



Pflichtenheft

Autor des Dokuments	Thomas Klepzig, Robert Schulze	Erstellt am	05.01.10
Dateiname	Pflichtenheft_Cassiopeia.pdf		
Seitenanzahl	18	SDDR	Vertraulich!

Historie der Dokumentversionen

Version	Datum	Autor	Änderungsgrund/Bemerkungen
0.1	05.01.10	Thomas Klepzig	Ersterstellung
0.2	13.01.10	Robert Schulze	Diagramme und Erläuterungen dazu
0.3	13.01.10	Thomas Klepzig	Technische Beschreibung
1.0	13.01.10	Thomas Klepzig	Fertigstellung

Tabelle 1: Historie der Dokumentenversionen

Inhaltsverzeichnis

Historie der Dokumentversionen.....	2
Inhaltsverzeichnis	2
1. Einleitung.....	4
1.1 Allgemeines.....	4
1.1.1 Zweck und Ziel dieses Dokuments.....	4
1.1.2 Projektbezug.....	4
1.1.3 Abkürzungen.....	4
1.2 Verteiler und Freigabe.....	5
1.2.1 Verteiler für das Pflichtenheft.....	5
2. Konzept und Rahmenbedingungen.....	5
2.1 Ziele des Anbieters.....	5
2.2 Ziele und Nutzen des Anwenders.....	5
2.3 Benutzer / Zielgruppe.....	7
2.4 Systemvoraussetzungen.....	7
2.5 Übersicht der Meilensteine.....	8
3. Technische Beschreibung.....	9
3.1 Lebenszyklus.....	9
3.2 Grafische Darstellung der Vorgehensweise ausgewählter Funktionen.....	10
3.2.1 Darstellung des Win32 Aufrufs <i>EnumWindows</i>	10

3.2.2 Darstellung der Funktion für transparente Fenster.....	11
3.2.3 Darstellung der Funktion für reduzierte Fenster.....	12
3.2.4 Darstellung des alternativen Fensterwechslers <i>WindowScroll</i>	13
3.3 Grundlegende Vorgehensweise der Anwendung.....	14
4. Test der Software.....	15
4.1 Vorgehensweise Konfigurationstest.....	15
4.1.1 Beschreibung.....	15
4.1.2 Versuchsanordnung.....	15
4.2 Vorgehensweise Test der exakten Funktionsfähigkeit der Features.....	16
4.2.1 Beschreibung.....	16
4.2.2 Versuchsanordnung.....	16
5. Freigabe / Genehmigung.....	16
6. Tabellenverzeichnis.....	17
7. Abbildungsverzeichnis.....	17
8. Anhang / Ressourcen.....	18

1. Einleitung

1.1 Allgemeines

1.1.1 Zweck und Ziel dieses Dokuments

Dieses Pflichtenheft beschreibt die Entwicklung eines Programms für erweiterte Funktionen unter Microsoft Windows mit dem Codenamen Cassiopeia. Weiterhin sollen Verbindlichkeiten des Auftragsnehmers (kurz AN) gegenüber des Vorschlages der SDDR festgelegt werden. Somit kann man das Pflichtenheft als Plan zur Erstellung der Software sehen, bei dem Auftraggeber (kurz AG) und AN zustimmen müssen.

1.1.2 Projektbezug

Wir als SDDR stehen schon seit 5 Jahren in engem Kontakt mit dem Auftragsnehmer und entwickelten schon mehrere Anwendungen dieser Art. Auch hier sehen wir uns durch unsere Erfahrung in diesem Softwareumfeld in erster Reihe. Wir glauben auch an die Zukunft dieses Projektes und wenn es gewünscht ist auch an eine Weiterführung.

