|  |
| --- |
| **NLRDMS (NL ERI Referentie Data)** |
|  |
|  |
| Interface Requirement Specification / Interface Design Description |
|  |

|  |  |  |
| --- | --- | --- |
| Datum |  |  |
| Status |  |  |

|  |
| --- |
| NLRDMS (NL ERI Referentie Data) |
|  |
|  |
| Interface Requirement Specification / Interface Design Description |
|  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Document ID** | **I** | IRSIDD-NLREF.NLRDMS-001-02 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Colofon |  | |
|  | | |
| Uitgegeven door |  | Rijkswaterstaat VWM/VFV |
| Informatie |  | Arjan Nieuwpoort |
| Uitgevoerd door |  | B. Bubberman |
| Datum |  |  |
| Status |  |  |
| Versienummer |  |  |

© 2021, Rijkswaterstaat RWS Dienst Verkeer en Scheepvaart. Alle rechten voorbehouden. Geen enkel deel van dit document mag worden vermenigvuldigd in welke vorm of door welke middelen dan ook zonder schriftelijke toestemming van de RWS Dienst Verkeer en Scheepvaart. Dit document is vertrouwelijk en mag alleen worden gebruikt voor de doeleinden waarvoor het is vrijgegeven.

Document History

| Version | Date | Modified by | Comments |
| --- | --- | --- | --- |
| 0.9 | 31-08-2015 | BB | Opzet |
| 0.9.4 | 10-11-2015 | BB/LS | Interne review LS (1e versie tbv ext review) |
| 1.0.0 | 16-11-2015 | BB | Input HWLS verwerkt |
| 1.0.1 | 07-12-2015 | BB | Input HK, Bjorn (CIV) verwerkt=> def versie |
| 1.0.2 | 06-10-2016 | BB | Update (correcties) nav de verhuizing en upload FIS->NLRDMS (RIS-idx) koppeling. |
| 1.0.3 | 23-07-2021 | BB | Tekstuele aanvullingen mbt refcodeOld gebruik. |
|  |  |  |  |

**Document Autorisatie**

| Autorisatie | Datum | Naam / functie | Handtekening |
| --- | --- | --- | --- |
| Review door |  |  |  |
| Geautoriseerd door |  |  |  |

**Inhoudsopgave**

[1 Scope 7](#_Toc78451317)

[1.1 Identificatie 7](#_Toc78451318)

[1.2 Systeemoverzicht 7](#_Toc78451319)

[1.3 Documentoverzicht 8](#_Toc78451320)

[1.3.1 Doel van de IRS-IDD 8](#_Toc78451321)

[1.3.2 Documentstructuur 8](#_Toc78451322)

[1.3.3 Aanwijzingen voor het lezen 9](#_Toc78451323)

[1.3.4 Beveiliging en intellectueel eigendom 10](#_Toc78451324)

[2 Aangehaalde documenten 11](#_Toc78451325)

[2.1 Normatieve documenten 11](#_Toc78451326)

[2.2 Informatieve documenten 12](#_Toc78451327)

[3 Interfaces 13](#_Toc78451328)

[3.1 Interface identificatie en diagrammen 13](#_Toc78451329)

[3.2 Algemene eisen 14](#_Toc78451330)

[3.3 NLR-EIF1 en NLR-EIF2 Gebruikersinterface 15](#_Toc78451331)

[3.3.1 Interface Eisen 15](#_Toc78451332)

[3.3.2 Ontwerpbeslissingen 15](#_Toc78451333)

[3.4 NLR-EIF3 Mutatie import/export, CSV bestandsinterface 15](#_Toc78451334)

[3.4.1 Interface Eisen 15](#_Toc78451335)

[3.4.2 Ontwerpbeslissingen 16](#_Toc78451336)

[3.5 NLR-EIF4 Referentiegegevens export, CSV bestandsinterface 18](#_Toc78451337)

[3.5.1 Interface Eisen 18](#_Toc78451338)

[3.5.2 Ontwerpbeslissingen 19](#_Toc78451339)

[3.6 NLR-EIF5 NLRDMS Webservice - WSDL/Soap interface 22](#_Toc78451340)

[3.6.1 Interface Eisen 22](#_Toc78451341)

[3.6.2 Ontwerpbeslissingen 24](#_Toc78451342)

[3.6.3 NLRDMS webservice interface, API functiebeschrijvingen 26](#_Toc78451343)

[3.6.3.1 matchByCode functie 27](#_Toc78451344)

[3.6.3.2 matchByName functie 29](#_Toc78451345)

[3.6.3.3 getMutatedTables functie 31](#_Toc78451346)

[3.6.3.4 getMutations functie 32](#_Toc78451347)

[3.6.3.5 getDataXML functie 34](#_Toc78451348)

[3.6.3.6 getRisDataXML functie 36](#_Toc78451349)

[3.6.3.7 mutateDataXML functie 37](#_Toc78451350)

[3.6.3.8 requestMutationXML functie 40](#_Toc78451351)

[3.6.4 NLRDMS Webservice interface, API datastructuren 42](#_Toc78451352)

[3.6.4.1 refdataReturnType 42](#_Toc78451353)

[3.6.4.2 refdataReqMutType 44](#_Toc78451354)

[3.6.4.3 mutatedTableReturnType 46](#_Toc78451355)

[3.6.4.4 statReturnType 46](#_Toc78451356)

[3.7 NLR-EIF6 ERDMS Webservice, WSDL/Soap interface 47](#_Toc78451357)

[3.7.1 Interface Eisen 47](#_Toc78451358)

[3.7.2 Ontwerpbeslissingen 48](#_Toc78451359)

[3.8 Prioriteit en afhankelijkheid van eisen 51](#_Toc78451360)

[4 Kwalificatiebepalingen 52](#_Toc78451361)

[5 Herleidbaarheid van eisen 53](#_Toc78451362)

[6 Opmerkingen 54](#_Toc78451363)

[6.1 Afkortingen en acroniemen 54](#_Toc78451364)

[6.2 Terminologie 56](#_Toc78451365)

[Bijlage A Beschrijving referentiegegevens velden 60](#_Toc78451366)

[A1 NLRDMS Webservice – WSDL 60](#_Toc78451367)

[A2 NLRDMS Webservice – XSD 63](#_Toc78451368)

[A3 NLRDMS Webservice – Veld formaten 99](#_Toc78451369)

[A4 NLRDMS Webservice - API Error codes 104](#_Toc78451370)

[Bijlage B NLRDMS Client implementatie stappen 108](#_Toc78451371)

**Inhoudsopgave figuren en eisen**

**Lijst van figuren**

[Figuur 1: NLRDMS context en externe interfaces 7](#_Toc78451372)

[Figuur 2: MIL-STD-498 / JSTD-016 documentenoverzicht 8](#_Toc78451373)

[Figuur 3: Identificatie van de NLRDMS interfaces 13](#_Toc78451374)

[Figuur 4: EIF5 NLRDMS webservice interface t.b.v. gekoppelde cliënt systemen 22](#_Toc78451375)

[Figuur 5: NLRDMS SOAP API operaties, berichtenscenario overzicht 25](#_Toc78451376)

[Figuur 6: EIF6 ERDMS webservice interface 47](#_Toc78451377)

[Figuur 7: Stroomdiagram, verwerken van ERDMS mutaties in NLRDMS 49](#_Toc78451378)

**Lijst van tabellen**

[Tabel 1: Normatieve documenten 12](#_Toc78451379)

[Tabel 2: Informatieve documenten 12](#_Toc78451380)

[Tabel 3: Kwalificatie methodes 52](#_Toc78451381)

**Lijst van IRS/IDD eisen en specificaties**

[IRS-NLR-001: NLRDMS interfaces en te gebruiken standaarden 14](#_Toc78451382)

[IDD-NLR-002: Implementatie NLRDMS Gebruikersinterface (EIF1 en EIF2) 15](#_Toc78451383)

[IRS-NLR-003: NLRDMS mutaties import/export (EIF3), kenmerken 15](#_Toc78451384)

[IDD-NLR-004: Implementatie NLRDMS mutatie import interface (EIF3.1) 16](#_Toc78451385)

[IDD-NLR-005: Implementatie NLRDMS mutatie export interface (EIF3.2) 17](#_Toc78451386)

[IRS-NLR-006: NLRDMS referentiegegevens (masterdump) export (EIF4.1) 18](#_Toc78451387)

[IRS-NLR-007: NLRDMS referentiegegevens (updates) export (EIF4.2) 19](#_Toc78451388)

[IDD-NLR-008: Implementatie Masterdump export interface (EIF4.1) 19](#_Toc78451389)

[IDD-NLR-009: Implementatie Updates export interface (EIF4.2) 20](#_Toc78451390)

[IRS-NLR-010: NLRDMS webservice interface (EIF5), technische kenmerken 23](#_Toc78451391)

[IDD-NLR-011: Implementatie NLRDMS webservice interface (SOAP API) (EIF5) 24](#_Toc78451392)

[IDD-NLR-012: Implementatie NLRDMS webservice interface, API functies (EIF5) 25](#_Toc78451393)

[IRS-NLR-013: ERDMS interface (EIF6) technische kenmerken 47](#_Toc78451394)

[IDD-NLR-014: Implementatie ERDMS interface (EIF6) in NLRDMS 48](#_Toc78451395)

[IDD-NLR-015: Implementatie ERDMS interface (EIF6), connectieparameters 48](#_Toc78451396)

[IDD-NLR-016: Implementatie NLRDMS <= ERDMS synchronisatie (EIF6.1) 49](#_Toc78451397)

[IDD-NLR-017: Implementatie NLRDMS => ERDMS synchronisatie (EIF6.2) 50](#_Toc78451398)

# Scope

## Identificatie

**Identification**. This paragraph shall contain a full identification of the system(s), the interfacing entities, and interfaces to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).

Dit document bevat zowel de *Interface Requirements Specification* (IRS) als de *Interface Design Description* (IDD) voor de verschillende interfaces tussen de NLRDMS Applicatie en de buitenwereld.

De belangrijkste NLRDMS interface in deze beschrijving, is de *interface voor derden* (zie Interface #5, paragraaf 3.6).

De interface voor derden is een webservice API interface. Deze interface verzorgt het kunnen koppelen van NL RIS (client)systemen aan het NLRDMS ten behoeve van het geautomatiseerd kunnen ophalen/aanleveren van de referentiegegevens (zie paragraaf 3.6).

Dit document wordt geïdentificeerd als **IRSIDD-NLREF.NLRDMS**.

Het doel en de verdere opbouw van dit document, inclusief leeswijzer, zijn terug te vinden in paragraaf 1.3.

## Systeemoverzicht

**System overview**. This paragraph shall briefly state the purpose of the system(s) and software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, acquirer, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.



Figuur 1: NLRDMS context en externe interfaces

 Dit symbool in de bovenstaande figuur kenmerkt een externe interface, die beschikbaar wordt gesteld door het systeem en gebruikt wordt door een ander extern systeem.

De NLRDMS applicatie voorziet, voor Rijkswaterstaat en haar partners in de landelijke functionaliteit met betrekking tot het (Europese) referentiegegevensbeheer. NLRDMS is daarmee *het* systeem in Nederland, gekoppeld met Europa, voor het beheren en distribueren van de actuele ERI referentiegegevens naar de Nederlandse RIS systemen en applicaties.

De hoofdfunctionaliteiten van de NLRDMS applicatie bestaan in het kort uit:

* Het publiceren (ter beschikking stellen) van de actuele RIS (ERI) referentiegegevens t.b.v. Nederlandse RIS systemen, applicaties en gebruikers, die te maken hebben met de Internationaal gestandaardiseerde ERI berichten (zie [EU164]) en of gegevensuitwisseling.
* Het kunnen beheren van de Nederlandse land specifieke gegevens en het uploaden van de relevante gegevens (NL RIS-index) naar het Europese referentiedata systeem, het ERDMS.

Dit document beschrijft de **externe** interfaces van NLRDMS, zoals benoemd in de [SSS-NLRDMS] paragraaf 3.3 (zie ook Figuur 1) en onderverdeeld in de volgende soorten:

* De *Gebruikersinterface (#1)* en *Gebruikersinterface (#2) voor de Administrator* (de NL Datamanager), zoals de menustructuur, formulieren en functies (zie paragraaf 3.3).
* De *Mutatie import interface (#3)*, waarmee mutaties uit een extern bestand via de GUI kunnen worden geïmporteerd (zie paragraaf 3.4).
* De *Referentiegegevens export interface (#4)* (Masterdumps en Updatebestanden), een interface voor het publiceren (handmatig opvragen) van de relevante referentiegegevens (zie paragraaf 3.5).
* De NL W*ebservice interface (#5)* voor externe gekoppelde (cliënt) systemen en of applicaties (zie paragraaf 3.6.
* De *ERDMS interface (#6),* ten behoeve van het automatisch synchroniseren met het ERDMS (Europese gegevens), die door ERDMS beschikbaar wordt gesteld (zie paragraaf 3.7).

## Documentoverzicht

**Document overview**. This paragraph shall summarize the **purpose** and **contents of this document** and shall describe any security or privacy considerations associated with its use.

### Doel van de IRS-IDD

Dit IRS-IDD beschrijft zowel de specificaties van, evenals het ontwerp van, de interfaces. Het IRS-IDD is een aanpassing van de J-STD-016 standaard voor Rijkswaterstaat en is een combinatie van het IRS en IDD. Tevens dient dit document als specificatie voor externe systemen die gebruik wensen te maken van deze interface(s).

Het document is vooral van belang voor ontwikkelaars, die een RIS cliëntsysteem willen aansluiten op de NLRDMS webservice interface (zie paragraaf 3.6).

### Documentstructuur

Dit combinatie document is gebaseerd op de IRS en IDD standaard uit de J-STD-016 standaard [JSTD].

In het onderstaande JSTD documentenoverzicht is het huidige (gecombineerde) document geel gemarkeerd.



Figuur 2: MIL-STD-498 / JSTD-016 documentenoverzicht

Het document is verder opgedeeld in de volgende hoofdstukken:

* Hoofdstuk 2 bevat verwijzingen naar andere documenten.
* Hoofdstuk 3 beschrijft de verschillende interface-onderdelen. Dit is opgesplitst in algemene eisen en eisen specifiek voor de geïdentificeerde interface.
* Hoofdstuk 4 beschrijft hoe de verschillende IRS/IDD eisen getest gaan worden.
* Hoofdstuk 5 traceert lokale eisen naar bovenliggende eisen.
* Hoofdstuk 6 geeft een overzicht van de gebruikte definities en afkortingen binnen het document.

### Aanwijzingen voor het lezen

De interface eisen en ontwerpbeslissingen van NLRDMS zijn vastgelegd door middel van diverse specificaties en begeleidende teksten in dit gecombineerde IRS/IDD document.

Waar in dit document wordt verwezen naar een gebruiker met de woorden *hij* of *zijn*, moet dit gelezen worden als zowel de mannelijke en vrouwelijke vorm: *hij* of *zij* en *zijn* of *haar*.

Per Externe Interface zijn de Interface eisen *en* de Ontwerpbeslissingen (de implementatie) beschreven.

Interface eisen in dit document worden geïdentificeerd door de tekst **'IRS-NLR-xxx**', waarbij 'xxx' het nummer van de eis is.

Ontwerpbeslissingen en implementatiezaken worden in dit document geïdentificeerd door de tekst ‘**IDD-NLR-xxx**’.

De IRS en IDD eisen zijn in dit document als aanklikbare (hyperlinks) verwijzingen opgenomen.

Elke eis of ontwerpbeslissing wordt uitgewerkt, voorzien van een unieke identificatie en de volgende aanvullende beschrijvende elementen:

* een korte beschrijvende *Titel* (steekwoorden), waarmee de belangrijkste kenmerken van die eis worden geïdentificeerd.
* een *Beschrijving,* waarin de eis of specificatie in tekst duidelijk en eenduidig wordt beschreven.
* een eventuele *Toelichting* en of extra achtergrond informatie ter verduidelijking van die eis.
* een eventuele *Referentie* en of verwijzing naar een (extern) normatief document ter onderbouwing en of verduidelijking van die eis.
* een eventuele *Herkomst* van de eis t.b.v. de herleidbaarheid, zijnde een verwijzing naar een (extern bovenliggend) document en of eis.
* de *Kwalificatie* van die eis, geeft aan hoe de eis beoordeelt/getoetst kan/moet worden (zie hoofdstuk 4 voor een beschrijving van de mogelijke kwalificaties).
* een eventueel van belang zijnde relatie (*Gerelateerd*) met andere eisen in dit document (dwarsverbanden).

De context informatie die in dit document is beschreven dient als aanvulling op de expliciete eisen.

Verwijzingen naar bovenliggende (externe) eisen zijn te herkennen aan een referentie “UC-NLREF” (use cases uit de OCD) *of* “SSS-NLR” (eisen uit de SSS).

Verwijzing naar externe documenten vindt plaats aan de hand van de desbetreffende documentnaam tussen []. Bijvoorbeeld [JSTD], verwijst naar de algemene J-STD-016 standaard (zie hoofdstuk 2 voor de lijst van referentie documenten).

### Beveiliging en intellectueel eigendom

Zie de auteursrechtvermelding op de voorpagina.

# Aangehaalde documenten

**Referenced documents**. This section shall list the number, title, revision, and date of all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.

## Normatieve documenten

| Referentie | Omschrijving |
| --- | --- |
| [JSTD] | EIA/IEEE J-STD-016:1995 Standard for Information Technology - Software Life Cycle Processes - Software development: Acquirer-Supplier Agreement. |
| [SSS-NLRDMS] | De *Systeem Subsysteem Specificatie van het NLRDMS*, een onderdeel van het NLREF system. |
| [OCD-NLREF] | De *Operational Concept Description van het NLREF systeem*, waar het NLRDMS een onderdeel van is. |
| [SOAP] | Simple Object Access Protocol (SOAP), v1.2 |
| [ERIRDMAINT] | MAINTENANCE PROCEDURES REFERENCE DATA AND RIS INDEX Version 2010-08-22  *Annex 3 to the report “consolidation of the RIS Index and Reference data”* |
| [ERDMS\_WSI] | De ERDMS webservice interface **v1.0g** is beschreven in het document: “*2011-0415 PLATINA RISDataManagement Service API Interface Final 20110728.doc*”  De ERDMS *vernieuwde* (voorkeurs) webservice Interface **v1.1b** is beschreven in ”*2015-0801 PLATINA RISDataManagement Service API Interface Final v11 20150801.doc*”.  Deze laatste interface is “RIS-index encoding guide v2.0” compliant. |
| [ERDMS\_WSDL] | De *ERDMS webservice Interface (API)* **1.0g** (d.d. juni 2011):  [https://webgate.ec.europa.eu/RIS/EUERDMS\_WS/v1/RefWebService?wsdl](https://webmail.rws.nl/owa/redir.aspx?REF=zkdGdAkKWfIpi2CTwLhdcl64SFIGU6b9NSXv3NedAW2GfqKbpp3TCAFodHRwczovL3dlYmdhdGUuZWMuZXVyb3BhLmV1L1JJUy9FVUVSRE1TX1dTL3YxL1JlZldlYlNlcnZpY2U_d3NkbA..)  De *vernieuwde* ERDMS Interface (API) **1.1b** (d.d. aug 2015):  <https://webgate.ec.europa.eu/RIS/EUERDMS_WS11/v11/RefWebService?wsdl>  Test/acceptatie omgevingen:  [https://webgate.acceptance.ec.europa.eu/RIS/EUERDMS\_WS/v1/RefWebService?wsdl](https://webmail.rws.nl/owa/redir.aspx?REF=1NfmjG1nYeul8ZM1fqokebf2LPZ9AQUH-FIYxUZhM-WGfqKbpp3TCAFodHRwczovL3dlYmdhdGUuYWNjZXB0YW5jZS5lYy5ldXJvcGEuZXUvUklTL0VVRVJETVNfV1MvdjEvUmVmV2ViU2VydmljZT93c2Rs) [https://webgate.acceptance.ec.europa.eu/RIS/EUERDMS\_WS11/v11/RefWebService?wsdl](https://webmail.rws.nl/owa/redir.aspx?REF=Wki8Q-Upw6XAKK_ZGTV6Xj2XOYRJ5qMrvU_SxoUOiLSGfqKbpp3TCAFodHRwczovL3dlYmdhdGUuYWNjZXB0YW5jZS5lYy5ldXJvcGEuZXUvUklTL0VVRVJETVNfV1MxMS92MTEvUmVmV2ViU2VydmljZT93c2Rs) |
| [NLRDMS\_GUI] | De *NLRDMS gebruikersinterface* is op te roepen via de volgende URL van NLRDMS.  De NLRDMS GUI URL:  *- Productie omgeving:* [*https://nlrdms.rws.nl*](https://nlrdms.rws.nl)  *- Acc omgeving:* [*https://acceptatie.nlrdms.rws.nl*](https://acceptatie.nlrdms.rws.nl)  *- Test omgeving:* [*https://test.nlrdms.rws.nl*](https://test.nlrdms.rws.nl) |
| [NLRDMS\_WSDL] | De **NLRDMS** **WSDL** **v1.1a** (juni 2013), die de webservice interface definieert, is op te vragen bij NLRDMS via een URL (zie eis IDD-NLR-011) en is ook in de bijlage van dit document opgenomen.  NLRDMS (web service) WSDL URL:  *Productie omgeving:*  [*https://nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl*](https://nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl)  Acceptatie omgeving:  [*https://acceptatie.nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl*](https://acceptatie.nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl)  *Test omgeving:* [*http://test.nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl*](http://test.nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl) |
| [NLRDMS\_SCHEMA] | Het NLRDMS schema is de WSDL behorende XSD **v1.1a** (juni 2013), die de datastructuren in de webservice interface definieert, is op te vragen bij NLRDMS (zie eis IDD-NLR-011) en is ook in de bijlage van dit document opgenomen. |
| [RFC4180] | Common Format and MIME Type for Comma-Separated Values (CSV) Files, oktober 2005. |
| [NLRDMS\_CLIENT] | Aansluitprocedure om een partnersysteem op het NLRDMS aan te sluiten (te koppelen met NLRDMS).  *“NLRDMS Client implementation steps” v0.3 dec 2013.* |
| [BIR] | “*Baseline Informatiebeveiliging Rijksdienst*” v1.0 dec 2012.  Richtlijnen voor de informatiebeveiliging van applicaties. |

Tabel 1: Normatieve documenten

## Informatieve documenten

| Referentie | Omschrijving |
| --- | --- |
| [BICS4\_UPD] | Specificaties van de Updatebestanden voor het Update mechanisme van de oude (inmiddels uitgefaseerde) BICS versie 4 applicatie. Via dit mechanisme konden de BICS versie 4 referentiegegevens (via een email bericht) op afstand worden bijgewerkt.  Document “*BICS update mechanisme v3.07*” d.d. 23 jan 2007. |
| [RISIDX\_ENC] | RIS-Index Encoding guide v2.0 d.d. 23.06.2015.  Beschrijft de opbouw en invul instructie voor de RIS-Index. |

Tabel 2: Informatieve documenten

# Interfaces

**Interface Requirements (IRS).** This section shall be divided into the following paragraphs to specify the requirements imposed on one or more systems, subsystems, configuration items, manual operations, or other system components to achieve one or more interfaces among these entities.

Each requirement shall be assigned a project-unique identifier to support testing and traceability and shall be stated in such a way that an objective test can be defined for it. Each requirement shall be annotated with associated qualification method(s) (see section 4) and traceability to system (or subsystem, if applicable) requirements (see section 5.a) if not provided in those sections.

The degree of detail to be provided shall be guided by the following rule: Include those characteristics of the interfacing entities that are conditions for their acceptance; defer to design descriptions those characteristics that the acquirer is willing to leave up to the developer.

If a given requirement fits into more than one paragraph, it may be stated once and referenced from the other paragraphs. If an interfacing entity included in this specification will operate in states and/or modes having interface requirements different from other states and modes, each requirement or group of requirements for that entity shall be correlated to the states and modes.

The correlation may be indicated by a table or other method in this paragraph, in an appendix referenced from this paragraph, or by annotation of the requirements in the paragraphs where they appear.

**Interface design (IDD)**. This section shall be divided into the following paragraphs to describe the interface characteristics of one or more systems, subsystems, configuration items, manual operations, or other system components.

If part or all of the design depends upon system states or modes, this dependency shall be indicated.

If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs.

If part or all of this information is documented elsewhere, it may be referenced. Design conventions needed to understand the design shall be presented or referenced.

## Interface identificatie en diagrammen

**3.1 Interface identification and diagrams**. For each interface identified in 1.1, this paragraph shall state the project-unique identifier assigned to the interface and shall identify the interfacing entities (systems, configuration items, users, etc.) by name, number, version, and documentation references, as applicable.

The identification shall state which entities have fixed interface characteristics (and therefore impose interface requirements on interfacing entities) and which are being developed or modified (thus having interface requirements imposed on them). One or more interface diagrams shall be provided, as appropriate, to depict the interfaces.

**3.x (Project-unique identifier of interface).** This paragraph (beginning with 3.2) shall identify an interface by project-unique identifier, shall briefly identify the interfacing entities, and shall be divided into subparagraphs as needed to state the requirements imposed on one or more of the interfacing entities to achieve the interface. If the interface characteristics of an entity are not covered by this IRS but need to be mentioned to specify the requirements for entities that are, those characteristics shall be stated as assumptions or as "When [the entity not covered] does this, the [entity being specified] shall...," rather than as requirements on the entities not covered by this IRS. This paragraph may reference other documents (such as data dictionaries, standards for communication protocols, and standards for user interfaces) in place of stating the information here. The requirements shall include the following, as applicable, presented in any order suited to the requirements, and shall note any differences in these characteristics from the point of view of the interfacing entities (such as different expectations about the size, frequency, or other characteristics of data elements):

a. Priority that the interfacing entity(ies) must assign the interface.

b. Requirements on the type of interface (such as real-time data transfer, storage-andretrieval of data, etc.) to be implemented.

c. Required characteristics of individual data elements that the interfacing entity(ies) must provide, store, send, access, receive, etc.

d. Required characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the interfacing entity(ies) must provide, store, send, access, receive, etc.

e. Required characteristics of communication methods that the interfacing entity(ies) must use for the interface.

f. Required characteristics of protocols the interfacing entity(ies) must use for the interface.

g. Other required characteristics, such as physical compatibility of the interfacing entities (dimensions, tolerances, loads, plug compatibility, etc.), voltages, etc.

De paragrafen in dit hoofdstuk beschrijven de externe interfaces zoals geïdentificeerd in de [SSS-NLRDMS] paragraaf 3.3 en zoals ook schematisch samengevat in Figuur 1 van dit document.

Onderstaande figuur geeft vanuit NLRDMS perspectief gezien een overzicht van de beschikbare interfaces (koppelingen) en de bijbehorende identificatie.



Figuur 3: Identificatie van de NLRDMS interfaces

De interfaces IRS-NLR-EIF1 en IRS-NLR-EIF2 voorzien in het bieden van de gebruikersinterface (menu’s, functies en formulieren) om de applicatie te kunnen bedienen en gegevens (handmatig) te kunnen invoeren.

De gewenste structuur van deze gebruikersinterfaces zijn gedefinieerd in de [SSS-NLRDMS], eis [SSS-NLR-008] en worden hier **niet** verder uitgewerkt.

De interface IRS-NLR-EIF3, beschikbaar gesteld door NLRDMS, voorziet in het kunnen importeren en exporteren via de GUI van mutaties middels een CSV-bestand in een bepaald formaat, met daarin de gewenste mutaties (zie paragraaf 3.4). Deze interface valt uiteen in:

* IRS-NLR-EIF3.1 Mutatie import CSV bestandsinterface.
* IRS-NLR-EIF3.2 Mutatie export CSV bestandsinterface.

De interface IRS-NLR-EIF4, beschikbaar gesteld door NLRDMS, geeft de mogelijkheid referentiegegevens en of mutaties te exporteren als CSV-bestand (zie paragraaf 3.5). Deze interface valt uiteen in:

* IRS-NLR-EIF4.1 Masterdump interface, een CSV bestand met een dump van de gewenste gegevens.
* IRS-NLR-EIF4.2 Updatebestand interface, een CSV-bestand met een dump van alle mutaties in een opgegeven periode.

De interface IRS-NLR-EIF5, beschikbaar gesteld door NLRDMS, voorziet in het via een webservice kunnen koppelen van clientsystemen (zie paragraaf 3.6). Deze interface valt uit:

* IRS-NLR-EIF5.1 webservice opvraag interface, waarbij gegevens alleen aan NLRDMS worden opgevraagd t.b.v. RIS applicaties die gebruik willen maken van referentiegegevens, zoals BICS.
* IRS-NLR-EIF5.2 webservice mutatie interface, waarbij het gekoppelde (geautoriseerde) systeem ook gegevens kan aanleveren aan het NLRDMS.

De interface IRS-NLR-EIF6, beschikbaargesteld door ERDMS, wordt gebruikt door NLRDMS voor de synchronisatie met de Europese database, het ERDMS (zie paragraaf 3.7).

## Algemene eisen

De eisen in deze paragraaf zijn op alle interfaces (zie Figuur 3) van toepassing.

| IRS-NLR-001: NLRDMS interfaces en te gebruiken standaarden | |
| --- | --- |
| Beschrijving: | NLRDMS stelt de volgende externe interfaces beschikbaar:   * IRS-NLR-EIF1 de gebruikersinterface voor de NL Datamanager. * IRS-NLR-EIF2 de gebruikersinterface voor de ‘gewone’ gebruiker. * IRS-NLR-EIF3 de mutatie import/export interface. * IRS-NLR-EIF4 de referentiegegevens export interface. * IRS-NLR-EIF5 de NL webservice interface voor het aansluiten van NL RIS clientsystemen.   Interfaces beschikbaar gesteld door andere systemen:   * IRS-NLR-EIF6 de koppeling met ERDMS.   De interfaces worden gebaseerd op gebruikelijke Internationale standaarden, zoals:   |  |  |  | | --- | --- | --- | | **Interface** | **Standaard** | **Beveiligd** | | EIF1,  EIF2 | Gebaseerd op de gangbare algemene userinterface standaarden voor web applicaties. | Ja (gedeeltelijk, rest na inloggen). | | EIF3 | Mutatie bestanden gebaseerd op de CSV standaard [RFC4180], veld volgorde bepaald door NLRDMS. | Ja (na inloggen) | | EIF4 | Export bestanden gebaseerd op de CSV standaard [RFC4180] en veld volgorde bepaald door NLRDMS. | Nee (Masterdumps),  Ja (Updates) | | EIF5 | Gebaseerd op [SOAP] over [HTTP], stijl “Document/literal (wrapped)”. | Ja (API beveiligd via generiek wachtwoord en voor upload is een account nodig). | | EIF6 | Gebaseerd op [SOAP] over [http] en conform [ERDMS\_WSI]. | Ja (ERDMS API beveiligd). |   Systemen en of applicaties, die gebruik willen maken van een NLRDMS interface, dienen te voldoen aan de gestelde standaarden en bijbehorende interface specificaties.  De (wel of niet) toegang tot de interfaces dient op systeem (technisch) niveau configureerbaar te zijn. |
| Toelichting: | - |
| Referentie: | - |
| Herkomst: | SSS-NLR-056 NLRDMS externe interfaces. |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

## NLR-EIF1 en NLR-EIF2 Gebruikersinterface

### Interface Eisen

De gebruikersinterfaces zijn al gedefinieerd en functioneel uitgewerkt in de [SSS-NLRDMS] paragraaf 3.3 en worden hier **niet** verder uitgewerkt.

