



Objektkatalog für das Straßen- und Verkehrswesen

Basiskonzepte im OKSTRA

Version: 1.11

Datum: 20.06.2024

Status: fertig

Dateiname: N0135.docx

Pfad: n.a.

Verantwortlich: Jochen Hettwer

OKSTRA®-Pflegestelle

interactive instruments GmbH
Bundeskanzlerplatz 2d
53113 Bonn

Herr Jochen Hettwer
Tel. 0228 91410 89
Email hettwer@interactive-instruments.de

Im Auftrag von

Bundesanstalt für Straßenwesen
Referat V2
Brüderstraße 53
51427 Bergisch Gladbach

Herr Gerd Kellermann
Tel. 02204 43 4201
Email kellermann@bast.de



1 UML-Profil für den OKSTRA	3
1.1 UML-Diagrammtypen	3
1.2 Basismodell	3
1.3 Arten von Klassen	3
1.3.1 Objektarten	4
1.3.2 Objektarten mit symbolischer Referenzierungsmöglichkeit	4
1.3.3 Elementare Datentypen	4
1.3.4 Komplexe Datentypen	5
1.3.5 Unionen	5
1.3.6 Schlüssel Tabellen	5
1.4 Beziehungen zwischen Klassen	6
1.4.1 Vererbungsbeziehungen	6
1.4.2 Assoziationen	6
1.5 Paketstruktur	7
1.6 Namen	7
1.7 Tagged Values	8
1.7.1 Einschränkungen auf Eigenschaften	8
2 Historisierung	9
3 Zulässige Koordinatenreferenzsysteme	10
3.1 Koordinatenreferenzsysteme für 2-D-Lageangaben	11
3.2 Koordinatenreferenzsysteme für 3-D-Positionsangaben	15
3.3 Koordinatenreferenzsysteme für Höhenangaben	15
4 Festlegungen für den Datenaustausch	16



1 UML-Profil für den OKSTRA

Das OKSTRA-Referenzmodell wird mit Hilfe der Modellierungssprache UML beschrieben. Die dabei einzuhaltenden Regeln sind in den folgenden Unterabschnitten aufgeführt.

1.1 UML-Diagrammtypen

Es werden ausschließlich UML-Klassendiagramme verwendet.

1.2 Basismodell

Als Basismodell werden folgende Pakete des „Iso harmonized model“ verwendet, das im Internet unter <https://www.seegrid.csiro.au/mirrors/iso-harmonized-model/> verfügbar ist:

- ISO 00639 Language Codes
- ISO 03166 Country Codes
- ISO 19103:2005 Schema Language
- ISO 19107:2003 Spatial Schema
- ISO 19108:2006 Temporal Schema
- ISO 19109 Application Schema
- ISO 19111 Referencing by Coordinates
- ISO 19115:2006 Metadata (Corrigendum)
- ISO 19136 GML
- ISO 19139 Metadata – XML Implementation
- ISO 19148 Linear Referencing
- ISO 19156:2011 Observations and Measurements

1.3 Arten von Klassen

Folgende Arten von Klassen können im UML-Modell des OKSTRA auftreten:

1. Objektarten,
2. Objektarten mit symbolischer Referenzierungsmöglichkeit,
3. elementare Datentypen,
4. komplexe Datentypen,
5. Unionen,
6. Schlüsseltabellen.



1.3.1 Objektarten

Eine Objektart umfasst logisch zusammengehörende Informationen und dient häufig (aber nicht in jedem Fall) zur Abbildung von Objekten der realen Welt. Objektarten besitzen eine eigenständige Objektidentität, d. h. die Existenz einer Instanz einer Objektart ist nicht an die Existenz anderer Instanzen gebunden. Objektarten können abstrakt oder instanzierbar sein. Von einer instanzierbaren Objektart können Instanzen erzeugt werden, die dann z. B. in Dateien für den Datenaustausch auftreten können. Von einer abstrakten Objektart können keine Instanzen erzeugt werden. Abstrakte Objektarten dienen daher hauptsächlich dazu, zusammengehörige Informationen zu bündeln und geschlossen weiterzuvererben.

Objektarten erben stets direkt oder mittelbar von der Objektart *OKSTRA_Objekt*. Diese Objektart enthält u. a. das Attribut „OKSTRA_ID“ zur optionalen Angabe eines Objektidentifikators in Form einer GUID.

Objektarten erhalten das Stereotype „**FeatureType**“.

1.3.2 Objektarten mit symbolischer Referenzierungsmöglichkeit

Objektarten mit symbolischer Referenzierungsmöglichkeit sind Objektarten, die in Assoziationen auch durch die Angabe eines symbolischen – in der Regel „fachlichen“ – Identifikators referenziert werden können. Abgesehen von dieser Besonderheit verhalten sie sich wie normale Objektarten.