1.1.3 Abkürzungen

AN – Auftragsnehmer

AG – Auftraggeber

CP – Cassiopeia

1.2 Verteiler und Freigabe

1.2.1 Verteiler für das Pflichtenheft

Rolle	Name	E-Mail
Projektteam	Robert Schulze	robert.schulze@it2007.ba-leipzig.de
Projektteam	Thomas Klepzig	thomas.klepzig@it2007.ba-leipzig.de
Auftraggeber	Christian Heller	christian.heller@ba-leipzig.de

Tabelle 2: Verteiler für das Pflichtenheft

2. Konzept und Rahmenbedingungen

2.1 Ziele des Anbieters

Es soll ein Programm entwickelt werden, welches erweiterte Funktionen für Windows-Benutzer bereitstellt. Es arbeitet dabei direkt mit dem Win32 Application Programming Interface (API) zusammen, was den Einsatz von nativen Windows Funktionen ermöglicht.

2.2 Ziele und Nutzen des Anwenders

Zuerst das formulierte Ziel aus dem Lastenheft:

Dies ist es, ein Programm zu entwickeln, das effizienzverbessernde Funktionen für Windows bereitstellt. Dabei enthält es eine Alternative zum Wechseln des aktuell vorn liegenden Fensters, eine Möglichkeit für Kurznotizen sowie eine Reihe von Funktionen für den komfortableren Umgang mit Fenstern, die sich direkt in das Systemmenü jedes Fensters integriert. Weiterhin wird die Anwendung für verschiedene Sprachen lokalisiert sein, vorerst nur für Englisch und Deutsch, frei konfigurierbar über das Programm Menü.

Die verschiedenen Features für den erweiterten Umgang mit Fenstern werden im Detail die folgenden sein:

Es gibt die Möglichkeit ein Fenster sofort ganz nach unten zu verschieben, also unter alle Fenster, sodass es direkt auf dem Desktop liegt und keine anderen Fenster verdecken kann. Weiterhin kann man ein Fenster auch so konfigurieren, dass es immer über allen Fenstern liegt, so dass ebenfalls kein anderes Fenster es überdecken kann und der Inhalt immer lesbar ist.

Ferner existiert eine Funktion, um die Ausmaße eines Fensters zurückzusetzen, falls sich dieses unabsichtlich außerhalb des sichtbaren Bereichs befindet und die Möglichkeit, den Prozess, der diesem Fenster zugehörig ist, unsauber zu beenden, falls die Anwendung beispielsweise abgestürzt ist.

Um den Fokus besser auf das aktuelle Fenster zu lenken, kann man beliebige Fenster auf die Größe ihrer Titelleiste verkleinern, die sich bei Berührung oder Aktivierung automatisch wieder vergrößern.

Des Weiteren gibt es eine Reihe von Funktionen, um gewisse Aktionen für Gruppen von Fenstern durchzuführen, beispielsweise das Minimieren verschiedener Instanzen derselben Anwendung.

Auch eine Veränderung der Transparenz von Fenstern wird unterstützt, dabei kann man wählen, ob diese permanent oder nur beim Deaktivieren des Fensters eintritt. Auf Wunsch kann man diesen Transparenzeffekt auch beim Verschieben bzw. bei der Größenänderung aktivieren.

Abgesehen von den bereits genannten Funktionen für Fenster gibt es noch vier weitere Elemente von CP. Darunter zählen die Desktop Notizen, die einen Notiz Zettel zur Verfügung stellen, der jederzeit per Tastaturkommando in den Vordergrund geholt werden kann und sich sonst immer im Hintergrund befindet, der Effekt „GlassConsole“, der dazu führt, dass jede geöffnete Eingabeaufforderung den Aero Glas Effekt bekommt, den alternativen Fensterwechsler namens „WindowScroll“, mit dem es möglich ist, in Richtung der z-Ebene von Fenstern durch diese zu wechseln, ohne sie dabei zu aktivieren und den Zeitplaner, mit dem es möglich ist, unkompliziert und schnell einen Alarm zu programmieren, der Text enthalten, ein Programm starten sowie eine Sounddatei wiedergeben kann.

2.3 Benutzer / Zielgruppe

Es gibt keine speziellen Ziel- oder Benutzergruppen. Dieses Produkt richtet sich an alle Nutzer von Windows, die im täglichen Umgang mit dem Rechner neue, komfortable Funktionen nutzen wollen.

