

**Plug-In Portierung auf NeuroCheck 6.0**SE-PW-079  
22. Oktober 2009

Autor: NeuroCheck GmbH, support@neurocheck.com, 0711-229646-31  
Stand: 22.10.2009  
Zusammenfassung: Dieses Dokument gibt Hinweise für die Portierung von Plug-Ins der NeuroCheck Version 5.1 auf das neue Plug-In Interface der NeuroCheck Version 6.0.

**1. Vorbemerkungen**

Ebenso wie NeuroCheck 6.0 einen bedeutenden **Technologiesprung** auf die neue Microsoft .NET Plattform erfahren hat, ändern sich auch die technologischen Grundlagen des NeuroCheck Plug-In Interfaces.

Deshalb müssen bestehende Plug-In DLLs, die ursprünglich für NeuroCheck Version 5.1 entwickelt wurden, auf die neue Technologie **portiert** werden, um in NeuroCheck 6.0 verwendet werden zu können.

Die Portierung bedeutet für die einzelnen Plug-In Projekte einen nicht unerheblichen **Programmieraufwand**.

Das neu gestaltete Plug-In Interface der NeuroCheck Version 6.0 bietet darüber hinaus eine großzügige **Erweiterung der Funktionalität** und zusammen mit der neuen .NET Technologie eine deutlich bessere Unterstützung zur Entwicklungszeit.

Vielleicht kann auch neue Standard-Funktionalität oder eines der neuen Konzepte in der NeuroCheck Version 6.0 Ihre Plug-In Funktionalität nahtlos ersetzen.

**2. Änderungen am Plug-In Interface****2.1. Microsoft .NET**

Wesentliche Neuerung an dem Plug-In Interface ist die zugrunde liegende Technologie. NeuroCheck 6.0 Plug-Ins sind Microsoft .NET Assemblies. Für die Portierung der Plug-In ist Erfahrung in der .NET Programmierung erforderlich.

Durch die Common Language Specification (CLS) des .NET Frameworks können Sie die verwendete Programmiersprache anhand persönlicher Kenntnisse aus einer Vielzahl möglicher .NET Sprachen wählen.

Es ist eine CLS-kompatible Entwicklungsumgebung notwendig. Wir empfehlen Microsoft .NET Visual Studio 2008. Hier ist die kostenfreie Express Version von Microsoft ausreichend.

## 2.2. Plug-In Editor

Einfache Plug-In DLLs lassen sich auch im integrierten Plug-In Editor der NeuroCheck Version 6.0 erstellen. Hier muss sich der Entwickler nicht in Interface-Definitionen einarbeiten, er schreibt im Wesentlichen lediglich das Codefragment der Execute-Routine. Für komplexere Funktionalitäten empfehlen wir allerdings die externe Entwicklung in einer professionellen Entwicklungsumgebung.

## 2.3. Struktur des Plug-In Interfaces

Die bewährte Grundstruktur einer NeuroCheck Plug-In ist beibehalten worden, so dass Sie die Logik bei der Portierung Ihrer Plug-In unverändert weiter verwenden können (es gibt weiterhin die Init, Execute, UnInit, etc. Aufrufe).

Die stark objektorientierte Gestaltung des neuen Interfaces werden Sie dabei als sehr vorteilhaft empfinden.

Eine ausführliche Beschreibung unseres neuen Interfaces finden Sie in der "Dokumentation des NeuroCheck 6.0 Plug-In-Interfaces" (PDF Dokumentation).

## 2.4. Erweiterungen der Plug-In API

Komplett neu in der NeuroCheck 6.0 API ist der Zugriff auf interne Bildverarbeitungs-Funktionalität. So können beispielsweise eigene Vorverarbeitungsschritte mittels eines einzigen Aufrufs durch bewährte NeuroCheck Algorithmen ersetzt werden.

Durch die Teilnahme am NeuroCheck Registerkonzept können 6.0 PlugIns auf beliebige globale Daten (z.B. Sollwertvorgaben) zugreifen, oder selbst Daten in das Register schreiben. Das sollte die übliche Vorgehensweise aus V5.1, alle Informationen zwischen Prüfschritten per Maßliste zu transportieren, überflüssig machen.

## 2.5. Prüfprogramm Know-How

Alte Prüfprogramme können nach der Portierung Ihrer Plug-In DLLs weiter verwendet werden, voraus gesetzt, es steht für jede im NeuroCheck 5.1 Prüfprogramm verwendete PlugIn DLL eine .NET Assembly entsprechend zur Verfügung.

