Technische Dokumentation MedPlaus 6

(S. Freudiger, S. Hasler)

28. April 2020

Freudiger EDV-Beratung
Zeughausgasse 16
Postfach
3001 Bern
Schweiz

www.freudiger.com

Inhaltsverzeichnis

1 Einführung

MedPlaus 6 ist das offizielle Programm für die Plausibilisierung der Daten der Medizinischen Statistik der Krankenhäuser. Es dient der Überprüfung von BFS-Records oder -exporten auf Ihre Korrektheit.

MedPlaus wird im Auftrag des Bundesamtes für Statistik (BFS) entwickelt und zur Verfügung gestellt. Es läuft unter allen aktuellen Windows-Versionen.

Die Applikation enthält Hunderte von Regeln, welche die Angaben des Minimaldatensatzes inklusive Diagnose- (ICD) und Operationskodes (CHOP) überprüfen. Neugeborenen- und Psychiatrie-Zusatzdatensätze werden ebenso getestet wie der MD-Datensatz.

MedPlaus 6 ist auf Geschwindigkeit optimiert. Selbst sehr grosse Dateien werden schnell geprüft.

1.1 Versionen

Es gibt drei MedPlaus-Versionen:

Version	**MedPlaus stand alone**	**MedPlaus Shell**	**MedPlaus DLL**
	MedPlaus stand alone	MedPlaus Shell	MedPlaus DLL
Für	Endbenutzer	Poweruser	Softwarehersteller
Beschreibung	Enthält ein Installationsprogramm und eine grafische Benutzeroberfläche. Damit können Mitarbeiter in Spitälern und Ämtern einfach Quartals- und Jahreslieferungen auf ihre Qualität hin überprüfen.	Dank ihrer überragenden Geschwindigkeit ist MedPlaus auf der Kommandozeile ideal für Auswertungen und die Plausibilisierung grosser Datenbestände.	Erlaubt die Integration der MedPlaus-Funktionalität in Produkten von Drittherstellern. Mit der DLL können einzelne Records überprüft werden.
Auslieferung	Installer-Datei (mp600ch.exe)	Teil der MedPlaus DLL-Distribution (zip-Datei)	MedPlaus DLL-Distribution (zip-Datei)

Abbildung 1.1: MedPlaus-Versionen

1.2 Download

Die Distributionen können von der MedPlaus-Seite der Freudiger EDV-Beratung heruntergeladen werden. Sie finden diese unter

- `http://www.freudiger.com/medplaus/` (Homepage auf Deutsch)
- `http://www.freudiger.com/f/medplaus/` (Homepage auf Französisch)

2 MedPlaus stand alone

Die stand alone-Version wird nicht in diesem Dokument beschrieben. Sie enthält eine umfassende Online-Hilfe. Diese kann nach erfolgter Installation aus dem Programm heraus aufgerufen werden.

