

Richtlinie BFS-RL 03-110
Standardbeschreibung Produktschnittstelle
Stahlbau - Teil 3: Implementierungsbereiche
und Konfirmitätsanforderungen

BFS-RL 03-110

**Standardbeschreibung Produktschnittstelle Stahlbau -
Teil 3: Implementierungsbereiche und
Konfirmitätsanforderungen**

**Empfehlungen des
Arbeitsausschusses Informationstechnologie**

Schnittstellenversion: April 2000
Stand des Dokumentes: 22.05.2000

Copyright-Klausel mit Haftungsausschluss

© Copyright - Klausel

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Die Autoren, der Verlag und der Hersteller können für fehlerhafte Angaben und deren Folgen keine Haftung übernehmen. Rechtsansprüche aus der Benutzung der vermittelten Daten sind daher ausgeschlossen. Für alle Hinweise und Verbesserungsvorschläge sind Herausgeber und Verlag stets dankbar. Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung von elektronischen Medien.

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, besonders die der Übersetzung, des Nachdrucks, der Bildentnahme, der Funksendung, der Wiedergabe auf photomechanischem oder ähnlichem Weg und der Nachspeicherung und Auswertung von Datenverarbeitungsunterlagen, bleiben auch bei Verwendung von Teilen des Werkes, der Verlag vorbehalten. Rechtsansprüche aus der Benutzung der vermittelten Daten sind ausgeschlossen. Bei gewerblichen Zwecken dienender Vervielfältigung ist an den Verlag gemäß § 54 UrhG eine Vergütung zu zahlen, deren Höhe mit dem Verlag zu vereinbaren ist.

Herausgeber:

bauforumstahl e. V., Düsseldorf

Vertrieb:

Stahlbau Verlags- und Service GmbH, Düsseldorf



STANDBESCHREIBUNG PRODUKTSCHNITTSTELLE STAHLBAU – TEIL 3:

IMPLEMENTIERUNGSBEREICHE UND KONFORMITÄTSANFORDERUNGEN

DSTV-Arbeitsausschuß EDV

Schnittstellenversion: April 2000
Stand dieses Dokumentes: 22.5.2000

Aufgestellt von:

E. Backx
M Falck
J. Friedrich
H.-W. Haller
C. Hörenbaum
M. Huhn
H.-D. Koch
H. Münninghoff
W. Rustler
C. Trinkner

1 Konformität

Im vorliegenden Teil 3 der Standardbeschreibung Produktschnittstelle Stahlbau werden Anforderungen an Implementierungen der Produktschnittstelle Stahlbau gestellt, die den Datenaustausch zwischen Software-Applikationen per Austauschdateien ermöglichen. Die dazu vorgenommene Unterteilung des Gesamtumfanges der Produktschnittstelle in sogenannte **Implementierungsbereiche** ermöglicht die in Teil 1 der Standardbeschreibung beschriebenen Anwendungsprozesse. Der Implementierungsbereich schreibt einer Implementierung vor, welche der in Teil 2 der Standardbeschreibung definierten Entities sie unterstützen muß, sofern dies nicht im folgenden Abschnitt 2 explizit eingeschränkt ist. Das Führen der Namen dieser Implementierungsbereiche durch Software-Applikationen soll die Konformität der Schnittstellenimplementierungen gewährleisten und dem Anwender deren genaue Anwendungsbereiche (und -grenzen) aufzeigen.

Eine Implementierung der Produktschnittstelle Stahlbau ist nur dann konform zur Standardbeschreibung Produktschnittstelle Stahlbau, wenn sie die in Abschnitt 2 gestellten Anforderungen erfüllt, und nur dann darf sie den Namen eines der in Abschnitt 4 definierten Implementierungsbereiche im Zusammenhang mit dem vom Konformitätsausschuß des DStV verliehenen Konformitätszeichen führen. Dies wird als Konformitätsaussage bezeichnet.

