

ERiC 38.2.4.0

ERiC-Tutorial

Inhalt

1	EINLEITUNG	4
1.1	Typographische Konventionen	4
1.2	Zweck des Dokuments und Zielgruppe	5
1.3	Was ist die ERiC API?	5
1.4	Wie kann die ERiC API verwendet werden?	5
1.5	Aufbau des ERiC-Tutorials	6
1.6	Abkürzungen	6
2	VORBEREITENDE MAßNAHMEN	7
2.1	Voraussetzungen.....	7
2.2	Registrieren und Anmelden für das Herunterladen der ERiC API	7
2.3	Herunterladen der ERiC API	9
2.4	Installieren der ERiC API	10
2.5	Die XML-Datei und die PDF-Dokumente des ERiC-Tutorials	11
2.6	ELSTER Forum für Entwickler	12
2.7	Das Beispielprogramm „ericdemo“ kompilieren und ausführen	13
2.8	Das Beispielprogramm „ericdemo“ und die Kernfunktionen der ERiC API	14
2.9	Verwendung der ERiC API-Referenz	18
3	VERARBEITUNG VON JAHRESSTEUERN AM BEISPIEL EST	19
3.1	Erste Aufgabe: Die ELSTER-XML Eingangsdaten am Beispiel ESt erstellen	20
3.1.1	Von den Daten des Steuerpflichtigen zu den ELSTER Feldern	21
3.1.2	Die Struktur der ELSTER eXML-Daten für die ESt.....	23
3.1.3	Bedeutung der Feldkennungen, -formate und -texte	27
3.1.4	Die ESt eXML Daten erstellen	28
3.2	Zweite Aufgabe: Verarbeitung der Steuerdaten	34
3.2.1	Schritt 1 - Implementieren Sie das empfohlene Vorgehen der Initialisierung beim Laden der ERiC Komponenten	35
3.2.2	Schritt 2 - Aufruf der ERiC API.....	36
3.2.3	Schritt 3 - Plausibilitätsprüfungen	41
3.2.4	Schritt 4 - Übermittlung an den ELSTER Annahmeserver.....	43
3.2.5	Schritt 5 - PDF-Erstellung	45
3.3	Dritte Aufgabe: Neue und geänderte Felder in die Steuersoftware integrieren	46
3.3.1	Beispiel zum VZ-Wechsel für die ESt von 2016 auf 2017	47
3.4	Vierte Aufgabe: Den TransferHeader mit EricCreateTH() erstellen	49
3.4.1	Schritt 1 - XML-Eingabedatei erstellen	49
3.4.2	Schritt 2 - EricCreateTH() in ericdemo aufrufen.....	50
4	ERIC OPTIMAL IN DIE STEUERSOFTWARE AM BEISPIEL EST 2020 INTEGRIEREN	52
5	FORTSCHRITTCALLBACKS AM BEISPIEL ERICDEMO IMPLEMENTIEREN	55
6	DATENABHOLUNG MIT ANHÄNGEN AM BEISPIEL „SONSTIGENACHRICHTEN“	58
6.1	Datenversand der Datenart SonstigeNachrichten mit defektem Anhang	59
6.2	XML-Anfrage erstellen, ob Daten abgeholt werden können	61
6.3	Abholung der bereitgestellten Daten	63
6.4	Dekodieren der Rückmeldung.....	64
6.5	Bestätigung der abgeholten Daten.....	65


7	ZUSAMMENFASSUNG	66
8	TABELLENVERZEICHNIS	67
9	ABBILDUNGSVERZEICHNIS.....	68

1 Einleitung

1.1 Typographische Konventionen

In diesem Dokument werden besondere Aspekte durch Formatierung hervorgehoben.

Tabelle 1-1 Typographische Konventionen

Formatierungs-konvention	Informationstyp
<i>EricBearbeiteVorgang()</i>	API-Funktionen und Variablen werden kursiv dargestellt.
<div style="border: 1px solid black; padding: 2px; display: inline-block;">ERICAPI DECL eric_fehler_t</div>	Code(-Beispiele) und Daten(-Strukturen) befinden sich in einem grauen Kasten.
<div style="border-top: 1px solid black; border-bottom: 1px solid black; padding: 5px; display: inline-block;"> HINWEIS: Hinweistext</div>	Hinweistexte werden oben und unten von einer Linie begrenzt. An der linken Seite zeigt ein Hand-Symbol auf den fett gedruckten Text „Hinweis:“.
<u>API-Referenz</u>	Hyperlinks sind blau unterstrichen.

1.2 Zweck des Dokuments und Zielgruppe

Das vorliegende Tutorial führt Sie anhand eines Beispiels einer Einkommensteuer (ESt) Schritt für Schritt durch die Steuersoftwareentwicklung mit ERiC (ELSTER Rich-Client).

Das Tutorial wendet sich an Softwareentwickler, die erstmalig eine Steuersoftware erstellen und hierzu ERiC einsetzen. Grundkenntnisse in der Softwareentwicklung werden vorausgesetzt, die verwendete Programmiersprache in diesem Tutorial ist C++.

Das Tutorial ist weitestgehend plattformunabhängig. Das ESt-Beispiel wird an der gebräuchlichsten Desktopplattform Windows demonstriert, Linux-, AIX- und macOS-Entwickler können analog vorgehen.

1.3 Was ist die ERiC API?

Die ERiC API ist eine Schnittstellenspezifikation, die die Erstellung von ELSTER-kompatibler Clientsoftware für Steuer-, Finanz- und Lohnbuchhaltung unterstützt. Die ERiC API steht allen auf www.elster.de registrierten Softwareherstellern zur Verfügung und ist für die Betriebssystemplattformen Microsoft Windows, macOS, Linux und AIX verfügbar. Lesen Sie im EHB¹, welche Steuerverfahren und -arten von der ERiC API unterstützt werden.

1.4 Wie kann die ERiC API verwendet werden?

Die ERiC API kann von der eingesetzten Programmiersprache verwendet werden, wenn diese das Einbinden von C-Bibliotheken unterstützt, das sind z.B. C++ oder Java mit JNA. Zur Integration in eine Anwendung können die ERiC Bibliotheken einerseits vom Linker der jeweiligen Entwicklungsumgebung eingebunden werden oder andererseits dynamisch unter Verwendung von "LoadLibrary" bzw. "dlopen" geladen werden. Das dynamische Laden können Sie am C++ Beispielprogramm „ericdemo“ sehen.

¹ siehe EHB, Kap. „Unterstützte Fachverfahren und Daten- / Steuerarten“

1.5 Aufbau des ERiC-Tutorials

Das Kap. [2 Vorbereitende Maßnahmen](#) beschreibt die Voraussetzungen, damit Sie erfolgreich mit ERiC ein Steuersoftwareprogramm erstellen können. Das ist beispielsweise die Installation und Konfiguration der IDE.

Das Kap. [3 Verarbeitung von Jahressteuern am Beispiel ESt](#) führt Sie mit einem ESt-Beispiel durch die Integration des ERiC in die Steuersoftware. Die Unterkapitel enthalten die Aufgaben:

- Kap. [3.1 Erste Aufgabe: Die ELSTER-XML Eingangsdaten am Beispiel ESt erstellen](#)
- Kap. [3.2 Zweite Aufgabe: Verarbeitung der Steuerdaten](#)
- Kap. [3.3 Dritte Aufgabe: Neue und geänderte Felder in die Steuersoftware integrieren](#)
- Kap. [3.4 Vierte Aufgabe: Den TransferHeader mit EricCreateTH\(\) erstellen](#)

Das Kap. [4 ERiC optimal in die Steuersoftware am Beispiel ESt 2020](#) zeigt am Beispiel ESt 2020, welche ERiC Plugins und Bibliotheken bei der Integration in die Steuersoftware notwendig sind bzw. weggelassen werden können und welche Vorteile sich daraus ergeben.

Das Kap. [5 Fortschrittcallbacks am Beispiel ericdemo implementieren](#) demonstriert die Verwendung von Fortschrittcallbacks.

Das Tutorial wird vom Kap. [7 Zusammenfassung](#) abgeschlossen.

1.6 Abkürzungen

EHB	ERiC-Entwicklerhandbuch.pdf
ERiC	ELSTER Rich Client
ESt	Einkommenssteuer
eXML	einheitliches ELSTER-XML Siehe EHB, Kap. „Einführung in die neue Nutzdatenstruktur“.
VZ	Veranlagungszeitraum

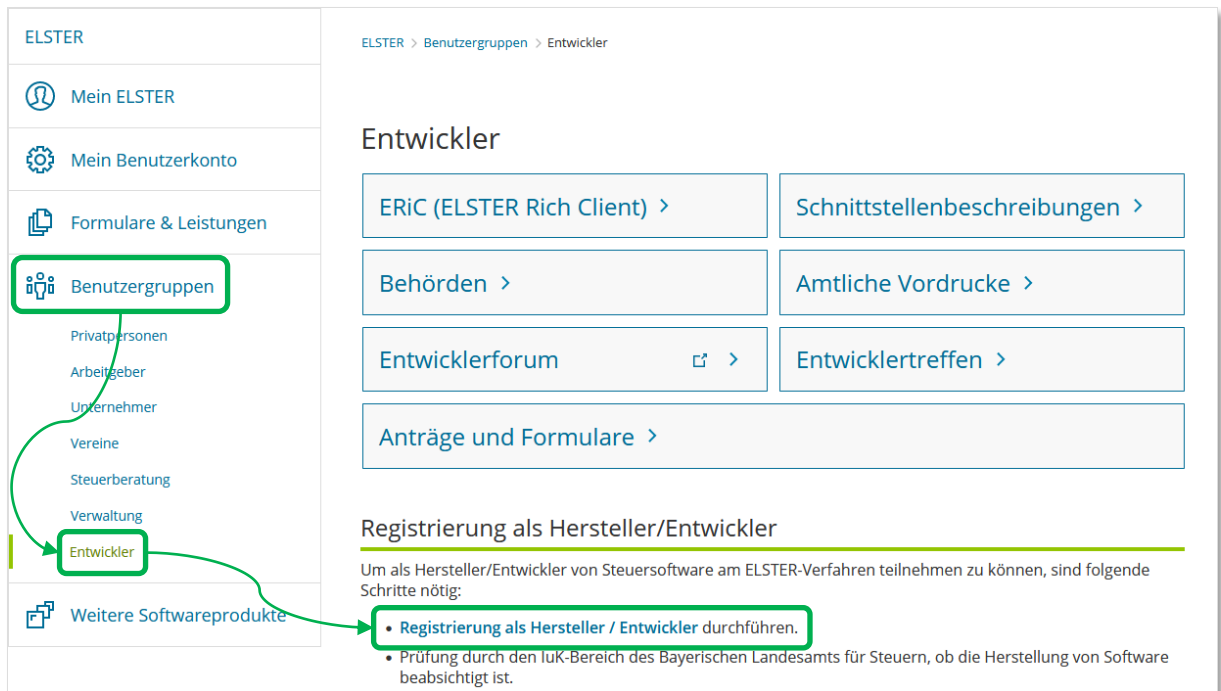
2 Vorbereitende Maßnahmen

2.1 Voraussetzungen

Es wird vorausgesetzt, dass auf dem Rechner des Steuersoftware-Entwicklers Microsoft Windows als Betriebssystem, das benötigte Visual C++ Redistributable Package und eine Microsoft Visual Studio Edition installiert sind.²

2.2 Registrieren und Anmelden für das Herunterladen der ERiC API

1. Registrieren Sie sich auf www.elster.de → **Benutzergruppe** → **Entwickler** als Softwarehersteller.



ELSTER

Mein ELSTER

Mein Benutzerkonto

Formulare & Leistungen

Benutzergruppen

- Privatpersonen
- Arbeitgeber
- Unternehmer
- Vereine
- Steuerberatung
- Verwaltung
- Entwickler**

Weitere Softwareprodukte

ELSTER > Benutzergruppen > Entwickler

Entwickler

ERIC (ELSTER Rich Client) > Schnittstellenbeschreibungen >

Behörden > Amtliche Vordrucke >

Entwicklerforum > Entwicklertreffen >

Anträge und Formulare >

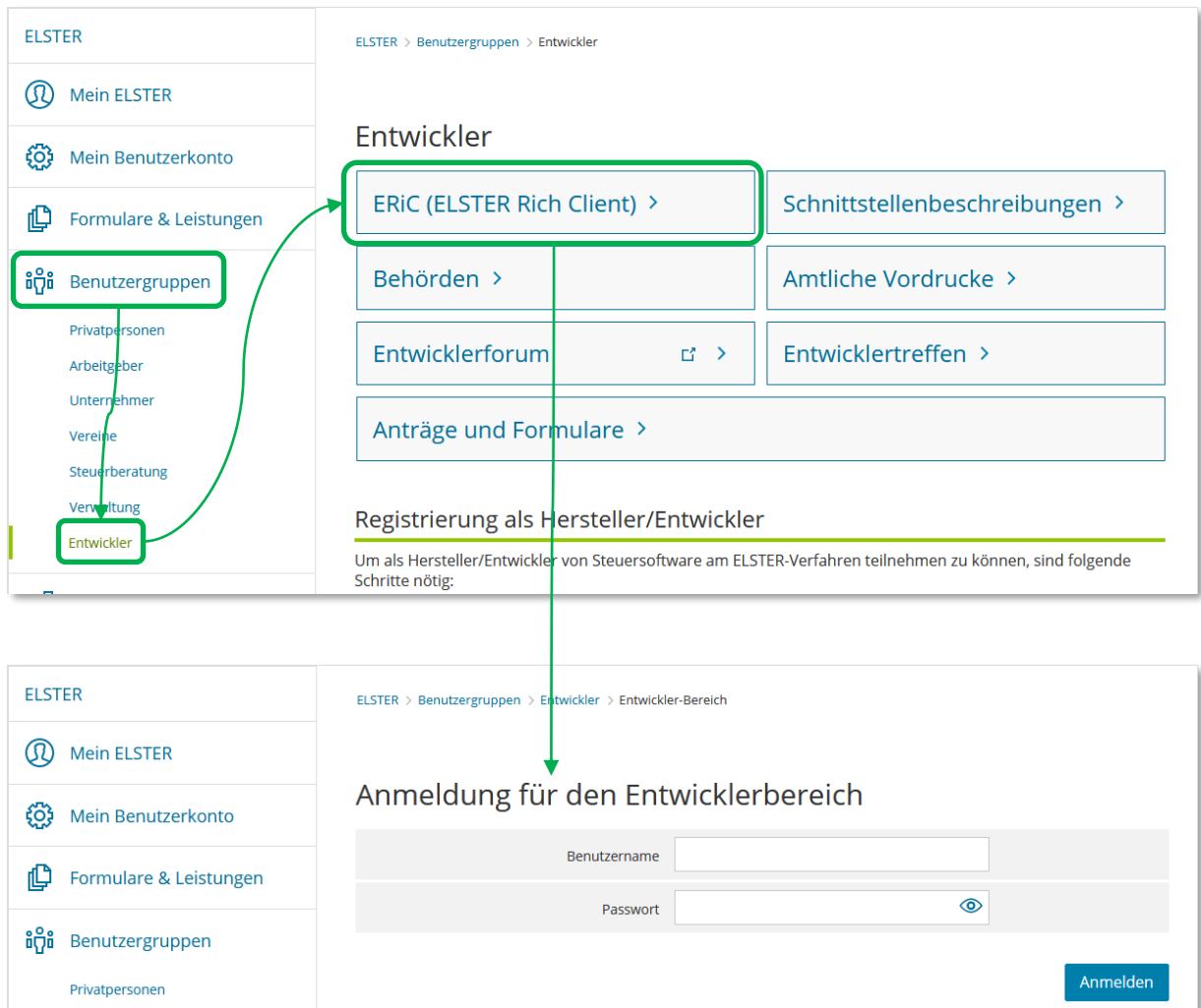
Registrierung als Hersteller/Entwickler

Um als Hersteller/Entwickler von Steuersoftware am ELSTER-Verfahren teilnehmen zu können, sind folgende Schritte nötig:

- **Registrierung als Hersteller / Entwickler durchführen.**
- Prüfung durch den IuK-Bereich des Bayerischen Landesamts für Steuern, ob die Herstellung von Software beabsichtigt ist.

² Siehe EHB (ERiC-Entwicklerhandbuch.pdf), Kap. "Mindestanforderungen an benötigte Software"

2. Melden Sie sich auf www.elster.de → **Benutzergruppen** → **Entwickler** → **ERiC (ELSTER Rich Client)** mit Ihren Hersteller-Logindaten an.



The image consists of two screenshots of the ELSTER website interface, illustrating the steps to reach the developer registration page.

Top Screenshot: The left sidebar shows the navigation menu with 'Benutzergruppen' highlighted. The main content area is titled 'Entwickler' and contains several links: 'ERiC (ELSTER Rich Client) >', 'Schnittstellenbeschreibungen >', 'Behörden >', 'Amtliche Vordrucke >', 'Entwicklerforum >', 'Entwicklertreffen >', and 'Anträge und Formulare >'. Below these links is a section titled 'Registrierung als Hersteller/Entwickler' with a sub-header 'Um als Hersteller/Entwickler von Steuersoftware am ELSTER-Verfahren teilnehmen zu können, sind folgende Schritte nötig:'.

Bottom Screenshot: The left sidebar shows the navigation menu with 'Benutzergruppen' highlighted. The main content area is titled 'Anmeldung für den Entwicklerbereich' and contains a login form with fields for 'Benutzername' and 'Passwort', and a blue 'Anmelden' button.

Nach dem Klick auf **Anmelden** wird die Seite www.elster.de/elsterweb/entwickler/infoseite/eric angezeigt.

3. Lesen Sie sorgfältig die Hinweise auf dieser Seite.
4. Klicken Sie für das gewünschte Release auf **Lizenzvertrag anzeigen**, lesen Sie ihn und klicken Sie dann auf **Lizenzvertrag akzeptieren**.
- Sie können nun mit dem Herunterladen der Dateien für die ERiC API beginnen, wie im Folgenden beschrieben.