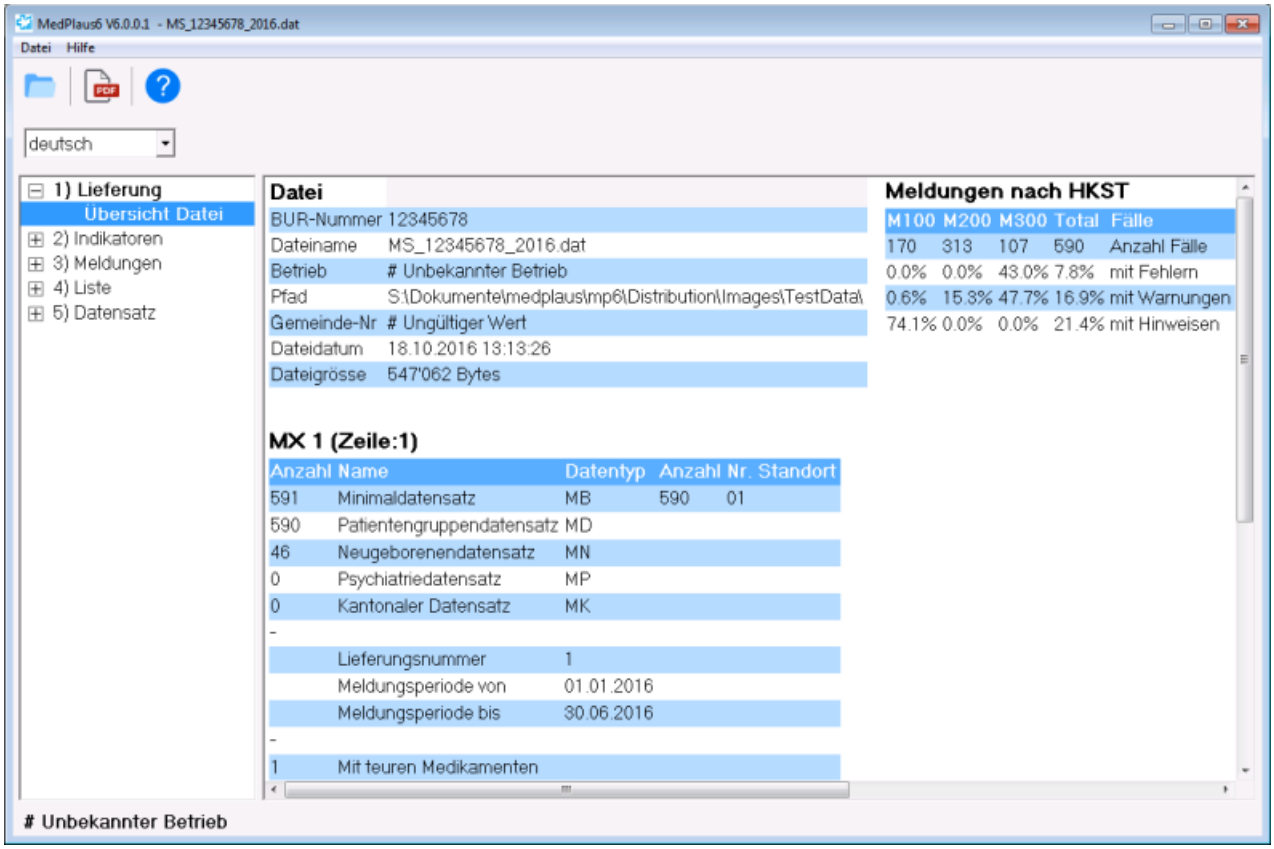

Abbildung 2.1: MedPlaus 6 stand alone

3 MedPlaus Shell

MedPlaus kann mit zahlreichen Parametern per Kommandozeile gestartet werden (z.B. als Batch). Der Aufruf gestaltet sich wie folgt:

- `medplaus6.exe [Options] [DataFilename]`

Der alleinige Aufruf von `medplaus6.exe` listet alle möglichen Parameter auf.

Es sind folgende [Options] möglich:

Parameter	Bedeutung
-de	Erstellt deutsche Texte für Log-Meldungen (default)
-fr	Erstellt französische Texte für Log-Meldungen
-log=FILENAME	Schreibt Meldungen in Datei FILENAME
-log	Schreibt Meldungen in Datei DataFilename.log
-report=FILENAME	Schreibt MX-Bericht in Datei FILENAME
-report	Schreibt MX-Bericht in Datei DataFilename.rep
-stat=FILENAME	Schreibt statistische Werte in Datei FILENAME
-stat	Schreibt statistische Werte in Datei DataFilename.stat
-f [formatString]	Verwendet das benutzerspezifisch definierte Ausgabeformat (s. MP_setFormatString())
-mppath=PATH	Legt den Pfad der Indexdateien (*.ix5) fest
-version	Anzeige der Version
-MKGR	Integration des kantonalen Zusatzdatensatzes Graubünden
-MKLU	Integration des kantonalen Zusatzdatensatzes Innerschweiz (LUSTAT)

Dabei gelten folgende Regeln und Konventionen:

- Die Reihenfolge der Parameter spielt keine Rolle
- Gross- und Kleinschreibung muss beachtet werden
- Bei allen Dateinamen sollte der Pfad angegeben werden. Ansonsten gilt das Verzeichnis, aus welchem der Aufruf erfolgt.

Beispiel: Programm mit Datei starten, inkl. Erstellung einer französischsprachigen Logdatei

- Klicken Sie auf den *Start*-Knopf
- Wählen Sie *Ausführen*
- Geben Sie folgenden Text ein (passen Sie dabei Laufwerks-, Verzeichnis- und Dateinamen Ihren Bedürfnissen an):
  `c:\programme\mp6042\medplaus6.exe -fr -log=c:\logfiles\2020\test.log test.dat`

4 MedPlaus DLL

Die Medplaus-DLL 6.0.4.2 ist betreffend Funktionsaufrufe mit Ausnahme der erweiterten MP_TestRec-Funktion, der neuen und optionalen MP_setFormatString()-Funktion und der Umbenennung einiger DLL-Dateien kompatibel zur Vorversion MedPlaus 5. Sie benötigt ein Minimum von 10 MByte Arbeitsspeicher, da sie alle benötigten Informationen in diesen lädt.

Die Datei medplaus6DLL.dll ist neu per se multilingual. Sie erweitert die Funktion MP_TestRec() um einen Sprachparameter. Der Sprachwechsel ist on-the-fly möglich.

Es wird auch eine 64-Bit Version der DLL (inkl. entsprechende MP6 DLL-Testgui) mitgeliefert. Die Datei nennt sich medplaus6DLL64bit.dll.

4.1 Installation

Die Dateien der DLL-Version sind in medplaus_dll_v6042_20200428.zip enthalten.

Entpacken Sie die Dateien in ein Verzeichnis und sprechen Sie die medplaus6DLL.dll[1] (bzw. medplaus6DLL64bit.dll) aus Ihrer Software an

Die DLL funktioniert ohne Installation von MedPlaus 6 stand alone.

4.1.1 Kompatibilität zu MedPlaus 5

Die MedPlaus 6 DLL weist bez. Kompatibilität zur MedPlaus 5 DLL folgende Unterschiede auf:

- Neuer Parameter cLanguage[2] im Funktionsaufruf MP_TestRec() der Standard-DLL
- Neue Dateinamen
  - medplaus6DLL.dll statt medplaus5DLL.dll
  - medplaus6DLL64bit.dll statt medplaus5DLL64bit.dll
- Neue Funktion MP_setFormatString()
  - Ohne Aufruf derselben gilt das bisherige MP5-kompatible Format als Default

4.1.2 Unterverzeichnisse

Im Verzeichnis \help befindet sich die Dokumentation.

Im Verzeichnis \DLL_BigLog befinden sich alternative Versionen der DLL. Sie bieten eine umfassende LOG-Funktionalität, die bei der Fehlersuche hilft. Die Protokolldatei wird in das Verzeichnis c:\temp geschrieben. Dieses muss vorhanden sein.

Diese Versionen sind deutlich langsamer als die Standard-DLLs.

[1] Alternativ können Sie die medplaus6.exe im Batchbetrieb mit den nötigen Parametern aufrufen (s. MedPlaus Shell).
[2] bisher Teil der separaten medplaus5DLL_ML.dll

4.1.3 Updates

In der Regel können innerhalb derselben MedPlaus-Version die Dateien der aktuellen zip-Distribution in die bestehende Umgebung hineinkopiert werden. Es genügt also, die früheren Dateien durch die neuen zu ersetzen.