2.4 Systemvoraussetzungen

Benötigte Ressourcen	
Festplattenspeicher	< 1 MB
Hauptspeicher	mindestens 10 MB
Betriebssystem	Windows Vista oder Windows Server 2008 und höher

Tabelle 3: Systemvoraussetzungen

Zusätzlich werden die Mindestanforderungen des gewählten Betriebssystems benötigt.

2.5 Übersicht der Meilensteine

Vorbereitungsphase	
Fertigstellung Pflichtenheft	13.01.10
Freigabe Pflichtenheft	
Programmierung der Software	
Konzepterstellung	13.11.09
Programmierung	30.11.09
Test der Software	
Konfigurationstest	30.01.10
Exakte Funktionsfähigkeit der Features	30.01.10
Einführung	
Projektvorstellung	23.02.10
Voraussichtlicher Verkaufsstart	01.04.10

Tabelle 4: Übersicht über die Meilensteine

Das Datum für die Freigabe des Pflichtenheft ist bekannt, sobald die Genehmigung für das Dokument vorliegt.

3. Technische Beschreibung

3.1 Lebenszyklus

Die *main*-Funktion des Programms ist übersichtlich gehalten. Diese ruft ausschließlich das Hauptmodul *WinTools* auf, welches wiederum das komplexere Untermodul *DeskNotes* aufruft und alle anderen kleineren Module beherbergt.

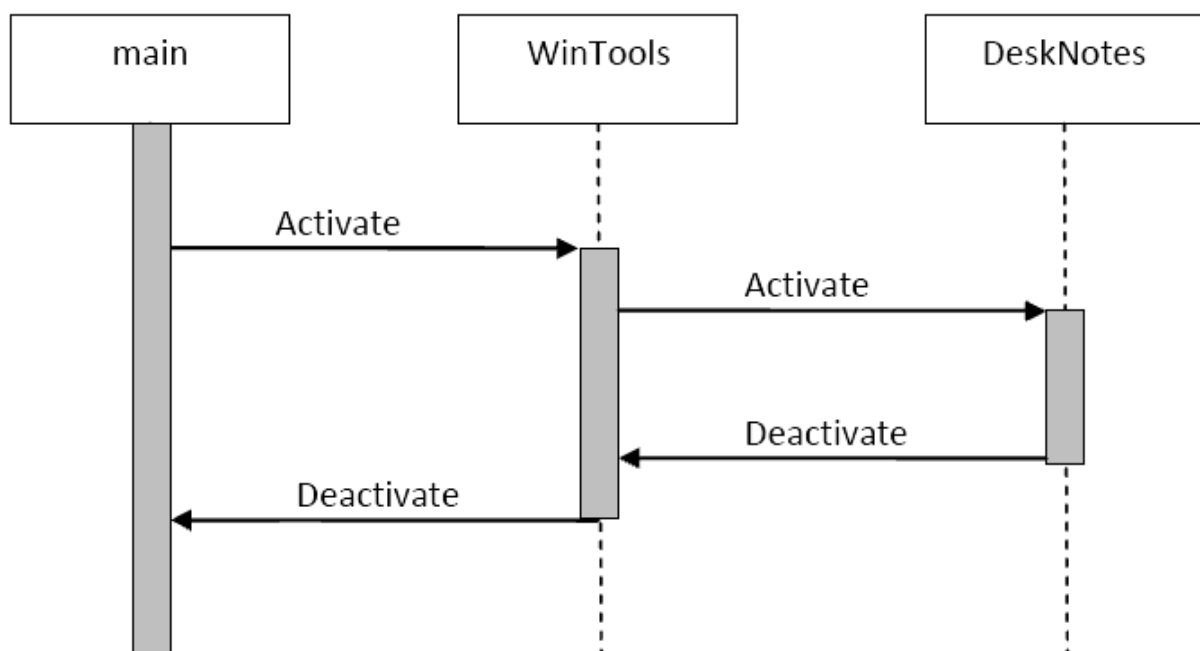


Abbildung 1: Lebenszyklus

Allerdings ist hierbei zu beachten, dass die *main* keine Klasse im engeren Sinne, sondern nur der vom Betriebssystem aufgerufene Einstiegspunkt ist.