Objektarten mit symbolischer Referenzierungsmöglichkeit erhalten das Stereotype „**FachId**“.

1.3.3 Elementare Datentypen

Elementare Datentypen sind Typangaben für eine einzelne Werte (eine Zahl, ein String etc.) und können als Typspezifikatoren in Attributen von Objektarten oder komplexen Datentypen verwendet werden.

Im OKSTRA-UML-Modell werden die folgenden elementaren Datentypen des Basispakets „ISO 19103:2005 Schema Language“ bzw. aus diesen per Vererbung abgeleitete Datentypen verwendet:

- *ClockTime*,
- *Date*,
- *Sequence<Bit>*,
- *Integer*,
- *Real*,
- *CharacterString*,
- *Boolean*.

Elementare Datentypen zur Repräsentation von Werten, die in irgendeiner Weise gemessen werden können, werden aus dem Typ *Measure* bzw. einer seiner Spezialisierungen (*Angle*, *Area*, *Length* etc.) abgeleitet. Es ist möglich, analog zu den bereits im Basispaket bestehenden Spezialisierungen des Typs *Measure* neue Spezialisierungen zur Repräsentierung einer bestimmten Art von Messwerten zu schaffen (so könnte man z. B. zur Angabe von Temperatur-Messwerten einen Typ *Temperature* aus dem Typ *Measure* ableiten. Sofern noch Typen für spezielle Temperatur-Maßeinheiten benötigt werden, könnte man dann aus dem Typ *Temperature* weitere Typen ableiten, z. B. *Grad_Celsius*, *Grad_Fahrenheit* oder *Kelvin*). Sofern ein Datentyp aus dem Typ *Measure* oder einer seiner Spezialisierungen abgeleitet wird, ist er mit dem Tagged Value *uomString* zu versehen, in dem eine Abkürzung für die verwendete Maßeinheit anzugeben ist (z. B. „m“ für den Typ *Meter*).



In die Kategorie der elementaren Datentypen fallen auch die folgenden Datentypen des Basispakets „ISO 19107:2003 Spatial Schema“ zur Darstellung von Geometrien.

Einfache Geometrie:

- Punktgeometrie: *GM_Point*,
- Liniengeometrie: *GM_Curve*,
- Flächengeometrie: *GM_Surface*,
- Volumengeometrie: *GM_Solid*.

Mehrfach-Geometrien:

- Punktgeometrien: *GM_MultiPoint*,
- Liniengeometrien: *GM_MultiCurve*,
- Flächengeometrien: *GM_MultiSurface*,
- Volumengeometrien: *GM_MultiSolid*.

Neue elementare Datentypen, die aus den Typen der Basispakete per Vererbung abgeleitet werden, werden mit dem Stereotype „**type**“ gekennzeichnet.

1.3.4 Komplexe Datentypen

Komplexe Datentypen können ebenso wie die elementaren Datentypen als Typspezifikatoren von Attributen in Objektarten oder anderen komplexen Datentypen verwendet werden. Im Unterschied zu den elementaren Datentypen können sie jedoch mehrere Informationen enthalten. Ein komplexer Datentyp kann folgende Arten von Eigenschaften besitzen:

- Attribute elementaren Typs,
- Attribute komplexen Typs,
- Attribute des Union-Typs,
- Assoziationen.

Da ein komplexer Datentyp im Gegensatz zu einer Instanz einer Objektart keine eigenständige Identität besitzt, kann eine von ihm ausgehende Assoziation nur in der Richtung von ihm weg navigiert werden (d. h. er verweist auf eine Objektart, diese aber nicht auf ihn zurück). Infolgedessen wird auch nur das navigierbare Ende mit einem Rollennamen und einer Kardinalitätsangabe versehen.

Komplexe Datentypen sind mit dem Stereotype „**DataType**“ zu versehen.


1.3.5 Unionen

Eine Union ist ein Datentyp, der ähnlich wie ein komplexer Datentyp mehrere Attribute elementaren bzw. komplexen Typs besitzen kann. Diese Attribute müssen optional und eindeutig sein; von ihnen darf in konkreten Daten maximal eines belegt werden. Eine Union kann keine Assoziationen besitzen.

Unionen sind mit dem Stereotype „**union**“ zu versehen.

1.3.6 Schlüsseltabellen

Eine Schlüsseltabelle ist ein Datentyp, der die Angabe eines Eintrags aus einem (ggf. bereits im UML-Modell definierten) Wertekatalog ermöglicht. Eine Schlüsseltabelle besitzt zwei oder mehr

	Objektkatalog für das Straßen- und Verkehrswesen Basiskonzepte im OKSTRA	Seite: 6 von 16 Name: N0135 Stand: 20.06.2024
--	---	--

Attribute vom Datentyp *CharacterString*. Das erste dieser Attribute muss den Rollennamen „Kennung“ besitzen und dient zur Abbildung der eindeutigen Kennung des jeweiligen Schlüsselzeileintrags. In den weiteren Attributen werden die der Kennung zugeordneten Werte angegeben (in den meisten Fällen ist der Kennung nur ein „Langtext“ zugeordnet).