Im Zweifelsfall bitte die Technische Dokumentation konsultieren. Sollte dies nicht weiterhelfen, steht Ihnen die Hotline der Freudiger EDV-Beratung zur Verfügung.

4.2 Lieferumfang

Die wichtigsten Dateien sind:

Dateiname	Kommentar
medplaus6DLL.dll	Produktive 32-Bit MedPlaus 6-DLL (multilanguage)
medplaus6DLL64bit.dll	Produktive 64-Bit MedPlaus 6-DLL (multilanguage)
medplaus6DLL_BigLog.dll	Alternative 32-Bit MedPlaus 6-DLL mit voller Log-Funktionalität
medplaus6DLL64bit_BigLog.dll	Alternative 64-Bit MedPlaus 6-DLL mit voller Log-Funktionalität
bur.ix5	Betriebsnummern-Index
chop.ix5	CHOP-Index (Version 2020 und Vorversionen)
icd.ix5	ICD-10 GM-Index (Version GM 2018 und Vorversionen)
natio.ix5	Nationalitäten-Index
plz.ix5	Postleitzahlen-Index
regio.ix5	CH Regionenkodes-Index
wregio.ix5	Weltregionen-Index
mp6show.exe	Separates Anzeigefenster für Datensatz und Meldungen aus mp6DllDynTest.exe und mp6DllDynTest64.exe
mp6DllDynTest.exe	GUI zum Testen der DLL-Aufrufe (32-Bit)
mp6DllDynTest64.exe	GUI zum Testen der DLL-Aufrufe (64-Bit)
mp60xx_dll_changes_d.pdf	Auflistung der Änderungen der jeweiligen MedPlaus DLL (pdf-Version)
mp60xx_dll_changes_d.md	Auflistung der Änderungen der jeweiligen MedPlaus DLL (Markdown-Version)
mp6042_dll_techDoc_d.pdf	TechDoc zur MedPlaus 6-DLL
MS_Schnittstellenkonzept_2020_d.pdf	BFS-Schnittstellenkonzept 2020

Die Indexdateien tragen auch in MedPlaus 6 weiterhin die Dateiendung ix5 (neues Format seit MedPlaus 5).

4.3 Meldungstypen

MedPlaus 6 kennt folgende Meldungen:

Meldungstyp	Kommentar
Fehler	Müssen korrigiert werden. Toleranzgrenze: max. 1% (gemäss BFS)
Warnung	Sollte beachtet/korrigiert werden. Toleranzgrenze: max. 50% (gemäss BFS)
Hinweis	Liefert zusätzliche Informationen zur Verbesserung der Datenqualität
Struktur	Deutet auf strukturelle Fehler hin (z. B. ein ungültiges Recordformat oder Zeichen). In einem solchen Fall wird der Rückgabewert -1 übermittelt.

4.4 Tests

4.4.1 Neue Testhierarchie/-nummern

Für MedPlaus 6 wurde die gesamte Testhierarchie überarbeitet. Die grundlegende Idee war, dass die für die Kodierung relevanten Meldungen zuoberst erscheinen, gefolgt von Tests auf administrative Daten und - zum Schluss - auf strukturelle Probleme (was v. a. IT und Softwarehersteller betrifft).

Mit dieser Überarbeitung besteht wieder eine stringente Testlogik für die kommenden Jahre.

Die MedPlaus-Tests haben deswegen neue Nummern erhalten.

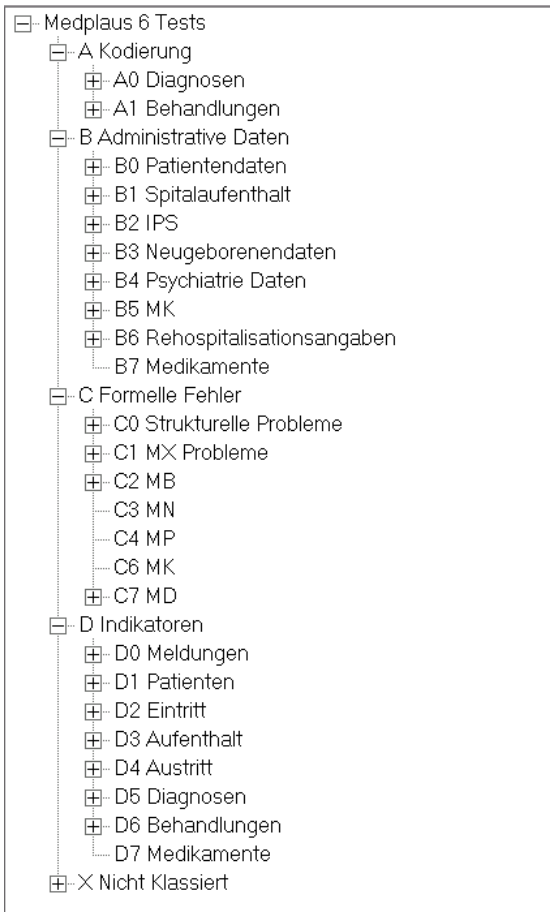

Abbildung 4.1: Die neue Testhierarchie

4.4.2 Plausibilisierungskonzept

Das Plausibilisierungskonzept beschreibt die in MedPlaus 6 integrierten Tests.

