

 **DESITE BIM**  
md pro

 **DESITE BIM**  
md

 **DESITE BIM**  
custom

 **DESITE BIM**  
share pro

 **DESITE BIM**  
share

 **DESITE BIM**  
touch

Managen Sie Ihr  
3D-Bauwerksmodell interaktiv.  
**Von Anfang an.**

**thinkproject**

# DESITE-Konfiguration mit Kommandozeilenargumenten

thinkproject



Alle Desktop-Programme der **DESITE**-Produktfamilie (md, md pro, custom, share, share pro) können mittels Kommandozeilenargumenten (command line switches) konfiguriert werden. Das bedeutet, dass beim Programmstart bestimmte Parameter an die ausführbare Datei (\*.exe) übergeben werden können.

Beispiele:

```
desitemd.exe "D:\Projects\SampleProject\mySampleProject.pfs"
```

```
desitemd.exe -srvPort 45456 -cltPort 45455 -lang "en"
```

Das erste Beispiel startet **DESITE** md und lädt sofort das angegebene Projekt. Auf diese Weise Dateien an Anwendungen zu übergeben, wird von so gut wie allen Windows-Programmen unterstützt. Genau dasselbe passiert übrigens, wenn Sie eine PFS-Datei doppelklicken – natürlich vorausgesetzt, dass **DESITE** md auf Ihrem Computer installiert ist.

Das zweite Beispiel ist **DESITE**-spezifischer. Es konfiguriert die TCP-Ports für DESITEs Remote-Schnittstelle. Außerdem wird Englisch als Oberflächen-Sprache festgelegt.

Diese Anleitung beschreibt, welche Parameter es gibt, was sie bewirken, wie sie an die Anwendung übergeben werden, und welche Unterschiede es zwischen den **DESITE**-Anwendungen zu beachten gibt.

## 1. Parameter-Übergabe

1.1 als Kommandozeilenargument

1.2 als Konfigurationsdatei

## 2. Parameter-Referenz

2.1 file

2.2 srvPort, cltPort

2.3 showWidget

2.4 webFormLocations

2.5 unlockWebForms

2.6 lang

2.7 scriptDebug

## 3. Unterstützung in **DESITE**-Anwendungen

**thinkproject**

# 1. Parameter-Übergabe

**thinkproject**



Um Parameter via Kommandozeile zu übergeben, werden sie mit Bindestrich (-) hinter den Programmnamen (.exe) geschrieben.

Beispiele:

```
desitecustom.exe -lang "en"
```

```
desitecustom.exe -srvPort 45456
```

```
desitecustom.exe -unlockWebForms
```

*(Die Bedeutung dieser Parameter interessiert an dieser Stelle noch nicht, das wird in der Parameter-Referenz behandelt.)*

Manche Parameter benötigen einen Wert (wie hier lang und srvPort), manche funktionieren wie einfache Schalter ohne Wert (wie hier unlockWebForms). Wenn ein Wert benötigt wird, wird er hinter dem Parameter mit einem einzelnen Leerzeichen als Trenner angegeben.

Beachten Sie die Datentypen der Werte. Im Beispiel steht "en" in Anführungszeichen, weil es eine Zeichenkette (string) ist, während 45456 ohne Anführungszeichen angegeben wird, da es sich um eine Port-Nummer handelt.

Die Argumente können natürlich auch kombiniert werden:

```
desitecustom.exe -lang "en" -srvPort 45456 -unlockWebForms
```

Alternativ können die Parameter in eine Konfigurationsdatei (JSON-Format) geschrieben und mittels `-configFile` an die Anwendung übergeben werden:

```
desitecustom.exe -configFile "E:\somePath\desiteConfig.json"
```

Die Konfigurationsdatei könnte etwa folgendermaßen aussehen:

```
{  
  "lang": "en",  
  "srvPort": 45456,  
  "unlockWebForms": true  
}
```

Dadurch würden dieselben Optionen gesetzt, wie wir es auf der vorigen Seite mit Kommandozeilenargumenten gemacht haben. Die JSON-Keys lauten 1:1 wie die Kommandozeilenargumente.

