

# REVERSE-ENGINEERING DER BASISKLASSEN FÜR DIE BENUTZEROBERFLÄCHE

---

Dokument zur Studienarbeit  
- Designphase -

Autoren	Thomas Kullmann, Günther Reinecker
Dokumentversion	1.2
Zustand	abgeschlossen
letzte Bearbeitung	06.06.02



**Inhalt**

<b>I</b>	<b>ÜBERBLICK</b> .....	<b>2</b>
<b>II</b>	<b>KLASSE TModalDLG</b> .....	<b>3</b>
<b>II.1</b>	<b>Attribute</b> .....	<b>3</b>
<b>II.2</b>	<b>Methoden</b> .....	<b>4</b>
<b>II.3</b>	<b>Bewertung</b> .....	<b>6</b>
<b>III</b>	<b>KLASSE TMODELESSDLG</b> .....	<b>7</b>
<b>III.1</b>	<b>Attribute</b> .....	<b>7</b>
<b>III.2</b>	<b>Methoden</b> .....	<b>7</b>
<b>III.3</b>	<b>Bewertung</b> .....	<b>9</b>
<b>IV</b>	<b>ATTRIBUTE</b> .....	<b>9</b>
<b>V</b>	<b>METHODEN</b> .....	<b>10</b>
<b>VI</b>	<b>ANHANG</b> .....	<b>10</b>
<b>VI.1</b>	<b>Verwandte Dokumente</b> .....	<b>10</b>
<b>VI.2</b>	<b>Tabellen</b> .....	<b>10</b>
<b>VI.3</b>	<b>Abbildungen</b> .....	<b>10</b>

**I Überblick**

Die Dokumentation wurde stark formalisiert, für die Layoutkonventionen siehe [1]. Die Klasse TModalDlg bietet als Basisklasse die Funktionalität zum Verwalten von Dialogfenstern. Man kann von ihr eine neue Dialogklasse ableiten und diese spezialisieren um sie an die eigenen Bedürfnisse anzupassen. Das Design der neuen Klasse ist damit durch die Basisklasse vorgegeben. Da dort fast die gesamte Nachrichtenbehandlung nur über eine Methode läuft, sind die Entwurfsmöglichkeiten von neuen Dialogklassen nicht sehr groß. Dadurch besitzen jedoch alle Dialogfenster einen ähnlichen Aufbau, was Reverse-Engineering Prozesse erheblich vereinfacht.

Weiterhin werden in der Basisklasse TModalDlg sehr viele ältere 16bit-Funktionen zur Steuerung von Dialogfenster benutzt. Da das Projekt noch dauerhaft auf 16bit-Systemen (Windows® 9.x, Me) lauffähig sein soll, ist dies auch in jedem Fall nötig. Aber im Hinblick auf eine Nutzung unter 32bit-Windows®-Versionen (NT, 2000, XP, ...), wären diese unbedingt auszutauschen.

**► Dokumentation des Istzustands**

Die Dokumentation bezieht sich auf den Quellcode der unten aufgelisteten Dateien (Stand: 24.04.2002). Nachfolgende Änderungen oder Neuimplementationen, können nicht berücksichtigt werden!

h-Dateien (Deklaration)	cpp-Dateien (Implementation)
• SWINTRAC.H	• DLG_TPL.CPP

Tabelle 1 „Auflistung der zum Subsystem zugehörigen Dateien“ (Quelle: selbst)



## II Klasse TModalDlg

Deklaration : SWINTRAC.H

Implementation : DLG\_TPL.CPP

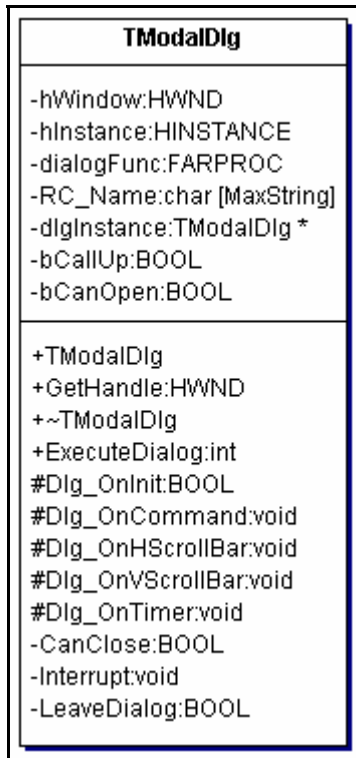


Abbildung 1 „UML-Klassendiagramm der Klasse TModalDlg“ (Quelle: Together®, Version 6.0)