Das passende Konzept zu dieser Version erscheint nach Fertigstellung der definitiven MedPlaus 6 stand alone-Version. Sie finden es unter:

- `http://www.freudiger.com/medplaus/` (Homepage auf Deutsch)
- `http://www.freudiger.com/f/medplaus/` (Homepage auf Französisch)

4.5 Beschreibung der Funktionen

4.5.1 Zeichensatzcodierung

Es gilt ASCII/ANSI (latin1). UTF wird nicht unterstützt und von den übermittelten Zeichen auch nicht benötigt.

4.5.2 Typenbeschreibung

Delphi	C	Beschreibung
Integer	int	32-Bit/64-Bit Integer
PChar	* char	32-Bit/64-Bit Pointer auf einen nullterminierten und 8-bit breiten String

4.5.3 function MP_Create(MPDir: PChar; cLanguage: Integer): Integer;

Initialisiert Testsystem: Erzeugt eine Instanz und lädt alle benötigten Ressourcentabellen in den Arbeitsspeicher.

Parameter	Beschreibung
MPDir	Das Verzeichnis, in dem sich die txt- und ix5-Dateien befinden
cLanguage	0: Deutsch 1: Französisch
Rückgabewert	<=0: Fehler >0: Handle des Testfensters (bei den weiteren Funktionen als HDL weiterverwendet)

4.5.4 function MP_setFormatString(formatString: PChar)

Neu ab MedPlaus 6: Definiert das Ausgabeformat der Meldungen.
Wird die Funktion nicht eingesetzt, gilt das Default-Ausgabeformat analog MedPlaus 5.

Parameter	Beschreibung
formatString	String, der das Ausgabeformat der Meldungen definiert. Die verfügbaren Felder werden über {{Domain.Feld}} angesprochen. Dazwischen können sich - ausser '{{' - beliebige Zeichen befinden.
Beispiele	{{MB.1}} gibt das Feld 0.1.V01 des MBs aus ('MB') {{MB.4}} gibt das Feld 0.1.V04 des MBs aus (z. B. 'BE') {{SYS.Version}} gibt die Version des Programms aus (z.B. '6.0.0.0 x64')
Domains	**MB** - alle Felder des MBs gemäss MS-Spezifikation des BFS sind über ihren Rang abrufbar **MD** - alle Felder des MDs gemäss MS-Spezifikation des BFS sind über ihren Rang abrufbar **SYS** - Systemmeldungen aus MedPlaus sind über nachfolgende Felder abrufbar
SYS-Domainfelder	*StatYear*: Statistikjahr *Version*: MedPlaus-Version *LineNr*: Zeilennummer *ErrorNr*: Lange Fehlernummer (> 5 Stellen bei Diagnosen, Behandlungen) *ErrorNrS*: Kurze Fehlernummer (5 Stellen. Ohne Nr. der Behandlung, Diagnose) *ErrorNrOld*: Lange Fehlernummer nach MP5-System. (mögliche Überschneidung bei neuen Tests ohne alte Fehlernummer) *MsgType*: Fehler, Warnung, Hinweis, ... *BUR*: Betriebsnummer *FID*: Fallidentifikationsnummer *HKST*: BFS-Hauptkostenstelle *KT*: Kanton *Text0*: kurzer Text (z.B. ohne '1. Nebendiagnose: ') *Text1*: langer Text, wie bisher (z.B. 'n. Nebendiagnose: ') *Text2*: abgekürzt: '1.ND: ' statt '1. Nebendiagnose: ' *InvalidData*: Wert, der Meldung verursacht (z.B. ungültiger Code)
Rückgabewert	0: Funktion erfolgreich ausgeführt

4.5.5 function MP_TestRec(HDL: Integer; Data: PChar; ShowWindow: Integer; MessageTypes: PChar; cLanguage: Integer = -1): Integer;

Ein MB-Datensatz mit oder ohne Zusatzdatensätze wird getestet.

Parameter	Beschreibung
HDL	Handle des Testfensters
Data	Pchar/* char-Pointer auf die Daten: MB-, MN-, MP-, MD- und MK-Records können in dieser Reihenfolge in Data übergeben werden. Die Records sind gemäss der BFS-Schnittstellendefinition aufgebaut. Verschiedene Records werden mit CR, LF oder CR + LF voneinander getrennt. Leerzeilen und unbekannte Datensätze werden dabei ignoriert
ShowWindow	0: MedPlaus-Meldungsfenster nicht anzeigen 1: MedPlaus-Meldungsfenster anzeigen
MessageTypes	Obsolet. Ein allfällig übergebener Parameter wird ignoriert
Rückgabewert	1 oder >1: Anzahl Fehler, Warnungen, Hints, Strukturfehler (unabhängig der Einstellung in MessageTypes) 0: Keine Fehler, Warnungen, Hints, Strukturfehler -1: Der übergebene Handle ist ungültig
cLanguage	0: Deutsch 1: Französisch

4.5.6 function MP_GetMessageList(HDL: Integer; var ErrCount: Integer; var WarnCount: Integer; var HintCount: Integer; var ControlCount: Integer; var StructCount: Integer; ErrList: PChar): Integer;

Gibt Anzahl und Liste der Meldungen von MP_TestRec() zurück.

Parameter	Beschreibung
HDL	Handle des Testfensters
ErrCount	Anzahl Fehler im Datensatz
WarnCount	Anzahl Warnungen im Datensatz
HintCount	Anzahl Hints im Datensatz
ControlCount	Wert 0, irrelevant
StructCount	Wert 0, irrelevant
ErrList	Übergibt alle Fehlermeldungen. Die einzelnen Meldungen werden mit CR + LF voneinander abgetrennt. ErrList muss (ErrCount + WarnCount) * 256 Bytes zur Verfügung stellen (siehe Rückgabewert von TestRec).
Rückgabewert	<0: Fehler 0: Funktion erfolgreich ausgeführt -1 = ungültiger Handle

4.5.7 function MP_Destroy(HDL: Integer): Integer;

Gibt die MedPlaus-Instanz/Ressourcen frei.