### Ontwerpbeslissingen

Gebruikers hebben alleen via de gebruikersinterface toegang tot de applicatie en bijbehorende gegevens.

| IDD-NLR-002: Implementatie NLRDMS Gebruikersinterface (EIF1 en EIF2) | |
| --- | --- |
| Beschrijving: | Gebruikers hebben alleen via de GUI toegang tot het systeem en de beschikbare gegevens. Bepaalde hoofdfuncties, zoals het muteren van gegevens, zijn beveiligd waarbij gebruikers via een inlogprocedure moeten inloggen op het systeem. Hiervoor hebben zij een geldig en geactiveerd account nodig. |
| Toelichting: | De accountgegevens en bijbehorende rol (de gebruikersadministratie) worden beheerd door de NL Datamanager. |
| Referentie: | [BIR] 11.2 Beheer van toegangsrechten.  [BIR] 11.5 Toegangsbeveiliging voor besturingssystemen. |
| Herkomst: | SSS-NLR-057 Externe interfaces, IRS-NLR-001 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | SSS-NLR-008: Gebruikersinterface, GUI opbouw. |

## NLR-EIF3 Mutatie import/export, CSV bestandsinterface

### Interface Eisen

De mutatie import interface voorziet in het aanleveren, invoeren van mutaties vanuit andere systemen via een door de gebruiker te importeren CSV-bestand.

Het CSV-bestand moet in basis voldoen aan algemene standaarden en een vastgestelde veld volgorde, waarbij een beperkte variatie mogelijk is, zoals: wel of geen header, gebruikte separator.

NLRDMS voorziet ook in het exporteren van eerder ingevoerde mutaties naar een CSV-bestand, in hetzelfde formaat als een te importeren mutatiebestand.

| IRS-NLR-003: NLRDMS mutaties import/export (EIF3), kenmerken | |
| --- | --- |
| Beschrijving: | De mutatie import/export interface biedt een mogelijkheid om een CSV-bestand met mutaties te importeren via de GUI.  Het formaat van het te importeren CSV bestand kan enigszins worden gespecificeerd bestaande uit: een vrij op te geven separator met of zonder headerregel.  Het formaat van het CSV bestand (velden en volgorde) is afhankelijk van het type referentiegegeven en gelijk aan het mutatie export formaat.  Voor de volgende gegevens is het toegestaan mutaties te importeren:   * RIS-Index (Nederlandse locaties). * ERI Locaties (Nederlandse locaties). * Eerste meldpunten t.b.v. Elektronisch melden. * Vaarweg-autoriteiten (ontvangende systemen). * Koppelgegevens meldpunten <=> vertrekplaatsen.   Deze import functie is alleen toegankelijk voor ingelogde gebruikers.  Het initiatief voor het gebruiken van deze import/export interface ligt bij de (ingelogde) NLRDMS gebruiker. |
| Toelichting: | De NLRDMS voorziet in het kunnen muteren (verwerken) van alleen de Nederlandse referentiegegevens. |
| Referentie: | - |
| Herkomst: | SSS-NLR-058 Mutatie import/export interface |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

### Ontwerpbeslissingen

Het import formaat (veld volgorde) is gelijk aan het export formaat en het bestand moet voldoen aan de algemene normen voor een tekstueel CSV-bestand.

| IDD-NLR-004: Implementatie NLRDMS mutatie import interface (EIF3.1) | |
| --- | --- |
| Beschrijving: | De mutatie **import** interface, toegankelijk voor een ingelogde gebruiker, voorziet in het kunnen inlezen (uploaden) van een mutatie CSV-bestand.  Het uploaden van het import bestand verloopt via het [HTTP] post commando.  De inhoud van het mutatie CSV-bestand dient qua formaat te voldoen aan [RFC4180].  Het import CSV-bestand bevat een optionele zelfbeschrijvende headerregel en een eenduidige separator.  De inhoud van het te importeren mutatiebestand dient de volgende kolommen te bevatten:   * Type referentiegegeven (kolom “reftype”):   *mogelijke waarden:*   * + “RISINDEX” voor RIS-Index mutaties.   + “LOCATION” voor ERI locaties mutaties.   + “ERIREPORTINGPOINT” 1e meldpunten mutaties.   + “ERIPROVIDER” vaarweg autoriteiten mutaties.   + “LOCSREPPOINTS” koppelgegevens meldpunten, vertrekplaatsen (mutaties). * Oude referentiecode (kolom “refcodeold”), * Referentiecode, * Referentiegegeven afhankelijke kolommen.   Indien het bestand niet kan worden geimporteerd (bestand voldoet niet aan de specificaties), dan wordt er een foutmelding gegeven en wordt er niets geïmporteerd. |
| Toelichting: | Het formaat (veld volgorde etc) van het mutatie importbestand is exact gelijk aan het mutatie exportbestand.  Een voorbeeld mutatie importbestand voor een bepaald referentiegegevens is eenvoudig aan te maken door een ingevoerde mutatie te exporteren (zie IDD-NLR-005). |
| Referentie: | - |
| Herkomst: | IRS-NLR-003 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | Zie ook IDD-NLR-005 Export interface |

| IDD-NLR-005: Implementatie NLRDMS mutatie export interface (EIF3.2) | |
| --- | --- |
| Beschrijving: | De mutatie **export** interface, toegankelijk voor een inglogede gebruiker, voorziet in het kunnen aanmaken van een mutatie CSV-bestand.  De inhoud van het aan te maken mutatie CSV-bestand dient qua formaat te voldoen aan [RFC4180].  Het export CSV-bestand wordt aangemaakt met een verplichte zelfbeschrijvende headerregel, waarin de veldnamen van het desbetreffende referentiegegeven staan.  Bij het aanmaken van het exportbestand wordt het “| “ teken (vertikale streep of staafje) als separator gebruikt.  De inhoud van het export mutatiebestand bestaat uit de volgende kolommen:   * Type referentiegegeven, * Oude referentiecode, * Referentiecode, * Referentiegegeven afhankelijke kolommen. |
| Toelichting: | Het formaat (veld volgorde etc) van het mutatie exportbestand is gelijk aan het mutatie importbestand. Zodoende kan het mutatie exportbestand altijd als voorbeeld worden gebruikt om mutaties te importeren.  De uitvoer van de export wordt rechtstreeks naar de gebruiker gestuurd (output in de browser). Hij kan de informatie vervolgens selecteren en met ‘Copy & Paste’ in elke gewenste tekst editor overnemen en daarna opslaan. |
| Referentie: | - |
| Herkomst: | IRS-NLR-003 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | Zie ook IDD-NLR-004 Import interface |

Hieronder enkele voorbeelden van een **NLRDMS mutatiebestand**:

ERI Locaties:

"reftype"|"refcodeold"|"srscode"|"riscode"|"locode"|"fwcode"|"kmcode"|"type"|"termcode"|"termtype"|"name\_LOCAL"|"name\_NL"|"name\_DE"|"name\_FR"|"name\_EN"|"term\_LOCAL"|"term\_NL"|"term\_DE"|"term\_FR"|"term\_EN"|"quayfrom"|"quayto"|"lat"|"lon"|"source"|"remarks"|"erased"|"ivs"|"ivscode"|"ivsname"|"ivsterm"|"ivsreport"|"exits"

"LOCATION"|""|"NLDOR"|"NLDOR001014538701166"|"NLDOR"|""|""|"64"|""|"0"|"DORDRECHT (NLDOR)"|"DORDRECHT (NLDOR)"|"DORDRECHT (NLDOR)"|"DORDRECHT (NLDOR)"|"DORDRECHT (NLDOR)"|""|""|""|""|""|""|""|"51.81670000"|"4.66670000"|""|"Kale UNLOC Specified (new) Code NLDOR already exists!"|"false"|"IVS90"|"3550500"|"DORDRECHT"|""|"false"|"1"|"G10UD"|""|""|""|""|""|""|""|""|""

Eerste meldpunten (ERI reporting points):

"reftype"|"refcodeold"|"reppoint"|"name\_LOCAL"|"name\_NL"|"name\_DE"|"name\_FR"|"name\_EN"|"source"|"remarks"|"erased"|"RIScode"|"Providerid"

"ERIREPORTINGPOINT"|""|"A00"|"IVS90 (NL)"|"IVS90 NEDERLAND (NL)"|"IVS90 (NL)"|"IVS90 (NL)"|"IVS90 (RIS NL)"|"BICS402"|"Test Reporting point"|"false"|"NLDOR001014538701166"|"IVS90"

## NLR-EIF4 Referentiegegevens export, CSV bestandsinterface

### Interface Eisen

De referentiegegevens export interface voorziet via de GUI in het (handmatig) exporteren van referentiegegevens en of updates in een bepaalde periode, o.a. om externe systemen bij te kunnen werken.

Deze export interface valt uiteen in:

* EIF4.1: het exporteren van alle actuele referentiegegevens (zogenaamde masterdumps).
* EIF4.2: het exporteren van de updates (mutaties) binnen een op te geven tijdsperiode.

Het aan te maken export CSV-bestand moet in de basis voldoen aan algemene standaarden, waarbij een bepaalde variatie mogelijk is, zoals: wel of geen header en de gebruikte separator.

| IRS-NLR-006: NLRDMS referentiegegevens (masterdump) export (EIF4.1) | |
| --- | --- |
| Beschrijving: | De referentiegegevens masterdump export interface voorziet (voor alle gebruikers) in het handmatig kunnen aanmaken van een CSV exportbestand (een zogenaande masterdump) voor een op te geven soort van referentiegegevens.  Het formaat van het CSV bestand kan enigszins worden gespecificeerd door de gebruiker, via de op te geven parameters:   * wel of geen headerregel, * vrij op te geven separator. * wel of geen “double quoted” waarden.   Verder kan de verzameling van te dumpen gegevens worden beïnvloed, door het opgeven van: een subcode, of de historie (verwijderde gegevens) wel of niet moet worden meegenomen.  Het exporteren van referentiegegevens is mogelijk voor alle type referentiegegevens.  Het initiatief voor het gebruiken van deze interface en het aanmaken van de masterdump bestanden ligt bij een NLRDMS gebruiker (wel of niet ingelogd). |
| Toelichting: | De zogenaamde masterdump export voorziet in het dumpen van de beschikbare referentiegegevens ten behoeve van de (initiële) vulling van externe systemen.  De masterdumps zijn via deze interface op te vragen door alle gebruikers (iedereen). |
| Referentie: | - |
| Herkomst: | SSS-NLR-059 Referentiegegevens export interface |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

| IRS-NLR-007: NLRDMS referentiegegevens (updates) export (EIF4.2) | |
| --- | --- |
| Beschrijving: | De updates export interface voorziet in het handmatig kunnen aanmaken van een zogenaamde Update voor een op te geven periode, bestaande uit een verzameling CSV update bestanden voor de diverse soorten referentiegegevens.  Deze export functie is alleen toegankelijk voor de NL Datamanager.  Het formaat van de CSV bestanden is verder niet te specificeren.  Het initiatief voor het gebruiken van deze interface en het aanmaken van de masterdump bestanden ligt bij de NLRDMS Datamanager. |
| Toelichting: | De updates export voorziet in het exporteren van de gemuteerde referentiegegevens over een bepaalde periode, ten behoeve van het (incrementeel) kunnen bijwerken van externe systemen. |
| Referentie: | - |
| Herkomst: | SSS-NLR-059 Referentiegegevens export interface / updatebestand |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

### Ontwerpbeslissingen

De masterdump voorziet in het dumpen van hele tabellen. Daarnaast voorziet de update export interface in het exporteren van verwerkte mutaties in een opgegeven periode.

| IDD-NLR-008: Implementatie Masterdump export interface (EIF4.1) | |
| --- | --- |
| Beschrijving: | De masterdump export interface voorziet in het kunnen aanmaken van een CSV exportbestand.  De inhoud van het aan te maken CSV-bestand dient qua formaat te voldoen aan [RFC4180].  Het masterdump CSV-bestand wordt aangemaakt op basis van de opgegeven parameters (voor mogelijkheden zie IRS-NLR-006).  De inhoud van het export bestand bestaat uit de volgende kolommen:   * Gegeven wel of niet (logisch) verwijderd (met historie), * Datum / tijd van laatste mutaties, * Oude referentiecode, * Referentiecode, * Referentiegegeven afhankelijke kolommen. |
| Toelichting: | De uitvoer van de masterdump (export) wordt rechtstreeks naar de gebruiker gestuurd (output in de browser). Hij kan de informatie vervolgens selecteren en met ‘Copy & Paste’ in elke gewenste tekst editor overnemen om het daarna op te slaan in een bestand.  Omdat het niet-privacy gevoelige gegevens betreft, is deze export interface voor iedereen toegankelijk. |
| Referentie: | - |
| Herkomst: | IRS-NLR-006 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

| IDD-NLR-009: Implementatie Updates export interface (EIF4.2) | |
| --- | --- |
| Beschrijving: | De updates export interface maakt een zogenaamde Update aan voor een opgegeven periode. De Update bevat een verzameling van CSV update bestanden, die voldoen aan de [RFC4180] standaard.  Het formaat van de CSV bestanden is verder niet te specificeren en wordt als volgt ingevuld:  Een algemeen CSV update formaat:   * een headerregel, * separator “| “, * geen “double quoted” waarden, * referentiegegeven afhankelijke velden.   *(deze bestanden beginnen met de naam “srs\_”)*  Een CSV update formaat voor de oude BICS versie 4:   * specificaties conform [BICS4\_UPD].   *(deze bestanden beginnen met de naam “bics\_” en worden vanwege het uitfaseren van BICS versie 4 niet meer gebruikt)*  De CSV update bestanden worden als één verzameling verpakt aan de gebruiker ter download beschikbaar gesteld. |
| Toelichting: | Het belangrijkste verschil tussen de masterdump export (zie IDD-NLR-008) en de update export (zie IDD-NLR-009) bestaat uit:   * In het masterdumpbestand worden de actuele gegevens geëxporteerd. Is ook de historie opgevraagd, dan worden ook alle logisch verwijderde en vervangen referentiegegevens geëxporteerd (gegevens gemarkeerd als logisch verwijderd). * In het updatebestand worden de betreffende referentiegegevens altijd als mutaties geëxporteerd (het ‘wel of niet logisch verwijderd’ zijn van het gegeven, wordt aangegeven zoals ingevoerd).   De uitvoer wordt aangemaakt in de vorm van een zogenaamd (te downloaden) .zip bestand. Het zip-bestand bevat een verzameling van bestanden: een updatebestand per referentiegegeven (in 2 formaten) met daarin de updates voor dat gegeven in de opgegeven periode.  Omdat dit specifieke functionaliteit betreft die voor de NL Datamanager van belang is, om partnersystemen handmatig te kunnen voorzien van updates, is deze functionaliteit alleen toegankelijk voor de NL Datamanager. |
| Referentie: | - |
| Herkomst: | IRS-NLR-007 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

Hieronder enkele voorbeelden van een **NLRDMS exportbestand** (masterdump):

ERI locaties:

erased|lastmod|refcodeold|srscode|riscode|locode|fwcode|kmcode|type|name\_LOCAL|name\_NL|name\_DE|name\_FR|name\_EN|termcode|termtype|term\_LOCAL|term\_NL|term\_DE|term\_FR|term\_EN|quayfrom|quayto|lat|lon|version|source|ivs|ivscode|ivsname|ivsterm|ivsreport|exits

true|2000‑01‑23 08:00||NLRTM102K1CETEM00018|NLRTM102K15576700018|NLRTM|102K1|00018|1|ROTTERDAM 3E PETROHAVEN|ROTTERDAM 3E PETROHAVEN|ROTTERDAM 3E PETROHAVEN|ROTTERDAM 3E PETROHAVEN|ROTTERDAM 3E PETROHAVEN|CETEM|1|CETEM CONTAINERS|CETEM CONTAINERS|CETEM CONTAINERS|CETEM CONTAINERS|CETEM CONTAINERS|||51.87000000|4.31000000|1||VBS|2559943|RDAM 3 PETROHAVEN||false|1

false|2000‑09‑19 08:00||GBEXE||GBEXE|||8|EXETER|EXETER|EXETER|EXETER|EXETER||0||||||||50.70000000|‑3.53330000|1||IVS90|3261004|EXETER||true|2

true|2000‑09‑19 08:00||DEMTQ||DEMTQ|||1|METTEN|METTEN|METTEN|METTEN|METTEN||0||||||||48.85000000|12.90000000|1||IVS90|1938100|METTEN||true|1

Eerste meldpunten (als “quoted values”):

"erased"|"lastmod"|"refcodeold"|"reppoint"|"name\_LOCAL"|"name\_NL"|"name\_DE"|"name\_FR"|"name\_EN"|"version"|"source"|"RIScode"|"Providerid"

"false"|"2013‑12‑03 17:09"|""|"A99"|"IVS90 (NL)"|"IVS90 NEDERLAND (NL)"|"IVS90  (NL)"|"IVS90 (NL)"|"IVS90 (RIS NL)"|"1"|"BICS402"|"NLDOR001014538701166"|"IVS90"

"false"|"2013‑12‑03 17:09"|""|"F01"|"FRANCE VNF"|"FRANKRIJK VNF"|"FRANKREICH VNF"|"FRANCE VNF"|"FRANCE VNF"|"1"|"BICS402"|"FRBET000000000000000"|"VNF"

"false"|"2013‑12‑03 17:09"|""|"G20"|"HCC ROTTERDAM"|"HCC ROTTERDAM"|"HCC ROTTERDAM"|"HCC ROTTERDAM"|"HCC ROTTERDAM"|"1"|"BICS402"|"NLRTM010424683300011"|"VBS"

"true"|"2013‑12‑03 17:09"|""|"G20A"|"ERASMUSBRUG R'DAM"|"ERASMUSBRUG R'DAM"|"ERASMUSBRUCKE, R'DAM"|"ERASMUS, PONT, R'DAM"|"ERASMUSBRIDGE ROTTERDAM"|"1"|"BICS402"|"NLRTM010424683300011"|"VBS"

Hieronder enkele voorbeelden van een **NLRDMS update/mutatiebestand**:

ERI locaties updatebestand:

erased|lastmod|refcodeold|srscode|riscode|locode|fwcode|kmcode|type|name\_LOCAL|name\_NL|name\_DE|name\_FR|name\_EN|termcode|termtype|term\_LOCAL|term\_NL|term\_DE|term\_FR|term\_EN|quayfrom|quayto|lat|lon|version|source|ivs|ivscode|ivsname|ivsterm|ivsreport|exits

false|2012‑03‑23 15:07||DENSS00000NOTNE00000||DENSS|00000|00000|1|NEUSS|NEUSS|NEUSS|NEUSS|NEUSS|NOTNE|1|NEUSS TRIMODAL (OPTIMODAL)|NEUSS TRIMODAL (OPTIMODAL)|NEUSS TRIMODAL (OPTIMODAL)|NEUSS TRIMODAL (OPTIMODAL)|NEUSS TRIMODAL (OPTIMODAL)|||51.21560000|6.70950000|3|terbi20.04|IVS90|0930300|NEUSS|NEUSS INTERMODAL TERMINAL|true|0

true|2012‑05‑16 16:57|NLGRM001501382601402|NLGAV001501382601402|NLGRM001501382601400|NLGAV|00150|01402|1|KEENT (BENEDEN GRAVE)|KEENT (BENEDEN GRAVE)|KEENT (BENEDEN GRAVE)|KEENT (BENEDEN GRAVE)|KEENT (BENEDEN GRAVE)|13826|0||||||||51.76640000|5.69370000|2|ivs90 uit het veld|IVS90|3578600|GRAVE||false|1

## NLR-EIF5 NLRDMS Webservice - WSDL/Soap interface

### Interface Eisen



Figuur 4: EIF5 NLRDMS webservice interface t.b.v. gekoppelde cliënt systemen

De NLRDMS webservice interface voorziet in het extern (via het Internet) beschikbaar stellen van bepaalde diensten (operaties, functie aanroepen) aan applicaties van derden. De communicatie verloopt via het zgn. “Simple Object Access Protocol” (SOAP) en de data is gestructureerd op basis van XML.

De mogelijkheden van de interface worden gepubliceerd via een zgn. “Web Service Description Language” (WSDL), die in de context van en door NLRDMS beschikbaar wordt gesteld (zie [NLRDMS\_WSDL]). Verder wordt gebruik gemaakt van de SOAP stijl “Document/literal (wrapped)”, waarbij de functies, bijbehorende parameters en datastructuren in de webservice geheel zijn vastgelegd in een XSD.

Deze synchrone webservice interface wordt gebruikt door de gekoppelde clientsystemen, die ERI referentiegegevens willen ophalen bij NLRDMS (in de [OCD-NLREF] synchronisatieclients genoemd) ***of*** bevoegd zijn om NL referentiegegevens aan te leveren (in de [OCD-NLREF] synchronisatieservers genoemd).

Tevens wordt deze interface door de interne NLRDMS controlefunctie gebruikt voor het opvragen en controleren of de recente gemuteerde referentiegegevens in NLRDMS opvraagbaar zijn en daarbij voldoen aan het webservice return formaat. Mocht dit niet het geval zijn dan zal de mutatie weer op de aanvraaglijst voor de Databeheerder worden teruggezet (zie ook [SSS-NLRDMS]).

In Bijlage B zijn de (procedure) stappen beschreven, die partijen moeten doorlopen, om een koppeling met NLRDMS te implementeren voor het synchroniseren van referentiegegevens.

| IRS-NLR-010: NLRDMS webservice interface (EIF5), technische kenmerken | |
| --- | --- |
| Beschrijving: | NLRDMS stelt een synchrone webservice interface via het Internet beschikbaar voor client applicaties. De bijbehorende WSDL wordt in de context van NLRDMS gepubliceerd.  De interface biedt een SOAP-gebaseerde API (hierna NLRDMS SOAP API genoemd) aan, zoals beschreven in [SOAP].  De NLRDMS SOAP API is gebaseerd op de stijl “Document/literal (wrapped)”, waarbij de datastructuren zijn vastgelegd in een XSD.  De NLRDMS SOAP API kan (indien aangevraagd) de data gecomprimeerd uitwisselen, dit om het dataverkeer te kunnen optimaliseren.  De NLRDMS SOAP API wordt beschikbaar gesteld via http, HTTPS en de toegang is beveiligd met een generiek usernaam, wachtwoord. Aanvullend zijn voor bepaalde mutaties functies ook specifieke accountgegevens nodig.  Het standaardpoortnummer voor de NLRDMS SOAP API is poort 80 voor HTTP en 443 voor HTTPS.  De NLRDMS interface kent de volgende synchrone functies:   * Opvraag functie(s) voor het opvragen van gegevens (EIF5.1):   + Het opvragen van alle (actuele) referentiegegevens (table dumps).   + Het opvragen of er mutaties zijn (aantallen) in een bepaalde periode.   + Het opvragen van alle mutaties in een bepaalde periode.   *(geen NLRDMS gebruikersaccount nodig).*   * Aanlever functie(s) voor het aanleveren van gegevens (EIF5.2):   + Het aanleveren van de land (NL) specifieke mutaties, die afhankelijk van het soort gegeven na goedkeuring worden gepubliceerd en of worden geupload naar ERDMS.   (*NLRDMS gebruikersaccount nodig*).  Wat betreft het gebruik van deze interface ligt het initiatief altijd bij de (gekoppelde) client applicaties van NLRDMS. |
| Toelichting: | De NLRDMS SOAP API functie aanroepen verlopen synchroon, dat wil zeggen dat de aanroep van een functie direct door NLRDMS wordt beantwoord en het resultaat ook direct wordt teruggeven aan de aanroepende client (in dezelfde call). |
| Referentie: | - |
| Herkomst: | SSS-NLR-057 Externe interfaces, toegankelijkheid.  SSS-NLR-060 NL webservice interface kenmerken. |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

### Ontwerpbeslissingen

Voor de implementatie van deze NLRDMS webservice interface zijn de volgende aspecten van belang:

* De connectieparameters (WSDL, URL etc.).
* De beschikbare (synchrone) interface (API) functies in NLRDMS voor het ophalen van gegevens (opvraag functies) en het opsturen van mutaties (aanleverfuncties).

| IDD-NLR-011: Implementatie NLRDMS webservice interface (SOAP API) (EIF5) | |
| --- | --- |
| Beschrijving: | De NLRDMS SOAP API voorziet in operaties die synchroon worden afgehandeld. Deze SOAP API ondersteunt primair HTTP en optioneel HTTPS *en* gebruikt als toegangsbeveiliging HTTP Basic Authentication (te configureren in de applicatie- / webserver).  De SOAP API voorziet in gestructureerde functie aanroepen, parameters en de bijbehorende datastructuren worden conform een XSD (zie [NLRDMS\_SCHEMA]) gevalideerd.  De SOAP API is gebaseerd op de UTF-8 karakter set.  Standaard is de NLRDMS SOAP API **WSDL** beschikbaar op:  **{host}/refwsclient/v1/NLRefWebService.wsdl**  De **end-point** van de NLRDMS SOAP-server is beschikbaar op:  **{host}/refwsclient/v1/NLRefWebService**  op poort 80 (HTTP) en poort 443 (HTTPS).  Hierbij moet **{host}** worden vervangen door de hostnaam of het IP-adres van NLRDMS server, zie ook [NLRDMS\_WSDL].  Bijvoorbeeld Productiesysteem:  [*http://nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl*](http://nlrdms.rws.nl/refwsclient/v1/NLRefWebService.wsdl) *of*  De cliënt applicatie bepaalt wanneer deze door NLRDMS beschikbaar gestelde interface wordt aangesproken. |
| Toelichting: | Merk op dat het versienummer van de interface verwerkt is in de URL, zodat er meerdere versies naast elkaar actief kunnen zijn.  De NLRDMS webservice interface specificatie lijkt op die van de ERDMS (EIF6), maar verschilt wel degelijk op details.  De parameters en datastructuren van de interface zijn vastgelegd in een XSD, die via een <xsd: import … > statement in de WSDL wordt ingelezen/geïmporteerd.  De NLRDMS WSDL en bijbehorende XSD zijn via de opgegeven URL op te vragen en ook opgenomen in de bijlage van dit document (zie [NLRDMS\_WSDL] en [NLRDMS\_SCHEMA]).  Om bepaalde API functies te kunnen gebruiken zijn NLRDMS accountgegevens nodig (zie IDD-NLR-012). |
| Referentie: | [NLRDMS\_WSDL] |
| Herkomst: | IRS-NLR-010 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

In onderstaand overzicht zijn de belangrijkste SOAP API operaties (functies) en bijbehorende SOAP berichtenscenario’s weergegeven, zoals ook gedefinieerd in de NLRDMS WSDL (zie ook [NLRDMS\_WSDL]):



Figuur 5: NLRDMS SOAP API operaties, berichtenscenario overzicht

De opvraagfuncties leveren doorgaans als resultaat een datastructuur met de opgevraagde informatie (de gevraagde referentiegegevens).

*Elk referentiegegeven (record) heeft een codeOld veld, waarmee wordt verwezen naar de eventuele vorige code. Dat is met name van belang als de code (key) van dat gegeven is gewijzigd, het codeOld veld geeft dan een referentie aan naar het vervangen en dus vervallen record. Verder bevat elke referentiegegeven een (oplopend) versienummer dat per code begint bij 1.*

De aanleverfuncties leveren als resultaat een status op inclusief een mogelijke foutcode en melding.

De operaties in de NLRDMS webservice bestaan uit: 6 opvraag/opzoek functies en 2 aanleverfuncties (zie ook IDD-NLR-012).

| IDD-NLR-012: Implementatie NLRDMS webservice interface, API functies (EIF5) | |
| --- | --- |
| Beschrijving: | De NLRDMS SOAP API kent de volgende synchrone operaties (API functies), zoals gedefinieerd in de WSDL (zie IDD-NLR-011):  matchByCode() Zoek een bepaald referentiegegeven op basis van de (deel)code en geef en een lijst(t.b.v. bijvoorbeeld opzoeklijsten).  matchByName() Zoek een bepaald referentiegegevens op basis van een deel van de naam (t.b.v. bijvoorbeeld opzoeklijsten).  getMutatedTables() Bepaal (opvraag van) de gemuteerde tabellen en per tabel het aantal gemuteerde records (in een bepaalde periode).  getMutations() Vraag gemuteerde records in een bepaalde periode op (t.b.v. het synchroniseren van databases).  getDataXML() Vraag de referentiegegevens per tabel en of deelcode op (t.b.v. vullen van tabellen).  getRisDataXML() Vraag de RIS-index gegevens op, op deelcode en of functiecodes (t.b.v. vullen van tabellen).  mutateDataXML() Voer een mutatie op, die verwerkt zal worden in het NLRDMS, incl. doorzetten naar ERDMS (NLRDMS accountgegevens nodig).  requestMutationXML() Voer een mutatie aanvraag op voor de overige niet land specifieke referentiegegevens (NLRDMS accountgegevens nodig). *Niet geïmplementeerd in NLRDMS.*  De beschrijvingen van deze functies zijn verder uitgewerkt in paragraaf 3.6.3. |
| Toelichting: | Om gebruik te kunnen maken van de API functies is een generiek id/wachtwoord combinatie nodig (http basic authentication).  Om mutaties aan te kunnen leveren bij NLRDMS dient ook een geldig NLRDMS gebruikersaccount (id + wachtwoord) als parameter meegegeven te worden. Op basis van het account wordt de gebruiker geautoriseerd (bepaald of hij mutaties mag aanleveren) en wordt ook het aanleveren van de gegevens beperkt tot alleen de gegevens van zijn eigen land (in dit geval NL).  De benodigde toegangsgegevens worden beschikbaar gesteld door de systeembeheerder van het NLRDMS. Zie ook de bijlage voor de beschrijving van de (procedure) stappen om een cliëntsysteem aan te sluiten op NLRDMS NLRDMS\_CLIENT]. |
| Referentie: | [NLRDMS\_WSDL] |
| Herkomst: | IRS-NLR-010 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

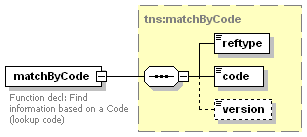
### NLRDMS webservice interface, API functiebeschrijvingen

De NLRDMS SOAP API bestaat uit diverse aan te roepen functies, zoals gedefinieerd in IDD-NLR-012), die in deze paragraaf verder worden uitgewerkt en beschreven, te weten:

* matchByCode(reftype, code, version), zie paragraaf 3.6.3.1.
* matchByName(reftype, lang, name, version), zie paragraaf 3.6.3.2.
* getMutatedTables(fromDate, toDate), zie paragraaf 3.6.3.3.
* getMutations(reftype, fromDate, toDate), zie pargaraaf 3.6.3.4.
* getDataXML(reftype, subcode, version), zie paragraaf 3.6.3.5.
* getRisDataXML(subcode, funcode, version), zie paragraaf 3.6.3.6.
* mutateDataXML(username, password, reftype, refData), zie paragraaf 3.6.3.7.
* requestMutationXML(username, password, reftype, inUserList, refData), zie paragraaf 3.6.3.8.

Alle functies van de NLRDMS webservice API kennen een synchroon karakter, waarbij de aanroep van de functie ook meteen het resultaat oplevert.

#### matchByCode functie



string [] matchByCode (

String reftype,

String code,

Long version)

Deze functie verzorgt het opvragen van een referentiegegeven op basis van een referentiecode. De gevonden gegevens (0 of meer) worden teruggegeven als een XML (choice) datastructuur van het complextype “refdataReturnType”.

De resultaat lijst (aantal teruggegeven records) wordt begrensd op 200 records. De “recCount” variabele in het resultaat geeft het totaal aantal beschikbare (gevonden) records aan, maar de resultaat lijst bevat daarvan slechts de eerste 200 records.

De functie “matchByCode()” is geschikt voor het opvragen van referentiecodes voor bijvoorbeeld opzoeklijsten.

Parameters:

reftype Referentiegegevens type, waarvan de gegevens moeten worden opgezocht.