Um dieses Konformitätszeichen führen zu dürfen, muß eine Software seine Schnittstellenimplementierung einer Konformitätsprüfung unterziehen. Diese Konformitätsprüfung läßt der DStV von unabhängigen Dienstleistern unter Überwachung durch die Universität Karlsruhe (TH) durchführen. Außerdem kann der Konformitätsausschuß bei Unstimmigkeiten der als unabhängige Schiedsstelle angerufen werden. Ansprechpartner sind in beiden Fällen:

Herr Hüller, DStV: contact@deutscherstahlbau.de oder

Herr Hörenbaum, Universität Karlsruhe: hoerenbaum@versuchsanstalt.uni-karlsruhe.de

2 Anforderungen an eine Implementierung

Im folgenden wird zwischen den Begriffen *Einlesen*, *Interpretieren* und *Erzeugen* unterschieden. Mit *Einlesen* ist das Einlesen einer produktschnittstellenkonformen Austauschdatei (der Einfachheit halber lediglich Austauschdatei) gemeint, ohne daß die Applikation die Datensätze „verstehen“ muß. *Interpretieren* bedeutet den Vorgang des Auswertens der Informationen aus einer Menge (in der Regel einer Teilmenge) dieser Datensätze. Mit *Erzeugen* wird das Schreiben von Daten in eine Austauschdatei bezeichnet, die in der erzeugenden Applikation entstanden sind oder die diese zuvor durch Einlesen einer Austauschdatei erhalten hat.

2.1 Anforderungen an Implementierungen, die Austauschdateien einlesen

Eine Implementierung der Implementierungsbereiche „*Statische Struktur Einlesen*“, „*Statik Einlesen*“, „*Statik ohne Ergebnisse Einlesen*“, „*Entwurf Einlesen*“, „*NC3D Einlesen*“, „*Bearbeitungen Einlesen*“ oder „*Verbindungen Einlesen*“ muß alle der folgenden Anforderungen erfüllen:

- Die Implementierung muß *alle* der im Teil 2 der Standardbeschreibung definierten Entities einlesen können.
- Nach dem Einlesen der Austauschdatei muß die Applikation *diejenigen* der ihrem Implementierungsbereich in Abschnitt 4 zugeordneten Datensätze korrekt interpretieren können, die sie nach ihren eigenen Fähigkeiten bearbeiten kann. Die Definition der korrekten Interpretation ist in Abschnitt 5 gegeben.
- Teil 2 der Standardbeschreibung läßt zur Beschreibung eines Sachverhaltes unter Umständen mehrere Möglichkeiten zu. Die Implementierung muß bezüglich der im o.g. Punkt beschriebenen Datensätze jede dieser Möglichkeiten korrekt interpretieren können.

2.2 Anforderungen an Implementierungen, die Austauschdateien erzeugen

Eine Implementierung der Implementierungsbereiche „*Statische Struktur Schreiben*“, „*Statik Schreiben*“, „*Statik ohne Ergebnisse Schreiben*“, „*Entwurf Schreiben*“, „*NC3D Schreiben*“, „*Bearbeitungen Schreiben*“ oder „*Verbindungen Schreiben*“ muß alle der folgenden Anforderungen erfüllen:

- Von der Implementierung erzeugte Austauschdateien müssen bezüglich Syntax und Aufbau dem Teil 2 der Standardbeschreibung Produktschnittstelle Stahlbau genügen.
- Die Beschreibung von Objekten und Sachverhalten in einer von der Implementierung erzeugten Austauschdatei muß korrekt gemäß Abschnitt 5 sein.
- Sollen Objekte und Sachverhalte, die mit den in Teil 2 der Standardbeschreibung definierten Sprachmitteln beschrieben werden können, in eine Austauschdatei geschrieben werden, *muß* dies durch diese Sprachmittel geschehen.
- Kann die Implementierung lediglich einen Teil der in Abschnitt 4 für einen Implementierungsbereich geforderten Datensätze erzeugen, muß sie an den Namen des Implementierungsbereiches immer diejenigen Teile anhängen, die sie *nicht* erzeugen kann, siehe dazu das Beispiel in Abschnitt 4.
- Das in Tabelle 2 angegebene Kürzel des Implementierungsbereiches muß im Attribut IMPLEMENTATION_LEVEL des Datensatzes vom Entity-Typ FILE_DESCRIPTION im HEADER-Abschnitt der von der Implementierung erzeugten Austauschdatei angegeben sein.

Anmerkung: Die in Abschnitt 4 gemachten Zuordnungen der Entity-Gruppen zu Implementierungsbereichen beziehen sich lediglich auf den DATA-Abschnitt der Austauschdateien. Unabhängig davon muß *jede* Austauschdatei einen gültigen HEADER-Abschnitt gemäß Teil 2 der Standardbeschreibung enthalten.