2.3 Herunterladen der ERiC API

1. Laden Sie für das vorliegende ERiC-Tutorial die Dokumentation, ein 64-Bit ERiC-Release für Windows und die amtlichen Vordrucke herunter:³

- **Dokumentation:** ERiC-*<Version>*-Dokumentation.zip
Dieses Paket beinhaltet im Verzeichnis Dokumentation\ das (ERiC-Entwicklerhandbuch.pdf (EHB), auf das in diesem Tutorial häufig verwiesen wird. Es ist die zentrale Dokumentation und beinhaltet detaillierte Beschreibungen für alle Themen, die für die Anwendungsentwicklung mit ERiC relevant sind.
- **ERiC-Pakete:** ERiC-*<Version>*-Windows-x86_64.jar
Die Windows jar-Datei ist ein Archiv und keine Bibliothek. Ein Entpackprogramm, z.B. 7-Zip, wird benötigt, um die darin enthaltenen ERiC-Bibliotheken zu extrahieren.
- **Amtliche Vordrucke:** Vordrucke_*_ERIC-*<Version>*.zip
Nachfolgend werden die ESt-Vordrucke für den Veranlagungszeitraum (VZ) 2020 benötigt. Die im Tutorial verwendeten Vordrucke sind in der Datei Vordrucke_archive_ERiC-*<Version>*.zip enthalten.

2. Navigieren Sie zu **Schnittstellenbeschreibungen**.



3. Laden Sie aus dem Entwickler-Bereich [Schnittstellenbeschreibungen und Dokumentation](#) die folgenden Dokumentationen herunter:

- ELSTER-Fehlerliste
- Dokumentation Steuernummern- und Identifikationsnummernprüfung






Zum Durcharbeiten des Tutorials benötigen Sie weitere Softwarepakete. Die passende Version der weiteren Softwarepakete kann im EHB, Kap. „Ergänzende Downloadpakete“ nachgelesen werden.

³ siehe EHB, Kap. „Inhalt des ERiC“

2.4 Installieren der ERiC API

Entpacken Sie die heruntergeladenen Pakete in dasselbe Verzeichnis, z.B.

- C:\ERiC-Entwicklung
- Ergebnis:

(C:) > ERiC-Entwicklung	
Name	Typ
 archive	Dateiordner
 ERiC-36.2.4.0	Dateiordner
 META-INF	Dateiordner
 Fehlerliste.pdf	Adobe Acrobat Document
 Fehlerliste.xml	XML-Dokument



HINWEIS:

Im weiteren Verlauf des Tutorials wird dieser Pfad zusammen mit dem ERiC-Basisverzeichnis zur besseren Lesbarkeit weggelassen.

Beispiel:

- Gesamter Pfad: „C:\ERiC-Entwicklung\ERIC-<version>\Dokumentation“
- Verkürzte Schreibweise: „Dokumentation“

2.5 Die XML-Datei und die PDF-Dokumente des ERiC-Tutorials

Zum ERiC-Tutorial der ERiC API gehören folgende Dateien und Dokumente:

Tabelle 2-1 Dateien und Dokumente des ERiC-Tutorials

Datei / Dokument	Verzeichnis Beschreibung
ERiC-Tutorial.pdf	Dokumentation\Tutorial Dieses Dokument
ESt_2020-Beispiel_Loesung.xml	Dokumentation\Tutorial\Beispiele XML-Lösungsdatei des ESt-Beispiels.
ericprint.pdf	Dokumentation\Tutorial\Beispiele PDF-Lösungsdatei des ESt-Beispiels.
MitteilungAbholungAnfrage.xml	Dokumentation\Tutorial\Beispiele XML-Lösungsdatei des ElsterDatenabholung-Beispiels.
SonstigeNachrichten_Rueckmeldungtest.xml	Dokumentation\Tutorial\Beispiele XML-Lösungsdatei des ElsterDatenabholung-Beispiels.



HINWEIS:

Halten Sie zusätzlich das EHB (ERiC-Entwicklerhandbuch.pdf) bereit, da sich das Tutorial vertiefend darauf bezieht.

Sie finden die Datei unter: Dokumentation\ERiC-Entwicklerhandbuch.pdf

2.6 ELSTER Forum für Entwickler

Im Herstellerforum können Sie nach weiterführenden Informationen zu Problemen suchen, die andere schon gelöst haben.⁴

1. Rufen Sie im Browser die Seite <https://forum.elster.de/herstellerforum> auf.
2. Melden Sie sich mit Ihren Hersteller-Logindaten an.

Startseite nach dem Login:



⁴ siehe EHB, Kap. "Probleme und mögliche Ursachen"

2.7 Das Beispielprogramm „ericdemo“ kompilieren und ausführen

Im vorangegangenen Kapitel haben Sie die Voraussetzungen erfüllt, um das mitgelieferte C++-Programmbeispiel „ericdemo“ übersetzen und starten zu können. In den nächsten Schritten führen Sie dies aus.

Kompilieren und Linken von „ericdemo“:

1. Navigieren Sie im Windows-Explorer nach „Windows-x86_64\Beispiel\ericdemo-cpp\“
2. Doppelklicken Sie auf die Datei „ericdemo-2017-x64.sln“, um das Beispielprojekt „ericdemo“ in der Microsoft IDE zu öffnen.
3. Übersetzen Sie das Beispielprojekt „ericdemo“.
4. Überprüfen Sie, ob das ausführbare Programm „ericdemo.exe“ in dem Verzeichnis „Windows-x86_64\Beispiel\ericdemo-cpp\x64\Debug\“ zu finden ist.

Ausführen von „ericdemo“:

1. Navigieren Sie in das Verzeichnis „Windows-x86_64\Beispiel\ericdemo-cpp\“
2. Doppelklicken Sie die Batchdatei „startedemo-x64.bat“.



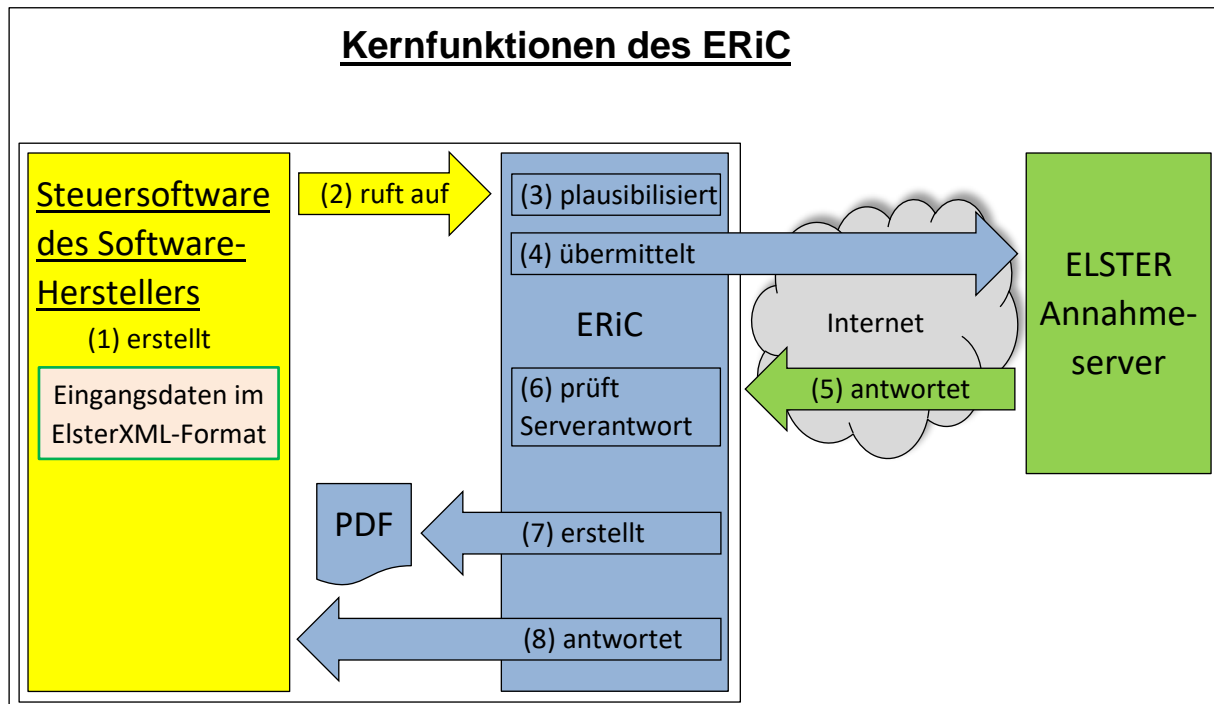
HINWEIS:

Achten Sie bei Problemen auf die Programmausgaben und prüfen Sie die Logdatei „eric.log“ auf Fehlermeldungen. Verifizieren Sie die korrekte Ausführung vorangegangener Schritte und beachten Sie die Hinweise in der Datei Liesmich.txt.

2.8 Das Beispielprogramm „ericdemo“ und die Kernfunktionen der ERiC API

Dieser Abschnitt vermittelt Ihnen am C++ Programmbeispiel „ericdemo“ die Kernfunktionen der ERiC API. Die in folgender Abbildung visualisierten Abläufe können Sie in der anschließenden Beschreibung und in den Codebeispielen nachvollziehen.

Abbildung 2-1 Die Kernfunktionen des ERiC



(1) Steuersoftware des Software-Herstellers erstellt ElsterXML-Datei

Die Steuersoftware stellt der ERiC API die Eingangsdaten im Elster-XML-Format⁵ bereit. Beim Beispielprogramm „ericdemo“ ist dies die Datei „ESt_2020.xml“.

- Gehen Sie hierzu in das Verzeichnis „Windows-x86_64\Beispiel\ericdemo-cpp“.
- Öffnen Sie die Datei „ESt_2020.xml“ mit einem Text- oder XML-Viewer.
 - Das Element `<DatenArt>` mit dem Wert „ESt“ wird vom übergeordneten Element `<TransferHeader>` ummantelt.
 - ERiC unterstützt Sie beim Testen durch die Bereitstellung von Test-Steuernummern und Test-Finanzämtern⁶. Das Tag `<Testmerker>`, im Beispiel ist es der Wert „700000004“, muss in den Testdaten gesetzt sein.

⁵ siehe EHB, Kap. „Ergänzende Downloadpakete“, Tabelle „Ergänzende Softwarepakete und Dateien – Schnittstellenbeschreibungen“, Bezeichnung des Aufklappbereichs „ElsterXML-Schnittstelle_V11_V*.zip“

⁶ siehe EHB, Kap. "Test Unterstützung bei der ERiC-Anbindung"

Auszug aus der Datei „ESt_2020.xml“:

```
<Elster>
  <TransferHeader version="11">
    ...
    <DatenArt>ESt</DatenArt>
    <Testmerker>700000004</Testmerker>
    ...
  </TransferHeader>
  <DatenTeil>
    ...
    ...
  </DatenTeil>
</Elster>
```

(2) Steuersoftware des Software-Herstellers ruft ERiC auf

Die XML-Eingangsdaten in der Datei ESt_2020.xml werden neben anderen Funktionsparametern im Beispielprogramm „ericdemo“ der ERiC API-Funktion *EricBearbeiteVorgang()* übergeben.

(3) und (4) Plausibilisieren und übermitteln an den ELSTER Annahmeserver

Alle Daten, die von der Steuersoftware an die ERiC-Schnittstelle übergeben werden, müssen eine Plausibilitätsprüfung durchlaufen.

Bei der Plausibilitätsprüfung werden die Daten auf formale Korrektheit und Vollständigkeit geprüft. Die formale Korrektheit bezieht sich auf die Prüfung des Formates, der maximalen Längen und der Gültigkeit des Feldes im Kontext der Steuerart und des Veranlagungszeitraums.

Nachdem ERiC die Funktionsparameter überprüft und die XML-Eingangsdaten erfolgreich plausibilisiert hat, erfolgt der verschlüsselte Datenversand an den ELSTER Annahmeserver.

1. Öffnen Sie die Datei *ericvorgang.cpp*.
2. Sehen Sie sich in der Methode *EricVorgang::ausfuehren()* folgenden API-Funktionsaufruf an:

```
const int rc = ericAdapter.EricBearbeiteVorgang(
    xmlDaten.c_str(), argParser.getDatenartVersion().c_str(),
    bearbeitungsFlags, &druckEinstellungen,
    (zertifikat && sende ? &(zertifikat->getVerschlüsselungsParameter()) :
        nullptr),
    argParser.getHatTransferHandle() ? &transferHandle : nullptr,
    ergebnisPuffer.handle(), serverantwortPuffer.handle() );
```

(5) ELSTER Annahmeserver antwortet

Der ELSTER Annahmeserver antwortet. Das Ergebnis der Plausibilitätsprüfung und die möglichen Fehlermeldungen der Serverantwort werden über den jeweiligen EricPuffer (*ergebnisPuffer* und *serverantwortPuffer*) von *EricBearbeiteVorgang()* an den Aufrufer über ein Handle zurückgegeben. Zur Erzeugung, Verwendung und Freigabe von Rückgabepuffern siehe *EricRueckgabepufferHandle* in der API-Referenz⁷.

Der Rückgabewert *ERIC_OK* signalisiert Erfolg, andere Werte einen Fehler. Die möglichen Fehlerwerte mit Fehlertexten⁸ können Sie in der API-Referenz und im Softwarepaket in der Datei *eric_fehlercodes.h*⁹ nachlesen.

(6) ERiC prüft Serverantwort

ERiC prüft anhand der Serverantwort, ob bei der serverseitigen Verarbeitung ein Fehler aufgetreten ist. Falls ja, wird die Bearbeitung im ERiC abgebrochen und der Funktionsaufruf mit einer entsprechenden Fehlermeldung beendet. Die Steuersoftware muss in diesem Fall die im Parameter *serverantwortXmlPuffer* zurückgelieferte Serverantwort auswerten.

(7) ERiC erstellt PDF-Datei

Nachdem die ERiC API mit *ERIC_OK* geantwortet hat, ist eine PDF-Datei von ERiC erstellt worden.

Sie sehen im nächsten Codebeispiel aus „ericdemo“, Datei *ericvorgang.cpp* in der Struktur *eric_druck_parameter_t*, wie die ERiC Druckeinstellungen und der PDF-Dateiname gesetzt sind.

```
static eric_druck_parameter_t holeDruckeinstellungen()
{
    eric_druck_parameter_t druckEinstellungen = {};
    druckEinstellungen.version = 2;
    druckEinstellungen.vorschau = 0;
    druckEinstellungen.ersteSeite = 0;
    druckEinstellungen.duplexDruck = 0;
    druckEinstellungen.pdfName = "ericprint.pdf";
    druckEinstellungen.fussText = nullptr;
    return druckEinstellungen;
}
```

Näheres über die zulässigen Datenfelder und Werte, siehe API-Referenz, *eric_types.h*
Struktur *eric_druck_parameter_t*

⁷ Sie lernen die API-Referenz im nächsten Kap. [2.9 Verwendung der ERiC API-Referenz](#) kennen.

⁸ Eine Fehlerauswertung mit Fehlerbehandlung in der Steuersoftware ist damit möglich.

⁹ Windows-x86_64\include\

(8) ERiC antwortet

Als Ergebnis des Funktionsaufrufs *EricBearbeiteVorgang()* wird die Serverantwort in den Parameter *serverantwortXmlPuffer* zurückgeliefert.

Im Fall von Fehlern bei der Plausibilisierung wird die Serverantwort in den Parameter *rueckgabeXmlPuffer* zurückgeliefert.

2.9 Verwendung der ERiC API-Referenz

Die ERiC API-Referenzdokumentation wird im HTML- und PDF-Format ausgeliefert.

1. Öffnen Sie die HTML-Hauptseite „Dokumentation/API-Referenz/HTML/index.html“ in einem Web-Browser Ihrer Wahl.

Über den oberen oder linken Navigationsbereich gelangen Sie zu den von ERiC verwendeten Headerdateien, Datenstrukturen und API-Funktionen.

2. Öffnen Sie beispielsweise über den linken Navigationsbereich die Datei „ericapi.h“, siehe [Abbildung 2-2](#).

Hier sehen Sie die Funktionssignatur und Referenzdokumentation zur API-Funktion *EricBearbeiteVorgang()*.

Abbildung 2-2 Beispiel einer Funktionssignatur in der ERiC API-Referenz



The screenshot shows the ERiC API-Referenz web interface. On the left, a navigation pane lists the contents of the 'ericapi.h' file, with 'EricBearbeiteVorgang' selected. The main area displays the function signature and its description.

```

ERICAPI_IMPORT int EricBearbeiteVorgang ( const char *          datenpuffer,
                                           const char *          datenartVersion,
                                           uint32_t             bearbeitungsFlags,
                                           const eric_druck_parameter_t * druckParameter,
                                           const eric_verschlüsselungs_parameter_t * cryptoParameter,
                                           EricTransferHandle *   transferHandle,
                                           EricRueckgabepufferHandle rueckgabeXmlPuffer,
                                           EricRueckgabepufferHandle serverantwortXmlPuffer
                                           )
    
```

Diese API-Funktion ist die zentrale Schnittstellenfunktion zur Kommunikation mit dem ELSTER-Annahmeserver.

