derartigen Ereignisse im Programm abzufangen. Die Ereignisse werden in Form von Fehlerziffern von Unterprogramm zu Unterprogramm nach oben zum Hauptprogramm gemeldet. Dort werden die Fehlerziffern dann zur Anzeige gebracht:



Die Fehlermeldung kann nur anhand des C-Programms analysiert werden. Das Programm arbeitet nach derartigen Fehlern in der Regel noch weiter. Da aber nicht klar ist, welche weiteren Konsequenzen der Fehler hat, sollten die Daten gesichert und das Programm beendet werden. Anschließend sollte das DOCsys dann erneut aufgerufen werden. Durch einen "work around" sollte der Versuch unternommen werden, die Fehlermeldung zu vermeiden. Wird die Sache zu "nervig", dann muß der [Support](#) 'ran. Ich hoffe, daß mit diesen Maßnahmen der Fehlererkennung "sang- und klanglose Untergänge" des Programms nur äußerst selten auftreten.

## Lizenz

Die Programmierung hat so manche Nacht gekostet. Unter Berücksichtigung dessen, was ich für die Programmierwerkzeuge ausgegeben habe, liegt mein Stundenlohn irgendwo **unter** 0 DM. Um diese Situation zu verbessern, könnte ich mir vorstellen das Programm evtl. mehr als einmal zu verkaufen: Daher ein gewisser Schutz.

- Das Programm darf vertragsgemäß nur innerhalb der vereinbarten Laufzeit der Lizenz genutzt werden.
- Das Programm darf gleichzeitig nur auf der vertraglich geregelten Anzahl an Computern gleichzeitig installiert sein.
- Das Programm darf über den vereinbarten Nutzerkreis hinaus nicht weitergegeben werden.

Das Programm DOCsys versucht daran zu erinnern, diese Vereinbarungen einzuhalten. Falls DOCsys nicht will, obwohl es doch eigentlich laufen sollte, dann bitte 'mal kurz beim [Support](#) melden.

## Preference File

You can use OpenUI preference files to set up your OpenUI environment. Preference files contain variables to allow OpenUI to locate files and determine preferences in your environment. Preference files are easy to customize for your environment, and multiple preference files may exist in any environment. This means that:

- Multiple OpenUI installations can exist independently on one platform.
- OpenUI applications with different preferences can run simultaneously on one platform.
- Both application developers and end-users can control preference variables.

### See also:

[Initialisierungsdatei](#)

[Preference files structure](#)

[Editing preference files](#)

[Field descriptors \(% substitution fields\)](#)

[Application preference files](#)

[Default search order](#)

## Preference file structure

Each preference file is divided into a number of sections. Sections group together related sets of preference variables. Sections are delimited by section tags enclosed in square brackets []. For example, [Color]. Within each section, you can assign values to a number of preference variables. For example

AutoCompact=TRUE

assigns the value true to the preference variable AutoCompact. There are some guidelines to follow when you are editing your preference files. Within a preference file, OpenUI looks for the following set of tags:

- [Color]
- [Environment]
- [Internationalization]
- [Remote Application]
- [Remote Environment]
- [Virtual Keys]
- [Font Variables]

Back to Start of Tutorial on the [Preference File](#).

## Editing preference files

Apply the following rules when directly editing OpenUI preference files:

- Keep <preference variable> = <value> assignments on separate lines. Don't extend them over more than one line
- Put a semicolon in the first character position of comment lines
- Use blank lines as you want to. OpenUI ignores them
- Don't put spaces in preference variable names
- You can put spaces in values. OpenUI discards the spaces between the key name or value and the "=" trailing white space on the value
- Don't have lines longer than 512 characters, including trailing newline characters
- Preference variable names and section names are not case-dependent
- Boolean preference variable can be set to true, false, TRUE, FALSE, yes, no, YES or NO

Back to Start of Tutorial on the [Preference File](#).

