Technical Specification

ISO/TS 15143-4

First edition
2025-02

Earth-moving machinery and mobile road construction machinery — Worksite data exchange —

Part 4: Worksite topographical data

Engins de terrassement et machines mobiles de construction de routes — Échange de données sur le chantier —

Partie 4: Données topographiques sur le chantier

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

Reference number
ISO/TS 15143-4:2025(en)

© ISO 2025

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

COPYRIGHT PROTECTED DOCUMENT

© ISO 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 127, *Earth-moving machinery,* Subcommittee SC 03, *Machine characteristics, electrical and electronic systems, operation and maintenance.*

A list of all parts in the ISO 15143 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Introduction

This document is the fourth part in a series of documents covering data communication involving earth-moving machinery, as defined in ISO 6165, on construction worksites.

ISO 15143-1 defines the architecture of a worksite information system used to transfer information from the earth-moving machinery (EMM) to various users via the worksite information system.

ISO 15143-2 defines the individual data elements in a data dictionary.

This document is intended to be used in conjunction with ISO 15143-1 and ISO 15143-2.

ISO/TS 15143-3 defines a set of data elements common to the status and health of EMM that is collected by multiple OEMs or third-party telematics suppliers and distributed over the internet to application software used to manage machine operation.

ISO/TS 15143-3 focuses on data for management of a fleet of machines, independent of worksite, while this document focuses on data for management of a worksite and the assets specific to that worksite.

Construction worksites vary greatly in size, tasks, complexity, and scope of work. Some worksites have more design models, stakeholders, tasks and local regulations than others. In addition, most worksites have different construction contractor companies involved in various phases of the construction process. Each company can use different technologies for executing tasks. Grade control is a common technology used for earth moving to enhance operator productivity by augmenting control of a machine's working tool using a 3D design model of the worksite. The main high-level elements of a grade control solution are on-machine hardware, local job correction, field measurement equipment, design models, and offboard software tools. This document takes into account the current state of the art in machine guidance technology (grade control, compaction, etc.).

Stakeholders on the worksite must be able to interact effectively in order to complete a construction project as efficiently as possible. ISO 15143-1 defines protocols enabling different vendors of various grade control systems to efficiently interoperate on the same worksite. This document defines specifications relating to solutions both on and off the worksite.

This document focuses on the earth-moving phase of a construction worksite. Earth-moving activities are needed for many applications, such as water runoff management, stripping and reclamation, sanitary landfills, worksite development and roadbuilding.

Conformance to this document means that the relevant solutions have the specified capabilities. However, this document does not imply designing solutions that require the end-users to use these features. Since there are so many variations in job sites, tasks, and business relationships between stakeholders on a project, possibly not all of the capabilities provided on specific sites will be used. While it is possible that the end-user will not use all of the capabilities provided, conformance to this document means that the solutions include the required capabilities. Some examples include the following:

— The ability to provide information on RTK data sources to the vendor integration system (VIS) is a required capability of the site management system (SMS); however, the end-user of a particular SMS on a specific site can choose to not provide information on RTK data sources to the VIS on that site.

— A VIS can provide information about asset operators; however, this document does not require the VIS end-user to enter this information.

It is the responsibility of the user of this document to determine the applicability of legal and regulatory requirements prior to use.

Certain nouns in this document are programmatic and key elements of implementation, such as types, enumerations, and URL elements. For clarity, these programmatic nouns are italicized and use a different font from the rest of the document.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Earth-moving machinery and mobile road construction machinery — Worksite data exchange —

Part 4: Worksite topographical data

1 Scope

This document specifies requirements for data exchange at the interface between earth-moving machinery, as defined in ISO 6165, mobile road construction machinery, as defined in ISO 22242, and the worksite information systems. It focuses on data for management of a worksite and the assets specific to that worksite.

This document includes:

a) methods of local position correction and localization, including standardization of RTK corrections;

b) method of implementation of a common design model;

c) types of data and methods of data exchange between servers. This includes application programming interfaces (APIs) to exchange data between servers regardless of vendor. These APIs focus on project data, as-built data, and production data. Field-equipment-to-server and machine-to-machine are not included.

This document covers both hardware mounted on earth-moving equipment and field measurement equipment.

This document does not define methods of on-machine data collection, on-machine communication protocol (e.g. CAN bus), wireless transmission of data to the vendor's server, or wireless transmission of data directly between machines onsite. This document also does not include design software requirements or other areas related to building information management (BIM). Data formats and transfer methods from design software to the SMS are not in scope.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 3779, *Road vehicles — Vehicle identification number (VIN) — Content and structure*

ISO 8601 (all parts), *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO 10261, *Earth-moving machinery — Product identification numbering system*

ISO/IEC 12113, *Information technology — Runtime 3D asset delivery format — Khronos glTF™ 2.0*

ISO/IEC 10118-3:2018, *IT Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

ISO 80000-1, *Quantities and units — Part 1: General*

IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*

IETF RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

IETF RFC 6455, *The WebSocket Protocol*

IETF RFC 7692:2015, *Compression Extensions for WebSocket*

IETF RFC 6749:2012, *The Oauth 2.0 Authorization Framework*

IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*

IETF RFC 1738, *Uniform Resource Locators (URL)*

IETC RFC 3629:2003, *UTF-8, a transformation format of ISO 10646*

IETF RFC 3339, *Date and Time on the Internet: Timestamps*

IETC RFC 7232:2014, *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*

RTCM 10403.3, *Differential GNSS (Global Navigation Satellite Systems) Services*

RTCM 10410.1, *Standard for Networked Transport of RTCM via Internet Protocol (Ntrip)*

RTCM 13500.1, *Radio Layer for Real-Time DGNSS Applications*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1 General terms relating to construction worksites and technology

3.1.1
as-built data
data which is used to document the shape and quality of work done on a worksite

Note 1 to entry: As-built data can be other than just topographic data.

3.1.2
contractor
person or organization that undertakes construction work in accordance with a contract

[SOURCE: ISO 15143-1:2010, 3.2.11]

3.1.3
design data
data prepared in the pre-construction phase presenting shape, structures and quality of the object of the construction concerned

[SOURCE: ISO 15143-1:2010, 3.4.7]

3.1.4
design file
file containing at least one design model

3.1.5
design model
virtual representation of design data comprised of design objects intended to represent an object on the construction worksite

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

3.1.6
design object
individual element in a design model (point, line, surface etc.)

3.1.7
field equipment
assets, such as grade control systems and rovers, that are connected with a VIS

3.1.8
grade control
system of 3D positioning devices (e.g. GNSS receivers), data radios, embedded computers and software to control the working tool on an earth-moving machine with respect to a design file

3.1.9
localization
series of mathematical computations that transform WGS-84 coordinates into local Cartesian coordinates and vice versa

Note 1 to entry: The conversion of geodetic positions produced using GNSS (WGS-84) positioning into local worksite northing, easting and elevation (orthometric height) coordinates.

3.1.10
northing, easting, and elevation
NEE
Cartesian axes X (easting), Y (northing), and Z (elevation) in surveying

Note 1 to entry: Elevation is also known as orthometric height.

3.1.11
production data
data which is used to determine progress towards a target outcome

Note 1 to entry: The measurement of progress can vary based on site. For example, streaks and acres are two measurements of progress.

3.1.12
project data
basic data related to a construction project passed from the design phase to the field

Note 1 to entry: Project data includes data related to the project, tasks, as well as references to design files.

3.1.13
site management system
SMS
system which serves end-users in managing a construction worksite by managing design objects, maintaining an aggregation of as-built and production data, and managing project data

Note 1 to entry: The SMS supports end-users in managing a mixed-fleet of machines on a worksite, and holds the authoritative data for a worksite.

3.1.14
site reporting system
SRS
system which serves end-users in analysing the performance of a construction worksite by providing means to report on or otherwise post-process data

Note 1 to entry: The SRS does not provide data, it only retrieves it.

3.1.15
streak
journal of working tool positions and attributes

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

3.1.16
survey
process of making the measurements that are necessary to determine the relative position of points above, on, or beneath the surface of the Earth, or to establish such points

[SOURCE: ISO 22932-1:2020, 3.2.1.32, modified — "Science or art" replaced with "process".]

3.1.17
survey point
location collected on the worksite from survey equipment or grade control equipped machines

Note 1 to entry: Survey points have several use cases, such as mapping existing objects, as-built reporting, progress reporting, or quality control testing.

3.1.18
topographic data
data in three-dimensional form representing the surface of the ground or any layer of the finished surface and comprised of as-of-present (or as-built) and as-designed data

[SOURCE: ISO 15143-1:2010, 3.2.46, modified — “Road” replaced with “ground”, and “road” replaced with “finished surface”.]

3.1.19
vendor integration system
VIS
system, connected to on-machine grade control solutions, which makes vendor machine data available to the SMS for a worksite, and makes SMS worksite data available to machines

3.1.20
working tool
component on an earth-moving machine that engages the dirt

Note 1 to entry: Examples of working tools are blades on a motor grader or the bucket on an excavator.

3.1.21
worksite
physical location or virtual representation of the operation of a fleet of mobile equipment generally identified as construction machines or measurement equipment where the machines are used to perform work

3.2 Terms specific to field positioning

3.2.1
base station
non-moving GNSS receiver that provides local worksite observables, often called corrections or signals

Note 1 to entry: These signals are used in RTK processing.

Note 2 to entry: See also *real-time kinematic* (3.2.14).

3.2.2
coordinate
one of a sequence of numbers designating the position of a point

[SOURCE: ISO 19111:2019, 3.1.5, modified — Note 1 to entry has been removed.]

3.2.3
coordinate reference system
coordinate system that is related to an object by a datum

[SOURCE: ISO 19111:2019, 3.1.9, modified — Notes to entry have been removed.]

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

3.2.4
datum
reference frame
parameter or set of parameters that realize the position of the origin, the scale, and the orientation of a coordinate system

[SOURCE: ISO 19111:2019, 3.1.15]

3.2.5
ellipsoid
reference ellipsoid
geometric reference surface embedded in 3D Euclidean space formed by an ellipse that is rotated about a main axis

[SOURCE: ISO 19111:2019, 3.1.22, modified — Domain and Note 1 to entry have been removed.]

3.2.6
flattening
ratio of the difference between the semi-major (a) and semi-minor axis (b) of an ellipsoid to the semi-major axis: $f = (a - b)/a$

[SOURCE: ISO 19111:2019, 3.1.28, modified — Symbol and Note 1 to entry have been removed.]

3.2.7
global navigation satellite system
GNSS
system consisting of several satellites in different orbital planes, which allow absolute navigation solutions as well as highly precise (e.g. differential) positioning and broadcasting of time due to the global coverage

[SOURCE: ISO 9849:2017, 3.1.5, modified — Note 1 to entry and all examples have been removed.]

3.2.8
GNSS receiver
electronic device that receives and processes the signals from GNSS satellites in order to provide position, velocity and time (of the receiver)

[SOURCE: ISO 9849:2017, 3.1.5.1]

3.2.9
measurement equipment
apparatus installed on a construction machine or placed on a worksite in order to acquire information on site conditions and situation

Note 1 to entry: Examples of measurement equipment include total stations, robotic total stations, theodolites, GNSS receivers, retroreflectors, 3D scanners, subsurface locators, drones etc.

[SOURCE: ISO 15143-1:2010, 3.2.25, modified — Note 1 to entry has been added.]

3.2.10
meridian
intersection of an ellipsoid by a plane containing the shortest axis of the ellipsoid

Note 1 to entry: This term is generally used the describe the pole-to-pole arc rather than the complete closed figure.

[SOURCE: ISO 19111:2019, 3.1.42]

3.2.11
positioning sensors
object or device used to measure points on a surface that are aggregated to create as-built topographic data

Note 1 to entry: Examples of positioning sensors include the tool-tip of the working tool on a machine, traditional field measurement equipment, unmanned aerial vehicles, LIDAR systems, stereo cameras, etc.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

3.2.12
projection
projected coordinate reference system

Note 1 to entry: A map projection is a coordinate conversion from an ellipsoidal coordinate system to a plane.

3.2.13
Radio Technical Commission for Maritime Services
RTCM
standards organization focused on in-depth radio communication and radio navigation areas

Note 1 to entry: The RTCM features two Special Committees that produce standards applicable to construction worksites:

Note 2 to entry: — SC 104, Differential GNSS Service — see RTCM 10403.3 *Differential GNSS (Global Navigation Satellite Systems) Services*, Version 3.2 or higher;

Note 3 to entry: — SC 135, Radio Layer for Real-Time DGNSS Applications.

3.2.14
real-time kinematic
RTK
real-time processing algorithm technique of mobile GNSS receivers using the carrier phase of GNSS observations for a positioning of the mobile GNSS receiver within a reference network, or from a single base station, in a low cm-level resolution

Note 1 to entry: In real-time kinematic (RTK) application, measurements of the phase of the signal's carrier wave are used to provide real-time corrections. By a data link from the reference station to the rover station, the corrections are transmitted to enhance the precision of the position up to cm-level.

[SOURCE: ISO 9849:2017, 3.1.5.4, modified — "or from a single base station" and "resolution" added.]

3.2.15
WGS-84
reference system used in the satellite-based positioning system NAVSTAR Global Positioning System (GPS)

Note 1 to entry: The World Geodetic System (WGS) is a standard for use in cartography, geodesy, and navigation. The latest revision is WGS-84.

[SOURCE: ISO 13184-2:2016, 3.11, modified — "WGS-84 coordinate system" changed to "WGS-84".]

3.3 Terms specific to data exchange

3.3.1
application programming interface
API
invocation method or associated parameter used by one piece of software to request actions from another piece of software

[SOURCE: ISO/IEC 18012-1:2004, 3.1.1. modified — "Collection of" removed.]

3.3.2
background drawing
non-measurable drawing that is intended for view-only purposes in the field

3.3.3
coarse-grained
authorization statements which can be minted in the token, such as the OAuth scopes

Note 1 to entry: An example is that a VIS can read but not write.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

3.3.4
endpoint
address for the provider's server for a specific API method

Note 1 to entry: Endpoints are typically accessed by a URL.

3.3.5
end-user
real person who has authentication and authorization access for a tenant within an SMS or VIS

3.3.6
equipment_ID
user-friendly name provided by an end-user and used to identify equipment

Note 1 to entry: The end-user provides this information to the VIS provider during registration.

Note 2 to entry: This concept is also identified in ISO/TS 15143-3.

3.3.7
fine-grained
detailed statements which are made in a policy, to which the token refers

Note 1 to entry: An example is that a VIS can read machine as-built for graders and bulldozers, but can only write data for bulldozers.

3.3.8
media type
description of the content type of a message in conformance with IETF RFC 2045

3.3.9
policy
authorization information stored in the SMS and VIS which grants access to worksite resources

3.3.10
system-of-record
information system considered to be the authoritative source for a specific piece of information

3.3.11
tenant
organization or customer being represented within an SMS or VIS

3.3.12
timestamp
date and time at which a data point is created

3.3.13
token
sequence of characters representing either a verified identity or access, or both

[SOURCE: ISO 20078-1:2021,3.2.8, modified — Notes to entry have been removed.]

3.3.14
VIS asset ID
identifier created in conformance with IETF RFC 4122 by a VIS for an asset such as an earth-moving machine

3.4 Terms specific to codelist

3.4.1
codelist
dictionary and visualization scheme for worksite design model, as-built and survey data objects

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

3.4.2
category code
substitutes CAD layer concept providing human readable names and visualization for a group of design model, as-built and survey data objects through codelist concept

3.4.3
feature code
type and codelist attributes for design, as-built and survey objects

3.4.4
codelist attribute
individual triple (name, type, unit) within a feature code or category code

3.4.5
code group
virtual group holding a number of category and feature codes

4 System overview

4.1 Service roles

To help manage their worksites, construction contractors choose a software system which serves as the site management system (SMS) role for their operation. This software:

— stores all design files and project data;

— collects and stores as-built and production data for the worksite;

— provides a way to manage and report on the worksite.

A worksite frequently utilizes grade control solutions from different vendors. These vendors provide systems which implement the vendor integration system (VIS) role. Server-to-server data communication happens between an SMS and one or more VIS. A VIS pulls worksite design files and project data from the SMS, and pushes as-built and production data from the worksite to the SMS. The VIS also pulls as-built and production data from the SMS, as needed.

Each VIS exchanges data with the connected grade control machines and other field equipment that it manages.

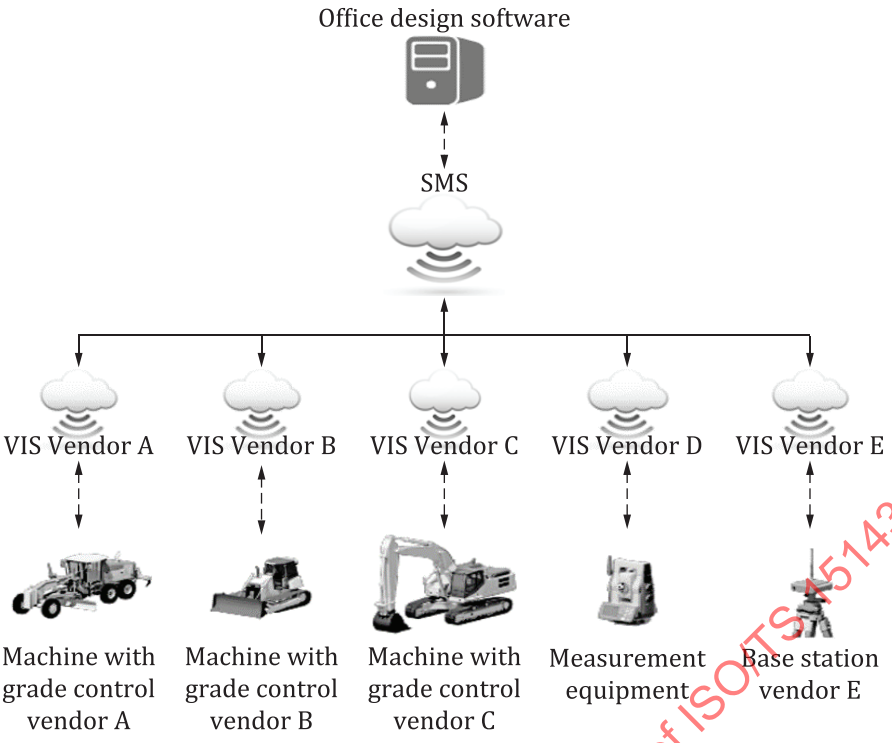
The data available from a worksite can be made available to other systems for the purpose of reporting or other post-processing. A vendor can provide a system for this purpose, referred to as a site reporting system (SRS). When acting as an SRS, the system shall not provide data, only retrieve it.

One software system can implement any combination of SMS, VIS and SRS roles.

Figure 4.1 illustrates the connection between VISs and SMS, and their relationship within the scope of this document.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved



Key

In scope of ISO/TS 15143-4 (this document).

Not in scope of ISO/TS 15143-4 (this document).

Figure 4.1 — Relationships between VIS and SMS

Due to the wide variety of VIS solutions and capabilities, not all requirements in this document are universally applicable to all VIS solutions. VIS solutions vary depending on scope of their product. Examples include:

— Not all VIS offer solutions that generate the breadth of data outlined in this document. For example, some VIS do not offer surveying or compacting solutions.

— There can be some VIS that do not offer solutions that leverage all included data (e.g., codelists).

On the other hand, an SMS will interact with multiple VIS's and in multiple use cases.

The following apply:

a) The requirements in the document apply equally to VIS and SMS, unless specifically noted in the relevant clause.

b) For solutions offered by the VIS or SMS, all relevant clauses in this document apply.

c) All requirements in this document are universally applicable to all SMS solutions.

4.2 Project data

Project data is the data passed from the office to the field required to describe a construction project happening at a worksite. It consists of metadata such as materials, planning data such as work orders, and references to design files. This document defines a set of data types for exchanging project data between an SMS on one end and either a VIS or an SRS on the other end.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

4.3 Design model exchange format

Worksite design information exists in various different formats and levels of detail. This information is harmonized and exported to a file in ISO LandXML format, as defined in Annex L. The ISO LandXML includes objects relevant to field equipment, including (but not limited to) triangle mesh surfaces, lines, alignments, and pipe networks.

The SMS, VIS and field equipment may use proprietary design model exchange formats internally. However, for data exchange between SMS and VIS from different vendors, they shall convert such proprietary formats to the design model exchange formats defined in this document.

4.4 Common localization

The SMS supplies information to a VIS which is necessary to enable its field equipment to have a consistent transformation of GNSS positions to the local northing, easting and elevation coordinates (NEE).

Field equipment that uses RTK GNSS field for positioning require the use of a base station, either local or remote. This document defines the methods of interoperability for field equipment and the base station operating on a worksite.

4.5 Units and formatting

All measurements in this document are in SI units in accordance with ISO 80000-1. Units are assumed to be base units except for cases where non-base units are explicitly stated with the measurement.

ISO LandXML design data shall conform to ISO 80000-1 and is expected to convey values in base SI units.

Certain nouns in this document are programmatic and key elements of implementation, such as types, enumerations, and URL elements. For clarity, these programmatic nouns are italicized and use a different font from the rest of the document.

4.6 As-built and production data exchange

As-built and production data includes survey data, machine implement positions and sensor measurements, and machine local triangulated surfaces. This document defines a set of data types for exchanging as-built and production data between an SMS on one end and a VIS or an SRS on the other end.

4.7 Concept diagram

See Figure 4.2 for an illustration of the high level scheme.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

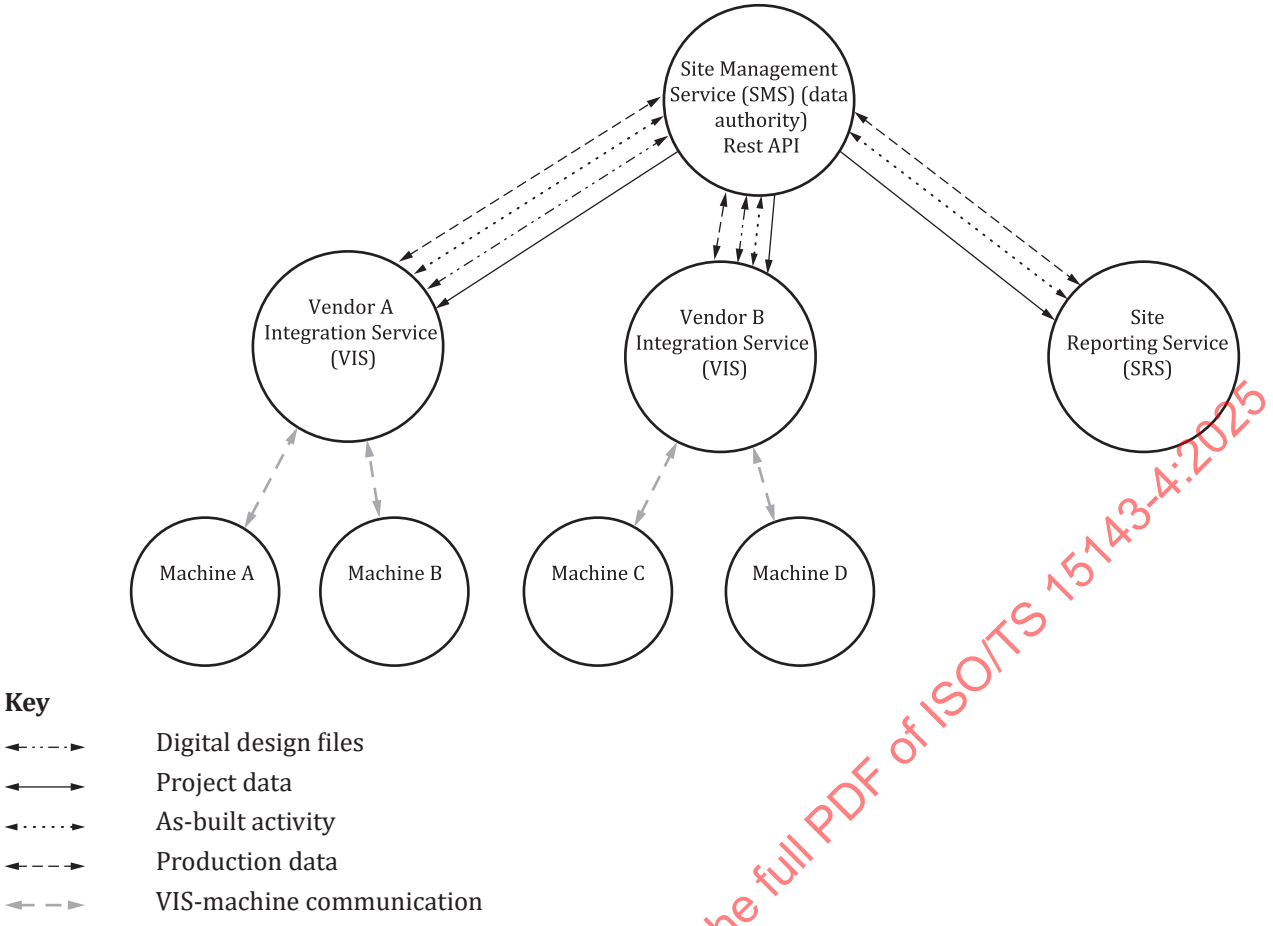


Figure 4.2 — Data flow

As shown in Figure 4.2, project managers administer the worksite from the SMS.

The SMS is the source-of-truth for the aggregated Project data and provides access to it via a REST API. The VIS collects vendor machine worksite data and pushes it to the SMS. The VIS pulls worksite data from the SMS and distributes it to vendor machines.

5 Method of data exchange between servers

5.1 General

The SMS shall implement REST and WebSocket APIs for all data exchange as described in Annex B, Annex F, Annex I and Annex J.

The SMS REST APIs should support HTTP2.0 in conformance with IETF RFC 7540 (or later). The SMS REST APIs shall support HTTP1.1 in conformance with IETF RFC 7231, secured with TLS. HTTP1.1 without TLS shall not be supported. A VIS should preference establishing HTTP2.0 (or later) client connections to an SMS, falling back to HTTP1.1 and TLS where HTTP2.0 (or later) is not available. For all SMS HTTP endpoints, the following requirements apply:

— The SMS shall support the HTTP header “Accept-Encoding” with value “gzip” and respond with “Content-Encoding” value “gzip”.

— For POST and PUT, the SMS shall support the HTTP header “Content-Encoding” with value “gzip”.

— A VIS shall POST and PUT with appropriate “Content-Type” headers, such as “application/json” for JSON bodies.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

— The SMS shall respond with appropriate "Content-Type" headers, such as "application/json" for JSON bodies.

— A VIS and an SRS shall follow SMS redirects.

5.2 API URL

A VIS or SRS shall obtain an `apiURL` for each SMS it connects to. The `apiURL` shall be used by VIS and SRS to form REST API routes as specified in Annex B. The method of how a VIS obtains the apiURL must be agreed upon between the VIS and SMS.

5.3 Discovering worksites

To facilitate the discovery of available SMS worksite resources, the SMS shall implement a worksite directory REST API as specified in Annex B. This API shall list worksites shared to a VIS or SRS end-user for data exchange. For each worksite listed, the SMS shall provide a `worksiteBaseURL`. The `worksiteBaseURL` shall be used by VIS and SRS to form REST API routes specific to the worksite, as detailed in the document.

5.4 Reporting unexpected errors

The SMS shall implement an error reporting REST API as specified in Annex B. VISs and SRSs should report unexpected errors (such as unexpected HTTP response codes and data which do not conform to schema) to the SMS using this API, allowing support staff at the SMS the opportunity to resolve issues in a timely manner.

5.5 Authorizing worksite data exchange

Data exchange, as discussed in this document, is between a VIS (or SRS) worksite resource and an SMS worksite resource. To facilitate data exchange, authorization relationships shall be established between the VIS (or SRS) and SMS worksites. While it is end-users who establish these relationships, once established the data exchange shall operate regardless of those end-users being logged into their respective system.

The SMS, VIS (or SRS) shall implement the mechanism needed for establishing these authorization relationships as specified in Annex A.

5.6 Versioning

In order to allow for publication of different versions of this document, and of the schemata used by this document, the following definitions apply:

— the major version shall be of the value 2024. Future versions of this document are expected to change this value to a numerically greater one;

— where files are published in accordance with Annex N, each published JSON Schema file shall have a minor version of the form YYYYMMDD, the date of publication of the schema file;

— the semantic version of a published JSON schema file shall be formed as `{major version}.{minor version}`.

Changes to types are managed by the maintenance agency as discussed in Annex N.

6 Method of project data exchange

6.1 General

An SMS shall be responsible for maintaining worksite project data. Project data shall be communicated as JSON objects as described in Annex E. An SMS shall make project data available by providing the endpoints specified in Annex F and in Clause 6.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

6.2 Obtaining project data

An SMS shall provide two separate APIs for obtaining data.

a) A long-poll event API to provide a synchronization mechanism for a VIS described in Clause F.3.

b) A REST API to provide current project data elements to an SRS.

A VIS shall use the synchronization mechanism to both initialize the VIS project data and maintain synchronization with the SMS project data. A VIS shall pass applicable project data to worksite equipment.

A SRS may obtain a snapshot of current project data by sending GET requests to the relevant endpoint for each data type, as specified in Clause F.4.

An SRS shall not contribute any project data or upload files.

6.3 Contributing project data

The SMS is responsible for maintaining project data and may use its own methods to create it.

A VIS may contribute project data that it owns by sending a POST request to the appropriate endpoint as per Clause F.12. For the purpose of this document, the VIS is considered to only own data concerning the types *asset* and *operator*.

On receipt of such data, where correctly authorized by policy, the SMS shall:

a) add the required metadata as described in Clause E.2;

b) not semantically modify the data in any other way;

c) return this composite object to the VIS as the response body of the request;

d) publish this object to any VISs currently subscribing to changes as per the above Clause 6.2.

A VIS may contribute an update to project data that it created by sending a PUT request to the appropriate endpoint as per Clause F.12.

A VIS may delete project data that it created by sending a DELETE request to the appropriate endpoint as per Clause F.12. The VIS should delete the *asset* and *operator* objects when they permanently leave the site. When a VIS deletes an *asset* object, the VIS should notify the physical asset that their association with the worksite has terminated and it should delete the worksite's data from its local storage.

A VIS may upload files (such as from a survey) to an SMS by following the workflow described in Clause F.7.

7 Method of as-built and production data exchange

7.1 General

As-built and production data originate from a variety of measuring equipment and positioning sensors in all phases of the construction process. Data collected by measuring equipment and positioning sensors is communicated from machine to VIS.

A VIS provides this data to the SMS using types defined in Annex G and Annex H. The SMS aggregates the data and makes the aggregated data available to connected VISs or SRSs using the same types. Further aggregation of the data, such as to provide reports, is not in the scope of this document.

The SMS shall implement REST and WebSocket APIs as defined in Annex I and Annex J to enable this worksite data exchange.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

There are two separate use-cases for as-built and production data.

— The need for an authoritative record of machine activity on the worksite. This use case is supported by the system-of-record data stream.

— The need for a low-latency data stream to support the here-and-now aspects of machine state, location, and posture. This use case is supported by the real-time data stream.

These two use cases trade off timeliness and reliability, with the system-of-record stream favouring reliability and the real-time stream favouring timeliness.

While the data representation in the two streams differ, the position and state described by them should match.

7.2 System-of-record data stream

The SMS shall implement a HTTP REST API for the system-of-record data stream as described in Annex I. This stream contains as-built and production information in self-contained blocks designed to be an authoritative data record.

The system-of-record data types are defined in Annex G. A VIS shall collect data objects of these types from worksite equipment and push them to the SMS. On receipt of these objects, the SMS shall append them to the system-of-record and make them available to polling VISs and SRSs. The SMS shall not modify the objects before making them available. If worksite equipment needs this data, the VIS shall poll the SMS API and pass the received objects to the equipment.

On the SMS, the system-of-record data stream is conceptualized as a sliding window of data aggregated across machines. The sliding window shall keep messages for a duration of at least 1 week from receipt. This allows for the SMS to persist aggregated data only for the duration of the window and for a VIS to resume pulling data from within the window in the event of system failure and subsequent recovery. The system-of-record shall present messages in the order they were received (transaction order), which is not necessarily the order in which events occurred in the field (temporal order).

7.3 Real-time data stream

The SMS shall implement a WebSocket API for the real-time data stream as described in Annex J. This is a low-latency stream providing a best-effort communication about the position and pose of machine assets on worksite. This document makes no hard real-time or reliability constraints on the transmission of data. Accordingly, the data shall not be used for safety related purposes.

The real-time data types are defined in Annex H. A VIS should publish data objects of these types to the SMS. On receipt of these objects, the SMS shall publish them to subscribed VISs. The SMS shall not modify the objects before publishing them. A VIS may subscribe to this API to communicate the locations of other vendors' assets to its own connected assets.

