

Deutsche Rentenversicherung Bund	ELENA Kernprüfung	Bereich 0560
	Funktionsbeschreibung Einbettung der Prüffunktionen in Anwendersysteme Programmiersprache C	

1. Allgemein:

Im Rahmen der Neukonzeption der Datenerfassungs- und Übermittlungsverordnung (DEÜV) wurde in Übereinstimmung mit den Spitzenverbänden der Krankenkassen, der Bundesagentur für Arbeit sowie der Deutschen Rentenversicherung vereinbart, eine zentral durch die Datenstelle der Träger der Deutschen Rentenversicherung und der Deutschen Rentenversicherung Bund Träger gefertigte Fehlerprüfung mit entsprechenden Kernfunktionen bereit zu stellen.

Diese „Funktion“, die den jeweiligen Anwendern zur Mindestprüfung der Datensätze der DEÜV dienen soll, wird von den verschiedenen Benutzern in ihre eigene Systemumgebung eingebunden. Sie ist auf Datensatzebene angesiedelt und enthält die Prüfungen der Anlage 9 des Gemeinsamen Rundschreibens.

Auf der Basis der Kernprüfung der DEÜV wurde die Kernprüfungskomponente für das ELENA-Verfahren entwickelt. Diese enthält grundsätzlich die Prüfungen der Anlage 5 der ELENA-Verfahrensbeschreibung mit Ausnahme der anwenderspezifischen Prüfungen bei der ZSS (z. B. MVDSz58).

Die Realisierung der Funktion erfolgt in C und müsste für alle gängigen Plattformen einsetzbar sein. Eine mögliche Einbettung der Funktion in die anwendereigene Systemumgebung ist in der Anlagen A beispielhaft dargestellt. Der Ablauf der Prüfungen ergibt sich aus der Anlage B.

Bearbeiter: Schmitt	Stand: 15.12.2009	Version: 01	Seite: 1 von 5
---------------------	-------------------	-------------	-------------------

Deutsche Rentenversicherung Bund	ELENA Kernprüfung	Bereich 0560
	Funktionsbeschreibung Einbettung der Prüffunktionen in Anwendersysteme Programmiersprache C	

2. Anlage A: Schnittstellenbeschreibung

Bei der nachfolgend dargestellten Beschreibung handelt es sich um die Schnittstelle zwischen der allgemein gültigen Prüffunktion und dem Anwender (Arbeitgeber, Dienstleister). Der Anwender bettet, wie bereits erwähnt, die Funktion in seine Verfahren ein und übergibt den jeweils zu prüfenden Datensatz. Dabei ist darauf zu achten, dass der übergebene Datensatz in einem ausreichend großen Datenbereich steht damit ggf. Fehlerbausteine angehängt werden können. Der zusätzliche Platzbedarf liegt bei 720 Bytes (9 Fehlerbausteine à 80 Bytes).

Für die Funktion steht die Bibliothek en055c.dll zur Verfügung, welche in die eigene Anwendung eingebunden wird (u. U. ist zusätzlich die Datei en055c.lib notwendig). Für das Einbinden von DLL-Bibliotheken stehen im Internet zahlreiche Tutorials zur Verfügung (z.B. http://www.codeguru.com/cpp/cpp/cpp_mfc/tutorials/article.php/c9855/).

Da der Datensatz durch die Prüffunktion eventuell verändert wird - z.B. werden im Fehlerfall die Längfelder sowie die Felder FEKZ und FEAN verändert - sollte zur Verfahrenssicherheit das einbettende Verfahren dafür sorgen, dass der Originaldatensatz erhalten bleibt.

Mit Hilfe des Operationscodes (**op_code**) wird der Prüfumfang des Datensatzes festgelegt. Zum Beispiel erfolgen die Prüfungen der Datensätze des Arbeitgebers durch den **op_code** = „AGELE“.

Nach erfolgter Prüfung ist es dem Benutzer überlassen, individuell auf den Returncode (**rpc**) zu reagieren.

Der Aufruf dieser Funktion „**plausi**“ könnte in C beispielsweise folgendermaßen aussehen: (Implizites Linken der DLL):

```
#include "plausi.h"

void main(){
    /* Benötigte Variablen */
    char version[20];
    char* op = "AGELE";
    char ptr[1000] = "Teststring";
    char datum[21] = "20100101000000000000";
    char rpc[5];
    char ver[3];

    printf("ELENA DLL Beispiel\n");

    /* Ausgabe der ELENA DLL Version */
    getVersion(version);
    printf("DLL Version: %s\n", version);

    /* Aufruf der Plauibilitätsprüfung */
    plausi(op, ptr, datum, rpc, ver);

    /* Ausgabe der Ergebnisse */
    printf("Datensatz mit fehlerbausteinen: %s\n", ptr);
    printf("Returncode: %s\n", rpc);
    printf("Version der Spezifikation der Kernprüfung: %s\n", ver);
}
```