De volgende waarden zijn mogelijk:

|  |  |
| --- | --- |
| ris\_idx | *The RIS Index (locations)* |
| eri\_location | *ERI (Ship Reporting Std) codes* |
| eri\_hscode | *Harmonised system codes (non dangerous)* |
| eri\_adncode | *Dangerous goods (ADN) codes* |
| eri\_conttype | *ISO container type codes* |
| eri\_packtype | *Inner package type codes (UN- rec 21)* |
| eri\_country | *Country codes* |
| eri\_shiptype | *Ship & transp. combination types (UN- rec 28)* |
| er\_erireportingpoint | *Electronic Reporting points (traffic centres)* |
| er\_eriprovider | *Electronic Reporting, providers (receiving systems)* |
| er\_locsreppoints  nts\_data | *Electronic Reporting,reporting points per location*  *Nts datatype* |
| nts\_barrage | *Nts barrage status codes* (not supported) |
| nts\_communication | *Nts communication means codes* (not supported) |
| nts\_country | *Nts country codes* (not supported) |
| nts\_direction | *Nts Fairway direction codes* (not supported) |
| nts\_direction\_max | *Nts compass direction codes* (not supported) |
| nts\_direction\_min | *Nts compass direction codes* (not supported) |
| nts\_ice\_accessibility | *Ice accessibility (status) codes* (not supported) |
| nts\_ice\_condition | *Ice condition (strong, weak) codes* (not supported) |
| nts\_ice\_classification | *Ice classification (navigatable) codes* (not supported) |
| nts\_ice\_situation | *Ice situation (navigation) codes.* (not supported) |
| nts\_indication | *Nts Indication (qualification) codes* (not supported) |
| nts\_interval | *Nts Time interval codes* (not supported) |
| nts\_language | *Nts language codes* (not supported) |
| nts\_limitation | *Nts fairway limitation codes* (not supported) |
| nts\_measure | *Nts (type of) measurement codes* (not supported) |
| nts\_position | *Nts fairway position code* (not supported) |
| nts\_reason | *Nts reason code* (not supported) |
| nts\_reference | *Nts (water)level reference code* (not supported) |
| nts\_regime | *Nts waterway regime code.* (not supported) |
| nts\_reporting | *Nts reporting regime* (not supported) |
| nts\_subject | *Nts subject code* (not supported) |
| nts\_tag | *Nts messages XML tag codes* (not supported) |
| nts\_target\_group | *Nts (ship) target group* (not supported) |
| nts\_type | *Nts Type of message* (not supported) |
| nts\_weather\_category | *Weather category* (not supported) |
| nts\_weather\_class | *Weather class* (not supported) |
| nts\_weather\_item | *Weather item message is about.* (not supported) |

code De code of subcode die wordt opgezocht.

Wildcards zijn mogelijk via het “%” en “\_” teken, te weten:

“%” (procent) = komt overeen met 1 of meer willekeurige karakters.

“\_” (onderstrepingsteken) = komt overeen met 1 willekeurig karakter.

*Bijvoorbeeld:*

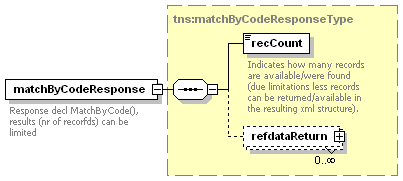
* + “NLRTM” = records moeten exact overeenkomen met de code.
  + “DEDUI%” = alle records beginnend met “DEDUI”,
  + “%00001” = alle records eindigend op “00001”.
  + “%RTM%” = alle records met “RTM” ergens in de code.

*(matching is hoofdletter ongevoelig)*

version Versienummer van het op te zoeken referentiegegeven, te weten:

* *Niet opgegeven* => selecteert alleen de actuele (alleen de *niet* verwijderde gegevens).
* “0” => selecteert alle versies van het gegeven (incl. de logisch verwijderde gegevens).
* *Overige waarden* (>0) => selecteert het gegeven met het opgegeven versienummer (indien beschikbaar).

Resultaat:



Het teruggegeven resultaat bestaat uit een XML structuur met de gevonden referentiegegevens, conform de datastructuur “refdataReturnType”.

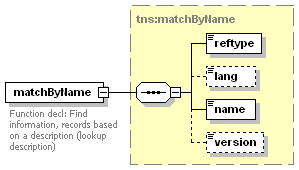
De beschrijving van de referentiegegevensvelden (in refdataReturn) en de betekenis kan worden teruggevonden in bijlage A3.

De “recCount” variabele geeft het aantal gevonden (in de database beschikbare) records aan. Het maximum aantal terug te geven records is, zoals eerder aangegeven, begrensd.

Als conform de opgegeven parameters geen referentiegegevens zijn gevonden, dan ontbreekt de “refdataReturn” structuur en bevat “recCount” de waarde 0.

In geval van fouten (verkeerde, onvoldoende parameters, interne fouten enz.) wordt een exceptie(fout) gegeven (in plaats van de XML structuur).

#### matchByName functie



string[] matchByName (

String reftype,

String lang,

String name,

Long version)

Deze functie verzorgt het opvragen van een referentiegegeven op basis van (een gedeelte van) een naam/beschrijving. De gevonden gegevens (0 of meer) worden teruggegeven als een XML (choice) datastructuur van het complextype “refdataReturnType”. De MatchByName() functie kan worden gebruikt om bepaalde sets van gegevens op te vragen op basis van (een gedeelte van) een naam of beschrijving.

De resultaat lijst (aantal teruggegeven records) word begrensd op 200 records. De “recCount” variabele in het resultaat geeft het totaal aantal beschikbare (gevonden) records aan, maar de resultaat lijst bevat daarvan slechts de eerste 200 records.

Parameters:

reftype Referentiegegevens type, waarvan de gegevens moeten worden opgezocht (zie matchByCode() voor mogelijke waarden).

lang Taal code van de beschrijving in de zoek-functie (gebaseerd op ISO 639-1).

*De ondersteunde taal codes* *zijn*:

“nl”=Nederlands, “fr”=Frans, “de”=Duits, “en”=Engels en “lc”=lokale naam, “bg”=Bulgaars, “cs”=Tsjechisch, “da”=Deens, “el”=Grieks, “es”=Spaans, “et”=Ests, “fi”=Fins, “hu”=Hongaars, “it”=Italiaans, “lt”=Litouws, “lv”=Lets, “pl”=Pools, “pt”=Portugees, “ro”=Roemeens, “sk”=Slovaaks, “sl”=Sloveens, “sv”=Zweeds, “hr”=Kroatisch, “ru”=Russisch, “sr”=Servisch.

*Speciale condities:*

* + *wanneer [lang] niet gespecificeerd is (de tag is niet aanwezig) dan zal op de lokale naam worden gezocht.*
  + *Wanneer bij [lang] een taalcode is ingevoerd die niet wordt ondersteund door het opgevraagde referentie type dan zal een foutmelding worden gegenereerd.*
  + *Wanneer [lang] niet is ingevuld (een lege string) dan zal er een foutmelding worden gegenereerd*
  + *Wanneer bij [lang] de taalcode lc is ingevoerd dan zal alleen op de code in de lokale namen/beschrijvingen worden gezocht.*
  + *Wanneer bij [lang] “%%” is ingevoerd dan zal er op alle taal velden in de naam / beschrijvingen worden gezocht.*

name De beschrijving of deel van de beschrijving waarop gezocht moet gaan worden in de referentie data. Indien het referentie type over andere aanvullende taalvelden beschikt (b.v. terminal naam voor locations, synoniemen voor ADN goederen enz), dan zullen deze velden ook worden doorzocht.

Invoer van de wildcard “%” is beschikbaar/mogelijk voor gebruik, bijv.:

- “ROTTERDAM” = het resultaat is alleen records die exact het woord ROTTERDAM hebben in de naam

- “OBER%” = geeft als resultaat alle records die beginnen met “OBER”,

- “%DAM” = geeft als resultaat alle records die eindigen met “DAM”.

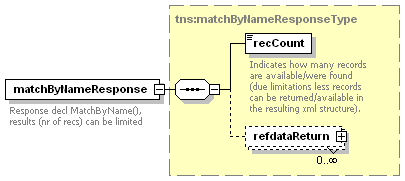
- “%WAAL%” = geeft als resultaat alle records die ergens in de naam de combinatie “WAAL” bevatten.

*(het overeenkomen is niet hoofdlettergevoelig).*

version Aanvraag van de data met een specifiek versie nummer.  
Wanneer bij ‘version’ een “0” is ingevoerd dan zullen alle versies van de opgevraagde records worden getoond, ook de records die zijn verwijderd.

Wanneer ‘version’ niet is gespecificeerd dan zullen alleen de actuele (geaccepteerde) records worden weergegeven (de laatste versie) Wanneer gezocht wordt op een specifiek versienummer dan zal de gespecificeerde versie van het record worden getoond (zelfs wanneer deze versie van het record de status verwijderd heeft).

Resultaat:



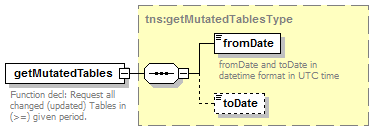
XML structuur van het resultaat = de data structuur van “refdataReturnType”.

Het resultaat geeft een lege structuur wanneer er geen records zijn gevonden.

De beschrijving van de referentiegegevensvelden (in refdataReturn) en de betekenis kan worden teruggevonden in bijlage A3.

Wanneer er een foutmelding wordt geconstateerd (ongeldige parameters, interne fouten) dan zal er een exceptie worden gegenereerd (zie bijlage A4 voor de beschrijving van de lijst van mogelijke fouten).

#### getMutatedTables functie



string[] getMutatedTables (

DateTime fromDate, toDate)

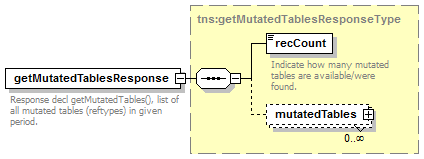
Deze functie aanvraag levert een lijst met alle referentietypes waar mutaties/wijzigingen voor zijn geweest in de opgevraagde periode. Door deze functie te gebruiken wordt duidelijk welke referentie tabellen precies zijn gewijzigd. Zodat selectief de mutaties/wijzigingen van specifieke referentie tabellen kunnen worden opgevraagd (types van referentie data).

Parameters:

fromDate,

toDate De datum parameters die de opgevraagde periode van de aanvraag weergeeft, inclusief “van” en “tot” datum/tijd. Er wordt geselecteerd op het veld “last update” = mutatie/wijzig datum van het record. De datum parameters moeten worden gespecificeerd in datum/tijd formaat in UTC tijd. Wanneer het veld “toDate” niet gespecificeerd is zal de huidige datum/tijd als default datum tot worden gebruikt.

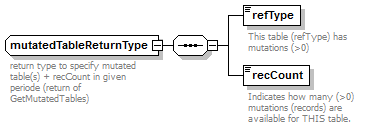
Resultaat:



XML structuur van het resultaat = de data structuur van “mutatedTables”.

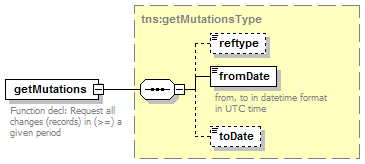
Het resultaat geeft een lege structuur wanneer er geen records zijn gevonden.

De structuur van “mutatedTables” geeft per referentie tabel (reftype) de aantallen van de beschikbare updates/wijzigingen weer (recCount):



Wanneer er een fout wordt geconstateerd (ongeldige parameters, interne fouten) dan zal er een exceptie worden gegenereerd. Zie bijlage A4 voor de beschrijving van de lijst van mogelijke fouten.

#### getMutations functie



string[] getMutations (

String reftype,

DateTime fromDate, toDate)

Deze functie haalt over een bepaalde op te geven periode alle gemuteerde informatie binnen. Met behulp van deze functie kunnen alle laatste updates (verwijderde, gewijzigde, toegevoegde records) worden opgehaald om de data in de ‘eigen’ database te kunnen synchroniseren met NLRDMS database.

Naast het opvragen van de mutaties in een bepaalde periode is er ook de mogelijkheid een gehele (actuele) set op te halen via de getDataXML() functie.

Voor de teruggegeven mutatieset in de getMutatationResponse geldt:

* De mutaties worden in een bepaalde chronologische volgorde teruggegeven, die samen met het beschikbare versienummer van belang is voor de verwerking. De volgorde in het resultaat is gebaseerd op de mutatiedatum + de volgorde van muteren in NLRDMS.
* Verwijderde records (mutaties) worden aangegeven met een “erased” waarde is ‘true’.
* Gewijzigde records (mutaties) kunnen worden herkend door een ingevulde waarde voor het veld “codeOld”. Het “codeOld” veld kan daarbij ook verwijzen naar een eventueel gewijzigde en vervangen (andere) code. D.w.z. dat in dat geval de oude code is vervallen en wordt vervangen door een nieuwe. In de gesynchroniseerde set dient de oude code dan als verwijderd gemarkeerd te worden.
* Een nieuw ingevoerd record (mutatie) kan worden herkend door een “codeOld” veld dat geen waarde heeft (leeg).
* Het versie nummer van een record specificeert de volgorde van de ingevoerde wijzigingen (mutaties per gelijke code) en is van belang bij het verwerken van de mutatie. De hoogste versie is de meest recente versie. Voor elke (nieuwe) referentiecode begint de waarde weer bij 1.

Hieronder volgen enkele voorbeelden voor de locatie referentiegegevens (eri\_location), die het gebruik van de getMutations resultaten illustreren.

Voorbeeld #1: code toevoegen en wijzigen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| lastupdate | erased | erilocCode | codeOld | version | Verwerkingsacties |
| 2021-06-22 11:21:59 | false | NLVEN0150P00ECT00014 |  | 1 | NLVEN0150P00ECT00014 wordt nieuw toegevoegd (geen codeOld, versie=1). |
| 2021-06-22 11:21:59 | false | NLVEN0150P00ECT00014 | NLVEN0150P00ECT00014 | 2 | Record NLVEN0150P00ECT00014 wordt gewijzigd. |

Voorbeeld #2: mutaties incl. verwijdering en code wijziging

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| lastupdate | erased | erilocCode | codeOld | version | Verwerkingsacties |
| 2021-06-01 11:22:30 | false | NLAMS00233RK17A00223 | NLAMS00233RK17A00223 | 7 | Record NLAMS00233RK17A00223 wordt gewijzigd. |
| 2021-06-22 08:13:22 | false | NLAMS00233RK17A00223 | NLAMS00233RK17A00223 | 6 | Mutatie voor NLAMS00233RK17A00223 wordt genegeerd (versie 6 < 7). |
| 2021-06-22 08:13:22 | true | NLAMS00233RK17A00223 | NLAMS00233RK17A00223 | 7 | NLAMS00233RK17A00223 wordt verwijderd (erased=true) |
| 2021-06-22 08:13:22 | false | NLAMS00233RK17A00220 | NLAMS00233RK17A00223 | 1 | NLAMS00233RK17A00223 wordt **vervangen** door NLAMS00233RK17A00220.  D.w.z. dat mocht NLAMS00233RK17A00223 bestaan, deze oude code wordt vervangen en als verwijderd moet worden gemarkeerd. |

Voorbeeld #3: code wijziging

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| lastupdate | erased | erilocCode | codeOld | version | Verwerkingsacties |
| 2021-06-22 08:13:22 | false | NLAMS00233RK17A00223 | NLAMS00233RK17A00220 | 1 | Code NLAMS00233RK17A00220 wordt gewijzigd in NLAMS00233RK17A00223.  De oude code NLAMS00233RK17A00220 wordt vervangen en moet worden verwijderd. Code NLAMS00233RK17A00223 wordt vervolgens nieuw toegevoegd. |

Belangrijke punten bij de verwerking van de getMutations resultaten zijn:

* De mutaties dienen in de volgorde verwerkt te worden, zoals ze in het resultaat worden teruggegeven.
* Het versienummer speelt per gelijke code een rol in de verwerking. Mutaties met een versienummer < huidige worden genegeerd, behalve bij een code wijzigingen (codeOld <> code), want dan begint het versienummer voor de nieuwe code bij 1.
* Het codeOld veld is van belang bij de verwerking, met name als het een gewijzigde code betreft, dan is de oude code vervangen en dient deze ook als verwijderd gemarkeerd te worden in de gesynchroniseerde set.
* Vanzelfsprekend dienen alle mutaties steeds in aaneengesloten periodes verwerkt te worden. Beginnend bij de meest recente mutatiedatum (lastupdate) van het te synchroniseren gegeven.

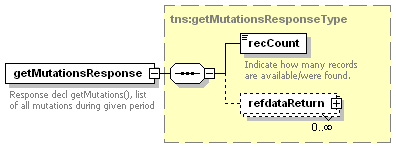
Parameters:

reftype Referentiegegevens type, waarvan de gegevens moeten worden opgezocht (zie matchByCode() voor mogelijke waarden).

fromDate,

toDate De datum parameters die de opgevraagde periode van de aanvraag weergeeft, inclusief “van” en “tot” datum/tijd. Er wordt geselecteerd op het veld last update = mutatie/wijzig datum van het record. De datum parameters moeten worden gespecifieerd in datum/tijd formaat in UTC tijd. Wanneer het veld “toDate” niet gespecificeerd is zal de huidige datum/tijd als default worden gebruikt.

Result:



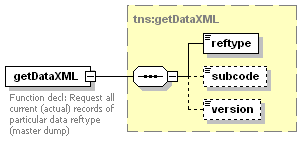
XML structuur van het resultaat = de data structuur van “refdataReturnType”.

Het resultaat geeft een lege structuur wanneer er geen records zijn gevonden.

De beschrijving van de referentiegegevensvelden (in refdataReturn) en de betekenis kan worden teruggevonden in bijlage A3.

Wanneer er een fout wordt geconstateerd (ongeldige parameters, interne fouten) dan zal er een exceptie/fout worden gegenereerd (zie bijlage A4 voor de beschrijving van de lijst van mogelijke fouten).

#### getDataXML functie



string[] getDataXML (

String reftype,

String subcode

Long version)

Deze functie haalt alle of een gedeelte van de op dit moment actuele records van een bepaald referentie type op (een complete masterdump van de referentie tabel)

Deze functie zorgt ervoor dat een complete tabel of een gedeelte van een tabel kan worden gedownload zodat met behulp daarvan een database kan worden gevuld.

*(Het verschil tussen de functies MatchByCode() en getDataXML() zit in het feit dat bij de MatchByCode functie het resultaat begrensd is en het resultaat van getDataXML niet begrensd is.*

Parameters:

reftype Referentiegegevens type, waarvan de gegevens moeten worden opgezocht (zie matchByCode() voor mogelijke waarden).

subcode Subcode om de gewenste subset op te vragen.

Wildcards “%” zijn beschikbaar voor gebruik, bijv.:

- “NLRTM” = het resultaat is alleen records die exact het woord NLRTM hebben als code

- “DEDUI%” = geeft als resultaat alle records die beginnen met “DEDUI”,

- “%00001” = geeft als resultaat alle records die eindigen met “00001”.

- “%RTM%” = geeft als resultaat alle records die ergens in de naam de combinatie “RTM” bevatten.

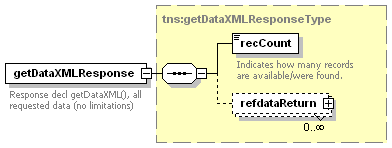
Wanneer de code leeg of niet gespecificeerd is dan zullen alle records worden geselecteerd (geen filter)

*(het overeenkomen/matchen is niet hoofdlettergevoelig).*

version Aanvraag van de data met een specifiek versie nummer.   
Wanneer bij ‘version’ een “0” is ingevoerd dan zullen alle versies van de opgevraagde records worden getoond, ook de records die zijn verwijderd.

Wanneer ‘version’ niet is gespecificeerd dan zullen alleen de actuele (geaccepteerde) records worden weergegeven (de laatste versie) Wanneer gezocht wordt op een specifiek versienummer dan zal de gespecificeerde versie van het record worden getoond (zelfs wanneer deze versie van het record de status verwijderd heeft).

Resultaat:



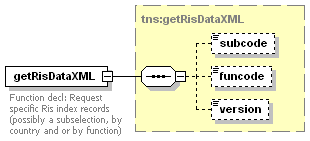
XML structuur van het resultaat = de data structuur van “refdataReturnType”.

Het resultaat geeft een lege structuur wanneer er geen records zijn gevonden.

De beschrijving van de referentiegegevensvelden (in refdataReturn) en de betekenis kan worden teruggevonden in bijlage A3.

Wanneer er een fout wordt geconstateerd (ongeldige parameters, interne fouten) dan zal er een exceptie/fout worden gegenereerd (zie bijlage A4 voor de beschrijving van de lijst van mogelijke fouten).

#### getRisDataXML functie



string[] getRisDataXML

String subcode

String funcode

Long version)

De getRisDataXML functie verzorgt het ophalen van alle of een deel van de RIS Index data. Deze functionaliteit maakt het mogelijk om van een specifiek land de RIS index data op te halen. Of om van een specifieke functie code de RIS index gegevens op te halen.

Deze functie maakt het mogelijk om de gehele RIS index tabel of een gedeelte van de gegevens van de RIS index tabel op te halen zodat er een eigen database kan worden opgebouwd.

H*et verschil tussen getRisDataXML() en getDataXML() is dat met de GetRisDataXML functie specifieke subsets (bijvoorbeeld op een functiecode) van de RIS Index kunnen worden opgevraagd, terwijl getDataXML een generieke functie is, waarbij alleen een subset kan worden opgevraagd die gebaseerd is op een code (wildcard).*

Parameters:

subcode RIS index sub code waar op moet worden gezocht. Alleen de records met een risindex code die aan deze code voldoet worden getoond. (b.v. subcode=“NL%” zal als resultaat een dataset leveren van de Nederlandse RIS index en “NLRTM%” zal resulteren in een dataset van Rotterdam).

Wildcards “%” zijn beschikbaar voor gebruik, bijv.:

- “NLRTM” = het resultaat is alleen records die exact het woord NLRTM hebben als code

- “DEDUI%” = geeft als resultaat alle records die beginnen met “DEDUI”,

- “%00001” = geeft als resultaat alle records die eindigen met “00001”.

- “%RTM%” = geeft als resultaat alle records die ergens in de naam de combinatie “RTM” bevatten.

Wanneer de code leeg of niet gespecificeerd is dan zullen alle records worden geselecteerd (geen filter)

*(het overeenkomen is niet hoofdlettergevoelig).*

Funcode RIS index functie code (Wildcards “%” zijn mogelijk). Voor het ophalen van RIS index records met een specifieke functie, b.v. alleen ankerplaatsen (berths).

In combinatie met de subcode parameter kunnen bijvoorbeeld alle ankerplaatsen voor een specifiek land worden opgevraagd (subcode=”NL%”, funcode=”berths%”). De mogelijke functie codes zijn opgesteld volgens de RIS index standard (zie hiervoor RIS encoding Guide).

Een lege string zal records leveren ongeacht de RIS function code.

*(het overeenkomen is niet hoofdlettergevoelig).*

Een aantal funcode voorbeelden:

-“berths\_%” = alle ankerplaatsen

-“trmnl” = alle terminals.

-“lok%” = identificatie van sluiscomplex met zijn onderdelen.

-“hrb%” = identificatie van haven gebied met zijn onderdelen.

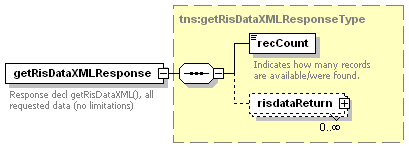
-“junctions” = knooppunten.

-“wtwgag” = peilschaal

Version Aanvraag van de data met een specifiek versie nummer.   
Wanneer bij ‘version’ een “0” is ingevoerd dan zullen alle versies van de opgevraagde records worden getoond, ook de records die zijn verwijderd.

Wanneer ‘version’ niet is gespecificeerd dan zullen alleen de actuele (geaccepteerde) records worden weergegeven (de laatste versie) Wanneer gezocht wordt op een specifiek versienummer dan zal de gespecificeerde versie van het record worden getoond (zelfs wanneer deze versie van het record de status verwijderd heeft).

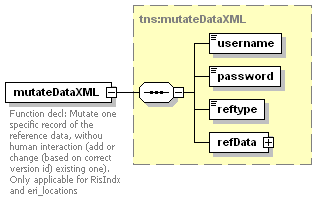
Resultaat:



XML structuur van het resultaat = de data structuur van “risdataReturnType” (ris\_idxType).

Het resultaat geeft een lege structuur wanneer er geen records zijn gevonden.

#### mutateDataXML functie



string[] mutateDataXML(

String user, passw,

String reftype,

String refData)

Deze functie verwerkt één (1) mutatie van een specifiek referentie type. Indien de aangeboden mutatie voldoet aan de verplichte geldigheids-eisen van NLRDMS dan zal de mutatie direct worden verwerkt en gepubliceerd. Wanneer de aangeboden mutatie incompleet of ongeldig is dan zal de mutatie worden geweigerd en zal de functie een foutmelding teruggeven.

Voor de autorisatie is de combinatie gebruikersnaam en wachtwoord (een geldig en geactiveerd NLRDMS gebruikersaccount) vereist. De gebruikersnaam wordt tevens gebruikt om te bepalen voor welk land de gebruiker geautoriseerd is om mutaties aan te leveren. Het thuisland van de gebruiker moet overeenkomen met de landcode van de mutatie (voor NLRDMS gebruikers is dat NL), anders wordt de mutatie bij voorbaat geweigerd. (rejected)

Deze functie kan alleen worden gebruikt voor land specifieke data zoals mutaties van de referentie types RIS index en ERI locations. Daarbij komt dat de aangeleverde data moet behoren bij het thuis land van de gebruiker die de mutatie aanvraagt.

Functie aanvragen van andere types van referentie data (niet land specifiek) zullen worden afgekeurd en er zal een foutmelding worden teruggeven.

Mutaties op alle andere types van referentie data moeten worden aangevraagd door gebruik te maken van de requestMutationXML() de validatie en verwerking van deze aanvragen zal worden uitgevoerd door de Europese Datamanager.

Afhankelijk van het referentietype zal voor de invoer de juiste referentie data structuur (XML) moeten worden gebruikt. De verplichte velden in de referentie data XML structuur moeten verplicht worden ingevuld anders wordt de mutatie-aanvraag afgekeurd.

Het versie element speelt binnen de referentie data structuur tevens een belangrijke rol. Wanneer het versienummer van de mutatie aanvraag verschilt van het versienummer van het al bestaande actuele record wat moet worden gewijzigd dan zal de mutatie aanvraag worden afgekeurd om te voorkomen dat er wijzigingen op de data worden aangebracht die zijn gebaseerd op oudere / vorige versies van de data. Wanneer het versienummer niet juist is dient de applicatie die de mutatie aanvraagt eerst de laatste / actuele versie van het record aan te vragen. Het versienummer van een mutatie record van een nieuw nog niet bestaand record moet de waarde 1 hebben.

De basis van de UN/Locodes (de eerste 5 posities van de “ris\_idx” en / of “eri\_location” code) van nieuwe locaties die de aanvrager wenst toe te voegen, dienen al in de database aanwezig te zijn. De basis UN/Locode (de eerste 5 posities) moet al aanwezig zijn in de “eri\_location” tabel. Wanneer dat niet het geval is, dan zal dat worden gemeld bij het opslaan van de mutatie. Het gegeven kan namelijk verderop in de keten (upload naar EU) hierop worden afgekeurd. Het is aan te bevelen om de basis UN/Locode eerst toe te voegen of deze te corrigeren. Dit waarschuwingsmechanisme is geïmplementeerd om te voorkomen dat mutaties worden toegevoegd aan de Centrale EU-database op basis van niet-bestaande (niet officieel geaccepteerde) UN/Locodes.

In het geval dat er een diakritisch karakter in de aangeleverde data voorkomt, dan wordt deze tijdens het goedkeuren en de publicatie van de mutatie (indien conversie is geconfigureerd) vervangen door het corresponderende niet-diakritische (simpele) karakter. Voor het vervangen van karakters wordt gebruik gemaakt van een door de systeembeheerder beheerde diakrieten conversietabel. Diakritische tekens zijn alleen in de zogenaamde ‘nationale’ naamvelden toegestaan. In die velden worden ze niet vervangen.

Alle acties van mutaties zullen worden gelogd en zijn gekoppeld aan de gebruikersnaam van de aanvrager van de mutatie zodat alle wijzigingen op de data zijn terug te herleiden.

Parameters:

gebruikersnaam,

wachtwoord Gebruikersnaam en wachtwoord van de RIS Data Management Service. De gebruiker moet een geldig en geactiveerd account gebruiken. Het account wordt gebruikt voor autorisatie en om te beoordelen wat het thuisland is van de gebruiker om te kunnen controleren of de aangevraagde mutaties wel zijn toegestaan.

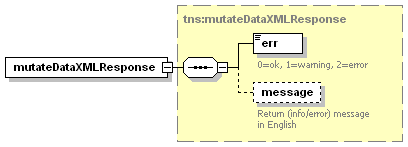
reftype Referentiegegevens type, waarvan de gegevens moeten worden opgezocht (zie matchByCode() voor mogelijke waarden).

*(****alleen*** *de land-afhankelijk data:* ***ris\_idx*** *en* ***eri\_locations*** *zijn toegestaan).*

refData Data (XML structuur) inhoudelijke referentie (**slechts een record**). de XML structuur is afhankelijk van het referentie type en is gebaseerd op “refdataReturnType”.

De beschrijving van de referentiegegevensvelden (in refdataReturn) en de betekenis kan worden teruggevonden in bijlage A3.

Resultaat:

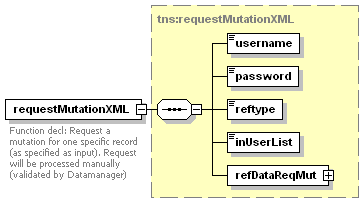


Deze functie geeft een status terug die gebaseerd is op de structuur van “statReturnType”.

Wanneer de mutatie voldoet aan de invoereisen *en* kan worden ingelezen, verwerkt en gepubliceerd dan zal de functie als resultaat OK teruggeven. In andere gevallen, zal er een warning/error status (waarschuwing/fout) terug worden gemeld met daarbij een Engelstalige beschrijving van de waarschuwing/fout.

Er zal een foutmelding (exceptie) worden teruggegeven wanneer er een fout optreedt. Zie bijlage A4 voor de beschrijving van de mogelijke fouten.

#### requestMutationXML functie



string[] mutateDataXML(

String user, passw,

String reftype,

Boolean inUserList,

String refDataReqMut)

**Deze functie is (nog) niet te gebruiken, omdat het uploaden van dit type aanvraag naar de Europese database (ERDMS) niet wordt ondersteund door de NLRDMS applicatie.**

De functie heeft als taak om een (1) mutatie aanvraag van een bepaald reftype te verwerken. De verwerking houdt in dat de mutaties interactief moet worden gevalideerd en verwerkt. De aanvraag zal in de gebruikers mutatie lijst moeten worden geplaatst (inUserList=true) **of** direct zichtbaar worden in de aangevraagde mutatie lijst van de NL Data manager / NL RIS Admin focal punt. Die dan vervolgens de mutatie dient te valideren en handmatig dient te verwerken. De mutaties die met behulp van deze functie worden aangevraagd moeten altijd handmatig worden gevalideerd (menselijke handeling is nodig).

De combinatie gebruikersnaam, wachtwoord (een geldig en geactiveerd NLRDMS gebruikersaccount) is nodig voor de autorisatie en om het thuis land van de gebruiker te kunnen bepalen. Voor ieder land is in ieder geval 1 gebruikersaccount beschikbaar in ERDMS.

De refdata structuur (XML) is afhankelijk van het gebruikte referentietype en deze structuur moet worden gebruikt om de invoer correct te laten verlopen. Al de verplichte velden in de “refdataReqMut” XML dienen te worden ingevuld. De set van verplichte velden is kleiner dan de set die is gebruikt voor de functie MutateDataXML(). In de basis dienen alleen de code en de beschrijving te worden ingevuld. De overige velden kunnen worden ingevuld maar dat is optioneel (de data manager zal de set indien nodig aanvullen en compleet maken.

In het geval dat er een diakritisch karakter in de aangeleverde data voorkomt, dan wordt deze tijdens het goedkeuren en de publicatie van de mutatie (indien de conversie is geconfigureerd) vervangen door het corresponderende niet-diakritische (simpele) karakter. Voor het vervangen van karakters wordt gebruik gemaakt van een door de systeembeheerder beheerde diakrieten conversietabel. Diakritische tekens zijn alleen in de zogenaamde ‘nationale’ naamvelden toegestaan. In die velden worden ze niet vervangen.

Alle acties van mutaties zullen worden gelogd en zijn gekoppeld aan de gebruikersnaam van de aanvrager van de mutatie zodat alle wijzigingen op de data zijn terug te herleiden.

Parameters:

gebruikersnaam,

wachtwoord Gebruikersnaam en wachtwoord van de RIS Data Management Service. De gebruiker moet een geldig en geactiveerd account gebruiken. Het account wordt gebruikt voor autorisatie en om te beoordelen wat het thuisland is van de gebruiker om te kunnen controleren of de aangevraagde mutaties wel zijn toegestaan.