Sofern für eine Schlüsselzeile ein Wertekatalog im UML-Modell definiert werden soll, ist er als Constraint mit dem Namen „Werteliste“ anzugeben.

Sollen in einer Schlüsselzeile zusätzlich zu einem ggf. im UML-Modell definierten Wertekatalog weitere Einträge verwendet werden dürfen, ist sie mit dem Tagged Value *extendable* zu kennzeichnen.

Schlüsselzeilen sind mit dem Stereotyp „**Schlüsselzeile**“ zu versehen.

1.4 Beziehungen zwischen Klassen

Zwischen den Klassen des Modells dürfen nur zwei Arten von Beziehungen existieren:

- Vererbungsbeziehungen (Generalisierungen) und
- Assoziationen.

Weitere in UML verfügbare Arten von Beziehungen wie z. B. Kompositionen oder Aggregationen sind nicht zulässig.

1.4.1 Vererbungsbeziehungen

Bei einer Vererbungsbeziehung zwischen zwei Klassen übernimmt die erbende Klasse alle Eigenschaften (Attribute und Assoziationen) von der vererbenden Klasse.

Im OKSTRA-UML-Modell ist nicht nur das Erben von einer anderen Klasse, sondern auch das Erben von mehreren anderen Klassen möglich (Mehrfachvererbung).

Im Hinblick auf die verschiedenen Arten von Klassen gelten für den Einsatz von Vererbungsbeziehungen folgende Regeln:

- Eine Objektart (Stereotyp „FeatureType“) darf nur von anderen Objektarten mit dem Stereotyp „FeatureType“ erben.
- Eine Objektart mit symbolischer Referenzierungsmöglichkeit (Stereotyp „FachId“) darf nur von anderen Objektarten (Stereotypes „FeatureType“ oder „FachId“) erben.
- Ein elementarer Datentyp (Stereotyp „type“) darf nur von anderen elementaren Datentypen erben (diese tragen dann ebenfalls den Stereotyp „type“ bzw. im Falle einiger aus dem ISO-Basismodell stammender Datentypen keinen Stereotyp).
- Ein komplexer Datentyp (Stereotyp „DataType“) darf nur von anderen komplexen Datentypen erben.
- Eine Union (Stereotyp "union") darf keine Vererbungsbeziehungen besitzen.
- Eine Schlüsselzeile (Stereotyp „Schlüsselzeile“) darf keine Vererbungsbeziehungen besitzen.

1.4.2 Assoziationen

Assoziationen können nur bestehen



- zwischen zwei Objektarten (unabhängig davon, ob diese symbolisch referenzierbar sind oder nicht),
- zwischen einem komplexen Datentyp und einer Objektart (unabhängig davon, ob diese symbolisch referenzierbar ist oder nicht).

Assoziationen zwischen zwei Objektarten sind stets beidseitig navigierbar. In diesem Fall sind beide Enden mit Kardinalitätsangaben und Rollennamen zu versehen.

Assoziationen zwischen einem komplexen Datentyp und einer Objektart sind nur in der Richtung vom komplexen Datentyp zur Objektart navigierbar. Daher ist in diesem Fall nur das vom Datentypen wegzeigende Ende mit einer Kardinalitätsangabe und einem Rollennamen zu versehen.

Im Gegensatz zu den Assoziationsenden werden die vollständigen Assoziationen nicht mit Namen versehen.

Ein Rollenname eines Assoziationsendes gehört logisch zur *gegenüberliegenden* Klasse. In Bezug auf diese Klasse muss der Name eindeutig sein (in der Menge der Assoziations- und Attributnamen, wobei auch geerbte Attribute und Assoziationen zu berücksichtigen sind).

Zeigt eine Assoziation auf eine Objektart mit symbolischer Referenzierungsmöglichkeit, dann kann die Angabe der Assoziation in dieser Richtung optional durch den symbolischen Identifikator des Zielobjekts erfolgen.

1.5 Paketstruktur

Die ISO-Pakete des Basismodells werden direkt unter dem „Model“ in einem Paket namens „ISO/TC211“ abgelegt.

Die OKSTRA-Pakete werden parallel dazu in einem Paket namens „OKSTRA x.xxx“ bzw. „OKSTRA x.xxx.y“ abgelegt, wobei für x.xxx die Hauptversionsnummer und für y die (optionale) Entwicklungsversionsnummer der jeweiligen OKSTRA-Version einzusetzen ist. Dieses Paket – das im Folgenden als „OKSTRA Root Package“ bezeichnet wird – ist mit dem Stereotype „**Application Schema**“ zu versehen.

