

Wie... verändere ich den Code meiner Anwendung?

Voraussetzungen

- Installiertes ClassiX[®] System
- Codewright oder ein anderer Texteditor
- Kenntnisse über:

Grundlegender Aufbau von ClassiX[®]	Widget
Konventionen	Action List
Klassen	Event
Vererbung (Inheritance)	Message
Modul	

Einführung/Aufgabenstellung

Hier lernen wir ein Werkzeug kennen, mit dem unsere ClassiX[®] Anwendung mit Hilfe von InstantView[®] zu verändert werden kann. InstantView[®] bietet ein Monitor-Fenster, welches Sie weitreichend unterstützt. Angeschlossen an das Monitorfenster ist das Hauptwerkzeug zur Entwicklung: ein Text Editor. Für größere Entwicklungen empfehlen wir den CodeWright Editor, der von uns standardmäßig mit InstantView[®] verknüpft wurde.

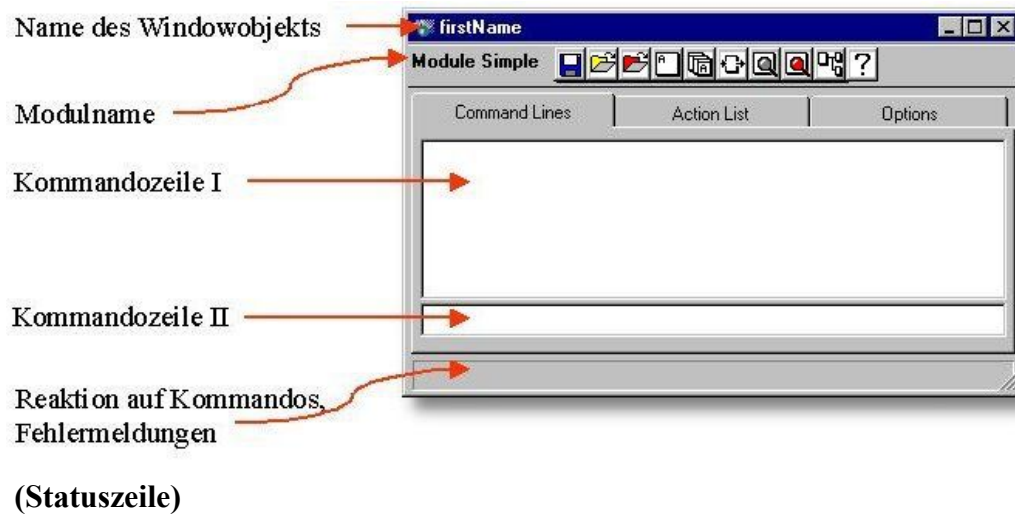
Inhalt

- Das Monitor-Fenster oder Request-Window öffnen
- Aufbau des Fensters
- Buttons und Kommandos
- Ausführen von Code
- 2. Reiter - Action List
- Veränderungen eines Moduls
- Zusammenfassung

Das Monitor-Fenster oder Request-Window öffnen










Sie öffnen das Monitor-Fenster, indem Sie die ALT-Taste halten, während Sie die rechte Maustaste auf einem Element (Widget) eines ClassiX[®]-Windows betätigen.

Der Aufbau des Fenster



Achtung:
Der angezeigte Modulname zeigt auf das aktuelle Objekt. Wenn es sich um ein *abgeleitetes* Modul handelt, in dem der Code nicht steht, dann wird das Modul aufgerufen, aus dem unser abgeleitetes Modul den Code *geerbt* hat.