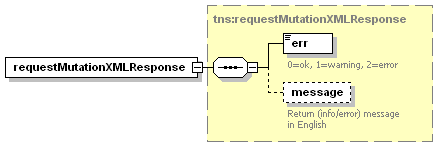
reftype Referentiegegevens type, waarvan de gegevens moeten worden opgezocht (zie matchByCode() voor mogelijke waarden). Het aanvragen van mutaties van NtS data is niet mogelijk.

inUserList Indien *true:* de mutatie aanvraag krijgt de status nieuw en zal bij de gebruiker in de nieuwe mutatie lijst getoond worden, in plaats van in de lijst van de aanvragen van de data manager (status requested). De gebruiker kan op deze manier de mutatie nog wijzigen en aanvullen (met behulp van de RIS Data Management Service GUI), voordat de aanvraag wordt doorgestuurd naar de Data manager.

Indien *false:* de mutatie aanvraag wordt direct in de Data manager aanvraag lijst geplaatst (status requested) en de Data manager dient de aanvraag te verwerken.

refDataReqMut Data (XML) bevat de inhoudelijke mutatie (**slechts een record**). XML structuur is afhankelijk van het referentie type en is gebaseerd op “refdataReqMutType”.

Resultaat:



Deze functie geeft een status terug die gebaseerd is op de structuur van “statReturnType”.

Wanneer de mutatie voldoet aan de invoereisen en kan worden ingelezen, verwerkt en gepubliceerd in NLRDMS, dan zal de functie als resultaat OK teruggeven. In andere gevallen, zal er een warning status (waarschuwing) terug worden gemeld met daarbij een Engelstalige beschrijving van de waarschuwing/fout.

*Wanneer de aanvraagde mutatie in een later stadium door de NL datamanager wordt geweigerd (rejected) dan zal de gebruikelijke manier van afhandeling (via de GUI) bij een afgekeurde mutatie door het RIS Data Management Service worden doorlopen. De gebruiker zal in dit geval de afgekeurde mutaties in het home gebruikers scherm terugvinden (na inloggen) en kan het van daaruitverder afhandelen.*

Er zal een foutmelding (exceptie) worden teruggegeven wanneer er een fout optreedt. Zie bijlage A4 voor de beschrijving van de mogelijke fouten.

### NLRDMS Webservice interface, API datastructuren

In deze paragraaf worden de belangrijkste (ComplexType) datastructuren beschreven van de webservice interface, zoals:

* *refdataReturnType*: een datastructuur voor het kunnen teruggeven van een (opgevraagd) referentiegegeven.
* *refdataReqMutType*: een datastructuur voor het kunnen opgeven van een (aan te leveren) referentiegegeven.
* *mutatedTableReturnType*: een datastructuur voor het kunnen teruggegeven van het aantal gemuteerde tabellen en records.
* *statReturnType*: een datastructuur voor het kunnen teruggeven van status informatie.

#### refdataReturnType

Het “refdataReturntype (complexType) bevat van de verschillende types van referentie data de gehele (choice) data structuur zoals deze gebruikt worden in RIS Data Management Service. Alle velden waarvan je gedwongen wordt om ze in te vullen (ook in de onderliggende database verplicht) zijn als verplicht gemarkeerd. De structuur geeft een complete weergave van het referentie data (refdata) record (alle velden die kunnen worden ingevuld zijn ingevuld).

De beschrijving van de referentiegegevensvelden (in *refdataReturn*) en de betekenis kan worden teruggevonden in bijlage A3. Deze structuur wordt voornamelijk gebruikt om de referentie data records weer te geven als antwoord op een informatie aanvraag functie zoals “matchByCode()”, “matchByName()”, “getDataXML()”, “getMutations()”. Daarbuiten wordt deze (afgedwongen) structuur ook gebruikt om de mutaties voor de “mutateDatXML()” functie te specificeren.

Het Referentie data type wordt gespecificeerd door het “reftype” element (dit is een enumerated type dat het referentie data type benoemd/weergeeft. Bijvoorbeeld, welke sectie van de keuze structuur ingevuld dienen te worden. Zie de functie “matchByCode()” voor de mogelijke waardes.

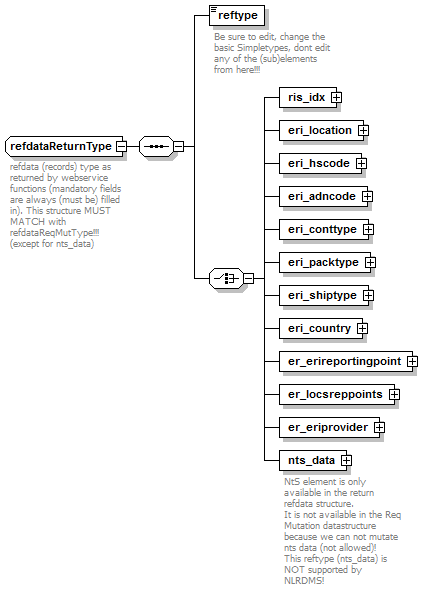
Belangrijke velden in de referentiedata structuur zijn:

* lastupdate: mutatiedatum.
* …code: de unieke referentiecode voor dat gegeven.
* codeOld: de oude eventueel vervangen code (belangrijk bij code wijzigingen).
* version: oplopend versienummer, beginnend bij 1 (per gelijke code).
* erased: true = gegeven is logisch verwijderd.

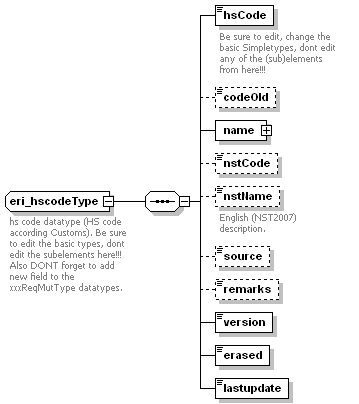
**Belangrijk**: De meeste referentie data bevat velden voor de verschillende talen die in de Europese Binnenvaart worden gebruikt. (maar niet alle talen worden ondersteund).

Zorg ervoor dat deze gegevensstructuur altijd hetzelfde blijft als "refdataReqMutType" structuur, met name bij het toevoegen of verwijderen van velden/elementen. Gebruik/definieer voor ieder element altijd het (basis) eenvoudige type. Zodat de elementen in beide structuren zijn gebaseerd op dezelfde data types.

*(Wees ervan bewust dat de keuze structuur "nts\_data" is opgenomen in de "refdataReturnType" maar niet in "refdataReqMutType", de rest van de keuze-structuur is/moet overeenkomen)*



Onderstaand een voorbeeld van een substructuur binnen “refdataReturnType”:



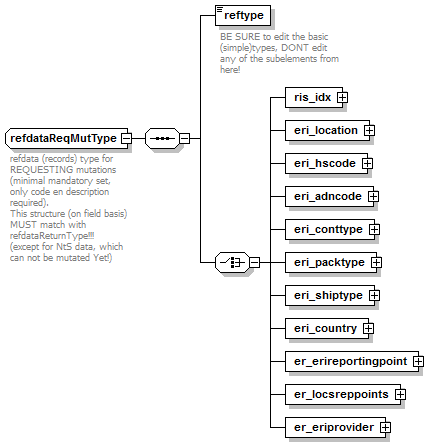
#### refdataReqMutType

Het “refdataReqMutType” bevat een complete (keuze) data structuur voor de referentie data zoals gebruikt in de NL RIS DataManagement Service voor het aanvragen van mutaties. De data structuur heeft (verplicht) de zelfde structuur als “refdataReturnType”, behalve voor NtS data. Echter de meeste velden zijn optioneel in deze structuur (“refdataReqMutType” is niet zo dwingend als refdataReturnType”).

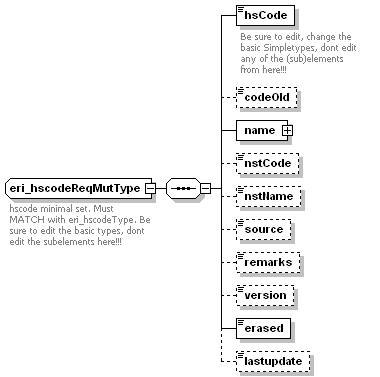
Zorg ervoor dat deze gegevensstructuur hetzelfde blijft als "refdataReturnType", met name bij het toevoegen of verwijderen van velden/elementen. Gebruik/definieer voor ieder element altijd het (basis) eenvoudige type. Zodat de elementen in beide structuren zijn gebaseerd op dezelfde data types.

*(Wees ervan bewust dat de keuze structuur "nts\_data" niet is opgenomen in “refdataReqMutType”, omdat NtS referentie gegevens niet kunnen worden gewijzigd/gemuteerd volgens de gebruikelijke geldende procedures.)*

Deze structuur wordt gebruikt als invoer voor het opvragen referentie mutatie records zoals in de functie: “requestMutationXML()”.

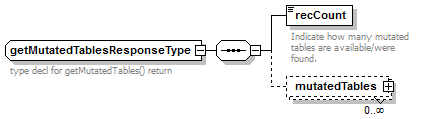


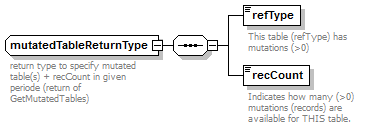
Onderstaand een voorbeeld van de keuze (sub)structuren binnen “refdataReqMutType”:



#### mutatedTableReturnType

Het "mutatedTableReturnType" bevat een resultaat structuur die wordt gebruikt om gemuteerde tabellen weer te geven. (tabellen = refdata met gemuteerde records in de opgegeven periode). Zie de functie: "getMutatedTables()". Zie de functie: “getMutatedTables()”.



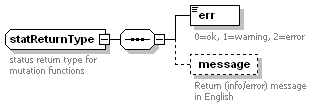


#### statReturnType

Het “statReturnType” bevat een (status) resultaat. Dit type wordt gebruikt voor het weergeven/teruggeven van een return status.

De status kan de waarde OK, waarschuwing of foutmelding krijgen. Deze structuur wordt gebruikt voor een status resultaat.

Wanneer er op de referentie gegevens een mutatie aanvraag wordt uitgevoerd, dan wordt er een antwoord gegeven in de structuur van statReturnType. *Bijvoorbeeld* als antwoord op de aanroepende functie: "requestMutationXML()"



Het resultaat van de returntype structuur zal zijn err=0 als de aanroep van de functie resulteert in een ok, de aanvraag is als verwacht verwerkt.

Wanneer er bij de functieaanroep een fout is opgetreden dan resulteert dit in de status err=2. Tevens wordt er dan ook een Engelse fouttekst (melding) teruggegeven.

Wanneer de aanvraag wel volgens de verwachte structuur is ingevoerd maar een record kan niet worden toegevoegd, omdat de data niet volledig is of niet voldoet aan de regels van uniciteit, dan resulteert dit in de status err=1. Dit wordt als waarschuwing verwerkt. Waarschuwingen kunnen ook worden teruggegeven wanneer de verwerking van de functie OK was, maar er enkele opmerkingen/waarschuwingen zijn (zie "message"). Er wordt dan een Engelse waarschuwing meegegeven met de reden waarom het record niet is toegevoegd of verwerkt.

## NLR-EIF6 ERDMS Webservice, WSDL/Soap interface

### Interface Eisen



Figuur 6: EIF6 ERDMS webservice interface

Deze interface, die beschikbaar wordt gesteld door ERDMS, wordt gebruikt door NLRDMS voor de koppeling met ERDMS om de referentiegegevens te kunnen synchroniseren met de Europese database. De ERDMS webservice interface kent een synchrone afhandeling.

De interface koppeling valt uiteen in 2 delen:

* EIF6.1: voor het ophalen van updates uit het ERDMS.
* EIF6.2: voor het opsturen van gegevens naar het ERDMS.

De in deze paragraaf beschreven implementatie (methodiek) van de koppeling met het ERDMS, kan ook als voorbeeld dienen voor het realiseren van een koppeling met NLRDMS (zie EIF5 in paragraaf 3.6).

| IRS-NLR-013: ERDMS interface (EIF6) technische kenmerken | |
| --- | --- |
| Beschrijving: | De interface [NLR\_EIF6] is gebaseerd op de interface specificaties van ERDMS (zie [ERDMS\_WSI]), waarbij de operaties synchroon worden afgehandeld.  De interface met ERDMS kent verkeer in twee richtingen:   * NLRDMS <= ERDMS (6.1), voor het downloaden van de mutaties uit ERDMS.   *(geen ERDMS account nodig).*   * NLRDMS => ERDMS (6.2), voor het (geautoriseerd) uploaden van NL wijzigingen naar het ERDMS.   *(een geldig geconfigureerd ERDMS account is nodig).*  Wat betreft het gebruik van deze interface ligt het initiatief altijd bij NLRDMS. |
| Toelichting: | Deze synchrone webservice interface maakt onderdeel uit van ERDMS en wordt door NLRDMS aangeroepen en gebruikt. |
| Referentie: | zie [ERDMS\_WSI] |
| Herkomst: | SSS-NLR-061 ERDMS interface kenmerken. |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

### Ontwerpbeslissingen

Voor de implementatie van deze interface zijn van belang:

* Connectieparameters (WSDL URL etc).
* De te gebruiken synchrone functies van ERDMS voor ophalen mutaties.
* Hoe de mutaties te verwerken in NLRDMS.
* De te gebruiken synchrone functies voor het opsturen van mutaties.
* Hoe de mutaties op te sturen.

| IDD-NLR-014: Implementatie ERDMS interface (EIF6) in NLRDMS | |
| --- | --- |
| Beschrijving: | Voor de koppeling met ERDMS wordt gebruik gemaakt van de bestaande beproefde ERDMS webservice interface **versie 1.0g**, via **http** (zie [ERDMS\_WSI]).  Operaties worden verzonden via de ERDMS SOAP API en synchroon afgehandeld door ERDMS.  De ERDMS SOAP API gebruikt een HTTP Basic Authentication toegangsbeveiliging.  De HTTP URL van de te gebruiken ERDMS WSDL is:  <http://risdatamanagement.ris.eu/refws/v1/RefWebService.wsdl> op poort 80.  NLRDMS bepaalt wanneer de gegevens via deze interface moeten worden opgehaald of opgestuuurd. |
| Toelichting: | In de toekomst moet rekening worden gehouden met het overschakelen naar een nieuwere versie 1.1. of hoger van de ERDMS interface. |
| Referentie: | zie ook [ERDMS\_WSI] en [ERDMS\_WSDL] |
| Herkomst: | IRS-NLR-013 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

| IDD-NLR-015: Implementatie ERDMS interface (EIF6), connectieparameters | |
| --- | --- |
| Beschrijving: | Om verbinding te maken en gegevens uit te wisselen over de ERDMS interface (EIF6) zijn connectieparameters nodig.  In NLRDMS zijn de volgende configureerbare (via configuratiebestand) connectie parameters voor ERDMS aanwezig:   * URL van ERDMS webservice (WSDL) end-point. * Generieke usernaam / wachtwoord om toegang te krijgen tot de ERDMS webservice functies. * Een eigenaar naam, waaronder de mutaties afkomstig van ERDMS in NLRDMS worden aangemaakt. * ERDMS accountgegevens (actief ERDMS account) om mutaties vanuit NLRDMS op te kunnen sturen naar ERDMS. |
| Toelichting: | - |
| Referentie: | zie [ERDMS\_WSI] |
| Herkomst: | IRS-NLR-013 |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

Het synchroniseren met ERDMS bestaat uit het ophalen van de mutaties uit ERDMS en het verwerken van de gevonden mutaties in NLRDMS conform onderstaand stroomschema.



Figuur 7: Stroomdiagram, verwerken van ERDMS mutaties in NLRDMS

| IDD-NLR-016: Implementatie NLRDMS <= ERDMS synchronisatie (EIF6.1) | |
| --- | --- |
| Beschrijving: | Voor het ophalen van gegevens bij ERDMS wordt gebruik gemaakt van de **ERDMS.GetMutations()** functie, die via een aparte NLRDMS taak op basis van een timer periodiek voor een bepaald tijdsvenster de mutaties bij ERDMS opvraagt.  Het ‘eindpunt’ van het tijdsvenster is de huidige tijd.  Het ‘startpunt’ van het tijdsvenster is gelijk aan de eindtijd van de laatste succesvolle ophaal actie minus 1 uur.  Het ophalen en verwerken van mutaties verloopt verder als volgt (zie ook Figuur 7):   * Maak verbinding met ERDMS op basis van de geconfigureerde connectieparameters. * Haal ERDMS mutaties op in opgegeven tijdsvenster. * Zoek per ontvangen mutatie of het referentiegegeven al bestaat. * Als het gegeven al bestaat controleer of het versienummer hoger is. Of als het versienummer gelijk is controleer dan of de mutatiedatum van de ERDMS mutatie nieuwer**[[1]](#footnote-2)** is.   + Zo ja markeer het bestaande gegeven dan als ‘vervangen’ en voeg de mutatie toe.   + Zo nee sla de mutatie dan over. * Als het gegeven nog niet bestaat, voeg dan de mutatie toe.   Na een succesvolle ophaal actie van GetMutations(), waarbij >0 mutaties zijn verwerkt, wordt het ‘startpunt’ opgeschoven naar het gebruikte ‘eindpunt’.  Het ‘startpunt’ (tijdstip) voor de volgende synchronisatie wordt als attribuutgegeven in een interne applicatie tabel opgeslagen. |
| Toelichting: | Uitgangspunt is dat de NLRDMS database initieel gevuld is en incrementeel wordt bijgewerkt via de GetMutations() functie.  Alleen mutaties met een hoger versienummer moeten worden verwerkt, zodat bij een eventuele synchronisatie vanaf een (overlappende) periode in het verleden, de reeds bestaande en verwerkte mutaties worden overgeslagen.. |
| Referentie: | zie [ERDMS\_WSI] |
| Herkomst: | SSS-NLR-051 ERDMS connector, ophalen gegevens. |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

| IDD-NLR-017: Implementatie NLRDMS => ERDMS synchronisatie (EIF6.2) | |
| --- | --- |
| Beschrijving: | Voor het opsturen van gegevens naar ERDMS wordt gebruik gemaakt van: **ERDMS.MutateDataXML()** functie. Via een aparte NLRDMS taak wordt op basis van een timer periodiek gecontroleerd of er mutaties beschikbaar zijn met de status “Aangevraagd ERDMS”. Dat zijn door de NL datamanager goedgekeurde mutaties in NLRDMS, die opgestuurd moeten worden naar ERDMS.  Het opsturen van mutaties verloopt verder als volgt:   * Maak verbinding met ERDMS op basis van de connectieparameters inclusief geldig ERDMS account. * Stuur mutaties één voor één op naar ERDMS. * Als het opsturen en afleveren bij ERDMS succesvol is (status “goedgekeurd ERDMS”), wordt de mutatie als geaccepteerd beschouwd en gepubliceerd in NLRDMS. * Als het opsturen mislukt:   + Omdat er geen (geautoriseerde) connectie kan worden gemaakt, *dan* blijft de status ongewijzigd (volgende keer opnieuw proberen).   + Omdat de MutateDataXML() een foutcode teruggeeft, *dan* wordt de mutatie afgekeurd en de fout tekst in het opmerkingen veld geregistreerd (correctie actie door NL datamanager nodig). |
| Toelichting: | Voor de op te sturen mutaties geldt dat voor bestaande gegevens in ERDMS, het versienummer van de mutatie overeen moet komen met het versienummer in ERDMS, anders wordt de mutatie door ERDMS afgekeurd. Dit om te kunnen bewaken dat opgegeven mutaties altijd op de juiste versie van een gegeven zijn gebaseerd. |
| Referentie: | zie [ERDMS\_WSI] |
| Herkomst: | SSS-NLR-052 ERDMS connector, opsturen gegevens. |
| Kwalificatie: | Demonstratie & Inspectie |
| Gerelateerd: | - |

## Prioriteit en afhankelijkheid van eisen

**Precedence and criticality of requirements (IRS)**. This paragraph shall be numbered as the last paragraph in Section 3 and shall specify, if applicable, the order of precedence, criticality, or assigned weights indicating the relative importance of the requirements in this specification.

Examples include identifying those requirements deemed critical to safety, to security, or to privacy for purposes of singling them out for special treatment. If all requirements have equal weight, this paragraph shall so state.

Niet van toepassing.

# Kwalificatiebepalingen

**Qualification provisions**. This section shall define a set of qualification methods and shall specify, for each requirement in Section 3, the qualification method(s) to be used to ensure that the requirement has been met. A table may be used to present this information, or each requirement in Section 3 may be annotated with the method(s) to be used. Qualification methods may include:

a. Demonstration: The operation of interfacing entities that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.

b. Test: The operation of interfacing entities using instrumentation or special test equipment to collect data for later analysis.

c. Analysis: The processing of accumulated data obtained from other qualification methods. Examples are reduction, interpretation, or extrapolation of test results.

d. Inspection: The visual examination of interfacing entities, documentation, etc.

e. Special qualification methods: Any special qualification methods for the interfacing entities, such as special tools, techniques, procedures, facilities, and acceptance limits.

De kwalificatiebepalingen geven aan hoe de beschreven eis getoetst moet/kan worden. De kwalificatie van de diverse eisen zijn als apart item terug te vinden in de beschrijvingen van de eisen.

Voor de kwalificatiebepalingen zijn één of meer van de onderstaande methoden van toepassing:

|  |  |
| --- | --- |
| Kwalificatie methode | Beschrijving |
| Demonstratie | De eis betreft een duidelijk waarneembaar aspect / functionaliteit van het systeem en er kan door demonstratie direct worden aangetoond dat wordt voldaan aan de eis, zonder dat hier speciale hulpmiddelen (test software, tool etc.) of verdere analyse voor nodig is.  Een voorbeeld is het aantonen dat een bepaalde menuoptie/functie beschikbaar is in de gebruikersinterface. |
| Test | Het aantonen van de werking van het systeem, of een deel van het systeem, door het uitvoeren van testen, waarbij gebruik wordt gemaakt van de echte functies van het systeem, eventueel ondersteund door test software en tools, om gegevens te verzamelen voor latere analyses en uiteindelijke beoordeling van de test (zie ook Analyse).  Een voorbeeld is het aantonen dat een bepaald (berekend) rapport aangemaakt kan worden en de getoonde resultaten juist zijn (zie ook Analyse). |
| Analyse | Het verwerken en analyseren van verzamelde gegevens verkregen uit andere kwalificatiemethoden (bijvoorbeeld Testen). Een theoretische analyse (inspectie, reductie, interpolatie, extrapolatie etc.) kan gebruikt worden om aannemelijk te maken dat het systeem aan de gestelde eisen voldoet, zal voldoen.  Een voorbeeld is het vergelijken van meetgegevens met gegevens uit een andere bron en of het narekenen (steekproeven) van door het systeem berekende gegevens. |
| Inspectie | Het uitvoeren van een (visuele) inspectie of het onderdeel voldoet aan de gestelde eisen. Dit kan toegepast worden op eisen aan de functionaliteiten van het systeem en of op eisen die betrekking hebben op het ontwerp en of de architectuur.  Inspectie wordt o.a. toegepast in combinatie met de Demonstratie methode om te verifiëren, dat het gedemonstreerde voldoet aan de gestelde eis. |
| Geen | Voor deze eis is geen kwalificatiemethode van toepassing. |

Tabel 3: Kwalificatie methodes

# Herleidbaarheid van eisen

**Requirements traceability**. For system-level specifications, this paragraph does not apply. For subsystem-level specifications, this paragraph shall contain:

a. Traceability from each subsystem requirement in this specification to the system requirements it addresses. (Alternatively, this traceability may be provided by annotating each requirement in Section 3.).

b. Traceability from each system requirement that has been allocated to the subsystem covered by this specification to the subsystem requirements that address it. All system requirements allocated to the subsystem shall be accounted for. Those that trace to subsystem requirements contained in Interface Requirements Specifications (IRSs) shall reference those IRSs.

Niet van toepassing of nader in te vullen.

# Opmerkingen

**Notes**. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

## Afkortingen en acroniemen

| **Afkortingen** | **Omschrijving** | |
| --- | --- | --- |
| **A** | **A** | |
| ADN | European Agreement concerning the International Carriage of Dangerous Goods by Inland Waterways | |
| ADNR | Accord européen relatif au transport des matières Dangereuses par voie de Navigation sur le Rhin: reglement voor het vervoer van gevaarlijke stoffen over de Rijn. Het ADNR2007 is geharmoniseerd met IMO-regelgeving en per 2011 is de “R” in de naam vervallen en wordt vanwege bredere toepassing dan alleen de Rijn nu gesproken over ADN. | |
| API | Application Programming Interface is een interface met daarachter bepaalde functionaliteiten, die door een systeem/applicatie ter beschikking wordt gesteld om bepaalde taken voor externe aanroepende applicaties te kunnen uitvoeren. | |
| **B** | **B** | |
| BICS | Binnenvaart Informatie en Communicatie Systeem, een extern systeem waarmee schepen (schippers) en bedrijven aan de wal (verladers) zich met behulp van een ERINOT EDI-bericht o.a. elektronisch kunnen aanmelden bij de vaarwegbeheerder (in Nederland het IVS90). | |
| **C** | **C** | |
| CSV-bestand,  CSV-formaat. | Een komma gescheiden bestand, of CSV-bestand, in het Engels Comma Separated Values, is een specificatie voor databestanden.  Het bestaat enkel uit tekstgegevens, waardoor het gemakkelijk geïmplementeerd (lezen en/of schrijven) kan worden en een brede verspreiding kent. Waarden worden in principe gescheiden door komma's (of ander scheidingsteken) op een regel (gescheiden door het nieuwe-regelteken). | |
| **D** | **D** | |
| DBDD | “Database Design Description” beschrijving van de database en zijn structuren (J-STD document). | |
| **E** | **E** | |
| EDI | Electronic Data Interchange. Een stelsel van standaards en afspraken voor het langs elektronische weg uit­wisselen van berichten tussen informatiesystemen van bedrijven en (Overheids‑) instellingen. | |
| ERDMS | European (ERI) Reference Data Management Service, het Europese systeem voor het beheren van de ERI referentiedata. | |
| ERI | Electronic Reporting International | |
| ERINOT | Electronic Reporting International NOTification (bericht) | |
| **F** | **F** | |
| **G** | **G** | |
| GUI | Graphical User Interface, de gebruikersinterface van een applicatie (grafisch en of alfanumeriek), met de indeling op het scherm van de menubalk, functietoetsen; uiterlijke verschijning van rubrieknamen en informatie uit de database. | |
| **H** | **H** | |
| HS  HS-code | Harmonized System, een stofcodering die in de plaats komt van de NSTR stofcodering en gebruikt wordt om niet gevaarlijke stoffen te identificeren. | |
| HTTP | HyperText Transfer Protocol | |
| HTTPS | HTTP + SSL | |
| **I** | | **I** |
| IDD | | “Interface Design Description” beschrijft de geïmplementeerde interfaces van het systeem (J-STD document). |
| IRS | | “Interface Requirement Specification” beschrijft de eisen m.b.t. de interfaces van het systeem (J-STD document). |
| ISRS | | International Ship Reporting Standard (bijv het ERINOT 1.2 bericht). |
| IVS90 | | Informatie en Volgsysteem voor de Scheepsvaart, die ERINOT (reis- lading) berichten kan ontvangen en verwerken en waarmee Rijkswaterstaat het scheepvaartverkeer op de Nederlands (hoofd)vaarwegen (op de diverse verkeersposten en sluizen) opvolgt. |
| **J** | | **J** |
| JSTD-016 | | Joint Standard 016, documentatie standaard voor het ontwikkelen van Software. |
| **K** | | **K** |
| **L** | | **L** |
| **M** | | **M** |
| **N** | | **N** |
| NLRDMS | | Nederlandse (NL) “Reference Data Management Service”, de Nederlandse databank met referentiecodes, die gebruikt worden in het E-Melden domein. |
| NSTR | | Nomenclature pour les statistiques de transports, een stofcodering (die niet meer in gebruik is) die gebruikt werd om een bepaalde stof t.b.v. statistiek doeleinden te identificeren. |
| **O** | | **O** |
| **P** | | **P** |
| PLATINA | | Een Europees project om de EU lidstaten te ondersteunen in het bijeenbrengen van de actoren, het aanjagen en het realiseren van (RIS) oplossingen voor de binnenvaart, zowel voor overheden als het bedrijfsleven (e.e.a. in het kader van het NAIADES action plan). |
| **Q** | | **Q** |
| **R** | | **R** |
| RIS | | Zie ook River Information Services; een indeling en definitie van voor binnenvaart van belangzijnde (ICT) Diensten. |
| RWS | | Rijkswaterstaat is de uitvoeringsorganisatie die in opdracht van de minister en staatssecretaris van Infrastructuur en Milieu (IenM) de nationale netwerken (weg en water) op duurzame wijze beheert en ontwikkelt. |
| **S** | | **S** |
| SCOM | | “Software Center Operator Manual” bevat de instructies voor beheerders van een softwaresysteem dat geïnstalleerd is in een computercentrum (J-STD document) |
| SOAP | | SOAP (aanvankelijk een afkorting voor Simple Object Access Protocol) is een computerprotocol dat wordt gebruikt voor communicatie tussen verschillende componenten. SOAP is een protocol dat XML-berichten stuurt, meestal over HTTP, maar ook over SMTP, HTTPS of FTP. |
| SSS | | “System Subsystem Specification” bevat de systeem (functionele) eisen voor het (gewenste) te realiseren systeem (J-STD document). |
| SUM | | “Software User Manual” de gebruikersinstructie (handleiding) voor het systeem. |
| **T** | | **T** |
| **U** | | **U** |
| UTF-8 | | UCS Transformation Format-8-bit, is een manier om Unicode/ISO 10646-tekens op te slaan, een zogenaamde tekencodering. |
| **V** | | **V** |
| VN-stof | | Gevaarlijke stof; volgens ADN- en/of IMO-classificaties (zie ook ADN). |
| **W** | | **W** |
| **X** | | **X** |
| XML | | XML staat voor eXtensible Markup Language. Deze taal is ontworpen om in bepaalde gegevens een structuur aan te brengen. En voorziet in het op een gestructureerde manier kunnen uitwisselen van informatie tussen systemen, via zogenaamde XML (tekst)berichten. |
| XSD | | **X**ML **S**chema **D**efinition (Language) is een taal voor het beschrijven van de structuur van XML-documenten. In een XSD of XML-Schema kan men schema's maken voor XML-documenten, waarmee deze formeel worden gespecificeerd, en op grond waarvan zij kunnen worden gevalideerd. Zeer simplistisch uitgedrukt kan een XML-Schema onder meer aangeven wat voor soort tags in een XML-document mogen voorkomen en welke waarden en formaten voor de data (invulling) van die tag gelden. |
| **Y** | | **Y** |
| **Z** | | **Z** |