Parameter	Beschreibung
HDL	Handle der MedPlaus-Instanz
Rückgabewert	0: Funktion erfolgreich ausgeführt -1 = ungültiger Handle

4.5.8 function MP_VersionAppName(): PChar;

Name der Applikation

Parameter	Beschreibung
Rückgabewert	Name der Applikation

4.5.9 function MP_VersionNumber(): PChar;

Versionsnummer der Applikation

Parameter	Beschreibung
Rückgabewert	Versionsnummer der Applikation

4.5.10 function MP_VersionDate(): PChar;

Datum der Applikation

Parameter	Beschreibung
Rückgabewert	Datum der Applikation

4.6 MP6 DLL-DynTest

Die Datei mp6DllDynTest.exe[3] startet eine grafische Oberfläche, die das Testen der DLL-Aufrufe (dynamischen Bindung) anhand eines Beispieldatensatzes erlaubt.

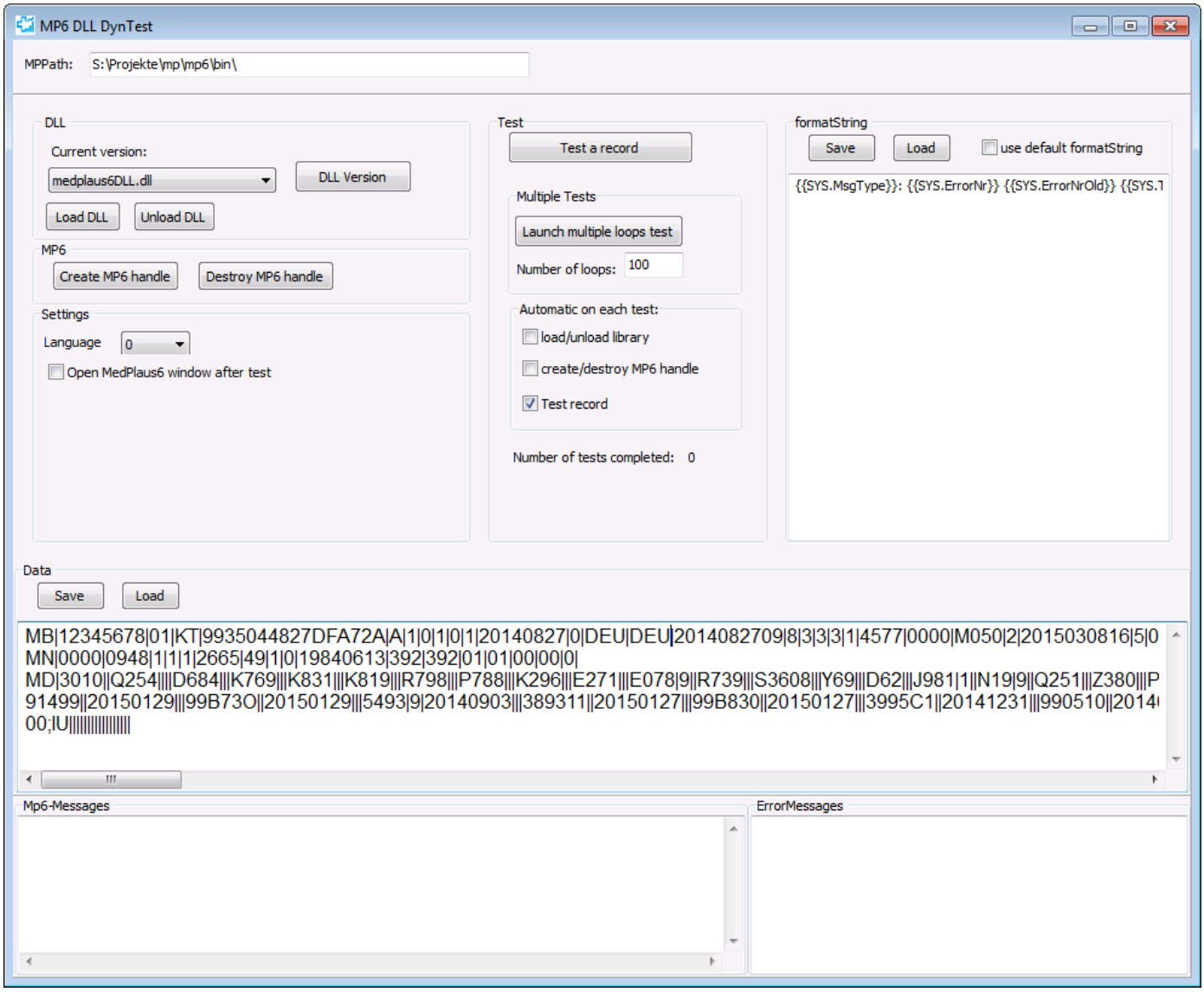

Abbildung 4.2: MP6 DLL-DynTest

[3]alternativ: mp6DllDynTest64.exe zum Testen der 64-Bit MedPlaus-DLL

4.6.1 Testvorbereitungen

Der grundsätzliche Ablauf ist der folgende:

1. Geben Sie Laufwerk und Pfad, in dem sich die MedPlaus DLL-Dateien befinden, an.
   Es muss unbedingt ein Backslash \ am Pfadende gesetzt werden (z. B. `c:\programme\mp6042\`)