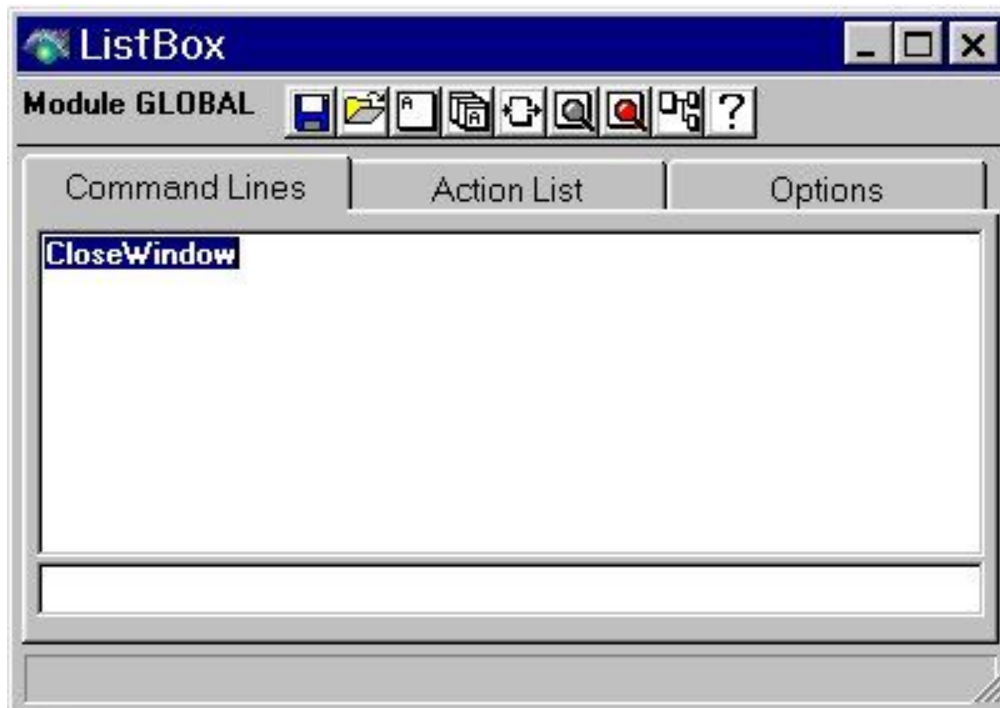
Buttons und Kommandos

- Die Buttons im Toolbar haben folgende Bedeutung:
 -  speichert das aktuelle Modul
 -  lädt das aktuelle Modul
 -  lädt das oberste der Basismodule des aktuellen Moduls inkl. aller Ableitungen
 -  ruft über einen Editor den Source-Code des ausgewählten Windowobjekts auf.
 -  ruft über einen Editor den Source-Code des 1. Auftretens des Windowobjekts auf.
 -  macht die Änderungen in der (edierten) Aktionsliste wirksam
 -  startet den (einfachen) InstantView®-**Debugger**
 -  aktiviert CodeWright als InstantView®-**Debugger**
 -  öffnet ein Window mit einer Übersicht sämtlicher Module des Projekts (Browser)
- In den Kommandozeilen können InstantView®-Anweisungen sofort ausgeführt werden.
 - Kommandozeile I ist für mehrzeilige Eingaben geeignet. mit <Enter> wird in die nächste Zeile gesprungen. Die Ausführung startet nach <Ctrl><Eingabe>. ESC löscht die Kommandozeilen.

- Kommandozeile II ist für einzeilige Eingaben gedacht - mit <Enter> wird in die Ausführung gestartet.
Vorher eingegebene Kommandos können mit Cursor Up/Down wiederholt werden.
ESC löscht die Kommandozeile.
- Mit F7 erhält man eine Übersicht der letzten 64 Kommandos.
- Mit F1 erreicht man die Dokumentation des InstantView®-Sprachelements, auf dem sich der Cursor befindet; dies gilt innerhalb der Kommandozeilen als auch für die Aktionsliste.
- Die Aktionsliste kann ediert werden.

Ausführen von Code

InstantView® Anweisungen (Statements) können direkt in die Kommandozeilen eingegeben werden und interaktiv ausgeführt werden.



`CloseWindow` schließt das Fenster, aus dem das Monitor-Fenster aufgerufen wurde, wenn die Anweisung mit <STRG><Enter> abgeschickt wird.

2. Reiter - Action List

Der Reiter "Action List" zeigt die im Code des [Moduls](#) vorhandene [Action List](#) des gerade bearbeiteten [Widget](#).

Die Action List beschreibt das dynamische Verhalten des Windowobjekts. Dieses Verhalten wird interaktiv durch Ereignisse gesteuert ([event driven](#)). Die Auslösung eines event erfolgt immer über eine Nachricht ([message](#)).