Innerhalb des OKSTRA Root Package besteht für jedes OKSTRA-Fachschemata ein eigenes Unterpaket, dessen Name dem Schemanamen entspricht. In diesen Unterpaketen sind die jeweiligen Objektarten des Fachschemas sowie die zugehörigen Klassen-Diagramme untergebracht.

Parallel zu den Unterpaketen für die OKSTRA-Fachschemata existiert das Unterpaket „Datentypen“, in dem die OKSTRA-eigenen Datentypen (sowohl die elementaren als auch die komplexen) mit den zugehörigen Diagrammen angesiedelt sind.


Ebenfalls parallel zu den Unterpaketen für die OKSTRA-Fachschemata existiert das Unterpaket „Schlüsseltabellen“, in dem sich die Schlüsseltabellen befinden.

1.6 Namen

Konventionen für alle Namen:

- Zulässige Zeichen: A-Z, a-z, 0-9, -, _
- Namensbestandteile werden durch - oder _ getrennt (möglichst kein CamelCase)

Zusätzliche Konventionen für die Namen von Paketen zur Aufnahme von Fachschemaschemata:

	Objektkatalog für das Straßen- und Verkehrswesen Basiskonzepte im OKSTRA	Seite: 8 von 16 Name: N0135 Stand: 20.06.2024
--	---	--

- Ein solcher Name beginnt stets mit „S_“

Zusätzliche Konventionen für die Namen von Klassen und Datentypen:

- Ein solcher Name beginnt nie mit „S_“

Zusätzliche Konventionen für die Rollennamen von Attributen und Assoziationen:

- Ein solcher Name beginnt nie mit „S_“

1.7 Tagged Values

Das OKSTRA Root Package besitzt folgende Tagged Values:

- *previous*: Nummer der OKSTRA-Vorgängerversion (Beispiel: „1.015“)
- *targetNamespace*: Namespace für die GML-Erzeugung. Der Wert dieses Tagged Values für die Version 2.016 beträgt beispielsweise „http://schema.okstra.de/2016/okstra“.
- *version*: Nummer der OKSTRA-Version (Beispiel: „2.016“)
- *xmlns*: Namespace-Präfix für die GML-Erzeugung (Wert: „okstra“).
- *xsdDocument*: Name der GML-Schemadatei für die GML-Erzeugung (Wert: „okstra.xsd“).

Die Unterpakete für die OKSTRA-Fachschemata sowie die Unterpakete „Datentypen“ und „Schlüsseltabellen“ besitzt jeweils den Tagged Value *xsdDocument*, in dem der Name der GML-Schemadatei angegeben wird, in die der Inhalt des jeweiligen Unterpaketes bei der GML-Erzeugung abgebildet wird (Wert: „Unterpaketname“ + „.xsd“).

1.7.1 Einschränkungen auf Eigenschaften

Einschränkungen hinsichtlich der Länge von Strings können mit den folgenden Tagged Values bei Datentypen, die aus dem Typ *CharacterString* abgeleitet werden, sowie bei Attributen, die diese Typen (oder direkt den Typ *CharacterString*) verwenden, vorgenommen werden:

- *fixed*: Der String muss genau die angegebene Länge besitzen.
- *minLength*: Der String darf minimal die angegebene Länge besitzen.
- *maxLength*: Der String darf maximal die angegebene Länge besitzen. (Gleichbedeutend mit *variable*, kompatibel zum XML-Schema-Facetten-Namen)
- *pattern*: Regulärer Ausdruck nach der ECMAScript-Syntax (wie JavaScript). Siehe <http://www.cplusplus.com/reference/regex/>
- *variable*: Der String darf maximal die angegebene Länge besitzen, die minimale Länge ist 0. (obsolet, wird zur Rückwärtskompatibilität beibehalten, zukünftig soll *maxLength* verwendet werden)

Einschränkungen bezüglich numerischer Werte können mit den folgenden Tagged Values bei den Datentypen *Integer*, *Real* und *Measure* abgeleiteten Typen sowie bei Attributen, die diese Typen (oder direkt die Typen *Integer* oder *Real*) verwenden, vorgenommen werden:

- *fractionDigits*: Genauigkeit als Zahl der zu berücksichtigenden Nachkommastellen (nicht beim Datentyp *Integer* oder Ableitungen daraus)
- *maxExclusive*: obere Grenze für Zahlenwerte, Grenze gehört nicht zum Wertebereich



- *maxInclusive*: obere Grenze für Zahlenwerte, Grenze gehört zum Wertebereich
- *minExclusive*: untere Grenze für Zahlenwerte, Grenze gehört nicht zum Wertebereich
- *minInclusive*: untere Grenze für Zahlenwerte, Grenze gehört zum Wertebereich

Datentypen, die aus dem Typ *Measure* oder einer seiner Spezialisierungen abgeleitet werden, besitzen darüber hinaus folgenden Tagged Value:

- *uomString*: Abkürzung für die verwendete Maßeinheit (z. B. „m“ für die Einheit „Meter“).