## Terminologie

| **Term** | **Omschrijving** |
| --- | --- |
| **A** | **A** |
| Authenticatie | Authenticatie is het mechanisme, waarbij nagegaan wordt of een bepaalde gebruiker of systeem daadwerkelijk de gebruiker of het systeem is, wie hij zegt dat hij is (bv de controle op inlognaam en juiste wachtwoord). Authenticatie is de tweede stap (na identificatie) in het toegangscontroleproces. |
| Autorisatie | Autorisatie is het mechanisme, waarbij na identificatie en authenticatie een gebruiker of systeem rechten (toegang) krijgt tot bepaalde functies en of informatie. |
| **B** | **B** |
| B-vlag | Een verkeersteken, dat zeeschepen moeten voeren als zij bepaalde (hoeveelheden) gevaarlijke stoffen aan boord hebben. Zie ook Kegel(s). |
| Berichtformaat | Met *berichtformaat*, wordt bedoeld het formaat waarin een bericht (hetzelfde bericht) is opgebouwd, zoals EDIFact of XML formaat. |
| Berichtsoort | Met *berichtsoort* wordt het type bericht bedoeld, zoals: een gevaarlijke lading melding (IFTDGN bericht), een passagierslijst melding (PAXLST bericht). |
| Berichtversie | Met *berichtversie* van een bericht, wordt bedoeld de verschillende mogelijke versies van hetzelfde bericht, zoals “ERINOT 1.2”, “ERINOT 2.0” etc. |
| BERMAN bericht | Een Internationaal gestandaardiseerd bericht, gebaseerd op het EDIFact “Berth Management” (BERMAN) bericht, bedoeld om een Ligplaats met bijbehorende diensten bij een (Haven)autoriteit aan te vragen. |
| Binnenvaartschip | Een schip bedoeld voor het transporteren van ladingen over binnenwateren. Voor binnenvaartschepen geldt een bepaald regime dat kan verschillen van zoals dat voor zeeschepen geldt. |
| **C** | **C** |
| Call-sign | De roepletters van een Zeeschip volgens het Lloyds-register. Zie ook Lloyds. |
| Casco | Een eenheid met zelfstandig drijfvermogen, die alleen of in combinatie met andere casco's aan het scheepvaart­verkeer kan deelnemen (varend of stilliggend). Elk casco kent een set van vaste gegevens (identificatie, afmetingen), die in principe niet veranderen en niets te maken hebben met de vervoerde lading. |
| **D** | **D** |
| Double quoted waarden | Tekstreeksen, die aan het begin en einde voorzien zijn van dubbele quote tekens (“”). |
| **E** | **E** |
| E-Melden | Het elektronisch melden van informatie, waar dat voorheen veelal op andere manieren ging, zoals op papier, fax of via spraak/marifoon. |
| E-Meldplicht | De verplichting om, als voldaan wordt aan bepaalde criteria, dan ook bepaalde informatie, verplicht elektronisch te melden (andere vorm van melden is niet toegestaan, zie ook [RPR], [BPR]). Sinds 2010 dienen schepen die > 20 containers vervoeren of een of meer gevaarlijke ladingen aan boord hebben, zich elektronisch te melden (via marifoon is voor die groep niet meer toegestaan). |
| ENC kaarten | Staat voor Electronic Navigation Chart (Elektronische Navigatie Kaart). Deze kaarten worden door alle professionele kaartviewers gebruikt, in de beroepsvaart. De kaarten zijn vectorkaarten. |
| ERI berichten | In het kader van Electronic Reporting International gestandaardiseerde elektronische berichten, zoals: het ERINOT bericht. |
| ERI-Locatie | ERI-locaties, zijn locatiecodes (laad-, los- en overige (lig)plaatsen) die relevant zijn voor het elektronisch melden. |
| ERINOT bericht | Een Internationaal gestandaardiseerd (notification) bericht, gebaseerd op het EDIFact IFTDGN bericht, om Reis- en (gevaarlijke) lading informatie aan vaarweg autoriteiten aan te kunnen melden. |
| ERIVOY bericht | Een Internationaal gestandaardiseerd bericht, gebaseerd op het EDIFact IFTSAI bericht, om Reisplan gegevens aan vaarweg autoriteiten aan te kunnen melden. |
| ERIMAN bericht | Een Internationaal gestandaardiseerd bericht om een elektronisch (lading) manifest aan te kunnen melden (nog niet toegepast in ERI domein). |
| **F** | **F** |
| **G** | **G** |
| **H** | **H** |
| **I** | **I** |
| International Ship Reporting Standard code  (ISRS-code) | De ISRS code is bedoeld om een locatie aan te duiden en bestaat uit: UN-landencode, UN-locatiecode, een vijf cijferig nummer dat de vaarweg(sectie) aanduidt, een vijf cijferige code die de terminal of het vaarwegpunt aanduidt, en tenslotte een vijf cijferige aanduiding van de hectometerpositie in de vaarweg(sectie) |
| **J** | **J** |
| **K** | **K** |
| Kegel(s) | Een verkeersteken, dat binnenschepen met een gevaarlijke lading moeten voeren. Afhankelijk van de soort, hoeveelheid gevaarlijke stoffen en het scheepstype moet een binnenschip één, twee of drie kegels voeren, welke een indicatie is voor de mate van gevaarlijkheid van dat transport. Voor een zeeschip geldt een zogenaamde B-vlag. |
| **L** | **L** |
| **M** | **M** |
| Meldplicht | De wettelijke verplichting van bepaalde categorieën schepen op bepaalde vaarwegen om zich te melden bij de Verkeerspost of Sluiscomplex (aan de vaarwegbeheerder). Dat kan via de marifoon op een bepaald kanaal of elektronisch via een elektronische meldapplicatie (bijv. BICS). |
| Mutatie | Een mutatie is een wijziging op een gegeven. Dat kan zijn: het invoeren van een nieuw gegeven (record) of doorvoeren van een wijziging op een bestaand gegeven. |
| **N** | **N** |
| **O** | **O** |
| **P** | **P** |
| **Q** | **Q** |
| **R** | **R** |
| Reis | Een reis is een beschrijving van het transport door een romp of meerdere rompen. Het transport wordt gekenmerkt door een herkomst en een bestemming met ladinggegevens. |
| Responsetijd | De gemiddelde tijd die het computersysteem nodig heeft voor het opbouwen van het antwoord bij beeldschermtoepassingen. Wordt gemeten vanaf het moment dat de gebruiker een functietoets of de Entertoets indrukt, tot aan het moment dat het antwoord van het systeem volledig op het beeldscherm wordt weergegeven. |
| RIS index  (RIS code) | Een (belangrijk) referentiepunt of object op of langs de vaarweg.  RIS indexen zijn er o.a. voor zogenaamde junctions (knooppunten), havens, ligplaatsen, sluizen enz. De RIS index speelt een rol in het E-Melden om een bepaalde locatie te kunnen identificeren De RIS index is opgebouwd uit de UN/locode, vaarwegcode, terminal of objectcode en een hectometercode (hectometerpositie in de vaarweg). |
| River Information Services | River Information Services is defined as an (ICT) concept for harmonised information services to support traffic and transport management in inland waterways including interfaces to other transport modes. |
| Romp | (Deel van) een varende eenheid. Een 4-baks vracht­duwbak-combinatie bestaat uit 5 rompen (1 duwboot en 4 bakken). |
| **S** | **S** |
| Samenstel | Zie Varende Eenheid |
| Scheepstype | Typering van schepen (casco, romp of varende eenheid) waarmee de verschillende kenmerken van een soort schip worden vastgelegd. Denk hierbij aan categorie (vrachtschip, tankschip, dienstschip), hoofdindeling (binnenvaart, zeevaart, recreatie) en maximum laadvermogen. Ook is vastgelegd in welke combinaties verschillende scheepstypen als één varende eenheid voor mogen komen (samensteltype). |
| Schip | Komt overeen met de entiteit Varende Eenheid uit het gegevensmodel. |
| Schipper | De verantwoordelijk aan boord voor de varende eenheid (kapitein), die ook verantwoordelijk is voor het melden (doorgaans via EDI berichten) van zijn reis- en ladinggegevens aan autoriteiten. |
| Seinvoering | Verkeerstekens, die door een schip met gevaarlijke lading gevoerd moeten worden op basis van de reglementering (B-vlag bij zeevaart, kegels bij binnenvaart). |
| Sluis | Een bouwkundig kunstwerk, bedoeld om schepen te schutten. Een Sluis verbindt twee stukken vaarweg met een verschillende waterstand. |
| Sluiscomplex | Een groep van één of meer bij elkaar gelegen sluizen die dezelfde vaarwegen verbinden en onder het beheer van één organisatorische eenheid vallen. |
| **T** | **T** |
| Terminal | Een laad- losplaats, ligplaats in een haven of langs een rivier, doorgaans bedoeld voor het laden en of lossen van lading en meestal voorzien van laad-, los faciliteiten (kranen). |
| Transport-combinatie | Zie Varende Eenheid |
| **U** | **U** |
| UN-Locode | United Nations Locationcode, een code opgebouwd uit een land en locatie code, waarmee belangrijke plaatsen in de wereld uniek kunnen worden geïdentificeerd. Zie [www.unece.org](http://www.unece.org) (UN recomendation 16). |
| **V** | **V** |
| Varende Eenheid (VE) | Enkelvoudig of samengesteld schip (schip dat uit één of meer casco's bestaat). Van de varende eenheid (VE) worden, naast de cascogegevens, ook variabele scheepsgegevens geregistreerd die (mede) bepaald worden door de samenstellende rompen en de daarin vervoerde lading(en).  Binnen een varende eenheid is één bepaald casco naamgevend en de VE zal onder die (hoofd)naam worden geregistreerd. |
| Verlader | Instantie (een bedrijf) die d.m.v. EDI (elektronische berichtuitwisseling) informatie levert over schepen en ladingen. |
| VN-nr. | Door de Verenigde Naties vastgesteld identificatienummer (incl. classificaties) voor het identificeren van gevaarlijke ADN stoffen. |
| **W** | **W** |
| Webservice | Een webservice kan omschreven worden als een interface van een applicatiecomponent, die toegankelijk is via standaard webprotocollen en waarbij meestal wordt gecommuniceerd via XML zonder menselijke tussenkomst. Een webservice maakt het mogelijk om op afstand (meestal over het Internet) vanaf een client (een (web)applicatie of component) een dienst op te vragen aan een server. Webservices spelen een groeiende rol in het denken over component-based systems. |
| **X** | **X** |
| **Y** | **Y** |
| **Z** | **Z** |
| Zeeschip | Zeeschepen zijn schepen die op zee mogen varen en zeewaardig zijn. Zeeschepen mogen ook over binnenwateren varen en kunnen zodoende ook beschikken over binnenvaart identificatienummers (ENI). |

1. Beschrijving referentiegegevens velden

Appendixes. Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling.

Appendixes shall be lettered alphabetically (A, B, etc.).

In deze bijlage zijn de technische specificaties en bijbehorende bestanden van de NLRDMS webservice interface (zie paragraaf 3.6) opgenomen, bestaande uit:

* NLRDMS Web Service Description Language (WSDL) file.
* NLRDMS XML Schema Definition (XSD) file.
* NLRDMS webservice foutcodes en omschrijvingen.
  1. NLRDMS Webservice – WSDL

De WSDL file in deze sectie, bevat de technische specificaties van de NLRDMS webservice interface ([NLR-EIF5]) en maakt gebruik van een bijbehorende XSD (zie volgende bijlage).

In de bijbehorende **XSD** zijn de volgende structuren voor de webservice vastgelegd:

* API functies, zoals de naam en parameters van de functies.
* Complex type structuren van generieke data typen (zie paragraaf 3.6.4).
* Simple type structuren, elke gebruikte variabele in een Complex type structuur is gebaseerd op een gestandaardiseerde Simple type structuur (bv: een lengte, breedte, diepte => type afstand).

|  |
| --- |
| Indien de onderstaande WSDL wordt gepubliceerd is het volgende van belang:   * Zorg er voor dat de juiste XSD wordt geïmporteerd in de WSDL (zie “<xsd:import namespace=…>”). * Zorg er voor dat de juiste end-point wordt gebruikt (zie “<soap:address location=…”>). * Zorg er voor dat Basic HTML authentication wordt geïmplementeerd op het opgegeven end-point. |

De **actuele** NLRDMS WSDL is altijd online op te vragen via de NLRDMS URL (zie IDD-NLR-011 en [NLRDMS\_WSDL]).

NLRDMS Webservice WSDL file:

<?xml version="1.0" encoding="UTF-8"?>

<!-- ===================================================================================== -->

<!-- NLRDMS Webservice interface WSDL. This WSDL contains the specification of the NL RefTool webservice -->

<!-- interface (the API) for the NL (client of ERDMS) RIS Datamanagement Service (NLRDMS). -->

<!-- -->

<!-- These NLRDMS specifications are different (additional functions+fixes) from ERDMS webservice specifications 1.0g!!! -->

<!-- -->

<!-- Using the functionality of the NL webservice interface it is possible to: -->

<!-- - request (find) information about a code or description (translate code to description etc). -->

<!-- - request all records from one of the refdata tables for synchronising local databases. -->

<!-- - request last updated records/mutations for synchronising local databases. -->

<!-- - request and specify updates/mutations. -->

<!-- Datastructures, datatypes (incl basictypes) and function parameters are defined in a separate XSD file -->

<!-- and are imported into the WSDL interface (see <types\xsd> section in this WSDL). -->

<!-- -->

<!-- The following interface functions are defined: -->

<!-- - matchByCode(reftype,code,version): find refdata based on a code. -->

<!-- - matchByName(reftype,name,version): find refdata based on a description. -->

<!-- - getMutatedTables(fromdate,todate): request list of mutated tables since (>=) given date. -->

<!-- - getMutations(reftype,fromdat,todate): request list of mutations (complete data) in given period. -->

<!-- - getDataXML(reftype,version): request all records of specified reftype (master dump). -->

<!-- - getRisDataXML(subcode,funcode,version): request all Ris index data, based on country, subc or subfunction (master dump). -->

<!-- - mutateDataXML(username,password,reftype,refdata): mutate specific record in NLRDMS database, without human interaction (only Ris idx and eri\_locations) -->

<!-- - requestMutationXML(username,password,refDataReqMut): request a mutation for specific record in NLRDMS database, req will be validated = human interaction. -->

<!-- -->

<!-- Journal, Releasenotes: -->

<!-- -->

<!-- v1.0 LS/BB mei 2010 Initial version -->

<!-- V1.0e LS/BB okt 2010 Final version for impl of the WS version 1.0 (identified as v1) -->

<!-- V1.0f LS/BB mrt 2011 Implementation extra fields for the RISindex -->

<!-- V1.0g LS/BB june 2011 Corrections xsd after testing (CR001-002) -->

<!-- V1.0h LS/BB jan 2013 Minor modifications in wsdl,xsd (splitting NL from EU: xmlns renamed to http://nlreftool + feedback/corrections from operational usage -->

<!-- V1.1a BB/LS jun 2013 Extra specific NL-RDMS (NL\_REF) functionality, added new operation + several significant changes, fixes in WS (XSD) interface) for NL RIS systems (VOS,BICS2 etc) -->

<!-- -->

<!-- ===================================================================================== -->

<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://nlreftool/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://nlreftool/" name="NLRefWebService">

<types>

<xsd:schema>

<!--=========================================================== -->

<!-- Specify the URL to the XSD (for docstyle wrapped) here!!! -->

<!-- (url's (endpoints) are different for different versions of the webservice) -->

<!--=========================================================== -->

<!-- <xsd:import namespace="http://rwsreftool/" schemaLocation="http://reftool.risexpertgroups.org:80/refws/v1/RefWebService.xsd"/> -->

<xsd:import namespace="http://nlreftool/" schemaLocation="NLRDMS\_WS\_XSD\_1.1a.xsd"/>

</xsd:schema>

</types>

<!-- Main NLRDMS message datatypes (soap document xml structures) for the webservice operations are defined as elements in separate (imported) XSD: NLRDMS\_WS\_XSD\_nnn.xsd -->

<message name="exception">

<part name="fault" element="tns:exception"/>

</message>

<message name="matchByCode">

<part name="parameters" element="tns:matchByCode"/>

</message>

<message name="matchByCodeResponse">

<part name="parameters" element="tns:matchByCodeResponse"/>

</message>

<message name="matchByName">

<part name="parameters" element="tns:matchByName"/>

</message>

<message name="matchByNameResponse">

<part name="parameters" element="tns:matchByNameResponse"/>

</message>

<message name="getMutatedTables">

<part name="parameters" element="tns:getMutatedTables"/>

</message>

<message name="getMutatedTablesResponse">

<part name="parameters" element="tns:getMutatedTablesResponse"/>

</message>

<message name="getMutations">

<part name="parameters" element="tns:getMutations"/>

</message>

<message name="getMutationsResponse">

<part name="parameters" element="tns:getMutationsResponse"/>

</message>

<message name="getDataXML">

<part name="parameters" element="tns:getDataXML"/>

</message>

<message name="getDataXMLResponse">

<part name="parameters" element="tns:getDataXMLResponse"/>

</message>

<message name="getRisDataXML">

<part name="parameters" element="tns:getRisDataXML"/>

</message>

<message name="getRisDataXMLResponse">

<part name="parameters" element="tns:getRisDataXMLResponse"/>

</message>

<message name="mutateDataXML">

<part name="parameters" element="tns:mutateDataXML"/>

</message>

<message name="mutateDataXMLResponse">

<part name="parameters" element="tns:mutateDataXMLResponse"/>

</message>

<message name="requestMutationXML">

<part name="parameters" element="tns:requestMutationXML"/>

</message>

<message name="requestMutationXMLResponse">

<part name="parameters" element="tns:requestMutationXMLResponse"/>

</message>

<!-- The NLRDMS webservice operations (functions and parameters of the webservice interface) are defined here -->

<portType name="RefWeb">

<operation name="matchByCode">

<input message="tns:matchByCode"/>

<output message="tns:matchByCodeResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

<operation name="matchByName">

<input message="tns:matchByName"/>

<output message="tns:matchByNameResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

<operation name="getMutatedTables">

<input message="tns:getMutatedTables"/>

<output message="tns:getMutatedTablesResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

<operation name="getMutations">

<input message="tns:getMutations"/>

<output message="tns:getMutationsResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

<operation name="getDataXML">

<input message="tns:getDataXML"/>

<output message="tns:getDataXMLResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

<operation name="getRisDataXML">

<input message="tns:getRisDataXML"/>

<output message="tns:getRisDataXMLResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

<operation name="mutateDataXML">

<input message="tns:mutateDataXML"/>

<output message="tns:mutateDataXMLResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

<operation name="requestMutationXML">

<input message="tns:requestMutationXML"/>

<output message="tns:requestMutationXMLResponse"/>

<fault message="tns:exception" name="exception"/>

</operation>

</portType>

<!-- The NLRDMS webservice function bindings are defined here -->

<binding name="RefWebPortBinding" type="tns:RefWeb">

<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>

<operation name="matchByCode">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

<operation name="matchByName">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

<operation name="getMutatedTables">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

<operation name="getMutations">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

<operation name="getDataXML">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

<operation name="getRisDataXML">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

<operation name="mutateDataXML">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

<operation name="requestMutationXML">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

<fault name="exception">

<soap:fault name="exception" use="literal"/>

</fault>

</operation>

</binding>

<!-- The NLRDMS webservice port and address (url, endpoint) bindings (per webservice version) are defined here, service name has a direct link with dirstruct in war -->

<service name="v1/NLRefWebService">

<port name="RefWebPort" binding="tns:RefWebPortBinding">

<!-- generic URL to the web service port (keywords: host & port) will be substituted runtime with the correct root url & port by an appl servlet. -->

<!-- <soap:address location="http://host:port/refwsclient/v1/NLRefWebService"/> will become i.e. <soap:address location="http://reftool.e-navigation.nl/refwsclient/v1/NLRefWebService"/> -->

<soap:address location="http://host:port/refwsclient/v1/NLRefWebService"/>

</port>

</service>

</definitions>

* 1. NLRDMS Webservice – XSD

De XSD file in deze sectie bevat de definities van alle datastructuren zoals gebruikt in de NLRDMS webservice interface [NLR-EIF5].

De gebruikte datastructuren in de XSD zijn op te delen in:

*Global elements*:

* API Function message definities (operatie/functienamen zoals gedefinieerd in de WSDL operation section), zoals: matchByCode, mutateDataXML().

*Complex types* definities voor:

* API Functie parameters (input operatie messages).
* Overige substructuren, zoals resultaat datastructuren: “refdataReturnType” en “refdataReqMutType” etc).

*Simple types* definities:

* Basis types zoals gebruikt in de diverse Complex types. Deze basic “simpletypes” worden vooral gebruikt om de verschillende items/velden als “typedef” vast te leggen, zoals gewichten, afmetingen etc. Deze kunnen worden gezien als de core components van de XSD.

NLRDMS Webservice XSD file:

<?xml version="1.0" encoding="UTF-8"?>

<!-- =================================================================================== -->

<!-- NL Refdata Management (NLRDMS Tool) Webservice interface XSD. This XSD contains the specification of the webservice -->

<!-- interface (operations) for the NLRDMS WSDL, and the corresponding function parameters incl. the datatypes and basic types. -->

<!-- -->

<!-- These NLRDMS specifications are different (additional functions+fixes) from ERDMS webservice specifications 1.0g!!! -->

<!-- -->

<!-- The following NLRDMS Tool WSDL (web service) interface functions are defined: -->

<!-- - matchByCode(): find refdata based on a code. -->

<!-- - matchByName(): find refdata based on a description. -->

<!-- - getMutatedTables(): request list of mutated tables since given date. -->

<!-- - getMutations(): request list of mutations (complete data) in given period. -->

<!-- - getDataXML(): request all records of specified reftype (master dump). -->

<!-- - getRisDataXML(): request all records of Ris index, based on country and or subfunction (master dump). -->

<!-- - mutateDataXML(): mutate specific record in reftool database, without human interaction (only Ris idx and eri\_locations) -->

<!-- - requestMutationXML(): request mutation for specific record, req will/must be validated = human interaction. -->

<!-- -->

<!-- Every WS function (operation) wil return (responses with) a XML datastructure as specified in this XSD. The find functions -->

<!-- will return the matched reference data (records of refdataReturnType), other function will return a status result. -->

<!-- -->

<!-- -->

<!-- SOME REMARKS for the structure of the defined NLRDMS datatypes etc in this XSD:-->

<!-- # The Soap operation/messages in the WSDL refers to elements in this XSD, where each elements is a global ComplexType.-->

<!-- # The Complextype structures are based on (sub)elements wich are based on Simpletypes (basic types, core components)!!! -->

<!-- # Edit the Basic Simpletypes ONLY, DON'T edit (sub)elements directly from within the ComplexTypes!!! -->

<!-- # Be aware that refdataReturnType has the same structure (on field basis) as refdataReqMutType. So keep them aligned!! -->

<!-- use Basic (logical) types for new elements = xsd Simpletypes (define a Simpletype (core component) for each new element -->

<!-- in those structures, and use it in both structures). -->

<!-- -->

<!-- Journal, Releasenotes: -->

<!-- -->

<!-- v1.0 LS/BB mei 2010 Initial (new style) version -->

<!-- v1.0b BB jun 2010 Added nts data in choice structure of refData and extended reftype, langType -->

<!-- v1.0c BB jul 2010 Corrections (andr=>adn, langc to lowercase, some corrections, added additional langs in eri tables) -->

<!-- v1.0d BB sep 2010 Corrections during impl: -->

<!-- -resolved name conflicts (iversion attribute, renamed all compex, simplatypes to ..Type) -->

<!-- -forgotten eri\_country enum added in reftypeType -->

<!-- v1.0e BB/LS okt 2010 Correction changes after consulting ERI/NtS/Exp Grps, Platina stakeholders) -->

<!-- Resulting in the final version 1.0 of the webservice (idetified as v1), Changes: -->

<!-- -iversion attribute and iversionType removed (version of WS is already identified in the endpoint url) -->

<!-- -maxWeightCodeInland ADN elements renamed (2x) to maxWeightCodeInlandBulkCont -->

<!-- -ris\_idxType\risidxCode changed to simpletype (instead of complex) -->

<!-- -ivsCode renamed to ivsVTSCode in eri\_locationType and er\_locationReqMutType -->

<!-- -recCount added to results of matchByCode, byName, GetMut, GetDataXML, GetRisDataXML -->

<!-- -refdataReqMutType\reftype changed type to reftypeType -->

<!-- -refdataReqMutType\eri\_hscode\name\loc changed type to goodnameType -->

<!-- -refdataReqMutType\eri\_adn\flagsbulk and flagstank change to type flagsType -->

<!-- -ris\_idxType\objname + ris\_idxReqMutType\objname: from mandatory -> optional -->

<!-- -ris\_idxType\wwname\loc + ris\_idxReqMutType\wwname\loc: from optional -> mandatory -->

<!-- -ris\_idxType\routename\loc + ris\_idxReqMutType\routename\loc: from optional -> mandatory -->

<!-- v1.0e2 BB/LS feb 2011 Textual Corrections in comments (replaced Reftool with ERDMS/NLRDMS) -->

<!-- v1.0e3 BB/LS feb 2011 Corrections for bugfixes the WS API v1.0 (some minimal alignements of field lengths with the DB) -->

<!-- -rissectnodeType (simpletype) added with maxlen 20 and used for ris\_IdxType.sectionode (sectnode is NOT a ris idxcode) -->

<!-- -erilocnameType (simpletype) changed maxlen 30 to 90 -->

<!-- -goodDgsNameType (simpletype) added to make dgs names diffrent from hs names and used in eri\_adn structures -->

<!-- -countryNameType(simpletype) changed maxlen 30 to 70 -->

<!-- V1.0f1 LS/BB mar 2011 Changes due CR001-002 -->

<!-- -CR001 ris\_idxType.locname changed Mandatory->Optional (still in discussion if this is useful, a better choice is to leave it M) -->

<!-- -CR001 ris\_idxType.wwname changed Optional->Mandatory -->

<!-- -CR001 ris\_idxReqMutType.locname changed Mandatory -> Optional -->

<!-- -CR001 ris\_idxReqMutType.wwname changed from Optional->Mandatory according to the CR001 -->

<!-- -CR002 ris\_idxType added 51 extra (new) fields starting with ris\_idxtype.commInfo -->

<!-- -CR002 ris\_idxReqMutType added 51 extra (new) fields starting with ris\_idxtype.commInfo -->

<!-- -CR002 shiptypeNameType length changed form 35 to 70 -->

<!-- -CR002 ris\_idxrisgaugeidType ris idx Gauge id track and trace changed in ris idx national Gauge Code type -->

<!-- V1.0g LS/BB june 2011 Corrections after testing (CR001-002) -->

<!-- -Updated several descriptions (xs:documentation) -->

<!-- V1.0h LS/BB jan 2013 Minor corrections in xsd (splitting NL from EU and minor corrections from operational usage) -->

<!-- -removed ComplexType def (extraclass level) for eri\_adncode in choice struct of refdataReturnType (feedback from impl) -->

<!-- -renamed namespace http://rwsreftool to http://nlreftool.-->

<!-- V1.1a BB/LS jun 2013 Significant changes for NLRDMS (extra NL specific functionality), added extra Operation + Tables + Structure changes -->

<!-- -added new operation: getMutatedTables(fromDate, toDate) -->

<!-- -added new refdata tables + structure: er\_provider, er\_erireportingpoint and er\_locsreppoints -->

<!-- -changed operation: getMutations(); parameter toDate changed to optional, meaning (defaults to ToDay) -->

<!-- -changed existing operation getMutations(fromDate, toDate); changed parameter 'toDate' from mandatory => optional -->

<!-- -changed simpletype: reftypeType; added extra enumerations er\_xxx -->

<!-- -added (new) complextypes er\_xxxType and er\_xxxReqMutType -->

<!-- -changed complextype: refdataReturnType; added extra er\_xxx choice structures for additional (new) tables -->

<!-- -changed complextype: refdataReqMutType; added extra er\_xxx choice structures for additional (new) tables -->

<!-- -changed simpletype: locVtsCodeType; renamed to providerIdType, an..20 -->

<!-- -changed refdataReturnType.eri\_adncode.adnClass from optional to mandatory -->

<!-- -changed refdataReturnType.eri\_adncode.imoClass from mandatory to optional -->

<!-- -changed simpletype SignalCodeType; length=1 to maxlen=1 (so it can be length 0 or 1) -->

<!-- -changed simpletype conesType; length=1 to maxlen=1 (so it can be length 0 or 1) -->

<!-- -changed simpletype flagsType; length=1 to maxlen=1 (so it can be length 0 or 1) -->

<!-- -changed complextype: refdataReturnType; added extra adn2013 fields (label4, aquaticToxicity, ChemicalStability, cmrProp, floater -->

<!-- -changed complextype: refdataReqMutType; added extra adn2013 fields -->

<!-- -added new simpletypes for and2013 ((label4, aquaticToxicity, ChemicalStability, cmrProp, floater) -->

<!-- -->

<!-- =================================================================================== -->

<!-- -->

<!-- below are the NLRDMS element definitions of the soap operations/messages as defined in the WSDL -->

<xs:schema targetNamespace="http://nlreftool/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://nlreftool/" version="1.0">

<xs:element name="matchByCode">

<xs:annotation>

<xs:documentation>Function decl: Find information based on a Code (lookup code)</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:matchByCodeType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="matchByCodeResponse">

<xs:annotation>

<xs:documentation>Response decl MatchByCode(), results (nr of recorfds) can be limited</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:matchByCodeResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="matchByName">

<xs:annotation>

<xs:documentation>Function decl: Find information, records based on a description (lookup description)</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:matchByNameType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="matchByNameResponse">

<xs:annotation>

<xs:documentation>Response decl MatchByName(), results (nr of recs) can be limited</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:matchByNameResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getDataXML">

<xs:annotation>

<xs:documentation>Function decl: Request all current (actual) records of particular data reftype (master dump)</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getDataXMLType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getDataXMLResponse">

<xs:annotation>

<xs:documentation>Response decl getDataXML(), all requested data (no limitations)</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getDataXMLResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getMutatedTables">

<xs:annotation>

<xs:documentation>Function decl: Request all changed (updated) Tables in (>=) given period.</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getMutatedTablesType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getMutatedTablesResponse">

<xs:annotation>

<xs:documentation>Response decl getMutatedTables(), list of all mutated tables (reftypes) in given period.</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getMutatedTablesResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getMutations">

<xs:annotation>

<xs:documentation>Function decl: Request all changes (records) in (>=) a given period</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getMutationsType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getMutationsResponse">

<xs:annotation>

<xs:documentation>Response decl getMutations(), list of all mutations during given period</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getMutationsResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getRisDataXML">

<xs:annotation>

<xs:documentation>Function decl: Request specific Ris index records (possibly a subselection, by country and or by function)</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getRisDataXMLType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="getRisDataXMLResponse">

<xs:annotation>

<xs:documentation>Response decl getRisDataXML(), all requested data (no limitations)</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:getRisDataXMLResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="mutateDataXML">

<xs:annotation>

<xs:documentation>Function decl: Mutate one (1) specific record of the reference data, without human interaction (add or change (based on correct version id) existing one). Only applicable for RisIndx and eri\_locations</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:mutateDataXMLType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="mutateDataXMLResponse">

<xs:annotation>

<xs:documentation>Response decl mutateDataXML(), status result of mutate call.</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:mutateDataXMLResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="requestMutationXML">

<xs:annotation>

<xs:documentation>Function decl: Request a mutation for one (1) specific record (as specified as input). Request will be processed manually (validated by Datamanager)</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:requestMutationXMLType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="requestMutationXMLResponse">

<xs:annotation>

<xs:documentation>Response decl requestMutationXML(), status result of request call</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:complexContent>

<xs:extension base="tns:requestMutationXMLResponseType"/>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="exception" type="tns:exceptionType">

<xs:annotation>

<xs:documentation>exception as used in faults or when an error is encounterd in NLRDMS (exception contains exception/error text prefixed with a err nr)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="matchByCodeType">

<xs:annotation>

<xs:documentation>type decl for matchByCode() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="reftype" type="tns:reftypeType"/>

<xs:element name="code" type="xs:string"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="matchByCodeResponseType">

<xs:annotation>

<xs:documentation>type decl for matchByCode() return</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="recCount" type="tns:recCountType">

<xs:annotation>

<xs:documentation>Indicates how many records are available/were found

(due limitations less records can be returned/available in the resulting xml structure).</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="refdataReturn" type="tns:refdataReturnType" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="matchByNameType">

<xs:annotation>

<xs:documentation>type decl for matchByName() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="reftype" type="tns:reftypeType"/>

<xs:element name="lang" type="tns:langType" minOccurs="0"/>

<xs:element name="name" type="xs:string"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="matchByNameResponseType">

<xs:annotation>

<xs:documentation>type decl for matchByName() return</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="recCount" type="tns:recCountType">

<xs:annotation>

<xs:documentation>Indicates how many records are available/were found

(due limitations less records can be returned/available in the resulting xml structure).</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="refdataReturn" type="tns:refdataReturnType" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getMutatedTablesType">

<xs:annotation>

<xs:documentation>type decl for getMutatedTables() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="fromDate" type="xs:dateTime">

<xs:annotation>

<xs:documentation>fromDate and toDate in datetime format in UTC time</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="toDate" type="xs:dateTime" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getMutatedTablesResponseType">