3.2 Grafische Darstellung der Vorgehensweise ausgewählter Funktionen

3.2.1 Darstellung des Win32 Aufrufs *EnumWindows*

Das folgende Diagramm zeigt die Verarbeitung mehrerer Fenster bezüglich einer Aktion die ausgeführt werden soll. Das heißt im Detail, dass die Startfunktion wie beispielsweise der Befehl „Minimize All“ oder auch „Restore Windows Of This Program“ und andere für jedes Fenster bzw. eine Auswahl davon eine bestimmte Aktion durchführen wie Minimieren oder Wiederherstellen.

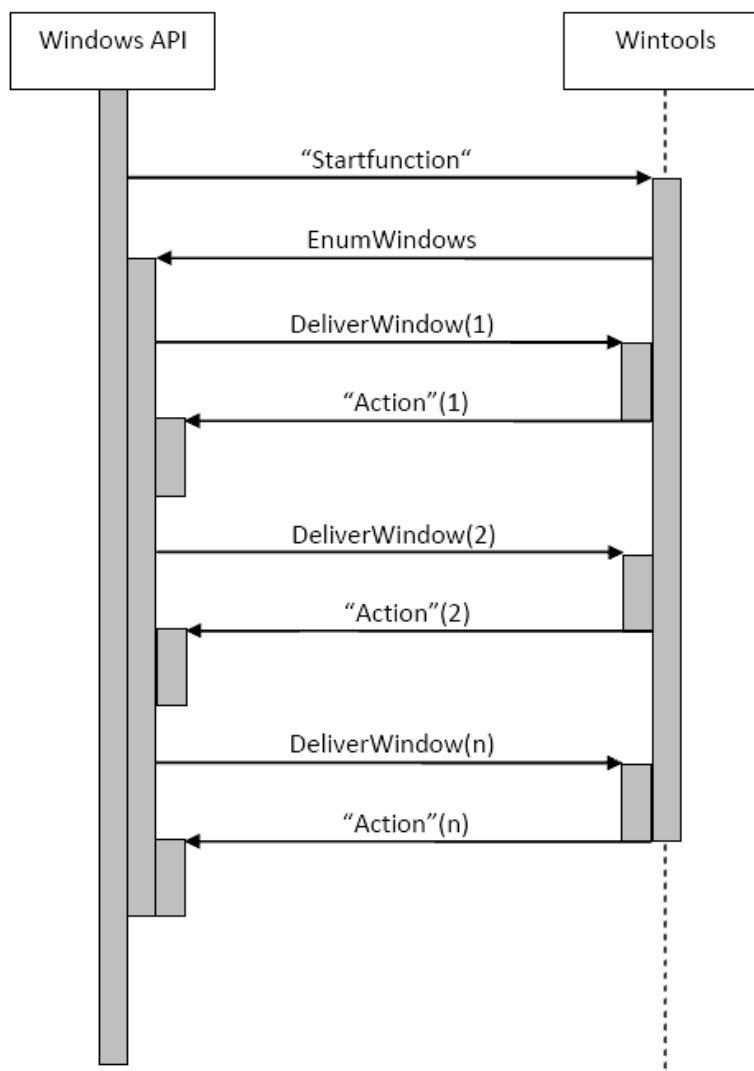


Abbildung 2: Darstellung des *EnumWindows*-Aufrufs

Die Win32 API wird zur Vereinfachung als ein Objekt einer Klasse betrachtet, da die Kapselung dieser in eine Klasse möglich ist und mittels dieser dann die Funktionen im Programm selbst genutzt werden können.