Schlüsseltabellen, in denen die Verwendung von nicht im Modell definierten Einträgen gestattet ist, besitzen den Tagged Value *extendable* (ohne Wertangabe).

Wenn eine Objektart ein Textattribut besitzt, das für die Aufnahme von Fachbedeutungen aus den Fachbedeutungslisten vorgesehen ist, ist sowohl die Objektart als auch das entsprechende Attribut mit dem Tagged Value *hatFachbedeutung* (ohne Wertangabe) zu kennzeichnen.

2 Historisierung

Das OKSTRA-UML-Modell enthält ein Historisierungsmodell auf Objektebene. Von der Instanz einer historisierbaren (und instanzierbaren) Objektart können dabei mehrere Versionen existieren, die unterschiedliche, zeitlich aufeinander folgende Gültigkeitsintervalle besitzen. Zwischen der Entstehung und dem Untergang der Instanz ist somit zu einem Zeitpunkt genau eine Version gültig. Die einzelnen Versionen sind darüber hinaus über Vorgänger- und Nachfolger-Assoziationen miteinander verknüpft.

Historisierbare Objektarten erben im Modell direkt oder mittelbar von der Objektart *historisches_Objekt*. Diese Objektart enthält sowohl Attribute zur Angabe des Gültigkeitsintervalls als auch die oben erwähnte Vorgänger-Nachfolger-Assoziation.

Wenn eine OKSTRA_ID für eine Instanz einer historisierenden Objektart vergeben wird, ist sie für alle Versionen der Instanz gültig. Mit der OKSTRA_ID allein kann somit nur die Menge der bestehenden Versionen identifiziert werden, nicht aber eine einzelne Version. Die Identifikation einer einzelnen Version erfolgt über eine Kombination aus der OKSTRA_ID und dem jeweiligen Gültigkeitsintervall.

Auf der Ebene der Instanzen sind Assoziationen zu historisierenden Objektarten stets multipel, da sie sich grundsätzlich auf alle Versionen der zu referenzierenden Instanz(en) beziehen. Welche Version(en) zu einem bestimmten Zeitpunkt als Assoziationspartner relevant sind, muss anhand der Gültigkeitsintervalle geklärt werden. Dies gilt auch bei der Referenzierung in Form von symbolischen Verweisen.

Im UML-Modell werden die Kardinalitäten von Assoziationen hingegen stets konzeptionell und ohne Berücksichtigung der Historisierung angegeben. Dies bedeutet, dass eine fachlich eindeutige Assoziation im UML-Modell auch als solche angegeben wird – ohne Berücksichtigung der Tatsache, dass in technischen Ableitungen aus dem Modell (z. B. Formatbeschreibungen) eindeutige Assoziationen zu historisierbaren Objektarten ggf. multipel werden müssen, um mehrere Versionen des historisierbaren Objekts referenzieren zu können.

Wenn eine technische Ableitung aus dem Modell multiple Assoziationen zu historisierbaren Objektarten benötigt, müssen diese im Rahmen des Ableitungsprozesses erzeugt werden.

Bei einer Assoziation zwischen zwei historisierenden Objektarten besteht auf der Instanzebene alternativ zu einer Verknüpfung mit allen Versionen der zu referenzierenden Instanz(en) auch die Möglichkeit, nur diejenigen Versionen untereinander zu verknüpfen, deren Gültigkeitsintervalle sich überlappen. Damit kann die Bestimmung der zu einem bestimmten Zeitpunkt relevanten



Assoziationspartner effizienter durchgeführt werden. Bei symbolischen Verweisen und bei Assoziationen zwischen historisierbaren und nicht historisierbaren Objektarten kann dieses Verfahren jedoch nicht angewendet werden.

3 Zulässige Koordinatenreferenzsysteme

Im OKSTRA dürfen ausschließlich die in der folgenden Tabelle aufgeführten Koordinatenreferenzsysteme verwendet werden. Im Datenformat OKSTRA-XML sind zur Angabe von Koordinatenreferenzsystemen die in der Spalte „AdV-Code“ enthaltenen Codes zu verwenden, die der „Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens“ (GeoInfoDok) der „Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland“ (AdV) entnommen sind. Gemäß den Vorgaben der GeoInfoDok sind im OKSTRA nur Koordinatenwerte ohne Streifen- und Zonenangaben zulässig.