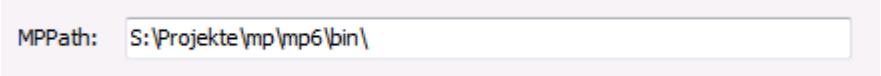

Abbildung 4.3: MP6 DLL-DynTest: Pfad

2. Wählen Sie danach die gewünschte MedPlaus-DLL aus
   - Default: medplaus6DLL.dll
   - Für Fehlersuche: medplaus6DLL_BigLog.dll
3. Laden Sie die DLL mit Klick auf *Load DLL*

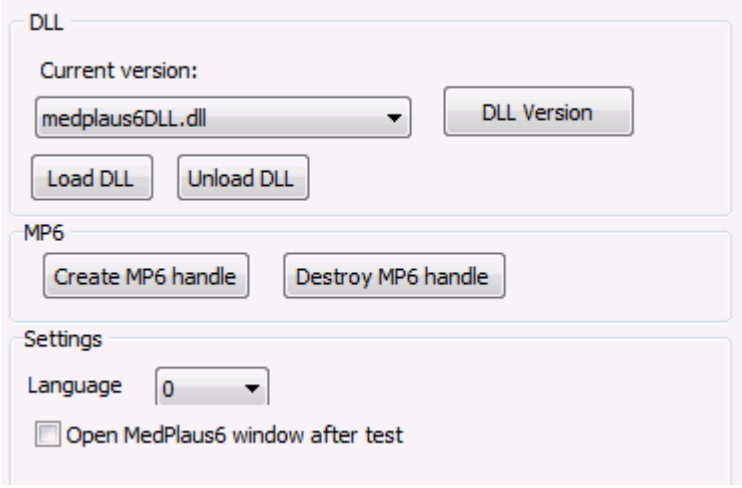

Abbildung 4.4: MP6 DLL-DynTest: Load DLL

4. Lassen Sie bei Bedarf weitere DLL-Informationen wie Version und Datum anzeigen (per Klick auf *DLL Version*)
5. Passen Sie die gewünschten Einstellungen unter 'Settings' an:
   a) die Sprache im Dropdown-Menü *Language*
      - 0: Deutsch
      - 1: Französisch
   b) Wenn die Option *Open MedPlaus6 window after test* aktiviert ist, werden Datensatz und Meldungen in einem eigenen MedPlaus-Fenster[4] angezeigt.
6. Erstellen Sie mit *Create MP6 handle* eine eindeutige Instanz

[4] mp6show.exe

4.6.2 Test ausführen

Abschnitt Test

Über *Test a record* wird ein Test einmalig ausgeführt.

- Der Testdatensatz befindet sich im Bereich 'Data'
- Die Resultate werden entweder im Bereich 'Messages' oder im separaten MedPlaus-Fenster angezeigt

Im Bereich 'Multiple Tests' kann ein Test beliebig oft in Folge ausgeführt werden. Der Wert unter 'Number of loops' bestimmt die Anzahl der Durchgänge.

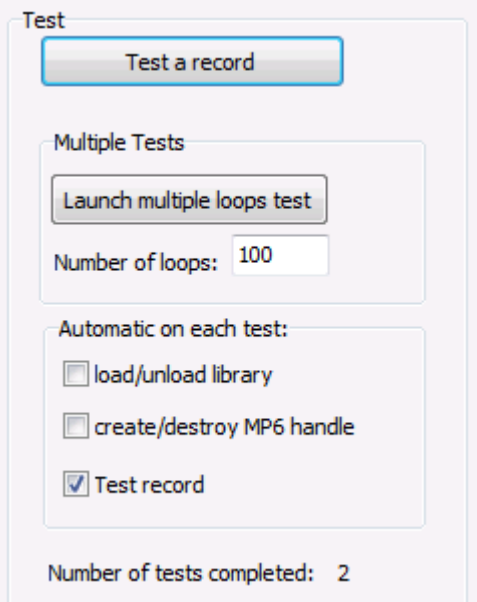

Abbildung 4.5: MP6 DLL-DynTest: ein-/mehrfache Test

Mit *Automatic on each test* können folgende Vorgänge automatisiert durchgeführt werden:

- DLL laden/entladen (*load/unload library*)
- Instanz erstellen/zerstören (*create/destroy MP6 handle*)
- Test des Datensatzes (*Test record*)

Bei mehrfacher Ausführung des Tests ist dies etwas langsamer, da die DLL jedes mal neu geladen wird.

Unterhalb der Testeinstellungen wird noch die Anzahl der abgeschlossenen Tests (*Number of tests completed:*) angezeigt.

Abschnitt Data

Der zu prüfende Datensatz (mit Zusatzdatensätzen) steht im Abschnitt 'Data'.

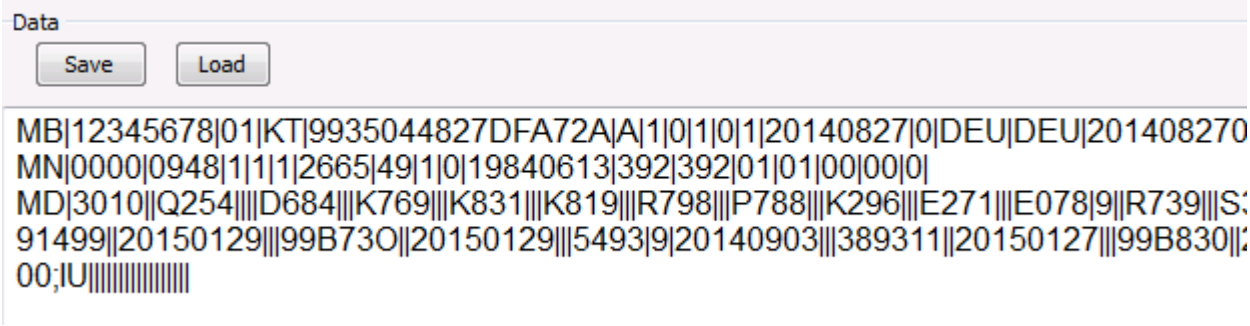

Abbildung 4.6: MP6 DLL-DynTest: Data-Bereich

Mit *Save* wird der Datensatz als Textdatei (mp6DllDynTestData.txt) gespeichert.
Mit *Load* wird der Inhalt von mp6DllDynTestData.txt eingelesen und dargestellt.