3 Verarbeitung von Jahressteuern am Beispiel ESt

Am Beispiel der Einkommensteuer für den Veranlagungszeitraum (VZ) 2020 mit ausgewählten Anlagen führt das nachfolgende Kapitel Schritt für Schritt durch die Integration des ERiC in die Steuersoftware:

- Kap. [3.1 Erste Aufgabe: Die ELSTER-XML Eingangsdaten am Beispiel ESt erstellen](#)
- Kap. [3.2 Zweite Aufgabe: Verarbeitung der Steuerdaten](#) (Plausibilisieren, Versenden, PDF-Druck)
- Kap. [3.3 Dritte Aufgabe: Neue und geänderte Felder in die Steuersoftware integrieren](#)
- Kap. [3.4 Vierte Aufgabe: Den TransferHeader mit EricCreateTH\(\) erstellen](#)

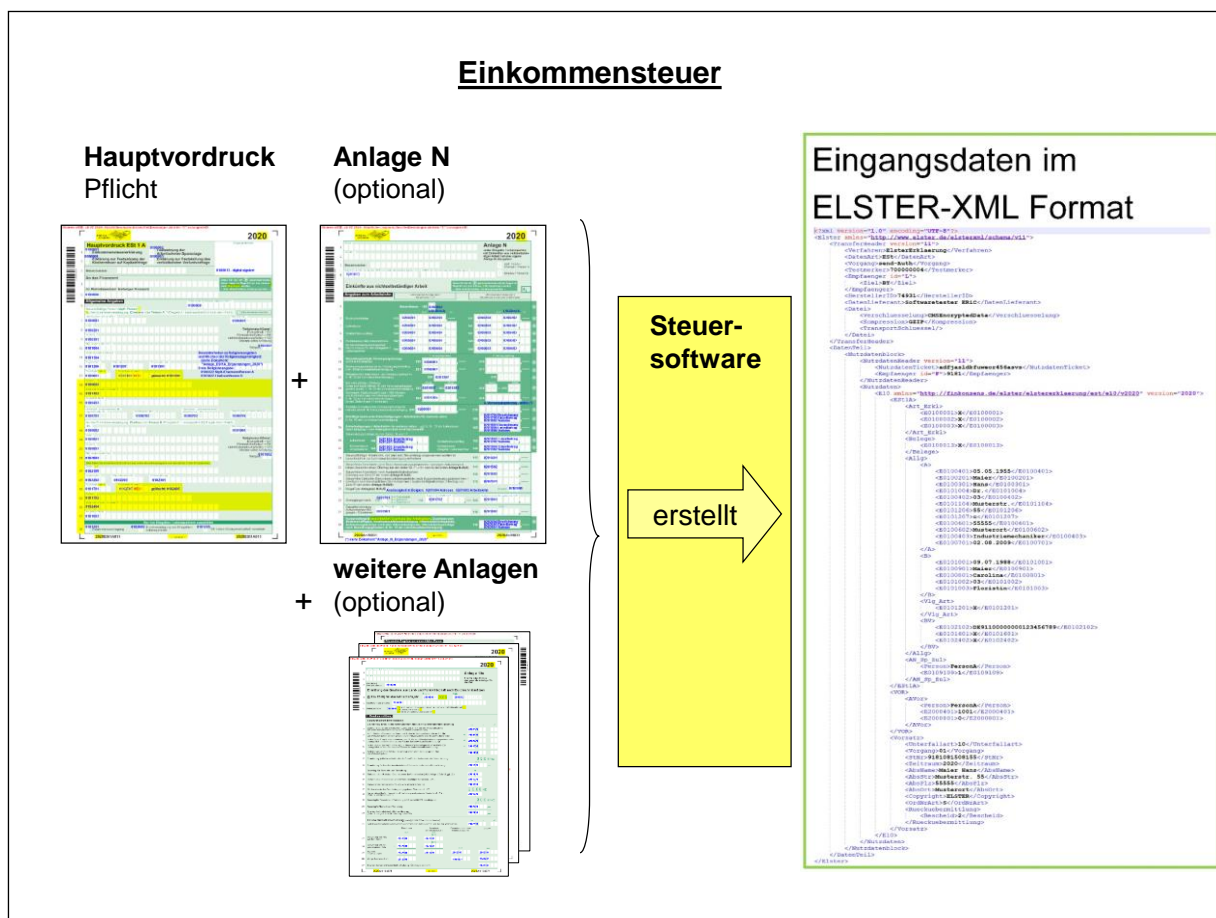
Anschließend erfahren Sie, wie Sie Ihre Steuersoftware an ein neues ERiC-Release anpassen können.

3.1 Erste Aufgabe: Die ELSTER-XML Eingangsdaten am Beispiel ESt erstellen

Steuersoftware, die mit ELSTER bzw. mit ERiC Datensätze an den ELSTER Annahmeserver übertragen möchte, muss aus den Eingaben des Benutzers einen gültigen, vollständigen ELSTER-XML Datensatz aufbauen und an ERiC übergeben. Dazu ist die Kenntnis der in einem VZ gültigen Felder für den Hauptvordruck mit optionalen Anlagen nötig.

[Abbildung 3-1](#) stellt schematisch die Aufgaben der Steuersoftware, ein ESt ELSTER-XML zu erstellen, dar.

Abbildung 3-1 Erstellung der ESt XML-Eingangsdaten



In den folgenden Abschnitten wird erklärt,

- wie die Eingaben des Steuerpflichtigen einem ELSTER-Feld zugeordnet werden, siehe Kap. [3.1.1 Von den Daten des Steuerpflichtigen zu den ELSTER Feldern](#)
- wie die ESt eXML-Daten für den VZ 2020 strukturiert sind, siehe Kap. [3.1.2 Die Struktur der ELSTER eXML-Daten für die ESt](#)
- wie Feldkennungen, -formate und -texte aufgebaut sind, siehe Kap. [3.1.3 Bedeutung der Feldkennungen, -formate und -texte](#)
- wie und wo Sie die benötigte Information zu gültigen ELSTER-Feldern und den zugehörigen Plausibilitätsprüfungen finden, siehe Kap. [3.1.4 Die ESt eXML Daten erstellen](#)

3.1.1 Von den Daten des Steuerpflichtigen zu den ELSTER Feldern

Ihr Steuerprogramm, in das Sie ERiC integrieren wollen, verwendet elektronische Formulare, in die der Steuerpflichtige seine Steuerdaten einträgt. Wie genau Ihre elektronischen Formulare aussehen, ist unerheblich. Aber die Formulare werden mindestens die Pflichtfelder der amtlichen Vordrucke für die ESt beinhalten.

In diesem Tutorial werden die (Pflicht-)Felder aus dem ESt Hauptvordruck und einige weitere Anlagen verwendet, um die ELSTER-XML Eingangsdaten aufzubauen.

In diesem Beispiel wird u.a. der Hauptvordruck, Dateiname „2020ESt1A011.png“, verwendet. Öffnen Sie den Hauptvordruck jetzt. Die Vordrucke finden Sie im Verzeichnis 2020\Jahressteuern\ESt\Grafiken_und_Erweiterungen_E10\

Abbildung 3-2 Ausschnitt aus 2020ESt1A011.png

Hinweise eXML ab VZ 2020: Sämtlichen numerischen Feldkennungen wird ein "E" vorangestellt; Markierungen "-----" von nicht umgesetzten Feldern entfallen.

A Anleitung vorhanden

2020

Hauptvordruck ESt 1 A

1 ☒ 0100001 Einkommensteuererklärung ☒ 0100002 Festsetzung der Arbeitnehmer-Sparzulage

2 ☒ 0100009 Erklärung zur Festsetzung der Kirchensteuer auf Kapitalerträge ☒ 0100003 Erklärung zur Feststellung des verbleibenden Verlustvortrags

3 Steuernummer 0100013 - digital signiert

An das Finanzamt

4 Bei Wohnsitzwechsel: bisheriges Finanzamt

5 0100006

Allgemeine Angaben

6 Steuerpflichtige Person (stpf. Person) 0100008

Nur bei Zusammenveranlagung: Ehemann oder Person A *) (Ehegatte A / Lebenspartner[in] A nach dem LPartG) *) Bitte Anleitung beachten.

Identifikationsnummer (IdNr.) 0100081

8 0100201 Name

9 0100301 Vorname

10 0101004 Titel, akademischer Grad

11 0101104 Straße (derzeitige Adresse)

12 0101206 Hausnummer Hausnummerzusatz 0101207 Adressergänzung 0101301

13 0100601 Postleitzahl (Inland) 0101405 NEU! Postleitzahl (Ausland) gelöscht: 0101404

14 0100602 Wohnort

15 0101403 Staat (falls Anschrift im Ausland)

B

C

D

E

F Besonderheiten zu Religionsangaben und Wechsel der Religionszugehörigkeit: (siehe Dokument: "Anlage_ESt1A_Ergaenzungen_2020") Freie Religionsangabe: 0100422 Stpf./Ehemann/Person A 0101022 Ehefrau/Person B

Religionsschlüssel: Evangelisch = EV Römisch-Katholisch = RK nicht kirchensteuerpflichtig = VD Weitere siehe Anleitung Religion 0100402

Geburtsdatum 0100401 M J J J J

- [A]** Beachten Sie den roten Hinweis in der Kopfzeile des Vordrucks. Die ESt für den Veranlagungszeitraum (VZ) 2020 verwendet die neue Nutzdatenstruktur (kurz eXML). Lesen Sie hierzu die ausführlichen Informationen zu eXML im EHB, Kap. „Einführung in die neue Nutzdatenstruktur“.
- [B]** Die amtlichen Vordrucke sind um Informationen für den Softwareentwickler erweitert (annotiert). Die auszufüllenden Felder sind auf den Vordrucken mit blauen Nummern, den Feldkennungen versehen.
- [C]** Änderungen zum Vorjahr sind gelb markiert.

[D] Falls Felder erstmalig in diesem Vordruck vorhanden sind, erfolgt eine rote Markierung mit dem Wort „**NEU!**“.

Beispielsweise ist die Feldkennung „0101405“ erstmalig verwendet und deshalb mit dem roten Wort „**NEU!**“ markiert.

[E] Gelöschte Feldkennungen sind unter „gelöscht:“ aufgezählt.

[F] Hinweise auf ergänzende Dokumente werden bei Bedarf aufgedruckt, hier z.B. „Besonderheiten zu Religionsangaben“.

Die eindeutigen Feldkennungen sind zusammen mit den zugehörigen, vom Steuerpflichtigen eingegebenen Daten (Werte) des Steuerfalls in die eXML Eingangsdaten zu übernehmen.

Über die Feldkennungen werden die vom Steuerpflichtigen erklärten Werte von den verarbeitenden Systemen der Finanzbehörde inhaltlich zugeordnet.

Wie die Ihnen nun bekannten Feldkennungen bei der Erstellung der eXML Eingangsdaten verwendet werden, erfahren Sie in den nächsten Abschnitten.

3.1.2 Die Struktur der ELSTER eXML-Daten für die ESt

Dieser und die folgenden Abschnitte setzen das installierte und lauffähige Beispielpogramm „ericdemo“ voraus. Bitte lesen Sie ggf. das Kap. [2 Vorbereitende Maßnahmen](#).

Die zu erstellende eXML-Datei mit den Steuerdaten beinhalten sowohl die Daten des Hauptvordrucks als auch die Daten der optionalen Anlagen. Sie sehen dies im Beispieldatensatz „Windows-x86_64\Beispiel\ericdemo-cpp\ESt_2020.xml“.

Die zugrundeliegende, annotierte eXML-Schemabeschreibung ist ausgehend von Dokumentation\Schnittstellenbeschreibungen\Erklärungssteuern\ESt_10_2020\Schema\elster11_E10_extern.xsd in E10-2020.xsd zu finden.

Der grundsätzliche, strukturelle Aufbau eines ELSTER eXML Eingangsdatensatzes ist in der folgenden Abbildung dargestellt. Zur besseren Lesbarkeit sind nur wesentliche Elemente enthalten.



Alle Erklärungssteuern folgen dieser Struktur.

Die Elemente `<TransferHeader>` und `<DatenTeil>` mit dem `<Nutzdatenblock>` und dem `<NutzdatenHeader>` sind für alle Datenarten gleich, nur das darunterliegende Element `<Nutzdaten>` mit seinem Inhalt ist spezifisch für Erklärungssteuern. Es wird empfohlen, den `<TransferHeader>` mit der API-Funktion `EricCreateTH()` zu erstellen, siehe Kap. [3.4](#).

3.1.2.1 Die Nutzdatenstruktur für eXML-Daten

Um sich mit der eXML Nutzdatenstruktur vertraut zu machen, öffnen Sie die Beispieldatei von ericdemo, ESt_2020.xml¹⁰. Sie sehen, dass jeder Feldkennung das Präfix „E“ vorangestellt ist. Das Präfix „E“ ist ein typisches Merkmal für eXML-Daten.

```
...
<Nutzdaten>
  <E10 xmlns=http://finkonsens.de/elster/elstererklaerung/est/e10/v2020 version="2020">
    <ESt1A>
      <Art_Erkl>
        <E0100001>X</E0100001>
        <E0100002>X</E0100002>
        <E0100003>X</E0100003>
      </Art_Erkl>
      <Belege>
        <E0100013>X</E0100013>
      </Belege>
      <Allg>
        <A>
          <E0100401>05.05.1955</E0100401>
          <E0100201>Maier</E0100201>
          <E0100301>Hans</E0100301>
        </A>
      </Allg>
    </E10>
  </Nutzdaten>
...
```

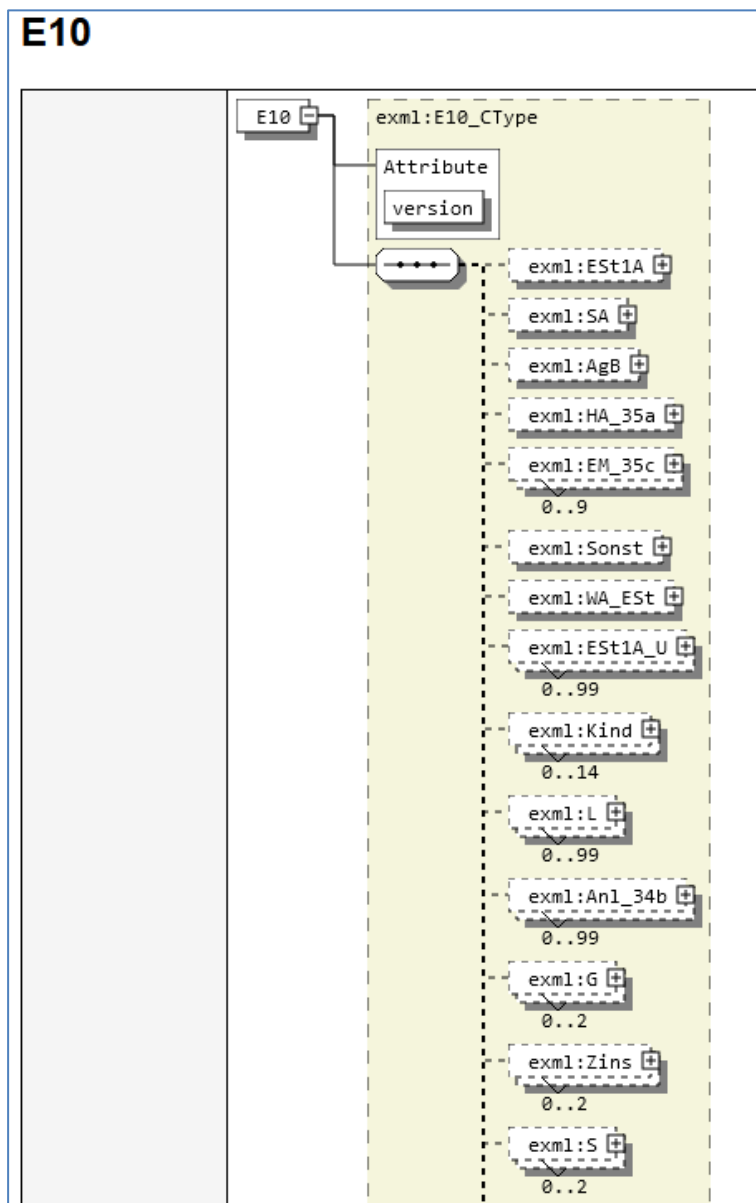
Das Schema und die Schemadokumentation sind für die Integration in die Steuersoftware unverzichtbare Hilfsmittel. Die Schemadokumentation wird im HTML-Format bereitgestellt und kann mit einem Internetbrowser, z.B. Firefox, verwendet werden.

¹⁰ Windows-x86_64\Beispiel\ericdemo-cpp\

Öffnen Sie für dieses Beispiel jetzt

„Dokumentation\Schnittstellenbeschreibungen\Erklaerungssteuern\Est_10_2020\SchemaDokumentation\E10-2020.html“ und klicken im linken oberen Bereich auf das Element „E10“, um sich mit der eXML Struktur vertraut zu machen:

Abbildung 3-3 Schema für ESt 2020



Damit sollte der Inhalt der Datei Est_2020.xml verständlich sein.

Beachten Sie, dass die Reihenfolge der Elemente strikt dem Schema folgt!

Aus dem annotierten, amtlichen Vordruck 2020ESt1A011.png entnehmen Sie, dass die Feldkennung „E0100401“ das Geburtsdatum der steuerpflichtigen Person beinhaltet. Vom Steuerpflichtigen wurde der Wert „05.05.1955“ angegeben. Ist das hier verwendete Format „dd.mm.yyyy“ richtig? Das eXML-Schema sagt darüber nichts aus.

Die Antwort auf diese Frage liefert das Feldformat. Es wird im folgenden Abschnitt in der Dokumentation der Plausibilitätsprüfungen vorgestellt. Sie beinhaltet Informationen über formale Prüfungen (Felder, Formatdefinitionen) und inhaltliche Prüfungen (Regeln, Werteberechnungen), die Sie benötigen.

3.1.3 Bedeutung der Feldkennungen, -formate und -texte

Öffnen Sie das Dokument „Zusatzinformationen_zur_Plausibilitaetspruefung.pdf“¹¹, denn es erklärt das Grundkonzept der Felder, Regeln und Kennzahlen und erläutert die Sprachkonstrukte für die Regelbedingungen, die Feldtypen und formale Prüfungen. Mit diesem Wissen ist das Verständnis für die Jahresdokumentation gegeben. Öffnen Sie nun die ESt Jahresdokumentation für den VZ 2020 „Jahresdokumentation_10_2020.xml“¹² mit Excel oder einem kompatiblen Programm.

Das Dokument enthält für den Hauptvordruck und jede Anlage einen Reiter mit allen im VZ 2020 für die ESt gültigen Kontexten, Feldkennungen, -formaten, -texten und Regeln. Suchen Sie auf dem Reiter „ESt1A – Felder“ in der Spalte „Name“ nach der Feldkennung „E0100401“, dem Geburtsdatum aus dem letzten Abschnitt.