<xs:annotation>

<xs:documentation>type decl for getMutatedTables() return</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="recCount" type="tns:recCountType">

<xs:annotation>

<xs:documentation>Indicate how many mutated tables are available/were found.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="mutatedTables" type="tns:mutatedTableReturnType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getMutationsType">

<xs:annotation>

<xs:documentation>type decl for getMutations() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="reftype" type="tns:reftypeType" minOccurs="0"/>

<xs:element name="fromDate" type="xs:dateTime">

<xs:annotation>

<xs:documentation>from, to in datetime format in UTC time</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="toDate" type="xs:dateTime" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getMutationsResponseType">

<xs:annotation>

<xs:documentation>type decl for getMutations() return</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="recCount" type="tns:recCountType">

<xs:annotation>

<xs:documentation>Indicate how many records are available/were found.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="refdataReturn" type="tns:refdataReturnType" nillable="true" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getDataXMLType">

<xs:annotation>

<xs:documentation>type decl for getDataXML() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="reftype" type="tns:reftypeType"/>

<xs:element name="subcode" type="xs:string" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getDataXMLResponseType">

<xs:annotation>

<xs:documentation>type decl for getDataXML() return</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="recCount" type="tns:recCountType">

<xs:annotation>

<xs:documentation>Indicates how many records are available/were found.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="refdataReturn" type="tns:refdataReturnType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getRisDataXMLType">

<xs:annotation>

<xs:documentation>type decl for getRisDataXML() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="subcode" minOccurs="0">

<xs:simpleType>

<xs:restriction base="xs:string"/>

</xs:simpleType>

</xs:element>

<xs:element name="funcode" minOccurs="0">

<xs:simpleType>

<xs:restriction base="xs:string"/>

</xs:simpleType>

</xs:element>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="getRisDataXMLResponseType">

<xs:annotation>

<xs:documentation>type decl for getRisDataXML() return</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="recCount" type="tns:recCountType">

<xs:annotation>

<xs:documentation>Indicates how many records are available/were found.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="risdataReturn" type="tns:ris\_idxType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="mutateDataXMLType">

<xs:annotation>

<xs:documentation>type decl for mutateDataXML() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="username" type="tns:userNameType"/>

<xs:element name="password" type="tns:passwordType"/>

<xs:element name="reftype" type="tns:reftypeType"/>

<xs:element name="refData" type="tns:refdataReturnType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="mutateDataXMLResponseType">

<xs:annotation>

<xs:documentation>type decl for mutateDataXML() return</xs:documentation>

</xs:annotation>

<xs:complexContent>

<xs:extension base="tns:statReturnType"/>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="requestMutationXMLType">

<xs:annotation>

<xs:documentation>type decl for requestMutationXML() call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="username" type="tns:userNameType"/>

<xs:element name="password" type="tns:passwordType"/>

<xs:element name="reftype" type="tns:reftypeType"/>

<xs:element name="inUserList" type="xs:boolean"/>

<xs:element name="refDataReqMut" type="tns:refdataReqMutType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="requestMutationXMLResponseType">

<xs:annotation>

<xs:documentation>type decl for requestMutationXML() return</xs:documentation>

</xs:annotation>

<xs:complexContent>

<xs:extension base="tns:statReturnType"/>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="exceptionType">

<xs:annotation>

<xs:documentation>exception type (as used in soap webservice fault parameter)</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="exception" type="xs:integer" minOccurs="0">

<xs:annotation>

<xs:documentation>exception code</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="message" type="xs:string" minOccurs="0">

<xs:annotation>

<xs:documentation>exception message in English</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="refdataReturnType">

<xs:annotation>

<xs:documentation>refdata (records) type as returned by webservice functions (mandatory fields are always (must be) filled in). This structure MUST MATCH with refdataReqMutType!!! (except for nts\_data)</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="reftype">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:simpleContent>

<xs:extension base="tns:reftypeType"/>

</xs:simpleContent>

</xs:complexType>

</xs:element>

<xs:choice>

<xs:element name="ris\_idx" type="tns:ris\_idxType"/>

<xs:element name="eri\_location" type="tns:eri\_locationType"/>

<xs:element name="eri\_hscode" type="tns:eri\_hscodeType"/>

<xs:element name="eri\_adncode" type="tns:eri\_adncodeType"/>

<xs:element name="eri\_conttype" type="tns:eri\_conttypeType"/>

<xs:element name="eri\_packtype" type="tns:eri\_packtypeType"/>

<xs:element name="eri\_shiptype" type="tns:eri\_shiptypeType"/>

<xs:element name="eri\_country" type="tns:eri\_countryType"/>

<xs:element name="er\_erireportingpoint" type="tns:er\_erireportingpointType"/>

<xs:element name="er\_locsreppoints" type="tns:er\_locsreppointsType"/>

<xs:element name="er\_eriprovider" type="tns:er\_eriproviderType"/>

<xs:element name="nts\_data" type="tns:nts\_dataType">

<xs:annotation>

<xs:documentation>NtS element is only available in the return refdata structure.

It is not available in the Req Mutation datastructure because we can not mutate nts data (not allowed)!

This reftype (nts\_data) is NOT supported by NLRDMS!</xs:documentation>

</xs:annotation>

</xs:element>

</xs:choice>

</xs:sequence>

</xs:complexType>

<xs:complexType name="refdataReqMutType">

<xs:annotation>

<xs:documentation>refdata (records) type for REQUESTING mutations (minimal mandatory set, only code en description required).

This structure (on field basis) MUST match with refdataReturnType!!!

(except for NtS data, which can not be mutated Yet!)</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="reftype" type="tns:reftypeType">

<xs:annotation>

<xs:documentation>BE SURE to edit the basic (simple)types, DONT edit any of the subelements from here!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:choice>

<xs:element name="ris\_idx" type="tns:ris\_idxReqMutType"/>

<xs:element name="eri\_location" type="tns:eri\_locationReqMutType"/>

<xs:element name="eri\_hscode" type="tns:eri\_hscodeReqMutType"/>

<xs:element name="eri\_adncode" type="tns:eri\_adncodeReqMutType"/>

<xs:element name="eri\_conttype" type="tns:eri\_conttypeReqMutType"/>

<xs:element name="eri\_packtype" type="tns:eri\_packtypeReqMutType"/>

<xs:element name="eri\_shiptype" type="tns:eri\_shiptypeReqMutType"/>

<xs:element name="eri\_country" type="tns:eri\_countryReqMutType"/>

<xs:element name="er\_erireportingpoint" type="tns:er\_erireportingpointReqMutType"/>

<xs:element name="er\_locsreppoints" type="tns:er\_locsreppointsReqMutType"/>

<xs:element name="er\_eriprovider" type="tns:er\_eriproviderReqMutType"/>

</xs:choice>

</xs:sequence>

</xs:complexType>

<xs:complexType name="mutatedTableReturnType">

<xs:annotation>

<xs:documentation>return type to specify mutated table(s) + recCount in given periode (return of GetMutatedTables)</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="refType" type="tns:reftypeType">

<xs:annotation>

<xs:documentation>This table (refType) has mutations (>0)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="recCount" type="tns:recCountType">

<xs:annotation>

<xs:documentation>Indicates how many (>0) mutations (records) are available for THIS table.</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="statReturnType">

<xs:annotation>

<xs:documentation>status return type for mutation functions, returning the status of the corresponding function call</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="err">

<xs:annotation>

<xs:documentation>0=ok, 1=warning, 2=error</xs:documentation>

</xs:annotation>

<xs:simpleType>

<xs:restriction base="xs:integer">

<xs:enumeration value="0"/>

<xs:enumeration value="1"/>

<xs:enumeration value="2"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="message" minOccurs="0">

<xs:annotation>

<xs:documentation>Return (info/error) message in English</xs:documentation>

</xs:annotation>

</xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="ris\_idxType">

<xs:annotation>

<xs:documentation>ris index datatype. Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="risidxCode" type="tns:risCodeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:risCodeType" minOccurs="0"/>

<xs:element name="unlocCC" type="tns:unlocCCType"/>

<xs:element name="unlocLC" type="tns:unlocLCType"/>

<xs:element name="wwsectCode" type="tns:fwCodeType"/>

<xs:element name="objcode" type="tns:risobjCodeType"/>

<xs:element name="hectomt" type="tns:kmCodeType"/>

<xs:element name="objfunc" type="tns:risobjfuncType"/>

<xs:element name="objname" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="locname" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="wwname">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="routename" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="relrisindex" type="tns:risCodeType" minOccurs="0"/>

<xs:element name="sectionnode" type="tns:rissectnodeType" minOccurs="0"/>

<xs:element name="lat" type="tns:latlonType"/>

<xs:element name="lon" type="tns:latlonType"/>

<xs:element name="relenc" type="tns:risrelencType" minOccurs="0"/>

<xs:element name="comminfo" type="tns:riscomminfoType" minOccurs="0"/>

<xs:element name="gaugeid" type="tns:risgaugeidType" minOccurs="0"/>

<xs:element name="vesselconvlength" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvbreadth" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvdraught" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvairdraught" type="tns:dimType" minOccurs="0"/>

<xs:element name="availablelength" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearancewidth" type="tns:dimType" minOccurs="0"/>

<xs:element name="availabledepth" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearanceheight" type="tns:dimType" minOccurs="0"/>

<xs:element name="applicabilityfromkm" type="tns:risapplikmType" minOccurs="0"/>

<xs:element name="applicabilitytokm" type="tns:risapplikmType" minOccurs="0"/>

<xs:element name="reflevel1code" type="tns:risreflevelcodeType" minOccurs="0"/>

<xs:element name="reflevel1value" type="tns:risreflevelvalueType" minOccurs="0"/>

<xs:element name="reflevel2code" type="tns:risreflevelcodeType" minOccurs="0"/>

<xs:element name="reflevel2value" type="tns:risreflevelvalueType" minOccurs="0"/>

<xs:element name="reflevel3code" type="tns:risreflevelcodeType" minOccurs="0"/>

<xs:element name="reflevel3value" type="tns:risreflevelvalueType" minOccurs="0"/>

<xs:element name="zeropoint" type="tns:riszeropointType" minOccurs="0"/>

<xs:element name="geodref" type="tns:risgeodrefType" minOccurs="0"/>

<xs:element name="catoftimesched" type="tns:riscatoftimeschedType" minOccurs="0"/>

<xs:element name="forshiptype" type="tns:risforshiptypeType" minOccurs="0"/>

<xs:element name="foruseofship" type="tns:risuseofshipType" minOccurs="0"/>

<xs:element name="lnktoextXMfiletimesched" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="lnktoextXMfilepastime" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="vesselconvlength2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvbreadth2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvdraught2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvairdraught2" type="tns:dimType" minOccurs="0"/>

<xs:element name="availablelength2" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearancewidth2" type="tns:dimType" minOccurs="0"/>

<xs:element name="availabledepth2" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearanceheight2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvlength3" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvbreadth3" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvdraught3" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvairdraught3" type="tns:dimType" minOccurs="0"/>

<xs:element name="availablelength3" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearancewidth3" type="tns:dimType" minOccurs="0"/>

<xs:element name="availabledepth3" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearanceheight3" type="tns:dimType" minOccurs="0"/>

<xs:element name="catoftimesched2" type="tns:riscatoftimeschedType" minOccurs="0"/>

<xs:element name="forshiptype2" type="tns:risforshiptypeType" minOccurs="0"/>

<xs:element name="foruseofship2" type="tns:risuseofshipType" minOccurs="0"/>

<xs:element name="lnktoextXMfiletimesched2" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="lnktoextXMfilepastime2" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="catoftimesched3" type="tns:riscatoftimeschedType" minOccurs="0"/>

<xs:element name="forshiptype3" type="tns:risforshiptypeType" minOccurs="0"/>

<xs:element name="foruseofship3" type="tns:risuseofshipType" minOccurs="0"/>

<xs:element name="lnktoextXMfiletimesched3" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="lnktoextXMfilepastime3" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="startdate" type="tns:lastupdateType"/>

<xs:element name="enddate" type="tns:lastupdateType" minOccurs="0"/>

<xs:element name="infodate" type="tns:lastupdateType"/>

<xs:element name="risidxRemarks" type="tns:remarksType" minOccurs="0">

<xs:annotation>

<xs:documentation>specific Ris idx remarks, reasons for change (descriptions as received with the ris idx)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0">

<xs:annotation>

<xs:documentation>remarks regarding the maintance of the ris idx</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_locationType">

<xs:annotation>

<xs:documentation>location datatype (ERI loc code according ERI). Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="erilocCode" type="tns:erilocCodeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:erilocCodeType" minOccurs="0"/>

<xs:element name="risCode" type="tns:risCodeType" minOccurs="0">

<xs:annotation>

<xs:documentation>Related (nearest or corresponding) RIS-idx (ISRS) code</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="loCode" type="tns:loCodeType"/>

<xs:element name="fwCode" type="tns:fwCodeType" minOccurs="0"/>

<xs:element name="termCode" type="tns:termCodeType" minOccurs="0"/>

<xs:element name="kmCode" type="tns:kmCodeType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:erilocNameType"/>

<xs:element name="NL" type="tns:erilocNameType"/>

<xs:element name="DE" type="tns:erilocNameType"/>

<xs:element name="FR" type="tns:erilocNameType"/>

<xs:element name="EN" type="tns:erilocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="locType" type="tns:locTypeType"/>

<xs:element name="termname" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="NL" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:erilocNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="quayFrom" type="tns:quayType" minOccurs="0"/>

<xs:element name="quayTo" type="tns:quayType" minOccurs="0"/>

<xs:element name="termType" type="tns:locTypeType"/>

<xs:element name="lat" type="tns:latlonType"/>

<xs:element name="lon" type="tns:latlonType"/>

<xs:element name="ivsVTSCode" type="tns:providerIdType">

<xs:annotation>

<xs:documentation>VTS/IVS (code) acting as the default reporting (receiving) point (when departing from this llocation)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="exits" type="tns:locExitsType">

<xs:annotation>

<xs:documentation>Nr of possible exits (routes) when departing from this location >1 means a reporting or routepoint has to be specified.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_hscodeType">

<xs:annotation>

<xs:documentation>hs code datatype (HS code according Customs). Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="hsCode" type="tns:hsCodeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:hsCodeType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:goodNameType"/>

<xs:element name="NL" type="tns:goodNameType"/>

<xs:element name="DE" type="tns:goodNameType"/>

<xs:element name="FR" type="tns:goodNameType"/>

<xs:element name="EN" type="tns:goodNameType"/>

<xs:element name="BG" type="tns:goodNameType"/>

<xs:element name="CS" type="tns:goodNameType"/>

<xs:element name="DA" type="tns:goodNameType"/>

<xs:element name="EL" type="tns:goodNameType"/>

<xs:element name="ES" type="tns:goodNameType"/>

<xs:element name="ET" type="tns:goodNameType"/>

<xs:element name="FI" type="tns:goodNameType"/>

<xs:element name="HU" type="tns:goodNameType"/>

<xs:element name="IT" type="tns:goodNameType"/>

<xs:element name="LT" type="tns:goodNameType"/>

<xs:element name="LV" type="tns:goodNameType"/>

<xs:element name="PL" type="tns:goodNameType"/>

<xs:element name="PT" type="tns:goodNameType"/>

<xs:element name="RO" type="tns:goodNameType"/>

<xs:element name="SK" type="tns:goodNameType"/>

<xs:element name="SL" type="tns:goodNameType"/>

<xs:element name="SV" type="tns:goodNameType"/>

<xs:element name="HR" type="tns:goodNameType"/>

<xs:element name="RU" type="tns:goodNameType"/>

<xs:element name="SR" type="tns:goodNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="nstCode" type="tns:nstCodeType" minOccurs="0"/>

<xs:element name="nstName" type="tns:goodNameType" minOccurs="0">

<xs:annotation>

<xs:documentation>English (NST2007) description.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_adncodeType">

<xs:annotation>

<xs:documentation>adn code datatype (ADN(R) according CCNR). Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="adnCode" type="tns:adnCodeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:adnCodeType" minOccurs="0"/>

<xs:element name="unnr" type="tns:unnrType"/>

<xs:element name="adnClass" type="tns:adnClassType"/>

<xs:element name="adnClassif" type="tns:adnClassType" minOccurs="0"/>

<xs:element name="imoClass" type="tns:adnClassType" minOccurs="0"/>

<xs:element name="packGrp" type="tns:packGrpType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:goodDgsNameType"/>

<xs:element name="NL" type="tns:goodDgsNameType"/>

<xs:element name="DE" type="tns:goodDgsNameType"/>

<xs:element name="FR" type="tns:goodDgsNameType"/>

<xs:element name="EN" type="tns:goodDgsNameType"/>

<xs:element name="BG" type="tns:goodDgsNameType"/>

<xs:element name="CS" type="tns:goodDgsNameType"/>

<xs:element name="DA" type="tns:goodDgsNameType"/>

<xs:element name="EL" type="tns:goodDgsNameType"/>

<xs:element name="ES" type="tns:goodDgsNameType"/>

<xs:element name="ET" type="tns:goodDgsNameType"/>

<xs:element name="FI" type="tns:goodDgsNameType"/>

<xs:element name="HU" type="tns:goodDgsNameType"/>

<xs:element name="IT" type="tns:goodDgsNameType"/>

<xs:element name="LT" type="tns:goodDgsNameType"/>

<xs:element name="LV" type="tns:goodDgsNameType"/>

<xs:element name="PL" type="tns:goodDgsNameType"/>

<xs:element name="PT" type="tns:goodDgsNameType"/>

<xs:element name="RO" type="tns:goodDgsNameType"/>

<xs:element name="SK" type="tns:goodDgsNameType"/>

<xs:element name="SL" type="tns:goodDgsNameType"/>

<xs:element name="SV" type="tns:goodDgsNameType"/>

<xs:element name="HR" type="tns:goodDgsNameType"/>

<xs:element name="RU" type="tns:goodDgsNameType"/>

<xs:element name="SR" type="tns:goodDgsNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="syn">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:goodSynType"/>

<xs:element name="NL" type="tns:goodSynType"/>

<xs:element name="DE" type="tns:goodSynType"/>

<xs:element name="FR" type="tns:goodSynType"/>

<xs:element name="EN" type="tns:goodSynType"/>

<xs:element name="BG" type="tns:goodSynType"/>

<xs:element name="CS" type="tns:goodSynType"/>

<xs:element name="DA" type="tns:goodSynType"/>

<xs:element name="EL" type="tns:goodSynType"/>

<xs:element name="ES" type="tns:goodSynType"/>

<xs:element name="ET" type="tns:goodSynType"/>

<xs:element name="FI" type="tns:goodSynType"/>

<xs:element name="HU" type="tns:goodSynType"/>

<xs:element name="IT" type="tns:goodSynType"/>

<xs:element name="LT" type="tns:goodSynType"/>

<xs:element name="LV" type="tns:goodSynType"/>

<xs:element name="PL" type="tns:goodSynType"/>

<xs:element name="PT" type="tns:goodSynType"/>

<xs:element name="RO" type="tns:goodSynType"/>

<xs:element name="SK" type="tns:goodSynType"/>

<xs:element name="SL" type="tns:goodSynType"/>

<xs:element name="SV" type="tns:goodSynType"/>

<xs:element name="HR" type="tns:goodSynType"/>

<xs:element name="RU" type="tns:goodSynType"/>

<xs:element name="SR" type="tns:goodSynType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="hsCode" type="tns:hsCodeType" minOccurs="0"/>

<xs:element name="nstCode" type="tns:nstCodeType" minOccurs="0"/>

<xs:element name="conesBulk" type="tns:conesType">

<xs:annotation>

<xs:documentation>0, 1, 2, 3, V</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="conesTank" type="tns:conesType"/>

<xs:element name="conesCont" type="tns:conesType"/>

<xs:element name="flagsBulk" type="tns:flagsType">

<xs:annotation>

<xs:documentation>0, B, V</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="flagsTank" type="tns:flagsType"/>

<xs:element name="warnCard" type="tns:warnCardType"/>

<xs:element name="warnId" type="tns:warnIdType"/>

<xs:element name="label1" type="tns:adnLabelType"/>

<xs:element name="label2" type="tns:adnLabelType"/>

<xs:element name="label3" type="tns:adnLabelType"/>

<xs:element name="label4" type="tns:adnLabelType"/>

<xs:element name="aquaticToxicity" type="tns:aquaticToxicityType"/>

<xs:element name="chemicalStability" type="tns:chemicalStabiltyType"/>

<xs:element name="cmrProperties" type="tns:cmrPropertyType"/>

<xs:element name="floater" type="tns:floatersinkerType"/>

<xs:element name="maxWeightCodeInlandBulkCont" type="tns:signalCodeType"/>

<xs:element name="maxWeightInlandBulkCont" type="tns:weightType">

<xs:annotation>

<xs:documentation>max weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="signalCodeInlandCont" type="tns:signalCodeType"/>

<xs:element name="minWeightInlandCont" type="tns:weightType">

<xs:annotation>

<xs:documentation>min weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="signalCodeInlandExcemp" type="tns:signalCodeType"/>

<xs:element name="minWeightInlandExcemp" type="tns:weightType">

<xs:annotation>

<xs:documentation>min weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="signalCodeSeaBulk" type="tns:signalCodeType"/>

<xs:element name="minWeightSeaBulk" type="tns:weightType">

<xs:annotation>

<xs:documentation>min weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_conttypeType">

<xs:annotation>

<xs:documentation>conttype datatype (ISO 6346). Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="contId" type="tns:contIdType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:contIdType" minOccurs="0"/>

<xs:element name="length" type="tns:dimType">

<xs:annotation>

<xs:documentation>dimensions in cm</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="width" type="tns:dimType"/>

<xs:element name="height" type="tns:dimType"/>

<xs:element name="weight" type="tns:weightType">

<xs:annotation>

<xs:documentation>empty weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:contNameType"/>

<xs:element name="NL" type="tns:contNameType"/>

<xs:element name="DE" type="tns:contNameType"/>

<xs:element name="FR" type="tns:contNameType"/>

<xs:element name="EN" type="tns:contNameType"/>

<xs:element name="BG" type="tns:contNameType"/>

<xs:element name="CS" type="tns:contNameType"/>

<xs:element name="DA" type="tns:contNameType"/>

<xs:element name="EL" type="tns:contNameType"/>

<xs:element name="ES" type="tns:contNameType"/>

<xs:element name="ET" type="tns:contNameType"/>

<xs:element name="FI" type="tns:contNameType"/>

<xs:element name="HU" type="tns:contNameType"/>

<xs:element name="IT" type="tns:contNameType"/>

<xs:element name="LT" type="tns:contNameType"/>

<xs:element name="LV" type="tns:contNameType"/>

<xs:element name="PL" type="tns:contNameType"/>

<xs:element name="PT" type="tns:contNameType"/>

<xs:element name="RO" type="tns:contNameType"/>

<xs:element name="SK" type="tns:contNameType"/>

<xs:element name="SL" type="tns:contNameType"/>

<xs:element name="SV" type="tns:contNameType"/>

<xs:element name="HR" type="tns:contNameType"/>

<xs:element name="RU" type="tns:contNameType"/>

<xs:element name="SR" type="tns:contNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_packtypeType">

<xs:annotation>

<xs:documentation>inner package datatype (UN Recom 21). Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="packId" type="tns:packIdType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:packIdType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:packNameType"/>

<xs:element name="NL" type="tns:packNameType"/>

<xs:element name="DE" type="tns:packNameType"/>

<xs:element name="FR" type="tns:packNameType"/>

<xs:element name="EN" type="tns:packNameType"/>

<xs:element name="BG" type="tns:packNameType"/>

<xs:element name="CS" type="tns:packNameType"/>

<xs:element name="DA" type="tns:packNameType"/>

<xs:element name="EL" type="tns:packNameType"/>

<xs:element name="ES" type="tns:packNameType"/>

<xs:element name="ET" type="tns:packNameType"/>

<xs:element name="FI" type="tns:packNameType"/>

<xs:element name="HU" type="tns:packNameType"/>

<xs:element name="IT" type="tns:packNameType"/>

<xs:element name="LT" type="tns:packNameType"/>

<xs:element name="LV" type="tns:packNameType"/>

<xs:element name="PL" type="tns:packNameType"/>

<xs:element name="PT" type="tns:packNameType"/>

<xs:element name="RO" type="tns:packNameType"/>

<xs:element name="SK" type="tns:packNameType"/>

<xs:element name="SL" type="tns:packNameType"/>

<xs:element name="SV" type="tns:packNameType"/>

<xs:element name="HR" type="tns:packNameType"/>

<xs:element name="RU" type="tns:packNameType"/>

<xs:element name="SR" type="tns:packNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_shiptypeType">

<xs:annotation>

<xs:documentation>shiptype (single and transport combos) datatype (UN Recom 28). Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="shipType" type="tns:shipTypeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:shipTypeType" minOccurs="0"/>

<xs:element name="dvkType" type="tns:dvkTypeType"/>

<xs:element name="craftType" type="tns:craftTypeType"/>

<xs:element name="combination" type="tns:combinationFlagType">

<xs:annotation>

<xs:documentation>True = transport cominination type, false=single ship</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:shiptypeNameType"/>

<xs:element name="NL" type="tns:shiptypeNameType"/>

<xs:element name="DE" type="tns:shiptypeNameType"/>

<xs:element name="FR" type="tns:shiptypeNameType"/>

<xs:element name="EN" type="tns:shiptypeNameType"/>

<xs:element name="BG" type="tns:shiptypeNameType"/>

<xs:element name="CS" type="tns:shiptypeNameType"/>

<xs:element name="DA" type="tns:shiptypeNameType"/>

<xs:element name="EL" type="tns:shiptypeNameType"/>

<xs:element name="ES" type="tns:shiptypeNameType"/>

<xs:element name="ET" type="tns:shiptypeNameType"/>

<xs:element name="FI" type="tns:shiptypeNameType"/>

<xs:element name="HU" type="tns:shiptypeNameType"/>

<xs:element name="IT" type="tns:shiptypeNameType"/>

<xs:element name="LT" type="tns:shiptypeNameType"/>

<xs:element name="LV" type="tns:shiptypeNameType"/>

<xs:element name="PL" type="tns:shiptypeNameType"/>

<xs:element name="PT" type="tns:shiptypeNameType"/>

<xs:element name="RO" type="tns:shiptypeNameType"/>

<xs:element name="SK" type="tns:shiptypeNameType"/>

<xs:element name="SL" type="tns:shiptypeNameType"/>

<xs:element name="SV" type="tns:shiptypeNameType"/>

<xs:element name="HR" type="tns:shiptypeNameType"/>

<xs:element name="RU" type="tns:shiptypeNameType"/>

<xs:element name="SR" type="tns:shiptypeNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_countryType">

<xs:annotation>

<xs:documentation>un country datatype (ISO 3166). Be sure to edit the basic types, dont edit the subelements here!!! Also DONT forget to add new field to the xxxReqMutType datatypes.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="countryCode" type="tns:countryCodeType">

<xs:annotation>

<xs:documentation>UN country ISO code 2 chars. Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:countryCodeType" minOccurs="0"/>

<xs:element name="cbsCode" type="tns:cbscountryCodeType"/>

<xs:element name="countryIsoCode3" type="tns:countryCode3Type"/>

<xs:element name="countryIsoNum" type="tns:countryIsoNumType"/>

<xs:element name="lloydsFlag" type="tns:lloydsflagType"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:countryNameType"/>

<xs:element name="NL" type="tns:countryNameType"/>

<xs:element name="DE" type="tns:countryNameType"/>

<xs:element name="FR" type="tns:countryNameType"/>

<xs:element name="EN" type="tns:countryNameType"/>

<xs:element name="BG" type="tns:countryNameType"/>

<xs:element name="CS" type="tns:countryNameType"/>

<xs:element name="DA" type="tns:countryNameType"/>

<xs:element name="EL" type="tns:countryNameType"/>

<xs:element name="ES" type="tns:countryNameType"/>

<xs:element name="ET" type="tns:countryNameType"/>

<xs:element name="FI" type="tns:countryNameType"/>

<xs:element name="HU" type="tns:countryNameType"/>

<xs:element name="IT" type="tns:countryNameType"/>

<xs:element name="LT" type="tns:countryNameType"/>

<xs:element name="LV" type="tns:countryNameType"/>

<xs:element name="PL" type="tns:countryNameType"/>

<xs:element name="PT" type="tns:countryNameType"/>

<xs:element name="RO" type="tns:countryNameType"/>

<xs:element name="SK" type="tns:countryNameType"/>

<xs:element name="SL" type="tns:countryNameType"/>

<xs:element name="SV" type="tns:countryNameType"/>

<xs:element name="HR" type="tns:countryNameType"/>

<xs:element name="RU" type="tns:countryNameType"/>

<xs:element name="SR" type="tns:countryNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="er\_erireportingpointType">

<xs:annotation>

<xs:documentation>The ERI reporting points a usercan select to send in his ERI msgs (each point relates to 1 provider=receiving authority VTS system).</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="repPointId" type="tns:repPointIdType">

<xs:annotation>

<xs:documentation>Unique ERI reporting point id (first 3 are significant to support aliases)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:repPointIdType" minOccurs="0"/>

<xs:element name="risCode" type="tns:risCodeType">

<xs:annotation>

<xs:documentation>Related (nearest or corresponding) RIS-idx (ISRS) code for the Reportingpoint</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="name">

<xs:annotation>

<xs:documentation>Supports only the basic ERI languages</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:repPointNameType"/>

<xs:element name="NL" type="tns:repPointNameType"/>

<xs:element name="DE" type="tns:repPointNameType"/>

<xs:element name="FR" type="tns:repPointNameType"/>

<xs:element name="EN" type="tns:repPointNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="eriProviderId" type="tns:providerIdType">

<xs:annotation>

<xs:documentation>Provder ID of the system(VTS/IVS) acting as the receiving system for this Reporting point.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="er\_locsreppointsType">

<xs:annotation>

<xs:documentation>The link between ERI locations and possible/available Reportingpoints</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="erilocCode" type="tns:erilandlocCodeType">

<xs:annotation>

<xs:documentation>Unique Land of ERI loc code (2 of 5-20 lang)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="erilocCodeOld" type="tns:erilandlocCodeType" minOccurs="0">

<xs:annotation>

<xs:documentation>De oude (te wijzigen) code is altijd een combinatie van erilocCode en repPoint</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="repPointId" type="tns:repPointIdType">

<xs:annotation>

<xs:documentation>Unique ERI reporting point id</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="repPointIdOld" type="tns:repPointIdType" minOccurs="0"/>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="er\_eriproviderType">

<xs:annotation>

<xs:documentation>The eri providers (systems and their address info), that can recieve ERI msgs</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="providerId" type="tns:providerIdType">

<xs:annotation>

<xs:documentation>Provder ID of the system(VTS/IVS) acting as the receiving system for ERI msgs.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:providerIdType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:providerNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="systemId" type="tns:systemIdType">

<xs:annotation>

<xs:documentation>SystemId (within the mailbox)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="systemEmail" type="tns:emailAddressType"/>

<xs:element name="contactPers" type="tns:contactNameType" minOccurs="0"/>

<xs:element name="contactTel" type="tns:telnrType" minOccurs="0"/>

<xs:element name="contactFax" type="tns:telnrType" minOccurs="0"/>

<xs:element name="contactAddress" type="tns:contactAddressType" minOccurs="0"/>

<xs:element name="contactPostalCode" type="tns:contactPostalCodeType" minOccurs="0"/>

<xs:element name="contactCity" type="tns:contactCitynameType" minOccurs="0"/>

<xs:element name="contactCountrycode" type="tns:countryCodeType">

<xs:annotation>

<xs:documentation>ISO3166 country code</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="receiveParty" type="tns:providerOptionsType">

<xs:annotation>

<xs:documentation>Can party receive msgs</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="invoiceParty" type="tns:providerOptionsType">

<xs:annotation>

<xs:documentation>Does party support (needs) Invoice info</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="responseParty" type="tns:providerOptionsType">

<xs:annotation>

<xs:documentation>Does party support responses</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="erinotMsgVersion" type="tns:msgVersionType">

<xs:annotation>

<xs:documentation>msgversion 0=not applicable</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="bermanMsgVersion" type="tns:msgVersionType"/>