### I.2 Attribute

► **TModalDlg\* dlgInstance** **PRIVATE**

speichert den Zeiger des modalen Dialogfensters, das sich derzeit im Vordergrund des XCTRL-Programms befindet (siehe `TheDialog`); Wenn derzeit kein modales Dialogfenster angezeigt wird, bleibt `dlgInstance` uninitialisiert (dieser Zustand wird mit `bCallUp == FALSE` gekennzeichnet). Nach dem Schließen dieses Dialogfensters wird der Zeiger wiederhergestellt (nur bei `bCallUp == TRUE`). wird im Konstruktor initialisiert

► **BOOL bCallUp** **PRIVATE**

gibt an, dass bereits ein modales Dialogfenster, vor diesem Fenster, angezeigt wurde; wird im Konstruktor initialisiert

**Redundant, `dlgInstance == 0` könnte kennzeichnen, dass kein Dialogfenster angezeigt wird**

► **FARPROC dialogFunc** **PRIVATE**

ist der Zeiger auf die CALLBACK-Funktion `DialogProc`; wird im Konstruktor initialisiert

► **HWND hWindow** **PRIVATE**

Fensterhandle, das das Fenster eindeutig identifiziert; wird bei Methode `OnInit` mit dem Wert des ersten Parameters initialisiert

**▶ HINSTANCE hInstance****PRIVATE**

enthält die ausführende Instanz der Anwendung; Die Instanz muss Windows beim Laden von Ressourcen als Parameter übergeben werden (aus [2]). wird im Konstruktor initialisiert

**▶ char RC\_Name[ MaxString ]****PRIVATE**

ist der Name der Ressource für das Dialogfenster; wird im Konstruktor mit dem Wert des ersten Parameters initialisiert

**▶ BOOL bCanOpen****PRIVATE**

wird nicht verwendet

## I.2 Methoden

---

**▶ TModalDlg( char\* )****PUBLIC**

Im Konstruktor wird eine Vielzahl der Attribute initialisiert (siehe I.0 Attribute). Der Parameter ist der Name der Ressource für das anzuzeigende Dialogfenster. Wenn bereits eine Instanz von TModalDlg existiert, wird dlgInstance das existierende Dialogobjekt zugewiesen.

**▶ virtual ~TModalDlg( void )****PUBLIC**

Der Standarddestruktor gibt den Zeiger auf die CALLBACK-Funktion wieder frei und stellt TheDialog wieder her.

**▶ virtual int ExecuteDialog( HWND )****PUBLIC**

erzeugt das Dialogfenster und zeigt es an; Der Rückgabewert kann zum Schließen des erzeugten Dialogfensters verwendet werden.

**▶ HWND GetHandle( void )****PUBLIC**

gibt den Inhalt von hWnd zurück

**▶ virtual void Dlg\_OnCommand( HWND, int,  
HWND, UINT )****PROTECTED**

wird zur Behandlung der Windowsbotschaften eingesetzt, die nicht explizit anderen Methoden<sup>1</sup> zugeordnet sind (diese Zuordnung findet bei DialogProc für ausgewählte Botschaften statt); Der erste Parameter ist ein Handle auf das Fenster, das die Botschaft ausgelöst hat, der Zweite ist die Kennung der Botschaft und die letzten beiden Parameter werden botschaftenspezifisch belegt (siehe [2]).

Hier werden bereits: IDOK, IDCANCEL und IDABORT behandelt, sie veranlassen alle das Schließen des Fensters.

**▶ virtual BOOL CanClose( void )****PRIVATE**

Der Rückgabewert gibt an, dass das Dialogfenster bei Windowsbotschaft IDOK geschlossen werden darf. gibt hier stets TRUE zurück

---

<sup>1</sup> sind Member von TModalDlg und besitzen meist das Suffix „Dlg\_On“



▶ **virtual BOOL LeaveDialog( void )** **PRIVATE**

wird in Methode Dlg\_OnCommand als Reaktion auf die dort behandelten Botschaften benutzt → Aufruf erfolgt vor dem Schließen des Fensters; gibt stets TRUE zurück

**Der Rückgabewert wird nie ausgewertet: Rückgabewert void ausreichend.**

▶ **virtual BOOL Dlg\_OnInit( HWND, HWND, LPARAM )** **PROTECTED**

wird bei der Initialisierung des Fensters durch WM\_INITDIALOG aufgerufen; hWnd wird initialisiert

▶ **virtual void DlgOnHScrollBar( HWND, HWND, UINT, int )** **PROTECTED**

wird aufgerufen sobald eine Bildlauf-Benachrichtigungsbotschaft einer horizontalen Bildlaufleiste ausgelöst wurde; ist leer implementiert

▶ **virtual void DlgOnVScrollBar( HWND, HWND, UINT, int )** **PROTECTED**

wie Methode DlgOnHScrollBar, nur für vertikale Bildlaufleisten; ist ebenfalls leer implementiert