Ein wichtiger Unterschied besteht allerdings bei `unlockWebForms`: Als Kommandozeilenargument wird für diesen Schalter kein Wert angegeben, in JSON ist der Wert `true`.

Wichtig: Wenn `-configFile` verwendet wird, werden alle weiteren evtl. angegebenen Kommandozeilenparameter ignoriert. Dies dient zur Vermeidung von Mehrdeutigkeiten bzw. Konflikten.

## 2. Parameter-Referenz

**thinkproject**





Laden Sie ein Projekt oder Modell mit `-file "filepath"`

Unterstützte **DESITE**-Anwendungen: md, md pro, custom, share, share pro (mit den jeweils unterst. Modellformaten)

Beispiel:

```
desitemd.exe -file "D:\Projects\SampleProject\mySampleProject.pfs"
```

Hinweis: Dies bewirkt dasselbe, wie wenn der Dateiname direkt übergeben wird:

```
desitemd.exe "D:\Projects\SampleProject\mySampleProject.pfs"
```

Setzen Sie die TCP-Ports für die Remote-Schnittstelle mit `-srvPort Number` und/oder `-cltPort Number`

Unterstützte **DESITE**-Anwendungen: md, md pro, custom, share, share pro

Beispiel:

```
desiteshare.exe -srvPort 45456 -cltPort 45455
```

**DESITE**-Anwendungen können Nachrichten über lokale TCP-Sockets senden und empfangen, um mit anderen Anwendungen zu kommunizieren.

Standardmäßig ist der Server-Port (listener) 45456 und der Client-Port (sender) 45455. Das Ändern dieser Werte kann notwendig sein, falls die Standard-Ports in Ihrer Netzwerk-Umgebung nicht zur Verfügung stehen, oder falls mehrere Instanzen von **DESITE**-Anwendungen gleichzeitig laufen.

Mehr Details zur **DESITE**-Remote-Schnittstelle finden Sie in der entsprechenden Dokumentation.

Blenden Sie beim Programmstart ein Widget oder Fenster ein mit `-showWidget`

Unterstützte **DESITE**-Anwendungen: md, md pro, custom, share\*, share pro\*

Beispiel:

```
desitecustom.exe -showWidget WebForms
```

Verwenden Sie diesen Schalter mit einem der folgenden Parameter, um das entsprechende Widget anzuzeigen:

- WebForms
- Issues (alternativ: Viewpoints)
- ObjectInfo (alternativ: DataSheet)
- ProjectStructure

Hinweis: Hierdurch wird nicht zwingend sichergestellt, dass das Fenster sichtbar im Vordergrund erscheint. Je nach der letzten gespeicherten Oberflächen-Konfiguration könnte es auch in einem nicht aktiven Tab liegen.

Dieser Schalter ersetzt `-showWebForms`. Dieser funktioniert weiterhin, doch zu bevorzugen ist `-showWidget WebForms`.

\* eingeschränkt, z.B. keine WebForms

Definieren Sie Formular-Quellen mit `-webFormLocations "url1[url2]"`

Unterstützte **DESITE**-Anwendungen: md, md pro, custom

Beispiel:

```
desitemd.exe -webFormLocations "https://www.thinkproject.com"
```

Seit Version 2.6 unterstützt **DESITE** neben der traditionellen lokalen `DbInputForm.html` auch andere Quellen als Startseite für die Formulare.

Das kann insbesondere dann nützlich sein, wenn Sie die gleichen Formulare in vielen Projekten benötigen.