Die alten Prüfprogramme werden hierzu aus NeuroCheck V5.1 als Xml Datei exportiert und mit der neuen Version 6.0 wieder importiert, anschließend können die Prüfprogramme wie gewohnt als Binärdatei (\*.chr) gespeichert werden.

### 3. Hinweise zum Portierungsvorgang

#### 3.1. Prüfen der Notwendigkeit

Zunächst sollten Sie prüfen, ob die spezifische Funktionalität Ihrer Version 5.1 Plug-In DLL in NeuroCheck 6.0 noch erforderlich ist. Möglicherweise ist diese Funktionalität jetzt bereits im Standard-Umfang der Software-Version 6.0 enthalten. Im Zweifel fragen Sie bei der NeuroCheck GmbH nach.

#### 3.2. Prüfen des Portierungsweges

Prüfen Sie, ob sich die Funktionalität Ihrer Version 5.1 Plug-In DLL auf dem einfachen Wege der integrierten Entwicklungsoberfläche (Plug-In Editor) nachbilden lässt. Hier könnten Sie sich ggfls. einiges an Portierungsaufwand ersparen, weil Ihnen das Framework der neuen Plug-In Schnittstelle durch die Software selbst abgenommen wird. Für kleinere Plug-Ins mit geringer Komplexität ist dies oft möglich.

#### 3.3. Umfangreicher Portierungsweg

Wenn Sie festgestellt haben, dass Sie die Funktionalität Ihrer Version 5.1 Plug-In DLL umfassend nachbilden müssen, empfehlen wir Ihnen, sich an folgenden Hinweisen zu orientieren:

##### 3.3.1. Template-Projekt

Als Basis für Ihre neue .NET Plug-In empfehlen wir, mit einem von uns bereitgestellten Beispiel-Projekt für Visual Studio 2008 zu beginnen (bitte beim NeuroCheck Support anfragen).

Die Bedeutung der einzelnen Klassen und Attribute, sofern diese nicht selbsterklärend sind, entnehmen Sie bitte unserer "Dokumentation des NeuroCheck 6.0 Plug-In-Interfaces".

##### 3.3.2. Manuelle Portierung

Die Hauptaufgabe wird nun darin bestehen, die Methoden-Rümpfe der Prüfschrittclassen manuell durch Modifizierung des ursprünglichen (meist C++) Codes zu implementieren.

In der Regel wird der Codeumfang durch Verwendung der neuen .NET Sprache und der umfangreichen .NET Klassenbibliothek deutlich kürzer, und damit zukünftig auch besser wartbar.

Die Beschreibungs-Strukturen der NeuroCheck Datenobjekte (z.B. sPI\_IMAGE → NcPIImage) sind prinzipbedingt sehr ähnlich in den beiden Interface Versionen, so kann die Logik beim Auslesen der Bild- oder Arbeitsbereichs-Informationen in der Regel beibehalten werden.

Eventuell können Sie ja auch ganze Codeabschnitte in der Execute durch Methoden der neuen NeuroCheck V6.0 API ersetzen (z.B. für die Bildfilterung).

##### 3.3.3. Dialoge

Durch die Verwendung der neuen Windows.Forms Oberfläche anstatt der alten MFC müssen die Parameterdialoge komplett neu geschrieben werden. Durch die

umfangreiche Unterstützung des Visual Studio Designers sind Implementierungen deutlich schneller, als unter der MFC, außerdem bekommt Ihr Parameterdialog ein ansprechendes, neues Look and Feel.

#### 3.3.4. Externe Bibliotheken

Sollten Sie bisher externe unmanaged (Win32) Bibliotheken verwendet haben, erkundigen Sie sich bitte beim Hersteller, ob zwischenzeitlich ein .NET SDK existiert, falls nicht, finden Sie in der Fachliteratur unter den Stichwörtern "P/Invoke" oder "DllImport" die notwendigen Mechanismen zur Integration von Win32 Bibliotheken.

## 4. Einschränkungen

1. API Funktionen zum Bitmap Management entfallen ersatzlos, stattdessen kann die .NET Klasse Bitmap verwendet werden.
2. Die API Funktion GetXmlString wird es in der neuen NeuroCheck 6.0 API nicht mehr geben. Der neue Lösungsansatz könnte über die Register laufen.