Abbildung 3-4 Ausschnitt aus der ESt Jahresdokumentation 2020

	A	B	C	D	E	F
1	Kontext	Name	Beschreibung	max. Zeilen	Format	Format als regulärer Ausdruck
12	Allg/A	E0100081	Identifikationsnummer	1	IDNr	[0-9]{11}
13	Allg/A	E0100401	Geburtsdatum	1	Datum TT.MM.JJJJ mit Zusatzprüfung	\\d\\d\\.\\d\\d\\.\\d\\d\\d\\d
14	Allg/A	E0100201	Name	1	String	.{1,25}
15	Allg/A	E0100301	Vorname	1	String	.{1,25}
16	Allg/A	E0101004	Titel, akademischer Grad	1	String	.{1,25}
	Allg/A	E0100402	Religion	1	Benutzerdefinierte Typdefinition: Religionsschluessel_ab_VZ_2014 - 3	\\Q11\\E \\Q03\\E \\Q02\\E \\Q05\\E \\Q10\\E \\Q16\\E \\Q13\\E \\Q14\\E \\Q15\\E \\Q17\\E \\Q18\\E \\Q22\\E \\Q23\\E
	Est1A - Kontexte	Est1A - Felder	Est1A - Erste Seite Felder	Est1A - Regeln	Est1A - Kennzahlen	Est1A - Texte
					SA - Kontexte	SA - Felder

Sie erfahren in der Spalte:

- Beschreibung: „Geburtsdatum“
- Max. Zeilen: „1“, diese Feldkennung darf maximal 1x vorkommen.
- Format: „Datum TT.MM.JJJJ mit Zusatzprüfung“, das Format für das Datum „05.05.1955“ ist richtig umgesetzt.
- Format als regulärer Ausdruck: `\d\d.\d\d.\d\d\d\d`, die formale Definition des Formats
- Max. Länge: „10“, die Länge des Feldes darf nicht größer als 10 sein.
- Pflichtfeld: „Ja“, der Wert muss angegeben werden
- Vordruckzeile: „7“, sehen Sie nach, ob in 2020Est1A011.png das Feld „E0100401“ in Zeile 7 steht.

¹¹ Dokumentation\Plausipruefungen\

¹² Dokumentation\Plausipruefungen\Erklaerungssteuern\ESt\UFA10\

3.1.4 Die ESt eXML Daten erstellen

Das Wissen der vorangegangenen Abschnitte ist notwendig, damit Sie in diesem Kapitel ausgehend von den ESt Vordrucken, die ESt eXML Daten erstellen können.

Verwenden Sie zum Erstellen der XML-Daten einen Texteditor Ihrer Wahl.

Die Einzelschritte sind:

- Kap. [3.1.4.1 Vom Vordruck zu den ESt eXML Daten](#)
- Kap. [3.1.4.2 Felder, die mehrfach angegeben werden können](#)

3.1.4.1 Vom Vordruck zu den ESt eXML Daten

Am Beispiel der Feldkennungen und -werte aus dem ESt 2020 Hauptvordruck und den eXML-Daten¹³ aus ericdemo, wird exemplarisch gezeigt, wie Sie Schritt für Schritt einen ESt eXML Datensatz erstellen können.

Öffnen Sie die Seite 1 des annotierten amtlichen Hauptvordrucks 2020ESt1A011.png¹⁴. Entnehmen Sie die benötigten Feldkennungen (rot und grau markiert):

Hinweise eXML ab VZ 2020: Sämtlichen numerischen Feldkennungen wird ein "E" vorangestellt; Markierungen "-----" von nicht umgesetzten Feldern entfallen.

Anleitung vorhanden

2020

Hauptvordruck ESt 1 A

0100001 Einkommensteuererklärung

0100002 Festsetzung der Arbeitnehmer-Sparzulage

0100009 Erklärung zur Festsetzung der Kirchensteuer auf Kapitalerträge

0100003 Erklärung zur Feststellung des verbleibenden Verlustvortrags

2

3 Steuernummer

0100013 - digital signiert

An das Finanzamt

Bei Wohnsitzwechsel: bisheriges Finanzamt

4

5 0100006

Allgemeine Angaben

6 Steuerpflichtige Person (stpfl. Person)

0100008

Nur bei Zusammenveranlagung: Ehemann oder Person A *) (Ehegatte A / Lebenspartner[in] A nach dem LPartG)

Identifikationsnummer (IdNr.)

0100081

Name

0100201

Vorname

0100301

Titel, akademischer Grad

0101004

Straße (derzeitige Adresse)

0101104

Hausnummer

0101206

Hausnummerzusatz

0101207

Adressergänzung

0101301

Postleitzahl (Inland)

0100601

Postleitzahl (Ausland)

0101405 NEU!

gelöscht: 0101404

Wohnort

0100602

Staat (falls Anschrift im Ausland)

0101403

Religionsschlüssel:

Evangelisch = EV

Römisch-Katholisch = RK

nicht kirchensteuerpflichtig = VD

Weitere siehe Anleitung

Religion

0100402

Besonderheiten zu Religionsangaben und Wechsel der Religionszugehörigkeit: (siehe Dokument: "Anlage_ESt1A_Ergänzungen_2020")

Freie Religionsangabe:

0100422 Stpfl./Ehemann/Person A

0101022 Ehefrau/Person B

Welche Felder im Vorsatz des Nutzdatenblocks und welche im Nutzdatenblock einzutragen sind, entnehmen Sie dem eXML Schema¹⁵ und der Schemadokumentation¹⁶.

Sie stellen fest, dass die in der obigen Abbildung grau eingekreisten Felder im Vorsatz und die rot eingekreisten Felder im Nutzdatenblock einzutragen sind.

¹³ eXML-Datei „ESt_2020.xml“

¹⁴ Vordruck: 2020\Jahressteuern\ESt\Grafiken_und_Erweiterungen_E10

¹⁵ Dokumentation\Schnittstellenbeschreibungen\Erklärungssteuern\ESt_10_2020\Schema

¹⁶ Dokumentation\Schnittstellenbeschreibungen\Erklärungssteuern\ESt_10_2020\SchemaDokumentation

Das Feld „Steuernummer“ kann nicht 1:1 in den eXML-Datensatz übernommen werden, es muss zuerst in das 13-stellige ELSTER-Steuernummerformat konvertiert werden. Hierzu gehen Sie wie folgt vor:

- Öffnen Sie das Dokument *Pruefung_der_Steuer_und_Steueridentifikatsnummer.pdf*¹⁷ und machen sich mit dem Inhalt vertraut, insbesondere mit Kap. 3 „Aufbau der Steuernummer im ELSTER-Steuernummerformat“.
- Entnehmen Sie dem Dokument *Finanzamtsdaten.xlsx*¹⁸, je nach Finanzamtsnummer, welche BuFa (Bundesfinanzamtsnummer) der Steuernummer zuzuordnen ist.
- Entnehmen Sie dem PDF-Dokument die Regel der 5ten Stelle der Steuernummer.
- Bilden Sie nun die ELSTER-Steuernummer und tragen Sie diese jetzt in die eXML Daten ein.

Programmatisch kann mit der API-Funktion *EricMakeElsterStnr()* eine Steuernummer im Format des Steuerbescheides in eine 13-stellige Steuernummer des ELSTER-Steuernummerformats umgewandelt werden.

Den Aufbau der 13-stelligen ELSTER-Steuernummer finden Sie auch im XML-Schema und der Schemadokumentation.

¹⁷ befindet sich im Downloadbereich für Entwickler:

<https://www.elster.de/elsterweb/entwickler/infoseite/schnittstellenbeschreibungen>

¹⁸ Dokumentation\Finanzamtsdaten.xlsx

3.1.4.2 Felder, die mehrfach angegeben werden können

Bisher konnte ein Feld des Hauptvordrucks einem Wert eindeutig zugeordnet werden. Es gibt Felder, die mehrmals verwendet werden. Die Felder von „Spenden und Mitgliedsbeiträge“ sind solche Felder. Eine eindeutige Feldzuordnung wird mit dem sog. „Kontext“ hergestellt, die „maximale Wiederholbarkeit“ gibt an, wie oft das Feld verwendet werden kann. Lesen Sie jetzt die weiterführenden Informationen zum „Kontext“ in Zusatzinformationen_zur_Plausibilitaetspruefung.pdf¹⁹ und im EHB, Kap. „Einführung in die neue Nutzdatenstruktur“.

Wie Sie ein mehrmals verwendetes Feld in die eXML Daten aufnehmen, lernen Sie anhand der folgenden Beispieldaten:

Tabelle 3-1 Beispieldaten für 2x „Spenden und Mitgliedsbeiträge“

Zeile	Feldkennung	Feldwert
6	E0104704	Musikverein
6	E0104705	200
6	E0104708	200
6	E0104704	Schützenverein
6	E0104708	100
6	E0104702	200
6	E0104703	300

¹⁹ Dokumentation\Plausipruefungen\

Die Felder „Spenden und Mitgliedsbeiträge“ befinden sich auf dem annotierten amtlichen Vordruck 2020AnlSonderausgaben401.png. Öffnen Sie nun diesen Vordruck:

Zuwendungen (Spenden und Mitgliedsbeiträge)									
Spenden und Mitgliedsbeiträge (ohne Beträge in den Zeilen 9 bis 12)				lt. Bestätigungen EUR		lt. Betriebsfinanzamt EUR			
5	zur Förderung steuerbegünstigter Zwecke an Empfänger im Inland	123	0108105			124	0108106		
6	zur Förderung steuerbegünstigter Zwecke an Empfänger im EU- / EWR-Ausland	133	0104702			134	0104703		
7	an politische Parteien (§§ 34g, 10b EStG)	127	0108701			128	0108702		
8	an unabhängige Wählervereinigungen (§ 34g EStG)	129	0108801			130	0108802		
siehe (*)									
Spenden in das zu erhaltende Vermögen (Vermögensstock) einer Stiftung									
9	2020 geleistete Spenden an Empfänger im Inland (lt. Bestätigungen / lt. Betriebsfinanzamt)	208	0108405			209			
10	2020 geleistete Spenden (lt. Bestätigungen / lt. Betriebsfinanzamt) an Empfänger im EU- / EWR-Ausland	224	0105502			225			
11	Von den Spenden in den Zeilen 9 und 10 sollen 2020 berücksichtigt werden	212	0108509			213			
12	2020 zu berücksichtigende Spenden aus Vorjahren in das zu erhaltende Vermögen (Vermögensstock) einer Stiftung, die bisher noch nicht berücksichtigt wurden	214	0108607			215			
0105902 - Summe der Umsätze, Löhne und Gehälter zur Ermittlung des Spendenhöchstbetrages (Kz 108)									

Die gewünschten Felder, z.B. „E0104708“ können im Abschnitt „Spenden und Mitgliedsbeiträge“ nicht gefunden werden, jedoch befindet sich am rechten Rand des Vordrucks für die Zeilen 5 – 8 eine Referenz auf die Fußnote „siehe (*)“.

Öffnen Sie nun das in der Fußnote referenzierte Dokument

„Anlage_Sonderausgaben_Ergaenzungen_2020.pdf“²⁰:

Ergänzungen Anlage Sonderausgaben 2020

Hinweis eXML ab VZ 2020: Sämtlichen numerischen Feldkennungen wird ein "E" vorangestellt.

Zeile	Spenden und Mitgliedsbeiträge (ohne Spenden in das zu erhaltende Vermögen einer Stiftung)	Beschreibung	Betrag lt. Bestätigungen	Betrag lt. Nachweis Betriebsfinanzamt	Summe der Beträge lt. Bestätigungen	Summe der Beträge lt. Nachweis Betriebsfinanzamt
5	zur Förderung steuerbegünstigter Zwecke an Empfänger im Inland	0108102	0108103	0108104	0108105 Kz 52.123	0108106 Kz 52.124
6	zur Förderung steuerbegünstigter Zwecke an Empfänger im EU- / EWR-Ausland	0104704	0104705	0104708	0104702 Kz 52.133	0104703 Kz 52.134
7	an politische Parteien (§§ 34g, 10b EStG)	0108716	0108717	0108718	0108701 Kz 52.127	0108702 Kz 52.128
8	an unabhängige Wählervereinigungen	0108804	0108805	0108806	0108801 Kz 52.129	0108802 Kz 52.130

Sie finden das gesuchte Feld E0104708 in Zeile 6, Spalte „Betrag lt. Nachweis Betriebsfinanzamt“ und stellen fest, dass sich dort keine weitere Information zu dem Feld befindet.

²⁰ Vordrucke: 2020\Jahressteuern\Est\Grafiken_und_Erweiterungen_E10\

In diesem Beispiel sollen diese Feldwerte zwei Mal angegeben werden, aber wie oft dürfen die Felder maximal verwendet werden? Die Antwort liefert der Wert „max. Wiederholbarkeit“ für den Kontext des Feldes aus der Jahresdokumentation²¹. Öffnen Sie die Jahresdokumentation und suchen Sie nach dem Feld E0104708. Sie finden es auf dem Tabellenblatt „SA - Felder“ und können dort den zugehörigen „Kontext“ „Zuw/Sp_MB/Foerd_st_beg_Zw_EU_EWR/Einz“ entnehmen. Mit diesem Kontext können Sie auf dem Tabellenblatt „SA - Kontexte“ den zugehörigen Wert „max. Wiederholbarkeit“ = 99 entnehmen. Das Feld E0104708 kann somit 99-mal angegeben werden.

Damit die Felder und Feldwerte an die richtige Position in das eXML eingefügt werden können, öffnen Sie das Schema und entnehmen die richtige Reihenfolge, siehe [Abbildung 3-3 Schema für ESt 2020](#).

Beachten Sie die Formate und Regeln aus der Jahresdokumentation für die Feldwerte.

Damit sind alle benötigten Informationen vorhanden, erweitern Sie nun die eXML-Daten und testen Sie diese z.B. mit ericdemo. Vergleichen Sie bei Bedarf die erstellten eXML-Daten mit der Beispiellösung ESt_2020-Beispiel_Loesung.xml.

Falls beim Erstellen der eXML-Daten die Reihenfolge nicht eingehalten wird, erhalten Sie folgende Fehlermeldung (eric.log):

```
element 'SA' is not allowed for content model
```

'SA' bezieht sich auf die hier verwendeten Beispieldaten in der Anlage Sonderausgaben.

Im folgenden Kap. [3.2 Zweite Aufgabe: Verarbeitung der Steuerdaten](#) lesen Sie, wie die eXML Daten plausibilisiert und an den ELSTER Annahmeserver übermittelt werden. Das sog. Freizeichnungsdokument wird im PDF-Format erstellt.

²¹

Dokumentation\Plausipruefungen\Erklaerungssteuern\ESt\UFA10\Jahresdokumentation_10_2020.xml

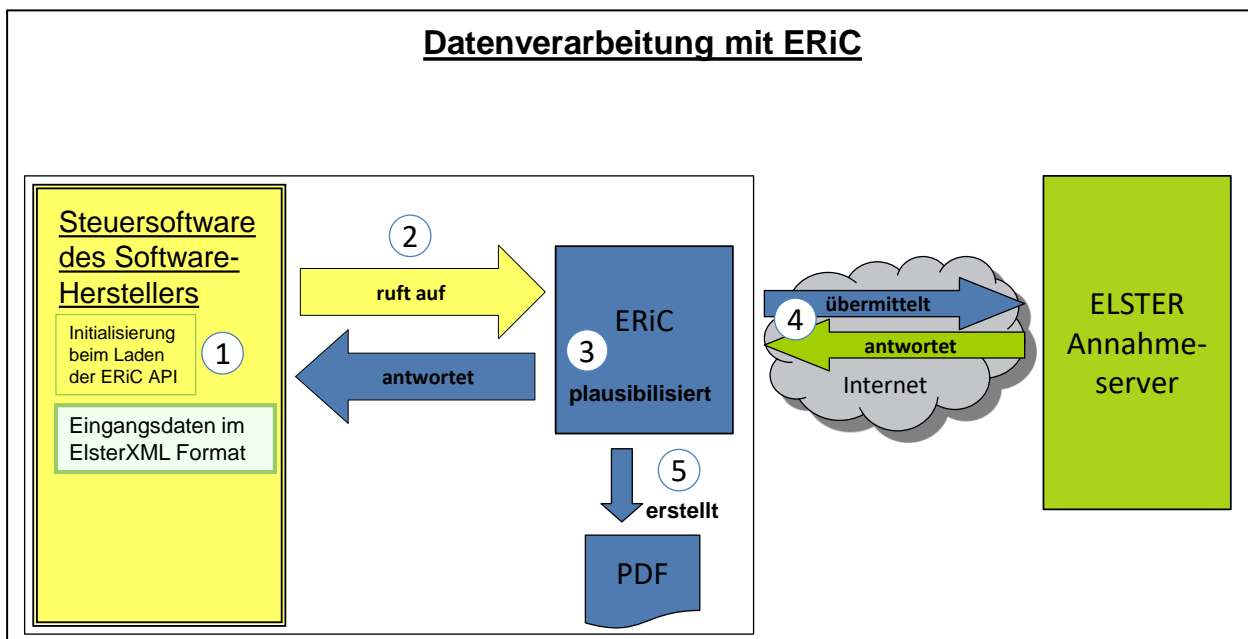
3.2 Zweite Aufgabe: Verarbeitung der Steuerdaten

Das von Ihnen in den vorangegangenen Abschnitten erweiterte ESt eXML werden Sie in diesem Kapitel mit ERiC verarbeiten. Sie erfahren:

- Wie Sie in Ihrer Software das empfohlene Vorgehen bei der Initialisierung beim Laden der ERiC Komponenten umsetzen können im
Kap. [3.2.1 Schritt 1 - Implementieren Sie das empfohlene Vorgehen der Initialisierung beim Laden der ERiC Komponenten](#)
- Wie Sie die ERiC API aufrufen im Kap. [3.2.2 Schritt 2 - Aufruf der ERiC API](#)
- Wie Sie die eXML Steuerdaten mit ERiC plausibilisieren im
Kap. [3.2.3 Schritt 3 - Plausibilitätsprüfungen](#)
- Wie Sie die eXML Steuerdaten an den ELSTER Annahmeserver übermitteln im
Kap. [3.2.4 Schritt 4 - Übermittlung an den ELSTER Annahmeserver](#)
- Wie das sog. Freizeichnungsdokument als PDF-Datei erstellt wird im
Kap. [3.2.5 Schritt 5 - PDF-Erstellung](#)

In der folgenden Abbildung, die sich an der [Abbildung 2-1](#) der ERiC API-Kernfunktionen orientiert, ist der schematische Ablauf der Verarbeitung der Steuerdaten mit ERiC dargestellt. Für sehr viele Anwendungsfälle ist dieser Ablauf gleich. Falls in einem gewählten Anwendungsfall die PDF-Datei nicht erstellt werden soll, entfällt (5).