On the SMS, the real-time data stream is conceptualized as a message broker. A VIS may publish and subscribe to the real-time data stream for a worksite.

A SRS shall not publish or subscribe to the real-time data stream.

8 Method of design model data exchange

8.1 Design file format

Source design models shall be presented in the design file format defined in Annex L, as it is communicated between the SMS and VIS. The VIS may convert the design file to a proprietary format prior to sending it to machine or field equipment. Conformance to this standard does not require grade control systems to directly use this design file format.

The design files communicated between an SMS on one end and a VIS or an SMS on the other shall validate against the ISO LandXML schema defined in Annex L.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Content of the provided ISO LandXML schema focuses on elements needed to execute the tasks completed by typical grade control systems on a worksite.

8.2 Background drawings

A worksite's project data may reference background drawings. A background drawing is used for aiding in visualization of the design, but shall not be used for any field measurements or the grade control system for automatic control of the working tool.

Background drawings shall use DXF file format, and shall:

— use meters as distance unit, indicated by setting the `$INSUNIT` header variable to value 6;

— follow the specification of the DXF file of AC1012(R13) or later;

— set the $`ACADVER` header variable to the version in use.

8.3 Codelist

Codelist is a common concept in the surveying industry. In this document, codelist is applied for design, as-built and survey data objects.

Codelist may be used in combination with design files at VIS file import phase to enrich the contents of the design model file. The design model format defined in Annex L natively does not offer such functionality. See Annex M for more details.

9 Method of local position correction and localization

9.1 General

When field equipment using GNSS positioning comes onto a worksite, it should use the worksite's base stations together with the localization that is defined and communicated through project data.

9.2 RTK GNSS position correction

Clause 9 shall apply only to those worksites that support RTK GNSS.

The base station shall make the base station observables available in accordance with RTCM 10403.3 (V3.3 or higher). Multi system message (MSM) shall be used.

This document does not require use of a narrow band UHF radio on the worksite. However, if a narrow band UHF radio is used then an RTCM 13500.1 (V1.0 or above), compatible radio shall be used. MSM3or MSM4 should be used with UHF delivery.

When remote base stations with IP (internet) based delivery are utilized for a worksite, RTCM 10410.1 (NTRIP V2.0 or above) shall be used to provide the connection.

9.3 Localization

Within this document, all coordinates are reported in a local datum rectangular system consisting of NEE coordinates. It can be used in reverse to transform site coordinates into WGS-84.

The SMS should provide the localization to the VIS as defined in Annex K, and the VIS should provide this information to field equipment in order to allow for a consistent transformation of GNSS positions to the local NEE coordinates. However, it is not required for VIS-connected field equipment to use the localization provided. Whenever the localization provided by the SMS is not used, users should ensure that the NEE values are consistent with the worksite coordinates.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

The localization system defined in Annex K does not attempt to implement every possible coordinate transformation. Rather, Annex K provides a solution for the most common transformations. When a worksite coordinate system is not able to use the defined localization system, each VIS and its connected field equipment shall use its own method for a consistent transformation to the worksite coordinates.

This document focuses on the transformation of coordinates generated using RTK methods. Therefore, time dependent transformations are not supported. If a VIS and its connected field equipment do not use the same base station observables source as defined by the SMS for the worksite, then the VIS should ensure that the position of its source of observables is consistent with the site base station in order to be able to use the localization provided by the SMS and VIS.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex A
(normative)

Authorization mechanisms for worksite data exchange

A.1 Authorization relationships between SMS and VIS worksite

Both SMS and VIS model a physical worksite with one or more virtual worksites. This annex concerns virtual worksites, which will subsequently be referred to as "worksites" without further qualification. SMS and VIS are typically multi-tenant systems. Accordingly, a worksite resource is owned by tenants at the VIS and SMS.

Data-exchange between VIS and SMS worksites shall be limited to a star topology. Each topology may have only one SMS worksite. This single-master limitation may be relaxed to support multi-master in future revisions of this document.

For a VIS, a star topology means that each (VIS, worksite) pair shall relate to at most one (SMS, worksite) pair. Accordingly, a VIS shall prevent end-users from establishing data-exchange to multiple SMS worksites for the one VIS worksite. For each VIS worksite, a set of VIS agents should operate to both push and pull worksite data (such as project data, as-built and production data system-of-record, as-built and production data real-time) to and from the SMS worksite. These agents shall transport data (using HTTP long-polling or WebSockets) for assets connected to the VIS worksite. Accordingly, these agents shall continue to run independently of the end-user (who initiated the data-exchange) being logged in.

A VIS shall maintain the following for each (VIS, worksite) to (SMS, worksite) relationship:

— the (SMS, worksite) `worksiteBaseURL` obtained from the worksite directory as specified in Annex B;

— authorization parameters and scopes (also obtained from the worksite directory) required for the authorization and token requests;

— security token and refresh token for each agent;

— granted OAuth2.0 scopes;

— any other access policy information, such as permissions specifying which assets on the VIS worksite may exchange data with the SMS worksite.

For an SMS, each (SMS, worksite) pair may relate to zero or more (VIS, worksite) pairs.

The SMS (or its delegated OAuth2.0 authorization server) shall maintain the following for each (SMS, worksite) to (VIS, worksite) relationship:

— issued refresh tokens;

— granted OAuth2.0 scopes;

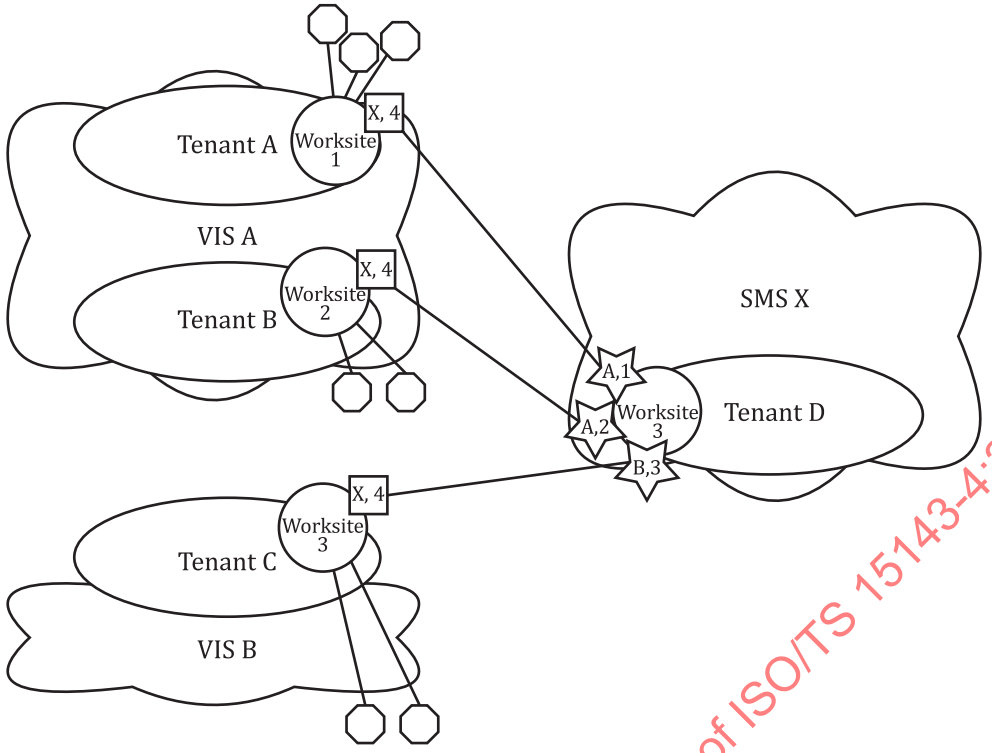
— any other access policy information, such as permissions specifying which assets on the SMS worksite may exchange data with the VIS worksite.

To avoid multi-master topologies, for a system which implements both SMS and VIS roles, on any worksite where the system operates as an SMS, it shall not also operate as a VIS, connecting to an upstream SMS. This does not preclude an SMS from having vendor-specific machines connecting directly to it.

Figure A.1 illustrates the high-level concept, showing the intended relationships needed for data-exchange between SMS and VIS worksites. Each worksite in the diagram represents the same physical worksite.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved



Key

Cloud shapes represent a VIS or SMS (most likely SaaS multi-tenant systems).

Ovals represent a tenant within a VIS or SMS.

Circles represent a worksite within a tenant.

Squares represent data maintained by the VIS for an SMS worksite relationship.

Stars represent data maintained by the SMS for a VIS worksite relationship.

Octagons represent a machine or device connected to a VIS.

Figure A.1 — Authorization relationship overview

A.2 OAuth2.0

A.2.1 General

Authorization between VIS and SMS is broken into layers.

a) The client credentials layer provides credentials for further layers.

b) The worksite directory access layer authorizes a VIS user to discover worksites available for data exchange at an SMS.

c) The worksite resource access layer authorizes a VIS to access SMS worksite-specific APIs, such as project data or system-of-record.

d) The access policy layer allows SMS and VISs to control fine-grained access, such as which assets participate in the data exchange.

These layers are described in Subclauses A.2.2 to A.2.4.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Refer to Annex C for relevant example end-user stories.

A.2.2 Client credential layer

Each VIS shall acquire OAuth2.0 client credentials (in conformance with IETF RFC 6749:2012, Subclause 1.3.4) that are required for the authorization code grant (in conformance with IETF RFC 6749:2012, Subclause 4.1) for each SMS it accesses. VIS client registration with an SMS (in conformance with IETF RFC 6749:2012, Clause 2) is outside the scope of this document.

A.2.3 Worksite directory access layer

The SMS shall implement a worksite directory API as described in Annex B. The worksite directory lists SMS worksite resources available to an authenticated identity at the SMS.

The SMS OAuth2.0 authorization server (AS) shall support Authorization Code Grant (in conformance with IETF RFC 6749:2012, Subclause 4.1) requests scoped with `iso15143-4.worksite_directory`. The SMS OAuth2.0 AS should issue a refresh token (in conformance with IETF RFC 6749:2012, Subclause 1.5) if an `offline_access` scope is requested.

A VIS shall obtain an OAuth2.0 access token using Authorization Code Grant (in conformance with IETF RFC 6749:2012, Subclause 4.1) with a scope of `iso15143-4.worksite_directory` for use with the SMS worksite directory. A VIS may long-poll the worksite directory for the purpose of discovering those worksites that it can establish data-exchange with, regardless of a VIS end-user being online. In this case, a VIS shall include an `offline_access` scope.

To establish data-exchange with a worksite listed in the worksite directory, a VIS shall use the worksite resource access layer described in A.2.4.

A.2.4 Worksite resource access layer

A.2.4.1 General

The basic conditions for authorizing worksite data-exchange are:

— a VIS end-user is not required to be the owner of the SMS worksite resource;

— a VIS end-user requests access on behalf of the VIS agents which perform worksite data exchange;

— granted access remains in effect independently of the VIS end-user;

— fine-grained consent must be obtained for VIS access to an SMS worksite.

These conditions are not supported by common OAuth2.0 usage. The basic OAuth2.0 mechanism is sufficient to implement coarse-grained authorization requests based on an owner's profile. For authorizing worksite data exchange, however, there are three problems which basic OAuth2.0 does not readily address. These problems and their solutions are:

a) The VIS end-user is possibly not the resource owner at the SMS. Accordingly, the SMS should provide a way to share worksite resources with external users.

b) Requests to specific SMS worksites require fine-grained authorization, such that a competitor VIS client can never access worksite data not intended by the SMS end-user. To overcome the lack of readily available OAuth2.0 fine-grained support, the worksite directory states optional authorization parameters and authorization scopes required for gaining access to each worksite. This requirement is detailed in Subclause A.2.4.2.

c) A VIS shall have access tokens for its long-running agents which continue to exchange worksite data independently of the VIS end-user who established the data exchange. Accordingly, OAuth2.0 Client Credentials Grant (in conformance with IETF RFC 6749:2012, Subclause 4.4) shall be used to gain access tokens for long-running VIS agents. This requirement is detailed in Subclause A.2.4.3.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

A summary of the flow to establish worksite data exchange between VIS and SMS is as follows: a VIS end-user authenticates at the SMS authorization server using OAuth2.0 Authorization Code Grant (in conformance with IETF RFC 6749:2012, Subclause 4.1). For the OAuth2.0 authorization and token requests, the VIS shall build the authorization endpoint URL (in conformance with IETF RFC 6749:2012, Subclause 3.1) and token endpoint URL (in conformance with IETF RFC 6749:2012, Subclause 3.2) using authorization parameters and scopes stated for the specific worksite in the worksite directory as described in Clause B.1. This access token is used to create a policy at the SMS as described in Clause B.2. With a policy now in place, OAuth2.0 Client Credentials Grant (in conformance with IETF RFC 6749:2012, Subclause 4.4) is used to obtain access tokens used by the VIS agents to access the SMS REST endpoints as detailed in Annex F, Annex I and Annex J. This flow is detailed in A.2.4.2 to A.2.4.4.

A.2.4.2 Access tokens for policy creation

Each worksite listed by the worksite directory includes the following fine-grained authorization information needed by a VIS to request access tokens for policy creation:

a) Authorization parameters: A VIS shall add worksite-specific URL parameters to the SMS authorization endpoint (in conformance with IETF RFC:2012 6749, Subclause 3.1) and token endpoint (in conformance with IETF RFC 6749:2012, Subclause 3.2) when authorizing for policy creation. This allows the SMS OAuth2.0 AS to use standards such as resource indicators (in conformance with IETF RFC 8707), or custom parameters (for JWT claims) for implementing fine-grained authorization.

b) Authorization scopes: a VIS shall include worksite-specific scopes (in conformance with IETF RFC 6749:2012, Subclause 3.3) along with the `iso15143-4.worksite_directory` scope when authorizing for policy creation. This allows SMS OAuth2.0 AS to alternatively use dynamic scopes for implementing fine-grained authorization.

c) A worksite URL: a VIS shall use the worksite URL as a root to form REST route URLs detailed in Clause B.2.

d) An `auto_create` flag: when this flag is true, authorization parameters and scopes shall not be provided.

An SMS should use standards such as resource indicators (in conformance with IETF RFC 8707), custom parameters for JWT claims, or dynamic scopes to implement fine-grained resource access. This means a VIS can only create a policy for a worksite when the VIS user performs a browser-agent authorization code flow specific to the worksite. Consequently, a VIS client can only access worksites which its end-users have previously explicitly authorized using a fine-grained authorization code flow. This is highly desirable when the SMS and VIS are competitors in the marketplace.

When the `auto_create` flag is true for a worksite, a VIS can create a policy for the worksite using the same access tokens as used for the worksite directory.

A.2.4.3 Access tokens for worksite data exchange

The SMS shall use OAuth2.0 Client Credential Grant (in conformance with IETF RFC 6749:2012, 4.4) to secure the SMS REST endpoints detailed in Annex F, Annex I and Annex J. The SMS shall permit access to worksites only where policies have been created for them by a VIS, as detailed in Clause B.2.

A VIS shall use the client credentials specified Subclause A.2.2 to request an access token, using OAuth2.0 Client Credentials Grant (in conformance with IETF RFC 6749:2012, Subclause 4.4.2), with scopes described in Subclause A.2.4.4. A VIS shall include the parameters and scopes stated in the policy when requesting tokens for the worksite as detailed in Clause B.2. A VIS shall use such access tokens for the SMS REST endpoints detailed in Annex F, Annex I and Annex J.

A.2.4.4 Authorization scopes

The following scopes apply when requesting access tokens for worksite resource access:

— `iso15143-4.project_data.read` grants project data read;

— `iso15143-4.project_data.write` grants project data write;

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

— `iso15143-4.production_data.sor.read` - grants as-built and production data system-of-record read;

— `iso15143-4.production_data.sor.write` grants as-built and production data system-of-record write;

— `iso15143-4.production_data.rt.read` grants as-built and production data real-time read;

— `iso15143-4.production_data.rt.write` grants as-built and production data real-time write.

A.2.5 Access policy layer

When implementing policies, an SMS should provide a mechanism for fine-grained control of the data made available to any VIS tenancy. Examples of such fine-grained control can include:

— In the worksite directory, only share worksites that the resource owner has explicitly marked to be shared with a particular VIS.

— In the project data API, only share *file_entry*s and *file_package*s to a VIS when there is a related *assignment* for an asset or operator on that VIS.

— Do not share anything through the system-of-record API to a particular VIS.

Likewise, a VIS should implement a mechanism to provide fine-grained control of the data pushed to an SMS for a worksite. Examples of such fine-grained control can include:

— For system-of-record, only publish data related to particular assets or asset types on the VIS worksite.

A.3 Authorization relationships between SMS and SRS worksite

An SRS shall authenticate with an SMS in the same way as described for a VIS. The following scopes shall not apply for an SRS-only client:

— `iso15143-4.project_data.write`

— `iso15143-4.production_data.sor.write`

— `iso15143-4.production_data.rt.write`

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex B
(normative)

Worksite directory SMS API

B.1 Worksite directory API

B.1.1 General

The SMS shall implement a REST API to get a directory of worksites which can be accessed by an authenticated end-user. Routes in this REST API are detailed within Clause B.3. Each route is relative to the `apiURL` as discussed in Clause 5.2.

A VIS may use this API to discover worksites when required by an end-user. In addition, a VIS may poll the worksite directory to obtain worksites which can be automatically created at the VIS. A VIS shall not poll the SMS worksite directory more often than every 30 seconds.

Since the worksite directory contents are expected to change relatively rarely, and a VIS can frequently poll the API to quickly find any new worksites it can auto-create, this API uses conditional requests (in conformance with IETF RFC 7232). An SMS shall include the last-modified header in its responses, containing the last modification timestamp for the representation of the full list of worksites it returns. A VIS should include this timestamp in the If-Modified-Since header of further requests, so that the SMS can respond with 304 Not Modified as applicable.

The API accepts a `limit` parameter that defines the number of objects returned per page. Each call returns the following:

— a page of objects;

— a `next` link to obtain the next page of objects, if there is one.

The exact detail of the next link is the responsibility of the SMS and the VIS shall treat it as opaque. That is, it shall not rely on it being structured in any particular way. The VIS shall page through all available objects in one session and not retain `next` links once used. The SMS shall make every effort to return all objects current at the time of call.

An SRS may use this API in the same way as a VIS does.

B.1.2 Authorizing

These endpoints shall be accessed with an access token obtained using OAuth2.0 Authorization Code Flow, as detailed in Subclause A.2.3. The SMS shall require the scope `iso15143-4.worksite_directory`.

Accordingly, worksite access tokens (as per Subclause A.2.4.2), and security tokens (as per Subclause A.2.4.3) shall not be used for worksite directory access.

B.1.3 Worksite directory routes

B.1.3.1 worksite_directory routes

B.1.3.1.1 GET {apiURL}/2024/worksite_directory

Description

Get a list of worksites accessible to the authenticated user of the SMS.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Parameters

This route shall support the parameters described in Table B.1.

Table B.1 — GET worksite_directory parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).
If-Modified-Since	header	*string*	Optional	If-Modified-Since header (in conformance with IETF RFC 7232:2014, Subclause 3.3).

Responses

This route shall provide responses as described in Table B.2.

Table B.2 — GET worksite_directory responses

Code	Schema	Description	Headers
200	*list_of_worksite*	Successful operation	Last-Modified
304	N/A	Not Modified	
401	N/A	Unauthorized	

Response headers

Responses from this route shall provide response headers as described in Table B.3.

Table B.3 — GET worksite_directory response headers

Header	Type	Description
Last-Modified	*string*	Last-Modified header (in conformance with IETF RFC 7232:2014, Subclause 2.2).

B.1.4 Worksite directory definitions

B.1.4.1 *list_of_worksite*

Description

A list of worksites, together with links for REST navigation.

Properties

A *list_of_worksite* shall have the properties defined in Table B.4.

Table B.4 — list_of_worksite properties

Name	Type	Use	Description
_links	*server_list_links*	Required	Links for REST navigation.
data	array of *worksites*	Required	An array of worksite.
limit	*integer*	Required	Number of entries per page of data, other than the last.

B.1.4.2 *server_list_links*

Description

A set of links supplied to enable pagination through a list of results.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Properties:

A *server_list_links* shall have the properties defined in Table B.5.

Table B.5 — server_list_links properties

Name	Type	Use	Description
next	*character_string_url*	Optional	Link to the next page of results.

B.1.4.3 *worksite*

A worksite entry with basic project and connection information.

Properties:

A *worksite* shall have the properties defined in Table B.6.

Table B.6 — worksite properties

Name	Type	Use	Description
authorization_ parameters	*character_string_1000*	Optional	URI-encoded parameters particular to this worksite to be added to the SMS authorization and token endpoints to authorize for this worksite.
auto_create	*boolean*	Required	A flag indicating if the VIS can create this worksite regardless of a user being logged in at the VIS. An SMS shall not set this to true unless the VIS end-user is also the resource owner at the SMS, and has consented (at the SMS) to this worksite being auto-created for the requesting VIS. This consent is required in order to prevent unexpected costs at the VIS due to the auto-create feature. The `authorization_parameters` and `scopes` fields shall not be supplied when `auto_create` is true.
message	*character_string_1000*	Optional	An optional message, useful for the case where the requesting user is not the resource owner of this worksite.
owner	*character_string_255*	Optional	An optional name of the owning company or user, useful for the case where the requesting user is not the resource owner of this worksite.
scopes	array of at most 4 *character_string_255*s	Optional	Scopes particular to this worksite to be used with authorize and token requests to authorize for this worksite.
worksite_base_url	*character_string_url*	Required	The `worksiteBaseURL` used by VIS and SRS to form routes specific to this worksite, as described in Subclause 5.3.
worksite_project_data	*worksite_project_data*	Required	Basic project details from project data.

B.1.4.4 *worksite_project_data*

Description

Information provided by an SMS to enable an end-user to select a worksite. This data shall agree with the data provided by the SMS in the `basic_project_data` object of the relevant worksite.

Properties

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

A *worksite_project_data* shall have the properties defined in Table B.7.

Table B.7 — worksite properties

Name	Type	Use	Description
beginning_date	*datetime_3339*	Optional	Anticipated or actual date construction began.
boundary	*points_wgs84*	Optional	Polygon enclosing a site. In the cases where systems treat this as 2D, then altitude ("alt") should have the value of zero.
completion_date	*datetime_3339*	Optional	Anticipated or actual date construction completed.
job_code	*character_string_4*	Optional	Job code for the project.
name	*character_string_1*	Required	Name for the project.
timezone	*character_string_1*	Optional	An IANA timezone string identifying the timezone nominally associated to the project.

B.2 Worksite policy API

B.2.1 General

Each route in each API is relative to the `worksiteBaseURL` as discussed in Subclause 5.3.

B.2.2 Authorizing

These endpoints shall be accessed with an access token obtained using OAuth2.0 Authorization Code Flow, as detailed in Subclause A.2.4.2. The SMS shall require the scope `iso151434.worksite_directory`.

B.2.3 Worksite policy routes

B.2.3.1 *policy routes*

B.2.3.1.1 POST {worksiteBaseURL}/2024/policy

Description

Create a policy for the worksite. The SMS shall implement this as an idempotent operation.

Parameters

This route shall support the parameters described in Table B.8.

Table B.8 — POST policy parameters

Name	Source	Type	Use	Description
body	body	*policy_request*	Required	Policy request.

Responses

This route shall provide responses as described in Table B.9.

Table B.9 — POST policy responses

Code	Schema	Description
200	*policy*	Successful operation.
201	*policy*	Successful operation.
400	N/A	No VIS policy_key provided.
401	N/A	Unauthorized.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

B.2.4 Worksite policy definitions

B.2.4.1 *policy*

Description

The object returned from successful policy creation.

Properties

A *policy* shall have the properties defined in Table B.10.

Table B.10 — Policy properties

Name	Type	Use	Description
scopes	array of at most 4 *character_string_255*s	Optional	Scopes particular to this worksite to be added to the SMS token requests as specified in Subclause A.2.4.3.
token_parameters	*character_string_1000*	Optional	URI-encoded parameters particular to the worksite to be added to the SMS token endpoints as specified in Subclause A.2.4.3.

B.2.4.2 policy_request

The policy request object shall have the properties defined in Table B.11.

Table B.11 — policy_request properties

Name	Type	Use	Description
policy_key	*character_string_1*	Required	An opaque and unique descriptor for the worksite in the VIS making the request. This is treated as an idempotency key by the SMS to differentiate VIS worksites being associated with the SMS worksite for the purposes of policy construction.
description	*character_string_255*	Optional	A human readable hint to aid SMS users in identifying the particular VIS worksite at the SMS.

B.3 Error notification API

B.3.1 General

The SMS shall implement a REST API to provide error notifications to the SMS. Routes in this REST API are detailed in Subclauses B.3.2 through to B.3.4. Each route is relative to the apiURL as discussed in Subclause 5.2.

B.3.2 Authorizing

These endpoints shall be accessed with an access token obtained using OAuth2.0 client credentials, as specified in Subclause A.2.2.

B.3.3 Error notification routes

B.3.3.1 *error_notification routes*

B.3.3.1.1 POST {apiURL}/2024/error_notification

Description

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Allows a VIS or an SRS to provide details of an unexpected error (such as unexpected response code, schema violation) to the SMS.

Parameters

This route shall support the parameters described in Table B.12.

Table B.12 — POST error_notification parameters

Name	Source	Type	Use	Description
body	body	*error_notification_post_body*	Required	An object describing an unexpected error.

Responses

This route shall provide responses as described in Table B.13.

Table B.13 — POST error_notification responses

Code	Schema	Description
200	*error_notification_response_body*	Successful operation.
401	N/A	Unauthorized.

B.3.4 Error notification definitions

B.3.4.1 *error_notification_post_body*

Properties

A *error_notification_post_body* shall have the properties defined in Table B.14.

Table B.14 — error_notification_post_body properties

Name	Type	Use	Description
description	*character_string_1000*	Required	An English description of the error, helpful to SMS support engineers.
id	*character_string_id*	Optional	The id of the entity relating to the error, where relevant.
time	*datetime_3339*	Required	The time at which the error occurred, which can differ to the time the error is reported.
url	*character_string_url*	Required	The URL being accessed which resulted in the error.
worksite_base_url	*character_string_url*	Optional	The `worksiteBaseURL` relating to the error, where relevant.

B.3.4.2 *error_notification_response_body*

The response object currently has no fields.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex C
(informative)

Server to server use cases

C.1 General

Each of these user-stories details how the SMS worksite directory can be used to achieve the security required by worksite data exchange in a competitive marketplace.

C.2 John

C.2.1 User story

The “John” end-user story relates to the case where an owner (“John”) wants to operate his mixed-fleet by using only his SMS.

John chooses an SMS and creates accounts there. John has a mixed-fleet and requires a few VISs to have his machines connect to his SMS. John must create an account at each VIS. John logs into each VIS and connects to his SMS, giving consent for the VIS to access the worksite directory of the SMS. This is done once for many worksites. Over time, John can log into his SMS and create a worksite, and inform the SMS which VISs can access the new SMS worksite. John can also inform the SMS which VISs can auto-create the worksite. This prevents John from having to log into each VIS, yet allows John to use machines connected to that VIS on his SMS worksite.

C.2.2 Worksite directory usage

The key part of this user story is that John gives consent for a VIS to access a worksite at the SMS. Therefore, when the SMS produces a directory listing for John’s identity at that VIS, it produces the worksite entry:

— without fine-grained `authorization_parameters` and `scopes` fields,

  which means the VIS can use existing access tokens to access the new worksite resource, without John having to perform an Authorization Code Flow particular to the worksite at the VIS;

— with the `auto_create` field as true (if John indicated the VIS can auto-create the site).

C.3 Bill

C.3.1 User story

The “Bill” end-user story relates to the case where an owner (“Bill”) wants to operate his mixed-fleet using both SMS and VIS.

Bill chooses an SMS and creates accounts there. Bill has a mixed-fleet and requires a few VISs to have his machines connect to his SMS. Bill must create an account at each VIS. Bill logs into each VIS and connects to his SMS giving consent for the VIS to access the worksite directory of the SMS. This is done once for many worksites. Over time, Bill can log into his SMS and create a worksite. Bill can also log into his VISs and create worksites, using features particular to that VIS. Bill logs into a VIS and browses the worksite directory of his SMS. Bill can connect a worksite of the VIS to a worksite listed in the SMS worksite directory, establishing worksite data-exchange. This connection must be performed from a browser agent.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

C.3.2 Worksite directory usage

The key part of this user story is that Bill gives consent for a VIS to access an SMS worksite from the VIS in a browser agent. When the SMS produces a directory listing for Bill's identity at that VIS, it produces the worksite entry:

— with fine-grained `authorization_parameters` and `scopes` fields particular to the worksite,

this means the VIS must have Bill perform an Authorization Code Flow in a browser agent to mint an access token particular to the worksite;

— without the `auto_create` field.

C.4 Sally and Bob

C.4.1 User story

The "Sally and Bob" end-user story is relevant to larger projects, where there are many sub-contractors operating on a large mixed-fleet earth-works project.

Sally is a general contractor using an SMS to manage a worksite. Sally desires a sub-contractor, Bob, to also operate at the worksite. Bob uses one or more VISs to manage his mixed-fleet. Using her SMS, Sally shares her worksite with Bob providing Bob's email address to the SMS. Bob logs into one of his VISs, and then connects to the worksite directory of Sally's SMS. Bob must register a user with Sally's SMS if he does not already have credentials for that SMS. Bob selects the worksite which Sally shared with him and connects it to one of his VIS worksites.

C.4.2 Worksite directory usage

The key part of this user story is that Bob does not own the worksite resource at the SMS. Sally uses the SMS to share (or invite) Bob to the worksite by stating Bob's identity. When the SMS produces a directory listing for Bob's identity at the VIS, it produces worksite entries:

— which are available to Bob because of shares with his verified email address;

— with fine-grained `authorization_parameters` and `scopes` fields particular to the worksite;

— where the `auto_create` field should not be true, as Bob should opt-in to the worksite as it can cost him;

— wherein Sally cannot consent to a site being created for Bob.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex D
(normative)

Shared data types

D.1 General

This annex defines a set of types that are shared across the different schemata specified by this document.

D.2 JSON schema types

Types shall be defined as JSON schemata in accordance with https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-01, and make use of the follwoing standard JSON schema names for JSON data types in conformance with IETF RFC 8259:

— *null*

— *boolean*

— *object*

— *array*

— *number*

— *string*

— *integer*

D.3 Simple types

D.3.1 General

The simple type descriptions in Clause D.3 follow and expand on types introduced in ISO 15143-2:2010 Table A.2.

D.3.2 Type "*asset_type*"

A machine type, such as those defined in ISO 6165, ISO 22242, and ISO 9849.

asset_type is of the enumeration:

— `backhoe_loader`

— `compact_tool_carrier`

— `dozer`

— `driller`

— `dumper`

— `excavator`

— `grader`

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

— horizontal_directional_drill

— landfill_compactor

— loader

— milling_machine

— paver

— piler

— pipe_layer

— roller

— scraper

— survey_equipment

— trencher

— other

D.3.3 Type "*character_string_1*"

A string of at most 100 characters.

D.3.4 Type "*character_string_1000*"

A string of at most 1000 characters.

D.3.5 Type "*character_string_2*"

A string of at most 5 characters.

D.3.6 Type "*character_string_255*"

A string of at most 255 characters.

D.3.7 Type "*character_string_3*"

A string of at most 10 characters.

D.3.8 Type "*character_string_4*"

A string of at most 20 characters.

D.3.9 Type "*character_string_5*"

A string of at most 40 characters.

D.3.10 Type "*character_string_6*"

A string of at most 60 characters.

D.3.11 Type "*character_string_b64*"

A base-64 encoded string (encoded using the url and filename safe alphabet in conformance with IETF RFC 4648, not padded), matching the pattern:

```
^[A-Za-z0-9_-]+$
```

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

D.3.12 Type "*character_string_id*"

The type that project data IDs belong to, 255 characters at most, matching the pattern:

```
^[0-9A-Za-z][0-9A-Za-z._~-]*$
```

D.3.13 Type "*character_string_url*"

A URL shall conform to IETF RFC 1738. A VIS shall support HTTP redirects when GETting a URL. A VIS shall provide security tokens when GETting a URL.

D.3.14 Type "*character_string_uuid*"

A universally unique identifier in conformance with IETF RFC 4122, 36 characters at most, matching the pattern:

```
^[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-9A-Fa- f]{12}$
```

D.3.15 Type "*datetime_3339*"

Date and time in UTC, formatted in conformance with IETF RFC 3339 as outlined at https://tools.ietf.org/html/rfc3339#section-5.6 with millisecond accuracy.

D.3.16 Type "*direction_decimal*"

A clockwise direction angle, in degrees between 0 and 360.