3.2.2 Darstellung der Funktion für transparente Fenster

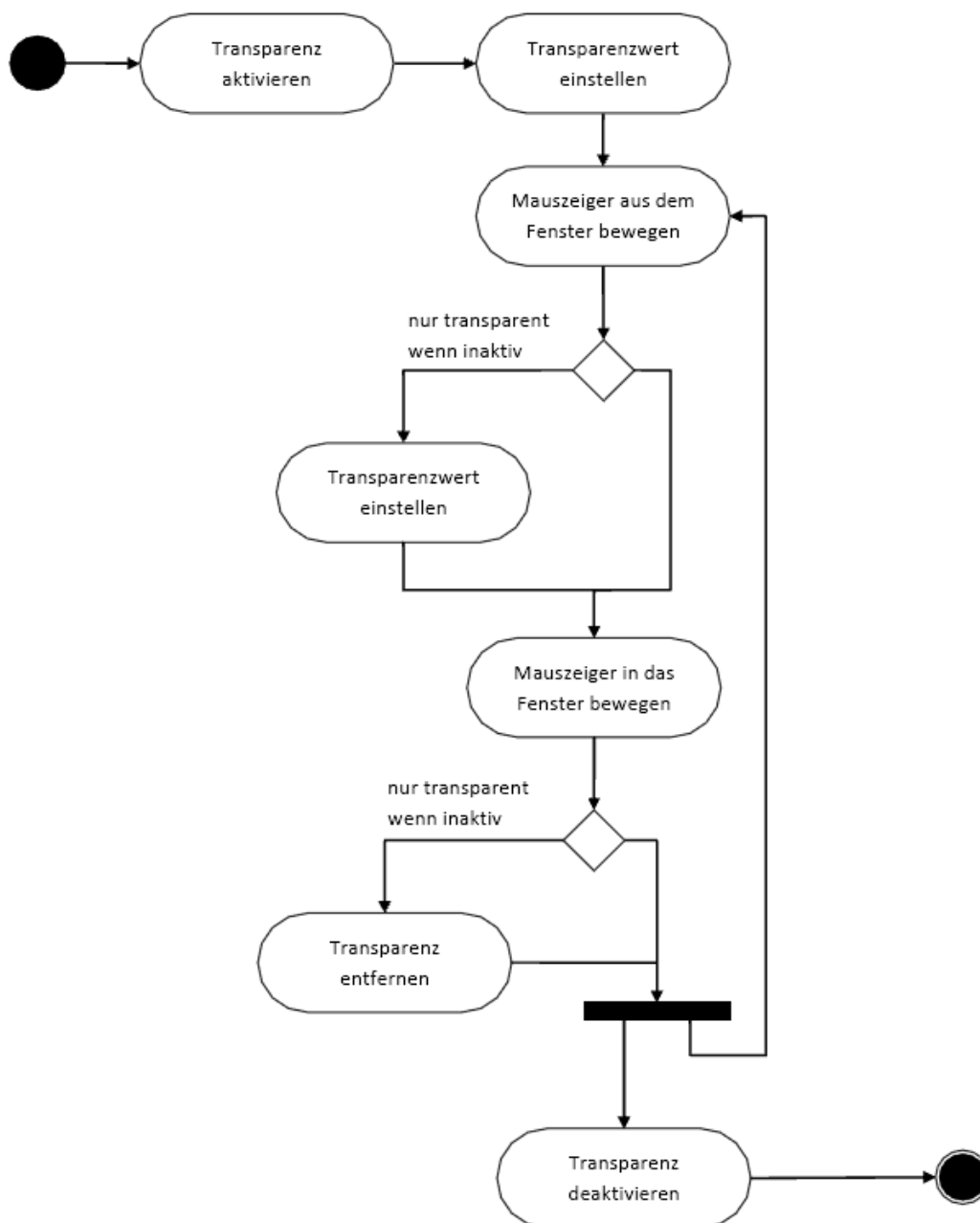


Abbildung 3: Transparenzeffekt für Fenster

Abbildung 3 veranschaulicht, wann ein Fenster, bei aktiviertem Transparenzeffekt, voll bzw. nur teilweise sichtbar ist.

3.2.3 Darstellung der Funktion für reduzierte Fenster

In dem folgenden Diagramm wird der Nachrichtenverkehr von Windows und den *WinTools* dargestellt. Dabei leiten die *WinTools* wiederum die Nachrichten an den *WindowReducer* weiter. Der *WindowReducer* übernimmt hier die eigentliche Logik und prüft, wann die Fenster auf ihre Titelleiste reduziert oder wieder normal groß dargestellt werden sollen. Die Funktion *ReduceOtherWindows* arbeitet intern mit dem im *EnumWindows* Diagramm (siehe Abbildung 2) dargestellten Verfahren zusammen.