Bearbeiter: Schmitt	Stand: 15.12.2009	Version: 01	Seite: 2 von 5
---------------------	-------------------	-------------	-------------------

Deutsche Rentenversicherung Bund	ELENA Kernprüfung	Bereich 0560
	Funktionsbeschreibung Einbettung der Prüffunktionen in Anwendersysteme Programmiersprache C	

Plausi.h:

```

#ifdef __PLAUSI_H__
#define __PLAUSI_H__

/* Windows */
#ifdef KERNPRUEF_DLL_EXPORTS
#define PLAUSI_DLL __declspec(dllexport)
#else
#define PLAUSI_DLL __declspec(dllimport)
#endif

#ifdef _WINDLL
#define PLAUSI_DLL
#endif

#ifdef __cplusplus
extern "C" {
#endif

    /****** C-Schnittstelle *****/

    /* Aufruf der ELENA Kernprüfung:
       op: Verfahrensmerkmal
       ptr: Datensatz, der ggf um Fehlerbausteine erweitert wird
       datum: Datum im Format jhjjmmtt (Datum)
                                   hhmss (Uhrzeit)
                                   msmsms (Millisekunden)
       rcp: Returncode -> wird von plausi gesetzt
       ver: ProgrammVersion -> wird von plausi gesetzt*/
    PLAUSI_DLL void plausi(char *op, char*ptr,
        char datum[20], char*rcp, char*ver);

    /* Ermittelt die Version der Kernprüfung */
    PLAUSI_DLL void getVersion(char version[20]);

    /******
**/

#ifdef __cplusplus
}
#endif

#endif

...

```

Bearbeiter: Schmitt	Stand: 15.12.2009	Version: 01	Seite: 3 von 5
---------------------	-------------------	-------------	-------------------

Deutsche Rentenversicherung Bund	ELENA Kernprüfung	Bereich 0560
	Funktionsbeschreibung Einbettung der Prüffunktionen in Anwendersysteme Programmiersprache C	

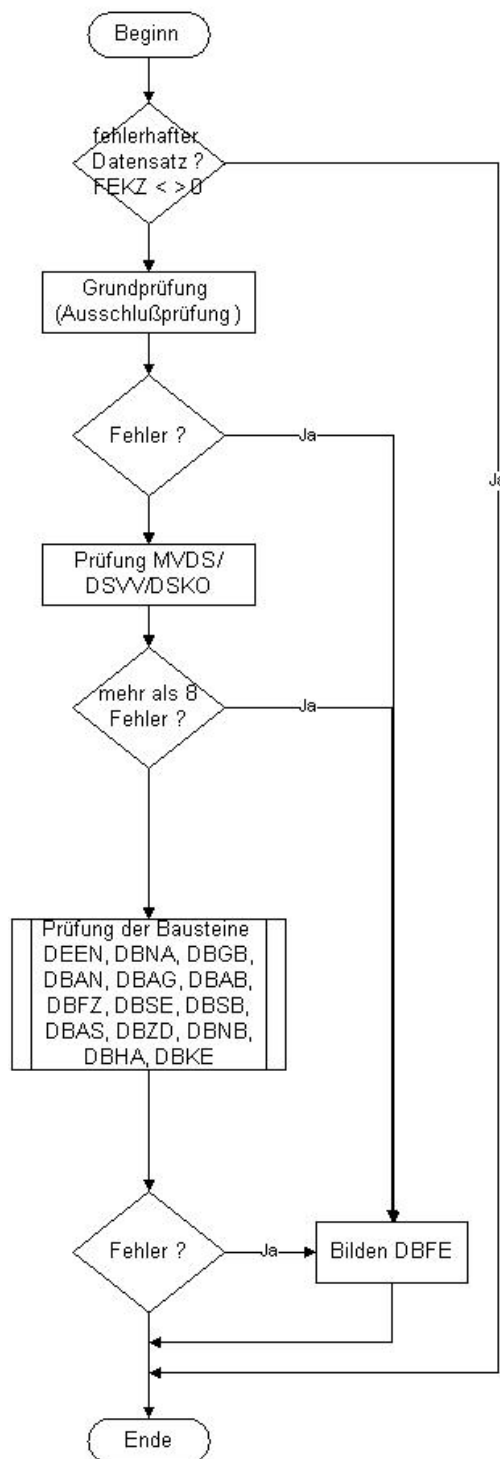
LG	Name	Erläuterung
⇒ 006	OP-CODE Techn. Name: op Typ: Zeiger auf char	1. Parameter: Operationscode OP-Code <i>Der OP-Code entspricht dem Verfahrensmerkmal (VFMM) im Vorlaufsatz zur DEÜV wie z.B.</i> <i>AGELE = Datensatz ist zu prüfen - Prüfumfang Arbeitgeber zur ZSS (MVDS)</i> <i>AGENR = Datensatz ist zu prüfen - Prüfumfang Arbeitgeber zur ZSS (DSVV)</i> <i>ZSELE = Datensatz ist zu prüfen - Prüfumfang ZSS zum Arbeitgeber</i> <i>ZSENR = Datensatz ist zu prüfen - Prüfumfang ZSS zum Arbeitgeber</i>
⇒ var var	MVDS / DSVV / DSKO DBxx Techn. Name: ptr Typ: Zeiger auf char	2. Parameter: zu prüfender Datensatz (bestehend aus) Fester Teil eines Datensatzes Variabler Teil ggf. mit einem oder mehreren Datenbausteinen. Nach der Prüfung wurde der Datensatz ggf. um Fehlerbausteine ergänzt.
⇒ 020	VERARBEITUNGS- DATUM Techn. Name: datum Typ: char[20]	3. Parameter: Systeminformationen Das Verarbeitungsdatum, das für die Prüfungen innerhalb der Datenbausteine benötigt wird. Das Feld ist nur mit numerischen Werten zu versorgen. Die Form lautet: <i>jhjmmmtt (Datum)</i> <i>hhmmss (Uhrzeit)</i> <i>msmsms (Millisekunden) - optional, sofern systemseitig lieferbar, jedoch mind. Nullen</i>
⇒ 005	Copy RETURNCODE Techn. Name: rcp Typ: Zeiger auf char	4. Parameter: Returncode Ergebnis der Operation <i>0000 = OK</i> <i>1000 = Fehler</i> <i>1001 = Verfahrensmerkmal nicht AGELE/AGENR</i> <i>1002 = Kennung nicht MVDS/DSVV/DSKO</i> <i>1100 = Datensatz wurde nicht geprüft, da bereits Fehlerkennzeichen gesetzt war</i> <i>1200 = nicht vorgesehen</i> <i>2000 = nicht vorgesehen</i> <i>9000 = nicht vorgesehen</i>
⇒ 003	PROGRAMMVERS Techn. Name: progvers Typ: Zeiger auf char	5. Parameter: PROGRAMMVERSION Dieser Rückgabeparameter dokumentiert die Versionsnummer des Kernprüfungsprogramms mit der die ELENA-Datensätze geprüft wurden.