D.3.17 Type "*file_entry_mime_type_enum*"

A permissible media type for a project data *file_entry*.

file_entry_mime_type_enum is one of the enumerations:

— `image/vnd.dxf:DXF Background Drawing`

— `application/xml:ISO LandXML`

— `application/pdf:PDF`

D.3.18 Type "*hexargbcolor*"

The color entity indicates the color of an element. Color may be represented as hexadecimal RGB triples, as in HTML, or as hexadecimal ARGB tuples, with the A indicating alpha of transparency. An alpha value of 00 is totally transparent and FF is totally opaque. If RGB is used, the A value is assumed to be FF. For instance, the RGB value "#800080" represents purple. An ARGB value of "#40800080" would be a transparent purple. Colors are defined in terms of the sRGB color space (refer to IEC 61966), matching the pattern:

```
^(#[\dA-F]{6,6}([\dA-F][\dA-F])?)$
```

D.3.19 Type "*interval_decimal*"

An angular spacing, in degrees between 0 and 180.

D.3.20 Type "*latitude_decimal*"

A latitude measured in degrees between -90 (South Pole) and 90 (North Pole).

D.3.21 Type "*longitude_decimal*"

A longitude measured in degrees between -180 (180 degrees West of Greenwich Meridian) and 180 (180 degrees East of Greenwich Meridian).

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

D.3.22 Type "*points_local*"

An array of at most 100 points in local format. Each array element shall be as defined in Table D.1.

Table D.1 — points_local fields

Field	Type	Use	Description
easting	*number*	Required	In meters.
elevation	*number*	Required	In meters.
northing	*number*	Required	In meters

D.3.23 Type "*points_wgs84*"

An array of at most 100 points in WGS-84 format. Each array element shall be as defined in Table D.2.

Table D.2 — points_wgs84 fields

Field	Type	Use	Description
alt	*number*	Required	Ellipsoid height in meters.
lat	*latitude_decimal*	Required	Latitude measured in degrees between -90 (South Pole) and 90 (North Pole).
lon	*longitude_decimal*	Required	Longitude measured in degrees between -180 (180 degrees West of Greenwich Meridian) and 180 (180 degrees East of Greenwich Meridian).

D.3.24 Type "*rotation_decimal*"

A clockwise rotation angle, in degrees between -180 and 180.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex E
(normative)

Project data types

E.1 Object definitions

This annex defines the data types for project data which are communicated between an SMS on one end and a VIS or an SRS on the other, using the API defined in Annex F.

Project data shall be provided as a set of JSON objects. Each such object shall conform to a JSON schema as specified in Annex N. JSON Objects shall be communicated using the UTF-8 encoding in conformance with IETF RFC 3629.

Objects are the smallest unit of project data that can be transferred. Accordingly, they are versioned and considered to be immutable. Any change to the value of an object is affected by issuing a new revision of the object.

Each project data object has attributes that are typed with a simple type or a subtype. Where the values for these attributes require units, those units shall be SI units unless otherwise explicitly stated in this document.

E.2 Definition "*project_metadata*"

Each object shall have a *project_metadata* entry as per Table E.1, which identifies the following required information:

Table E.1 — project_metadata fields

Field	Type	Use	Description
at	*datetime_3339*	Required	Datetime that this object was last edited. Newer revisions of an object should have later at values.
id	*character_string_id*	Required	Unique identifier of the object in this collection of project data. Required to be unique for the project data for a given worksite.
type	*character_string_1*	Required	One of the types from the schema.json file.
v	*character_string_255*	Required	Semantic version of the schema set.

E.3 Type definitions

E.3.1 General

The project data types defined in JSONSchema files available as discussed in Annex N have intended uses detailed in Clause E.3.

Changes to types are managed by the maintenance agency as specified in Annex N. Changes are likely to be infrequent and should follow a ratification process similar to that discussed in ISO 15143-1, amended as necessary. The schemata files contain a "v" property that shall be amended to reflect the semantic version so that implementors can maintain correctness.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

E.3.2 Type "*basic_project_data*"

Each worksite has one-and-only-one object of this type, which contains fields for the data that relate to the project as a whole.

On receiving a *basic_project_data* object, a VIS should pass the object's file package and the files they contain to all assets.

A *basic_project_data* shall be as defined in Table E.2.

Table E.2 — basic_project_data fields

Field	Type	Use	Description
background_image	*character_string_url*	Optional	URL for an image file for the site.
beginning_date	*datetime_3339*	Optional	Anticipated or actual date construction began.
boundary	*points_wgs84*	Optional	Polygon enclosing a site.
completion_date	*datetime_3339*	Optional	Anticipated or actual construction completed.
control_points	*character_string_id*	Optional	Reference to a *file_package* containing the control points for the project.
default_rtk	*character_string_id*	Optional	Default RTK settings.
geo_features	*character_string_id*	Optional	Reference to a *file_package* containing the avoidance zones for the project.
job_code	*character_string_1*	Optional	Job code for the project.
meta	*project_metadata* with id=basic_project_data and type=basic_project_data	Required	Metadata for this *basic_project_data.*
name	*character_string_1*	Required	Human oriented name for *basic_project_data.*
timezone	*character_string_1*	Optional	An IANA timezone string identifying the timezone nominally associated to the project.

E.3.3 Type "*localization*"

E.3.3.1 *General*

Each worksite has zero-or-one object of this type, which contains fields for the data that relate to the localization for the project as a whole.

A *localization* shall be as defined in Table E.3.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.3 — Localization fields

Field	Type	Use	Description
azimuth	*string*, one of(*north_az, south_az*)	Required	The direction of the 0 Azimuth on the worksite.
datum_transformation	*datum_transformation*	Optional	The datum transformation is a parametric datum that converts from WGS-84 positions into the local datum.
grid_orientation	*string*, one of(*NE, NW, SE, SW*)	Required	The direction of the positive coordinate on the worksite.
horizontal_adjustment	*horizontal_adjustment*	Optional	The horizontal adjustment is used to adjust the NEE coordinates that have been produced by the datum or projection. This also allows for local coordinate adjustment on published coordinates system coordinates.
meta	*project_metadata* with id=localization and type=localization	Required	Metadata for this localization.
projection	*projection*	Required	The projection will have one object based on the type of the projection. Field implementations are required to support the most common projections.
vertical_adjustment	*vertical_adjustment*	Optional	The vertical adjustment is used to provide a local height system. The object may contain zero, one or both of "*inclined_plane*" or "*geoid_model*".

E.3.3.2 Definition "*datum_transformation*"

The datum transformation is a parametric datum that converts from WGS-84 positions into the local datum. A *datum_transformation* shall be as defined in Table E.4.

Table E.4 — datum_transformation fields

Field	Type	Use	Description
ellipsoid	*ellipsoid*	Required	Local ellipsoid.
transformation	*datum_3p, datum_7p* or *datum_grid_ref*	Required	Datum transformation method.

E.3.3.3 Definition "*datum_3p*"

3-parameter datum translation.

A *datum_3p* shall be as defined in Table E.5.

Table E.5 — datum_3p fields

Field	Type	Use	Description
dx	*number*	Required	Translation X.
dy	*number*	Required	Translation Y.
dz	*number*	Required	Translation Z.
type	the value *"datum_3p"*	Required	

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

E.3.3.4 Definition "*datum_7p*"

7-parameter datum transformation (translation+rotation+scale).

A *datum_7p* shall be as defined in Table E.6.

Table E.6 — datum_7p fields

Field	Type	Use	Description
dS	*number*	Required	Scale factor (unitless).
dX	*number*	Required	Translation X.
dY	*number*	Required	Translation Y.
dZ	*number*	Required	Translation Z.
rX	*rotation_decimal*	Required	Rotation X.
rY	*rotation_decimal*	Required	Rotation Y.
rZ	*rotation_decimal*	Required	Rotation Z.
type	The value "*datum_7p*"	Required	

E.3.3.5 Definition "*datum_grid_ref*"

Datum grid transformation.

A *datum_grid_ref* shall be as defined in Table E.7.

Table E.7 — datum_grid_ref fields

Field	Type	Use	Description
type	The value "*datum_grid_ref*"	Required	
url	*character_string_url*	Required	URL of a JSON file which contains a *datum_grid* object. The SMS shall ensure that the content of this URL remains unchanged over time.

E.3.3.6 Definition "*ellipsoid*"

An *ellipsoid* shall be as defined in Table E.8.

Table E.8 — ellipsoid fields

Field	Type	Use	Description
inverse_flattening	*number*	Required	Local ellipsoid flattening (inverse).
semi_major_axis	*number*	Required	Local ellipsoid earth radius.

E.3.3.7 Definition "*horizontal_adjustment*"

A *horizontal_adjustment* shall be as defined in Table E.9.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.9 — horizontal_adjustment fields

Field	Type	Use	Description
origin_east	*number*	Required	East origin.
origin_north	*number*	Required	North origin.
rotation_angle	*rotation_decimal*	Required	Rotation about origin.
scale	*number*	Required	Scale factor.
translation_east	*number*	Required	Translation east.
translation_north	*number*	Required	Translation north.

E.3.3.8 Definition "*projection*"

A *projection* shall be as defined in Table E.10.

Table E.10 — projection fields

Field	Type	Use	Description
parameters	*projection_parameters*	Required	Projection parameters. Used parameters vary by projection type.
projection_type	*string, one of(transverse_mercator, mercator, lambert_1, lambert_2, oblique_mercator, oblique_stereographic, albers)*	Required	Projection type.

E.3.3.9 Definition "*projection_parameters*"

Parameters for projection objects.

A *projection_parameters* shall be as defined in Table E.11.

Table E.11 — projection_parameters fields

Field	Type	Use	Description
central_latitude	*latitude_decimal*	Optional	Geodetic latitude of the point from which the values of both the geographical coordinates on the ellipsoid and the grid coordinates on the projection are deemed to increment or decrement for computational purposes.
central_longitude	*longitude_decimal*	Optional	Geodetic longitude of the point from which the values of both the geographical coordinates on the ellipsoid and the grid coordinates on the projection are deemed to increment or decrement for computational purposes.
false_easting	*number*	Optional	Value assigned to the ordinate (north or south) axis of the projection grid at the natural origin.
false_northing	*number*	Optional	Value assigned to the abscissa (east or west) axis of the projection grid at the natural origin.
latitude_no	*latitude_decimal*	Optional	Latitude of the point at which the azimuth of the central line for an oblique projection is defined.
line_azimuth	*direction_decimal*	Optional	Direction (north zero) of the great circle which is the centre line of an oblique projection. The azimuth is given at the projection centre.
line_scale	*number*	Optional	Factor by which the map grid is reduced or enlarged during the projection process, defined by its value at the projection centre.
longitude_no	*longitude_decimal*	Optional	Longitude of the point at which the azimuth of the central line for an oblique projection is defined.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.11 *(continued)*

Field	Type	Use	Description
parallel_1	*latitude_decimal*	Optional	Geodetic latitude of one of the parallels of intersection of the cone with the ellipsoid. Scale is true along this parallel.
parallel_2	*latitude_decimal*	Optional	Geodetic latitude of one of the parallels at which the cone intersects with the ellipsoid. Scale is true along this parallel.
rectified_angle	*rotation_decimal*	Optional	Angle at the natural origin of an oblique projection through which the natural coordinate reference system is rotated to make the projection north axis parallel with true north.
scale	*number*	Optional	Factor by which the map grid is reduced or enlarged during the projection process, defined by its value at the natural origin.

E.3.3.10 Definition "*vertical_adjustment*"

Vertical adjustment defined as an inclined plane and/or geoid,

A *vertical_adjustment* shall be as defined in Table E.12.

Table E.12 — vertical_adjustment fields

Field	Type	Use	Description
geoid_model	*geoid_model*	Optional	Description of the geoid, either in this object data stream or in an external file.
inclined_plane	*inclined_plane*	Optional	Inclined plane.

E.3.3.11 Definition "*geoid_model*"

A *geoid_model* shall be as defined in Table E.13.

Table E.13 — geoid_model fields

Field	Type	Use	Description
type	*string*, one of *(geoid_ref, geoid_model_ref)*	Optional	

A geoid_model shall also contain attributes as defined in exactly one of the following subschemata:

a) geoid_model_ref;

b) geoid_ref.

Subschema geoid_model_ref

A geoid_model_ref shall be as defined in Table E.14.

Table E.14 — geoid_model_ref fields

Field	Type	Use	Description
type	the value *"geoid_model_ref"*	Required	
url	*character_string_url*	Required	URL of a JSON file which contains a *geoid* object. The SMS shall ensure that the content of this URL remains unchanged over time.

Subschema geoid_ref

A geoid_ref shall be as defined in Table E.15.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.15 — geoid_ref fields

Field	Type	Use	Description
ref	*character_string_id*	Required	Reference to a *geoid.*
type	the value *"geoid_ref"*	Required	

E.3.3.12 Definition "*inclined_plane*"

An *inclined_plane* shall be as defined in Table E.16.

Table E.16 — inclined_plane fields

Field	Type	Use	Description
constant_adjustment	*number*	Required	Constant adjustment.
origin_east	*number*	Required	East origin.
origin_north	*number*	Required	North origin.
slope_east	*number*	Required	Scale factor for east slope, unitless.
slope_north	*number*	Required	Scale factor north slope, unitless.

E.3.3.13 Definition "*datum_grid*"

A *datum_grid* shall be as defined in Table E.17.

Table E.17 — datum_grid fields

Field	Type	Use	Description
column_count	*integer*	Required	Number of columns in "separations" matrix.
interpolation_method	*string*, one of (*bilinear, biquadratic, spline*)	Required	Interpolation method. One of "bilinear", "biquadratic", or "spline".
interpolation_window	*number*	Required	Interpolation window length.
latitude_interval	*interval_decimal*	Required	Angular spacing between "separations" matrix rows.
longitude_interval	*interval_decimal*	Required	Angular spacing between "separations" matrix cols.
name	*character_string_255*	Required	Name of datum grid.
origin_latitude	*latitude_decimal*	Required	Latitude origin (first "row" of "separations"). Latitude value of NW (top left).
origin_longitude	*longitude_decimal*	Required	Longitude origin (first "column" of "separations"). Longitude value of NW (top left).
row_count	*integer*	Required	Number of rows in "separations" matrix.
separations_height	array of *numbers*	Optional	Height separation data matrix in row major order.
separations_latitude	array of *numbers*	Required	Latitude separation data matrix in row major order.
separations_longitude	array of *numbers*	Required	Longitude separation data matrix in row major order.

E.3.4 Type "*geoid*"

A *geoid* shall be as defined in Table E.18.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.18 — geoid fields

Field	Type	Use	Description
column_count	*integer*	Required	Number of columns in "separations" matrix.
interpolation_method	*string*, one of (*bilinear, spline*)	Required	Interpolation method. One of either "bilinear" or "spline".
interpolation_window	*number*	Required	Interpolation window length.
latitude_interval	*interval_decimal*	Required	Angular spacing between "separations" matrix rows.
longitude_interval	*interval_decimal*	Required	Angular spacing between "separations" matrix cols.
meta	*project_metadata* with type=geoid	Required	Metadata for this geoid.
name	*character_string_255*	Required	Name of geoid model.
origin_latitude	*latitude_decimal*	Required	Latitude origin (first "row" of "separations"). Latitude value of NW (top left).
origin_longitude	*longitude_decimal*	Required	Longitude origin (first "column" of "separations"). Longitude value of NW (top left).
row_count	*integer*	Required	Number of rows in "separations" matrix.
separations	array of *numbers*	Required	Geoid separation data matrix in row major order.

E.3.5 Type "*rtk*"

E.3.5.1 General

Settings required to obtain correction data for localizations.

Each project may provide a default object of this type, which contains fields for the data that relate to the RTK and correction settings for the project as a whole. This default object is provided in the *basic_project_data* object.

Each project may provide additional RTK objects in the case of multiple base stations; in this case, each *work_order* may specify the particular settings to use.

A *rtk* shall be as defined in Table E.19.

Table E.19 — rtk fields

Field	Type	Use	Description
latitude	*latitude_decimal*	Optional	Geodetic latitude of the RTK.
longitude	*longitude_decimal*	Optional	Geodetic longitude of the RTK.
meta	*project_metadata* with type=*rtk*	Required	Metadata for this *rtk.*
name	*character_string_1000*	Optional	A name used to refer to the RTK.
ntrip	*ntrip_settings*	Optional	The settings for RTK where provided by an NTRIP base server.
uhf	*uhf_settings*	Optional	The settings for RTK where provided by a narrow band UHF radio.

E.3.5.2 Definition "*ntrip_settings*"

The settings for RTK where provided by an NTRIP base server A *ntrip_settings* shall be as defined in Table E.20.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.20 — ntrip_settings fields

Field	Type	Use	Description
mount_point	*character_string_1*	Required	Base station and mount point name.
password	*character_string_1*	Optional	
server_host	*character_string_1*	Required	IP address or DNS name of the server.
server_port	*integer*	Optional	Port number of the server for non-TLS connections.
tls_port	*integer*	Optional	port number of the server for TLS connections.
username	*character_string_1*	Optional	

If an SMS provides *ntrip_settings*, it shall provide at least one of *server_port* and *tls_port*. It should also provide a *tls_port*.

E.3.5.3 Definition "*uhf_settings*"

This object provides the settings required to access a narrow band UHF radio. Messages shall be sent using the protocol in conformance with RTCM 13500.1.

An *uhf_settings* shall be as defined in Table E.21.

Table E.21 — uhf_settings fields

Field	Type	Use	Description
center_frequency	*integer*	Required	The center frequency for the base station transmission, in Hz.

E.3.6 Type "*asset*"

A piece of equipment used on a worksite.

For assets capable of performing multiple types of work, such as a machine equipped with both a drill and an excavator bucket, a separate *asset* object shall be created for each type.

An *asset* shall be as defined in Table E.22.

Table E.22 — asset fields

Field	Type	Use	Description
asset_class	*asset_type*	Required	The type of the asset.
equipment_id	*character_string_255*	Optional	A user-friendly string identifying an asset such as a machine.
id	*character_string_uuid*	Required	The VIS asset ID, an identifier determined by the VIS for this asset, generated in conformance with IETF RFC 4122. On any worksite, no two assets shall have the same VIS asset id. A VIS should make this unique per asset per worksite.
meta	*project_metadata* with type=asset	Required	Metadata for this asset.
oem_iso_id	*character_string_255*	Optional	This is either a PIN or VIN. The pin shall be as defined by ISO 10261. The VIN shall be as defined by ISO 3779.
serial_number	*character_string_255*	Optional	This identifies the specific instance of the asset for those assets that are not identified by a PIN or a VIN.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

E.3.7 Type "*codelist*"

E.3.7.1 General

Code lists enhance the usability and robustness of design model and as-built data.

Code lists can be used for traditional survey use, where features of the built environment are identified by using human-readable names of features. In addition to survey use, code lists can be used on the file import phase of ISO LandXML design files. This should ensure human-readable feature names and harmonize the visualization of categories and features across all files of the worksite.

High level elements within code list (or taxonomy):

— category code — defines group where features may belong;

— feature code — defines human readable name;

— visualization including color and style — harmonizes the visualization across many files.

The SMS, VIS and field equipment should use codelists together with design model to ensure that:

— all items on the worksite (design models, as-built objects, production snapshots, etc.) share common visualization scheme (colors, linestyle, linewidth);

— alphanumeric feature codes of design objects are presented using a human-readable description.

Additional information is available in Annex M.

Codelist is a list of codes, codegroups and attributes. A *codelist* shall be as defined in Table E.23.

Table E.23 — codelist fields

Field	Type	Use	Description
attributes	Array of at most 1000 *codelist_attributes*	Optional	Array of attributes.
codegroups	Array of at most 100 *codelist_codegroups*	Optional	Array of codegroups.
codes	Array of at most 100000 *codelist_codes*	Optional	Array of codes.
creator	*character_string_255*	Optional	Creator of the codelist.
description	*character_string_255*	Optional	Description of the codelist.
meta	*Project_metadata* with type=codelist	Required	Metadata for this codelist.
name	*character_string_255*	Required	Human readable codelist name.
timestamp	*datetime_3339*	Optional	Creation and modification time of the codelist.
version	*character_string_255*	Optional	Version of the codelist.

E.3.7.2 Definition "*codelist_attribute*"

Freeform attribute.

A *codelist_attribute* shall be as defined in Table E.24.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.24 — codelist_attribute fields

Field	Type	Use	Description
id	*character_string_id*	Required	Unique ID of attribute.
name	*character_string_255*	Required	Name of attribute.
type	*attribute_type*	Required	Attribute type.
unit	*character_string_5*	Optional	Attribute unit as defined by ISO 80000-1, containing optional unit prefix and unit symbol (ie. m,kg...).

E.3.7.3 Definition "*codelist_code*"

Feature code.

A *codelist_code* shall be as defined in Table E.25.

Table E.25 — codelist_code fields

Field	Type	Use	Description
attribute_refs	Array of at most 100 *attribute_refs*	Optional	List of attribute references linked to this code.
codegroups	Array of at most 100 *character_string_ids*	Optional	List of codegroup IDs linked to this code.
fulldesc	*character_string_255*	Optional	Full description of code.
name	*character_string_4*	Required	This is the code.
shortcut	*character_string_2*	Optional	To enhance UX, short "quick dial sequence" for most commonly used codes.
shortdesc	*character_string_1*	Required	Short description of code for field equipment with limited screen size.

A *codelist_code* shall also contain attributes as defined in exactly one of the following subschemata:

a) *codelist_code_area*

b) *codelist_code_category*

c) *codelist_code_line*

d) *codelist_code_point*

e) *codelist_code_surface*

Subschema *codelist_code_area*

Code type for codes defining a geographical area.

A *codelist_code_area* shall be as defined in Table E.26.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.26 — codelist_code_area fields

Field	Type	Use	Description
areacolor	*hexargbcolor*	Optional	Area color.
areastyle	*codelist_areastyles*	Optional	Area fill style.
linecolor	*hexargbcolor*	Optional	Line color.
linestyle	*codelist_linestyles*	Optional	Line style.
linewidth	*number*	Optional	Line width.
pointcolor	*hexargbcolor*	Optional	Point color.
pointstyle	*codelist_pointstyles*	Optional	Point style.
type	The value "*area*"	Required	

Subschema *codelist_code_category*

Code type for category codes defining a geographical feature.

A *codelist_code_category* shall be as defined in Table E.27.

Table E.27 — codelist_code_category fields

Field	Type	Use	Description
areacolor	*hexargbcolor*	Optional	Area color.
areastyle	*codelist_areastyles*	Optional	Area fill style.
linecolor	*hexargbcolor*	Optional	Line color.
linestyle	*codelist_linestyles*	Optional	Line style.
linewidth	*number*	Optional	Line width.
pointcolor	*hexargbcolor*	Optional	Point color.
pointstyle	*codelist_pointstyles*	Optional	Point style.
surfacecolor	*hexargbcolor*	Optional	Surface color.
type	the value "*category*"	Required	

Subschema *codelist_code_line*

Code type for codes defining a line shaped feature.

A *codelist_code_line* shall be as defined in Table E.28.

Table E.28 — codelist_code_line fields

Field	Type	Use	Description
linecolor	*hexargbcolor*	Optional	Line color.
linestyle	*codelist_linestyles*	Optional	Line style.
linewidth	*number*	Optional	Line width.
pointcolor	*hexargbcolor*	Optional	Point color.
pointstyle	*codelist_pointstyles*	Optional	Point style.
type	the value "*line*"	Required	

Subschema *codelist_code_point*

Code type for codes defining a point.

A *codelist_code_point* shall be as defined in Table E.29.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.29 — codelist_code_point fields

Field	Type	Use	Description
pointcolor	*hexargbcolor*	Optional	Point color.
pointstyle	*codelist_pointstyles*	Optional	Point style.
type	the value "*point*"	Required	

Subschema *codelist_code_surface*

Code type for codes defining a surface.

A *codelist_code_surface* shall be as defined in Table E.30.

Table E.30 — codelist_code_surface fields

Field	Type	Use	Description
surfacecolor	*hexargbcolor*	Optional	Surface color.
type	the value "*surface*"	Required	

E.3.7.4 Definition "*codelist_codegroup*"

Group of codes.

A *codelist_codegroup* shall be as defined in Table E.31.

Table E.31 — codelist_codegroup fields.

Field	Type	Use	Description
deactivated_for_measurements	*boolean*	Optional	If true, the codes belonging to this group shall be used only for model data import and should not be available for mapping on survey equipment or grade control system.
id	*character_string_id*	Required	Unique ID of code group.
name	*character_string_255*	Required	Human readable name of code group.

E.3.7.5 Definition "*codelist_linestyles*"

The *codelist_linestyles* definition is of the enumeration:

— `solid`

— `dashed`

— `dotted`

— `dashdot`

— `dashdotdot`

E.3.7.6 Definition "*codelist_pointstyles*"

The *codelist_pointstyles* definition is of the enumeration:

— `dot`

— `triangle`

— `rectangle`

— `diamond`

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

— star

— circle

— filled_triangle

— filled_rectangle

— filled_diamond

— filled_star

— filled_circle

— cross

— plus

— tree

— pin

— rectangle_dot

— rectangle_plus

— rectangle_cross

— rectangle_pin

— rectangle_circle

— rectangle_circle_dot

— rectangle_circle_plus

— diamond_cross

— circle_dot

— circle_plus

— circle_cross

— circle_pin

— circle_circle

— circle_filled_circle

E.3.7.7 Definition "*codelist_areastyles*"

The *codelist_areastyles* definition is of the enumeration:

— solid

— grid

— checkered

— horz_lines

— vert_lines

— dots

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

E.3.7.8 Definition "*attribute_type*"

The *attribute_type* definition is of the enumeration:

— `integer` : json integer type

— `number` : json number type

— `text` : json text type

— `datetime_3339` : json string type as datetime_3339.

E.3.7.9 Definition "*range*"

A range of permitted values. An extra constraint is that min < max. *range* can be applied only for integer or number attribute types. The range is inclusive of minimum and maximum values.

A *range* shall be defined as in Table E.32.

Table E.32 — range fields

Field	Type	Use	Description
max	*number*	Required	Maximum value.
min	*number*	Required	Minimum value.

E.3.7.10 Definition "*attribute_ref*"

Attribute reference links an *attribute* to feature or category code.

An *attribute_ref* shall be as defined in Table E.33.

Table E.33 — attribute_ref fields

Field	Type	Use	Description
attribute	*character_string_id*	Required	The ID of an attribute.
choices	array of between 2 and 100 *integers*, array of between 2 and 100 *numbers*, array of between 2 and 100 *character_string_255s*, or array of between 2 and 100 *datetime_3339s*	Optional	Choice list of values. No choices means all values are allowed.
default	*integer, number, character_string_255*, or *datetime_3339*	Optional	Default value.
optionality	*string*, one of(*optional, required, constant*)	Optional	Attribute optionality.
range	*range*	Optional	A range of permitted values. Range can be applied only for integer or number attribute types.

E.3.8 Type "*file_entry*"

In order to facilitate discovery and transference of large objects, the project data provides a type to provide metadata for file objects.

Where such files are logically grouped into a single item, such metadata is presented as a *file_package* as detailed in Subclause E.3.9.

Downloading the file contents is described in Clause F.6.

A *file_entry* shall be as defined in Table E.34.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.34 — file_entry fields

Field	Type	Use	Description
checksum	*string* matching the regular expression `^[0-9a-f]{64}$`	Required	SHA-256 hash (in accordance with ISO/IEC 10118-3) of file content. This shall be the checksum obtained from the file downloaded from the download_url.
created_at	*datetime_3339*	Required	Identifies when the file was created.
description	*character_string_1000*	Optional	Text describing the file contents and intended usage.
download_url	*character_string_url*	Required	Location from where the content of the file can be obtained.
meta	*Project_metadata* with type=file_entry	Required	Metadata for this *file_entry.*
mime_type	*file_entry_mime_type_enum*	Required	Media type of the file content.
name	*character_string_255*	Required	Name for file, meaningful to end-users. Names are informational, are not intended to be represented in a file system and are not required to be unique.
size	*integer*	Required	File size in bytes. This shall be the size of the file downloaded from the *download_url.*
version	*character_string_255*	Optional	Human-readable string giving the version of the file.

E.3.9 Type “file_package”

The *file_package* type provides a mechanism to aggregate the file metadata described in Subclause E.3.8.

This type is provided in order to facilitate presenting such metadata as a set of files, as it is considered that this reflects usual industry practice that files are supplied in sets to be loaded on an asset. As a consequence, *file_package*s should be small.

File packages can contain files in formats dictated by Annex L.

A *file_package* shall be as defined in Table E.35.

Table E.35 — file_package fields

Field	Type	Use	Description
files	array of at most 1000 *character_string_ids*	Required	References to the *file_entries* contained in the file package.
meta	*project_metadata* with type=*file_package*	Required	Metadata for this *file_package.*
name	*character_string_1*	Required	Human-oriented name for the *file_package.*
revision_note	*character_string_1000*	Optional	Descriptive textual information suitable for display to users.

For each element of the files array of the fileset, the SMS shall provide a *file_entry* as detailed in the preceding section.

E.3.10 Type “*material*”

A definition for a material used on site for any purpose, enabling it to be identified in a unique and consistent way throughout the project. This definition neccessitates further extension in future parts.

A *material* shall be as defined in Table E.36.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.36 — material fields

Field	Type	Use	Description
meta	*project_metadata* with type=material	Required	Metadata for this material.
name	*character_string_1*	Required	Human-oriented name for the material.

E.3.11 Type "*operator*"

A definition for any person who must be identified in the project for any purpose, enabling them to be identified in a unique and consistent way throughout the project. This definition necessitates further extension in future parts.

An *operator* shall be as defined in Table E.37.

Table E.37 — operator fields

Field	Type	Use	Description
meta	*Project_metadata* with type=operator	Required	Metadata for this operator.
name	*character_string_1*	Optional	Text used by the VIS to refer to the operator, whether by name or by some ID or by other means.

E.3.12 Type "*work_order*"

A work order refers to a collection of fields that are relevant to performing a single piece of work.

Several fields from the mission data are as defined in ISO 15143-2:2010 Table A.2. These fields have names that end with _1.

On receiving a *work_order*, a VIS should pass the work order's file packages and the files they contain to assets it intends to use to complete the work.

A VIS shall report to SMS how it has processed the work orders it has received by sending *work_order_confirmations* using the route specified in Clause F.16. If an SMS wishes to receive a new confirmation message, it may publish a new revision of the work order by increasing its timestamp. This can be useful to confirm that updated files within the work order's file packages have been transmitted to assets.

It is expected that this type can necessitate further extension in future parts.

A *work_order* shall be as defined in Table E.38.

Table E.38 — work_order fields

Field	Type	Use	Description
codelist	*character_string_id*	Optional	Reference to codelist used by the work order.
description	*character_string_1000*	Required	Description of the work to be performed.
file_packages	array of *character_string_ids*	Optional	References to *file_packages*.
foreman_name_1	*character_string_5*	Optional	Name of the person employed by a contractor to be in charge of site labour.
meta	*project_metadata* with type=*work_order*	Required	Metadata for this *work_order.*
name	*character_string_1*	Required	Human-oriented name for the *work_order.*
rtk	*character_string_id*	Optional	RTK settings to be used for this work order, if not the default settings.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.38 *(continued)*

Field	Type	Use	Description
work_area_boundary	*points_local*	Optional	Range of operation and space which organizes the range of construction by the asset.
work_area_name_1	*character_string_3*	Optional	Unique name of operation range where the range corresponds to a work order.
work_time_start_1	*datetime_3339*	Optional	Date and time in UTC of work starting provided for the work; starting time of operation allocated to the work.
work_time_end_1	*datetime_3339*	Optional	Date and time in UTC of finished work provided for the work; ending time of operation allocated to the work.

E.3.13 Type "*assignment*"

E.3.13.1 General

An assignment specifies either the set of assets or operators, or both, which should execute a work order. An *assignment* shall be as defined in Table E.39.

Table E.39 — assignment fields

Field	Type	Use	Description
assignees	array of at most 100 *assignees*	Required	Array of objects referring to operators, assets or asset types. An entity is included in the assignment if one or more of the assignees accepts it.
meta	*project_metadata* with type=assignment	Required	Metadata for this assignment.
work_order	*character_string_id*	Required	Reference to a *work_order.*

E.3.13.2 Definition "*assignee*"

An *assignee* shall be as defined in Table E.40.

Table E.40 — assignee fields

Field	Type	Use	Description
type	string, one of (*asset_type*, *asset*, *operator*, *all_assets*)	Required	The rule type used to accept entities for assignments.

An assignee shall also contain attributes as defined in exactly one of the following subschemata:

a) *operator_ref*

b) *asset_ref*

c) *asset_type_ref*

d) *all_assets*

Subschema operator_ref

Accepts the stated operator.