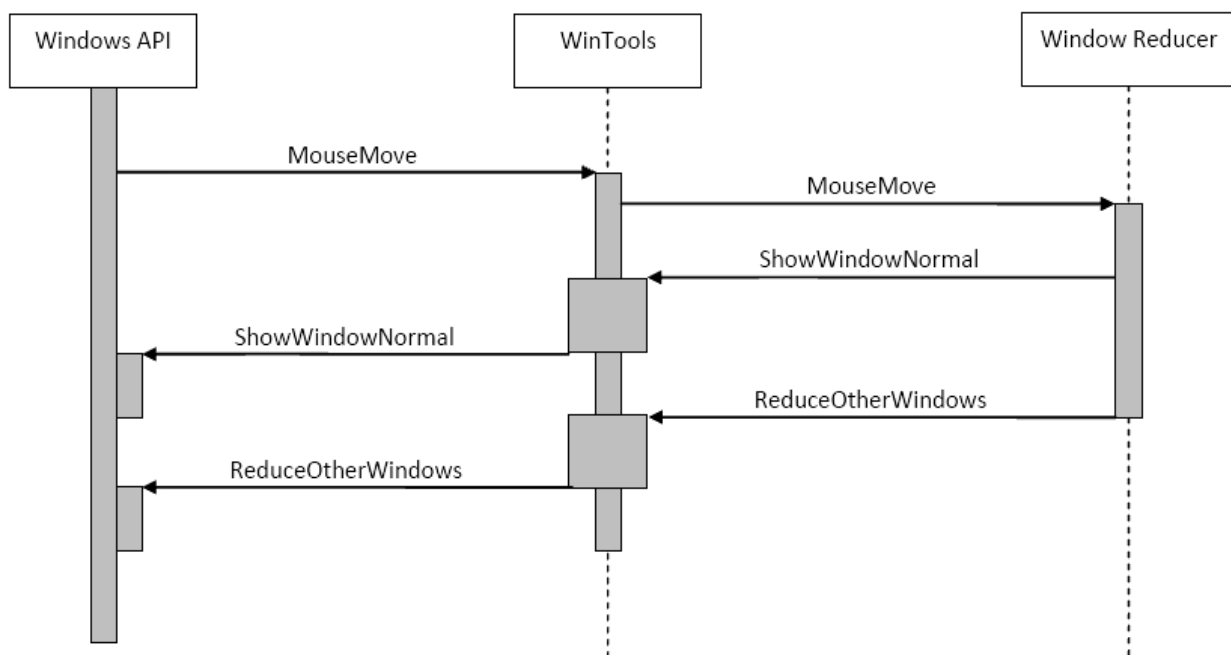


Abbildung 4: Reduzieren von Fenstern

Die Win32 API wird zur Vereinfachung erneut als ein Objekt einer Klasse betrachtet.