Abbildung 3-5 Datenverarbeitung mit ERiC



3.2.1 Schritt 1 - Implementieren Sie das empfohlene Vorgehen der Initialisierung beim Laden der ERiC Komponenten

Mit Ihrer zu erstellenden Steuersoftware sind die ERiC Basisbibliotheken und Plugins²², siehe EHB, Kap. „Empfohlenes Vorgehen bei der Initialisierung“, mit auszuliefern.

Die folgenden Punkte zeigen die zur Initialisierung notwendigen Schritte des ERiC am Beispiel der Implementierung in „ericdemo“. Öffnen Sie hierzu eric.cpp²³.

1. Der Suchpfad für die ERiC Bibliotheken wird gesetzt.

```
System::setzeBibliotheksSuchpfad(homeDir.c_str())
```

In der Methode *setzeBibliotheksSuchpfad* wird die folgende Systemfunktion²⁴ aufgerufen:

```
SetDllDirectoryA(verzeichnis)
```

Diese Funktion muss in Ihrer Applikation entsprechend aufgerufen werden.

2. die Bibliothek „ericapi“ wird mit der Funktion *ladeEricApi(homeDir)* dynamisch geladen:

```
...
libEricApi =
Resolve::library(System::dateiPfad(apiVerzeichnis,ericapiDateiname).c_str());
...
EricBearbeiteVorgangPtr =
    ladeFunktion<EricBearbeiteVorgangFun>("EricBearbeiteVorgang", libEricApi);

EricGetHandleToCertificatePtr =
    ladeFunktion<EricGetHandleToCertificateFun>("EricGetHandleToCertificate",
        libEricApi);

EricCloseHandleToCertificatePtr =
    ladeFunktion<EricCloseHandleToCertificateFun>("EricCloseHandleToCertificate",
        libEricApi);
...
```

3. Die ERiC API-Funktion

```
EricInitialisiere(homeDir.c_str(), logDir.c_str())
```

sucht im Verzeichnis „homeDir“ rekursiv nach den ERiC Plugins und „logDir“ setzt das Verzeichnis, in das die Logdatei eric.log geschrieben wird.

Zum Beenden der Anwendung ist, um alle ERiC Plugins geordnet zu entladen, *EricBeende()* aufzurufen.

²² Windows-x86_64\dll* und Windows-x86_64\dll\plugins2*

²³ Windows-x86_64\Beispiel\ericdemo-cpp\ericdemo\eric.cpp

²⁴ Hier am Beispiel Windows.

3.2.2 Schritt 2 - Aufruf der ERiC API

Sie haben im Kap. [3.1](#) die ESt eXML-Daten²⁵ für den VZ 2020 erweitert und im Kap. [3.2.1](#) das empfohlene Vorgehen bei der Initialisierung implementiert. Damit sind alle Vorbereitungen zum Aufruf der ERiC API-Funktionen abgeschlossen.

3.2.2.1 Bedeutung und Aufruf der ERiC API-Funktion *EricBearbeiteVorgang()*

Die ERiC API-Funktion *EricBearbeiteVorgang()* nimmt Steuerdaten im ELSTER-XML Format entgegen, plausibilisiert und versendet sie, gibt mit dem Rückgabewert *rc* und dem *ergebnisPuffer* sowie dem *serverantwortPuffer* Auskunft über den Status der Einlieferung am ELSTER Annahmeserver und erstellt je nach Anwendungsfall eine PDF-Datei.

EricBearbeiteVorgang() wird für alle von ERiC unterstützten Datenarten verwendet. Deshalb lernen Sie diese ERiC API-Funktion jetzt näher kennen.

²⁵ Alternativ kann die Lösungsdatei „ESt_2020-Beispiel_Loesung.xml“ verwendet werden.

Schauen wir uns an, mit welchen Parametern *EricBearbeiteVorgang()* aufgerufen wird. Öffnen Sie hierfür die Datei *ericvorgang.cpp*²⁶, Methode *EricVorgang::ausfuehren()*, des Beispielprogramms „ericdemo“.

```
const int rc = ericAdapter.EricBearbeiteVorgang(
    xmlDaten.c_str(), argParser.getDatenartVersion().c_str(),
    bearbeitungsFlags, &druckEinstellungen, verschluesselungsParameter,
    argParser.getHatTransferHandle() ? &transferHandle : nullptr,
    ergebnisPuffer.handle(), serverantwortPuffer.handle() );
```

Die Funktionsparameter für das Beispiel sind:

- Der Eingangsparameter *xmlDaten* enthält die aus Kap. 3.1 erstellten ESt Steuerdaten im ELSTER XML-Format.
- Aus der Datenartversionmatrix²⁷ suchen Sie mit den Vorgaben:
Datenart = ‚EST‘ und VZ = ‚2020‘ die *datenartVersion* = ‚EST_2020‘ heraus. Im Beispiel mit „ericdemo“ übergeben Sie die *datenartVersion* als Kommandozeilenparameter. Die *datenartVersion* muss zu den XML-Eingangsdaten passen.
- Mit den Bearbeitungsflags *bearbeitungsFlags* spezifizieren Sie, wie die *xmlDaten* verarbeitet werden sollen. Für eine Plausibilisierung weisen Sie den Wert *ERIC_VALIDIERE*²⁸ zu. Mehrere Werte werden durch oder-Verknüpfungen zugewiesen.
- *&druckEinstellungen* ist ein Zeiger auf die Struktur *eric_druck_parameter_t*. und definiert die ERiC Druckfunktionalität, z.B. hier ohne Vorschau druck:

```
druckEinstellungen.version = 2;
druckEinstellungen.vorschau = 0;
druckEinstellungen.ersteSeite = 0;
druckEinstellungen.duplexDruck = 0;
druckEinstellungen.pdfName = "ericprint.pdf";
druckEinstellungen.fussText = nullptr;
```

Weitere Informationen zum PDF-Druck erfahren Sie im EHB, Kap. „Anwendungsfälle von *EricBearbeiteVorgang()*“, Abschnitt „Druckkennzeichnung der Anwendungsfälle“.

²⁶ Windows-x86_64\Beispiele\ericdemo-cpp\ericdemo\ericvorgang.cpp

²⁷ siehe Dokumentation\Datenartversionmatrix.xml und EHB, Kap. „datenartVersion – Definition und Verwendung“

²⁸ Für weitere Werte der Bearbeitungsflags, siehe *eric_types.h* in der API-Referenz.

- *verschluesselungsParameter* ist ein Zeiger auf die Struktur *eric_verschluesselung_parameter_t*. Mit ihr kann ein Zertifikat (Zertifikatshandle und PIN) für einen authentifizierten Versand übergeben werden. Wird kein Zertifikat benötigt, ist der Parameter ein *nullptr*.
Ob eine Datenart authentifiziert („send-Auth“) versendet werden kann oder sogar muss, ist der Eigenschaften-Tabelle der jeweiligen Datenart im EHB zu entnehmen.
Für den Testfall ESt 2020 benötigen Sie ein Zertifikat. Ein Testzertifikat können Sie aus dem Downloadbereich [Schnittstellenbeschreibungen](#) herunterladen. Siehe Kap. [3.2.2.2 Download des Testzertifikats](#).
- Der Parameter *transferHandle* ist nur in dem Sonderfall einer ElsterDatenabholung zu setzen. In allen anderen Fällen ist einfach ein *nullptr* zu übergeben. Daher wird das Transferhandle in diesem Beispiel nicht näher betrachtet.
- Die Parameter *ergebnisPuffer* und *serverantwortPuffer* sind vom Typ *EricRueckgabepufferHandle*. Beide müssen vor Übergabe an *EricBearbeiteVorgang()* mit *EricRueckgabepufferErzeugen()* angelegt werden. Im Code-Ausschnitt aus dem Beispielprogramm "ericdemo" geschieht dies durch die Instanziierung der Helferklasse *EricPuffer*, welche ebenfalls Teil des "ericdemo"-Projektes ist.

Mit den vorbereiteten Parametern können Sie *EricBearbeiteVorgang()* aufrufen und den Rückgabewert sowie die Inhalte von *ergebnisPuffer* und *serverantwortPuffer* auswerten. Zugriff auf die Pufferinhalte ermöglicht die API-Funktion *EricRueckgabepufferInhalt()*, im Code-Beispiel ausgeführt von der Methode *inhalt()* der Helferklasse *EricPuffer*.

Im *serverantwortPuffer* ist nach dem Versand von Daten die Antwort des ELSTER Annahmeservers enthalten. Mögliche serverseitige Fehlermeldungen können mit der API-Funktion *EricGetErrormessagesFromXMLAnswer()* aus dieser Antwort extrahiert werden.

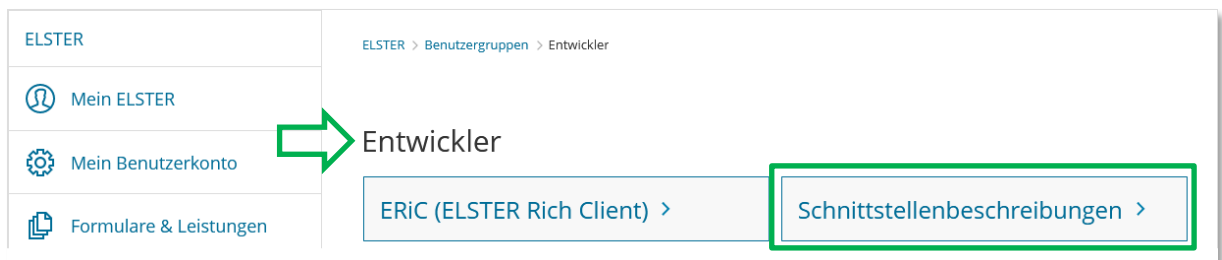
3.2.2.2 Download des Testzertifikats

Für den Testfall ESt 2020 benötigen Sie ein Test-Zertifikat. In den folgenden Schritten wird beschrieben, wo Sie Test-Zertifikate herunterladen können.

Hier finden Sie die Test-Zertifikate:

1. Klicken Sie auf <https://www.elster.de/elsterweb/entwickler/login> und melden Sie sich mit Ihren Hersteller-Logindaten im Entwickler-Bereich an.

Die Seite **Entwickler** öffnet sich.



2. Klicken Sie auf **Schnittstellenbeschreibungen**.

Die Seite **Schnittstellenbeschreibungen und Dokumentationen** öffnet sich.

3. Scrollen Sie zum Abschnitt **Dokumentationen** und laden Sie dort die **Test-Zertifikate** herunter.



3.2.2.3 Fehlerbehandlung am Beispiel der ERiC API-Funktion *EricBearbeiteVorgang()*

Ist ein Fehler bei der Verarbeitung aufgetreten ist der Rückgabewert, in diesem Beispiel „rc“, ungleich `ERIC_OK`. Ist der Rückgabewert gleich `ERIC_GLOBAL_PRUEF_FEHLER`, enthält der Rückgabepuffer Fehlermeldungen aus der Plausibilisierung der Steuerdaten. Der Themenbereich „Plausibilitätsprüfung“ ist im nächsten Kap. [3.2.3](#) ausführlich beschrieben.

Mit der ERiC API-Funktion *EricHoleFehlerText()* können Sie den Fehlertext zum Fehlercode ermitteln. Verschaffen Sie sich einen Überblick über die ERiC Fehlercodes indem Sie in der API-Referenz unter „Dateien“ `eric_fehlercodes.h` öffnen:

Abbildung 3-6 ERiC Fehlercodes in der API-Referenz

Typdefinitionen
<code>typedef enum eric_fehlercode eric_fehlercode_t</code>
Aufzählungen
<pre>enum eric_fehlercode { ERIC_OK = 0, ERIC_GLOBAL_UNKNOWN = 610001001, ERIC_GLOBAL_PRUEF_FEHLER = 610001002, ERIC_GLOBAL_HINWEISE = 610001003, ERIC_GLOBAL_FEHLERMELDUNG_NICHT_VORHANDEN = 610001007, ERIC_GLOBAL_KEINE_DATEN_VORHANDEN = 610001008, ERIC_GLOBAL_NICHT_GENUEGEND_ARBEITSSPEICHER = 610001013, ERIC_GLOBAL_DATEI_NICHT_GEFUNDEN = 610001014, ERIC_GLOBAL_HERSTELLER_ID_NICHT_ERLAUBT = 610001016, ERIC_GLOBAL_ILLEGAL_STATE = 610001017, ERIC_GLOBAL_FUNKTION_NICHT_ERLAUBT = 610001018, ERIC_GLOBAL_ECHTFALL_NICHT_ERLAUBT = 610001019, ERIC_GLOBAL_NO_VERSAND_IN_BETA_VERSION = 610001020, ERIC_GLOBAL_TESTMERKER_UNGUELTIG = 610001025, ERIC_GLOBAL_DATENSATZ_ZU_GROSS = 610001026, ERIC_GLOBAL_VERSCHLUESSELUNGS_PARAMETER_NICHT_ERLAUBT = 610001027,</pre>

Ein optionales Element `<ElsterInfo>` in der XML-Serverantwort enthält Hinweise für den Endanwender, beispielsweise über den Ablauf des verwendeten Zertifikats. Diese können Sie in Ihrer Steuersoftware dem Endanwender zur Verfügung stellen.

3.2.3 Schritt 3 - Plausibilitätsprüfungen

In diesem Kapitel lernen Sie den Umgang mit fehlerhaften Datensätzen. Im nachfolgenden Beispiel werten Sie den API-Rückgabewert und den *ergebnisPuffer*, der die Fehlermeldung enthält, aus.

Ändern Sie in Ihren erstellten eXML-Daten²⁹ das Geburtsdatum im Hauptvordruck des Steuerpflichtigen wie folgt, damit ein Plausibilitätsfehler auftritt:

```
<E0100401>05.17.1955</E0100401>
```

Wenn Sie jetzt „ericdemo“ in der Kommandozeile³⁰ ausführen, erhalten Sie folgende Fehlermeldung im *ergebnisPuffer*:

```
Sendestatus: Fehler während der Plausibilitätsprüfung, Datensatz nicht plausibel. Zur Ermittlung der fehlgeschlagenen Plausibilitätsprüfungen muss der Rückgabepuffer (Parameter "rueckgabeXmlPuffer") ausgewertet werden.

Rückgabe:
<?xml version="1.0" encoding="UTF-8"?>
<EricBearbeiteVorgang xmlns="http://www.elster.de/EricXML/1.0/EricBearbeiteVorgang">
  <FehlerRegelpruefung>
    <Nutzdatenticket>adfjasldkfweor456asvs</Nutzdatenticket>
    <Feldidentifikator>/Est1A[1]/Allg[1]/A[1]/E0100401[1]</Feldidentifikator>
    <Mehrfachzeilenindex>1</Mehrfachzeilenindex>
    <LfdNrVordruck>1</LfdNrVordruck>
    <RegelName>formalePruefung</RegelName>
    <FachlicheFehlerId>datumFormatFalsch</FachlicheFehlerId>
    <Text>Das Datum im Feld '$/Est1A[1]/Allg[1]/A[1]/E0100401[1]$_' ist nicht gültig. (Erforderliches Format: TT.MM.JJJJ).</Text>
  </FehlerRegelpruefung>
</EricBearbeiteVorgang>
```

Kontext mit Feldkennung = „E0100401“

Fehlercode = „datumFormatFalsch“

Zunächst öffnen Sie die Logdatei „eric.log“. Hier sehen Sie, dass der Aufruf mit dem Returncode „610001002“ beendet wurde.

```
ERROR: Fehler während der Plausibilitätsprüfung, Datensatz nicht plausibel. ...
INFO: }} Ausstieg >>EricMtBearbeiteVorgang<< rc = 610001002
```

Den korrespondierenden Fehlertext (rot markiert) zum Returncode „610001002“ sehen Sie in der API-Referenz, Datei *eric_fehlercodes.h*.

ERIC_GLOBAL_PRUEF_FEHLER	[610001002] Fehler während der Plausibilitätsprüfung, Datensatz nicht plausibel.
ERIC_GLOBAL_HINWEISE	[610001003] Hinweise während der Plausibilitätsprüfung, Datensatz ist aber plausibel.

Programmatisch ist *EricHoleFehlerText()* zu verwenden.

Beim Fehlercode = „datumFormatFalsch“ handelt es sich um einen formalen Fehler. Im Dokument *Zusatzinformationen_zur_Plausibilitaetspruefung.pdf*³¹ finden Sie die dazugehörige Prüfbedingung und Fehlerbeschreibung.

Die Plausibilitätsfehler werden unterschieden in:

- Formale Fehler innerhalb eines Feldes:
 - *Zusatzinformationen_zur_Plausibilitaetspruefung.pdf*³²
 - *Formale_Plausipruefungen.xml*³³

- Formale_Plausipruefungen_Steuerartuebergreifend.xml³²
- Inhaltliche Fehler / Plausibilitäten:
 - Dokumentation\Plausipruefungen\Erklaerungssteuern\ES\UFA10*.xml

Das Dokument „Zusatzinformationen_zur_Plausibilitaetspruefung.pdf“ informiert Sie auch über die in der obigen Dokumentation verwendeten Terminologie und Struktur, bzw. über die Regelsprache und Regeln.

Öffnen Sie jetzt die Datei „Jahresdokumentation_10_2020.xml“³⁴, um einen Überblick über die inhaltlichen Fehler / Plausibilitätsfehler der ESt für den Veranlagungszeitraum (VZ) 2020 zu bekommen.

Für den Hauptvordruck und jede ESt Anlage sind jeweils fünf Tabellenblätter (Kontexte, Felder, Regeln, Kennzahlen und Texte) vorhanden.

Das vorletzte Tabellenblatt „Allg. Information“ enthält eine Liste der Vordrucke mit der Angabe des maximal zulässigen Wertes für lfd. Nr. Vordruck, eine Legende über die verwendete farbliche Kennzeichnung und eine Liste inkl. Beschreibung der sogenannten Formatkennzeichen³⁵.

Das letzte Tabellenblatt „Formale Fehler“ listet die Fehler-IDs, Beschreibung und Fehlertexte der in der Regelsprache auftretenden formalen Fehler für diesen VZ auf.

Die Vielzahl der Plausibilitätsfehler in der Jahresdokumentation resultiert aus der Menge der Felder und deren Abhängigkeiten zueinander.