An operator_ref shall be as defined in Table E.41.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table E.41 — operator_ref fields

Field	Type	Use	Description
operator	*character_string_id*	Required	The ID of an operator.
type	the value "*operator*"	Required	

Subschema asset_ref

Accepts the stated asset.

An asset_ref shall be as defined in Table E.42.

Table E.42 — asset_ref fields

Field	Type	Use	Description
asset	*character_string_id*	Required	The ID of an asset.
type	the value "*asset*"	Required	

Subschema asset_type_ref

Accepts all assets of the stated asset_type.

An *asset_type_ref* shall be as defined in Table E.43.

Table E.43 — asset_type_ref fields

Field	Type	Use	Description
asset_type	*asset_type*	Required	An asset type name.
type	the value "*asset_type*"	Required	

Subschema all_assets

Accepts all assets.

An *all_assets* shall be as defined in Table E.44.

Table E.44 — all_assets fields.

Field	Type	Use	Description
type	the value "*all_assets*"	Required	

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex F
(normative)

Project data SMS API

F.1 General

The SMS is responsible for maintaining project data and providing project data to VISs and SRSs through the following REST APIs. Each route in each API is relative to the `worksiteBaseURL` as discussed in Subclause 5.3.

Objects are communicated as JSON objects. Each object has a type, which is defined by a JSON Schema from Clause E.3. Objects shall validate against the JSONSchema.

For communication purposes, all objects contain provenance data elements described in Clause E.2. These elements are communicated inside a JSON frame with the key “meta”. Objects communicated by a VIS are sent to the SMS without this metadata; the required data is supplied by the SMS.

F.2 Authorizing

These endpoints shall be accessed with an access token obtained using OAuth2.0 Client Credentials Grant, as detailed in Subclause A.2.4.3. The SMS shall require the scope `iso15143-4.project_data.read` or `iso15143-4.project_data.write` for endpoints which read and write data respectively.

F.3 Project data synchronization

F.3.1 General

To facilitate efficient data synchronization, the SMS shall provide a route as discussed in F.3 and detailed in Clause F.8. This route provides a stream of events. Differences in access policies will result in different event streams which can contain different subsets of the project data.

For this route, the SMS shall communicate a cursor value specific to the VIS and access policy accessing the project data. This cursor value shall be used as part of the data transport rather than an attribute of the project data objects themselves. The SMS shall ensure that cursor values are strictly increasing such that when a VIS is up to date at a particular cursor value it can request changes from that value and will receive all further data communicated without receiving data already known to it.

The SMS shall ensure referential integrity at each cursor value, even if the SMS applies an access policy to restrict the objects seen by a VIS. Every reference to an object shall be to an object available at that cursor value.

Cursor values shall be communicated as strings. This document does not specify format for them other than the requirement that they be at most 64 characters long and consist only of alphanumeric characters, underscores, minus signs, or slashes. A blank string shall constitute a legal cursor value, and is interpreted as a request for all current values regardless of cursor. A cursor value requested by a VIS shall be either a blank string or a value which has been previously communicated by the SMS.

An SMS may choose to provide every revision of every project data object through this mechanism, but it is only required to provide revisions that have not been superseded by later revisions.

F.3.2 Initialization

The VIS can initiate and maintain project data synchronization with the SMS by issuing a series of GET requests to the route

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

```
GET {worksiteBaseURL}/2024/project_data_events
```

as described in Subclause F.8.1.1. This route may be given as one of the following:

— a cursor query parameter to request data from a particular point in the data synchronization sequence;

— a `max_items` query parameter to limit the number of items returned;

— a `long_poll` query parameter to limit the wait time.

If items are available from the given cursor, the SMS shall return them without waiting. Otherwise, the SMS shall wait up to `long_poll` seconds for new items to become available. The SMS shall return the new items immediately. If no new items become available during the long poll, the SMS shall return an empty array.

The SMS shall make every attempt to honour the request parameters.

F.3.3 Processing

The route described in F.3.2 returns an array of JSON objects.

Each such object has two fields:

— a `t` field, a string which expresses the event type, and is either `delete` or `upsert`

— a `d` field, an object which expresses the event data.

If the `t` value is `delete`, the `d` field identifies the object to be deleted, as per F.9.3. The VIS shall make no further reference to this object.

If the `t` value is `upsert`, the `d` field provides the data object to be inserted, formatted as one of the types in Clause E.3.

F.4 Obtaining objects for reporting

An SMS shall make project data available to an SRS by providing type-scoped GET endpoints as specified in Clause F.12. These routes provide a paginated view of the current data for each project data type. Each such route accepts a *limit* parameter that defines the number of objects returned per page.

Each call returns:

— a page of objects;

— a next link to obtain the next page of objects, if there is one.

The exact detail of the next link is the responsibility of the SMS and the SRS shall treat it as opaque. That is, it shall not rely on it being structured in any particular way. The SRS is expected to page through all available objects in one session and not retain `next` links once used. The SMS shall make every effort to return all objects current at the time of call.

These routes shall provide only the most recent revision of any object.

F.5 Upserting objects

F.5.1 Upserting by the SMS

The SMS is responsible for maintaining project data and uses its own methods to create and amend it, outside the scope of this document.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.5.2 Upserting by a VIS

The VIS is expected to contribute files using the API described in Clause F.7. In addition, a VIS can provide and manage data for the following project data types:

— assets;

— operators.

For each of these types, an SMS shall provide (as specified in Clause F.12):

a) a POST route to enable a VIS to insert objects of that type;

b) a PUT route to enable a VIS to update objects that it contributed;

c) a DELETE route to enable a VIS to delete objects that it contributed.

This mechanism facilitates both insertion and updating, as the whole object is supplied on each update. In particular, this document does not require the use of the HTTP PATCH verb.

Authorization is achieved via the policy documents as per Clause 5.

The SMS shall track the VIS which owns (inserted) each object. The SMS shall reject updates from a VIS where the object is owned by another VIS.

Before referencing an asset or operator in any as-built and production data messages sent using the APIs discussed in Annex I or Annex J, a VIS shall POST these details and ensure that the SMS returns a 201 status from the POST. A VIS shall make every effort to avoid POSTing the same asset more than once.

F.5.3 Revisioning

Since some project data types contain references to objects of other types, the question arises as to what happens when a new revision of a referred object is created. To this end, the SMS can follow a defined strategy to track changes to referred objects. For example:

— The SMS will possibly not perform any additional actions.

— The SMS will possibly release new revisions of referring objects by increasing the `at` timestamp.

— The SMS will possibly avoid updating existing objects where possible, and instead, for each new object revision, delete the old object and create a whole new object. In this case, the SMS must similarly recreate any objects referring to the deleted object.

F.6 Downloading files

A VIS may download files from URLs included in *file_entry* objects.

The SMS shall ensure that the `download_url` returns the required content for any *file_entry* that has not been deleted or superseded. An SMS is not required to maintain file content for *file_entry* objects that have been superseded. The SMS shall ensure that the content of each `download_url` remains unchanged over time. That is, if the file contents change, an SMS shall publish a new revision of the *file_entry* object with a different `download_url`.

The SMS shall authorize file downloads using the same mechanism that is used for other project data API calls. Thus, downloads shall be proxied by the VIS.

The VIS may cache downloads.

An SRS may download files in the same way as a VIS does.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.7 Uploading files

F.7.1 General

A VIS may upload files to the SMS, such as survey files in the format defined in Annex L. To facilitate uploading files, the SMS shall provide routes as detailed in Clause F.10. These routes allow for uploading a file in several separate parts, which are to be catenated together to reconstitute the entire file. Where a file is sufficiently small, the VIS shall upload a single part.

An SRS shall not upload files.

F.7.2 Initialization

A VIS shall initiate a multipart file upload by sending a POST request to the route documented in Subclause F.10.1.1. On success, the SMS shall return an `upload_id` to identify the particular upload.

F.7.3 Content uploading

The VIS shall then upload the various parts of the file by sending a PUT request to the route documented in Subclause F.10.1.4 by ID. The body of this request shall be the content to be uploaded.

Each part is identified by a part number, which is an integer between 1 and the number of parts. The VIS shall supply the part number in the PUT request as a query parameter. The VIS shall upload a part for each number. The SMS shall ensure that each part is PUT by the same VIS.

No part shall exceed 1 Gb in size. Other than the final part, no part shall be less than 200 Mb in size.

F.7.4 Completion

Once all parts have been uploaded, the VIS shall signal that the upload is complete by sending a POST request to the route documented in Subclause F.10.1.3.

On receiving this POST request, the SMS shall:

— check that all parts have been uploaded correctly;

— check that no part is missing, i.e. that there is a part for each integer between 1 and the number of parts;

— assemble the uploaded parts in part-number order;

— confirm that the checksum of the assembled file agrees with the checksum provided as part of the POST body;

— emit a *`file_entry`* object detailing the file and providing a download link for it.

F.7.5 Cancellation

If it is necessary, a VIS can cancel an upload in progress by sending a DELETE request to the route documented in Subclause F.10.1.2.

In addition, an SMS may terminate a file upload if the process described in F.7 is not completed within 1 hour.

F.8 Event routes

F.8.1 project_data_events routes

F.8.1.1 GET {worksiteBaseURL}/2024/project_data_events

Description

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Provide changes to project data. The API supports long polling where, should data not be available immediately, the SMS shall wait (within the long poll timeout) to gather messages before responding. The mechanism is described in Clause F.3.

Parameters:

This route shall support the parameters described in Table F.1.

Table F.1 — GET project_data_events parameters

Name	Source	Type	Use	Description
cursor	query	*String* matching the regular expression `^[a-zA-Z0-9_/-]{0,64}$`	Optional	The cursor from when to return project data. The cursor should be empty, or a value returned by the SMS in response to a previous call.
max_items	query	*integer* between 1 and 1000	Optional	The maximum number of items to return, with valid values between 1 and 1000, defaults to 1000.
long_poll	query	*integer* between 0 and 60	Optional	Defines maximum interval (in seconds) to wait before returning data. If zero or missing, all available data (up to max_items) is returned immediately.

Responses:

This route shall provide responses as described in Table F.2.

Table F.2 — GET project_data_events responses

Code	Schema	Description
200	*project_data_events_get_body*	Successful operation.
400	N/A	Bad parameter data.
401	N/A	Authentication failure.
404	N/A	Not found.
409	N/A	Conflict: the project data state has changed in ways that cannot be easily resolved. A VIS should discard its current state and request a feed with no cursor.
410	N/A	Gone: the worksite is no longer available. A VIS should not issue further requests.

F.9 Event definitions

F.9.1 project_data_event

A *project_data_event* object shall have contents as defined exactly in one of the types:

— *project_data_event_delete*

— *project_data_event_upsert*

F.9.2 project_data_event_delete

A notice instructing a VIS to delete a project data element.

Properties:

A *project_data_event_delete* shall have the properties defined in Table F.3.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.3 — project_data_event_delete properties

Name	Type	Use	Description
d	*project_data_event_delete_data*	Required	Data identifying the object to delete.
t	the value "*delete*"	Required	Type flag identifying this object as a deletion.

F.9.3 project_data_event_delete_data

Properties:

A *project_data_event_delete_data* shall have the properties defined in Table F.4.

Table F.4 — project_data_event_delete_data properties

Name	Type	Use	Description
id	*character_string_id*	Required	ID of the project data element to delete.
type	string, one of(*asset, assignment, codelist, file_entry, file_package, geoid, localization, material, operator, basic_project_data, rtk, work_order*)	Required	Type of the project data element to delete.

F.9.4 project_data_event_upsert

A notice instructing a VIS to upsert a project data element.

Properties:

A *project_data_event_upsert* shall have the properties defined in Table F.5.

Table F.5 — project_data_event_upsert properties

Name	Type	Use	Description
d	*project_data_event_upsert_data*	Required	Data identifying the object to upsert.
t	the value "*upsert*"	Required	Type flag identifying this object as an upsert.

F.9.5 project_data_event_upsert_data

A *project_data_event_upsert_data* object shall have contents as defined in exactly one of the types:

— *asset*

— *assignment*

— *codelist*

— *file_entry*

— *file_package*

— *geoid*

— *localization*

— *material*

— *operator*

— *basic_project_data*

— *rtk*

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

— *work_order*

F.9.6 project_data_events_get_body

An array of objects representing deletes or upserts of project data elements.

Properties:

A *project_data_events_get_body* shall have the properties defined in Table F.6.

Table F.6 — project_data_events_get_body properties

Name	Type	Use	Description
items	array of *project_data_events*	Required	An array of project_data_events.
next_cursor	*string* matching the regular expression `^[a-zA-Z0-9_/-]{0,64}$`	Required	An opaque cursor, at most 64 characters made out of the regular expression `^[a-zA-Z0-9_/-]{0,64}$`

F.10 File routes

F.10.1 file_uploads routes

F.10.1.1 POST {worksiteBaseURL}/2024/file_uploads

Description

Initiate a multi-part file upload.

Responses:

This route shall support the parameters described in Table F.7.

Table F.7 — POST file_uploads responses

Code	Schema	Description
200	*file_upload_response*	Successful operation, the SMS has the data and will not lose it.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.10.1.2 DELETE {worksiteBaseURL}/2024/file_uploads/{id}

Description

Cancel a multi-part file upload. Any uploaded parts will be discarded.

Parameters

This route shall support the parameters described in Table F.8.

Table F.8 — DELETE file_uploads/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_1*	Required	ID to identify the file, provided from the SMS.

Responses

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

This route shall provide responses as described in Table F.9.

Table F.9 — DELETE file_uploads/{id} responses

Code	Schema	Description
204	N/A	File has been deleted successfully.
400	N/A	Invalid ID supplied.
404	N/A	No such upload has been initiated.

F.10.1.3 POST {worksiteBaseURL}/2024/file_uploads/{id}

Description

Complete a multi-part file upload. The uploaded parts will be assembled into a complete file.

Parameters

This route shall support the parameters described in Table F.10.

Table F.10 — POST /file_uploads/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_1*	Required	ID to identify the file, as provided from the SMS.
number_of_parts	query	*integer* between 1 and 10000	Required	Total number of parts uploaded.
body	body	*file_upload_complete_request*	Required	Fields used to create a file entry.

Responses

This route shall provide responses as described in Table F.11.

Table F.11 — POST /file_uploads/{id} responses

Code	Schema	Description
200	*file_upload_complete_response*	Successful operation, the SMS has the data and will not lose it.
400	N/A	Invalid ID supplied, checksum does not match, or not all parts have been uploaded.
404	N/A	No such upload has been initiated or the SMS has deleted the upload due to it taking too long.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.10.1.4 PUT {worksiteBaseURL}/2024/file_uploads/{id}

Description

Upload a single part of a multi-part file upload.

Parameters

This route shall support the parameters described in Table F.12.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.12 — PUT /file_uploads/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_1*	Required	ID to identify the file, as provided from the SMS.
part_number	query	*integer* between 1 and 10000	Required	Part number of the part being uploaded.
body	body	*binary*	Required	Content for the part being uploaded.

Responses

This route shall provide responses as described in Table F.13.

Table F.13 — PUT/file_uploads/{id} responses

Code	Schema	Description
204	N/A	Successful operation, the SMS has the data and will not lose it.
400	N/A	Supplied object cannot be parsed.
401	N/A	VIS will possibly not upload this file because it is not authenticated correctly, is not permitted by policy, or the VIS is attempting to upload to a file initiated by another VIS.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.
413	N/A	Supplied part is too large.

F.11 File definitions

F.11.1 file_upload_complete_request

Fields provided when a file upload is completed.

Properties:

A *file_upload_complete_request* shall have the properties defined in Table F.14.

Table F.14 — file_upload_complete_request properties

Name	Type	Use	Description
checksum	*string* of at most 64 characters	Required	SHA-256 (in accordance ISO/IEC 10118-3:2018) hash of file content.
created_at	*datetime_3339*	Required	Identifies when the file was created.
description	*character_string_1000*	Optional	Text describing the file contents and intended usage.
mime_type	*file_entry_mime_type_enum*	Required	Media type for the file content.
name	*character_string_255*	Required	Name for file, meaningful to end-users. Names are informational, are not intended to be represented in a file system, and are not required to be unique.
size	*integer*	Required	Total file size, in bytes.

F.11.2 file_upload_complete_response

The body returned when a multipart file upload is completed.

Properties:

A *file_upload_complete_response* shall have the properties defined in Table F.15.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.15 — file_upload_complete_response properties

Name	Type	Use	Description
file_entry_id	*character_string_id*	Required	ID for file entry for the uploaded file

F.11.3 file_upload_response

The body returned when a multipart file upload is initiated.

Properties:

A *file_upload_response* shall have the properties defined in Table F.16.

Table F.16 — file_upload_response properties

Name	Type	Use	Description
upload_id	*character_string_1000*	Required	Transient ID for file upload

F.12 REST query routes

F.12.1 General

This API provides a REST API for an SRS to retrieve data from an SMS. The SMS shall implement all routes defined here if it chooses to support SRSs.

F.12.2 Assets routes

F.12.2.1 GET {worksiteBaseURL}/2024/types/assets

Description

Returns a paginated view of all current asset data for a worksite.

Parameters

This route shall support the parameters described in Table F.17.

Table F.17 — GET types/assets parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.18.

Table F.18 — GET types/assets responses

Code	Schema	Description
200	*list_of_assets*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.12.2.2 GET {worksiteBaseURL}/2024/types/assets/{id}

Description

Returns a single asset object for a worksite.

Parameters

This route shall support the parameters described in Table F.19.

Table F.19 — GET types/assets/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the asset.

Responses

This route shall provide responses as described in Table F.20.

Table F.20 — GET types/assets/{id} responses

Code	Schema	Description
200	*asset*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.3 Assignments routes

F.12.3.1 GET {worksiteBaseURL}/2024/types/assignments

Description

Returns a paginated view of all current assignment data for a worksite.

Parameters

This route shall support the parameters described in Table F.21.

Table F.21 — GET types/assignments parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.22.

Table F.22 — GET types/assignments responses

Code	Schema	Description
200	*list_of_assignments*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.12.3.2 GET {worksiteBaseURL}/2024/types/assignments/{id}

Description

Returns a single assignment object for a worksite.

Parameters

This route shall support the parameters described in Table F.23.

Table F.23 — GET types/assignments/{id} parameters

Name	Source	Type	Use	Description
id	Path	*character_string_id*	Required	ID of the assignment.

Responses

This route shall provide responses as described in Table F.24.

Table F.24 — GET types/assignments/{id} responses

Code	Schema	Description
200	*assignment*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.4 basic_project_data routes

F.12.4.1 GET {worksiteBaseURL}/2024/types/basic_project_data

Description

Returns a paginated view of all current *basic_project_data* data for a worksite. This view will always return a single object as each worksite has a single *basic_project_data* object. The view is included in this document for ease of consistent processing of objects.

Parameters

This route shall support the parameters described in Table F.25.

Table F.25 — GET types/basic_project_data parameters

Name	Source	Type	Use	Description
limit	Query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.26.

Table F.26 — GET types/basic_project_data responses

Code	Schema	Description
200	*list_of_basic_project_data*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.12.4.2 GET {worksiteBaseURL}/2024/types/basic_project_data/basic_project_data

Description

Returns the single *basic_project_data* object for a worksite.

Responses

This route shall provide responses as described in Table F.27.

Table F.27 — GET types/basic_project_data/basic_project_data responses

Code	Schema	Description
200	*basic_project_data*	Successful operation.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.5 Codelists routes

F.12.5.1 GET {worksiteBaseURL}/2024/types/codelists

Description

Returns a paginated view of all current codelist data for a worksite.

Parameters

This route shall support the parameters described in Table F.28.

Table F.28 — GET /types/codelists parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.29.

Table F.29 — GET types/codelists responses

Code	Schema	Description
200	*list_of_codelists*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.5.2 GET {worksiteBaseURL}/2024/types/codelists/{id}

Description

Returns a single codelist object for a worksite.

Parameters

This route shall support the parameters described in Table F.30.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.30 — GET types/codelists/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the codelist.

Responses

This route shall provide responses as described in Table F.31.

Table F.31 — GET types/codelists/{id} responses

Code	Schema	Description
200	*codelist*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.6 File_entries routes

F.12.6.1 GET {worksiteBaseURL}/2024/types/file_entries

Description

Returns a paginated view of all current `file_entry` data for a worksite.

Parameters

This route shall support the parameters described in Table F.32.

Table F.32 — GET types/file_entries parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.33.

Table F.33 — GET types/file_entries responses

Code	Schema	Description
200	*list_of_file_entries*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.6.2 GET {worksiteBaseURL}/2024/types/file_entries/{id}

Description

Returns a single `file_entry` object for a worksite.

Parameters

This route shall support the parameters described in Table F.34.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.34 — GET types/file_entries/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the file entry.

Responses

This route shall provide responses as described in Table F.35.

Table F.35 — GET types/file_entries/{id} responses

Code	Schema	Description
200	*file_entry*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.7 file_packages routes

F.12.7.1 GET {worksiteBaseURL}/2024/types/file_packages

Description

Returns a paginated view of all current `file_package` data for a worksite.

Parameters

This route shall support the parameters described in Table F.36.

Table F.36 — GET types/file_packages parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.37.

Table F.37 — GET types/file_packages responses

Code	Schema	Description
200	*list_of_file_packages*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.7.2 GET {worksiteBaseURL}/2024/types/file_packages/{id}

Description

Returns a single *file_package* object for a worksite.

Parameters

This route shall support the parameters described in Table F.38.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.38 — GET types/file_packages/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the file package.

Responses

This route shall provide responses as described in Table F.39.

Table F.39 — GET types/file_packages/{id} responses

Code	Schema	Description
200	*file_package*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.8 Geoids routes

F.12.8.1 GET {worksiteBaseURL}/2024/types/geoids

Description

Returns a paginated view of all current `geoid` data for a worksite.

Parameters

This route shall support the parameters described in Table F.40.

Table F.40 — GET types/geoids parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.41.

Table F.41 — GET types/geoids responses

Code	Schema	Description
200	*list_of_geoids*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.8.2 GET {worksiteBaseURL}/2024/types/geoids/{id}

Description

Returns a single geoid object for a worksite.

Parameters

This route shall support the parameters described in Table F.42.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.42 — GET types/geoids/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the geoid.

Responses

This route shall provide responses as described in Table F.43.

Table F.43 — GET types/geoids/{id} responses

Code	Schema	Description
200	*geoid*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.9 Localizations routes

F.12.9.1 GET {worksiteBaseURL}/2024/types/localizations

Description

Returns a paginated view of all current `localization` data for a worksite. This view will always return at most a single object as each worksite has at most one localization. The view is included in this document for ease of consistent processing of objects.

Parameters

This route shall support the parameters described in Table F.44.

Table F.44 — GET types/localizations parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.45.

Table F.45 — GET types/localizations responses

Code	Schema	Description
200	*list_of_localizations*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.9.2 GET {worksiteBaseURL}/2024/types/localizations/{id}

Description

Returns a single localization object for a worksite.

Parameters

This route shall support the parameters described in Table F.46.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.46 — GET types/localizations/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the localization, which is fixed value = “localization”.

Responses

This route shall provide responses as described in Table F.47.

Table F.47 — GET types/localizations/{id} responses

Code	Schema	Description
200	*localization*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.10 Materials routes

F.12.10.1GET {worksiteBaseURL}/2024/types/materials

Description

Returns a paginated view of all current `material` data for a worksite.

Parameters

This route shall support the parameters described in Table F.48.

Table F.48 — GET types/materials parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.49.

Table F.49 — GET types/materials responses

Code	Schema	Description
200	*list_of_materials*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.10.2GET {worksiteBaseURL}/2024/types/materials/{id}

Description

Returns a single material object for a worksite.

Parameters

This route shall support the parameters described in Table F.50.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.50 — GET types/materials/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the material.

Responses

This route shall provide responses as described in Table F.51.

Table F.51 — GET types/materials/{id} responses

Code	Schema	Description
200	*material*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.11 Operators routes

F.12.11.1GET {worksiteBaseURL}/2024/types/operators

Description

Returns a paginated view of all current `operator` data for a worksite.

Parameters

This route shall support the parameters described in Table F.52.

Table F.52 — GET types/operators parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.53.

Table F.53 — GET types/operators responses

Code	Schema	Description
200	*list_of_operators*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.11.2GET {worksiteBaseURL}/2024/types/operators/{id}

Description

Returns a single operator object for a worksite.

Parameters

This route shall support the parameters described in Table F.54.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.54 — GET types/operators/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the operator.

Responses

This route shall provide responses as described in Table F.55.

Table F.55 — GET types/operators/{id} responses

Code	Schema	Description
200	*operator*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.12 RTKS routes

F.12.12.1GET {worksiteBaseURL}/2024/types/rtks

Description

Returns a paginated view of all current `rtk` data for a worksite.

Parameters

This route shall support the parameters described in Table F.56.

Table F.56 — GET types/rtks parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses

This route shall provide responses as described in Table F.57.

Table F.57 — GET types/rtks responses

Code	Schema	Description
200	*list_of_rtks*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.12.2GET {worksiteBaseURL}/2024/types/rtks/{id}

Description

Returns a single rtk object for a worksite.

Parameters

This route shall support the parameters described in Table F.58.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.58 — GET types/rtks/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the RTK.

Responses:

This route shall provide responses as described in Table F.59.

Table F.59 — GET types/rtks/{id} responses

Code	Schema	Description
200	*rtk*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.13 Work_orders routes

F.12.13.1GET {worksiteBaseURL}/2024/types/work_orders

Description

Returns a paginated view of all current `work_order` data for a worksite.

Parameters:

This route shall support the parameters described in Table F.60.

Table F.60 — GET types/work_orders parameters

Name	Source	Type	Use	Description
limit	query	*integer* between 1 and 100	Optional	Maximum number of values to return (default: 100).

Responses:

This route shall provide responses as described in Table F.61.

Table F.61 — GET types/work_orders responses

Code	Schema	Description
200	*list_of_work_orders*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.12.13.2GET {worksiteBaseURL}/2024/types/work_orders/{id}

Description

Returns a single `work_order` object for a worksite.

Parameters:

This route shall support the parameters described in Table F.62.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.62 — GET types/work_orders/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the work_order.

Responses:

This route shall provide responses as described in Table F.63.

Table F.63 — GET types/work_orders/{id} responses

Code	Schema	Description
200	*work_order*	Successful operation.
400	N/A	Supplied object cannot be parsed.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.13 REST query definitions

F.13.1 list_links

A set of links supplied to enable iteration through a list of results. The SMS shall provide a “next” link except in the case when there are no further results.

Properties:

A *list_links* shall have the properties defined in Table F.64.

Table F.64 — list_links properties

Name	Type	Use	Description
next	*character_string_url*	Optional	Link to the next collection of results.

F.13.2 list_of_assets

A list of assets, together with links for REST navigation.

Properties:

A *list_of_assets* shall have the properties defined in Table F.65.

Table F.65 — list_of_assets properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *assets*	Required	An array of assets.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.3 list_of_assignments

A list of assignments, together with links for REST navigation.

Properties:

A *list_of_assignments* shall have the properties defined in Table F.66.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.66 — list_of_assignments properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *assignments*	Required	An array of assignments.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.4 list_of_basic_project_data

A list containing the single basic_project_data object, together with (useless) links for REST navigation. This method is included so that VISs that call the GET methods in an automated way do not have to treat basic_project_data as a special case.

Properties:

A *list_of_basic_project_data* shall have the properties defined in Table F.67.

Table F.67 — list_of_basic_project_data properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *basic_project_datas*	Required	An array of basic_project_data.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.5 list_of_codelists

A list of codelists, together with links for REST navigation.

Properties:

A *list_of_codelists* shall have the properties defined in Table F.68.

Table F.68 — list_of_codelists properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *codelists*	Required	An array of codelists.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.6 list_of_file_entries

A list of file_entries, together with links for REST navigation.

Properties:

A *list_of_file_entries* shall have the properties defined in Table F.69.

Table F.69 — list_of_file_entries properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *file_entries*	Required	An array of File_Entries.
limit	*integer*	Required	Number of entries per page of data, other than the last.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.13.7 list_of_file_packages

A list of file_packages, together with links for REST navigation.

Properties:

A *list_of_file_packages* shall have the properties defined in Table F.70.

Table F.70 — list_of_file_packages properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *file_packages*	Required	An array of File_Packages.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.8 list_of_geoids

A list of geoids, together with links for REST navigation.

Properties:

A *list_of_geoids* shall have the properties defined in Table F.71.

Table F.71 — list_of_geoids properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *geoids*	Required	An array of geoids.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.9 List of localizations

A list of localizations, of which there are zero-or-one, together with links for REST navigation.

Properties:

A *list_of_localizations* shall have the properties defined in Table F.72.

Table F.72 — list of localizations properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *localizations*	Required	An array of localizations.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.10 list_of_materials

A list of materials, together with links for REST navigation.

Properties:

A *list_of_materials* shall have the properties defined in Table F.73.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.73 — list_of_materials properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *materials*	Required	An array of materials.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.11 list_of_operators

A list of operators, together with links for REST navigation.

Properties:

A *list_of_operators* shall have the properties defined in Table F.74.

Table F.74 — list_of_operators properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *operators*	Required	An array of operators.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.12 list_of_rtks

A list of rtks, together with links for REST navigation.

Properties:

A *list_of_rtks* shall have the properties defined in Table F.75.

Table F.75 — list_of_rtks properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *rtks*	Required	An array of RTKS.
limit	*integer*	Required	Number of entries per page of data, other than the last.

F.13.13 list_of_work_orders

A list of work_orders, together with links for REST navigation.

Properties:

A *list_of_work_orders* shall have the properties defined in Table F.76.

Table F.76 — list_of_work_orders properties

Name	Type	Use	Description
_links	*list_links*	Required	Links for REST navigation.
data	array of *work_orders*	Required	An array of *work_orders.*
limit	*integer*	Required	Number of entries per page of data, other than the last.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.14 REST command routes

F.14.1 General

This API provides communication from a VIS to an SMS. The SMS shall implement all routes defined here.

F.14.2 Assets routes

F.14.2.1 POST {worksiteBaseURL}/2024/types/assets

Description

Create a new asset data entry for a worksite. The fields of the asset shall have the values given in the corresponding fields in the request body. The metadata will be supplied by the SMS.

Parameters:

This route shall support the parameters described in Table F.77.

Table F.77 — POST types/assets parameters

Name	Source	Type	Use	Description
body	body	*asset_upsert_request*	Required	Asset data request metadata.

Responses:

This route shall provide responses as described in Table F.78.

Table F.78 — POST types/assets responses

Code	Schema	Description
201	*asset*	Successful operation, the SMS has the data and will not lose it. A VIS shall not refer to assets in data provided to the as-built and production data APIs without having first POSTed the asset using this route and received a 201 response.
400	N/A	Supplied object cannot be parsed.
401	N/A	The user's policy does not permit posting asset entries.
403	N/A	The SMS does not support posting asset entries.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.14.2.2 DELETE {worksiteBaseURL}/2024/types/assets/{id}

Description

Remove an existing asset data entry for a worksite.

Parameters

This route shall support the parameters described in Table F.79.

Table F.79 — DELETE types/assets/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the asset to delete.

Responses

This route shall provide responses as described in Table F.80.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.80 — DELETE types/assets/{id} responses

Code	Schema	Description
204	N/A	Successful operation.
401	N/A	The user's policy does not permit deleting `asset` entries.
403	N/A	The SMS does not support deleting `asset` entries.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.14.2.3 PUT {worksiteBaseURL}/2024/types/assets/{id}

Description

Update an existing asset data entry for a worksite.

Parameters

This route shall support the parameters described in Table F.81.

Table F.81 — PUT types/assets/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of the asset to update.
body	body	*asset_upsert_request*	Required	Asset data.

Responses

This route shall provide responses as described in Table F.82.

Table F.82 — PUT types/assets/{id} responses

Code	Schema	Description
200	*asset*	Successful operation, the SMS has the data and will not lose it.
400	N/A	Supplied object cannot be parsed.
401	N/A	The user's policy does not permit updating asset entries.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.14.3 operators routes

F.14.3.1 POST {worksiteBaseURL}/2024/types/operators

Description

Create a new operator data entry for a worksite. The metadata will be supplied by the SMS.

Parameters

This route shall support the parameters described in Table F.83.

Table F.83 — POST types/operators parameters

Name	Source		Type	Use	Description
body	body		*operator_upsert_request*	Required	Operator data without metadata.

Responses

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

This route shall provide responses as described in Table F.84.

Table F.84 — POST /types/operators responses

Code	Schema	Description
201	*operator*	Successful operation, the SMS has the data and will not lose it.
400	N/A	Supplied object cannot be parsed.
401	N/A	The user's policy does not permit posting operator entries.
403	N/A	The SMS does not support posting operator entries.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.14.3.2 DELETE {worksiteBaseURL}/2024/types/operators/{id}

Description

Remove an existing operator data entry for a worksite.