3.2.4 Darstellung des alternativen Fensterwechslers *WindowScroll*

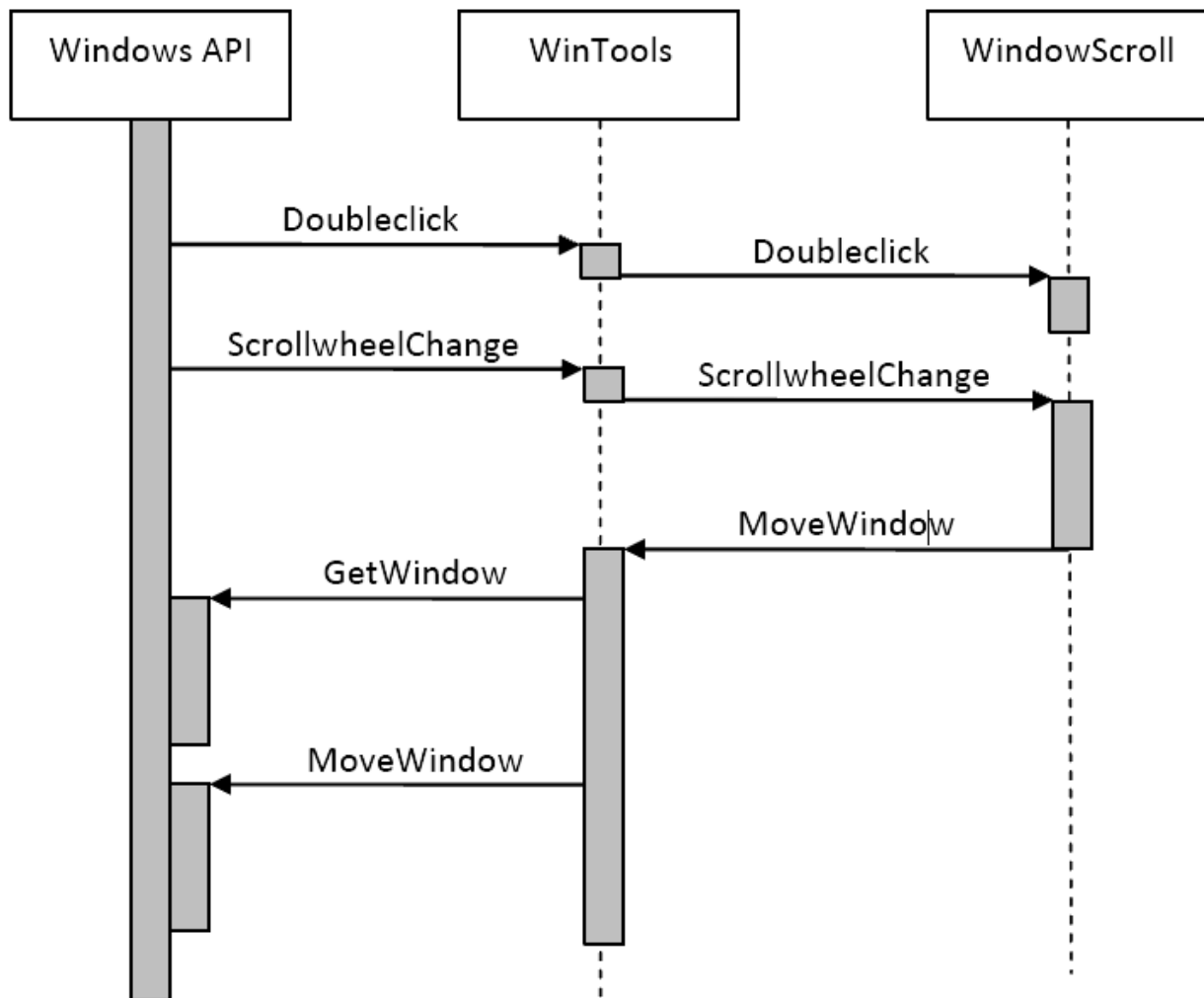


Abbildung 5: Funktionsweise *WindowScroll*

Die Doppelklick-Benachrichtigung aktiviert die *WindowScroll* Routine, anschließend kann dann durch Drehung des Mausekklings nach oben oder unten die Fensterreihenfolge manipuliert werden, dabei ist auch ein mehrfaches Scrollen möglich.

Die *MoveWindow* Funktion bewegt – je nach dem ob nach das Mausekkling nach oben oder unten gedreht wurde – das vorderste Fenster nach hinten oder das hinterste nach vorn.

Auch hier wird die Win32 API wieder vereinfacht dargestellt.

3.3 Grundlegende Vorgehensweise der Anwendung

Wie bereits im Lebenszyklus erläutert, wird im Haupteinstiegspunkt der Anwendung die Klasse *WinTools* angesprochen. Diese ist eine Klasse mit ausschließlich statischen Elementen, sodass eine Instanz davon nicht benötigt wird.