Sofern es zu einem Koordinatenreferenzsystem länderspezifische Ausprägungen gibt, ist eine textliche Beschreibung nur in der ersten Tabellenzeile aufgeführt, in der das System erscheint. In den folgenden Tabellenzeilen finden sich dann die Kürzel der jeweiligen Bundesländer bzw. des Bundes (DE = Deutschland, BW = Baden-Württemberg, BY = Bayern, BE = Berlin, HB = Hansestadt Bremen, HH = Hansestadt Hamburg, HE = Hessen, NI = Niedersachsen, NW = Nordrhein-Westfalen, RP = Rheinland-Pfalz, ST = Sachsen-Anhalt, SH = Schleswig-Holstein, SL = Saarland, TH = Thüringen).

Im OKSTRA-XML-Datenformat können zur Angabe von Koordinatenreferenzsystemen (im attribute „srsName“) alle Einträge der drei Schlüsseltabellen Koordinatenreferenzsystem_2D, Koordinatenreferenzsystem_3D und Koordinatenreferenzsystem_Höhe verwendet werden. Die konkrete Angabe erfolgt über den Langtext des jeweiligen Eintrags (= Kurzbezeichnung des Systems in der GeoInfoDok). Um eine Kombination eines 2D- Koordinatenreferenzsystems mit einem Höhenreferenzsystem anzugeben, werden analog zur GeoInfoDok die Kurzbezeichnungen der beiden Systeme mit einem Stern (*) als Trennzeichen hintereinander aufgeführt. In beiden Fällen ist noch der URN-Namespace „urn:adv:crs“ davorzusetzen.

Beispiel 1: Die Angabe des 2D-Koordinatenreferenzsystems DE_DHDN_3GK2 lautet

urn:adv:crs:DE_DHDN_3GK2

Beispiel 2: Die Kombination des 2D-Koordinatenreferenzsystems DE_DHDN_3GK2 und des Höhenreferenzsystems DE_DHHN92_NH lautet

urn:adv:crs:DE_DHDN_3GK2*DE_DHHN92_NH

Zulässig sind Koordinatenangaben ausschließlich wie folgt:

1. 2D-Lageangabe
2. 2D-Lageangabe in Kombination mit einer Höhenangabe
3. 3D-Positionsangabe



3.1 Koordinatenreferenzsysteme für 2-D-Lageangaben

Beschreibung	AdV-Code
ETRS89, ellipsoidische (geographische) Koordinaten	ETRS89_Lat-Lon
WGS84, ellipsoidische (geographische) Koordinaten	WGS84_Lat-Lon
System ETRS89/UTM (Zonen 32 und 33)	ETRS89_UTM32 ETRS89_UTM33
DHDN, Gauß-Krüger-3-Grad-Streifen allgemein (Streifen 2, 3, 4, und 5)	DE_DHDN_3GK2 DE_DHDN_3GK3 DE_DHDN_3GK4 DE_DHDN_3GK5
DHDN, Gauß-Krüger-3-Grad-Streifen altes Lagefestpunktfeld (Reichsdreiecksnetz) DE	DE_DHDN_3GK2_RDN DE_DHDN_3GK3_RDN DE_DHDN_3GK4_RDN DE_DHDN_3GK5_RDN
BY	DE_DHDN_3GK2_BY120 DE_DHDN_3GK3_BY120 DE_DHDN_3GK4_BY120 DE_DHDN_3GK5_BY120
BE	DE_DHDN_3GK2_BE200 DE_DHDN_3GK3_BE200 DE_DHDN_3GK4_BE200 DE_DHDN_3GK5_BE200
HH	DE_DHDN_3GK2_HH100 DE_DHDN_3GK3_HH100 DE_DHDN_3GK4_HH100 DE_DHDN_3GK5_HH100
HE	DE_DHDN_3GK2_HE120 DE_DHDN_3GK3_HE120 DE_DHDN_3GK4_HE120 DE_DHDN_3GK5_HE120
NI	DE_DHDN_3GK2_NI200 DE_DHDN_3GK3_NI200 DE_DHDN_3GK4_NI200 DE_DHDN_3GK5_NI200
NW	DE_DHDN_3GK2_NW101