Parameters

This route shall support the parameters described in Table F.85.

Table F.85 — DELETE types/operators/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of operator to delete.

Responses

This route shall provide responses as described in Table F.86.

Table F.86 — DELETE types/operators/{id} responses

Code	Schema	Description
204	N/A	Successful operation.
401	N/A	The user's policy does not permit deleting `operator` entries.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.14.3.3 PUT {worksiteBaseURL}/2024/types/operators/{id}

Description

Update an existing operator data entry for a worksite.

Parameters

This route shall support the parameters described in Table F.87.

Table F.87 — PUT types/operators/{id} parameters

Name	Source	Type	Use	Description
id	path	*character_string_id*	Required	ID of operator to update.
body	body	*operator_upsert_request*	Required	Operator data.

Responses

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

This route shall provide responses as described in Table F.88.

Table F.88 — PUT types/operators/{id} responses

Code	Schema	Description
200	*operator*	Successful operation, the SMS has the data and will not lose it.
400	N/A	Supplied object cannot be parsed.
401	N/A	The user's policy does not permit posting operator entries.
404	N/A	Not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

F.15 REST command definitions

F.15.1 asset_upsert_request

Data POSTed by a VIS to create or update an asset as per Subclause F.14.2.1.

Properties

An *asset_upsert_request* shall have the properties defined in Table F.89.

Table F.89 — asset_upsert_request properties

Field	Type	Use	Description
asset_class	*asset_type*	Required	The type of the asset.
equipment_id	*character_string_255*	Optional	A user-friendly string identifying an asset such as a machine.
id	*character_string_uuid*	Required	The VIS asset ID, an identifier determined by the VIS for this asset, generated in conformance with IETF RFC 4122. On any worksite, no two assets shall have the same VIS asset ID. A VIS should make this unique per asset per worksite.
oem_iso_id	*character_string_255*	Optional	This is either a PIN or VIN. The PIN shall be as defined by ISO 10261. The VIN shall be as defined by ISO 3779.
serial_number	*character_string_255*	Optional	This identifies the specific instance of the asset for those assets that are not identified by a PIN or a VIN.

F.15.2 operator_upsert_request

Data POSTed by a VIS to create or update an operator as per Subclause F.14.3.1.

Properties

A *operator_upsert_request* shall have the properties defined in Table F.90.

Table F.90 — operator_upsert_request properties

Name	Type	Use	Description
name	*character_string_1*	Optional	

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

F.16 Confirmation routes

F.16.1 work_order_confirmations routes

F.16.1.1 POST {worksiteBaseURL}/2024/work_order_confirmations

Description

Allows a VIS to confirm to an SMS that a work order and associated files have been received by an asset.

Parameters

This route shall support the parameters described in Table F.91.

Table F.91 — POST work_order_confirmations parameters

Name	Source	Type	Use	Description
body	Body	*work_order_confirmation_post_body*	Required	An object describing the confirmation.

Responses

This route shall provide responses as described in Table F.92.

Table F.92 — POST work_order_confirmations responses

Code	Schema	Description
200	*work_order_confirmation_response_body*	Successful operation.
401	N/A	Unauthorized.

F.17 Confirmation definitions

F.17.1 work_order_confirmation_post_body

Sent from a VIS to an SMS to confirm the result of sending a work order and its associated file packages to an asset. The confirmation is final, the VIS shall not report anything if it is still retrying or possibly expecting a confirmation from the asset.

Properties

A *work_order_confirmation_post_body* shall have the properties defined in Table F.93.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table F.93 — work_order_confirmation_post_body properties

Name	Type	Use	Description
asset	*character_string_id*	Required	The ID of the asset.
at	*datetime_3339*	Required	The time at which the assignment was confirmed.
description	*character_string_1000*	Optional	If the result is FAIL, an English description of the failure.
id	*character_string_id*	Required	The ID of the work order being confirmed.
operator	*character_string_id*	Optional	The ID of the `operator` of the asset, if known.
result	*work_order_-confirmation_result_enum*	Required	Processing result.
rev	*datetime_3339*	Required	Revision of the work order as defined by its “at” attribute.

F.17.2 work_order_confirmation_response_body

The response object currently has no fields.

F.17.3 work_order_confirmation_result_enum

Legal values for the confirmation:

— `OK` : The asset has received the work order and its associated file packages.

— `FAIL` : Delivery failed.

— `UNKNOWN` : Delivery attempted, result unknown.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex G
(normative)

As-built and production system-of-record types

G.1 System-of-record types

G.1.1 General

This annex defines the system-of-record data stream types which are communicated between an SMS on one end and a VIS or an SRS on the other, using the API defined in Annex I.

System-of-record types shall be communicated as messages encoded as JSON objects. Each such message shall conform to JSON schemas discussed in Annex N. JSON objects shall be communicated using the UTF-8 encoding in conformance with IETF RFC 3629. Where values recorded as attributes of system-of-record JSON objects require units, those units shall be SI units unless otherwise stated in this document.

G.1.2 Definition "abprod_sor_metadata"

Each system-of-record message shall have a "meta" field of type *abprod_sor_metadata*.

An *abprod_sor_metadata* shall be as defined in Table G.1.

Table G.1 — abprod_sor_metadata fields

Field	Type	Use	Description
asset	*character_string_uuid*	Required	The ID of the project data asset produced by this VIS for the asset this meta data refers to.
id	*character_string_uuid*	Required	A unique ID (generated in conformance with IETF RFC 4122) to identify a message such that the message contents can be reissued by the VIS to SMS.
type	*string, one of(streak, tin, survey, ref_survey)*	Required	The as-built and production data system-of-record message types.
v	*character_string_255*	Required	Semantic version of the schema set.

G.2 Type "streak"

G.2.1 General

A *streak* message provides a set of atomic fragments each containing a journal of measurements of working tool-tip positions. Attribute information (such as temperature measured at the working tool) can also be stated.

The *streak* message is considered to be atomic. All data related to the state of the asset is provided in a single message. Accordingly, the asset contribution for a time period can be applied to a surface using this single message.

A *streak* should typically contain 30 seconds to 5 minutes worth of measurement data.

Existing *streak* objects shall only be amended to add new *streak_fragment_measurements* to the end of the last existing *streak_fragment*, and to add new *streak_fragments* to the end of fragments list. Ordering by time ascending shall be retained in both cases.

A *streak* shall be as defined in Table G.2.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table G.2 — streak fields

Field	Type	Use	Description
fragments	array of between 1 and 100 *streak_fragments*	Required	Streak fragments ordered by time ascending.
meta	*abprod_sor_metadata* with type=streak	Required	*abprod_sor_metadata* with type=streak.

G.2.2 Definition "*streak_fragment*"

Streak fragments are required by changes to the asset state (worked material, configuration of working tool shapes etc). All measurements within a *streak_fragment* share the same asset state. Fragments shall appear in order of ascending measurement "at" values. For any asset, the ranges of fragment "at" shall not overlap.

Every fragment shall contain at least two journalled measurements. The last measurement in a fragment may match the first measurement in the following fragment in order to produce a continuous streak across fragments. Alternatively, where the last measurement from a fragment does not match the first measurement in the following fragment, there is an explicit gap in the streak. The VIS shall not issue fragments for an asset that is stationary.

The measurements in a fragment form a journal. Any attribute set in a measurement retains the set value for subsequent measurements until explicitly overridden. The first measurement frame in a fragment shall set the relevant attributes for the asset type, to set the initial conditions of the journal.

A *streak_fragment* shall be as defined in Table G.3.

Table G.3 — streak_fragment fields

Field	Type	Use	Description
measurements	array of between 2 and 6000 *streak_fragment_measurements*	Required	An array of measurements giving the time, position and other attributes of the working tool shapes.
state	*streak_fragment_state*	Required	State common to all measurements in a fragment.

G.2.3 Definition "*streak_fragment_state*"

The stateful context for all measurements within a fragment. These include references to objects in project data, codelist references and shape definitions, etc.

A *streak_fragment_state* shall be as defined in Table G.4.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table G.4 — streak_fragment_state fields

Field	Type	Use	Description
id	*character_string_uuid*	Optional	Allows streak objects to be tied together into a continuous streak for possible reporting purposes.
proj_ref	*proj_ref*	Optional	Contains references to objects in project data such as the working material.
purpose_st	*purpose_enum*	Optional	A statement relating to the purpose of the measurement work being done.
sensor_quality	*sensor_quality*	Optional	A statement relating to the quality of the geospatial measurements within the fragment expressed by identifying used positioning technology.
shapes	array of between 1 and 6 *shape_configurations*	Required	Defines the working tool configuration for the fragment. The length of the array **S** is equal to the number of working tools.

G.2.4 Definition "*proj_ref*"

Contains references to project data objects.

A *proj_ref* shall be as defined in Table G.5.

Table G.5 — proj_ref fields

Field	Type	Use	Description
attributes	array of between 0 and 100 *codelist_attribute_values*	Optional	Array of ID-value pairs of *code_list codelist_attributes*. This array contains the values set for attributes assigned to category or feature code.
category_code	*character_string_4*	Optional	Reference to the *code_list codelist_code* by name.
code_list	*character_string_id*	Required if any of *["category_code", "feature_code", "attributes"]* is set	Reference to a project data *codelist.*
feature_code	*character_string_4*	Optional	Reference to the *code_list codelist_code* by name.
material	*character_string_id*	Optional	Reference to a project data *material.*
operator	*character_string_id*	Optional	Reference to a project data *operator.*
reference_object	*reference_object*	Optional	Reference to an ISO LandXML project data file and element within.
work_order	*character_string_id*	Optional	Reference to a project data *work_order.*

Further project reference fields are subject to the maintenance agency,

G.2.5 Definition "*codelist_attribute_value*"

A tuple identifying a *codelist_attribute* and a value.

A *codelist_attribute_value* shall be as defined in Table G.6.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table G.6 — codelist_attribute_value fields

Field	Type	Use	Description
attribute	*character_string_id*	Required	The ID of the *codelist_attribute.*
value	*integer, number, character_string_255* or *datetime_3339*	Required	The value of the attribute.

G.2.6 Definition "*reference_object*"

Defines a reference to an ISO LandXML project data *file_entry* and a named element within.

A *reference_object* shall be as defined in Table G.7.

Table G.7 — reference_object fields

Field	Type	Use	Description
element_name	*character_string_id*	Required	The reference object element name within the file.
file	*project_file_rev*	Required	The project data *file_entry* id and revision.

G.2.7 Definition "*project_file_rev*"

Defines a reference to a revisioned project data *file_entry*.

A *project_file_rev* shall be as defined in Table G.8.

Table G.8 — project_file_rev fields

Field	Type	Use	Description
id	*character_string_id*	Required	File entry ID.
rev	*datetime_3339*	Required	File entry revision as given by its `at` value.

G.2.8 Definition "*sensor_quality*"

The *sensor_quality* definition is of the enumeration:

— `rtk`

— `total_station`

— `other`

A VIS should map used positioning sensor technology to this enumeration. The SMS may use the *sensor_quality* as a statement of position quality based on the sensor technology present.

This quality enumeration is subject to the maintenance agency.

G.2.9 Definition "*purpose_enum*"

purpose_enum is the following enumeration:

— `asbuilt`: The measurement is intended for final as-built data reporting.

— `current`: The measurement is intended for production data reporting.

A VIS shall not state the optional "purpose_st" field unless the purpose of the measurement work is explicitly determined by the work-flow processes defined by the VIS system.

G.2.10 Definition "*shape_configuration*"

A *shape_configuration* shall be as defined in Table G.9.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table G.9 — shape_configuration fields

Field	Type	Use	Description
ic_sensors	array of between 1 and 4 *ic_sensor_infos*	Optional	Optional array defining the set of unique IC sensor kinds for the working tool.
n	*integer* between 1 and 4	Required	The number of measured points **N** for the working tool.
smf	Nullable array of between 1 and 4 *surface_modelling_function_enums*	Required	Describes how the working tool shape affects a simulated surface. A null value indicates that the shape shall not affect the surface.

G.2.11 Definition "*ic_sensor_info*"

An *ic_sensor_info* defines intelligent compaction (IC) sensor information for a working tool.

An *ic_sensor_info* shall be as defined in Table G.10.

Table G.10 — ic_sensor_info fields

Field	Type	Use	Description
kind	*ic_sensor_kind_enum*	Required	The kind of IC sensor.

G.2.12 Definition "*ic_sensor_kind_enum*"

This enumeration defines the intelligent compaction sensor used.

ic_sensor_kind_enum is the following enumeration:

— `evib1`

— `evib2`

— `hmv`

— `cmv`

— `kb`

— `mdp`

— `omega`

— `fdvk`

— `ccv`

— `other`

This enumeration is subject to the maintenance agency.

G.2.13 Definition "*surface_modelling_function_enum*"

This enumeration defines how a working tool shape should affect a machine measured surface. *surface_modelling_function_enum* is the following enumeration:

— `height`: the working tool shape should affect the height of the surface;

— `pass`: the working tool shape should increment pass counts;

— `temp`: the working tool shape has a temperature value with each vertex;

— `cv`: the working tool shape has a compaction value with each vertex.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

G.2.14 Definition "*streak_fragment_measurement*"

A *streak_fragment_measurement* is a frame containing position and other attributes for the working tool shapes. The *streak_fragment* "measurements" field is an array of these measurements, which animate the position and other attributes of the asset and its working tool shapes over time.

Streak fragment measurements form a journal. Accordingly, optional *streak_fragment_measurement* fields are not required to be stated unless they have changed from the previous measurement or status in the fragment. If they are not stated in the *streak_fragment_measurement*, it implies they hold the value from the last *streak_fragment_measurement* in which they were set. A value of null for any field indicates the measurement or state is being cleared to an undefined value.

A *streak_fragment_measurement* shall be as defined in Table G.11.

Table G.11 — streak_fragment_measurement fields

Field	Type	Use	Description
at	*integer*	Required	Measurement timestamp, encoded as the number of milliseconds in accordance with the ISO 8601 time epoch.
h	nullable *number*	Optional	Heading of the machine asset. The azimuth is in radians in the local site cartesian frame, with a positive yaw turning the front of the asset to the right (clockwise).
reverse	nullable *boolean*	Optional	The asset is in reverse gear.
s	array of between 1 and 6 *streak_shapes*	Required	An array of length **S** (the number of working tools), matching *the streak_fragment_state* shapes definition.
v	nullable *number*	Optional	Velocity in meters per second.

The *streak_shape* field "s" is of length **S** (the number of working tools) matching the *streak_fragment_state* "shapes" definition. Shape indexes match, such that the shape at index *j* refers to the same working tool for all measurements in the fragment.

Further measurement attributes are subject to the maintenance agency.

G.2.15 Definition "*streak_shape*"

A *streak_shape* defines the positions and attributes for a working tool shape, within a *streak_fragment_measurement*.

Streak fragment measurements form a journal. Accordingly, optional *streak_shape* fields are not required to be stated unless they have changed from the previous measurement or status in the fragment. If they are not stated in the *streak_shape* it implies they hold the value from the last *streak_shape* in which they were set. A value of null for any field indicates the measurement or state is being cleared to an undefined value.

A *streak_shape* shall be as defined in Table G.12.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table G.12 — streak_shape fields

Field	Type	Use	Description
auto	nullable *boolean*	Optional	The working tool is in automatic mode.
cv	nullable array of *numbers*	Optional	An array with length matching *shape_configuration ic_sensors*. Each entry *cv[j]* gives a raw compaction value for the working tool, as defined by *shape_configuration ic_sensors[j].*
ncv	nullable array of *numbers*	Optional	An array with length matching *shape_configuration ic_sensors*. Each entry *cv[j]* gives a calibrated normalized compaction value for the working tool, as defined by *shape_configuration ic_sensors[j].*
p	nullable array of 3 *numbers*	Optional	An array of length **N** matching *shape_configuration n* (the number of measured points along the working tool). Each entry *p[j*] is an array of 3 numbers, giving the local NEE coordinates for measured point j.
q	nullable array of 2 *numbers*	Optional	An array of 2 numbers, the horizontal and vertical accuracy within 1 standard deviation. These apply to all measured points.
smc	nullable *surface_modelling_condition_enum*	Optional	Surface modelling condition indicates how the shape shall be applied to a machine measured surface. A null value indicates that the shape shall not be applied.
t	nullable array of *numbers*	Optional	An array of length **N** matching *shape_configuration* n (the number of measured points along the working tool). Each entry *t[j*] is a number giving a temperature measured in degrees Celsius for measured point *j.*
va	nullable *number*	Optional	The vibration amplitude of the working tool in millimeters.
vf	nullable *number*	Optional	The vibration frequency of the working tool in cycles per second.
vj	nullable *boolean*	Optional	Vibration jump is detected for the working tool.
vs	nullable *boolean*	Optional	The vibration state of the working tool. Values are: true = vibration on, false = vibration off.

Further shape attributes are subject to the maintenance agency.

G.2.16 Definition "*surface_modelling_condition_enum*"

This enumeration indicates how the SMS should update its current as-built surface model based on received streak data. Received data should only be applied to the modelled as-built surface when the enumerated condition holds.

surface_modelling_condition_enum is the following enumeration:

— `false`: never apply the shape;

— `true`: always apply the shape;

— `lower_or_no_existing`: apply the shape when either lower than existing surface height or no surface height exists;

— `higher_or_no_existing`: apply the shape when either higher than existing surface height or no surface height exists;

— `has_existing`: apply the shape when surface height exists;

— `lower_and_has_existing`: apply the shape when lower than existing surface height;

— `higher_and_has_existing`: apply the shape when higher than existing surface height.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

G.3 Type "*tin*"

G.3.1 General

The *tin* message provides a small triangulated irregular network (TIN) which defines topographic data local to an asset. This can be calculated by edge processing at the machine (such as by scanner hardware, stereo camera depth map on the asset). This message is used to augment working tool as-built information with topographic data available from other machine sensors.

The *tin* message contains NEZ points in the local grid coordinates space, and shall be a surface which is relatively local and near to the asset which produces the message.

A *tin* message shall not exceed an area of 30 meter radius. The SMS can reject a *tin* message larger than this radius.

The *tin* message type should be produced at low frequency per asset: typically less than 1 per 10 seconds. Existing *tin* objects shall not be amended in any way.

A *tin* shall be as defined in Table G.13.

Table G.13 — tin fields

Field	Type	Use	Description
at	*integer*	Required	The time at which the TIN was captured, encoded as the number of milliseconds in accordance with the ISO 8601 time epoch.
faces	array of between 1 and 10000 arrays of 3 *integers*	Required	Array of triangle [p0,p1,p2], where pN indexes the points array (from 0 inclusive). Face normal shall be defined as clockwise winding order like in the DTM.
meta	*abprod_sor_metadata* with type=tin	Required	*abprod_sor_metadata* with type=tin
points	array of between 3 and 30000 arrays of 3 *numbers*	Required	Array of local NEE coordinates in meters.
state	*tin_state*	Required	State for the TIN capture.

G.3.2 Definition "*tin_state*"

State context for a tin message. These include references to objects in project data etc. A *tin_state* shall be as defined in Table G.14.

Table G.14 — tin_state fields

Field	Type	Use	Description
proj_ref	*proj_ref*	Optional	Contains references to objects in project data such as the working material.
purpose_st	*purpose_enum*	Optional	A statement relating to the purpose of the measurement work being done.
sensor_quality	*sensor_quality*	Optional	A statement relating to the quality of the geospatial measurements within the TIN, expressed by identifying the set of position technologies used in sensor aggregation.

G.4 Type "*survey*"

G.4.1 General

The survey message provides a set of surveyed points as part of the as-built and production stream.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Surveyed points with matching line_id form a line. Points in a line are ordered by increasing at timestamp. More than one survey object can contain points for a single line.

Existing *survey* objects shall only be amended to delete *survey_points*. New points shall not be added and existing points shall not be modified. When a point is deleted, it shall be removed from the list of survey's points and its ID shall be added to the list of deleted points.

A *survey* shall be as defined in Table G.15.

Table G.15 — survey fields

Field	Type	Use	Description
deleted_points	array of at most 1000 *character_string_uuids*	Optional	Array of IDs of previously included points that have been deleted.
meta	*abprod_sor_metadata* with type=survey	Required	*abprod_sor_metadata* with type=survey
points	array of at most 1000 *survey_points*	Required	Array of surveyed points.
state	*survey_state*	Required	State for the survey.

G.4.2 Definition "*survey_state*"

State context for a survey message. These include references to objects in project data etc. A *survey_state* shall be as defined in Table G.16.

Table G.16 — survey_state fields

Field	Type	Use	Description
proj_ref	*proj_ref*	Optional	Contains references to objects in project data such as the working material. Applies to all points except where point contains override.
purpose_st	*purpose_enum*	Optional	A statement relating to the purpose of the measurement work being done.
sensor_quality	*sensor_quality*	Optional	A statement relating to the quality of the geospatial measurements within the survey, expressed by identifying the set of position technologies used in sensor aggregation.

G.4.3 Definition "*survey_point*"

Position and attributes for a surveyed point.

A *survey_point* shall be as defined in Table G.17.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table G.17 — survey_point fields

Field	Type	Use	Description
at	*integer*	Required	The time at which the point was surveyed, encoded as the number of milliseconds in accordance with the ISO 8601 time epoch.
attr	array of between 1 and 10 *codelist_attribute_values*	Optional	Array of id-value pairs of *code_list_attributes*. This array contains the values set for attributes assigned to category or feature code.
dh	*number*	Optional	Horizontal difference in meters between design object and survey point.
dn	*number*	Optional	Difference in meters between design object and survey point in surface normal direction (applicable only for surface and line reference objects).
dz	*number*	Optional	Vertical difference in meters between design object and survey point.
id	*character_string_uuid*	Required	Unique survey point identifier (in conformance with IETF RFC 4122).
line_id	*character_string_uuid*	Optional	Unique line identifier (in conformance with IETF RFC 4122).
p	array of 3 *numbers*	Required	Local NEE coordinates in meters.
proj_ref	*proj_ref*	Optional	Overrides state.proj_ref for this point.
q	array of 2 *numbers*	Optional	An array of 2 numbers: the horizontal and vertical accuracy estimate in meters, within 1 standard deviation.

G.5 Type "*ref_survey*"

A *ref_survey* message provides a survey by reference to a project data *file_entry*, and an element name within the entry. Survey files may be uploaded to the SMS as detailed in Clause F.7.

Existing *ref_survey* objects shall not be amended in any way.

A *ref_survey* shall be as defined in Table G.18.

Table G.18 — ref_survey fields

Field	Type	Use	Description
meta	*abprod_sor_metadata* with type=ref_survey	Required	*abprod_sor_metadata* with type=ref_survey
state	*survey_state*	Required	State for the survey.
survey	*reference_object*	Required	Reference to an ISO LandXML project data file and element within, which contains the survey content.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex H
(normative)

As-built and production real-time types

H.1 Real-time types

H.1.1 General

This annex defines the real-time data stream types which are communicated between SMS and VIS, using the API defined in Annex J.

Real-time types shall be communicated as messages encoded as JSON objects. Each such message shall conform to JSON schema as discussed in Annex N. JSON objects shall be communicated using the UTF-8 encoding in conformance with IETF RFC 3629. Where values recorded as attributes of system-of-record JSON objects require units, those units shall be SI units unless otherwise stated in this document.

The *asset_configuration* and *replicate* message types are used to communicate the location and shape of machine assets in soft real-time on a connected worksite.

The goal of these data types is to support features such as:

— Viewing an asset in real-time on a remote display (2D or 3D).

— Determining working tool positions for real-time as-built surface modeling.

— Determining points of interest for an asset digital twin.

— Validating asset calibration.

— Workflow and process control.

Forward kinematic terms (such as the angle between an excavator boom and stick) are used to express how an asset is currently positioned in three dimensional space. These terms, along with sensor values, are communicated over the network at high frequency in a *replicate* message. Information which is more static (such as the width of the asset) is communicated less often, as needed, in the *asset_configuration* message. Splitting the stream into these two primary types (*asset_configuration* and *replicate*) allows many points of interest and much graphical detail to be communicated in fewer bytes.

H.1.2 Definition "*abprod_rt_metadata*"

Each real-time message shall have a "meta" field of type *abprod_rt_metadata*. An *abprod_rt_metadata* shall be as defined in Table H.1.

Table H.1 — abprod_rt_metadata fields

Field	Type	Use	Description
asset	*character_string_uuid*	Required	The ID of the project data asset produced by this VIS for the asset this meta data refers to.
at	*integer*	Required	Unless specified otherwise, this is the time at which the event occurred at the asset. It is encoded as the number of milliseconds in accordance with the ISO 8601 time epoch.
type	*string*, one of(*state, asset_configuration*, *replicate*)	Required	The as-built and production data real-time message types.
v	*character_string_255*	Required	Semantic version of the schema set.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

H.2 Type "*state*"

H.2.1 General

State messages should appear for each asset at the start of a stream to provide context for each asset currently operating on the worksite. As state changes occur, state messages appear on the stream to set the new state. If a state is to be completely cleared, a state message may be sent which has no *proj_ref_rt* field. To clear individual values a VIS sends null values for the relevant fields in *proj_ref_rt*.

A *state* shall be as defined in Table H.2.

Table H.2 — state fields

Field	Type	Use	Description
meta	*abprod_rt_metadata* with type=state	Required	*abprod_rt_metadata* with type=state
proj_ref_rt	*proj_ref_rt*	Optional	Contains references to objects in project data.

H.2.2 Definition "*proj_ref_rt*"

Contains references to project data objects which form the current state of an asset. A *proj_ref_rt* shall be as defined in Table H.3.

Table H.3 — proj_ref_rt fields

Field	Type	Use	Description
attributes	nullable array of between 0 and 100 *codelist_attribute_value_rts*	Optional	Array of ID-value pairs of *code_list codelist_attributes*. This array contains the values set for attributes assigned to category or feature code.
category_code	nullable *character_string_4*	Optional	Reference to the *code_list codelist_code* by name.
code_list	nullable *character_string_id*	Required if any of ["category_code", "feature_code", "attributes"] is set	Reference to a project data *codelist.*
feature_code	nullable *character_string_4*	Optional	Reference to the *code_list codelist_code* by name.
material	nullable *character_string_id*	Optional	Reference to a project data *material.*
operator	nullable *character_string_id*	Optional	Reference to a project data *operator.*
reference_object	nullable *reference_object_rt*	Optional	Reference to an ISO LandXML project data file and element within.
work_order	nullable *character_string_id*	Optional	Reference to a project data *work_order.*

Further project reference fields are subject to the maintenance agency.

H.2.3 Definition "*codelist_attribute_value_rt*"

A tuple identifying a *codelist_attribute* and a value.

A *codelist_attribute_value_rt* shall be defined in Table H.4.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table H.4 — codelist_attribute_value_rt fields

Field	Type	Use	Description
attribute	*character_string_id*	Required	The ID of the *codelist_attribute.*
value	*integer*, *number*, *character_string_255* or *datetime_3339*	Required	The value of the attribute.

H.2.4 Definition "*reference_object_rt*"

Defines a reference to an ISO LandXML project data *file_entry* and a named element within. A *reference_object_rt* shall be as defined in Table H.5.

Table H.5 — reference_object_rt fields

Field	Type	Use	Description
element_name	*character_string_id*	Required	The reference object element name within the file.
file	*project_file_rev_rt*	Required	The project data *file_entry* ID and revision.

H.2.5 Definition "*project_file_rev_rt*"

Defines a reference to a revisioned project data *file_entry*.

A *project_file_rev_rt* shall be as defined in Table H.6.

Table H.6 — project_file_rev_rt fields

Field	Type	Use	Description
id	*character_string_id*	Required	File entry ID.
rev	*datetime_3339*	Required	File entry revision as given by its `at` value.

H.3 Type "*asset_configuration*"

H.3.1 General

An *asset_configuration* message defines the static configuration of a machine asset on the worksite. The *asset_configuration* describes the asset's articulation, size, calibrated measurements, attachment points and references to 2D and 3D model data, etc.

An *asset_configuration* message can describe articulated assets such as:

— an excavator with stick, boom and attached implement;

— a crane with a crane arm;

— a paving machine with a translating screed;

— a light vehicle;

— truck and trailer;

— a boom gate.

An *asset_configuration* message shall only appear on a stream when there are changes to the configuration or calibration of the asset. Such changes should happen at low frequency, thereby minimizing stream bandwidth. A moving asset, or asset which has sensor values changing should not result in new *asset_configuration* messages on the stream.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

An *asset_configuration* message is referenced by *replicate* messages. Accordingly, an *asset_configuration* message shall appear on a stream before any *replicate* message which references it.

An *asset_configuration* shall be as defined in Table H.7.

Table H.7 — asset_configuration fields

Field	Type	Use	Description
asset_class	*asset_type*	Required	The type of the asset.
components	array of *components*	Required	The set of *components* which make up the *asset_configuration.*
id	*character_string_uuid*	Required	Uniquely identifies the *asset_configuration*. This shall be a new UUID for any change (such as implement or calibration change) made to the *asset_configuration*. SMS and VIS may use this as a caching key. This does not identify an asset, but identifies a unique configuration and calibration of an asset.
meta	*abprod_rt_metadata* with type=asset_configuration	Required	*abprod_rt_metadata* with type=asset_configuration

H.3.2 Definition "*component*"

An *asset_configuration* is defined as a collection of components. The main body of an asset is represented by a *component*. Accordingly, an *asset_configuration* shall contain at least one component. Implements which can be changed, such as an excavator bucket, can be represented by further components.

It is not intended that components be used to represent an asset's fixed articulated hierarchy (such as the boom and stick of an excavator). Components should be used for the parts of an asset which are changeable by the operator (implements such as excavator buckets).

A *component* shall be as defined in Table H.8.

Table H.8 — component fields

Field	Type	Use	Description
component_type	*component_type_enum*	Required	A registered name for the category of *component*.
interfaces	array of *interfaces*	Required	The set of *interfaces* which define the *component*.

H.3.3 Definition "*component_type_enum*"

This enum is subject to the maintenance agency.

— `main_body`: Used for the main body of an asset. An asset_configuration shall only have one main_body component.

— `implement`: Used for an implement or attachment.

— `other`

H.3.4 Definition "*interface*"

A *component* is defined as a collection of *interfaces,* which provides an extensible way to define a component. Foundational *interfaces* are defined in Table H.9.

Extra *interfaces* may be defined by vendors to extend real-time functionality.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

An *interface* shall be as defined in Table H.9.

Table H.9 — interface fields

Field	Type	Use	Description
interface_type	string matching the regular expression `^[a-z_]+\.[a-z_]+$`	Required	Broken into two parts separated by a dot. The first part is a namespace, which represents a company or organization defining the interface. The second part names their interface. Matches: `^[a-z_]+\.[a-z_]+$`

An *interface* shall also contain attributes as defined in exactly one of the following subschemata:

a) *interface_position*

b) *interface_nodes*

c) *interface_points_of_interest*

d) *interface_collision_shapes*

e) *interface_asbuilt_shapes*

f) *interface_replicate*

g) *interface_attach*

h) *interface_render_assets*

i) *interface_unknown*

H.3.5 Referencing objects

Any JSON object which is a descendant of an *interface* may have an "id" field of type string. The value of the "id" field shall be unique within the interface JSON hierarchy. For example:

```
{
   "interface_type ": "xyz.weather",
   "values" :
   [
      {
           "id": "temperature",
           "degrees": 32.0
      },
      {
           "id": "humidity",
           "rel": 87
      }
   ]
}
```

In the example, vendor XYZ has defined a "weather" interface with two uniquely identified objects: "temperature" and "humidity". ID values only must be unique within the scope of the vendor XYZ "weather" interface. ID values shall match the regular expression:

```
^[a-z_]+$
```

Any JSON object within a *component* may be referenced using a 3 dot object referencing scheme:

```
<namespace>.<interface>.<object id>
```

For example, the "temperature" object in the previous example is referenced using the 3 dot object reference string:

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

```
"xyz.weather.temperature"
```

All 3 dot object references shall resolve to a valid object within the *asset_configuration*. A 3 dot object reference which does not resolve shall cause a load failure.

H.3.6 Referencing values

A JSON field of type number, within an object having an "id" field may be referenced using a 4 dot value referencing scheme:

```
<namespace>.<interface>.<object id>.<value>
```

Accordingly, the "humidity" in the previous example is referenced with the 4 dot value reference string:

```
"xyz.weather.humidity.rel"
```

Value field names shall match the regular expression:

```
^[a-z_]+$
```

All 4 dot value references shall resolve to a valid number within the *asset_configuration*. A 4 dot value reference which does not resolve shall cause a load failure.

H.4 Interface definitions

H.4.1 General

H.4 describes the foundational ISO *interface* definitions which should be provided by a VIS to the SMS for an *asset_configuration*.