Diese Klasse stellt die Hauptschnittstelle zum Betriebssystem dar, da hier alle für CP bedeutsame Nachrichten und Ereignisse eingehen. Auch das Integrieren sämtlicher Funktionen in das Systemmenü jedes Fensters geschieht hier.

Um diese Integration effektiv zu gewährleisten, gibt es mindestens zwei Möglichkeiten:

Man kann einerseits jede Nachrichtenschleife jeder offenen Anwendung überwachen und bei dem entsprechendem Klick auf ein Systemmenüeintrag die Nachricht abfangen, die das zugehörige Fenster und damit die zugehörige Anwendung bekommt. Dies ist jedoch sehr aufwändig und mit Verlust von Geschwindigkeit gekoppelt.

Daher habe wir uns als SDDR für eine alternative Vorgehensweise entschieden. Wir nutzen einen sogenannten *WinEventHook*, durch den uns Windows verschiedene Ereignisse mitteilt, so auch, dass auf ein Menüeintrag geklickt wurde. Übergeben bekommt man eine ID, eine Kennzeichnung, die von CP festgelegt wurde und somit ist bekannt, welcher Menüeintrag geklickt wurde.

Um nun noch zu ermitteln das Fenster zu ermitteln, dessen Systemmenü das Ereignis ausgelöst hat, bedient man sich eines weiteren Ereignisses: Das Öffnen eines Menüs. Empfängt man so ein Öffnen eines Systemmenüs und daraufhin ein Klick auf ein Menüeintrag, so hat man alles, was man benötigt, um darauf entsprechend reagieren zu können.

So kann man nun verschiedene Effekte aktivieren oder auslösen, wie zum Beispiel die Funktion „TopMost“. Die Aktivierung dieses Features führt dazu, dass es kein normales Fenster verdecken kann, es ist immer an oberster Stelle in der z-Ebene der Fenster.

4. Test der Software

4.1 Vorgehensweise Konfigurationstest

4.1.1 Beschreibung

In diesem Test werden die verschiedenen Konfigurationsmöglichkeiten geprüft. Dies geschieht durch Testen jeder möglichen Konfiguration.

4.1.2 Versuchsanordnung

- Programm starten
- eine Option ändern
- Programm beenden

Dies war der erste Teil des Programmtests. Danach folgt

- Programm starten
- Prüfen ob die Option noch gesetzt ist

Dies sollte zur Folge haben dass die Option noch gesetzt ist. Diesen Test wiederholt man für alle möglichen setzbaren Optionen und Konfigurationen.

4.2 Vorgehensweise Test der exakten Funktionsfähigkeit der Features

4.2.1 Beschreibung

In diesem Test werden die verschiedenen Funktionen auf Korrektheit und Fehlerfreiheit überprüft.

4.2.2 Versuchsanordnung

- Programm starten
- eine Funktion ausführen/aktivieren
- eine Funktion deaktivieren
- eine Funktion ausführen/aktivieren

Dies sollte zur Folge haben dass die gewählte Funktion entsprechend den Erwartungen arbeitet und sich ordnungsgemäß abschaltet bzw. aktiviert . Diesen Test wiederholt man für alle vorhandenen Features des Programms.

5. Freigabe / Genehmigung

Die Genehmigung erfolgt nach Durchsicht des Pflichtenheftes und mit nachstehender Unterschrift durch den AG.

Datum	13.01.10
Unterschrift Auftraggeber	
Unterschrift Projektleiter	

6. Tabellenverzeichnis

Tabelle 1: Historie der Dokumentenversionen.....	2
Tabelle 2: Verteiler für das Pflichtenheft.....	5
Tabelle 3: Systemvoraussetzungen.....	7
Tabelle 4: Übersicht über die Meilensteine.....	8

7. Abbildungsverzeichnis

Abbildung 1: Lebenszyklus.....	9
Abbildung 2: Darstellung des <i>EnumWindows</i> -Aufrufs.....	10
Abbildung 3: Transparenzeffekt für Fenster.....	11
Abbildung 4: Reduzieren von Fenstern.....	12
Abbildung 5: Funktionsweise <i>WindowScroll</i>	13

8. Anhang / Ressourcen