³⁴ Dokumentation\Plausipruefungen\Erklaerungssteuern\ES\UFA10

³⁵ siehe EHB, Kap. „Das ERiC Dokumentationspaket“

3.2.4 Schritt 4 - Übermittlung an den ELSTER Annahmeserver

Ist die Plausibilisierung fehlerfrei abgeschlossen, übermittelt der ERiC abhängig vom Anwendungsfall die Daten des Steuerpflichtigen zum ELSTER Annahmeserver. Dieser nimmt ebenfalls Dateneingangsprüfungen vor und meldet dann die Annahme oder Abweisung der Daten zurück an den ERiC. Die Antwort wird nun über den Returncode bzw. im Fehlerfall über den Rückgabepuffer der Funktion *EricBearbeiteVorgang()* an den Aufrufer gemeldet.

Öffnen Sie jetzt die zuvor verwendeten eXML-Daten und korrigieren Sie den Fehler im Datum. Führen Sie „ericdemo“ von der Kommandozeile erneut aus:

```
Sendestatus: Verarbeitung fehlerfrei.

Rückgabe:
<?xml version="1.0" encoding="UTF-8"?>
<EricBearbeiteVorgang xmlns="http://www.elster.de/EricXML/1.0/EricBearbeiteVorgang">
  <Erfolg>
    <Telenummer>AAP</Telenummer>
  </Erfolg>
</EricBearbeiteVorgang>

Serverantwort:
<?xml version="1.0" encoding="UTF-8"?><Elster xmlns="http://www.elster.de/elsterxml/schema/v11"><TransferHeader version="11"><Verfahren>Elst
erErklaerung</Verfahren><DatenArt>Est</DatenArt><Vorgang>send-Auth</Vorgang><TransferTicket>et08129gipxgnj50g74iz2d5a9mzoemq</TransferTicket
><Testmerker>700000004</Testmerker><Empfaenger id="L"><Ziel>BY</Ziel></Empfaenger><HerstellerID>74931</HerstellerID><DatenLieferant>MIAGCSqG
SIb3DQEHbqCAMIACAQAwgAYJKoZIhvcNAQcBMBQGCCqGSIb3DQMHBAhdfE+5PKKcz6CA
```

Sie sehen, dass der Versand mit Zertifikat („send-Auth“) erfolgte.

War der Versand fehlerfrei (*ERIC_OK*), antwortet „ericdemo“ mit „Sendestatus: Verarbeitung fehlerfrei.“. Der *rueckgabeXMLPuffer* besteht im Erfolgsfall aus:

- der Telenummer als Zuordnungskriterium im Finanzamt (dreistellige Zeichenkette)
- dem optionalen Ordnungsbegriff (nur bei Neuaufnahme³⁶)

Der Parameter *serverantwortXMLPuffer* besteht aus:

- der Serverantwort im XML-Format (im obigen Screenshot abgeschnitten, Details im nächsten Kap. [3.2.4.1 Auswertung der Serverantwort im XML-Format](#))

Die Einträge in eric.log bestätigen Ihnen ebenfalls die erfolgreiche Datenübermittlung.

³⁶ Bei der Neuaufnahme werden Steuerdaten ohne Steuernummer übermittelt. Welche Datenarten dies unterstützen und welche Besonderheiten zu beachten sind, siehe EHB, Kap. „Neuaufnahmen (Steuerdaten ohne Steuernummer)“

3.2.4.1 Auswertung der Serverantwort im XML-Format

Im Beispiel des *serverantwortXMLPuffer* sehen Sie, dass kein Fehler vorliegt, der `<Code>` ist 0 und im `<Text>` wird bestätigt, dass die Daten erfolgreich angenommen wurden.

```
<?xml version="1.0" encoding="UTF-8"?>
<Elster xmlns="http://www.elster.de/elsterxml/schema/v11">
  <TransferHeader version="11">
    ...
    ...
    <RC>
      <Rueckgabe>
        <Code>0</Code>
        <Text>Daten wurden erfolgreich angenommen.</Text>
      </Rueckgabe>
    ...
    ...
  </RC>
</TransferHeader>
<DatenTeil></DatenTeil>
</Elster>
```

Tritt ein Fehler auf und ist der Rückgabewert *ERIC_TRANSFER_ERR_XML_THEADER* oder *ERIC_TRANSFER_ERR_XML_NHEADER*, dann müssen das „eric.log“ und der *serverantwortXMLPuffer* ausgewertet werden. Hierzu kann entweder die Serverantwort selbst geparkt werden oder Sie verwenden *EricGetErrorMessageFromXMLAnswer()*³⁷. Diese Funktion liefert Ihnen alle Fehlermeldungen, den Returncode und das Transferticket. Mit dem Transferticket lässt sich jede Steuererklärung identifizieren, sie ist auch dem ericprint.pdf³⁸ aufgedruckt.

³⁷ Die API-Referenz enthält die Funktionssignatur und Funktionsbeschreibung.

³⁸ In der mitgelieferten Beispieldatei ericprint.pdf finden Sie das Transferticket auf jeder Seite am linken Rand vertikal aufgedruckt.

3.2.5 Schritt 5 - PDF-Erstellung

Nach einer fehlerfreien Verarbeitung wird abhängig vom Anwendungsfall ein PDF-Dokument erstellt. Stellen Sie sicher, dass ein schreibender Zugriff in das Arbeitsverzeichnis³⁹ möglich ist.

Öffnen Sie jetzt das erstellte PDF-Dokument „ericprint.pdf“ im Arbeitsverzeichnis.

Die Seite 1 sollte wie folgt aussehen:

Finanzamt München (181) Überschusseinkünfte		Steuernummer 181/815/08155	Datum der Ausfertigung: 26.03.2021 Seite 1 von 4
*** Testfall ***		*** Testfall ***	2020
Hauptvordruck Est 1 A			
Art der Erklärung			
Einkommensteuererklärung			
Festsetzung der Arbeitnehmer - Sparzulage			
Erklärung zur Feststellung des verbleibenden Verlustvortrags			
Allgemeine Angaben			
Steuerpflichtige Person. Nur bei Zusammenveranlagung: Ehemann oder Person A (Ehegatte A / Lebenspartner(in) A nach dem LPartG)			
7	Geburtsdatum	05.05.1955	
8	Name	Maier	
9	Vorname	Hans	
10	Titel, akademischer Grad	Dr.	
10	Religion	Römisch-katholisch	
11	Straße (derzeitige Adresse)	Musterstr.	
12	Hausnummer	55	
12	Hausnummerzusatz	c	
13	Postleitzahl (Inland)	55555	
14	Wohnort	Musterort	
16	Ausgeübter Beruf	Industriemechaniker	
17	Verheiratet / Lebenspartnerschaft begründet seit dem	02.08.2009	
Nur bei Zusammenveranlagung: Ehefrau oder Person B (Ehegatte B / Lebenspartner(in) B nach dem LPartG)			
Bitte füllen Sie die Zeilen 22 bis 26 nur aus, wenn die Adressangaben von den Zeilen 11 bis 15 abweichen			
18	Geburtsdatum	09.07.1988	
19	Name	Maier	
20	Vorname	Carolina	
21	Religion	Römisch-katholisch	
27	Ausgeübter Beruf	Floristin	
Bei Ehegatten / Lebenspartnern: Veranlagungsart			
28	Zusammenveranlagung		
Bankverbindung			
31	IBAN (inländisches Geldinstitut)	DE911000000001234567	
		89	
34	Kontoinhaber ist die steuerpflichtige Person, nur bei Zusammenveranlagung: Ehemann / Person A		
34	Kontoinhaber ist die Ehefrau / Person B		
Antrag auf Festsetzung der Arbeitnehmer-Sparzulage: Steuerpflichtige Person / Ehemann / Person A			
42	Für alle vom Anbieter übermittelten Vermögensbildungsbescheinigungen wird die Festsetzung der Arbeitnehmer-Sparzulage beantragt	1 (= Ja)	
ELSTER		gedruckt mit ERIC-Print 33.3.2/EST_2020 33.3.2	

Im Kap. [2.8](#) haben wir einen Testmarker in die eXML-Daten eingetragen, um den Datensatz als Testfall zu kennzeichnen. Der Testmarker führt auch dazu, dass das erzeugte PDF mit dem in Rot gedruckten „***** Testfall *****“ eine Kennzeichnung als Testfall erhält.

3.3 Dritte Aufgabe: Neue und geänderte Felder in die Steuersoftware integrieren

Damit Sie Ihre Steuersoftware an einen neuen Veranlagungszeitraum (VZ) oder Anmeldungszeitraum (AZ) anpassen können, sehen Sie sich folgende Dokumente an:

- In der Deltadokumentation⁴⁰ werden die Änderungen gegenüber dem letzten ERiC Update bzw. Patch dargestellt. Eine Deltadokumentation für einen VZ/AZ existiert aber nur dann, wenn es an dem VZ/AZ fachliche Änderungen zur vorangegangenen ERiC-Version gegeben hat.
- Die Jahresdokumentation⁴¹ hingegen enthält die Änderungen gegenüber dem Vorjahres-Zeitraum aus dem letzten Patch des ERiC-Vorjahres-Release. Dabei sind sämtliche Änderungen der bisher für dieses ERiC-Release ausgelieferten Updates oder Patches mit enthalten.

Alle Änderungen (Feld- und Regeländerungen) in der Delta- oder/und der Jahresdokumentation sind farbig hervorgehoben. Eine programmatische Auswertung ist über die Tabellenspalte „Änderungsinformation“ möglich. Die Änderungen sind auch in den annotierten amtlichen Vordrucken farbig abgebildet.

³⁹ Kann mit *EricInitialisiere()* gesetzt werden, siehe API-Referenz.

⁴⁰ Dokumentation\Deltadokumentation*

⁴¹ Dokumentation\Plausipruefungen*

3.3.1 Beispiel zum VZ-Wechsel für die ESt von 2016 auf 2017

Öffnen Sie den Vordruck „2017AnlKind021.png“⁴² (Anlage Kind, Seite 1) um Änderungen anhand der farbigen Markierung zu identifizieren. Beispielsweise ist die Feldkennung „0501513“ und „0501610“ neu hinzugekommen und der Beschreibungstext „Angaben für ein volljähriges Kind“ wurde geändert.

Hingegen sind die Feldkennungen „0501310“, „0501509“, „0501609“ und „0501702“ ab VZ 2017 nicht mehr möglich.

Abbildung 3-7 Ausschnitt aus Vordruck 2017AnlKind021.png

15	Der Wohnsitz oder gewöhnliche Aufenthalt des anderen Elternteiles ist nicht zu ermitteln oder der Vater des Kindes ist amtlich nicht feststellbar	0501513 05 NEU! 1 = Ja
Angaben für ein volljähriges Kind		gelöscht: 0501310, 0501509, 0501609, 0501702
<p>Das Kind</p> <ul style="list-style-type: none"> – befand sich in einer Schul-, Hochschul- oder Berufsausbildung, – befand sich in einer Übergangszeit von höchstens vier Monaten (z. B. zwischen zwei Ausbildungsabschnitten), – konnte eine Berufsausbildung mangels Ausbildungsplatzes nicht beginnen oder fortsetzen und / oder – hat ein freiwilliges soziales oder ökologisches Jahr (Jugendfreiwilligendienstgesetz), einen europäischen / entwicklungspolitischen Freiwilligendienst, einen Freiwilligendienst aller Generationen (§ 2 Abs. 1a SGB VII), einen Internationalen Jugendfreiwilligendienst, Bundesfreiwilligendienst oder einen Anderen Dienst im Ausland (§ 5 Bundesfreiwilligendienstgesetz) geleistet. (Folgt diese Abschnitte unmittelbar aufeinander, sind sie zu einem Zeitraum zusammenzufassen.) <p>1. Zeitraum 2. Zeitraum</p> <p>vom bis vom bis</p>		
16	80 0501610 (MZI=12) NEU!	81

⁴² Vordrucke\archive\2017\Jahressteuern\ESt\Grafiken_und_Erweiterungen_UFA10\

Die neue Feldkennung „0501610“ soll in der Steuersoftware aufgenommen werden. Hierzu müssen Sie das Feldformat kennen und wissen, ob es Abhängigkeiten zu anderen Feldern und Regeln gibt. Das erfahren Sie aus dem Dokument „Jahresdokumentation_10_2017.xml“ bzw. „Deltadokumentation_10_2017.xml“. Die „10“ im Dateinamen ist die Unterfallart (UFA). Diese kann dem EHB, Kap. „Unterstützte Fachverfahren und Daten- / Steuerarten“ entnommen werden.

Öffnen Sie jetzt das Dokument „Jahresdokumentation_10_2017.xml“ mit Excel oder einer kompatiblen Anwendung.

Angaben_fuer_ein_volljaehrigen_Kind	0501408	Bezeichnung der Schul-, Hochschul- oder Berufsausbildung	12	String
Angaben_fuer_ein_volljaehrigen_Kind	0501408 (alt)	Bezeichnung der Schul-, Hochschul- oder Berufsausbildung	6	String
Angaben_fuer_ein_volljaehrigen_Kind	0501802	vom - bis	6	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_fuer_ein_volljaehrigen_Kind	0501905	vom - bis	2	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_fuer_ein_volljaehrigen_Kind	0502009	vom - bis	2	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_fuer_ein_volljaehrigen_Kind	0501610	Zeitraum vom - bis	12	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_fuer_ein_volljaehrigen_Kind	0501348	Ausbildungsabschnitt vom - bis	6	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_fuer_ein_volljaehrigen_Kind	0501509	vom - bis	6	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_fuer_ein_volljaehrigen_Kind	0501609	vom - bis	6	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_fuer_ein_volljaehrigen_Kind	0501702	vom - bis	6	DatumBereich TT.MM.JJJJ-TT.MM.JJJJ mit Zusatzprüfung
Angaben_zur_Erwerbstaetigkeit_eines_volljaehrigen_Kindes	0502204	Das Kind hat bereits eine erstmalige Berufsausbildung oder ein Erststudium abgeschlossen (1 = Ja / 2 = Nein)	1	JaNein (1 2)
Angaben_zur_Erwerbstaetigkeit_eines_volljaehrigen_Kindes	0502401	Falls Zeile 21 mit Ja beantwortet wurde: Das Kind war erwerbstätig (kein Ausbildungsverhältnis) (1 = Ja / 2 = Nein)	1	JaNein (1 2)
Angaben_zur_Erwerbstaetigkeit_eines_volljaehrigen_Kindes	0502401 (alt)	Falls Zeile 23 mit Ja beantwortet wurde: Das Kind war erwerbstätig (kein Ausbildungsverhältnis) (1 = Ja / 2 = Nein)	1	JaNein (1 2)
Angaben_zur_Erwerbstaetigkeit_eines_volljaehrigen_Kindes	0502501	Das Kind übte eine / mehrere geringfügige Beschäftigung(en) im Sinne der §§ 8, 8a SGB IV (sogenannter Minijob) aus (1 = Ja / 2 = Nein)	1	JaNein (1 2)
Angaben_zur_Erwerbstaetigkeit_eines_volljaehrigen_Kindes	0502502	Beschäftigungszeitraum vom - bis	12	DatumBereich TT.MM-TT.MM mit Zusatzprüfung
Angaben_zur_Erwerbstaetigkeit_eines_volljaehrigen_Kindes	0502616	Das Kind übte andere Erwerbstätigkeiten aus (1 = Ja / 2 = Nein)	1	JaNein (1 2)
Angaben_zur_Erwerbstaetigkeit_eines_volljaehrigen_Kindes	0502617	Erwerbszeitraum vom - bis	12	DatumBereich TT.MM-TT.MM mit Zusatzprüfung

Suchen Sie nach der Feldkennung „0501610“ im Arbeitsblatt „Kind – Felder“. Neue Zeilen sind rot markiert, gelbe Zeilen signalisieren eine Änderung, graue und weiße Zeilen enthalten keinerlei Änderungen. Der gefundenen Zeile entnehmen Sie das Feldformat, die Beschreibung und stellen fest, dass es sich um ein optionales Feld handelt. Mit diesen Informationen können Sie das neue Feld in Ihre Steuersoftware, wie bereits in den vorigen Kapiteln beschrieben, aufnehmen und die XML-Daten erweitern.

Gehen Sie nun analog für alle weiteren geänderten Felder vor, achten Sie hierbei auf Abhängigkeiten zu anderen Feldern und Regeln. Validieren Sie das aktualisierte XML mit „ericdemo“, um diese Aufgabe abzuschließen.

3.4 Vierte Aufgabe: Den TransferHeader mit EricCreateTH() erstellen

Neben dem XML-Datenteil muss das Steuerdaten-XML einen TransferHeader enthalten. Zur Erstellung des TransferHeaders ist die Verwendung der API-Funktion *EricCreateTH()* empfohlen, da sie folgende Vorteile bietet:

- Die manuelle, fehlerträchtige Erstellung des TransferHeaders entfällt.
- Pflichtelemente und -werte werden automatisch erzeugt und korrekt befüllt, z.B. das Element `<Datei>` wird mit den korrekten Werten versorgt.
- Die richtige Reihenfolge der Elemente laut Schemadefinition wird gewährleistet.
- Der TransferHeader im XML bleibt schemakonform unabhängig vom ERiC Release.
- Es kann das gesamte Steuerdaten-XML oder nur der Datenteil im Parameter `xml` übergeben werden.

Die Verwendung von *EricCreateTH()* im C++ Beispiel `ericdemo` wird nachfolgend demonstriert.

3.4.1 Schritt 1 - XML-Eingabedatei erstellen

Entnehmen Sie aus der Datei `EST_2020.xml` den Datenteil und speichern diesen in einer neuen Datei ab, z.B. in `EST_2020_Datenteil.xml`.

Schematischer Aufbau der Datei `EST_2020_Datenteil.xml`:

```
<DatenTeil>
  <Nutzdatenblock>
    <NutzdatenHeader version="11">
      ...
      ...
      ...
    </NutzdatenHeader>
    <Nutzdaten>
      ...
      ...
      ...
    </Nutzdaten>
  </Nutzdatenblock>
</DatenTeil>
```

3.4.2 Schritt 2 - EricCreateTH() in ericdemo aufrufen

Der typedef, die Funktionszeiger, der Wrapper und die Implementierung der Proxymethode für *EricCreateTH()* sind bereits in ericdemo enthalten und können ohne Anpassung verwendet werden, siehe eric.h und eric.cpp.