The *interface* definitions work together, and relate to one another. Accordingly, a single calibration measurement on an asset can relate to many values across *interfaces* and *components* in an *asset_configuration*. For example, the length of an excavator boom will apply to stick translation, collision shape extents, and render asset scale.

H.4.2 Definition "*interface_position*"

The position interface is used to translate and rotate an asset relative to a worksite localization. Within a *component*, an *interface_position* is mutually exclusive with an *interface_attach*.

An *interface_position* shall be as defined in Table H.10.

Table H.10 — interface_position fields

Field	Type	Use	Description
interface_type	the value "*iso.position*"	Required	Constant value.
position	*position_local*	Required	Local position.

H.4.2.1 Definition "*position_local*"

A *position_local* shall be as defined in Table H.11.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table H.11 — position_local fields

Field	Type	Use	Description
easting	*number*	Required	Distance in meters.
elevation	*number*	Required	Distance in meters.
id	the value "*local*"	Required	Required for 4 dot value referencing.
northing	*number*	Required	Distance in meters.
pitch	*number*	Required	In radians, a pitch of 0 is a level asset with a positive pitch bringing the front of the asset up.
roll	*number*	Required	In radians, a roll of 0 is a level asset with a positive roll tilling the tip of the asset to the right.
yaw	*number*	Required	In radians, a yaw of 0 is facing due north with a positive yaw turning the front of the asset to the right.

H.4.3 Definition "*interface_nodes*"

The nodes interface defines an articulated hierarchy for the *component*. It models the skeletal structure of an asset (such as the body, boom or stick hierarchy in an excavator). This allows asset updates to send only the forward kinematic terms such as for boom angle and stick angle to replicate the asset remotely.

interface_nodes depends on either *interface_position* or *interface_attach*.

interface_nodes shall be as defined in Table H.12.

Table H.12 — interface_nodes fields

Field	Type	Use	Description
interface_type	the value "*iso.nodes*"	Required	Constant value.
nodes	array of *nodes_nodes*	Required	Root nodes, which inherit the spatial reference frame defined by *interface_position* or *interface_attach*.

H.4.3.1 Definition "*nodes_node*"

Each node defines a translation, rotation, and scale. The coordinate space is SAE J670, right handed, and Z down with front facing in the positive X axis. Rotations are applied in the order of roll, followed by pitch, followed by yaw.

A *nodes_node* shall be as defined in Table H.13.

Table H.13 — nodes_node fields

Field	Type	Use	Description
id	*character_string_255*	Required	node ID, used for 3 dot and 4 dot reference scheme.
nodes	array of *nodes_nodes*	Optional	Child nodes, which inherit the spatial reference frame of this node.
rx	*number*	Required	In radians, rotation around X-axis.
ry	*number*	Required	In radians, rotation around Y-axis.
rz	*number*	Required	In radians, rotation around Z-axis.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table H.13 *(continued)*

Field	Type	Use	Description
sx	*number*	Required	Scaling factor, X-axis.
sy	*number*	Required	Scaling factor, Y-axis.
sz	*number*	Required	Scaling factor, Z-axis.
tx	*number*	Required	In meters, X-axis translation.
ty	*number*	Required	In meters, Y-axis translation.
tz	*number*	Required	In meters, Z-axis translation.

H.4.4 Definition "*interface_points_of_interest*"

The points-of-interest interface defines points of interest on the asset, where each point of interest states a 3 dot objects reference to a *nodes_node*.

interface_points_of_interest depends on *interface_nodes*.

Points of interest are referenced by other interfaces, such as *interface_asbuilt_shapes*.

An *interface_points_of_interest* shall be as defined in Table H.14.

Table H.14 — interface_points_of_interest fields

Field	Type	Use	Description
interface_type	the value "*iso.points_of_interest*"	Required	Constant value.
points	array of *poi_points*	Required	Points of interest.

H.4.4.1 Definition "*poi_point*"

A point defined by a translation in relation to a *nodes_node*.

A *poi_point* shall be as defined in Table H.15.

Table H.15 — poi_point fields

Field	Type	Use	Description
id	*character_string_255*	Required	Point ID, used for 3 dot and 4 dot reference scheme.
node_ref	string matching the regular expression `^iso\.nodes\.[a-z_]+$`	Required	3 dot object reference to a nodes_node. Matches `^iso\.nodes\.[a-z_]+$`
tx	*number*	Required	In meters, X-axis translation, in relation to node_ref.
ty	*number*	Required	In meters, Y-axis translation, in relation to node_ref.
tz	*number*	Required	In meters, z-axis translation, in relation to node_ref.

H.4.5 Definition "*interface_collision_shapes*"

The collision-shapes interface defines an estimate volume of the component for collision avoidance. This interface should not be used for real-time or safety critical collision detection purposes.

interface_collision_shapes depends on *interface_nodes*.

An *interface_collision_shapes* shall be as defined in Table H.16.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table H.16 — interface_collision_shapes fields

Field	Type	Use	Description
interface_type	the value "*iso.collision_shapes*"	Required	Constant value.
shapes	array of *collision_shapes*	Required	Collision shapes.

H.4.5.1 Definition "*collision_shape*"

Either an axis-aligned-bounding-box (aabb) or sphere relative to a referenced *nodes_node*. A *collision_shape* shall be as defined in Table H.17.

Table H.17 — collision_shape fields

Field	Type	Use	Description
cx	*number*	Required	In meters, shape center X, in relation to node_ref.
cy	*number*	Required	In meters, shape center Y, in relation to node_ref.
cz	*number*	Required	In meters, shape center Z, in relation to node_ref.
id	*character_string_255*	Required	Shape ID, used for 3 dot and 4 dot reference scheme.
node_ref	string matching the regular expression `^iso\.nodes\.[a-z_]+$`	Required	3 dot object reference to a nodes_node. Matches `^iso\.nodes\.[a-z_]+$`
type	*string*, one of(*aabb, sphere)*	Required	

A collision_shape shall also contain attributes as defined exactly in one of the following subschemata:

a) *collision_shape_aabb*

b) *collision_shape_sphere*

Subschema collision_shape_aabb

A *collision_shape_aabb* shall be as defined in Table H.18.

Table H.18 — collision_shape_aabb fields

Field	Type	Use	Description
hex	*number*	Required	In meters, half-extents X.
hey	*number*	Required	In meters, half-extents Y.
hez	*number*	Required	In meters, half-extents Z.
type	the value "*aabb*"	Required	

Subschema collision_shape_sphere

A *collision_shape_sphere* shall be as defined in Table H.19.

Table H.19 — collision_shape_sphere fields

Field	Type	Use	Description
radius	*number*	Required	In meters, radius.
type	the value "*sphere*"	Required	

H.4.6 Definition "*interface_asbuilt_shapes*"

The as-built shapes interface defines working tool shapes, such as a blade edge, or the bottom of roller drum. These shapes may be use for simulating machine measured surfaces.

interface_asbuilt_shapes depends on *interface_points_of_interest*.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

An *interface_asbuilt_shapes* shall be as defined in Table H.20.

Table H.20 — interface_asbuilt_shapes fields

Field	Type	Use	Description
interface_type	the value "*iso.asbuilt_shapes*"	Required	Constant value.
shapes	array of *asbuilt_shapes*	Required	A shape for each working tool.

H.4.6.1 Definition "*asbuilt_shape*"

A geometry description of a working tool.

An *asbuilt_shape* shall be as defined in Table H.21.

Table H.21 — asbuilt_shape fields

Field	Type	Use	Description
functions	array of *surface_modelling_function_enum_rts*	Required	The set of surface modelling functions which shall be updated by this as-built shape. "height" comes from point_ref; "pass" (passcount) is implied; others shall be mapped with constants in the vertices.
id	*character_string_255*	Required	Shape ID, used for 3 dot and 4 dot reference scheme.
smc	*surface_modelling_condition_i_enum*	Required	Surface modelling condition, which may be replicated to indicate how the shape shall be applied to a simulated surface.
type	*string*, one of (*line*)	Required	
vertices	array of *asbuilt_vertexs*	Required	2 vertices for type="line".

H.4.6.1.1 Definition "surface_modelling_function_enum_rt"

This enumeration defines how a working tool shape should affect a machine measured surface.

surface_modelling_function_enum_rt is the following enumeration:

— `height`: The working tool shape should affect the height of the surface.

— `pass`: The working tool shape should increment pass counts.

— `temp`: The working tool shape has a temperature value with each vertex.

— `cv`: The working tool shape has a compaction value with each vertex.

H.4.6.1.2 Definition "*surface_modelling_condition_i_enum*"

This enumeration indicates how the SMS should update its current as-built surface model based on received real-time data. Received data should only be applied to the modelled as-built surface when the enumerated condition holds.

surface_modelling_function_i_enum is the following enumeration:

— 0 : Never apply the shape.

— 1 : Always apply the shape.

— 2 : Apply the shape when either lower than existing surface height or no surface height exists.

— 3 : Apply the shape when either higher than existing surface height or no surface height exists.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

— 4 : Apply the shape when surface height exists.

— 5 : Apply the shape when lower than existing surface height.

— 6 : Apply the shape when higher than existing surface height.

H.4.6.2 Definition "*asbuilt_vertex*"

Defines a vertex within an *asbuilt_shape*.

An asbuilt_vertex shall be as defined in Table H.22.

Table H.22 — asbuilt_vertex fields

Field	Type	Use	Description
constants	array of *asbuilt_constants*	Optional	Maps functions to referenced values.
point_ref	string matching the regular expression `^iso\.points_of_interest\.[a-z_]+$`	Required	3 dot object reference to a poi_point.

H.4.6.3 Definition "*asbuilt_constant*"

Maps a 4 dot referenced value to an as-built function such as "temp".

An *asbuilt_constant* shall be as defined in Table H.23.

Table H.23 — asbuilt_constant fields

Field	Type	Use	Description
function	string, one of(*temp, cv*)	Required	As-built function.
value_ref	string matching the regular expression `^[a-z_]+\.[a-z_]+\.[a-z_]+\.[a-z_]+$`	Required	4 dot value reference, which supplies the value for the as-built function at this vertex.

H.4.7 Definition "*interface_replicate*"

H.4.7.1 General

The replicate interface describes how to efficiently replicate the constantly varying parts of a *component* over a network to a remote VIS or SMS. It has a manifest of 4 dot value references from within the *component* that must be replicated to the same *component* in a remote *asset_configuration* of the same UUID.

A real-time implementation by an SMS or VIS shall maintain an in-memory document object model (DOM) for each unique *asset_configuration* which is in operation on a worksite. This DOM shall hold double precision real values for each field which is a target of a 4 dot value reference.

The definition "*interface_replicate*" "manifest" forms a set of value packing and unpacking instructions. Each 4 dot value reference listed in the manifest is taken from the component DOM and packed into a binary buffer to be sent over the network. The same definition "*interface_replicate*" "manifest" is used by the remote side to unpack the binary buffer onto the values in the remote component DOM.

An *interface_replicate* shall be as defined in Table H.24.

Table H.24 — interface_replicate fields

Field	Type	Use	Description
interface_type	the value "*iso.replicate*"	Required	Contant value.
manifest	array of *replicate_manifest_items*	Required	A sequence of 4 dot value references and byte packing parameters.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

H.4.7.2 Definition "*replicate_manifest_item*"

A 4 dot value reference to be byte packed or unpacked.

A *replicate_manifest_item* shall be as defined in Table H.25.

Table H.25 — replicate_manifest_item fields

Field	Type	Use	Description
type	*string*, one of(*float, double, int8, int16, int32)*	Required	The value is cast to and from this type for byte packing.
value_ref	*string* matching the regular expression `^[a-z_]+\.[a-z_]+\.[a-z_]+\.[a-z_]+$`	Required	4 dot value reference, which supplies the value to replicate.

A *replicate_manifest_item* shall also contain attributes as defined in exactly one of the following sub schemata:

a) *replicate_manifest_item_real*

b) *replicate_manifest_item_int*

Subschema replicate_manifest_item_real

A *replicate_manifest_item_real* shall be as defined in Table H.26.

Table H.26 — replicate_manifest_item_real fields

Field	Type	Use	Description
type	*string*, one of*(float, double)*	Required	

Subschema replicate_manifest_item_int

A *replicate_manifest_item_int* shall be as defined in Table H.27.

Table H.27 — replicate_manifest_item_int fields

Field	Type	Use	Description
discrete	*replicate_discrete*	Optional	Parameters required for casting integer types (int8, int16, int32).
type	*string*, one of(*int8, int16, int32*)	Required	

When packing and unpacking, the manifest items shall be processed in their array sequence. If the manifest item is integer type, a quantization occurs according to the *replicate_discrete* parameters stated in the "discrete" field. The values shall be serialized in little-endian.

H.4.7.3 Definition "*replicate_discrete*"

Parameters used to quantize (shift and scale) between referenced value and the integer type. A *replicate_discrete* shall be as defined in Table H.28.

Table H.28 — replicate_discrete fields

Field	Type	Use	Description
discrete_max	*number*	Required	Maximum projects to integer discrete_max.
discrete_min	*number*	Required	Minimum projects to integer discrete_min.
max	*number*	Required	Referenced value is clamped to this max.
min	*number*	Required	Referenced value is clamped to this min.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Pseudo code to pack a value as the stated integer type (int8,int16,int32), where p = the "discrete" object:

```
clampedV = min(max(p.min, value), p.max)
rangeV = p.max - p.min
normV = (clampedV - p.min) / (rangeV)
rangeD = p.discrete_max - p.discrete_min
intV = p.discrete_min + (normV * rangeD)
pack_bytes = little_endian(int_type(intV))
```

Unpacking is the reverse.

H.4.8 Definition "*interface_attach*"

The attach interface is used to connect hierarchies of *components* within an *asset_configuration* together. For example, a bucket node in an implement component can be attached to an excavator stick node in a body component. *interface_attach* shall be stated in the child *component*.

interface_attach depends on *interface_nodes*. Within a component, *interface_attach* is mutually exclusive with *interface_position*.

Multiple *interface_attach* can form a hierarchy more than one level deep (components attached to components attached to components). Accordingly, real-time implementations should build a dependency chain of attachments, and update the transforms of each *interface_nodes* object according to the dependency order inferred from the dependency chain.

An *interface_attach* shall be as defined in Table H.29.

Table H.29 — *interface_attach* fields

Field	Type	Use	Description
interface_type	the value "*iso.attach*"	Required	Constant value.
local_node_ref	string matching the regular expression `^iso\.nodes\.[a-z_]+$`	Optional	A 3 dot object reference to a root *nodes_node* in this component. Matches `^iso\.nodes\.[a- z_]+$`
parent_component_index	*integer*	Required	The index of the parent *component* within the *asset_configuration* components array.
parent_node_ref	string matching the regular expression `^iso\.nodes\.[a-z_]+$`	Required	A 3 dot object reference to a *nodes_node* in the parent *component* at which this component attaches. Matches `^iso\.nodes\.[a- z_]+$`

H.4.9 Definition "*interface_render_assets*"

H.4.9.1 General

The render-assets interface defines 3D-model assets which relate to nodes in the *component* such that the asset can be graphically rendered. The 3D-model assets shall be hosted by a VIS which produces them.

An *interface_render_assets* shall be as defined in Table H.30.

Table H.30 — interface_render_assets fields

Field	Type	Use	Description
assets	array of *render_assets*	Required	A set of URL references to 3D-assets, mapping the assets shapes to *nodes_node*.
interface_type	the value "*iso.render_assets*"	Required	Constant value.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

H.4.9.2 Definition "*render_asset*"

A URL reference to a 3D-asset with a mapping from its shapes to *nodes_node*. A *render_asset* shall be as defined in Table H.31.

Table H.31 — render_asset fields

Field	Type	Use	Description
resource	*render_resource*	Required	URL reference and media type for the 3D-asset.
shapes	array of *render_shapes*	Required	A set of shapes which map from the 3D-assets shapes to *nodes_node*.
view	*string*, one of(*top, front, right, 3d)*	Required	Describes the projection this asset can be useful for. Orthographic (top, front, right) are useful for 2D renderings.

H.4.9.3 Definition "*render_resource*"

Media type and URL.

A *render_resource* shall be as defined in Table H.32.

Table H.32 — render_resource fields

Field	Type	Use	Description
href	*character_string_url*	Required	The hosted asset shall be publically accessible without authorization.
mime_type	*string*, one of (*model/gltf+-json, model/gltf-binary*)	Required	The media type of the referenced 3D-asset.

H.4.9.4 Definition "*render_shape*"

A mapping from a named shape in the 3D-asset, and its target *node_ref* location. A *render_shape* shall be as defined in Table H.33.

Table H.33 — render_shape fields

Field	Type	Use	Description
asset_transform	*character_string_255*	Required	The element name in the scene graph of the referenced 3D-asset which contains the shapes to be rendered under node_ref.
asx	*number*	Required	The authored X-axis scale of the shape as it is in the referenced 3D-asset.
asy	*number*	Required	The authored Y-axis scale of the shape as it is in the referenced 3D-asset.
asz	*number*	Required	The authored Z-axis scale of the shape as it is in the referenced 3D-asset.
id	*character_string_255*	Required	Shape ID, used for 3 dot and 4 dot reference scheme.
node_ref	string matching the regular expression `^iso\.nodes\.[a-z_]+$`	Required	3 dot object reference to a nodes_node. Matches `^iso\.nodes\.[a-z_]+$`
sx	*number*	Required	The measured (desired) X-axis scale of the shape.
sy	*number*	Required	The measured (desired) Y-axis scale of the shape.
sz	*number*	Required	The measured (desired) Z-axis scale of the shape.

The scaling matrix pushed before rendering the shape is (sx/asx, sy/asy, sz/asz).

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

H.4.9.5 Producing a 3D-asset

There are several things to consider when producing a 3D-asset:

— How is the *interface_nodes* scene graph transferred to the 3D-asset scene graph in a 3D rendering library? This is a complex problem, but with a few conventions it is made simple.

— Different applications can desire different projected views of a 3D-asset, such as perspective view, orthographic plan view, orthographic profile view. Different 3D-assets can be required to optimize for different views.

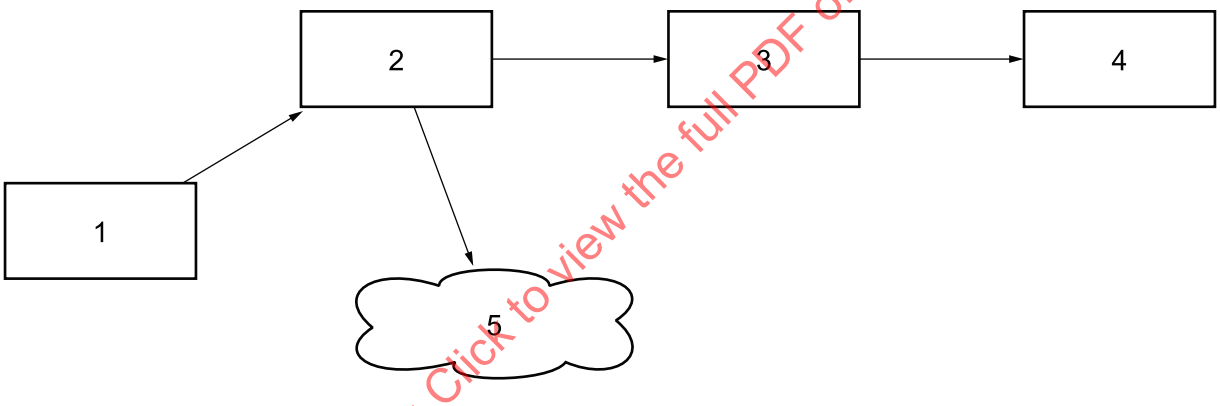
— To achieve interoperability, a VIS shall provide 3d assets in glTF format in conformance with ISO/IEC 12113.

— The 3D-asset interchange format shall be published in model space SAE J670, right handed, Z down with front facing in positive X axis.

— The model space of an applications 3D rendering library will most likely not be the same. A conversion matrix shall be applied by the renderer.

H.4.9.6 Transferring the real-time scene graph transform hierarchy to the render asset hierarchy

The following convention shall be adopted when building *interface_nodes* and *interface_render_assets*, in order to make configuring and rigging 3D-assets as easy as possible. Figure H.1 represents a simple excavator.



Key

1 The “body” *nodes_node* of an excavator.

2 The “boom” *nodes_node*. The translation of the boom is an offset from the body origin to the center of the boom pivot. The length of the boom is along the positive X-axis, and the boom rotates around the Y-axis.

3 The “stick” *nodes_node*. The positive X-axis translation “tx” of the stick is length of the boom. The length of the boom shall not be written into any scale values (e.g. “sx”).

4 The “bucket” *nodes_node*. The positive X-axis translation “tx” of the bucket is the length of the stick. The length of the stick shall not be written into any scale values (e.g. “sx”).

5 The *interface_render_assets* has a “boom” *render_shape*, which has a scale. The length of the boom will be used to calculate this “boom” shape scale.

Figure H.1 — Scene graph convention

A render 3D-asset scale shall not be written into a *nodes_node*. Lengths of booms shall be written into *nodes_node* translation, not scale. For example, the length of a boom is written into the translation of the child stick *nodes_node*. This avoids descendant scaling problems.

The terms required to build a shape scale matrix are provided in each *render_shape*. This scale matrix does not form part of the *interface_nodes* hierarchy. The length of a boom can be safely applied into this shape scale matrix without influencing the scale of child *nodes_node* transforms. The shape scale matrix should be pushed onto the rendering matrix stack before rendering the *render_shape* 3D-asset.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

H.4.9.7 Matrix stack

The matrix stack in a 3D library should be pushed as follows:

a) The projection matrix for orthographic or perspective projection, as determined by the 3D library (such as OpenGL).

b) An optional space conversion matrix, which can transform from application world space to the 3d library space. This is only needed if the application world space is different to the 3d library space. This can be handled by the 3D library. This is mentioned to make apparent the handedness (left handed/right handed) of each of the spaces involved. Any polygon drawn in one handedness shall have its winding changed (or back face culling order flipped) if the final transform is in the other handedness.

c) The view matrix (which is the inverse camera matrix).

d) The asset (machine) matrix. This is a matrix which transforms the asset configuration to its position and rotation, according to localized coordinate convention as it relates to application world space. The *interface_position* interface may be used for this, and may involve WGS-84, ECEF, or some localized coordinate system.

e) A space conversion matrix, which will transform from SAE J670 to application world space.

f) The *nodes_node* matrix hierarchy.

g) The *render_shape* shape scaling matrix, (sx/asx, sy/asy, sz/asz).

H.4.10 Definition "*interface_unknown*"

Matches any other interface, allowing this schema to validate *asset_configurations* with extended interfaces. An *interface_unknown* shall be as defined in Table H.34.

Table H.34 — interface_unknown fields

Field	Type	Use	Description
interface_type	string matching the regular expression `^[a-z_]+\.[a-z_]+$`	Required	Any value matching the pattern that does not equal any of the other interface_type values.

H.5 Type "*replicate*"

H.5.1 General

The *replicate* message is used to communicate the values within an *asset_configuration* which change at high frequency. Accordingly, the replicate message references an *asset_configuration* by UUID.

The *interface_replicate* from each component in the *asset_configuration* is used to pack a byte buffer, which is URL-safe no-padding base64 encoded and added to the "cm" array of the *replicate*. If an *asset_configuration* has two *component*, the "cm" array used to remotely represent it will have two base64 string entries. If one of the components has no *interface_replicate*, then the "cm" array element shall be the JSON null value.

Upon receiving a *replicate* message with "cm" array of base64 encoded strings, the remote side shall decode base64, and de-serialize the bytes using the appropriate *components interface_replicate* "manifest".

A *replicate* shall be as defined in Table H.35.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table H.35 — replicate fields

Field	Type	Use	Description
ac	*character_string_uuid*	Required	Identifies the *asset_configuration* used to encode/decode cm.
cm	array of nullable *character_string_b64s*	Required	The component manifests array contains an entry for each component in the referenced *asset_configuration*. Each entry is a URL-safe no-padding base64 encoding of the component replicate manifest buffer. The length should match the number of components in the *asset_configuration.*
meta	*abprod_rt_metadata* with type=*replicate*	Required	*abprod_rt_metadata* with type=replicate

H.5.2 DOM and static typed languages

In cases where the *asset_configuration* JSON is loaded into static types rather than a DOM, the static types shall hold double precision class members for each possible 4 dot value reference.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex I
(normative)

As-built and production data system-of-record SMS API

I.1 Messages

I.1.1 General

For a worksite, the SMS shall implement a system-of-record as a sliding window of messages. A message issued to the system-of-record shall be persistent and available to a VIS for retrieval for at least one week.

Messages are issued and retrieved using routes as described in this annex. Each route is relative to the `worksiteBaseURL` as discussed in Clause 5.3.

I.1.2 Authorizing

These endpoints shall be accessed with an access token obtained using OAuth2.0 Client Credentials Grant, as detailed in Subclause A.2.4.3. The SMS shall require the scope `iso15143-4.production_data.sor.read` or `iso15143-4.production_data.sor.write` for endpoints which read and write data respectively.

I.1.3 Issuing messages

A VIS shall issue messages to the SMS system-of-record using the route defined in Subclause I.2.1.2.

— A VIS should push messages as soon as it is able to do so (within 5 minutes) after receipt from an asset.

— A VIS shall persist and retry failed posts to ensure reliable message delivery.

— A VIS shall make best efforts to avoid issuing the same message twice.

— A VIS shall push messages according to its relevant access policy.

— A VIS may push messages out-of-order with respect to the “at” time, as there is no hard ordering requirement.

— A VIS may amend a system-of-record message previously issued, and reissue the message with the same *abprod_sor_metadata* “ID” but possibly different message content. A VIS should avoid reissuing messages with any frequency. Each system-of-record type may restrict the kind of allowed amendments. An SMS shall accept reissues that respect these restrictions, including the trivial case where the reissue does not modify the message content.

— A VIS shall ensure the project data assets identified by *abprod_sor_metadata* “asset” have successfully been POSTed to the SMS as specified in Clause F.5.2 prior to issuing *sor_type* messages for the asset.

— The SMS shall track the VIS which owns each asset as specified in Clause F.5.2. The SMS shall reject messages from a VIS where the *abprod_sor_metadata* “asset” is owned by another VIS.

An SRS shall not issue messages.

I.1.4 Retrieving messages

A VIS may subscribe to messages issued to the SMS system-of-record using the route defined in Subclause I.2.1.1. Messages are retrieved from the sliding window using a cursor. The SMS cursor is relative

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

to a position in the sliding window rather than an occurred-at time. Accordingly, message *abprod_sor_metadata* "at" times may be out of order in the response.

— The SMS shall filter the messages sent to a VIS according to relevant access policies.

— The SMS shall not filter messages by the issuer.

— If no cursor is provided in the request, the SMS shall return data from the start of the sliding window.

— If items are available from the given `cursor`, the SMS shall return them without waiting. Otherwise, the SMS shall wait up to `long_poll` seconds for new items to become available. The SMS shall return the new items immediately. If no new items become available during the long poll, the SMS shall return an empty array and `next_cursor` (which may not equal `cursor` in the case of authorization-filtered messages).

— SMS and VIS which process the system-of-record shall treat reissued messages as an UPSERT operation, totally replacing messages seen earlier in the record of the same *abprod_sor_metadata* "id".

An SRS may subscribe to the messages in the same way as a VIS does.

I.2 System-of-record routes

I.2.1 sor routes

I.2.1.1 GET {worksiteBaseURL}/2024/asbuilt/sor

Description

Retrieve messages from the SMS system-of-record sliding window. Refer to Subclause I.1.4 for details.

Parameters:

This route shall support the parameters described in Table I.1.

Table I.1 — GET /asbuilt/sor parameters

Name	Source	Type	Use	Description
cursor	Query	*string* matching the regular expression `^[a- zA-Z0-9_/-]{0,64}$`	Optional	The cursor from when to return system-of-record measurements, as returned by the SMS from a previous call.
max_items	Query	*integer* between 1 and 1000	Optional	The maximum number of items to return, defaults to 1000.
long_poll	Query	*integer* between 0 and 60	Optional	Defines maximum interval (in seconds) to wait before returning data. If zero or missing, all available data (up to max_items) is returned immediately.

Responses

This route shall provide responses as described in Table I.2.

Table I.2 — GET asbuilt/sor responses

Code	Schema	Description
200	*sor_get_body*	Successful operation.
400	N/A	Bad parameter data, including where the cursor is not within sliding window.
404	N/A	Worksite not found.
410	N/A	Gone: the worksite is no longer available at the SMS.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

I.2.1.2 POST {worksiteBaseURL}/2024/asbuilt/sor

Description

Issue messages to the SMS system-of-record sliding window. Refer to Subclause I.1.3 for details.

Parameters

This route shall support the parameters described in Table I.3.

Table I.3 — POST asbuilt/sor parameters

Name	Source	Type	Use	Description
body	body	*sor_post_body*	Required	An object containing an array of system-of-record messages. The messages are from assets connected to the VIS. The SMS shall append to the system-of-record in the order messages appear in the array.

Responses

This route shall provide responses as described in Table I.4.

Table I.4 — POST asbuilt/sor responses

Code	Schema	Description
201	*sor_response_body*	Successful operation, the SMS has the data and will not lose it.
400	N/A	Bad data, bad JSON or bad schema.
401	N/A	The VIS is not authenticated correctly, or this operation is not permitted by the policy in place between VIS and SMS.
403	N/A	A forbidden operation. An SMS may reject data that specifies assets not belonging to a VIS.
404	N/A	Worksite not found.
409	N/A	Conflict, asset not stated or amendment not allowed.
410	N/A	Gone: the project is no longer available at the SMS.

I.3 System-of-record definitions

I.3.1 *sor_get_body*

Properties

A *sor_get_body* shall have the properties defined in Table I.5.

Table I.5 — sor_get_body properties

Name	Type	Use	Description
items	array of *sor_types*	Required	An array of *sor_type.*
next_cursor	*string* – matching the regular expression `^[a-zA-Z0-9/-]{0,64}$`	Required	An opaque cursor, at most 64 characters made out of the regular expression `^[a-zA-Z0-9_/-]{0,64}$`

I.3.2 *sor_post_body*

Properties

A *sor_post_body* shall have the properties defined in Table I.6.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table I.6 — sor_post_body properties

Name	Type	Use	Description
items	Array of *sor_types*	Required	An array of *sor_type.*

I.3.3 *sor_response_body*

The response object currently has no fields.

I.3.4 *sor_type*

One of the system-of-record types as defined in Annex G.

A *sor_type* object shall have contents as defined in exactly one of the types:

— *streak*

— *tin*

— *survey*

— *ref_survey*

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex J
(normative)

As-built and production data real-time SMS API

J.1 Messages

J.1.1 General

For a worksite, the SMS shall implement a message broker in a pub-sub manner. Real-time messages published by a VIS to the SMS message broker are repeated to other VIS which are subscribed to the message broker. Each worksite is treated as a channel on the message broker, such that messages published to one worksite shall not be repeated to VIS subscribers on other worksites.

The SMS shall use websockets in conformance with IETF RFC 6455 to implement the API. The SMS shall support WebSocket upgrade over HTTP1.1 secured by TLS. The SMS may support WebSocket over HTTP2.0 in conformance with IETF RFC 8441. VIS agents may attempt SMS WebSocket connections over HTTP2.0 in conformance with IETF RFC 8441. And in case of failure, VIS agents shall fall back to WebSocket upgrade over HTTP1.1 secured by TLS. The SMS and VIS shall support WebSocket Compression in conformance with IETF RFC 7692. The SMS and VIS shall use context takeover to maximize stream compression in conformance with IETF RFC 7692:2015, Subclause 7.1.1. The websocket established between VIS and SMS should use context takeover in conformance with IETF RFC 7692:2015, Subclause 7.1.1, such that the lz77 sliding window is shared for subsequent messages.

Each route described in this annex is relative to the `worksiteBaseURL` as discussed in Subclause 5.3.

J.1.2 Authorizing

These endpoints shall be accessed with an access token obtained using OAuth2.0 Client Credentials Grant, as detailed in Subclause A.2.4.3. The SMS shall require the scope `iso15143-4.production_data.rt.read` or `iso15143-4.production_data.rt.write` for endpoints which read and write data respectively.

J.1.3 Publishing real-time messages

A VIS should publish real-time messages to the SMS message broker using the route defined in Subclause J.2.1.1.

— A VIS shall write JSON text messages on the websocket, of type rt_type.

— If a VIS does not collect data in real time from its connected field equipment, it should not use this publish mechanism.

— A VIS should only publish messages for assets which adhere to the relevant access policy.

— A VIS should close the websocket if there has been a prolonged period of asset inactivity.

— The SMS may close the websocket if there has been a prolonged period of inactivity.

— A VIS should create an agent which connects the websocket to publish a stream of rt_type in the case where there is real time machine activity for a worksite, and no agent is already running for that worksite.