2.3 Anforderungen an Implementierungen, die „round-trip-fähig“ sind

Eine Implementierung, die Austauschdateien sowohl einlesen als auch erzeugen kann, und die die unten aufgeführten Mindestanforderungen bezüglich der Pflege des von ihr nicht interpretierten Datenbestandes erfüllt, ist „round-trip-fähig“. Diese Fähigkeit beinhaltet logischerweise die Daten aller Implementierungsbereiche, nicht nur die der von ihr beim Einlesen oder Schreiben beanspruchten. Die round-trip-Fähigkeit wird mit der Unterstützung des Implementierungsbereiches „Round Trip“ ausgedrückt.

- Die Implementierung muß alle Anforderungen der Abschnitte 2.1 und 2.2 erfüllen.
- Erzeugt die Implementierung eine Austauschdatei aus Daten, die sie ganz oder teilweise aus vorherigem Einlesen einer Austauschdatei erhalten hat, müssen nach dem Erzeugen der Datei alle Datensätze, die die Applikation nicht interpretieren konnte, unverändert erhalten bleiben.
- Erzeugt die Implementierung eine Austauschdatei aus Daten, die sie ganz oder teilweise aus vorherigem Einlesen einer Austauschdatei erhalten hat, müssen alle Datensätze mit Objektidentität (siehe Teil 2 der Standardbeschreibung) aus der zuvor eingelesenen Datei erhalten bleiben. In diesen Datensätzen darf das Attribut ID dabei nicht verändert werden. Alle weiteren Attribute dürfen verändert werden.
- Erzeugt die Implementierung eine Austauschdatei aus Daten, die sie ganz oder teilweise aus vorherigem Einlesen einer Austauschdatei erhalten hat, müssen Verweise von Datensätzen mit Objektidentität auf andere Datensätze mit Objektidentität, die die Applikation nicht interpretieren konnte, erhalten bleiben.

3 Gruppierung der Entities

Entitygruppe	Zugehörige Entities gemäß Teil 2
Allgemeine Daten	OWNER BUILDING_SITE ARCHITECT SPECIFICATION VERTEX POLYLINE VERTEX_LIST POLYLINE_LIST RASTER_1 RASTER_2 RASTER_3 RASTER_LINE STRUCTURE CUSTOM_ENTITY MATERIAL CROSS_SECTION COMPOSITE_CROSS_SECTION
Statikdaten I	SUBSTRUCTURE NODE ELEMENT
Statikdaten II	CROSS_SECTION_ELEMENT CSE_LIST BOUNDARY_CONDITION ELEMENT_NODE_CONNECTIVITY ELEMENT_ECCENTRICITY ELASTIC_SUPPORT ACTION ACTION_COMBINATION_RULE ACTION_COMBINATION NODAL_ACTION ELEMENT_ACTION
Statikdaten III	NODAL_REACTION ELEMENT_REACTION
Entwurfsdaten	MEMBER MEMBER_LOCATION
Konstruktionsdaten	CUTOUT HOLE SIGNATURE COLD_BEND
Beschichtungsdaten	SURFACE
Verbindungsdaten	CONNECTION CONNECTION_ELEMENT_LOCATION BOLT BOLT_ELEMENT WELD GLUE

Tabelle 1: Entity-Gruppen

4 Implementierungsbereiche mit zugehörigen Entity-Gruppen

Tabelle 2 ordnet den **Implementierungsbereichen** die Entity-Gruppen aus Abschnitt 3 zu.

Implementierungsbereich	Entity-Gruppen	Kürzel
„ <i>Statische Struktur Einlesen</i> “	Allgemeine Daten Statikdaten I	1(IN)
„ <i>Statische Struktur Schreiben</i> “	Allgemeine Daten Statikdaten I	1(OUT)
„ <i>Statik ohne Ergebnisse Einlesen</i> “	Allgemeine Daten Statikdaten I Statikdaten II	7(IN)
„ <i>Statik ohne Ergebnisse Schreiben</i> “	Allgemeine Daten Statikdaten I Statikdaten II	7(OUT)
„ <i>Statik Einlesen</i> “	Allgemeine Daten Statikdaten I Statikdaten II Statikdaten III	2(IN)
„ <i>Statik Schreiben</i> “	Allgemeine Daten Statikdaten I Statikdaten II Statikdaten III	2(OUT)
„ <i>Entwurf Einlesen</i> “	Allgemeine Daten Entwurfsdaten	3(IN)
„ <i>Entwurf Schreiben</i> “	Allgemeine Daten Entwurfsdaten	3(OUT)
„ <i>NC3D Einlesen</i> “	Allgemeine Daten Entwurfsdaten Konstruktionsdaten	4(IN)
„ <i>NC3D Schreiben</i> “	Allgemeine Daten Entwurfsdaten Konstruktionsdaten	4(OUT)
„ <i>Bearbeitungen Einlesen</i> “	Allgemeine Daten Entwurfsdaten Konstruktionsdaten Beschichtungsdaten	5(IN)
„ <i>Bearbeitungen Schreiben</i> “	Allgemeine Daten Entwurfsdaten Konstruktionsdaten Beschichtungsdaten	5(OUT)
„ <i>Verbindungen Einlesen</i> “	Verbindungsdaten	6(IN)
„ <i>Verbindungen Schreiben</i> “	Verbindungsdaten	6(OUT)
„ <i>Round Trip</i> “	Alle	8