Veränderung eines Moduls

In der Kommandozeile II definieren wir zunächst eine Nachricht:

Syntax:

Message(MSG1, MSG2, ..., MSGn)

oder Kurzform


Msg(MSG1, MSG2, ..., MSGn)

MSG1...MSGn: Messagenamen

Mit Message (Kurzform Msg) soll der Identifikator "TRANSNORM" als Message-Name definiert werden:

Msg(TRANSNORM)

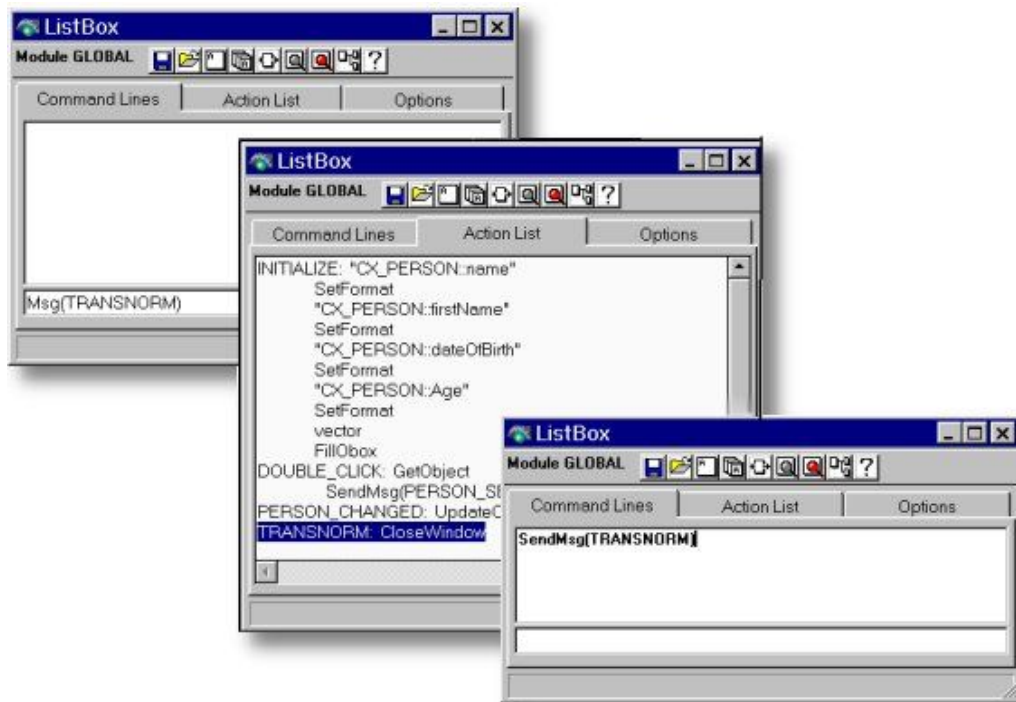
<Enter> führt das Statement aus. In der Statuszeile erscheint bei korrekter Syntax die Meldung ok.

Dann wird zu dem Reiter "Action List" gewechselt. Am Ende der Liste wird dem Message-Namen "TRANSNORM" eine Aktion zugewiesen, in diesem Fall die Funktion CloseWindow. Der Button  macht die Änderungen in der Action List wirksam.

Wird in einer der Kommandozeilen die Message "TRANSNORM" gesendet, wird das Anwendungsfenster geschlossen:

SendMsg(TRANSNORM)

<Enter>



Bemerkung: Die gemachten Veränderungen in der Action List gehen beim Beenden von ClassiX® verloren. Für bleibende Änderungen im Code siehe ...

Problembhebung

Gab es Probleme? Wurde statt o.k. ein Syntaxfehler angezeigt? Dann wurden die Konventionen für den InstantView® Code nicht eingehalten: vielleicht wurde statt `SendMsg(TRANSNORM)` - `sendmsg(transnorm)` geschrieben. Die Groß- und Kleinschreibung muss beachtet werden. Bitte noch einmal „**Wie... lauten die Code-Konventionen in meiner Anwendung**“ ansehen!

Zusammenfassung

Wir haben gelernt, wie ein Tool zur Veränderung unserer ClassiX® Anwendung zu öffnen und zu bedienen ist. Zusätzlich lernten wir eine Message zu definieren, ihr eine Aktion zuzuweisen und über die Message die Ausführung der Aktion auszulösen. Jetzt wissen wir, dass die Auslösung eines Events grundsätzlich über eine Message erfolgt.

Weiterführendes

Wie... wird eine Spalte in eine Listbox eingefügt?

Wie... funktioniert Vererbung in ClassiX®?

Wie... findet und verändert man abgeleiteten (vererbten) Code?