— Where multiple asset objects exist for a single physical multi-type asset (as specified in Subclause E.3.5), a VIS shall publish data for at most one of the asset objects at any one time.

— The SMS shall track the VIS which owns each asset per Subclause F.5.2. The SMS shall reject messages from a VIS where the abprod_rt_metadata “asset” is owned by another VIS.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

An SRS shall not publish real-time messages.

J.1.4 Subscribing to real-time messages

A VIS may subscribe to real-time messages from the SMS message broker using the route defined in Subclause J.2.2.1.

— The SMS shall write JSON text messages on the websocket, of type *rt_type*.

— The SMS should only write messages to subscribed VIS for assets which adhere to the relevant access policy.

— The SMS shall maintain a copy of the state of all assets and upon subscription of a VIS shall communicate the set of all states which adhere to the relevant access policy.

— The level of confidence that a VIS has in the accuracy of another asset's state decreases as the state message age increases (as measured by the *abprod_rt_metadata* "at" field). Since an asset's state is expected to be updated frequently, it is likely that a VIS using a state older than 15 seconds has an inaccurate view of that asset's actual state.

— A VIS should only subscribe to the real-time message stream when it needs to relay information to downstream connected field equipment. Accordingly, a VIS should not have agents continually subscribed.

An SRS shall not subscribe to the real-time stream.

J.2 Real time routes

J.2.1 rtpub routes

J.2.1.1 GET {worksiteBaseURL}/2024/asbuilt/rtpub

Description

Publish messages to the SMS message broker. Refer to Subclause J.1.3 for details.

Responses:

This route shall provide responses as described in Table J.1.

Table J.1 — GET asbuilt/rtpub responses

Code	Schema	Description
101	N/A	Successful protocol switch for web socket upgrade.
400	N/A	Bad parameter data.
401	N/A	Unauthorized.
410	N/A	Gone: the project is no longer available at the SMS.

J.2.2 rtsub routes

J.2.2.1 GET {worksiteBaseURL}/2024/asbuilt/rtsub

Description

Subscribe to real-time messages from the SMS message broker. Refer to Subclause J.1.4 for details.

Responses:

This route shall provide responses as described in Table J.2.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table J.2 — GET asbuilt/rtsub responses

Code	Schema	Description
101	N/A	Successful protocol switch for web socket upgrade.
400	N/A	Bad parameter data.
401	N/A	Unauthorized.
410	N/A	Gone: the project is no longer available at the SMS.

J.3 Real time definitions

J.3.1 *rt_type*

One of the real-time types as defined in Annex H.

A *rt_type* object shall have contents as defined in exactly one of the types:

— *state*

— *asset_configuration*

— *replicate*

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex K
(normative)

Localization

K.1 Overview

The localization process is used to transform WGS-84 positions into local NEE coordinates (orthometric heights). It can be used in reverse to transform site coordinates into WGS-84.

SMSs shall implement the API for providing this information to field systems via a VIS. Field systems should use the localization when provided by the SMS. The data provided shall conform to the schemata presented in Subclause E.3.3 and Subclause E.3.5.

This document does not try to represent every coordinate system transformation that is in use. It aims to cover the most common transformations that are used in 95 % of coordinate transformations today, without the complex or rare systems.

Field systems are not required to use the localization system. For sites that use a coordinate system that is not supported then each of the field systems is responsible for being able to work in a representation of the site coordinate system.

The localization system requires that the WGS-84 positions are homogeneous. For differential solutions, such as RTK, if multiple base stations are used, the base station coordinates shall be consistent.

On sites where a mixture of techniques, such as precise point positioning (PPP) and differential methods (e.g. RTK) is being used, the user is required to make the coordinates of the positioning methods consistent.

Time dependent coordinate transformations are not supported.

K.2 Units

All values shall be recorded in SI units.

— Distance values shall be recorded in meters.

— Angle shall be recorded in decimal degrees.

  — Angles are clockwise from 0.

  — North is 0 degrees.

— Latitude values shall be recorded in decimal degrees, with northern latitudes positive.

— Longitude values shall be recorded in decimal degrees, with eastern longitudes positive.

K.3 Localization process

The localization process consists of a number of steps, most of which are optional. They are applied in the following order.

a) Datum transformation, which may include a change of ellipsoid to a local one.

b) Projection.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

c) Horizontal adjustment.

d) Vertical adjustment.

The projection is the only required element in the localization transformation set.

K.4 Localization object

K.4.1 General

The localization object contains subobjects for each of the transformation steps.

In the cases that the coordinate system cannot be represented by this document, no localization object shall be provided.

K.4.2 Datum transformation (*datum_transformation*)

K.4.2.1 General

The datum transformation is a parametric datum that converts from WGS-84 positions into the local datum.

The transformation from the global to the local geodetic datum is generally accomplished by means of a 7-parameter transformation. The ISO 19111 specifies the linearized, or approximate, transformation formula. The parameters found in many publications as well as databases like the EPSG or European initiatives, and all the IGS and IERS parameters correspond to this formula.

ISO 19111 also defines the strict formula of a 7-parameter transformation, this calculation method is not used in this standard.

Only datum grid, 3 and 7 parameter transformations are supported.

When a datum transformation is not provided, WGS-84 should be used for the datum.

The current WGS-84 ellipsoid should be used.

At time of publication, the WGS-84 values are:

— semi major axis = 6378137;

— semi minor axis = 6356752.31424518;

— inverse flattening = 298.257223563.

K.4.2.2 Datum sign convention

The sign convention is such that a positive rotation of the frame about an axis is defined as a clockwise rotation of the coordinate reference frame when viewed from the origin of the Cartesian coordinate reference system in the positive direction of that axis. That is, a positive rotation about the Z-axis only from source coordinate reference system to target coordinate reference system, will result in a smaller longitude value for the point in the target coordinate reference system.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

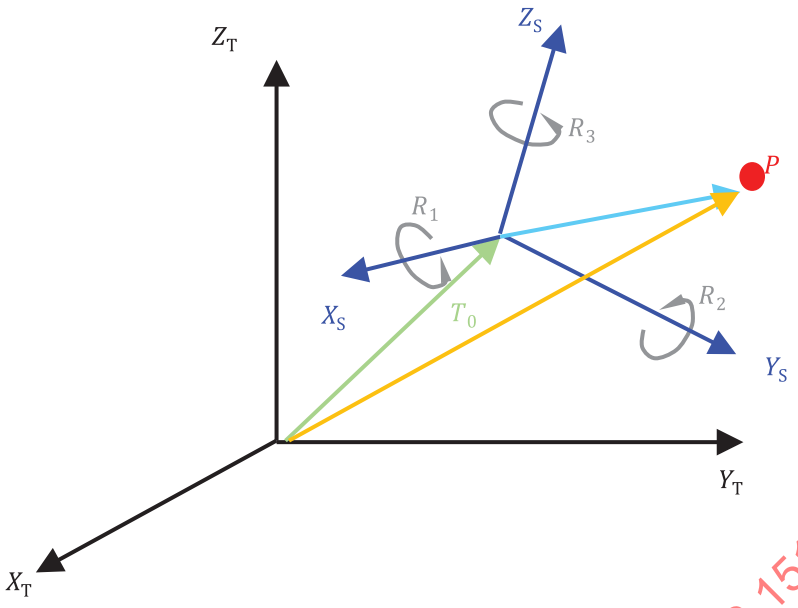


Figure K.1 — Datum sign convention

K.4.3 Projection (*projection*)

The *projection* object is required.

The *projection* will have one object based on the type of the projection. Field implementations are required to support the most common projections: *transverse_mercator, mercator, lambert_1, lambert_2, oblique_mercator, oblique_stereographic*, and *albers*.

K.4.4 Horizontal adjustment (*horizontal_adjustment*)

K.4.4.1 General

The horizontal adjustment is used to adjust the NEE coordinates that have been produced by the datum and projection. This also allows for local coordinate adjustment on published coordinates systems.

The *horizontal_adjustment* object is optional.

K.4.4.2 Order of application

a) Rotation.

b) Scale factor

c) Translation.

The origin in the adjustment is where the rotation and scale factor is applied from.

Rotation angle is such that a positive rotation is clockwise rotation.

The application of the scale factor is:

```
Adjusted Northing = (N - origin_north) * scale + origin_north
Adjusted Easting = (E - origin_east) * scale + origin_east
```

The application of the translation is:

```
Adjusted Northing = N + translation_north
Adjusted Easting = E + translation_east
```

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

K.4.5 Vertical adjustment (*vertical_adjustment*)

K.4.5.1 General

The vertical adjustment is used to provide a transformation from ellipsoid to orthometric heights. The *vertical_adjustment* object is optional.

The object may contain zero, one or both of:

— an *inclined_plane*

— a *geoid_model*

When there are 0 objects in the *vertical_adjustment* record, this indicates that ellipsoid heights and orthometric heights are the same. This is equivalent to an *inclined_plane* with all 0 values.

When both are provided, the geoid model should be applied first.

K.4.5.2 *inclined_plane* application

Origin is the location where the slope correction starts from.

The inclined plane correction is:

```
Inclined Plane Adjustment =
   (N - origin_north) * slope_north + (E - origin_east) * slope_east + constant_adjustment
```

The elevation of the point is:

```
Adjusted Elevation = Ellipsoid Height + Geoid Height Adjustment + Inclined Plane Adjustment
```

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Annex L
(normative)

ISO LandXML

L.1 Background

A software application implementing a BIM file format standard is generally required to support a well-defined subset of the data schema and referenced data.

ISO LandXML specification defines the workflow and mandatory model data exchange on infrastructure construction projects.

LandXML is a specialized XML data file format containing civil engineering and survey measurement data commonly used in the land development and transportation industries.

It was developed by an open community of volunteer organizations and individuals over almost a decade, until the schema version 1.2 was ratified and standardized on August 15, 2008 and published on the LandXML.org website.

LandXML 1.2 can support several data exchange scenarios in infrastructure construction and therefore can provide a practical short-term solution, if its use is well-defined and proper documentation (that is, a user-manual) is provided.

This also provides a fast-track solution for software vendors supporting LandXML1.2 on their existing products.

The ISO LandXML documentation is comprised of this document, accompanied by the following support material:

— ISO LandXML schema.

  — LandXML v1.2 schema with specific restrictions, as xsd. This is for validation purposes only. See Annex N for file location.

— Mathematical formulas of transition curves (source:landxml.org).

ISO LandXML is based on Finnish InfraModel4 (version 4.0.4) but defines an even more reduced subset of LandXML 1.2 functionality to fit with the data exchange requirements of this document.

Files following this specification are fully LandXML1.2 compatible, but not all LandXML 1.2 files are compatible with this specification due to the restricted subset of elements and constraints.

This document only covers final design to grade control data exchange scenario. Other data exchange scenarios, such as design to design or construction to asset information management are not included in this document.

This document does not limit aggregation of all design objects into a single file. In some cases, users can choose to group related objects in a single file, especially when objects are referenced to each other. However, for practical work, it is a good idea to split the large design data set into multiple reasonable sized files. This enables fluent management of work orders for later work planning for several reasons. For example, increased flexibility in managing data when designs are changing and updated, better data security among many sub-contractors in one project and less data for each field system and operator to handle.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

L.2 Fundamental concepts and assumptions

The LandXML 1.2 schema consists of specification of data types for civil engineering and survey measurement data. For practical application in civil and infrastructure design and construction scenarios common engineering concepts must be defined, indicating use of data types for particular scenarios. The following chapters define such common concepts, which are applied at elements having specific use. Such concepts also form the basis of this annex, detailed specifications that adapt the scope and rules of the LandXML1.2 schema for targeted applications within the civil and infrastructure industry.

Each concept defines a set of elements and attributes, with constraints and parameters set for particular attributes and instance types.

L.3 File headers

L.3.1 General

The following items are defined in the header:

— The root element of the file (<*LandXML*>)

  — Units of measurement (<*Units*>)

  — Worksite information (<*Project*>)

  — The feature extension used by this specification (<*FeatureDictionary*>)

  — Application information (<*Application*>)

  — Author information (<*Author*>)

L.3.2 XML file container

XML files shall use UTF-8 encoding, and encoding attribute shall be set on XML header.

The root element (<*LandXML*>) of the transfer file is used by software to check the validity of the file structure.

A <*LandXML*> shall be as defined in Table L.1.

Table L.1 — <LandXML> fields

Field	Type	Use	Description
One <*Units*> element	<*Units*>	Required	Worksite units.
Zero or one <*Project*> element	<*Project*>	Optional	Identifies the worksite.
One <*Application*> element	<*Application*>	Required	Identifies the application which has produced the file.
One <*FeatureDictionary*> element	<*FeatureDictionary*>	Required	Describes the feature extensions (ISO15143-4) used by the feature elements of design objects.
Zero or more <*Surfaces*> elements	<*Surfaces*>	Optional	A collection of surface models.
Zero or more <*Alignments*> elements	<*Alignments*>	Optional	A collection of alignments.
Zero or more <*CgPoints*> elements	<*CgPoints*>	Optional	A collection of COGO points. (Cg = COGO = Coordinate Geometry).
Zero or more <*Parcels*> elements	<*Parcels*>	Optional	A collection of 2D boundaries.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table L.1 *(continued)*

Field	Type	Use	Description
Zero or more *<PlanFeatures>* elements	*<PlanFeatures>*	Optional	A collection of planimetric features not otherwise defined by the schema, such as building footprints, guard rails, tree lines, light poles or signage. Typically, a *PlanFeatures* element will contain a collection of similar items.
Zero or more *<Roadways>* elements	*<Roadways>*	Optional	A collection of *PlanFeatures* which belong to specific route.
Zero or more *<PipeNetworks>* elements	*<PipeNetworks>*	Optional	A collection of pipe networks.
date	*date*	Required	UTC date.
time	*time*	Required	UTC time.
version	*string*	Required	LandXML version 1.2.

L.3.3 Units

The units used in the file are defined by the *<Units>* element. Only certain metric SI system units are allowed, and those are defined under the sub-element *<Units><Metric>*. The following table lists valid units.

Radians are used as default direction (directionUnit) and angular (angularUnit) units. These are defined counter-clockwise from the base direction. In angular definitions the base direction is east, and in direction definitions it is north.

A *<Metric>* shall be as defined in Table L.2.

Table L.2 — <Metric> fields

Field	Type	Use	Description
areaUnit	*metArea*	Required	Fixed value:squareMeter
linearUnit	*metLinear*	Required	Fixed value:meter
volumeUnit	*metVolume*	Required	Fixed value:cubicMeter
temperatureUnit	*metTemperature*	Required	Fixed value:celsius
pressureUnit	*metPressure*	Required	Fixed value:milliBars
diameterUnit	*metDiameter*	Required	Fixed value:meter
widthUnit	*metWidth*	Required	Fixed value:meter
heightUnit	*metHeight*	Required	Fixed value:meter
elevationUnit	*elevationType*	Required	Fixed value:meter
angularUnit	*angularType*	Required	Fixed value:radians
directionUnit	*angularType*	Required	Fixed value:radians
velocityUnit	*metVelocity*	Required	Fixed value:kilometersPerHour

L.3.4 *Project*

Optional worksite information element.

Name is only a required attribute if the element is present. The description can contain, for example, the project long name or code, but this is optional. The optional state attribute can be used to describe the state of the project and its content.

A *<Project>* shall be as defined in Table L.3.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table L.3 — <Project> fields

Field	Type	Use	Description
name	*character_string_id*	Required	Worksite identifier.
desc	*character_string_255*	Optional	Human readable name of worksite.
state	*stateType*	Optional	State of worksite.

A *stateType* is defined as per Table L.4.

Table L.4 — <stateType> definition

Type	Basetype	Restriction	Description
stateType	*string*	One of (abandoned, destroyed, existing, proposed)	Object state.

L.3.5 *Application*

The mandatory *<Application>* element describes the software which creates the file.

An *<Application>* shall be as defined in Table L.5.

Table L.5 — <Application> fields

Field	Type	Use	Description
One *<Author>* element	*<Author>*	Required	
name	*character_string_255*	Required	Application name.
version	*character_string_255*	Required	Application version.
desc	*character_string_255*	Optional	Application description.
manufacturer	*character_string_255*	Optional	Software vendor name.
manufacturerURL	*character_string_255*	Optional	Software homepage url.
timeStamp	*datetime*	Optional	UTC timestamp of last save.

L.3.6 *Authors*

Information of the author of the file is recorded in the optional sub-element *<Application><Author>*

An *<Author>* shall be as defined in Table L.6.

Table L.6 — <Author> fields

Field	Type	Use	Description
createdBy	*character_string_255*	Optional	Author name.
createdByEmail	*character_string_255*	Optional	Author e-mail.
company	*character_string_255*	Optional	Company name.
companyURL	*character_string_255*	Optional	Company homepage URL.
timeStamp	*datetime*	Optional	UTC timestamp of last save.

L.3.7 Feature dictionary

The *<FeatureDictionary>* identifies the specification source of extensions used in the file, and the point of access to their documentation.

The contents of *<Feature>* elements shall follow the source specification. LandXML 1.2 files in general can contain extensions from several different sources.

In ISO LandXML file transfer, proper recognition and interpretation is required only for the extensions documented in this specification (e.g. for the type coding systems used in such file). The dictionary for ISO

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

LandXML extensions shall be specified using the *<FeatureDictionary>* element as defined in Table L.7. The name is fixed to "ISO15143-4" for the extensions defined in this specification, and exactly the same name shall be set in every *<Feature>* for the optional source attribute (the *<Feature>* attribute code being labelled with ISO15143-4_ -prefix). The <version> shall match with the version number of ISO LandXML verification schema.

A *<FeatureDictionary>* shall be as defined in Table L.7.

Table L.7 — <FeatureDictionary> fields

Field	Type	Use	Description
name	*string*	Required	Fixed feature dictionary name (ISO15143-4).
version	*character_string_255*	Optional	Version number.

Proprietary extensions can be used in addition to ones specified in this specification's feature dictionary. Each source of such extension specifications shall be identified as their own *<FeatureDictionary>* instance, with unique name. That name shall be set as the value of attribute source in every *<Feature>* according to that dictionary. Proprietary *<Feature>* instances should not have attribute code labelled with the ISO15143-4_ prefix.

L.4 Taxonomy

L.4.1 General

Identification of design objects is based on taxonomy where each object belongs to an object category and has feature code identifying its type in a machine readable manner.

This document ensures machine readability by harmonizing use of feature codes and categories.

L.4.2 Object category

Object categories are used for grouping feature codes. Different structural surfaces have their dedicated category codes. All objects in the group, for example breaklines of the structure, have their own individual feature codes. Object categories follow single code list inside one worksite. Hence, multiple design models can represent parts of the same structural layer. When as-built is stored, it is identified based on object category and can be used for tracking progress and quality of the structure in machine readable manner independently of design model. Category code can be lead from layer structure of owner's CAD data if such system does not exist.

L.4.3 Feature code

Individual feature codes are set for:

— Data point groups, in the *<DataPoints>* element

— Breaklines in the *<BreakLine>* element

— Surfaces, in the *<Surfaces>* or *<Surface>* element

— Alignments, in the *<Alignments>* or *<Alignment>* element

— Points, in the *<CgPoints>* or *<CgPoint>* element

— String line layers, in the "ISO15143-4_StringlineLayer(s)" *<Feature>* extension element

— Other infrastructures in appropriate elements, such as *<PlanFeature>*, *<PipeNetwork>*, *<Pipe>* or *<Struct>*

Object categories and feature codes are set in the child elements, or in the parent element whose children inherit the values. However, if value is set on both levels, child value overrides parent value.

Object categories and feature codes are set using the ISO15143-4_Taxonomy *<Feature>* extension see Subclause L.15.1.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

L.4.4 Code list

Code list supporting taxonomy is part of the ISO/TS 15143-4 (this document) project data definition. The definition is available in Annex M. The code list enables following features for object categories and feature codes:

— Human readable name of category or feature.

— Visualization parameters (ie. color, linestyle).

L.5 Surface mesh

L.5.1 General

Surface mesh is a triangulated representation of the topography, which may also contain the original source data used for the triangulation process (as datapoints and boundaries). Breaklines are not typically used to form surfaces, but if transferred under *<Surface>* as source data, they shall coincide with the triangulation defining the surface, i.e. each pair of consecutive breakline points given in 3D coordinates, shall match the coordinates of two vertices of a triangle.

Surface meshes are used for soil topmost surface and ground layer models (top surface of each layer) as described in L.5. They are also used for structural models of roads, streets, railways, waterways and areal structures, as described in Clauses 4 to 8.

Surfaces and source data is described as surface groups *<Surfaces>*, which are made of individual *<Surface>* elements. The name of every surface group and the name of each surface within the file are unique.

A *<Surfaces>* shall be as defined in Table L.8.

Table L.8 — <Surfaces> fields

Field	Type	Use	Description
At least one *<Surface>* element	*<Surface>*	Required	Individual surface model.
Zero or one ISO15143-4_Taxonomy *<Feature>* element	ISO15143-4_Taxonomy *<Feature>*	Optional	ISO15143-4_Taxonomy Feature extension.
desc	*character_string_255*	Optional	Human readable name.
state	*stateType*	Optional	Surface group state. Child “Surface” elements inherit the state if set here.

A *<Surface>* shall be as defined in Table L.9.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table L.9 — <Surface> fields

Field	Type	Use	Description
Zero or one *<SourceData>* element	*<SourceData>*	Optional	The collection of data that was used to define the surface.
Zero or one *<Definition>* element	*<Definition>*	Optional	The collection of faces and points that defined the surface.
Zero or more ISO15143-4_Taxonomy *<Feature>* elements	ISO15143-4_Taxonomy *<Feature>*	Optional	ISO15143-4_Taxonomy Feature extension.
name	*character_string_id*	Required	Unique identifier.
desc	*character_string_255*	Required	Human readable name.
state	*stateType*	Optional	Surface state, overrides parent setting if set.

L.5.2 Current and planned surfaces

An existing surface is defined by setting the state of the *<Surfaces>* or *<Surface>* element to "existing". If it is proposed, the state is set to "proposed". If all the surfaces within a surface group have the same state, it is possible to set the state on a higher level in the surface group *<Surfaces>*. The state for an individual surface is set if the constituent triangle meshes, data points and breaklines are in the same state.

L.5.3 Taxonomy of surface meshes

Object category and feature code shall be set for following elements:

— In the element *<DataPoints>* for source data points.

— In the element *<BreakLine>* for breaklines.

— In the element *<Surface>* for single surface.

— In the element *<Surfaces>* for surfaces group.

L.5.4 Source data

L.5.4.1 General

The source data is described by the element *<SourceData>*. Refer to Figure L.1. This element has no attributes. Source data consists of:

— Source data points *<DataPoints>*

— Breaklines *<BreakLines>*

— Boundaries *<Boundary>*

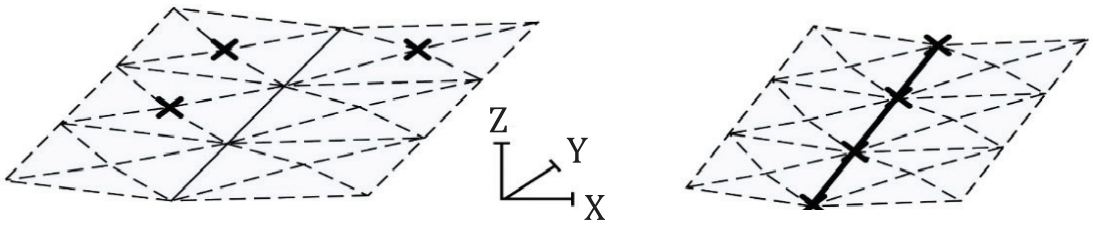


Figure L.1 — Surface source data

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

L.5.4.2 Data points

Source data points are described by the element , sorting every point group into individual elements.

A <*DataPoints*> shall be as defined in Table L.10.

Table L.10 — <DataPoints> fields

Field	Type	Use	Description
At least one <*PntList3D*> element	<*PntList3D*>	Required	A sequential space delimited list of 3D coordinates. Primarily used to store lists of NEE for terrain surface data.
Zero or one ISO15143-4_ Taxonomy <*Feature*> element	ISO15143-4_Taxonomy <*Feature*>	Optional	ISO15143-4_Taxonomy Feature extension.
name	*character_string_id*	Optional	Unique identifier of datapoint group.
desc	*character_string_255*	Optional	Human readable name.
state	*stateType*	Optional	State of datapoints.

The point group is classified by ISO15143-4_Taxonomy feature. The individual data points in the <*DataPoints*> point group are presented as a 3D coordinate list in the <*PntList3D*> element, values separated by spaces.

<*PntList3D*>northing1 easting1 elevation1 northing2 easting2 elevation2... </PntList3D>

L.5.4.3 *Breaklines*

In the breakline group <*BreakLines*>, where each <*BreakLine*> is presented in its own element. The breakline group has no attributes.

A <*Breakline*> shall be as defined in Table L.11.

Table L.11 — <Breakline> fields

Field	Type	Use	Description
One <*PntList3D*> element	<*PntList3D*>	Required	A sequential space delimited list of 3D coordinates. Primarily used to store lists of NEE for terrain surface data.
Zero or one ISO15143-4_ Taxonomy <*Feature*> element	ISO15143-4_Taxonomy <*Feature*>	Optional	ISO15143-4_Taxonomy Feature extension.
name	*character_string_id*	Required	Unique identifier.
desc	*character_string_255*	Optional	Human readable name.
brkType	*breakLineType*	Optional	Breakline type.
state	*stateType*	Optional	Breakline state.

In <*BreakLine*> the breakline type *brkType* defines the use in software. A *breakLineType* is defined per Table L.12.

Table L.12 — <breakLineType> definition

Type	Basetype	Restriction	Description
breakLineType	string	One of (standard, wall, proximity, nondestructive)	Breakline type enum

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Breaklines are classified by ISO15143-4_Taxonomy and they can be given individual names. The constituent points of a <*BreakLine*> are described as a 3D coordinate list in the <*PntList3D*> element, values separated by spaces.

<*PntList3D*>northing1 easting1 elevation1 northing2 easting2 elevation2../<*PntList3D*>

L.5.4.4 *Boundaries*

Additionally, it is also possible to define boundaries of the mesh source data in the optional boundary group <*Boundaries*>, where each <*Boundary*> is presented in its own element. The boundary group has no attributes. In <*Boundary*> the mandatory attributes boundary type *bndType* and edge trim *edgeTrim* define the use in software.

A <*Boundary*> shall be as defined in Table L.13.

Table L.13 — <Boundary> fields

Field	Type	Use	Description
One <*PntList3D*> element	<*PntList3D*>	Required	A sequential space delimited list of 3D coordinates. Primarily used to store lists of NEE for terrain surface data.
Zero or one ISO15143-4_Taxonomy <*Feature*> element	ISO15143-4_Taxonomy <*Feature*>	Optional	ISO15143-4_Taxonomy Feature extension.
name	*character_string_id*	Required	Unique identifier.
bndType	*surfBndType*	Required	Boundary type. One of (outer, void, island).
edgeTrim	*boolean*	Required	Edge trimming.
desc	*character_string_255*	Optional	Human readable name of boundary.
area	*double*	Optional	Area of boundary.
state	*stateType*	Optional	State of boundary.

A *surfBndType* is defined per Table L.14.

Table L.14 — <surfBndType> definition

Type	Basetype	Restriction	Description
surfBndType	*string*	One of (outer, void, island)	Surface boundaries can be one of three types: outer, void, island.

The points of a <*Boundary*> are described as a 2D coordinate list in the <*PntList2D*> element, values separated by spaces <*PntList2D*>northing1 easting1 northing2 easting2 . . . </*PntList2D*>

L.5.5 Triangulated mesh surface

L.5.5.1 General

Surface geometry is described as triangulated meshes, as shown in Figure L.2. Each surface is defined under the <*Definition*> in terms of boundaries, exterior features and holes. A triangular mesh is defined in two steps; first by defining the vertices of the triangular faces as surface points, and then each individual face by three vertices. The surface points used as vertices are assigned unique identifier IDs within the same surface definition (<*Surface*><*Definition*>) element. The face definitions are done by referring to the ID numbers of the vertex points. Additionally, the surface is given a type code according to the declared ISO15143-4_Taxonomy.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

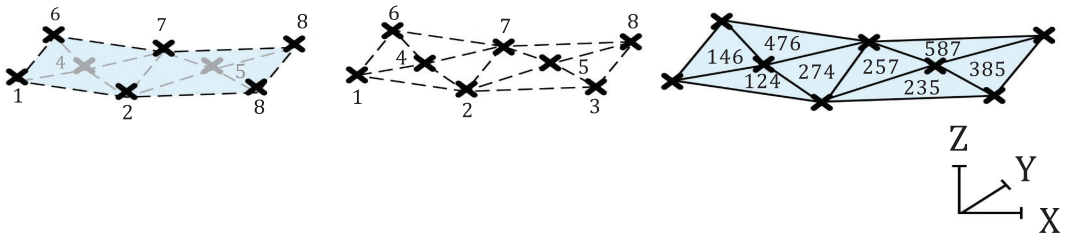


Figure L.2 — Triangulated mesh surface

It is mandatory to set *surfType* to "TIN" when describing a triangular mesh. The precision of the mesh model depends on the available software and data. It is possible to optionally describe a 2D surface area area2DSurf, 3D surface area area3DSurf and the maximum elevation elevMax and minimum elevation elevMin.

A <*Definition*> shall be as defined in Table L.15.

Table L.15 — <Definition> fields

Field	Type	Use	Description
One <*Pnts*> element	<*Pnts*>	Required	The collection of points that define the triangle edge points. The ID values are referenced by the surface faces and breaklines.
At least one <*Faces*> element	<*Faces*>	Required	The collection of faces that define the surface. The faces are defined by 3 (TIN) points, as indicated by the fixed "*surfType*" attribute.
surfType	*surfTypeEnum*	Required	Surface type, fixed value = TIN
area2DSurf	*double*	Optional	2D area of surface.
area3DSurf	*double*	Optional	3D area of surface.
elevMax	*double*	Optional	Maximum elevation of surface.
elevMin	*double*	Optional	Minimum elevation of surface.

A *surfTypeEnum* is defined per Table L.16.

Table L.16 — <surfTypeEnum> definition

Type	Basetype	Restriction	Description
surfTypeEnum	*string*	One of (TIN)	TIN is the acronym for "Triangulated irregular network", a surface comprised of 3 point faces.

L.5.5.2 Vertices

The vertex point group <*Pnts*> contains a listing of individual vertices <*P*>, which are each assigned an individual number ID. These numbers are referenced in the triangulation.

A <*Pnts*> shall be as defined in Table L.17.

Table L.17 — <Pnts> fields

Field	Type	Use	Description
At least 3 <*P*> elements	<*P*>	Required	Vertex point. The ID values are referenced by the surface faces for the coordinate values.

A <*P*> shall be as defined in Table L.18.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table L.18 — <P> fields

Field	Type	Use	Description
content	3 of type *Point*	Required	3D point.
name	*character_string_id*	Optional	Unique identifier.
desc	*character_string_255*	Optional	Human readable name.
state	*stateType*	Optional	State enum.
code	*character_string_255*	Optional	Feature code.
id	*unsigned integer*	Required	Point identifier, which is used to link datapoint by “F” element when defining a triangle face.

A Point is defined as per Table L.19.

Table L.19 — <Point> definition

Field	Type	Use	Description
Point	Whitespace delimited list of *doubles*	Required	A text value that is a space delimited list of doubles. It is used as the base type to define point coordinates in the form of “northing easting” or “NEE” as well as point lists of 2D or 3D points with items such as surface boundaries or “station elevation”, “station offset” lists for items such as profiles and cross sections.

The vertices of a surface definition, minimum three of them, are given as <*P*> elements, where the 3D coordinates are separated by spaces.

<P id=“1”>northing1 easting1 elevation1</P>

<P id=“2”>northing2 easting2 elevation2</P>

<P id=“3”>northing3 easting3 elevation3</P>

L.5.5.3 Faces

The triangulation is defined in the <*Faces*> collection. It consists of consecutive list of faces <*F*>. The order of the faces implicitly defines the index number of each triangle (1,2,. . .). Each face is defined by referencing three vertex ID numbers. Triangle normal direction is defined with the winding order of face vertices. Clockwise winding order defines a front facing triangle as shown in Figure L.3.

A <*Faces*> shall be as defined in Table L.20.

Table L.20 — <Faces> fields

Field	Type	Use	Description
At least one <*F*> elements	<*F*>	Required	Individual triangle face. The 3 space separated numbers represent the vertices on the face. Each number is a reference to the “ID” attribute of a surface point “P”.