Diese Funktionen sind hilfreich, wenn Sie viele Änderungen am zu prüfenden Testdatensatz vornehmen. Sie können diese in einem Texteditor statt in der Test-Gui vornehmen.

Abschnitt Messages: Meldungen

Der linke Bereich des Abschnitts Meldungen zeigt MedPlaus 6-Meldungen (Fehler, Warnungen und Hinweise) an, die aus dem Test des Datensatzes resultieren.

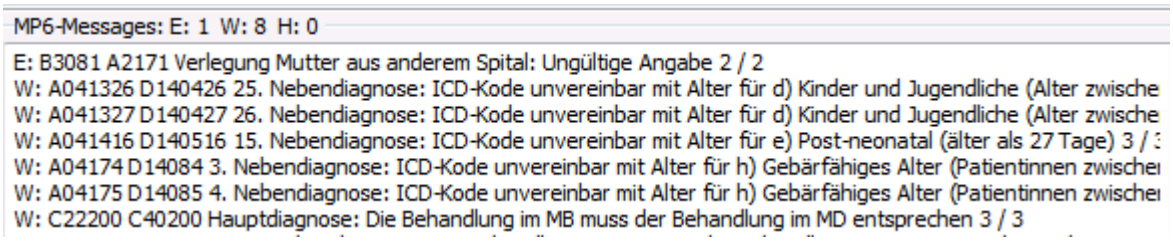

Abbildung 4.7: MP6 DLL-DynTest: MP6-Meldungen

Beachten Sie bitte:

- Im obersten Teil wird die Anzahl der angezeigten Fehler (E), Warnungen (W) und Hinweise (H) aufgelistet
- Die Meldungen sind nicht sortiert
- Das Ausgabeformat entspricht dem MedPlaus-Standard (default) oder dem selbst definierten Format (MP_setFormatString())

Abschnitt Messages: Fehlermeldungen

Der rechte Bereich des Abschnitts Meldungen zeigt Fehlermeldungen des Programms an.

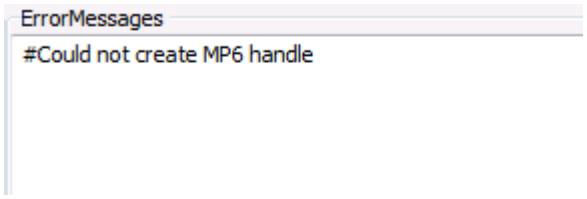

Abbildung 4.8: MP6 DLL-DynTest: Fehlermeldungen

4.6.3 Zusatzinformationen

Wechsel der DLL

Um die DLL zu wechseln, muss zuerst die alte entfernt werden. Nachfolgend das Vorgehen:

1. Klicken Sie auf *Destroy MP6 handle*, um diesen zu zerstören
2. Entfernen Sie die DLL per Klick auf *Unload DLL*
3. Laden Sie die neue DLL (s. Testvorbereitungen)

Wenn im Bereich *Automatic on each test* die Punkte *load/unload library* und *create/destroy MP6 handle* aktiviert sind, entfallen die ersten beiden Schritte. Es kann direkt eine neue DLL ausgewählt und mit dieser getestet werden.

Datenansicht im MedPlaus-Fenster

Ist *Open MedPlaus6 window after test* aktiviert, werden der getestete Datensatz und die Resultate in einem separaten Fenster ausgegeben. Im oberen Teil erscheinen die Inhalte, im unteren werden die gefundenen Fehler, Warnungen und Hinweise aufgelistet.

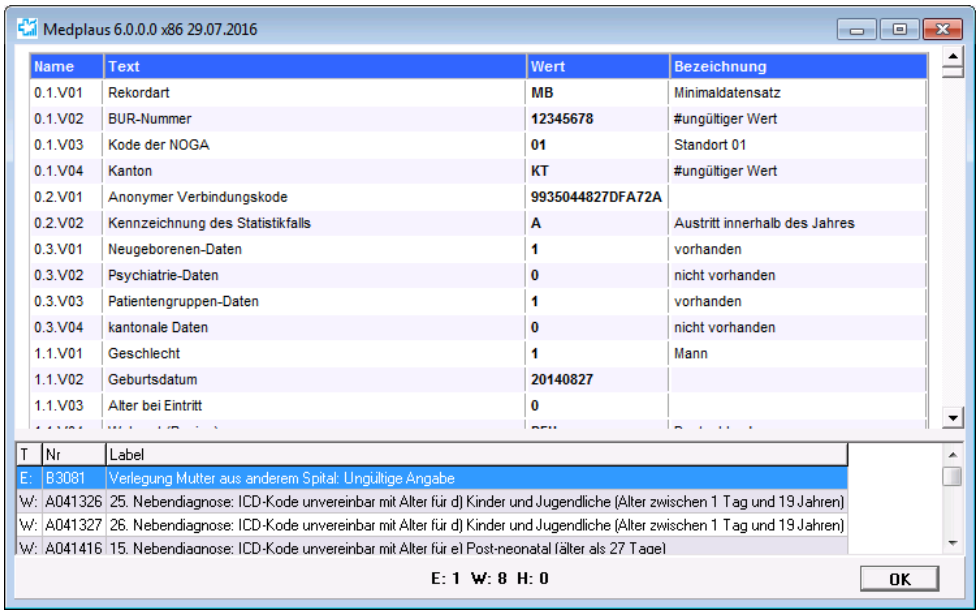

Abbildung 4.9: mp6show.exe

Die Titelzeile informiert über Version und Datum der MedPlaus-DLL.
Die Statuszeile listet die Anzahl der angezeigten Fehler (E), Warnungen (W) und Hinweise (H) auf.