	DE_DHDN_3GK3_NW101 DE_DHDN_3GK4_NW101 DE_DHDN_3GK5_NW101
RP	DE_DHDN_3GK2_RP101 DE_DHDN_3GK3_RP101 DE_DHDN_3GK4_RP101 DE_DHDN_3GK5_RP101
ST	DE_DHDN_3GK2_ST200 DE_DHDN_3GK3_ST200 DE_DHDN_3GK4_ST200 DE_DHDN_3GK5_ST200
SH	DE_DHDN_3GK2_SH200 DE_DHDN_3GK3_SH200 DE_DHDN_3GK4_SH200 DE_DHDN_3GK5_SH200
TH	DE_DHDN_3GK2_TH200 DE_DHDN_3GK3_TH200 DE_DHDN_3GK4_TH200 DE_DHDN_3GK5_TH200
SL	DE_DHDN_3GK2_SL159 DE_DHDN_3GK3_SL159 DE_DHDN_3GK4_SL159 DE_DHDN_3GK5_SL159
DHDN, Gauß-Krüger-3-Grad-Streifen (landesweit vollständig erneuerte Systeme) BW	DE_DHDN_3GK2_BW100 DE_DHDN_3GK3_BW100 DE_DHDN_3GK4_BW100 DE_DHDN_3GK5_BW100
HB	DE_DHDN_3GK2_HB100 DE_DHDN_3GK3_HB100 DE_DHDN_3GK4_HB100 DE_DHDN_3GK5_HB100
NI	DE_DHDN_3GK2_NI000 DE_DHDN_3GK3_NI000 DE_DHDN_3GK4_NI000 DE_DHDN_3GK5_NI000
NI, ST	DE_DHDN_3GK2_NI100



	DE_DHDN_3GK3_NI100 DE_DHDN_3GK4_NI100 DE_DHDN_3GK5_NI100
NW	DE_DHDN_3GK2_NW177 DE_DHDN_3GK3_NW177 DE_DHDN_3GK4_NW177 DE_DHDN_3GK5_NW177
RP	DE_DHDN_3GK2_RP180 DE_DHDN_3GK3_RP180 DE_DHDN_3GK4_RP180 DE_DHDN_3GK5_RP180
HE	DE_DHDN_3GK2_HE100 DE_DHDN_3GK3_HE100 DE_DHDN_3GK4_HE100 DE_DHDN_3GK5_HE100
SL	DE_DHDN_3GK2_SL197 DE_DHDN_3GK3_SL197 DE_DHDN_3GK4_SL197 DE_DHDN_3GK5_SL197
DHDN, Gauß-Krüger-3-Grad-Streifen (partiell erneuerte Systeme) BY	DE_DHDN_3GK2_BY110 DE_DHDN_3GK3_BY110 DE_DHDN_3GK4_BY110 DE_DHDN_3GK5_BY110
HE	DE_DHDN_3GK2_HE110 DE_DHDN_3GK3_HE110 DE_DHDN_3GK4_HE110 DE_DHDN_3GK5_HE110
SH	DE_DHDN_3GK2_SH210 DE_DHDN_3GK3_SH210 DE_DHDN_3GK4_SH210 DE_DHDN_3GK5_SH210
TH	DE_DHDN_3GK2_TH210 DE_DHDN_3GK3_TH210 DE_DHDN_3GK4_TH210 DE_DHDN_3GK5_TH210
NI	DE_DHDN_3GK2_NI210



	DE_DHDN_3GK3_NI210 DE_DHDN_3GK4_NI210 DE_DHDN_3GK5_NI210
NW	DE_DHDN_3GK2_NW119 DE_DHDN_3GK3_NW119 DE_DHDN_3GK4_NW119 DE_DHDN_3GK5_NW119
NW	DE_DHDN_3GK2_NW131 DE_DHDN_3GK3_NW131 DE_DHDN_3GK4_NW131 DE_DHDN_3GK5_NW131
NW	DE_DHDN_3GK2_NW133 DE_DHDN_3GK3_NW133 DE_DHDN_3GK4_NW133 DE_DHDN_3GK5_NW133
NW	DE_DHDN_3GK2_NW158 DE_DHDN_3GK3_NW158 DE_DHDN_3GK4_NW158 DE_DHDN_3GK5_NW158
NW	DE_DHDN_3GK2_NW163 DE_DHDN_3GK3_NW163 DE_DHDN_3GK4_NW163 DE_DHDN_3GK5_NW163
NW	DE_DHDN_3GK2_NW166 DE_DHDN_3GK3_NW166 DE_DHDN_3GK4_NW166 DE_DHDN_3GK5_NW166
NW	DE_DHDN_3GK2_NW173 DE_DHDN_3GK3_NW173 DE_DHDN_3GK4_NW173 DE_DHDN_3GK5_NW173
NW	DE_DHDN_3GK2_NW174 DE_DHDN_3GK3_NW174 DE_DHDN_3GK4_NW174 DE_DHDN_3GK5_NW174
NW	DE_DHDN_3GK2_NW175