Sie können mehrere URLs angeben. Setzen Sie bei Verwendung des Kommandozeilenarguments ein Semikolon als Trenner. Kommt eine Konfigurationsdatei zum Einsatz, verwenden Sie die JSON-Listennotation (auch bei nur einer URL):

```
{  
  "webFormLocations":["https://www.thinkproject.com"]  
}
```

Erteilen Sie eine allgemeine WebForms-Freigabe mit `-unlockWebForms`

Unterstützte **DESITE**-Anwendungen: md, md pro, custom

Beispiel:

```
desitecustom.exe -unlockWebForms
```

Verwenden Sie diesen Schalter, wenn Sie verhindern wollen, dass die Benutzung von WebForms und anderen in **DESITE** eingebetteten Browsern explizit freigeschaltet werden muss - ein Sicherheitsfeature, welches mit Version 2.6.1 eingeführt wurde.

Hinweis: Aus Sicherheitsgründen raten wir von der routinemäßigen Verwendung dieses Schalters ab. Er sollte nur bei der WebForms-Entwicklung zum Einsatz kommen, wenn die Abfrage sonst nur stören würde, oder wenn Sie wirklich sicher sind, nur Projekte aus vertrauenswürdigen Quellen zu öffnen.

Setzen Sie die Sprache mit `-lang "langCode"`

Unterstützte **DESITE**-Anwendungen: md, md pro, custom, share, share pro

Beispiel:

```
desiteshare.exe -lang "en"
```

Die so gesetzte Spracheinstellung hat Vorrang vor der entsprechenden Einstellung in der Anwendungskonfiguration, überschreibt diese jedoch nicht (override, nicht overwrite).

Wenn also beispielsweise in den Programmoptionen als Sprache Englisch eingestellt ist, und Sie starten mit dem Parameter `-lang "de"`, wird für diese Session die Oberfläche in deutscher Sprache lokalisiert. Starten Sie aber dann die Anwendung noch einmal ohne Parameter, wird wieder die Englisch-Einstellung angewendet.

Gültige Werte (**DESITE** 2.6): en, de, fr, it, es, nl, pl.

Aktivieren Sie den JavaScript/WebForms-Debugger mit `-scriptDebug [port]`

Unterstützte **DESITE**-Anwendungen: md, md pro

Beispiel:

```
desitemd.exe -scriptDebug
```

Wenn Script-Debugging aktiv ist, lösen JavaScript-Fehler in Makros ein Debug-Fenster aus, in dem Sie lokale Variablen einsehen oder Haltepunkte setzen können, etc.

Außerdem ist Remote-Debugging von WebForms möglich. Starten Sie dazu einen separaten Chromium-basierten Browser (wie Google Chrome oder Microsoft Edge) und navigieren Sie zu folgender Adresse:

<http://localhost:1000>.

Wählen Sie die zu untersuchende WebForm unter den "Inspectable Pages" aus. Anschließend können Sie Ihren Code mit den ganz normalen Chromium-Entwicklertools analysieren und debuggen.

Der Standard-Port für den Remote-Debugger ist 1000. Sie können aber einen anderen Port als Parameter übergeben, wie z.B. `"-scriptDebug 1234"` und die entsprechende Adresse: <http://localhost:1234>.

Bitte beachten Sie, dass `-scriptDebug` sich nachteilig auf die Leistung auswirken kann. Sie sollten die Funktion daher nur während der Entwicklung bzw. während des Testens verwenden, nicht im produktiven Einsatz.

# 3. Unterstützung in DESITE-Anwendungen

thinkproject





	md pro	md	custom	share pro	share
file	✓	✓	✓	✓	✓
srvPort, cltPort	✓	✓	✓	✓	✓
showWidget	✓	✓	✓	✓	✓
webFormLocations	✓	✓	✓	(keine Webforms)	(keine Webforms)
unlockWebForms	✓	✓	✓	(keine Webforms)	(keine Webforms)
lang	✓	✓	✓	✓	✓
scriptDebug	✓	✓	✗	✗	✗