Bearbeiter: Schmitt	Stand: 15.12.2009	Version: 01	Seite: 4 von 5
---------------------	-------------------	-------------	-------------------

Deutsche Rentenversicherung Bund	ELENA Kernprüfung	Bereich 0560
	Funktionsbeschreibung Einbettung der Prüffunktionen in Anwendersysteme Programmiersprache C	

3. Anlage B: Programmablaufplan

Abbildung 1 Programmablaufplan



Das Ablaufdiagramm beschreibt die allgemeine Fehlerprüfungen analog der DEÜV, die für alle Anwender identisch sind. Dem Programm ist jeweils nur ein Datensatz zu Prüfung zu übergeben. Vorlauf- und Nachlaufsätze werden nicht geprüft und sind nicht zu übergeben.

Ist der eingehende Datensatz fehlerhaft (FEKZ < > 0), wird das Programm beendet und der Datensatz ungeprüft an den Anwender zurückgegeben.

Fehlerfreie Datensätze werden zu Beginn einer formalen Prüfung unterzogen, die den grundsätzlichen Aufbau überprüft.

Enthält der Datensatz Fehler in der Länge, Kennung, Verfahrenskennzeichen des Datensatzes oder der Reihenfolge der Datenbausteine, wird die Prüfung beendet und der Datensatz mit einem entsprechenden Return-Code an den Anwender zurückgegeben.

Nach der Grundprüfung wird der MVDS/DSVV/DSKO im Einzelnen geprüft.

Bei mehr als acht Fehlern wird die Prüfung beendet und der Abschnitt zur Bildung der Datenbausteine Fehler aufgerufen.

Nach der Prüfung des MVDS/DSVV/DSKO werden die vorhandenen Datenbausteine gem. der Reihenfolge im Datensatz geprüft. Bei mehr als acht Fehlern innerhalb des Datensatzes wird die Prüfung beendet.

Zum Schluss wird geprüft, ob der Abschnitt zur Bildung der Datenbausteine Fehler aufzurufen ist. In diesem Abschnitt werden für jeden Fehler aus der Fehler-Nummer und dem Texte ein Datenbaustein Fehler gebildet und an den Datensatz angehängt.

Anschließend wird der geprüfte Datensatz ggf. mit Datenbausteinen Fehler und einem Return-Code an den Anwender zurückgegeben.