▶ **virtual void Dlg\_OnTimer( HWND, UINT )** **PROTECTED**

diese Methode wird aufgerufen, sobald ein Timerereignis aufgetreten ist; ist leer implementiert

▶ **virtual void Interrupt( void )** **PRIVATE**

wird aufgerufen; wenn das spezielle Tastaturereignis VK\_ESCAPE ausgelöst wurde; ist nicht weiter implementiert



### III.0 Bewertung

Metrik	Kennung (min, max)	Wert
Klasse		
,Lines Of Code' inkl. Leer- und Kommentarzeilen	LOC (0, 1000)	141
,LOC of Implementation'*	LOCI	110
,LOC of Declaration'*	LOCD	31
,Number Of Attributes'	NOA (0, 30)	7
,Number Of Operations'	NOO (0, 50)	16
,Number Of Members' Attribute + Methoden	NOM = NOA + NOO	23
,Number Of Constructors'	NOCON (0, 5)	1
,Number Of Overridden Methods'	NOOM (0, 10)	0
,Percentage of Private Members'	PPrivM	40
,Percentage of Protected Members'	PProtM (0, 10)	44
,Percentage of Public Members'	PPubM	16
,Weighted Methods Per Class'	WMPC1 (0, 30)	21
Attribute		
,Attribute Complexity'	AC	57
Methoden		
,Maximum Number Of Parameters'	MNOP (0, 4)	<b>5</b>
,Cyclomatic Complexity'	CC	21
Kommentare		
,Number Of Comments'*	NOC	3
,True Comment Ratio'	TCR (5, 400)	7

**Tabelle 2** „ausgewählte Metriken der Klasse TModalDLG“ (Quelle: Together<sup>®</sup>, Version 6.0)

\* Diese Metriken sind nicht Bestandteil von Together, sondern wurden manuell ermittelt.



## I Klasse TModelessDlg

Deklaration : SWINTRAC.H

Implementation : DLG\_TPL.CPP

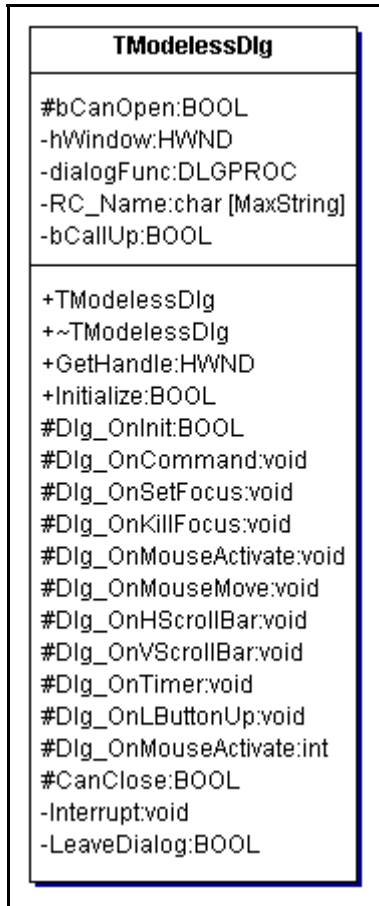


Abbildung 2 „UML-Klassendiagramm der Klasse TModelessDlg“ (Quelle: Together<sup>®</sup>, Version 6.0)

### I.2 Attribute

---

▶ TModelessDlg* dlgInstance	PRIVATE
▶ BOOL bCallUp	PRIVATE
▶ DLGPROC dialogFunc	PRIVATE
▶ HWND hWindow	PRIVATE
▶ char RC_Name[ MaxString ]	PRIVATE

### I.2 Methoden

---

▶ TModelessDlg( char* )	PUBLIC
▶ virtual ~TModelessDlg( void )	PUBLIC
▶ HWND GetHandle( void )	PUBLIC



▶	BOOL	Initialize( void )	PUBLIC
▶	virtual void	Dlg_OnCommand( HWND, int, HWND, UINT )	PROTECTED
▶	virtual BOOL	Dlg_OnInit( HWND, HWND, LPARAM )	PROTECTED
▶	virtual void	DlgOnHScrollBar( HWND, HWND, UINT, int )	PROTECTED
▶	virtual void	DlgOnVScrollBar( HWND, HWND, UINT, int )	PROTECTED
▶	virtual void	Dlg_OnSetFocus( HWND, HWND )	PROTECTED
▶	virtual void	Dlg_OnKillFocus( HWND, HWND )	PROTECTED
▶	virtual void	Dlg_OnMouseActivate( HWND, int, HWND, UINT )	PROTECTED
▶	virtual void	Dlg_OnMouseMove( HWND, int, HWND, UINT )	PROTECTED
▶	virtual int	Dlg_OnMouseActivate( HWND, HWND, UINT,	PROTECTED
▶	virtual void	Dlg_OnLButtonUp( HWND, int, int, UINT )	PROTECTED
▶	virtual void	Dlg_OnTimer( HWND, UINT )	PROTECTED
▶	virtual BOOL	CanClose( void )	PROTECTED
▶	virtual void	Interrupt( void )	PRIVATE
▶	virtual BOOL	LeaveDialog( void )	PRIVATE