Fügen Sie die folgende neue Methode „erzeugeTransferHeader“ in ericvorgang.cpp am Ende ein. Der API-Aufruf von *EricCreateTH()* verwendet in diesem Beispiel feste Übergabewerte:

```
int EricVorgang::erzeugeTransferHeader(std::string& ergebnis)
{
    EricPuffer ergebnisPuffer(ericAdapter);

    int rc = ericAdapter.EricCreateTH(
        xmlDaten.c_str(),           // zuvor eingelesene Steuerdaten
        "ElsterErklaerung",        // Verfahren
        "Est",                      // Datenart
        "send-Auth",               // Vorgang
        "7000000004",              // Testmerker
        "74931",                   // HerstellerID
        "Softwaretester ERiC",      // Datenlieferant
        "MeineSteuerSoftware 2.4",  // versionClient
        NULL,                      // publicKey
        ergebnisPuffer.handle()); // Rückgabepufferhandle

    ergebnis = ergebnisPuffer.inhalt();

    if (rc == 0)
        xmlDaten = ergebnis;

    return rc;
}
```

Zur neuen Methode "EricVorgang::erzeugeTransferHeader" muß in ericvorgang.h hinter der Deklaration leseDatensatz() auch eine passende Methodendeklaration eingefügt werden:

```
/** @brief Reichere ein XML für eine Est 2020 mit einem Elster-Transferheader
an
*/
int erzeugeTransferHeader(std::string& ergebnis);
```

Fügen Sie den folgenden Code mit dem neuen Methodenaufruf „createTH“ in ericdemo.cpp nach dem Einlesen des Datensatzes

(*vorgang.leseDatensatz (argParser.getDatensatzDatei ())*;) ein.

```
//TransferHeader mit EricCreateTH() erstellen
System::titelZeile("Den TransferHeader mit EricCreateTH() erstellen.");
std::string ergebnisXML;

fehlerkode = vorgang.erzeugeTransferHeader(ergebnisXML);

if (fehlerkode == ERIC_OK)
    std::cout << ergebnisXML << std::endl;
```

Übersetzen und linken Sie ericdemo, um es als ausführbares Programm für die Konsole zu erstellen.

Führen Sie auf der Konsole „startedemo-x64 /v ESt_2020 /x ESt_2020_Datenteil.xml“ aus und verfolgen Sie die Ausgaben von ericdemo.

Der Rückgabepuffer von *EricCreateTH()* enthält das mit dem TransferHeader angereicherte XML. Es wird auf der Konsole ausgegeben. Kontrollieren Sie in der Konsole den erstellten TransferHeader.

Der Aufruf von *EricCreateTH()* war erfolgreich: Die Übergabewerte sind zusammen mit den erstellten Pflichtelementen und -werten im TransferHeader zu finden.

4 ERiC optimal in die Steuersoftware am Beispiel ESt 2020 integrieren

Nach dem Abschluss der Entwicklungs- und Testphase der Steuersoftware wird der ERiC in die Steuersoftware integriert. Dieser Abschnitt zeigt anhand des Beispiels ESt 2020, welche ERiC Bibliotheken mindestens in die an den Endkunden auszuliefernde Steuersoftware zu integrieren sind, und welche Vorteile sich daraus ergeben.

Lesen Sie zuerst die Grundlagen im EHB, Kap. „Beste Strategie für die Integration einer ERiC Auslieferung in die Steuersoftware“.

Voraussetzungen für das Beispiel ESt 2020 für Windows, 64-Bit:

- Sie haben das ERiC Softwarepaket ERiC-*<version>*-Windows-x86_64.jar⁴³ installiert. Für das Beispiel verwenden wir die *<version>* 37.2.6.0.
Für Informationen zur Auswahl des benötigten ERiC Softwarepakets siehe EHB, Kap. „Das ERiC Softwarepaket auswählen“.
- Sie haben das Dokumentations-Paket ERiC-*<version>*-Dokumentation.zip installiert.

So ermitteln Sie die notwendigen ERiC Plugin-Bibliotheken:

- Öffnen Sie die Datei Dokumentation\Datenartversionmatrix.xml und ermitteln Sie für die datenartVersion⁴⁴ „ESt_2020“ den ERiC Plugin-Bibliotheksnamen:

	1	2	3	4	5	6	7
1	Verfahren	Datenart(en)	Jahr / Version	datenartVersion-Parameter	Plugin-Bibliothek	Verzeichnis	Bemerkung
10	ElsterErklaerung	ESt	<Jahr>: ab 2012	ESt_<Jahr>	checkEST_<Jahr>	plugins2	neue Nutzdatenstruktur ab 2020

[A] Es ist die ERiC Plugin-Bibliothek „checkEst_2020.dll“ zu verwenden.

[B] Die ERiC Plugin-Bibliothek „commonData.dll“ wird grundsätzlich immer benötigt.⁴⁵

- Alle anderen ERiC Plugin-Bibliotheken können Sie im Verzeichnis plugins2 löschen.⁴⁵

⁴³ siehe EHB, Kap. „Inhalt des ERiC“

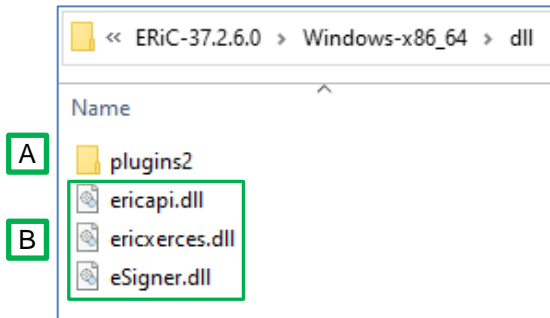
⁴⁴ siehe EHB, Kap. „datenartVersion – Definition und Verwendung“

⁴⁵ siehe EHB, Kap. „Dynamische Programmbibliotheken“ → „ERiC Plugins in den Verzeichnissen „plugins2“ und „plugins““



HINWEIS:

Im Gegensatz zu den ERiC Plugin-Bibliotheken müssen die ERiC Basisbibliotheken komplett in die Steuersoftware übernommen werden.



[A] ERiC Plugin-Bibliotheken im Verzeichnis plugins2.

[B] ERiC Basisbibliotheken im Verzeichnis dll.

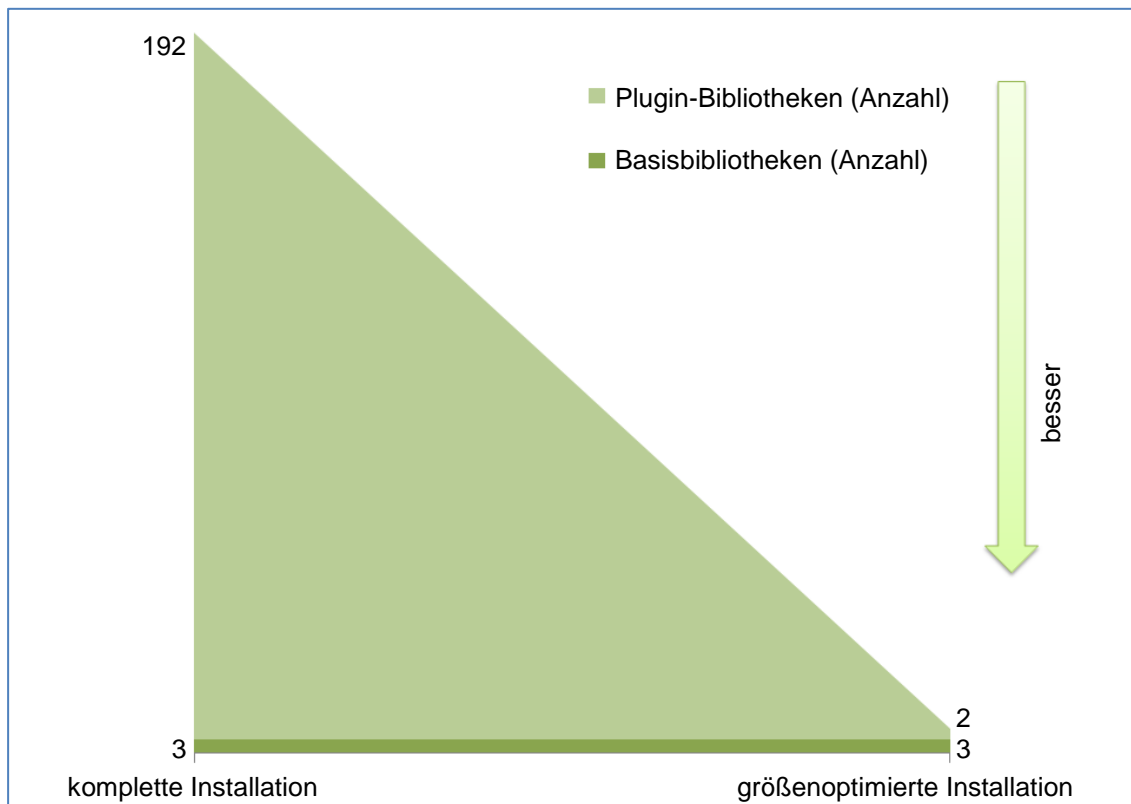
Mit den vorangegangenen Schritten wurden die ERiC Plugin-Bibliotheken für das Beispiel ESt 2020 auf die benötigte minimale Anzahl reduziert. Die daraus resultierenden Vorteile sind in der [Tabelle 4-1](#) und der [Abbildung 4-1](#) zusammengefasst.

Tabelle 4-1 Beispiel ESt 2020: Komplette versus größenoptimierter Installation mit ERiC 37.2.6

	Komplette Installation Dateien / Speicherplatz	Gelöschte Dateien / Speicherplatz	Größenoptimierte Installation Dateien / Speicherplatz
Basisbiblio- theken	3 / 15,6 MiB ⁴⁶	0 / 0 MiB	3 / 15,6 MiB
Verzeichnis plugins2	192 / 1,32 GiB	190 / 1,30 GiB	2 / 21,1 MiB
Summe	195 / 1,34 GiB	190 / 1,30 GiB	5 / 36,7 MiB

⁴⁶ 1 MiB = 1024 x 1024 Byte = 1.048.576 Byte, 1 MB = 1.000.000 Byte, siehe auch <https://de.wikipedia.org/wiki/Bin%C3%A4rpr%C3%A4fix> und <https://de.wikipedia.org/wiki/Byte#Vergleich>

Abbildung 4-1 Anzahl ERiC Plugin-Bibliotheken und Basisbibliotheken der kompletten gegenüber der größenoptimierten ERiC Installation



Es wird empfohlen, die Vorteile der größenoptimierten ERiC Installation zu nutzen:

- Der benötigte Speicherplatz wird reduziert.
- Der ERiC Ladevorgang wird beschleunigt, ebenso das Entladen.
- Die ERiC Funktionalität wird auf den gewünschten Umfang beschränkt.

5 Fortschrittcallbacks am Beispiel ericdemo implementieren

Die Verarbeitungsschritte von *EricBearbeiteVorgang()* und deren Fortschritte können dem Endanwender durch Callbackfunktionen angezeigt werden.

Lesen Sie hierzu die Grundlagen im EHB, Kap. „Funktionen für Fortschrittcallbacks“, und informieren Sie sich über die zugehörigen Funktionssignaturen und Typen in der API-Referenz (ericapi.h und eric_types.h).

Die folgende Abbildung zeigt das gewünschte Ergebnis in der Beispielanwendung ericdemo:

- Der globale Fortschrittcallback soll die Variable *pos* und *max* mit einem von der *id* abhängigen Text ausgeben.
- Der lokale Fortschrittcallback soll eine einfach textbasierte Fortschrittanzeige implementieren.

Abbildung 5-1 Die Ausgaben der Fortschrittcallbacks in ericdemo

```

xxx Validiere Steuerdatensatz xxx
0/2: XML einlesen
[#####]
1/2: Validieren
[#####]
2/2: Validieren
Validiere steuersatz.xml: ok
  
```

Sehen Sie sich im Folgenden die entsprechenden Codeausschnitte der Implementation in ericdemo an.

Der Fortschritt wird von der Klasse *CallbackHandler* auf der Konsole visualisiert. Die Funktion *mapId()* ordnet jeder *id* eines Verarbeitungsschritts von *EricBearbeiteVorgang()* einen passenden Text zu.

Die Methode *globalerFortschritt()* fügt den Inhalt der Variablen *pos*, *max* und den Text aus *mapId()* zu einer Ausgabezeile zusammen, beispielsweise „0/2: XML Einlesen“.

Die Fortschrittsanzeige innerhalb eines Verarbeitungsschrittes wird von *fortschritt()* erstellt, beispielsweise „[#####]“.

```
std::string mapId(uint32_t id) {
    switch (id) {
        case ERIC_FORTSCHRITTCALLBACK_ID_EINLESEN:
            return "XML einlesen";
        case ERIC_FORTSCHRITTCALLBACK_ID_VORBEREITEN:
            return "Versand vorbereiten";
        case ERIC_FORTSCHRITTCALLBACK_ID_VALIDIEREN:
            return "Validieren";
        case ERIC_FORTSCHRITTCALLBACK_ID_SENDEN:
            return "Versenden";
        case ERIC_FORTSCHRITTCALLBACK_ID_DRUCKEN:
            return "Drucken";
        default:
            throw std::runtime_error("Unbekannte ID");
    }
}

void CallbackHandler::globalerFortschritt(uint32_t id, uint32_t pos, uint32_t max) const {
    std::cout << pos << "/" << max << ": " << mapId(id) << std::endl(*stdout);
}

void CallbackHandler::fortschritt(uint32_t id, uint32_t pos, uint32_t max) {
    if (letzteId != 0 && letzteId != id) {
        System::out << std::string(78 - letzteSpalte, '#') << ']' <<
        std::endl(*stdout);
        letzteId = 0;
        letzteSpalte = 0;
        fortschritt(id, pos, max);
    } else {
        if (pos == 0) {
            (System::out << '[' << (*stdout);
            letzteId = id;
            letzteSpalte = 0;
        } else if (pos == max) {
            System::out << std::string(78 - letzteSpalte, '#') << ']' <<
            std::endl(*stdout);
            letzteId = 0;
            letzteSpalte = 0;
        } else {
            unsigned int spalte = (pos * 78) / max;
            (System::out << std::string(spalte - letzteSpalte, '#'))
            (*stdout);
            letzteSpalte = spalte;
        }
    }
}
```


Der Aufruf der Methoden `globalerFortschritt()` und `fortschritt()` erfolgt jeweils über eine Callbackfunktion in `callbackhandler.cpp`. Diese Funktionen sind erforderlich, weil die ERiC-Schnittstelle nicht objektorientiert sein darf und daher die Methoden nicht direkt am `CallbackHandler` aufgerufen werden können.

Die beiden Callbackfunktionen erwarten im zweiten Parameter `userData` einen Zeiger auf den `CallbackHandler` an dem die Methoden `globalerFortschritt()` und `fortschritt()` aufgerufen werden:

```
static void STDCALL globalerFortschrittAdapter(uint32_t id, uint32_t pos,
uint32_t max, void *userData) {
    const CallbackHandler *const handler = reinterpret cast<const
CallbackHandler *>(userData);
    handler->globalerFortschritt(id, pos, max);
}

static void STDCALL fortschrittAdapter(uint32_t id, uint32_t pos, uint32_t
max, void *userData) {
    CallbackHandler *const handler = reinterpret cast<CallbackHandler
*>(userData);
    handler->fortschritt(id, pos, max);
}
```

Damit `EricBearbeiteVorgang()` die beiden Callbackfunktionen aufruft und ihnen dabei den `CallbackHandler` übergibt, sind die Funktionen (erster Parameter) und der Handler (zweiter Parameter) mit den API-Funktionen `EricRegistriereGlobalenFortschrittCallback()` bzw. `EricRegistriereFortschrittCallback()` zu registrieren:

```
eric.EricRegistriereGlobalenFortschrittCallback(globalerFortschrittAdapter,
this);
eric.EricRegistriereFortschrittCallback(fortschrittAdapter, this);
```

Das Abmelden der beiden Callback-Funktionen im Destructor nicht vergessen:

```
eric.EricRegistriereGlobalenFortschrittCallback(NULL, NULL);
eric.EricRegistriereFortschrittCallback(NULL, NULL);
```

Damit sind die wichtigsten Codestellen in `ericdemo` für die Fortschrittcallbacks betrachtet, die Ausgabe erfolgt wie in [Abbildung 5-1](#) zu sehen.

6 Datenabholung mit Anhängen am Beispiel „SonstigeNachrichten“

Die wichtigsten Verarbeitungsschritte für die meisten Datenarten sind bereits im Kap. [3](#) anschaulich beschrieben. Ergänzend hierzu unterstützen derzeit einige und in Zukunft immer mehr Datenarten die asynchrone Datenabholung (mit Anhängen). Die Datenabholung mit Anhängen und die Auswertung der Antwort erfolgt in mehreren Schritten, die am nachfolgenden Beispiel „SonstigeNachrichten“ mit ericdemo veranschaulicht werden:

- Datenversand der Datenart SonstigeNachrichten mit defektem Anhang, siehe Kap. [6.1](#)
- XML-Anfrage erstellen, ob Daten abgeholt werden können, siehe Kap: [6.2](#)
- Abholung der bereitgestellten Daten, siehe Kap. [6.3](#)
- Dekodieren der Rückmeldung, siehe Kap. [6.4](#)
- Bestätigung der abgeholten Daten, siehe Kap: [6.5](#)

Bevor Sie mit dem nachfolgenden, praktischen Teil beginnen, machen Sie mit den Grundlagen der Datenabholung und den Anhängen vertraut:

- Siehe EHB, Kap. „Anhänge im Elster-XML“ mit Unterkap. „Rückmeldung zu den Anhängen“
- Siehe EHB, Kap. „Datenabholung“ mit Unterkap. „Abholung von Anhängen“

Verwenden Sie das Schema mit zugehöriger Schemadokumentation des Beispiels „SonstigeNachrichten“ und „MitteilungAbholung“ als Referenz zum Nachschlagen in den Verzeichnissen:

- Dokumentation\Schnittstellenbeschreibungen\Sonstige\SonstigeNachrichten_21\
- Dokumentation\Schnittstellenbeschreibungen\Sonstige\ElsterDatenabholung_18\

Für die Übermittlung der Beispieldaten mit dem Vorgang „sendAuth“ verwenden Sie, wenn möglich, Ihr eigenes Steueridentifikationsnummer (IdNr) Zertifikat. Sollte dies nicht möglich sein, kann das im Tutorial bereitgestellte Zertifikat *test-softidnr-pse.pfx* verwendet werden. Da dieses Testzertifikat aber von mehreren Personen gleichzeitig verwendet werden kann, sind Seiteneffekte, insbesondere bei der Datenabholung nicht auszuschließen. Nachfolgend wird als Platzhalter für das Zertifikat *<MeinIdNrZertifikat>* und zugehörige PIN *<MeinePIN>* verwendet.