## Field descriptors (% substitution fields)

Some preference variables can be constructed using field descriptors. The field descriptors are expanded when OpenUI reads the preference file. Both end-users and application developers can use field descriptors when building preference files. Supported field descriptors are:

%%      %.      %;      %:

These fields expand to a single instance of the second character in the field. For example, %% expands to "%". The expressions %<separator> expand to a single instance of the platform-specific separator. Only the field separator for the platform is meaningful, expressions containing field separators for other platforms are ignored. The platform-specific separators are for Windows a semicolon (;).

%A      Application

The pathname of the directory in which the current application executable resides. For Windows: The pathname of the application directory. Note:

The pathname does not include a trailing separator. That is (using Microsoft Windows), if the application is in \bin, the value of %A is "bin", not "bin\".

%B      Base

This field descriptor is for Unix only.

%E      Environment

The value of the environment variable of the same name as the preference variable. %E can only be used in a preference file:

[Environment] NLSPATH=%E

%L      Language

The value of the LANG environment variable, on supported platforms. The LANG environment variable has three components: lang, \_territory, and .codeset. Use %l for the lang component (the language being used), %t for the \_territory component (used to specify the geographic area or native dialect of the language being used), and %c for the .codeset component (used to specify the multinational character set used to represent the language).

%N      Name

The name of the target file. This field is used in different ways by different preference variables:

|              |  |
|--------------|--|
| NLSPATH      | The name of the application executable.  |
| OUIDICTPATH  | The name of the application executable.  |
| OUIHELPPATH  | The name of the current module.          |
| OUIIMAGEPATH | The name of the image file to be opened. |
| OUIPREFSPATH | The name of the application executable.  |

%P      Preference file pathname

The pathname of the directory in which the current preference file resides. This value should not be used outside preference files.

%S      Default suffix

The default suffix for preference files. The platform-specific default suffixes are:

For Windows:      .ini

%U      User preferences

The pathname of the location of the current user's preference variables. For Windows: The Windows directory.



%V      Version

The major version number of the current OpenUI release.

%W      Working directory

The pathname of the current working directory.

%X      Extra

The value that this preferences variable would have if this specification had not been found. This is useful for adding values to existing preference variables. For example, using Microsoft Windows, if you want to add \mydict to the dictionary path in your user preference file, but still locate all the usual directories on the path specified in the system preference file, add the entry OUIDICTPATH=\mydict;%X to your user preference file. %X must occur at the end of the line. Any specifications occurring after %X are ignored.

Back to Start of Tutorial on the [Preference File](#).

## Application preference files

You can specify application-specific preference variables in an application-specific preference file. There may be a system-wide preferences file for a particular application, or end-users may have their own, customized preference file for an application. Refer to the discussion on the default search order for preferences files to determine how application preference files are named, where they are located, and how preference variables will be set.

Back to Start of Tutorial on the [Preference File](#).

## Default search order

The default search order for preference files is:

- 1.) An end-user's application-specific preference file
- 2.) A system-wide application-specific preference file
- 3.) An end-user's general OpenUI preference file
- 4.) A system-wide general OpenUI preference file

The location of these preference files varies from platform to platform.

For Windows the default search order for preference files is:

- 1.) In the Windows directory, <application>.ini, where <application> is the name of the application to be started.
- 2.) In the application directory, <application>.ini, where <application> is the name of the application to be started.
- 3.) In the Windows directory, openui<version>.ini, where <version> is the current OpenUI release number.
- 4.) In the application directory, openui<version>.ini, where <version> is the current OpenUI release number.

Back to Start of Tutorial on the [Preference File](#).

## About ...

---

DOCsys ist ein Produkt von

# Applied Science

## Ingenieurbüro Dieter Scholz

---

Prof. Dr.-Ing. Dieter Scholz, MSME  
Seevering 53  
21629 Neu Wulmstorf

Tel: 040 - 409 716 46  
Fax: 040 - 409 716 47

E-mail: [Scholz@fzt.fh-hamburg.de](mailto:Scholz@fzt.fh-hamburg.de)  
WWW: <http://www.fh-hamburg.de/pers/Scholz/AppliedScience/>