**III.0 Bewertung**

Metrik		Kennung (min, max)	Wert
Klasse			
,Lines Of Code' inkl. Leer- und Kommentarzeilen	LOC (0, 1000)		126
,LOC of Implementation'*	LOCI		84
,LOC of Declaration'*	LOCD		42
,Number Of Attributes'	NOA (0, 30)		6
,Number Of Operations'	NOO (0, 50)		17
,Number Of Members' Attribute + Methoden	NOM = NOA + NOO		23
,Number Of Constructors'	NOCON (0, 5)		1
,Number Of Overridden Methods'	NOOM (0, 10)		0
,Percentage of Private Members'	PPrivM		20
,Percentage of Protected Members'	PProtM (0, 10)		52
,Percentage of Public Members'	PPubM		20
,Weighted Methods Per Class'	WMPC1 (0, 30)		<b>21</b>
Attribute			
,Attribute Complexity'	AC		48
Methoden			
,Maximum Number Of Parameters'	MNOP (0, 4)		<b>4</b>
,Cyclomatic Complexity'	CC		21
Kommentare			
,Number Of Comments'*	NOC		3
,True Comment Ratio'	TCR (5, 400)		6

**Tabelle 3** „ausgewählte Metriken der Klasse TModelessDlg“ (Quelle: Together<sup>®</sup>, Version 6.0)

\* Diese Metriken sind nicht Bestandteil von Together, sondern wurden manuell ermittelt.

**I Attribute**

► **TModalDlg\* TheDialog = NULL** **GLOBAL**

Deklaration des aktiven, modalen Dialogobjektes; die CALLBACK-Funktion DialogProc leitet ausgewählte Windows-Botschaften an Membermethode dieses Objekts weiter; wird in Methode TModalDlg::ExecuteDialog initialisiert

► **extern HINSTANCE hModuleInstance** **GLOBAL**

um auf das in M\_MAIN.CPP definierte Attribut auch in DLG\_TPL.CPP zugreifen zu können; ist die eindeutige Kennung der Instanz der Anwendung

► **TModelessDlg\* TheModeless = NULL** **GLOBAL**



## V Methoden

► **BOOL CALLBACK DialogProc( HWND, UINT, WPARAM, LPARAM ) GLOBAL**

statische CALLBACK-Funktion zur Behandlung der Windowsbotschaften des aktiven, modalen Dialogfensters; der erste Parameter ist das Handle des Dialogfensters; der Zweite ist die Kennung der Windows-Botschaft; der Dritte und Vierte werden botschaftenspezifisch belegt (siehe [2]); gibt TRUE zurück, wenn eine Botschaft behandelt werden konnte, ansonsten FALSE

Zuordnung Botschaften → Methodenaufrufe der Klasse TModalDlg:

WM\_COMMAND → Dlg\_OnCommand(...)  
 WM\_INITDIALOG → Dlg\_OnInit(...)  
 WM\_HSCROLL → Dlg\_OnHScrollBar(...)  
 WM\_VSCROLL → Dlg\_OnVScrollBar(...)  
 WM\_TIMER → Dlg\_OnTimer( void )  
 WM\_CHAR → Dlg\_Interrupt( void )

► **BOOL CALLBACK ModelessProc( HWND, UINT, WPARAM, LPARAM ) GLOBAL**

## VI Anhang

### I.0 Verwandte Dokumente

- [1] „Layoutkonventionen und Steuerelemente“, Version 1.0 von Thomas Kullmann und Günther Reinecker  
 [2] „Borland 5.02 Hilfe“

### II.0 Tabellen

TABELLE 1 „AUFLISTUNG DER ZUM SUBSYSTEM ZUGEHÖRIGEN DATEIEN“ (QUELLE: SELBST) .....	2
TABELLE 2 „AUSGEWÄHLTE METRIKEN DER KLASSE TMODALDLG“ (QUELLE: TOGETHER®, VERSION 6.0) .....	6
TABELLE 3 „AUSGEWÄHLTE METRIKEN DER KLASSE TMODELESSDLG“ (QUELLE: TOGETHER®, VERSION 6.0) .....	9

### III.0 Abbildungen

ABBILDUNG 1 „UML-KLASSENDIAGRAMM DER KLASSE TMODALDLG“ (QUELLE: TOGETHER®, VERSION 6.0) .....	3
ABBILDUNG 2 „UML-KLASSENDIAGRAMM DER KLASSE TMODELESSDLG“ (QUELLE: TOGETHER®, VERSION 6.0) .....	7