6.1 Datenversand der Datenart SonstigeNachrichten mit defektem Anhang

Öffnen Sie die Beispieldaten ***SonstigeNachrichten_Rueckmeldungstest.xml***⁴⁷ aus dem Dokumentationspaket mit einem Texteditor. Neben den bekannten XML-Daten ist im Nutzdatenteil der base64codierte Anhang im Tag <Anhaenge> zu erkennen. Damit eine Antwort, eine sog. Rückübermittlung erfolgt, ist das Tag <RueckmeldungGewuenscht> auf **true** gesetzt. Damit Sie per E-Mail benachrichtigt werden, fügen Sie das Tag <EmailRueckmeldung> mit Ihrer E-Mailadresse ein und speichern das geänderte Beispieldaten-XML im ericdemo-Verzeichnis.

Achten Sie auf den Testmerker bei allen zu versendenden Testdaten. Der Testmerker muss auf den Wert „700000001“ gesetzt sein, nur dann sind alle Schritte der Datenabholung im Testmodus möglich.

Mit ericdemo sollen diese Beispieldaten authentifiziert versendet werden. Kopieren Sie Ihr Zertifikat in das ericdemo Verzeichnis, falls dies nicht möglich ist, verwenden Sie das in der ericdemo beigefügte Zertifikat *test-softidnr-pse.pfx*. Nachdem Sie ericdemo mit Ihrer Entwicklungsumgebung erstellt haben, kann der Datenversand auf der Kommandozeile erfolgen, beispielhaft der ericdemo Windows 64-Bit Aufruf⁴⁸:

```
startedemo-x64.bat /v SonstigeNachrichten_21 /x  
SonstigeNachrichten_Rueckmeldungstest.xml /c <MeinIdNrZertifikat> /p <MeinePIN>
```

Ergebnis: Der Datensatz wurde erfolgreich an den Annahmeserver übermittelt.

```
Sendestatus: Verarbeitung fehlerfrei.

Rückgabe:
<?xml version="1.0" encoding="UTF-8"?>
<EricBearbeiteVorgang xmlns="http://www.elster.de/EricXML/1.0/EricBearbeiteVorgang">
  <Erfolg/>
</EricBearbeiteVorgang>

Serverantwort:
<?xml version="1.0" encoding="UTF-8"?><Elster xmlns="http://www.elster.de/elsterxml/schema/v11"><TransferHeader version="11"><Verfahren>ElsterNachricht</Verfahren><DatenArt>SonstigeNachrichten</DatenArt><Vorgang>send-Auth</Vorgang><TransferTicket>et11700chmu6no31w9zo4ti1mj10gyzi</TransferTicket><Testmerker>700000001</Testmerker><Empfaenger id="L"><Ziel>HE</Ziel></Empfaenger><HerstellerID>74931</HerstellerID><DatenLieferant>MIAGCSqGSIB3DQEHbqCAMIACAQAwgAYJKoZIhvcNAQcBMBQGCCqGSIB3DQMHBAjpcNzXNcqITKCA
BCCIw8peMotytVdOfImS1r1vYT3Zders2rmXyEyK8emVaAAAAAAAAAAAAAAAA=
</DatenLieferant><EingangsDatum>20220427132719</EingangsDatum><Datei><Verschlüsselung>CMSEncryptedData</Verschlüsselung><Kompression>GZIP</Kompression><TransportSchlüssel>MIAGCSqGSIB3DQEHbqCAMIACAQAwgAYJKoZIhvcNAQcBMBQGCCqGSIB3DQMHBAiKvSaunRl6qCA
BBGgN/enPQOQZnJfa/VMZMHZMj0yTfpXH0AAAAAAAAAAAAAAAA
</TransportSchlüssel></Datei><RC><Rueckgabe><Code>0</Code><Text>Daten wurden erfolgreich angenommen.</Text>
</Rueckgabe><Stack><Code></Code><Text></Text></Stack></RC></TransferHeader><DatenTeil></DatenTeil></Elster>

Bitte drücken Sie die Eingabetaste
```

⁴⁷ Dokumentation\Schnittstellenbeschreibungen\Sonstige\SonstigeNachrichten_21\Beispiele\

⁴⁸ Der Aufruf erfolgt in einer Zeile, Zeilenumbrüche sind zu ignorieren.

Die Daten werden nun in den Finanzamtsservern verarbeitet. Die Rückmeldung erfolgt nicht sofort, sondern asynchron und steht möglicherweise erst am nächsten Tag zur Abholung bereit. Einen Hinweis auf die abzuholenden Daten erhalten Sie mit der gewünschten E-Mailbenachrichtigung:

Sehr geehrte Dame, sehr geehrter Herr,

Ihre am 27.04.2022, 13:27:19 Uhr an das Finanzamt übermittelten Anhänge konnten nicht vollständig verarbeitet werden.

Weitere Informationen entnehmen Sie bitte Ihrem Steuerprogramm/-portal.

Mit freundlichen Grüßen

Ihr Finanzamt

Sie entnehmen der E-Mail, dass eine vollständige Verarbeitung nicht möglich war. Weitere Informationen erhalten Sie in den bereitgestellten Daten, die nun in den nächsten Schritten abzuholen sind.

6.2 XML-Anfrage erstellen, ob Daten abgeholt werden können

Im EHB, Kap. „Informationen zur Datenart MitteilungAbholung (Datenabholung)“ erfahren Sie, dass mit der Datenart „MitteilungAbholung“ das Anfrage-XML zu erstellen ist.

Es sollen nur neue, noch nicht abgeholte Daten angefragt werden. Die „Nutzdaten“ sind wie folgt zu definieren:

```
<Nutzdaten>
  <Datenabholung version="18">
    <Anfrage einschraenkung="neue" />
  </Datenabholung>
</Nutzdaten>
```

Ihr erstelltes Anfrage-XML sollte der bereitgestellten Lösung⁴⁹ sehr ähnlich sein. Beachten Sie, dass im Rahmen der Datenabholung beim ersten Aufruf von *EricBearbeiteVorgang()* das Transferhandle mit 0 zu übergeben ist:

***startedemo-x64.bat /v MitteilungAbholung_18 /x MitteilungAbholungAnfrage.xml /c
<MeinIdNrZertifikat> /p <MeinePIN> /t 0***

⁴⁹ Dokumentation\Tutorial\Beispiele\MitteilungAbholungAnfrage.xml

Der Rückgabewert des Transferhandles wird für alle weiteren Aufrufe von *EricBearbeiteVorgang()* benötigt. Entnehmen Sie den Wert der Ausgabe auf der Kommandozeile (blau umrahmt). Nachfolgend wird statt Ihrem Wert der Platzhalter *<MeinTransferhandle>* verwendet.

```
Sendestatus: Verarbeitung fehlerfrei.

Transferhandle 977163844

Rückgabe:
<?xml version="1.0" encoding="UTF-8"?>
<EricBearbeiteVorgang xmlns="http://www.elster.de/EricXML/1.0/EricBearbeiteVorgang">
  <Erfolg/>
</EricBearbeiteVorgang>

Serverantwort:
<?xml version="1.0" encoding="UTF-8"?><Elster xmlns="http://www.elster.de/elsterxml/schema/v11"><TransferHeader version="11"><Verfahren>ElsterDatenabholung</Verfahren><DatenArt>MitteilungAbholung</DatenArt><Vorgang>send-Auth</Vorgang><TransferTicket>et11748ov6fib16gxui3dd6219i1kyt5</TransferTicket><Testmarker>700000001</Testmarker><Empfaenger id="L"><Ziel>CS</Ziel></Empfaenger><HerstellerID>74931</HerstellerID><DatenLieferant>MIAGCSqGSIB3DQEHbqCAMIACAQAwgAYJKoZIhvcNAQcBMBQGCCqGSIB3DQMHBAjMrpF0sD1T7KCA
BCAB3dn8r0fx99qqN1sxJRYTBfKlv1kXppFIz5diihZ3gAAAAAAAAAAAAA=
</DatenLieferant><EingangsDatum>20220427161211</EingangsDatum><Datei><Verschluesselung>CMSEncryptedData</Verschluesselung><Kompression>GZIP</Kompression><TransportSchluessel>MIAGCSqGSIB3DQEHbqCAMIACAQAwgAYJKoZIhvcNAQcBMBQGCCqGSIB3DQMHBAjMrpF0sD1T7KCA
BBGwpuZrdlhyKZprvDUyCftDq4nwYy8OuNUAAAAAAAAAAAAA
</TransportSchluessel></Datei><RC><Rueckgabe><Code>0</Code><Text>Ihr Request wurde bearbeitet</Text></Rueckgabe><Stack><Code>0</Code><Text>Ihr Request wurde bearbeitet</Text></Stack></RC><VersionClient>ERiC:
2007.1.0.0</VersionClient></TransferHeader><DatenTeil><Nutzdatenblock><NutzdatenHeader version="11"><Nut
zdatenTicket>1</NutzdatenTicket><Empfaenger id="L">CS</Empfaenger><Hersteller><ProduktName>Programm</Pro
duktName><ProduktVersion>V 1.0</ProduktVersion></Hersteller><DatenLieferant>Lieferant</DatenLieferant><R
C><Rueckgabe><Code>0</Code><Text>Ihre Anfrage wurde bearbeitet, es wurde(n) 1 Element(e) gefunden.</Text
></Rueckgabe><Stack><Code>0</Code><Text>Ihre Anfrage wurde bearbeitet, es wurde(n) 1 Element(e) gefunden
.</Text></Stack></RC></NutzdatenHeader><Nutzdaten><Datenabholung version="12"><Anfrage einschraenkung="n
eue"><Id groesse="1900" steuerart="AnhangRueckmeldung">be1177e509i3b541rebrp567rtrkxhojg</Id></Anfrage></
Datenabholung></Nutzdaten></Nutzdatenblock></DatenTeil></Elster>

Bitte drücken Sie die Eingabetaste
```

Der rot umrahmte Wert mit dem Tag *<ID>* ist die bereitgestellte und im folgenden abzuholende Nachricht. Der Wert *be1177e509i3b541rebrp567rtrkxhojg* muss durch Ihren Wert ersetzt werden, nachfolgend wird deshalb stattdessen *<MeineAbholungID>* als Platzhalter verwendet.

6.3 Abholung der bereitgestellten Daten

Aus der Serverantwort des vorangegangenen Abschnitts muss das neue XML zur Abholung der konkreten Rückmeldedaten, Datenart „MitteilungAbholung“, erstellt werden. Kopieren Sie das obige Anfrage-XML und ändern Sie es wie folgt:

```
<Nutzdaten>
  <Datenabholung xmlns="http://finkonsens.de/elster/elsterdatenabholung/v2"
    version="18">
    <Empfangsbestaetigung id="<MeineAbholungID>" />
  </Datenabholung>
</Nutzdaten>
```

Speichern Sie das geänderte Anfrage-XML unter dem neuen Namen `MitteilungAbholungmitID.xml` ab.

Übergeben Sie das zurückgelieferte Transferhandle `ericdemo` in der Kommandozeile und speichern Sie die Serverantwort in einer Datei:

```
startedemo-x64.bat /v MitteilungAbholung_18 /x MitteilungAbholungmitID.xml /c  
<MeinIdNrZertifikat> /p <MeinePIN> /t <MeinTransferhandle> /s Serverantwort.xml
```

Die Antwort XML-Daten liegen im Tag `<Datenpaket>` base64-kodiert vor. Sie werden diese im nächsten Schritt mit `ericdemo` dekodieren und somit in ein für Menschen lesbares Format umwandeln.

6.4 Dekodieren der Rückmeldung

Aus der *Serverantwort.xml* kopieren Sie nur die base64-kodierten Daten und speichern sie in einer neuen Datei „Base64-kodierteRueckmeldung.b64“. Diese Datei übergeben Sie im nächsten Schritt an ericdemo zum Dekodieren mit *EricDekodiereDaten()*:

startdemo-x64.bat /e /x Base64-kodierteRueckmeldung.b64

ericdemo gibt nun die dekodierten, lesbaren XML-Daten auf der Kommandozeile aus:

```
*** Entschluesslung der Daten aus der Datei "Base64-kodierteRueckmeldung.b64" ***

Entschluessele mit Zertifikat: test-softidnr-pse_3072.pfx

Sendestatus: Verarbeitung fehlerfrei.

Rückgabe:
<TransferTicket>et11700chmu6no31w9zo4ti1mj10gyzi</TransferTicket><NutzdatenTicket>1</NutzdatenTicket><Datenart>Son
stigeNachrichten</Datenart><Anrede>Sehr geehrte ELSTER-Nutzerin,
sehr geehrter ELSTER-Nutzer,</Anrede><Referenz>der Anhang zu Ihrem am 27.04.2022 um 13:27:19 Uhr unter der Transfe
rticket-Nummer et11700chmu6no31w9zo4ti1mj10gyzi abgegebenen Formular wurde maschinell überprüft.</Referenz><Anhaen
ge><Anhang><Dateibezeichnung>Defekt.pdf</Dateibezeichnung><Rueckmeldung>Eine Gefährdung der Finanzverwaltungs-Syst
eme durch den Anhang Defekt.pdf konnte nicht ausgeschlossen werden. Daher wurde der Anhang aus Sicherheitsgründen
gelöscht. Die Löschung ist im System der Finanzverwaltung dokumentiert und für die Beschäftigten des Finanzamts er
kennbar.</Rueckmeldung></Anhang></Anhaenge><Mittelteil>Dies hat keine Auswirkung auf den Inhalt des Formulars.</Mi
ttelteil><Anweisung>Bitte reichen Sie den entsprechenden Anhang in Papierform bei Ihrem Finanzamt nach, da er anso
nsten dem Finanzamt nicht bekannt wird. Geben Sie bitte an, dass der Anhang zum oben genannten Formular gehört.</A
nweisung><Grussformel>Mit freundlichen Grüßen

Ihre Finanzverwaltung</Grussformel>

Bitte drücken Sie die Eingabetaste
```

Sie entnehmen dem Text, dass der Anhang gelöscht wurde. Im letzten Schritt erfolgt die Bestätigung der abgeholten Daten.

6.5 Bestätigung der abgeholten Daten

Damit die bereitgestellten Daten nicht erneut abgeholt werden, muss die erfolgreiche Abholung quittiert werden. Hierzu kopieren Sie MitteilungAbholungAnfrage.xml nach MitteilungAbholungBestaetigung.xml und ändern den Nutzdatenteil wie folgt:

```
<Nutzdaten>
  <Datenabholung xmlns="http://finkonsens.de/elster/elsterdatenabholung/v2"
                  version="18">
    <Empfangsbestaetigung id="<MeineAbholungID>" />
  </Datenabholung>
</Nutzdaten>
```

Ersetzen Sie den Platzhalter *<MeineAbholungID>* durch die in Abschnitt 6.2 ermittelte „Abholung ID“ und speichern Sie die MitteilungAbholungBestaetigung.xml, um sie dann mit ericdemo zu versenden:

```
startedemo-x64.bat /v MitteilungAbholung_18 /x MitteilungAbholungBestaetigung.xml  
/c <MeinIdNrZertifikat> /p <MeinePIN> /t <MeinTransferhandle>
```

Sie entnehmen der Serverantwort, dass der Versand der Bestätigung erfolgreich war. Alle Schritte der Datenabholung sind somit erfolgreich ausgeführt worden.

7 Zusammenfassung

In diesem Tutorial lernten Sie mit dem Beispielprogramm „ericdemo“ und dem ESt-Beispiel ein Steuerprogramm zu erstellen, die Steuerdaten mit ERiC zu verarbeiten und die Änderungen, wie z.B. einen VZ-Wechsel oder eine Datenabholung, in Ihre Software zu integrieren.

Das ESt-Beispiel des Tutorials ist minimalistisch, vermittelt aber das Handwerkszeug, um selbständig alle weiteren Anlagen, die Sie in Ihrer Steuersoftware anbieten wollen, zu implementieren.

Für die Verarbeitung und Einbindung anderer Datenarten in Ihre Software können Sie ähnlich vorgehen. Das EHB unterstützt Sie hierbei und verweist auf weitere, notwendige Dokumente sowie das Herstellerforum im Internet.

Mit der API-Referenz erhalten Sie einen Überblick über die ERiC API-Funktionen und Datenstrukturen, die Sie in Ihre Steuersoftware integrieren können.

8 Tabellenverzeichnis

Tabelle 1-1	Typographische Konventionen	4
Tabelle 2-1	Dateien und Dokumente des ERiC-Tutorials	11
Tabelle 3-1	Beispieldaten für 2x „Spenden und Mitgliedsbeiträge“	31
Tabelle 4-1	Beispiel ESt 2020: Komplette versus größenoptimierter Installation mit ERiC 37.2.6	53

9 Abbildungsverzeichnis

Abbildung 2-1	Die Kernfunktionen des ERiC.....	14
Abbildung 2-2	Beispiel einer Funktionssignatur in der ERiC API-Referenz	18
Abbildung 3-1	Erstellung der ESt XML-Eingangsdaten	20
Abbildung 3-2	Ausschnitt aus 2020ESt1A011.png	21
Abbildung 3-3	Schema für ESt 2020	25
Abbildung 3-4	Ausschnitt aus der ESt Jahresdokumentation 2020	27
Abbildung 3-5	Datenverarbeitung mit ERiC.....	34
Abbildung 3-6	ERiC Fehlercodes in der API-Referenz	40
Abbildung 3-7	Ausschnitt aus Vordruck 2017AnlKind021.png	47
Abbildung 4-1	Anzahl ERiC Plugin-Bibliotheken und Basisbibliotheken der kompletten gegenüber der größenoptimierten ERiC Installation	54
Abbildung 5-1	Die Ausgaben der Fortschrittcallbacks in ericdemo	55