Der Button *OK* schliesst das Fenster. Man befindet sich wieder auf der MP 6 DLL DynTest-Oberfläche (bzw. MP 6 DLL DynTest 64-Bit).

Abschnitt formatString

Der formatString ist ein neues Format, welches mit MedPlaus 6 eingeführt wird. Er erlaubt die individuelle Erstellung von Zeichenketten mit Konstanten und Variablen.

Detaillierte Informationen zum Format finden Sie im Abschnitt 'function MP_setFormatString()'.

Nutzung

- Individuelle Formatierung der Ausgabe der Meldungen im Bereich MP6-Messages der Test-Gui
- Individuelle Formatierung für Gebrauch mit der Funktion MP_GetMessageList()

Mit formatString können z. B. mit wenig Aufwand SQL-Statements generiert werden. Diese erlauben das Speichern und Auswerten von MedPlaus-Meldungen in einer Datenbank.

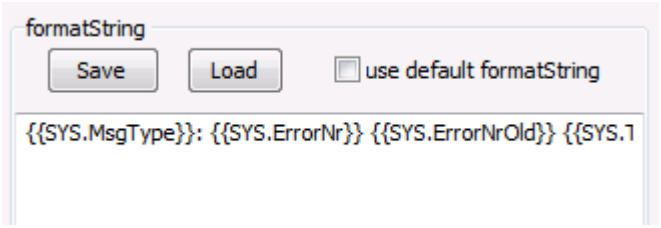

Abbildung 4.10: MP6 DLL-DynTest: formatString

Mit *Save* wird der Formatstring als Textdatei (mp6DllDynTestFormatString.txt) gespeichert.
Mit *Load* wird der Inhalt von mp6DllDynTestFormatString.txt eingelesen und dargestellt.

Diese Funktionen sind hilfreich, wenn Sie viele Änderungen am individuellen Formatstring vornehmen. Sie können diese in einem Texteditor statt in der Test-Gui vornehmen.

Wenn *use default formatString* aktiviert ist, wird der MedPlaus 6-Standard benutzt. Dieser entspricht demjenigen aus MedPlaus 5.

4.7 Fehlersuche

MedPlaus DLL wird mit schnellen, produktiven DLL-Versionen in 32-und 64-Bit ausgeliefert, die keine LOG-Dateien erstellen. Sollten Sie bei der DLL-Integration auf Probleme stossen, gehen Sie folgendermassen vor:

1. Sichern Sie die produktive DLL (medplaus6DLL.dll oder medplaus6DLL64bit.dll)
2. Benutzen Sie statt der produktiven MedPlaus 6 DLL die Version aus dem Verzeichnis \DLL_BigLog. Diese bietet eine umfassende LOG-Funktionalität, die hilft, das Problem zu ermitteln.
   a) Kopieren Sie die BigLog-DLL[5] in das Stammverzeichnis und benennen Sie sie um nach medplaus6DLL.dll.
   b) Sprechen Sie die DLL an.
   c) Analysieren Sie danach die Datei medplaus6DLL_debug.txt im Verzeichnis c:\temp[6].

   Ist das Problem gelöst, können Sie wieder die produktive MedPlaus 6 DLL-Version in das Stammverzeichnis kopieren.
3. Ist die Ursache des Problems weiterhin unklar, gehen Sie bitte folgendermassen vor:
   a) Löschen Sie den Inhalt der Datei medplaus6DLL_debug.txt im Verzeichnis c:\temp
   b) Sprechen Sie die BigLog-DLL erneut an
   c) Senden Sie uns den kompletten Inhalt der LOG-Datei per E-Mail an medplaus@freudiger.com

[5] analoges Vorgehen bez. 64-Bit Version, unter Anpassung der Dateinamen
[6] c:\temp muss bereits existieren, damit die Debug-Logdatei hineingeschrieben werden kann

5 Diverses

5.1 Hersteller-Support

Bei weiteren Fragen erhalten Sie Support von der Freudiger EDV-Beratung. Senden Sie eine genaue Problembeschreibung (evtl. mit Sourcecode-Ausschnitten und anonymisierten Printscreens) inkl. Angabe zu den benutzten Versionen an medplaus@freudiger.com.

Falls Sie sensible Daten übermitteln möchten, können Sie dies über unser BESTAT-Portal durchführen. Wenden Sie sich diesbezüglich an unsere Hotline (031 318 17 24).

5.2 Dokumenten-History

28.04.2020:	Nachführen der Dokumentation (Stand MedPlaus 6.0.4.2-DLL)
04.03.2020:	Nachführen der Dokumentation (Stand MedPlaus 6.0.4.1-DLL)
13.12.2019:	Nachführen der Dokumentation (Stand MedPlaus 6.0.4.0-DLL)
09.07.2019:	Nachführen der Dokumentation (Stand MedPlaus 6.0.3.1-DLL)
20.12.2018:	Nachführen der Dokumentation (Stand MedPlaus 6.0.3.0-DLL)

5.3 Versionierung

Version:	6.042
Letzte Änderung:	28.04.2020
Status:	Public