<xs:element name="erimanMsgVersion" type="tns:msgVersionType"/>

<xs:element name="paxlstMsgVersion" type="tns:msgVersionType"/>

<xs:element name="erivoyMsgVersion" type="tns:msgVersionType"/>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="ris\_idxReqMutType">

<xs:annotation>

<xs:documentation>ris index minimal dataset type for requesting mutations. MUST MATCH with ris\_idxType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="risidxCode" type="tns:risCodeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:risCodeType" minOccurs="0"/>

<xs:element name="unlocCC" type="tns:unlocCCType" minOccurs="0"/>

<xs:element name="unlocLC" type="tns:unlocLCType" minOccurs="0"/>

<xs:element name="wwsectCode" type="tns:fwCodeType" minOccurs="0"/>

<xs:element name="objcode" type="tns:risobjCodeType" minOccurs="0"/>

<xs:element name="hectomt" type="tns:kmCodeType" minOccurs="0"/>

<xs:element name="objfunc" type="tns:risobjfuncType"/>

<xs:element name="objname" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="locname" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="wwname">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="routename" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:rislocNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="relrisindex" type="tns:risCodeType" minOccurs="0"/>

<xs:element name="sectionnode" type="tns:rissectnodeType" minOccurs="0"/>

<xs:element name="lat" type="tns:latlonType" minOccurs="0"/>

<xs:element name="lon" type="tns:latlonType" minOccurs="0"/>

<xs:element name="relenc" type="tns:risrelencType" minOccurs="0"/>

<xs:element name="comminfo" type="tns:riscomminfoType" minOccurs="0"/>

<xs:element name="gaugeid" type="tns:risgaugeidType" minOccurs="0"/>

<xs:element name="vesselconvlength" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvbreadth" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvdraught" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvairdraught" type="tns:dimType" minOccurs="0"/>

<xs:element name="availablelength" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearancewidth" type="tns:dimType" minOccurs="0"/>

<xs:element name="availabledepth" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearanceheight" type="tns:dimType" minOccurs="0"/>

<xs:element name="applicabilityfromkm" type="tns:risapplikmType" minOccurs="0"/>

<xs:element name="applicabilitytokm" type="tns:risapplikmType" minOccurs="0"/>

<xs:element name="reflevel1code" type="tns:risreflevelcodeType" minOccurs="0"/>

<xs:element name="reflevel1value" type="tns:risreflevelvalueType" minOccurs="0"/>

<xs:element name="reflevel2code" type="tns:risreflevelcodeType" minOccurs="0"/>

<xs:element name="reflevel2value" type="tns:risreflevelvalueType" minOccurs="0"/>

<xs:element name="reflevel3code" type="tns:risreflevelcodeType" minOccurs="0"/>

<xs:element name="reflevel3value" type="tns:risreflevelvalueType" minOccurs="0"/>

<xs:element name="zeropoint" type="tns:riszeropointType" minOccurs="0"/>

<xs:element name="geodref" type="tns:risgeodrefType" minOccurs="0"/>

<xs:element name="catoftimesched" type="tns:riscatoftimeschedType" minOccurs="0"/>

<xs:element name="forshiptype" type="tns:risforshiptypeType" minOccurs="0"/>

<xs:element name="foruseofship" type="tns:risuseofshipType" minOccurs="0"/>

<xs:element name="lnktoextXMfiletimesched" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="lnktoextXMfilepastime" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="vesselconvlength2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvbreadth2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvdraught2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvairdraught2" type="tns:dimType" minOccurs="0"/>

<xs:element name="availablelength2" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearancewidth2" type="tns:dimType" minOccurs="0"/>

<xs:element name="availabledepth2" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearanceheight2" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvlength3" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvbreadth3" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvdraught3" type="tns:dimType" minOccurs="0"/>

<xs:element name="vesselconvairdraught3" type="tns:dimType" minOccurs="0"/>

<xs:element name="availablelength3" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearancewidth3" type="tns:dimType" minOccurs="0"/>

<xs:element name="availabledepth3" type="tns:dimType" minOccurs="0"/>

<xs:element name="clearanceheight3" type="tns:dimType" minOccurs="0"/>

<xs:element name="catoftimesched2" type="tns:riscatoftimeschedType" minOccurs="0"/>

<xs:element name="forshiptype2" type="tns:risforshiptypeType" minOccurs="0"/>

<xs:element name="foruseofship2" type="tns:risuseofshipType" minOccurs="0"/>

<xs:element name="lnktoextXMfiletimesched2" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="lnktoextXMfilepastime2" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="catoftimesched3" type="tns:riscatoftimeschedType" minOccurs="0"/>

<xs:element name="forshiptype3" type="tns:risforshiptypeType" minOccurs="0"/>

<xs:element name="foruseofship3" type="tns:risuseofshipType" minOccurs="0"/>

<xs:element name="lnktoextXMfiletimesched3" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="lnktoextXMfilepastime3" type="tns:rislinkextxmlType" minOccurs="0"/>

<xs:element name="startdate" type="tns:lastupdateType" minOccurs="0"/>

<xs:element name="enddate" type="tns:lastupdateType" minOccurs="0"/>

<xs:element name="infodate" type="tns:lastupdateType" minOccurs="0"/>

<xs:element name="risidxRemarks" type="tns:remarksType" minOccurs="0">

<xs:annotation>

<xs:documentation>specific Ris idx remarks/reasons for change, as received with the ris idx</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0">

<xs:annotation>

<xs:documentation>remarks regarding the maintance of the ris idx</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_locationReqMutType">

<xs:annotation>

<xs:documentation>location minimal set. Must MATCH with eri\_locationType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="erilocCode" type="tns:erilocCodeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:erilocCodeType" minOccurs="0"/>

<xs:element name="risCode" type="tns:risCodeType" minOccurs="0"/>

<xs:element name="loCode" type="tns:loCodeType" minOccurs="0"/>

<xs:element name="fwCode" type="tns:fwCodeType" minOccurs="0"/>

<xs:element name="termCode" type="tns:termCodeType" minOccurs="0"/>

<xs:element name="kmCode" type="tns:kmCodeType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:erilocNameType"/>

<xs:element name="NL" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:erilocNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="locType" type="tns:locTypeType" minOccurs="0"/>

<xs:element name="termname" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="NL" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:erilocNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:erilocNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="quayFrom" type="tns:quayType" minOccurs="0"/>

<xs:element name="quayTo" type="tns:quayType" minOccurs="0"/>

<xs:element name="termType" type="tns:locTypeType" minOccurs="0"/>

<xs:element name="lat" type="tns:latlonType" minOccurs="0"/>

<xs:element name="lon" type="tns:latlonType" minOccurs="0"/>

<xs:element name="ivsVTSCode" type="tns:providerIdType" minOccurs="0"/>

<xs:element name="exits" type="tns:locExitsType" minOccurs="0"/>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_hscodeReqMutType">

<xs:annotation>

<xs:documentation>hscode minimal set. Must MATCH with eri\_hscodeType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="hsCode" type="tns:hsCodeType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:hsCodeType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:goodNameType"/>

<xs:element name="NL" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="BG" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="CS" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="DA" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="EL" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="ES" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="ET" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="FI" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="HU" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="IT" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="LT" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="LV" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="PL" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="PT" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="RO" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="SK" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="SL" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="SV" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="HR" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="RU" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="SR" type="tns:goodNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="nstCode" type="tns:nstCodeType" minOccurs="0"/>

<xs:element name="nstName" type="tns:goodNameType" minOccurs="0"/>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_adncodeReqMutType">

<xs:annotation>

<xs:documentation>adn minimal set. Must MATCH with eri\_adncodeType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="adnCode" type="tns:adnCodeType" minOccurs="0">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:adnCodeType" minOccurs="0"/>

<xs:element name="unnr" type="tns:unnrType"/>

<xs:element name="adnClass" type="tns:adnClassType" minOccurs="0"/>

<xs:element name="adnClassif" type="tns:adnClassType" minOccurs="0"/>

<xs:element name="imoClass" type="tns:adnClassType" minOccurs="0"/>

<xs:element name="packGrp" type="tns:packGrpType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:goodDgsNameType"/>

<xs:element name="NL" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="BG" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="CS" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="DA" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="EL" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="ES" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="ET" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="FI" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="HU" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="IT" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="LT" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="LV" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="PL" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="PT" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="RO" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="SK" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="SL" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="SV" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="HR" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="RU" type="tns:goodDgsNameType" minOccurs="0"/>

<xs:element name="SR" type="tns:goodDgsNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="syn" minOccurs="0">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="NL" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="DE" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="FR" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="EN" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="BG" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="CS" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="DA" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="EL" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="ES" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="ET" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="FI" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="HU" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="IT" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="LT" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="LV" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="PL" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="PT" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="RO" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="SK" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="SL" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="SV" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="HR" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="RU" type="tns:goodSynType" minOccurs="0"/>

<xs:element name="SR" type="tns:goodSynType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="hsCode" type="tns:hsCodeType" minOccurs="0"/>

<xs:element name="nstCode" type="tns:nstCodeType" minOccurs="0"/>

<xs:element name="conesBulk" type="tns:conesType" minOccurs="0">

<xs:annotation>

<xs:documentation>0, 1, 2, 3, V (Inland)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="conesTank" type="tns:conesType" minOccurs="0"/>

<xs:element name="conesCont" type="tns:conesType" minOccurs="0"/>

<xs:element name="flagsBulk" type="tns:flagsType" minOccurs="0">

<xs:annotation>

<xs:documentation>0, B, V (Seagoing)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="flagsTank" type="tns:flagsType" minOccurs="0"/>

<xs:element name="warnCard" type="tns:warnCardType" minOccurs="0"/>

<xs:element name="warnId" type="tns:warnIdType" minOccurs="0"/>

<xs:element name="label1" type="tns:adnLabelType" minOccurs="0"/>

<xs:element name="label2" type="tns:adnLabelType" minOccurs="0"/>

<xs:element name="label3" type="tns:adnLabelType" minOccurs="0"/>

<xs:element name="label4" type="tns:adnLabelType" minOccurs="0"/>

<xs:element name="aquaticToxicity" type="tns:aquaticToxicityType" minOccurs="0"/>

<xs:element name="chemicalStability" type="tns:chemicalStabiltyType" minOccurs="0"/>

<xs:element name="cmrProperties" type="tns:cmrPropertyType" minOccurs="0"/>

<xs:element name="floater" type="tns:floatersinkerType" minOccurs="0"/>

<xs:element name="maxWeightCodeInlandBulkCont" type="tns:signalCodeType" minOccurs="0"/>

<xs:element name="maxWeightInlandBulkCont" type="tns:weightType" minOccurs="0">

<xs:annotation>

<xs:documentation>max weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="signalCodeInlandCont" type="tns:signalCodeType" minOccurs="0"/>

<xs:element name="minWeightInlandCont" type="tns:weightType" minOccurs="0">

<xs:annotation>

<xs:documentation>min weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="signalCodeInlandExcemp" type="tns:signalCodeType" minOccurs="0"/>

<xs:element name="minWeightInlandExcemp" type="tns:weightType" minOccurs="0">

<xs:annotation>

<xs:documentation>min weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="signalCodeSeaBulk" type="tns:signalCodeType" minOccurs="0"/>

<xs:element name="minWeightSeaBulk" type="tns:weightType" minOccurs="0">

<xs:annotation>

<xs:documentation>min weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_conttypeReqMutType">

<xs:annotation>

<xs:documentation>conttype minimal set. Must MATCH with eri\_conttypeType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="contId" type="tns:contIdType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:contIdType" minOccurs="0"/>

<xs:element name="length" type="tns:dimType" minOccurs="0">

<xs:annotation>

<xs:documentation>dimensions in cm</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="width" type="tns:dimType" minOccurs="0"/>

<xs:element name="height" type="tns:dimType" minOccurs="0"/>

<xs:element name="weight" type="tns:dimType" minOccurs="0">

<xs:annotation>

<xs:documentation>empty weight in kg</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:contNameType"/>

<xs:element name="NL" type="tns:contNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:contNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:contNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:contNameType" minOccurs="0"/>

<xs:element name="BG" type="tns:contNameType" minOccurs="0"/>

<xs:element name="CS" type="tns:contNameType" minOccurs="0"/>

<xs:element name="DA" type="tns:contNameType" minOccurs="0"/>

<xs:element name="EL" type="tns:contNameType" minOccurs="0"/>

<xs:element name="ES" type="tns:contNameType" minOccurs="0"/>

<xs:element name="ET" type="tns:contNameType" minOccurs="0"/>

<xs:element name="FI" type="tns:contNameType" minOccurs="0"/>

<xs:element name="HU" type="tns:contNameType" minOccurs="0"/>

<xs:element name="IT" type="tns:contNameType" minOccurs="0"/>

<xs:element name="LT" type="tns:contNameType" minOccurs="0"/>

<xs:element name="LV" type="tns:contNameType" minOccurs="0"/>

<xs:element name="PL" type="tns:contNameType" minOccurs="0"/>

<xs:element name="PT" type="tns:contNameType" minOccurs="0"/>

<xs:element name="RO" type="tns:contNameType" minOccurs="0"/>

<xs:element name="SK" type="tns:contNameType" minOccurs="0"/>

<xs:element name="SL" type="tns:contNameType" minOccurs="0"/>

<xs:element name="SV" type="tns:contNameType" minOccurs="0"/>

<xs:element name="HR" type="tns:contNameType" minOccurs="0"/>

<xs:element name="RU" type="tns:contNameType" minOccurs="0"/>

<xs:element name="SR" type="tns:contNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_packtypeReqMutType">

<xs:annotation>

<xs:documentation>packtype minimal set. Must MATCH with eri\_packtypeType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="packId" type="tns:packIdType">

<xs:annotation>

<xs:documentation>Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:packIdType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:packNameType"/>

<xs:element name="NL" type="tns:packNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:packNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:packNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:packNameType" minOccurs="0"/>

<xs:element name="BG" type="tns:packNameType" minOccurs="0"/>

<xs:element name="CS" type="tns:packNameType" minOccurs="0"/>

<xs:element name="DA" type="tns:packNameType" minOccurs="0"/>

<xs:element name="EL" type="tns:packNameType" minOccurs="0"/>

<xs:element name="ES" type="tns:packNameType" minOccurs="0"/>

<xs:element name="ET" type="tns:packNameType" minOccurs="0"/>

<xs:element name="FI" type="tns:packNameType" minOccurs="0"/>

<xs:element name="HU" type="tns:packNameType" minOccurs="0"/>

<xs:element name="IT" type="tns:packNameType" minOccurs="0"/>

<xs:element name="LT" type="tns:packNameType" minOccurs="0"/>

<xs:element name="LV" type="tns:packNameType" minOccurs="0"/>

<xs:element name="PL" type="tns:packNameType" minOccurs="0"/>

<xs:element name="PT" type="tns:packNameType" minOccurs="0"/>

<xs:element name="RO" type="tns:packNameType" minOccurs="0"/>

<xs:element name="SK" type="tns:packNameType" minOccurs="0"/>

<xs:element name="SL" type="tns:packNameType" minOccurs="0"/>

<xs:element name="SV" type="tns:packNameType" minOccurs="0"/>

<xs:element name="HR" type="tns:packNameType" minOccurs="0"/>

<xs:element name="RU" type="tns:packNameType" minOccurs="0"/>

<xs:element name="SR" type="tns:packNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_shiptypeReqMutType">

<xs:annotation>

<xs:documentation>shiptype minimal set. Must MATCH with eri\_shiptypeType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="shipType" type="tns:shipTypeType">

<xs:annotation>

<xs:documentation>UN rec 28. Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:shipTypeType" minOccurs="0"/>

<xs:element name="dvkType" type="tns:dvkTypeType" minOccurs="0"/>

<xs:element name="craftType" type="tns:craftTypeType" minOccurs="0"/>

<xs:element name="combination" type="tns:combinationFlagType">

<xs:annotation>

<xs:documentation>True = transport cominination type, false=single ship</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:shiptypeNameType"/>

<xs:element name="NL" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="BG" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="CS" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="DA" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="EL" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="ES" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="ET" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="FI" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="HU" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="IT" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="LT" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="LV" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="PL" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="PT" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="RO" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="SK" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="SL" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="SV" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="HR" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="RU" type="tns:shiptypeNameType" minOccurs="0"/>

<xs:element name="SR" type="tns:shiptypeNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="eri\_countryReqMutType">

<xs:annotation>

<xs:documentation>country minimal set. Must MATCH with eri\_countryType. Be sure to edit the basic types, dont edit the subelements here!!!</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="countryCode" type="tns:countryCodeType">

<xs:annotation>

<xs:documentation>UN country ISO code 2 chars. Be sure to edit, change the basic Simpletypes, dont edit any of the (sub)elements from here!!!</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:countryCodeType" minOccurs="0"/>

<xs:element name="cbsCode" type="tns:cbscountryCodeType" minOccurs="0"/>

<xs:element name="countryIsoCode3" type="tns:countryCode3Type" minOccurs="0"/>

<xs:element name="countryIsoNum" type="tns:countryIsoNumType" minOccurs="0"/>

<xs:element name="lloydsFlag" type="tns:lloydsflagType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:countryNameType"/>

<xs:element name="NL" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="BG" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="CS" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="DA" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="EL" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="ES" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="ET" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="FI" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="HU" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="IT" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="LT" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="LV" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="PL" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="PT" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="RO" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="SK" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="SL" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="SV" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="HR" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="RU" type="tns:countryNameType" minOccurs="0"/>

<xs:element name="SR" type="tns:countryNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="er\_erireportingpointReqMutType">

<xs:annotation>

<xs:documentation>ERI reporting points minimal set. Structure MUST Match with "er\_erireportingpointType". Be sure to edit ONLY the basic types.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="repPointId" type="tns:repPointIdType">

<xs:annotation>

<xs:documentation>Unique ERI reporting point id (first 3 are significant to support aliases)</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:repPointIdType" minOccurs="0"/>

<xs:element name="risCode" type="tns:risCodeType" minOccurs="0">

<xs:annotation>

<xs:documentation>Related (nearest or corresponding) RIS-idx (ISRS) code for the Reportingpoint</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="name">

<xs:annotation>

<xs:documentation>Supports only the basic ERI languages</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:repPointNameType"/>

<xs:element name="NL" type="tns:repPointNameType" minOccurs="0"/>

<xs:element name="DE" type="tns:repPointNameType" minOccurs="0"/>

<xs:element name="FR" type="tns:repPointNameType" minOccurs="0"/>

<xs:element name="EN" type="tns:repPointNameType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="eriProviderId" type="tns:providerIdType" minOccurs="0">

<xs:annotation>

<xs:documentation>Provder ID of the system(VTS/IVS) acting as the receiving system for this Reporting point.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="er\_locsreppointsReqMutType">

<xs:annotation>

<xs:documentation>ERI locations and possible/available Reportingpoints minimal set. Structure MUST match "er\_locreppointsType". Be sure to edit the basic types ONLY</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="erilocCode" type="tns:erilandlocCodeType">

<xs:annotation>

<xs:documentation>Unique ERI loc code</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="erilocCodeOld" type="tns:erilandlocCodeType" minOccurs="0">

<xs:annotation>

<xs:documentation>De oude (te wijzigen) code is altijd een combinatie van erilocCode en repPoint</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="repPointId" type="tns:repPointIdType">

<xs:annotation>

<xs:documentation>Unique ERI reporting point id</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="repPointIdOld" type="tns:repPointIdType" minOccurs="0"/>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="er\_eriproviderReqMutType">

<xs:annotation>

<xs:documentation>ERI providers (systems and their address info) minimal set. Structure MUST match "er\_providerType". Be sure to edit ONLY the basic types.</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="providerId" type="tns:providerIdType">

<xs:annotation>

<xs:documentation>Provder ID of the system(VTS/IVS) acting as the receiving system for ERI msgs.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="codeOld" type="tns:providerIdType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:providerNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="systemId" type="tns:systemIdType" minOccurs="0"/>

<xs:element name="systemEmail" type="tns:emailAddressType" minOccurs="0"/>

<xs:element name="contactPers" type="tns:contactNameType" minOccurs="0"/>

<xs:element name="contactTel" type="tns:telnrType" minOccurs="0"/>

<xs:element name="contactFax" type="tns:telnrType" minOccurs="0"/>

<xs:element name="contactAddress" type="tns:contactAddressType" minOccurs="0"/>

<xs:element name="contactPostalCode" type="tns:contactPostalCodeType" minOccurs="0"/>

<xs:element name="contactCity" type="tns:contactCitynameType" minOccurs="0"/>

<xs:element name="contactCountrycode" type="tns:countryCodeType" minOccurs="0">

<xs:annotation>

<xs:documentation>ISO3166 country code</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="receiveParty" type="tns:providerOptionsType" minOccurs="0"/>

<xs:element name="invoiceParty" type="tns:providerOptionsType" minOccurs="0"/>

<xs:element name="responseParty" type="tns:providerOptionsType" minOccurs="0"/>

<xs:element name="erinotMsgVersion" type="tns:msgVersionType" minOccurs="0">

<xs:annotation>

<xs:documentation>msgversion 0=not applicable</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="bermanMsgVersion" type="tns:msgVersionType" minOccurs="0"/>

<xs:element name="erimanMsgVersion" type="tns:msgVersionType" minOccurs="0"/>

<xs:element name="paxlstMsgVersion" type="tns:msgVersionType" minOccurs="0"/>

<xs:element name="erivoyMsgVersion" type="tns:msgVersionType" minOccurs="0"/>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType" minOccurs="0"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="nts\_dataType">

<xs:annotation>

<xs:documentation>NtS datatype (all the nts tables are represented by 1 data structure)</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="ntstype" type="tns:ntstypeType">

<xs:annotation>

<xs:documentation>Just a string (no enum) to return nts data type of the nts\_data returned! reftype already contains datatype.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="ntscode" type="tns:ntsCodeType"/>

<xs:element name="codeOld" type="tns:ntsCodeType" minOccurs="0"/>

<xs:element name="name">

<xs:complexType>

<xs:sequence>

<xs:element name="Loc" type="tns:ntsNameType"/>

<xs:element name="NL" type="tns:ntsNameType"/>

<xs:element name="DE" type="tns:ntsNameType"/>

<xs:element name="EN" type="tns:ntsNameType"/>

<xs:element name="FR" type="tns:ntsNameType"/>

<xs:element name="BG" type="tns:ntsNameType"/>

<xs:element name="CS" type="tns:ntsNameType"/>

<xs:element name="DA" type="tns:ntsNameType"/>

<xs:element name="EL" type="tns:ntsNameType"/>

<xs:element name="ES" type="tns:ntsNameType"/>

<xs:element name="ET" type="tns:ntsNameType"/>

<xs:element name="FI" type="tns:ntsNameType"/>

<xs:element name="HU" type="tns:ntsNameType"/>

<xs:element name="IT" type="tns:ntsNameType"/>

<xs:element name="LT" type="tns:ntsNameType"/>

<xs:element name="LV" type="tns:ntsNameType"/>

<xs:element name="PL" type="tns:ntsNameType"/>

<xs:element name="PT" type="tns:ntsNameType"/>

<xs:element name="RO" type="tns:ntsNameType"/>

<xs:element name="SK" type="tns:ntsNameType"/>

<xs:element name="SL" type="tns:ntsNameType"/>

<xs:element name="SV" type="tns:ntsNameType"/>

<xs:element name="HR" type="tns:ntsNameType"/>

<xs:element name="RU" type="tns:ntsNameType"/>

<xs:element name="SR" type="tns:ntsNameType"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="source" type="tns:sourceType" minOccurs="0"/>

<xs:element name="remarks" type="tns:remarksType" minOccurs="0"/>

<xs:element name="version" type="tns:refrecVersionType"/>

<xs:element name="erased" type="tns:erasedType"/>

<xs:element name="lastupdate" type="tns:lastupdateType"/>

</xs:sequence>

</xs:complexType>

<xs:simpleType name="adnCodeType">

<xs:annotation>

<xs:documentation>adn (dgs goods) internal adn code (unnr + seqnr: to make each adn good unique)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="7"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="adnClassType">

<xs:annotation>

<xs:documentation>adn class and classifications codes</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="adnLabelType">

<xs:annotation>

<xs:documentation>adn (dgs) labels applicable for a dgs good</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="aquaticToxicityType">

<xs:annotation>

<xs:documentation>adn aquatic toxicity, tanker transports (indication for dgs, usuallly liquids)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="15"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="cbscountryCodeType">

<xs:annotation>

<xs:documentation>Older CBS (number) code (3 digits) use by statistics organisations (backw comp) but replaced by newer iso 3166 (2 letters).</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="3"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="chemicalStabiltyType">

<xs:annotation>

<xs:documentation>adn chemical stability (indication), tanker transports</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="10"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="cmrPropertyType">

<xs:annotation>

<xs:documentation>adn CMR properties (tanker transports)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="10"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="contIdType">

<xs:annotation>

<xs:documentation>iso 6346 container type (code)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="4"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="contNameType">

<xs:annotation>

<xs:documentation>containertype textual description</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="combinationFlagType">

<xs:annotation>

<xs:documentation>indication (flag) wheater shiptype code is a single ship or a transport combination</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:boolean"/>

</xs:simpleType>

<xs:simpleType name="contactNameType">

<xs:annotation>

<xs:documentation>contact (person) name type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="35"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="contactAddressType">

<xs:annotation>

<xs:documentation>contact address text type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="35"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="contactCitynameType">

<xs:annotation>

<xs:documentation>contact city name Type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="35"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="contactPostalCodeType">

<xs:annotation>

<xs:documentation>contact International postal code (text) type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="10"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="countryCodeType">

<xs:annotation>

<xs:documentation>Country iso 3166 code (2 alpha)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="2"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="countryCode3Type">

<xs:annotation>

<xs:documentation>Country iso 3166 code (3 alpha)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="3"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="countryNameType">

<xs:annotation>

<xs:documentation>Countryname type (name of the country)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="countryIsoNumType">

<xs:annotation>

<xs:documentation>Country iso 3166 numcode (3 digits)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="3"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="craftTypeType">

<xs:annotation>

<xs:documentation>techn directive craft type code (eu hull db)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="2"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="conesType">

<xs:annotation>

<xs:documentation>inland cones 0,1,2,3,V (as used for DGS goods and cones calculations)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="1"/>

<xs:enumeration value="0"/>

<xs:enumeration value="1"/>

<xs:enumeration value="2"/>

<xs:enumeration value="3"/>

<xs:enumeration value="V"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="dimType">

<xs:annotation>

<xs:documentation>dimension type (length, width, draught etc) in [cm] (no decimals)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long"/>

</xs:simpleType>

<xs:simpleType name="dvkTypeType">

<xs:annotation>

<xs:documentation>older 2 digit dvk shiptype code (replaced by unrec</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="2"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="erasedType">

<xs:annotation>

<xs:documentation>record erased (not actual anymore) flag type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:boolean"/>

</xs:simpleType>

<xs:simpleType name="erilandlocCodeType">

<xs:annotation>

<xs:documentation>combined land or location code (to specify a country or any location within a country)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:minLength value="2"/>

<xs:maxLength value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="erilocCodeType">

<xs:annotation>

<xs:documentation>eri location code type as used for eri\_locations</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:minLength value="5"/>

<xs:maxLength value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="erilocNameType">

<xs:annotation>

<xs:documentation>eri location name type as used in eri\_locations</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="90"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="emailAddressType">

<xs:annotation>

<xs:documentation>email addres Type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="50"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="flagsType">

<xs:annotation>

<xs:documentation>flags 0,B,V (dang goods indication seagoing vessels) </xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="1"/>

<xs:enumeration value="0"/>

<xs:enumeration value="B"/>

<xs:enumeration value="V"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="floatersinkerType">

<xs:annotation>

<xs:documentation>adn floater or sinker (indication), tanker transports</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="fwCodeType">

<xs:annotation>

<xs:documentation>fairway code (text)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="goodDgsNameType">

<xs:annotation>

<xs:documentation> ADN dangerous goodnames, description</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="255"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="goodNameType">

<xs:annotation>

<xs:documentation>non-dangerous goodnames, description</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="goodSynType">

<xs:annotation>

<xs:documentation>ADN dangerous good synonym name,description</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="512"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="hsCodeType">

<xs:annotation>

<xs:documentation>HS code as used by customs</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="10"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="kmCodeType">

<xs:annotation>

<xs:documentation>hectometre code (hectomere in [hm] (no decimals) on the fairway)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="lastupdateType">

<xs:annotation>

<xs:documentation>lastupdate, last modification date time of this record</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:dateTime"/>

</xs:simpleType>

<xs:simpleType name="langType">

<xs:annotation>

<xs:documentation>one of the supported languages (basic and additional set), based on ISO 639-1</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="2"/>

<xs:enumeration value="%%"/>

<xs:enumeration value="lc"/>

<xs:enumeration value="nl"/>

<xs:enumeration value="de"/>

<xs:enumeration value="fr"/>

<xs:enumeration value="en"/>

<xs:enumeration value="bg"/>

<xs:enumeration value="cs"/>

<xs:enumeration value="da"/>

<xs:enumeration value="el"/>

<xs:enumeration value="es"/>

<xs:enumeration value="et"/>

<xs:enumeration value="fi"/>

<xs:enumeration value="hu"/>

<xs:enumeration value="it"/>

<xs:enumeration value="lt"/>

<xs:enumeration value="lv"/>

<xs:enumeration value="pl"/>

<xs:enumeration value="pt"/>

<xs:enumeration value="ro"/>

<xs:enumeration value="sk"/>

<xs:enumeration value="sl"/>

<xs:enumeration value="sv"/>

<xs:enumeration value="hr"/>

<xs:enumeration value="ru"/>

<xs:enumeration value="sr"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="latlonType">

<xs:annotation>

<xs:documentation>lat long (WGS decimal [.] coordinates) coordinate type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:double"/>

</xs:simpleType>

<xs:simpleType name="loCodeType">

<xs:annotation>

<xs:documentation>un locode (2 country + 3 location code)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="locExitsType">

<xs:annotation>

<xs:documentation>Number of possible (departure) exist (routes) for a location</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:integer"/>

</xs:simpleType>

<xs:simpleType name="locTypeType">

<xs:annotation>

<xs:documentation>location type, terminal type (a number indicating type of location,terminal)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:integer"/>

</xs:simpleType>

<xs:simpleType name="lloydsflagType">

<xs:annotation>

<xs:documentation>lloyds flag code for a certain country (seagoing)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="3"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="msgVersionType">

<xs:annotation>

<xs:documentation>msgVersion (of the msg to send out) type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="10"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="nstCodeType">

<xs:annotation>

<xs:documentation>NST code (corresponing nst2007 EU statistics codes for goods).</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="6"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="ntsCodeType">

<xs:annotation>

<xs:documentation>NtS (Notices to Skippers) code/Value datatype</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="50"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="ntsNameType">

<xs:annotation>

<xs:documentation>NtS name/description (belonging to a ntsCode) datatype</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="255"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="ntstypeType">

<xs:annotation>

<xs:documentation>NtS subtype (sub data type, there are several nts tables)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string"/>

</xs:simpleType>

<xs:simpleType name="packGrpType">

<xs:annotation>

<xs:documentation>ADN dangerous good packing group I, II, III</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="packIdType">

<xs:annotation>

<xs:documentation>UN rec 21 inner package code</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:minLength value="2"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="packNameType">

<xs:annotation>

<xs:documentation>Un rec 21 inner package name/description</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="100"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="passwordType">

<xs:annotation>

<xs:documentation>Password type (belonging to a user, used to identify a user and its role and home country).</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string"/>

</xs:simpleType>

<xs:simpleType name="providerIdType">

<xs:annotation>

<xs:documentation>VTS/IVS code/id of receiving system active in certain area and acting as a (ERI) receiving point (several reporting point could point to one VTS receving system)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="providerNameType">

<xs:annotation>

<xs:documentation>VTS/IVS name/receiving system name of the local authority (ie MIB Germany, IVS90 (NL))</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="providerOptionsType">

<xs:annotation>

<xs:documentation>options Type of options the provider (receiving system) supports, such as response msg (true/false)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="1"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="quayType">

<xs:annotation>

<xs:documentation>quay type (terminals), indicating quay nrs/id's</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="recCountType">

<xs:annotation>

<xs:documentation>record counter (nr of records available or in a set)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long"/>