	DE_DHDN_3GK3_NW175 DE_DHDN_3GK4_NW175 DE_DHDN_3GK5_NW175
NW	DE_DHDN_3GK2_NW176 DE_DHDN_3GK3_NW176 DE_DHDN_3GK4_NW176 DE_DHDN_3GK5_NW176
System 40/83, GK-3-Grad (Streifen 4 und 5)	DE_40-83_3GK4 DE_40-83_3GK5
System 42/83, GK-3-Grad (Streifen 3, 4 und 5)	DE_42-83_3GK3 DE_42-83_3GK4 DE_42-83_3GK5
System 42/83, GK-6-Grad (Streifen 2 und 3)	DE_42-83_6GK2 DE_42-83_6GK3
System RD/83, GK-3-Grad (Streifen 4 und 5)	DE_RD-83_3GK4 DE_RD-83_3GK5

3.2 Koordinatenreferenzsysteme für 3-D-Positionsangaben

Beschreibung	AdV-Code
System ETRS89, ellipsoidische (geographische) Koordinaten incl. ellipsoidischer Höhe	ETRS89_Lat-Lon-h
System ETRS89, räumliche kartesische Koordinaten	ETRS89_X-Y-Z
System WGS84, räumliche kartesische Koordinaten	WGS84_X-Y-Z

3.3 Koordinatenreferenzsysteme für Höhenangaben

Beschreibung	AdV-Code
Höhe im System des DHHN 12	DE_DHHN12_NOH
Höhe im "Horizont 71" (BW)	DE_DHHN12_BW130
Normalorthometrische Höhe im System des DHHN 85	DE_DHHN85_NOH
Normalhöhe im System des SNN 76	DE_SNN76_NH
Normalhöhe im System des SNN 56	DE_SNN56_NH
Normalhöhe im System des DHHN 92	DE_DHHN92_NH
Normalhöhe im System des DHHN 2016	DE_DHHN2016_NH



Höhe im Nordseeküstennivellement I	DE_DHHN12_NOH_NKNI
Höhe im Nordseeküstennivellement II	DE_DHHN12_NOH_NKNII
Normalorthometrische Höhe im Nivellementnetz 1960 (Westblock)	DE_NIV60_NOH
Höhe im Ostseeküstennivellement	DE_OKN-I_NOH
Höhe im "alten System" der preußischen Landesaufnahme	DE_ALT_NN
Höhe im "vorläufigen System" in Bayern	DE_VORL_NOH_BY901

4 Festlegungen für den Datenaustausch

OKSTRA-XML-Dateien sind bei ihrer Erzeugung durch eine Software ausschließlich mit der Dateinamenserweiterung *.xml zu versehen, um eine universelle Verwendbarkeit der Dateien zu ermöglichen. Darüber hinaus können Softwareanwendungen auch nach dem gzip-Verfahren komprimierte OKSTRA-XML-Dateien erzeugen bzw. erwarten. Die Dateinamenserweiterung einer solchermaßen komprimierten OKSTRA-XML-Datei muss dann *.xml.gz lauten. Der Einsatz anderer Komprimierungsverfahren oder die Verwendung anderer Dateinamenserweiterungen sind nicht zulässig.

Die Möglichkeit zur Komprimierung / Dekomprimierung einer OKSTRA-XML-Datei von Hand (z. B. für den Versand per Email) bleibt von diesen Festlegungen unberührt.

Beim Einsatz des OKSTRA-XML-Datenformats für einen dateibasierten Datenaustausch wird aufgrund der dabei auftretenden Dateigrößen die oben beschriebene softwareseitige Komprimierung der OKSTRA-XML-Dateien nach dem gzip-Verfahren dringend empfohlen.

Für die Verschlüsselung von OKSTRA-Daten mit sensiblen Inhalten (z.B. personenbezogenen Daten) wird die Verwendung folgender Werkzeuge empfohlen:

- 7zip: Mit diesem Programm können einzelne Dateien oder ganze Verzeichnisbäume in ein Archiv (7z-Datei) verpackt und dabei verschlüsselt und mit einem Passwort gesichert werden. Gleichzeitig findet eine Komprimierung mit sehr hoher Kompressionsstärke statt. Sofern ein gutes Passwort gewählt und sicher übermittelt wird, ist eine hohe Sicherheit der Daten gewährleistet. Zusätzlich zu den Daten können auch die Namen der in einem Archiv enthaltenen Dateien verschlüsselt werden.
- VeraCrypt: Diese Anwendung ermöglicht die Anlage verschlüsselter Container-Dateien und ihre Anbindung als Laufwerk. Zu übertragende Dateien werden auf das Laufwerk kopiert. Sofern ein gutes Passwort gewählt und sicher übermittelt wird, ist aufgrund der starken Verschlüsselung des Containers eine sehr hohe Sicherheit der Daten gewährleistet. Im Gegenzug zu 7zip findet bei VeraCrypt keine Komprimierung der Daten statt, außerdem ist stets eine vollständige Container-Datei zu übertragen.