Tabelle 2: Implementierungsbereiche mit zugeordneten Entity-Gruppen

Erfüllt eine Implementierung die Anforderungen aus mehreren Implementierungsbereichen, werden im HEADER die Kürzel in der in Tabelle 2 gegebenen Reihenfolge durch „/“ getrennt aneinander gehängt.

Kann eine Implementierung, wie in Abschnitt 2.1 beschrieben, nicht alle Entities eines Implementierungsbereiches erzeugen, müssen die nicht unterstützten Daten in Klammern an den Namen des Implementierungsbereiches angehängt werden.

Beispiel: Ein Statikprogramm soll beliebige statische Systeme erzeugen und berechnen können, ausgenommen elastische Bettungen. Es kann diese Systeme mit Ergebnissen aus statischen Berechnungen in Austauschdateien schreiben und entsprechende Dateien einlesen. Außerdem kann es aus einem statischen System einen Entwurf generieren, den es in eine Austauschdatei schreiben kann, dieses aber nicht für blechförmige Bauteile. Der im HEADER-Abschnitt der von der Implementierung erzeugten Austauschdatei vorkommende Datensatz vom Entity-Typ FILE_DESCRIPTION sieht dann folgendermaßen aus:

```
...
FILE_DESCRIPTION('Beispiel fuer einen Implementierungsbereich'),
                '2 (IN) /2 (OUT) /3 (OUT) ');
...
```

Der Programmhersteller darf die Implementierung folgendermaßen nennen (Konformitätsaussage):

Implementierung der Produktschnittstelle Stahlbau, Version 4/2000 in den Implementierungsbereichen „*Statik Einlesen, Statik Schreiben (ohne elastische Bettung), Entwurf Schreiben (ohne Bleche)*“

5 Korrekte Interpretation von Datensätzen und korrekte Beschreibung von Objekten und Sachverhalten

Eine Interpretation von zuvor eingelesenen Datensätze aus einer Austauschdatei durch eine Software-Applikation gilt dann als korrekt,

- wenn alle diejenigen Objekte und Sachverhalte, die diese Datensätze enthalten, der Applikation zur Bearbeitung zur Verfügung stehen,
- oder wenn eine Implementierung des Implementierungsbereiches „*Statische Struktur Einlesen*“ aus eingelesenen Datensätzen der Entity-Gruppen *Allgemeine Daten* und *Statikdaten I* in der Applikation eine Struktur erzeugt, deren Schwerelinien mit denen der zu interpretierenden Datensätze übereinstimmt, und die keine ungewollten Lücken, Überschneidungen oder Überstände aufweist. Dies kann interaktiv an der Programmoberfläche geschehen.

Die Beschreibung von Objekten und Sachverhalten, die in einer Applikation erzeugt wurden, gilt in einer Austauschdatei als korrekt,

- wenn die Beschreibung durch Datensätze gemäß Teil 2 der Standardbeschreibung Produktschnittstelle Stahlbau erfolgt ist, so daß die Objekte und Sachverhalte in Applikation und Austauschdatei „identisch“ sind.
- oder wenn eine Implementierung des Implementierungsbereiches „*Statische Struktur Schreiben*“ aus einer Konstruktion oder einem Entwurf Datensätze der Entity-Gruppen *Allgemeine Daten* und *Statikdaten I* gemäß Teil 2 der Standardbeschreibung Produktschnittstelle Stahlbau erzeugt, durch die eine Struktur beschrieben wird, deren Schwerelinien mit denen der Konstruktion oder des Entwurfes übereinstimmt, und die keine ungewollten Lücken, Überschneidungen oder Überstände aufweist. Dies kann interaktiv an der Programmoberfläche geschehen.