</xs:simpleType>

<xs:simpleType name="reftypeType">

<xs:annotation>

<xs:documentation>reftype Type (reftool data reftype, such as: risidx, eri\_location, eri\_hscode, nts\_data)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:enumeration value="ris\_idx"/>

<xs:enumeration value="eri\_location"/>

<xs:enumeration value="eri\_hscode"/>

<xs:enumeration value="eri\_adncode"/>

<xs:enumeration value="eri\_conttype"/>

<xs:enumeration value="eri\_packtype"/>

<xs:enumeration value="eri\_shiptype"/>

<xs:enumeration value="eri\_country"/>

<xs:enumeration value="er\_erireportingpoint"/>

<xs:enumeration value="er\_locsreppoints"/>

<xs:enumeration value="er\_eriprovider"/>

<xs:enumeration value="nts\_barrage"/>

<xs:enumeration value="nts\_communication"/>

<xs:enumeration value="nts\_country"/>

<xs:enumeration value="nts\_direction"/>

<xs:enumeration value="nts\_direction\_max"/>

<xs:enumeration value="nts\_direction\_min"/>

<xs:enumeration value="nts\_ice\_accessibility"/>

<xs:enumeration value="nts\_ice\_condition"/>

<xs:enumeration value="nts\_ice\_classification"/>

<xs:enumeration value="nts\_ice\_situation"/>

<xs:enumeration value="nts\_indication"/>

<xs:enumeration value="nts\_interval"/>

<xs:enumeration value="nts\_language"/>

<xs:enumeration value="nts\_limitation"/>

<xs:enumeration value="nts\_measure"/>

<xs:enumeration value="nts\_position"/>

<xs:enumeration value="nts\_reason"/>

<xs:enumeration value="nts\_reference"/>

<xs:enumeration value="nts\_regime"/>

<xs:enumeration value="nts\_reporting"/>

<xs:enumeration value="nts\_subject"/>

<xs:enumeration value="nts\_tag"/>

<xs:enumeration value="nts\_target\_group"/>

<xs:enumeration value="nts\_type"/>

<xs:enumeration value="nts\_weather\_category"/>

<xs:enumeration value="nts\_weather\_class"/>

<xs:enumeration value="nts\_weather\_item"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="refrecVersionType">

<xs:annotation>

<xs:documentation>RefData record version numbering (automaically, starts with 1 and increments on updates, requesting version 0=dont care)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long"/>

</xs:simpleType>

<xs:simpleType name="remarksType">

<xs:annotation>

<xs:documentation>remarks string (remarks about the maintance of a record or any other remarks)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="512"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="repPointIdType">

<xs:annotation>

<xs:documentation>reporting point id/code of the local authority/traffic centre where to report (ie D01, D01a; first 3 are significant to support aliases)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="4"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="repPointNameType">

<xs:annotation>

<xs:documentation>reporting point (local authority/traffic centre) name (ie MIB Duisburg)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risCodeType">

<xs:annotation>

<xs:documentation>ris index (ISRS) code type (full ris index code)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="rislocNameType">

<xs:annotation>

<xs:documentation>locationames as used in the ris index</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risrelencType">

<xs:annotation>

<xs:documentation>ris idx related ENC's type as used in ris idx (list of ENC id's)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="50"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risobjCodeType">

<xs:annotation>

<xs:documentation>ris idx object code type (single code for a junction, termcode etc padded), subpart of the ris idx.</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="riscomminfoType">

<xs:annotation>

<xs:documentation>ris idx communication information TXTDSC</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risobjfuncType">

<xs:annotation>

<xs:documentation>ris idx object function code type (junction, berths etc)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="10"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="rissectnodeType">

<xs:annotation>

<xs:documentation>ris idx section node id</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risgaugeidType">

<xs:annotation>

<xs:documentation>ris idx national Gauge Code type (can be utilised to store any national coding/id/link for gauges)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risapplikmType">

<xs:annotation>

<xs:documentation>ris idx applicability from / to km (in hectometres (no decimals))</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long"/>

</xs:simpleType>

<xs:simpleType name="risreflevelcodeType">

<xs:annotation>

<xs:documentation>ris idx reference level code (indicating the ref level)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risreflevelvalueType">

<xs:annotation>

<xs:documentation>ris idx reference level value [cm]</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long"/>

</xs:simpleType>

<xs:simpleType name="risgeodrefType">

<xs:annotation>

<xs:documentation>ris idx geod reference</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="riszeropointType">

<xs:annotation>

<xs:documentation>ris idx zero point [cm]</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long"/>

</xs:simpleType>

<xs:simpleType name="riscatoftimeschedType">

<xs:annotation>

<xs:documentation>ris idx category of time schedule (cattab)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="35"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risforshiptypeType">

<xs:annotation>

<xs:documentation>ris idx for ship type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="risuseofshipType">

<xs:annotation>

<xs:documentation>ris idx use of ship useshp</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="rislinkextxmlType">

<xs:annotation>

<xs:documentation>ris idx link to external XML file</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="256"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="shipTypeType">

<xs:annotation>

<xs:documentation>UNrec 28 shiptype (4 digits)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="4"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="shiptypeNameType">

<xs:annotation>

<xs:documentation>shiptype name (textual) description</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="sourceType">

<xs:annotation>

<xs:documentation>source string, to specify the source of this record/data or the update.</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="70"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="signalCodeType">

<xs:annotation>

<xs:documentation>ADN signal code (to group dgs for cones/flags calculations)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="1"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="systemIdType">

<xs:annotation>

<xs:documentation>Unique System/Provider/ (mailbox)account id</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long">

<xs:totalDigits value="9"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="termCodeType">

<xs:annotation>

<xs:documentation>terminal code type (remark: termcode is the only part of eri locs with variable length, historical reasons).</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="5"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="telnrType">

<xs:annotation>

<xs:documentation>telephone/fax number Type</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="20"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="unlocCCType">

<xs:annotation>

<xs:documentation>unlocode countrycode part (first 2 of unlocode) as used in ris idx</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="2"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="unlocLCType">

<xs:annotation>

<xs:documentation>unlocode location code part (last 3 of unlocode) as used in ris idx</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="3"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="unnrType">

<xs:annotation>

<xs:documentation>UN nr (4 digits) adn nr for identifying ADN dangerous goods (according CCNR)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:length value="4"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="userNameType">

<xs:annotation>

<xs:documentation>reftool username(reftool accountname) type, to idetify a user and its role.</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string"/>

</xs:simpleType>

<xs:simpleType name="weightType">

<xs:annotation>

<xs:documentation>cargo weight type, integer nr (units [tons], [kg]) depending on element)</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:long"/>

</xs:simpleType>

<xs:simpleType name="warnIdType">

<xs:annotation>

<xs:documentation>adn warning (dgs) card id (since adn2013 this is used as a reference to ADN Tabel A and or Table C</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="7"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="warnCardType">

<xs:annotation>

<xs:documentation>adn warning card </xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:maxLength value="7"/>

</xs:restriction>

</xs:simpleType>

</xs:schema>

* 1. NLRDMS Webservice – Veld formaten

In deze bijlage is een opsomming van de belangrijkste velden en formaten terug te vinden, zoals gebruikt en gedefinieerd in de webservice interface. Deze definities (data typen) zijn terug te vinden in de **XSD** van de webservice (zie bijlage A2).

De kolommen in de onderstaande overzichten, hebben de volgende betekenis:

1. Kolom *“Veld”*: naam van het gegevensveld.
2. Kolom *“Type”*: het formaat van het gegeven (An..25=Alfanumeriek maximaal 25 lang, AN5=Alfanumeriek exact 5 lang, N9=Numeriek 9 lang).
3. Kolom *“Omschrijving”*: beschrijft de betekenis van het veld.
   * 1. RIS-index velden

De RIS-index kent de volgende informatie velden (zie ook de RIS Encoding Guide):

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| risidxCode | AN20 | Unieke 20 letterige RIS-index, opgebouwd uit 5 letterige UN-locode, 5 letterige vaarwegcode, 5 letterige objectcode en 5 cijferige kmcode. |
| codeOld | AN20 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen risidxCode). |
| unlocCC | AN2 | UN landcode |
| unlocLC | AN3 | UN locatiecode (samen met unlocLC de UN-locode). |
| wwsectCode | AN5 | Vaarwegcode |
| objcode | AN5 | Objectcode |
| kmcode | AN5 | Kilometrering in hectometers |
| objname\Loc | AN..70 | Objectnaam |
| objname\locNational | AN..70 | Objectnaam in lokale / nationale spelling |
| locname\Loc | AN..70 | Locatienaam |
| locname\locNational | AN..70 | Locatienaam in lokale / nationale spelling |
| wwname\Loc | AN..70 | Vaarwegnaam |
| wwname\locNational | AN..70 | Vaarwegnaam in lokale / nationale spelling |
| routename\Loc | AN..70 | Routenaam |
| routename\locNational | AN..70 | Routenaam in nationale spelling (buiten gebruik) |
| positionCode | AN..5 | Positie code (links, rechts etc.) |
| relrisindex | AN20 | Related RIS-index waaraan deze is “opgehangen”. |
| sectionnode | AN..20 | Vaarweg sectie |
| Lat | N | Lat coordinaat in decimale notatie |
| Lon | N | Long coordinaat in decimale notatie |
| relenc | AN..50 | Related ENC’s |
| comminfo | AN..70 | Communicatie info |
| gaugeid | AN..20 | Peilschaal id (nationaal) |
| vesselconvlength | N | Vaartuig/konvooi lengte in cm. |
| vesselconvbreadth | N | Vaartuig/konvooi breedte in cm. |
| vesselconvdraught | N | Vaartuig/konvooi diepgang in cm. |
| vesselconvairdraught | N | Vaartuig/konvooi doorvaart hoogte in cm. |
| availablelength | N | Beschikbare (passeer) lengte in cm. |
| clearancewidth | N | Beschikbare (passeer) breedte in cm. |
| availabledepthcode | AN..20 | Beschikbare diepte, referentievlak code |
| availabledepth | N | Beschikbare (passeer) diepte in cm. |
| clearanceheightcode | AN..20 | Beschikbare doorvaart hoogte, referentievlak code |
| clearanceheight | N | Beschikbare doorvaart hoogte in cm. |
| referencegaugeISRS | AN20 | Van toepassing zijnde peilschaal (RIS-index) |
| applicabilityfromkm | N | Van toepassing vanaf kilometrering (in rhm) |
| applicabilitytokm | N | Van toepassing tot kilometrering (in rhm) |
| reflevel1code | AN..20 | Referentievlak code 1 |
| reflevel1value | N | Referentievlak waarde in cm |
| reflevel2code | AN..20 | Referentievlak code 2 |
| reflevel2value | N | Referentievlak waarde in cm |
| reflevel3code | AN..20 | Referentievlak code 3 |
| reflevel3value | N | Referentievlak waarde in cm |
| zeropoint | N | Hoogte van nulpunt t.o.v. refvlak in cm. |
| geodref | AN..70 | Geodetische referentie voor het nulpunt. |
| catoftimesched | AN..35 | Categorie vh tijdschema (bediening), conform ECDIS |
| forshiptype | AN..70 | Van toepassing voor schiptype, conform ECDIS |
| foruseofship | AN..70 | Van toepassing voor type gebruik, conform ECDIS |
| lnktoextXMfiletimesched | AN..256 | Extern tijdschema link. |
| lnktoextXMfilepastime | AN..256 | Passeertijden schema. |
| Vesselconvlength2 | N | Vaartuig/konvooi lengte in cm. |
| Vesselconvbreadth2 | N | Vaartuig/konvooi breedte in cm. |
| Vesselconvdraught2 | N | Vaartuig/konvooi diepgang in cm. |
| Vesselconvairdraught2 | N | Vaartuig/konvooi doorvaart hoogte in cm. |
| Availablelength2 | N | Beschikbare (passeer) lengte in cm. |
| Clearancewidth2 | N | Beschikbare (passeer) breedte in cm. |
| Availabledepthcode2 | AN..20 | Beschikbare diepte, referentievlak code |
| Availabledepth2 | N | Beschikbare (passeer) diepte in cm. |
| Clearanceheightcode2 | AN..20 | Beschikbare doorvaart hoogte, referentievlak code |
| Clearanceheight2 | N | Beschikbare doorvaart hoogte in cm. |
| Referencegauge2ISRS | AN20 | Van toepassing zijnde peilschaal (RIS-index) |
| Vesselconvlength3 | N | Vaartuig/konvooi lengte in cm. |
| Vesselconvbreadth3 | N | Vaartuig/konvooi breedte in cm. |
| Vesselconvdraught3 | N | Vaartuig/konvooi diepgang in cm. |
| Vesselconvairdraught3 | N | Vaartuig/konvooi doorvaart hoogte in cm. |
| Availablelength3 | N | Beschikbare (passeer) lengte in cm. |
| Clearancewidth3 | N | Beschikbare (passeer) breedte in cm. |
| Availabledepthcode3 | AN..20 | Beschikbare diepte, referentievlak code |
| Availabledepth3 | N | Beschikbare (passeer) diepte in cm. |
| Clearanceheightcode3 | AN..20 | Beschikbare doorvaart hoogte, referentievlak code |
| Clearanceheight3 | N | Beschikbare doorvaart hoogte in cm. |
| Referencegauge3ISRS | AN20 | Van toepassing zijnde peilschaal (RIS-index) |
| Catoftimesched2 | AN..35 | Categorie vh tijdschema (bediening), conform ECDIS |
| Forshiptype2 | AN..70 | Van toepassing voor schiptype, conform ECDIS |
| Foruseofship2 | AN..70 | Van toepassing voor type gebruik, conform ECDIS |
| lnktoextXMfiletimesched2 | AN..256 | Extern tijdschema link. |
| lnktoextXMfilepastime2 | AN..256 | Passeertijden schema. |
| Catoftimesched3 | AN..35 | Categorie vh tijdschema (bediening), conform ECDIS |
| Forshiptype3 | AN..70 | Van toepassing voor schiptype, conform ECDIS |
| Foruseofship3 | AN..70 | Van toepassing voor type gebruik, conform ECDIS |
| lnktoextXMfiletimesched3 | AN..256 | Extern tijdschema link. |
| lnktoextXMfilepastime3 | AN..256 | Passeertijden schema. |
| startdate | DT | Start datumtijd van de geldigheid van deze index |
| enddate | DT | Eind datumtijd v/d geldigheid |
| infodate | DT | Datumtijd v/d informatie (datumtijd v publicatie). |
| risidxRemarks | AN..512 | Opmerkingen mbt deze RIS index |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* + 1. ERI-locatie velden

ERI-locatie kent de volgende informatie velden:

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| erilocCode | AN20 | Unieke 20 letterige ERI-locatie code, opgebouwd uit 5 letterige UN-locode, 5 letterige vaarwegcode, 5 letterige terminalcode en 5 cijferige kmcode. |
| codeOld | AN20 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen erilocCode). |
| riscode | AN20 | Gerelateerde RIS-index (ophangpunt van deze loc) |
| unlocode | AN5 | UN locatiecode |
| fwcode | AN5 | Vaarwegcode |
| termcode | AN5 | Terminal code |
| kmcode | AN5 | Kilometrering in hectometers |
| Name (velden) | AN..90 | Locatie naam (in 5 notaties/spellingen) |
| loctype | N | Type locatie, o.a.: 0=no info, 1=on fairway, 2=routepoints,4=sea port, 8=hinterland, 64=basic code etc. |
| Termname (velden) | AN..90 | Terminal naam (in 5 notaties/spellingen) |
| quayfrom | AN..5 | Kadenr van |
| quayto | AN..5 | Kadenr tot |
| Termtype | N | Type: 1=cont, 2=tank, 4=drybulk, 10,11,12=routepoints etc. |
| Lat | N | Lat coördinaat van ligplaats in decimale notatie |
| Lon | N | Long coördinaat in decimale notatie |
| ivsVTSCode | AN..20 | VTS code (code v/h ontvangende systeem) |
| Exits | N | Aantal mogelijke vertrek richtingen , 0 of >1 dan zijn er meer mogelijkheden en is extra informatie (1e meldpunt) voor deze vertrekplaats nodig. |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* + 1. ERI HS-code velden

Een HS-code (niet-gevaarlijke stoffen) kent de volgende informatie velden:

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| hsCode | AN10 | Unieke 10 cijferige HS-code (1e 6 zijn significant) |
| codeOld | AN10 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen hsCode). |
| Name (velden) | AN..70 | Naam van de stof (in 25 notaties/spellingen), alleen de velden: Loc, NL, DE, FR, EN worden gebruikt. |
| nstCode | AN..6 | Bijbehorende NST statistiek categorie (NST2007) |
| nstName | AN..70 | Bijbehorende Engelse NST omschrijving |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* + 1. ERI ADN-code velden

Een ADN-code (gevaarlijke stoffen) kent de volgende informatie velden:

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| adnCode | AN7 | Unieke 7 cijferige code (1e 4 = VN nr + seq nr) |
| codeOld | AN7 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen adnCode). |
| unnr | AN4 | UN nummer |
| adnClass | AN..5 | ADN Klasse |
| adnClassif | AN..5 | ADN Classificatie |
| imoClass | AN..5 | IMO Klasse (sinds ADN2003 gelijk aan ADN klasse) |
| packGrp | AN..5 | Verpakkingsgroep (I, II, III) |
| Name (velden) | AN..255 | Naam van de stof (in 25 notaties/spellingen), alleen de velden: Loc, NL, DE, FR, EN worden gebruikt. |
| Syn (velden) | AN..255 | Aanvullende info voor de stof (in 25 notaties/spellingen), alleen de velden: Loc, NL, DE, FR, EN worden gebruikt. |
| hsCode | AN..6 | Bijbehorende HS-code categorie (WCO HS-code) |
| nstCode | AN..6 | Bijbehorende NST statistiek categorie (NST2007) |
| conesbulk | AN1 | Kegels bulk transport |
| conesTank | AN1 | Kegels Tank transport |
| conesCont | AN1 | Kegels Container transport |
| flagsBulk | AN1 | Vlag zeevaart bulk transport |
| FlagsTank | AN1 | Vlag zeevaart tank transport |
| warnCard | AN..7 | Gevarenkaart (vervallen sinds ADN2003) |
| warnId | AN..7 | Referentie naar de ADN tabel (tabel A, C of AC) |
| Label1 | AN..5 | ADN Etiket 1 |
| Label2 | AN..5 | ADN Etiket 2 |
| Label3 | AN..5 | ADN Etiket 3 |
| Label4 | AN..5 | ADN Etiket 4 |
| aquaticToxicity | AN..15 | Milieu info, giftigheid |
| chemicalStability | AN..10 | Milieu info, chemische stabiliteit |
| cmrProperties | AN..10 | Milieu info, CMR eigenschappen |
| floater | AN..5 | Milieu info, drijvende of zinkende substantie. |
| maxWeightCodeInlandBulkCont | AN1 | Max gewicht code |
| maxWeightInlandBulkCont | N | Maximum gewicht in kg |
| signalCodeInlandCont | AN1 | Gewichtscode container vervoer |
| minWeightInlandCont | N | Minimum gewicht in kg |
| signalCodeInlandExcemp | AN1 | Vrijstellingscode |
| minWeightInlandExcemp | N | Vrijstellingsgewicht in kg |
| signalCodeSeaBulk | AN1 | Gewichtscode Zeevaart Bulk |
| minWeightSeaBulk | N | Minimum gewicht zeevaart in kg |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* + 1. ERI Containertype velden

Een containertype kent de volgende informatie velden:

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| contId | AN4 | Unieke 4 cijferige type codering (conform ISO) |
| codeOld | AN4 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen contId code). |
| length | N | Lengte van de container in cm |
| width | N | Breedte van de container in cm |
| height | N | hoogte van de container in cm |
| weight | N | Leeg gewicht van de container in kg |
| Name (velden) | AN..70 | Naam van containertype (in 25 notaties/spellingen), alleen de velden: Loc, NL, DE, FR, EN worden gebruikt. |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* + 1. ERI Packagetype velden

Een packagetype (binnenverpakking) kent de volgende informatie velden:

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| packId | AN2 | Unieke 2 cijferige type codering. |
| codeOld | AN2 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen packId code). |
| Name (velden) | AN..100 | Naam van verpakkingstype (in 25 notaties/spellingen), alleen de velden: Loc, NL, DE, FR, EN worden gebruikt. |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* + 1. ERI Shiptype velden

Een scheepstype kent de volgende informatie velden:

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| ShipType | AN4 | Unieke 4 cijferige type codering. |
| codeOld | AN4 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen shipType code). |
| dvkType | AN2 | Uitgefaseerde (oude) DVK scheeptype code |
| craftType | AN2 | Bijbehorende craft type zoals bekend in de EUHULL |
| Combination | B | True dan betrefteen samenstel (transp combo) typering |
| Name (velden) | AN..70 | Naam van verpakkingstype (in 25 notaties/spellingen), alleen de velden: Loc, NL, DE, FR, EN worden gebruikt. |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* + 1. ERI Country velden

Een landcode kent de volgende informatie velden:

| **Velden** | **Type** | **Omschrijving** |
| --- | --- | --- |
| CountryCode | AN2 | Unieke 2 letterige type codering (ISO 3166) |
| codeOld | AN2 | Oude vorige code (i.g.v. code wijziging verwijst dit veld naar de vervangen CountryCode). |
| cbsCode | AN3 | Uitgefaseerde (verouderde) CBS land codering |
| countryIsocode3 | AN3 | 3 letterige ISO 3166 codering |
| countryIsoNum | AN3 | 3 cijferige ISO 3166 (numerieke) codering |
| lloydsFlag | AN3 | 3 letterige (verouderde) Lloyds codering |
| Name (velden) | AN..70 | Naam van hetland (in 25 notaties/spellingen), alleen de velden: Loc, NL, DE, FR, EN worden gebruikt. |
| source | AN..70 | Bron van het gegeven. |
| remarks | AN..512 | Opmerkingen m.b.t. het beheer van dit gegeven. |
| version | N | Versienummer van dit gegeven |
| Erased | B | Boolean True=gegeven is logisch verwijderd. |
| Lastupdate | DT | Mutatie datumtijd. |

* 1. NLRDMS Webservice - API Error codes

In deze bijlage worden de mogelijke “NL RIS Data management Service” fouten (excepties) en bijbehorende fout omschrijving opgesomd.

Het fout nummer aan het begin van bijna elke foutmelding van “NL RIS Data management Service” is geformatteerd in een bepaald vast formaat, zodat automatisch interpretatie mogelijk is.

Foutnummer formaat is “Ecc-ooo-tt”, waar:

* “E“: is the indicatie en start van het foutnummer.
* “cc”: is de subsysteem indicatie.
* “ooo”: is het bijbehorende object, waar de fout is opgetreden.
* “tt”: fout categorie.

Voorbeeld (meldingen zijn altijd in het Engels): “E21-045-07: WARNING Eri Location record .... does not exist. The mutation request is rejected”

* 21 = ERI location (subsysteem) fout.
* 045 = Related UN-locode field error.
* 07 = Does not exist error.
  + 1. Subsystem indication, error numbers

// 00 System errors

// 01 General errors

// 02 User accountdata - errors

// 03 Mutation general - errors

// 04 Import, export - errors

// 05 Mail (SMTP) - errors

// 06 Webservice general - errors

// 20 RIS Index - errors

// 21 ERI Location - errors

// 22 ERI HS code - errors

// 23 ERI ADN code - errors

// 24 ERI Container type - errors

// 25 ERI Package type - errors

// 26 ERI Country code - errors

// 27 ERI Shiptype - errors

// 28 nts Barrage - errors

// 29 nts Communication code - errors

// 30 nts Country code - errors

// 31 nts\_direction - errors

// 32 nts\_direction\_max - errors

// 33 nts\_direction\_min - errors

// 34 nts\_ice\_accessibility - errors

// 35 nts\_ice\_classification - errors

// 36 nts\_ice\_condition - errors

// 37 nts\_ice\_situation - errors

// 38 nts\_indication - errors

// 39 nts\_interval - errors

// 40 nts\_language - errors

// 41 nts\_limitation - errors

// 42 nts\_measure - errors

// 43 nts\_position - errors

// 44 nts\_reason - errors

// 45 nts\_referencelevel - errors

// 46 nts\_regime - errors

// 47 nts\_reporting - errors

// 48 nts\_subject - errors

// 49 nts\_tag - errors

// 50 nts\_target\_group - errors

// 51 nts\_type - errors

// 52 nts\_weather\_category - errors

// 53 nts\_weather\_class - errors

// 54 nts\_weather\_item - errors

* + 1. Object id, error numbers

// 000 Unkown item

// 001 Records selection (data set)

// 002 Selected, specified record

// 003 Text field

// 004 Number field

// 005 Identifier field

// 006 Reference data type field

// 007 Reference code field

// 008 Related ref code (refcode old) field

// 009 Name / description field

// 010 Remarks field

// 011 Source field

// 012 Record version field

// 013 Status field

// 014 UN Country code 3 field

// 015 UN Country nr 3 field

// 016 CBS Country field

// 017 Lloyds flagcode field

// 018 Related NST code field

// 019 Related NST name field

// 020 ADN class field

// 021 ADN classification field

// 022 IMO class field

// 023 ADN packinggroup field

// 024 ADN aggregation status field

// 025 ADN warning card field

// 026 Related HS-code field

// 027 ADN label info field

// 028 ADN Cones info field

// 029 ADN Flags info field

// 030 ADN Signal Code Inland Container field

// 031 ADN Signal Code Sea Bulk field

// 032 ADN Signal Code Inland Exemptions field

// 033 ADN max weight code Inland Bulk & Container field

// 034 ADN Synonym field

// 035 ADN min weight inland container field

// 036 ADN max weight inland Bulk & Ccontainer field

// 037 ADN min weight sea bulk field

// 038 ADN min weight inland exemptions

// 039 Length field

// 040 Width field

// 041 Height field

// 042 Weight field

// 043 Related RIS idx (ISRS) code field

// 044 Related ERI-location code field

// 045 Related UN-Locode field

// 046 Related Fairway code field

// 047 Related KM code (hecotmetre) field

// 048 Related terminal/object code field

// 049 Related Craft type code field

// 050 Related Hull type code field

// 051 Quay code field

// 052 IVS field

// 053 IVS code field

// 054 IVS name field

// 055 Terminal name field

// 056 IVS exit port(s)

// 057 Location type field

// 058 Terminal type field

// 059 Position info (lat/long) field

// 060 RIS idx object function field

// 061 RIS idx object name field

// 062 Location name field

// 063 Waterway name field

// 064 Route name field

// 065 Reasons for change field

// 066 RIS idx section node field

// 067 Related ENC's

// 068 RIS idx communication info field

// 069 RIS idx Gauge id field

// 070 RIS idx Reference level field

// 071 RIS idx Geodetic reference field

// 072 RIS idx Category of time schedule field

// 073 RIS idx For ship type field

// 074 RIS idx Link to external XML file time schedule field

// 075 RIS idx Link to external XML file passage time field

// 076 RIS idx vessel/convoy length

// 077 RIS idx vessel/convoy width

// 078 RIS idx vessel/convoy draught

// 079 RIS idx vessel/convoy airdraft

// 080 RIS idx available length

// 081 RIS idx clearance width

// 082 RIS idx available dept

// 083 RIS idx clearance height

// 084 RIS idx applicable from/to field

// 085 RIS idx zeropoint field

// 086 RIS idx Reference level value field

// 087 Start date value field

// 088 End date value field

// 089 Info date value field

// 090 Last update value field

// 091 Import line

// 092 Seperator field

// 093 User login name field

// 094 User name field

// 095 User password field

// 096 User 1st role field

// 097 User 2nd role field

// 098 User country code field

// 099 Mail message

// 100 Mail sender/recipient field

// 101 Mail subject field

// 102 Mail body field

// 103 Mail server field

// 104 Reject remarks

// 105 Mutation record

// 900 System I/O error

// 901 File/data error

// 902 Internal data

// 903 Internal data item/tag

// 904 Method, datatype

// 905 Authentication (login)

// 906 Configuration

// 907 Database

* + 1. Error type numbers

// 00 General error type (none of the other errors)

// 01 Invalid error (Item invalid (ie number format etc))

// 02 Empty error (Item not filled in)

// 03 Text too long/short (text to long/short)

// 04 Number too big/small (number to big to small)

// 05 Data format error (invalid/ not expected format (illegal code, diacritic etc))

// 06 Number format error (invalid/ not expected format (illegal number etc))

// 07 Does not exist error (not available/not exists/not found errors)

// 08 NOT allowed (operation not allowed/not enough privs)

// 09 NOT implemented (operation not implemented/unknown)

1. NLRDMS Client implementatie stappen

Appendixes. Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling.

Appendixes shall be lettered alphabetically (A, B, etc.).

In deze bijlage worden de stappen beschreven voor het implementeren van een (webservice) koppeling tussen NLRDMS en Nederlandse cliënt systemen, zoals benoemd in de [OCD\_NLREF], use-case **UC-NLREF-022**. De zogenaamde verkorte, samengevatte aansluitprocedure.

De Nederlandse cliënt systemen worden gekoppeld via de NL webservice interface (interface (#5) in Figuur 1).

Het aanvragen van een account op NLRDMS, of het opvragen van meer informatie over het aansluiten, verloopt via de NL datamanager.

De aansluitprocedure is uitgebreid in een apart memo beschreven (zie [**NLRDMS\_CLIENT**]).

Het e-mail (contact) adres van de NL datamanager is terug te vinden op de inlog pagina van NLRDMS.

Van de NLRDMS applicaties zijn er 2 versies operationeel, te weten:

1. De zogenaamde Productie versie: dat is de operationele versie met de actuele referentiegegevens, die ook in de diverse operationele ketens worden gebruikt.
2. De zogenaamde TEST (acceptatie) versie: dat is de versie en omgeving, die gebruikt moet worden tijdens de ontwikkeling van een cliënt. Deze versie wordt ook gebruikt voor het uitvoeren van de diverse testen en de acceptatietesten voordat kan worden overgeschakeld naar de productie versie. De Test versie bevat test referentiegegevens, die niet persé gelijk is aan de productie data.

De belangrijkste stappen om een koppeling te implementeren met de NLRDMS zijn:

1. Vraag de benodigde informatie en NLRDMS documentatie aan bij de NL Datamanager (via de contactgegevens op de NLRDMS login pagina), minimaal bestaande uit:
   * *ERDMS Maintenance Procedures Reference Data and RIS Index,* die de update procedures op Internationaal niveau vastleggen [ERIRDMAINT].
   * *De NLRDMS Interface Specificatie (dit document [IRSIDD-NLRDMS]),* die de beschikbare interface operaties/functies beschrijft.
2. Vraag een account aan voor de Testomgeving van NLRDMS.

Een geldig en actief gebruikersaccount is nodig om op de applicatie in te kunnen loggen (via de GUI) en handmatige mutatieverzoeken in te voeren en aan te vragen.

Hetzelfde gebruikersaccount is ook nodig voor het kunnen aanleveren van referentiegegevens via de NL webservice functies: MutateDataXML() en RequestMutationXML() functions.

Als er alleen referentiegegevens opgevraagd worden, dan is er geen (individueel) gebruikersaccount nodig en volstaat een generiek account om toegang te krijgen tot de webservice functies van NLRDMS.

1. De koppelingspartner implementeert de koppeling gebruikmakend van de NLRDMS Testomgeving.
2. De koppelingspartner voert zijn eigen testen uit gedurende de ontwikkelingsfase.
   * Test of alle web service functions in the cliënt correct werken (m.n. de GetDataXML, en Mutate functions).
   * Controleer de resultaten via de GUI (check of aangeleverde data beschikbaar is),
3. In de eindfase worden bepaalde testen in samenwerking met de NL Datamanager uitgevoerd. Pas als deze testen succesvol zijn verlopen kan de stap naar Productie plaatsvinden.
4. De NL Datamanager creëert een nieuw productie account op NLRDMS en communiceert dit samen met de aansluitgegevens naar de koppelingspartner.

+++

1. In ERDMS kunnen mutaties zitten met hetzelfde versienummer maar met een recentere mutatiedatum, ontstaan door een onvolkomenheid in de versienummering in ERDMS. [↑](#footnote-ref-2)