A <*F*> shall be as defined in Table L.21.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table L.21 — <F> fields

Field	Type	Use	Description
content	Whitespace delimited list of *integers*	Required	Face definition.
i	*integer*	Optional	Invalid triangle flag. If i=1, the face should be hidden.
n	*FaceType*	Optional	Surrounding faces defined as integer numbers. if there is a face adjacent to the face described.
b	*unsigned integer*	Optional	Coincidence with breakline. Sum value indicating which edges touch a breakline(value 0-7), where edge 1 =1, edge 2 = 2, and edge 3 = 4. For example, if b = 4 this indicates that only edge 3 touches breakline.

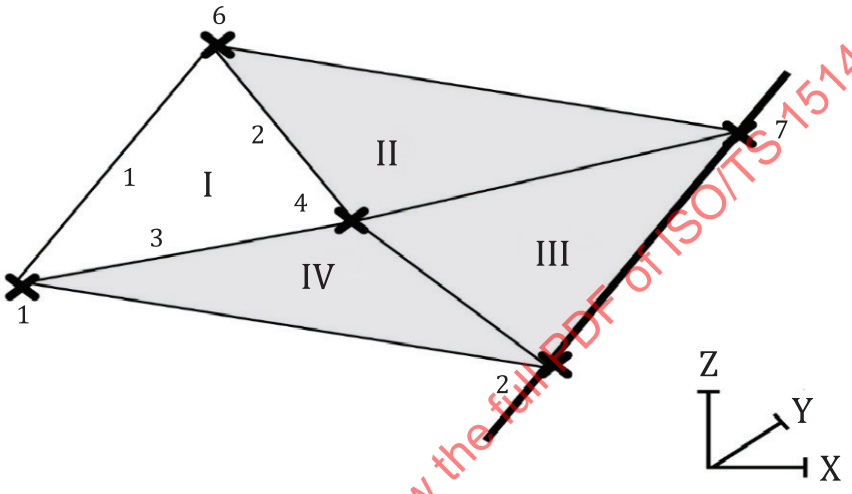


Figure L.3 — Triangle faces

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

L.5.6 Soil topmost surface

The digital elevation model in <*Surfaces*> contains the description of the existing soil surface as one or more <*Surface*>. It consists of the vertices of the component faces <*Pnts*> and the faces <*Faces*> as explained in Clause L.5. In files following this specification, it is also possible to assign source data points and breaklines to the surface. An ISO15143-4_Taxonomy type coding provides surface classification.

L.5.7 Geotechnical model

The geotechnical model (<*Surfaces*>) contains a description of all the surfaces (<*Surface*>) between ground layers. Individual layer surfaces are constructed as explained in Subclause L.5.4. An ISO15143-4_Taxonomy type coding provides surface classifications.

L.6 Route planning across domains

L.6.1 General

Routes encompass highways, local roads and private roads, waterways and railways. Each route has one continuous stationing reference alignment and a vertical alignment. A route plan consists of parametric route alignments, and combination of either their line models or triangulated meshes, or both.

This document discusses different route types individually, including roads, streets, railways, and waterways.

© ISO 2025 – All rights reserved

An alignment group *<Alignments>* consists of several alignments *<Alignment>*. Alignments can be described in two ways:

— Geometric alignment.

— Line string.

Refer to Figure L.4 for an illustration of alignment types. Geometric alignments describe parameters of the horizontal and vertical elements of an alignment. A line string is a description where consecutive points are connected by line segments. Geometric alignments are typically used to describe the reference alignment of a route as well as other important geometric descriptions such as kerb. Other route breaklines are usually described as alignment line strings.

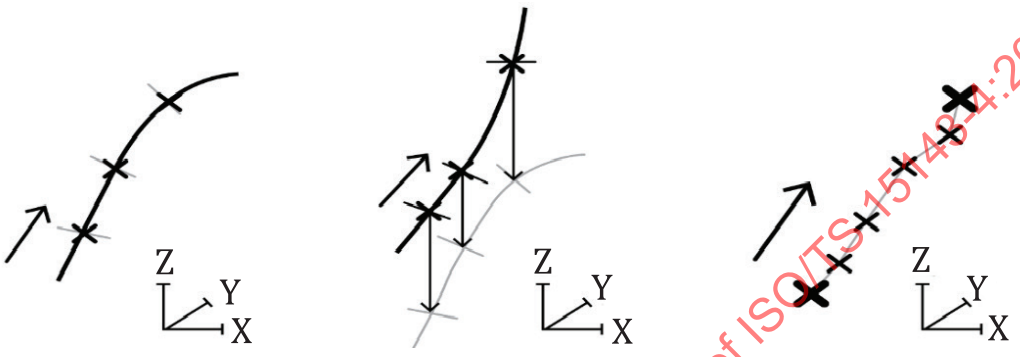


Figure L.4 — Alignment types

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

L.6.2 Route description

Route description is driven by stationing reference alignment. All geometry lines of the same route are given in the same alignment collection *<Alignments>*, each as separate alignment *<Alignment>*.

L.6.3 Composing alignments

L.6.3.1 General

An *<Alignments>* shall be as defined in Table L.22.

Table L.22 — <Alignments> fields

Field	Type	Use	Description
At least One *<Alignment>* element	*<Alignment>*	Required	Geometric alignment, horizontal or horizontal + vertical.
Zero or one *ISO15143-4_Taxonomy <Feature>* element	ISO15143-4_Taxonomy *<Feature>*	Optional	ISO15143-4_Taxonomy Feature extension.
Zero or one *ISO15143-4_StringlineLayers <Feature>* element	ISO15143-4_StringlineLayers *<Feature>*	Optional	ISO15143-4_- StringlineLayers feature extension.
desc	*character_string_255*	Optional	Human readable name.
state	*stateType*	Optional	Alignment group state. Child “Surface” elements inherit the state if set here.

The alignments within a file do not have to be presented in any particular order. It is, however, advisable to first describe geometric alignments and then line strings. It is recommended that descriptions of alignments are given in an intuitive fashion. If the state is set for the entire alignment group *<Alignments>* the *<Alignment>* elements will inherit the state attribute from the parent element.

© ISO 2025 – All rights reserved

An *<Alignment>* shall be as defined in Table L.23.

Table L.23 — <Alignment> fields

Field	Type	Use	Description
One *<CoordGeom>* element	*<CoordGeom>*	Required	A sequential list of line and/or curve and/or spiral elements. After the sequential list of elements, an optional vertical geometry can be defined as a profile, which can be as simple as a list of PVIs (point to point 3D line string).
Zero or one *<Cant>* element	*<Cant>*	Optional	The cant element represents a railway cant/superelevation alignment.
Zero or more *<StaEquation>* elements	*<StaEquation>*	Optional	"staAhead" is the new station value and "staIncrement" indicates whether or not the station values increase or decrease.
Zero or one *<Profile>* element	*<Profile>*	Optional	Profile, aka vertical geometry or long section.
Zero or one *<CrossSects>* element	*<CrossSects>*	Optional	Collection of cross sections.
Zero or one ISO15143-4_Taxonomy <Feature> element	*<Feature>*	Optional	ISO15143-4_Taxonomy Feature extension.
name	*character_string_id*	Required	Unique identifier of alignment.
length	*double*	Required	Alignment length.
staStart	*double*	Required	Alignment station offset (ie. if long alignment is split to parts).
desc	*character_string_255*	Required	Human readable alignment name.
state	*stateType*	Optional	Alignment state.

L.6.3.2 Taxonomy of alignments

Object category and feature code shall be set either for each *<Alignments>* parent element, whose children inherit the values, or for each *<Alignment>* element which value overrides the parent value in case both are defined. The object category and feature code is set by the ISO15143-4_Taxonomy *<Feature>* extension. For more details, see Subclause L.15.1.

L.6.4 Alignment geometry

L.6.4.1 General

The geometric alignment contains the horizontal and vertical alignment information. See Figure L.5 for an illustration of horizontal and vertical alignment. The horizontal alignment information is described in the *<CoordGeom>* and the corresponding vertical geometry in the element *<Profile><ProfAlign>* when available. For the connection between horizontal and vertical geometry, it is mandatory that the geometric description is continuous. Continuous means that all elements are connected in a chain and their tangents meet. The horizontal geometry is described using a 2D or 3D coordinate representation. In case of 2D horizontal geometry, the final elevation values along the alignment can only be produced after vertical geometry is applied. The illustration below shows the horizontal and vertical geometry definition and their connection principle. The optional staStart attribute in *<Line>*, *<Curve>*, *<Spiral>* and *<Profile>* shall not be used for calculating horizontal or vertical coordinates, but can be used for precise stationing equations.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

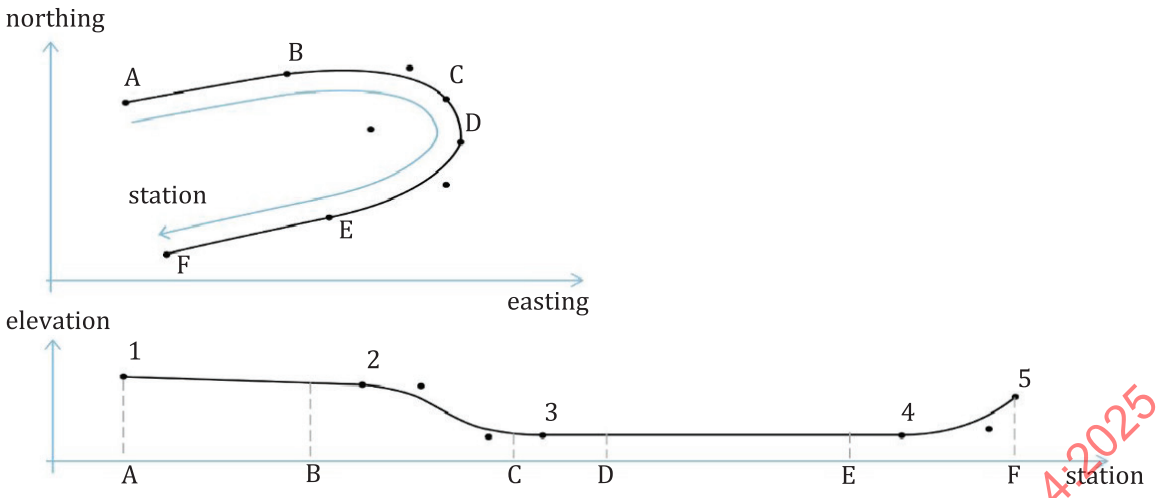


Figure L.5 — Horizontal and vertical alignment

L.6.4.2 Horizontal geometry

Horizontal alignment geometry is in the *<CoordGeom>* element.

A *<CoordGeom>* shall be as defined in Table L.24.

Table L.24 — <CoordGeom> fields

Field	Type	Use	Description
Zero or one *<Line>* elements	*<Line>*	Optional	Line between two points.
Zero or one *<IrregularLine>* elements	*<IrregularLine>*	Optional	Container for series of 2D or 3D points which form a breakline.
Zero or more *<Curve>* elements	*<Curve>*	Optional	The distance from the start to the center provides the radius value. The rotation attribute “rot” defines whether the arc travels clockwise or counter-clockwise from the start to end point.
Zero or one *<Spiral>* element	*<Spiral>*	Optional	An “infinite” spiral radius is denoted by the value “INF”. This conforms to XML Schema which defines infinity as “INF” or “-INF” for all numeric datatypes.

The elements of horizontal alignments:

— *<Line>*

— *<Curve>*

— *<Spiral>*

See Figure L.6 for an illustration of these elements.The horizontal alignment is a listing of consecutive elements, starting at the staStart of the parent *<Alignment>*.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

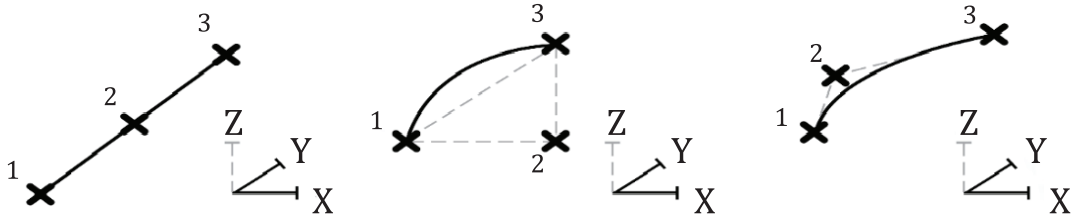


Figure L.6 — Line, curve and spiral

L.6.4.3 Line

A *<Line>* is defined by *<Start>* and *<End>* 2D or 3D coordinates. A *<Line>* shall be as defined in Table L.25.

Table L.25 — <Line> fields

Field	Type	Use	Description
One *<Start>* element	*<Start>*	Required	2D or 3D start point.
One *<End>* element	*<End>*	Required	2D or 3D end point.
desc	*character_string_255*	Optional	Human readable alignment name.
staStart	*double*	Optional	Station offset of line element.
length	*double*	Optional	Line element length.

The format for the *<Start>* and *<End>* coordinates of a *<Line>*, the 2D or 3D coordinates are separated by spaces.

<Start> northing1 easting1</Start>

<Start> northing1 easting1 altitude1</Start>

*<End>*northing2 easting2</End>

*<End>*northing2 easting2 altitude2</End>

L.6.4.4 Curve

A circular arc *<Curve>* is defined by *<Start>* *<Center>* and *<End>* 2D or 3D coordinates.

The distance from the *<Start>* to the *<Center>* provides the radius value of the curve if radius attribute is not present.

A *<Curve>* shall be as defined in Table L.26.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table L.26 — <Curve> fields

Field	Type	Use	Description
One <*Start*> element	<*Start*>	Required	2D or 3D startpoint.
One <*Center*> element	<*Center*>	Required	Represents a 2D or 3D center point defined by either a coordinate text value (“north east” or “NEE") or a CgPoint number reference “pntRef” attribute.
One <*End*> element	<*End*>	Required	2D or 3D endpoint.
Zero or one <*PI*> element	<*PI*>	Optional	Represents a 2D or 3D point of intersection defined by either a coordinate text value (“north east” or “NEE”) or a CgPoint number reference “pntRef” attribute.
rot	*rotDir*	Required	Direction of rotation, one of (cw,ccw).
desc	*character_string_255*	Optional	Human readable name.
staStart	*double*	Optional	Station offset.
length	*double*	Optional	Curve length.
radius	*double*	Optional	Radius of curve.
dirStart	*direction*	Optional	Direction at start of curve.
dirEnd	*direction*	Optional	Direction at end of curve.
chord	*double*	Optional	Length of chord.

A *rotDir* is defined per Table L.27.

Table L.27 — <rotDir> definition

Type	Basetype	Restriction	Description
rotDir	*string*	One of (cw, ccw)	Rotation direction.

The <*Start*> , <*Center*> and <*End*> of a <*Curve*>, the 2D or 3D coordinates are separated by spaces.

<*Start*> northing1 easting1</Start>

<*Start*> northing1 easting1 altitude1</Start>

<*Center*>northing2 easting2</Center>

<*Center*>northing1 easting1 altitude2</Center>

<*End*>northing3 easting3</End>

<*End*>northing1 easting1 altitude3</End>

L.6.4.5 Transition curve

A <*Spiral*> is defined by <*Start*> , point of intersection of the end tangents <*PI*> and <*End*> Coordinates. Allowed transition curve type are:

— An Euler spiral “clothoid”.

— Bi-quadratic parabola “biquadraticParabola”.

— Third-degree spiral “cubic”.

A <*Spiral*> shall be as defined in Table L.28.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

Table L.28 — <Spiral> fields

Field	Type	Use	Description
One <*Start*> element	<*Start*>	Required	2D or 3D start point.
One <*PI*> element	<*PI*>	Required	Represents a 2D or 3D point of intersection defined by either a coordinate text value (“north east” or “NEE”) or a CgPoint number reference “pntRef” attribute.
One <*End*> element	<*End*>	Required	2D or 3D end point.
length	*double*	Required	Total length.
radiusEnd	*double*	Required	Radius at end of spiral or infinite (INF).
radiusStart	*double*	Required	Radius at start of spiral or infinite (INF).
rot	*rotDir*	Required	Direction of rotation, one of (cw,ccw).
spiType	*spiralType*	Required	Spiral type.
desc	*character_string_255*	Optional	Human readable name.
Constant	*double*	Optional	Spiral constant parameter, default = 1.0.
dirStart	*direction*	Optional	Direction at start of spiral.
dirEnd	*direction*	Optional	Direction at end of spiral.
staStart	*double*	Optional	Station offset.

Type <*rotDir*> is defined per Table L.26 above. A *spiralType* is defined per Table L.29.

Table L.29 — <spiralType> definition.

Type	Basetype	Restriction	Description
spiralType	*string*	One of (biquadratic, clothoid, cubic)	Spiral type enumeration.

The <*Start*>, point on intersection of start and end tangents <*PI*> and <*End*> are defined as 2D coordinates separated by spaces.

<*Start*> northing1 easting1</Start>

<*PI*>northing2 easting2</PI>

<*End*>northing3 easting3</End>

L.6.4.6 Vertical geometry

The vertical geometry is described in the <*Profile*><*ProfAlign*> element under the same <*Alignment*> with the horizontal geometry definition. See Figure L.7 for an illustration of vertical geometry elements. Each horizontal geometry can have only one (or 0) vertical geometry. The elements of the vertical geometry are:

— Point of vertical intersection <*PVI*>.

— Vertical circular arc <*CircCurve*>.

— Parabolic vertical curve <*ParaCurve*>.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

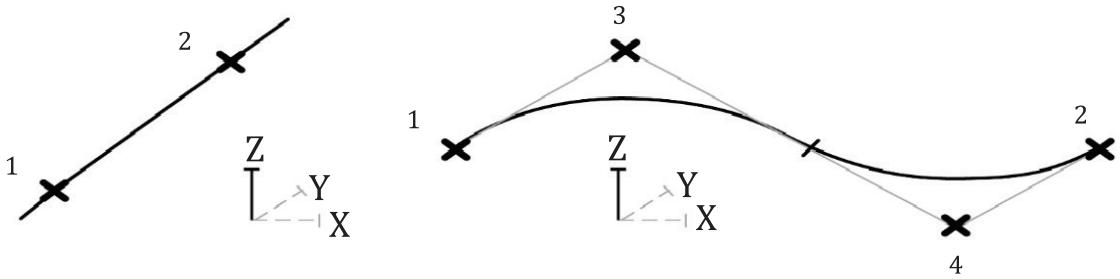


Figure L.7 — Vertical geometry elements

A <*Profile*> shall be as defined in Table L.30.

Table L.30 — <Profile> fields

Field	Type	Use	Description
Atleast one <*ProfAlign*> element	<*ProfAlign*>	Required	ProfAlign element represents a vertical alignment.
desc	*character_string_255*	Optional	Human readable name of vertical geometry.
name	*character_string_id*	Optional	Unique identifier of vertical geometry.
staStart	*double*	Optional	Station offset of vertical geometry.
state	*stateType*	Optional	State of vertical geometry.

A <*ProfAlign*> shall be as defined in Table L.31.

Table L.31 — <ProfAlign> fields

Field	Type	Use	Description
Zero or more <*PVI*> elements	<*PVI*>	Optional	A point of vertical intersection with a space delimited “station elevation”.
Zero or more <*ParaCurve*> elements	<*ParaCurve*>	Optional	A point of vertical intersection with a space delimited “station elevation” and a parabolic vertical curve defined by the “length” attribute.
Zero or more <*UnsymParaCurve*> elements	<*UnsymParaCurve*>	Optional	A point of vertical intersection with a space delimited “station elevation” and a parabolic asymmetric vertical curve defined by the “lengthIn” and “lengthOut attributes.
Zero or more <*CircCurve*> elements	<*CircCurve*>	Optional	A point of vertical intersection with a space delimited “station elevation” and circular curve defined by the length and radius attributes.
name	*character_string_id*	Required	Unique identifier.
desc	*character_string_255*	Optional	Human readable name.
state	*stateType*	Optional	State of profile align point.

L.6.4.7 Point of vertical intersection

The first and last element of the vertical profile is always a point of vertical intersection<*PVI*>.

A point of vertical intersection <*PVI*> marks the ends of the line segments of a vertical geometry. A point of vertical intersection is described by a station and an elevation. These are separated by a space.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

A <*PVI*> shall be as defined in Table L.32.

Table L.32 — <PVI> fields

Field	Type	Use	Description
content	2 of type of double	Required	Whitespace delimited station elevation pair.
desc	*character_string_255*	Optional	Human readable name.

L.6.4.8 Vertical curve

Vertical circular arcs may be combined into S-curves or compound curves.

The location of the <*CircCurve*> is defined by the station and elevation, separated by spaces.

A <*CircCurve*> shall be as defined in Table L.33.

Table L.33 — <CircCurve> fields

Field	Type	Use	Description
content	2 of type of *double*	Required	Whitespace delimited station elevation pair.
length	*double*	Required	Curve length.
radius	*double*	Required	Curve radius.
desc	*character_string_255*	Optional	Human readable name.

L.6.5 Parabolic curve

The parabolic curve <*ParaCurve*> is defined by attribute length and by space separated station and elevation values for vertical intersection point.

A <*ParaCurve*> shall be as defined in Table L.34.

Table L.34 — <ParaCurve> fields

Field	Type	Use	Description
content	2 of type of *double*	Required	Whitespace delimited station elevation pair.
length	*double*	Required	Curve length.
desc	*character_string_255*	Optional	Human readable name.

L.6.6 Line strings

L.6.6.1 General

Line strings are defined in <*CoordGeom*><*IrregularLine*> element as 2D or 3D points. In case of 2D points, a vertical <*Profile*> element can be applied for 3D representation.

L.6.6.2 Line string

A line string shall have sub-elements to define its <*Start*> and <*End*> coordinates. Intermediate 2D or 3D points shall be defined either as <*PntList2D*> or <*PntList3D*>.

The <*Start*> and <*End*> points of an <*IrregularLine*> are individual coordinates are separated by spaces.

The 2D point list <*PntList2D*> consists of 2D coordinates of intermediate points and start and end points, separated by spaces.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

<*PntList2D*>northing1 easting1 northing2 easting2 northing3 easting3... </*PntList2D*>

The 3D point list <*PntList3D*> consists of 3D coordinates of intermediate points and start and end points, separated by spaces.

<*PntList3D*>northing1 easting1 elevation1 northing2 easting2 elevation2 northing3 easting3 eleva- tion3... </*PntList3D*>

An <*IrregularLine*> shall be as defined in Table L.35.

Table L.35 — <IrregularLine> fields

Field	Type	Use	Description
One <*Start*> element	<*Start*>	Required	2D or 3D start point.
One <*End*> element	<*End*>	Required	2D or 3D end point.
One <*PntList2D*> or <*PntList3D*> element	<*PntList2D*> *or* <*PntList3D*>	Required	A sequential space delimited list of 2D or 3D coordinates.
desc	*character_string_255*	Optional	Human readable name.
staStart	*double*	Optional	Station offset of *irregularline.*
length	*double*	Optional	Total length.

L.6.7 String line representation of surface

An alignment group <*Alignments*> is a collection of either geometric alignments or line strings, or both. The string line model (see Figure L.8) used in this documentation is based on the Leica RoadRunner <*RR_StringLineLayers*> / <*RR_StringlineLayer*> feature extensions, but is renamed as <*ISO15143-4_ StringLineLayers*> / <*ISO15143-4_String- lineLayer*>.

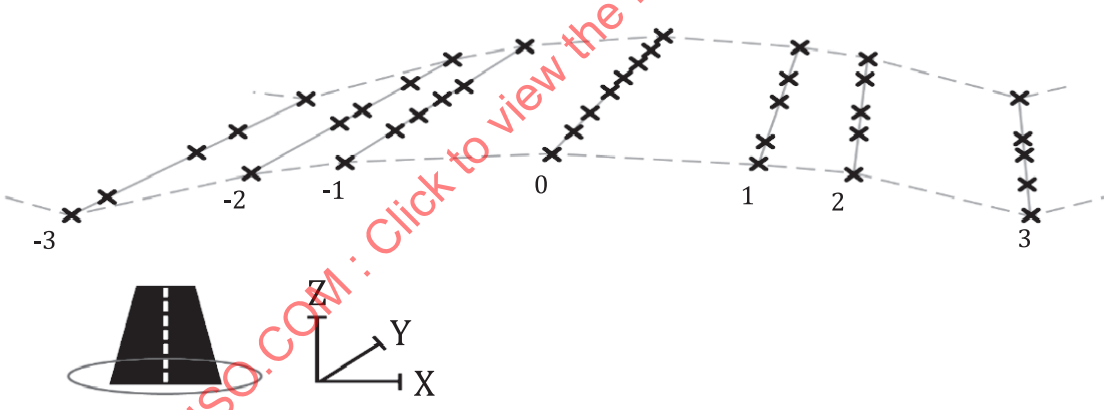


Figure L.8 — String line surface

The string line model of a <*Alignments*> alignment group is defined by the ISO15143-4_StringlineLayers <*Feature*> extension. The string line model is a surface generated from set of alignments listed in ISO15143-4_ StringlineLayer <*Feature*> child element. Unique names of alignments are used as identifiers. Each surface of the string line model <*ISO15143-4_StringlineLayer*> shall have a unique name.

The procedure for constructing a new surface in the string line model in the "ISO15143-4_StringlineLayers" extension:

a) The surface <*ISO15143-4_StringlineLayer*> is assigned a unique name.

b) The line string alignments are listed from left to right towards increasing stationing. The list is separated by commas.

c) The centerline property value is set. Centerline shall be one of the alignments listed.

d) The category code shall be set via ISO15143-4_Taxonomy <feature> extension.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

A line string may belong to multiple surfaces.

L.6.8 Cross-section model

The cross-sections along an alignment may be set in *<Alignment><CrossSects><CrossSect>* as a geometric representation. Cross-sections may provide parametric information at given station intervals (e.g. each 20 meters or at transitions). Additional domain specific parameters are set in the ISO15143-4_CrossSect extension along with the cross-section elements.

The cross-section parameters describe the situation at a given station. The cross-section parameters are presented at a station where a value begins or stops changing.

A *<CrossSects>* shall be as defined in Table L.36.

Table L.36 — <CrossSects> fields

Field	Type	Use	Description
At least one *<CrossSect>* element	*<CrossSect>*	Required	Individual cross section.
Zero or one ISO15143-4_Taxonomy *<Feature>* element	ISO15143-4_Taxonomy *<Feature>*	Optional	ISO15143-4_Taxonomy Feature extension.
desc	*character_string_255*	Optional	Human readable name.
name	*character_string_id*	Optional	Unique identifier.
state	*stateType*	Optional	State of cross section collection.

A *<CrossSect>* shall be as defined in Table L.37.

Table L.37 — <CrossSect> fields

Field	Type	Use	Description
Zero or more *<DesignCrossSectSurf>* elements	*<DesignCrossSectSurf>*	Optional	Each structural layer has its own "DesignCrossSectSurf".
Zero or one *ISO15143-4_CrossSect <Feature>* element	*ISO15143-4_CrossSect <Feature>*	Optional	ISO15143-4_Crossect Feature extension
Zero or one *ISO15143-4_Taxonomy <Feature>* element	*ISO15143-4_Taxonomy <Feature>*	Optional	ISO15143-4_Taxonomy Feature extension
sta	*double*	Required	Station number of cross section (staStart+2d distance along alignment).
name	*character_string_id*	Optional	Unique identifier.
desc	*character_string_255*	Optional	Human readable name.

The planned cross-sections may be represented by design cross-section surfaces set in *<DesignCrossSectSurf>* where each cross-section point is given by *<CrossSectPnt>* as a space separated value pairs “Offset Elevation”. See Figure L.9 for an illustration of cross sections.

Offsets are measured horizontally in linear units from the center line (positive values to the right).

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

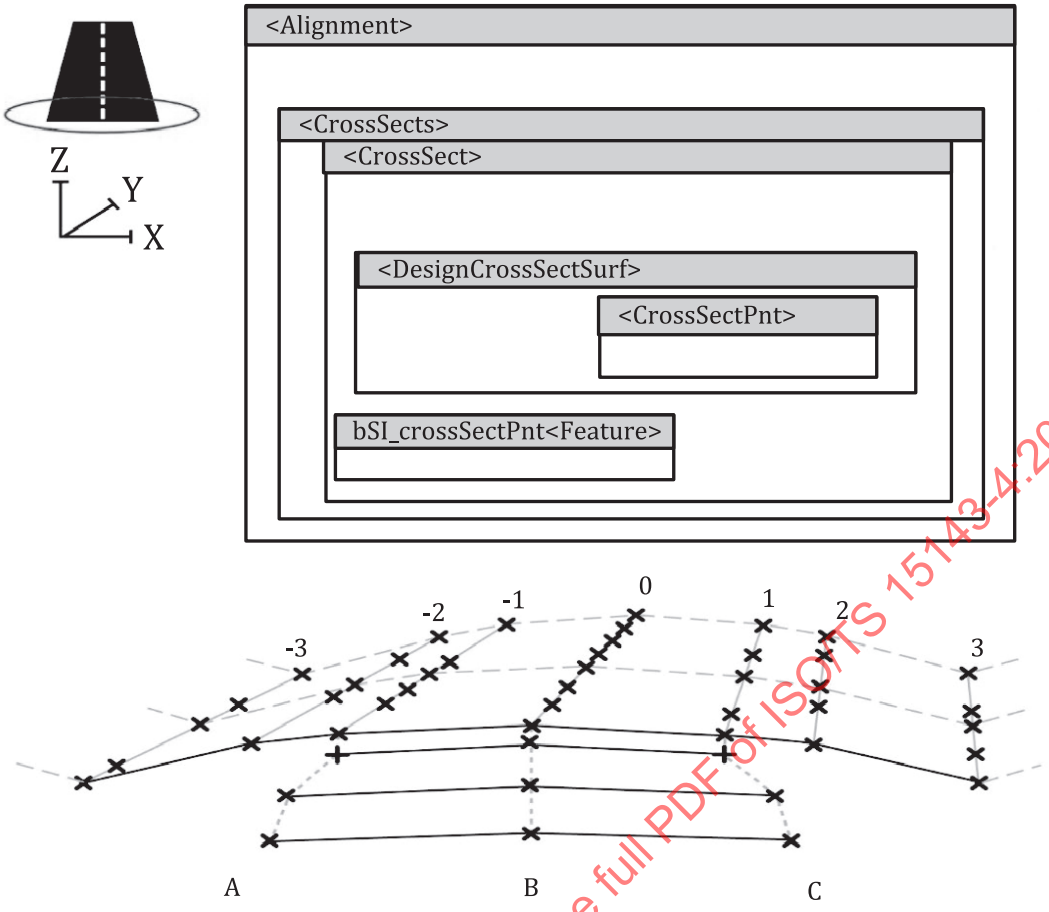


Figure L.9 — Cross sections

A *<DesignCrossSectSurf>* shall be as defined in Table L.38.

Table L.38 — <DesignCrossSectSurf> fields

Field	Type	Use	Description
Zero or more *<CrossSectPnt>* elements	*<CrossSectPnt>*	Optional	Cross section point.
Zero or one ISO15143-4_Taxonomy *<Feature>* element	ISO15143-4_Taxonomy *<Feature>*	Optional	ISO15143-4_Taxonomy Feature extension.
name	*character_string_id*	Optional	Unique identifier.
desc	*character_string_255*	Optional	Human readable name.
state	*stateType*	Optional	State enum.

A *<CrossSectPnt>* shall be as defined in Table L.39.

Table L.39 — <CrossSectPnt> fields

Field	Type	Use	Description
content	2 of type *point*	Required	2D point.
name	*character_string_id*	Optional	Unique identifier.
desc	*character_string_255*	Optional	Human readable name.
state	*stateType*	Optional	State enum.
code	*character_string_255*	Optional	Feature code.
dataFormat	*dataFormatType*	Required	Format of CrossSectPnt datafield.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

A *dataFormatType* is defined per Table L.40.

Table L.40 — <dataFormatType> definition

Type	Basetype	Restriction	Description
dataFormatType	*string*	One of (Offset Elevation, Slope Distance)	Cross section point data format enumeration.

L.6.9 Cross-section domain specific parameters

Cross-section parameters refer to parametric information complementing the model represented as surface or stringline models. These include design parameters such as the widths and superelevations of roads.

L.6.10 Structural model

Structural model of a route can be represented as a combination of stringline layers, surface meshes and cross-sections. Stringline layers are most efficient for data exchange of machine readable structural model. However, stringline layers can be problematic, for example, on intersection, roundabouts and extremely tight corners. In these cases, surface mesh can provide a more accurate structural model. Cross-sections shall not be used to generate structural model. However, cross-sections can provide useful design parameters, for example for railway track placement, and cross-sections can be used to validate the geometry of structural model.

L.7 Roads and streets

L.7.1 General

A road or street alignment is defined according to the process defined in Clause L.6. A single design file can contain several route plans from several domains, i.e. the same file can contain, for example, road, street, and railway plans.

Street designs often interface with water supply and sewerage systems that may be described in the same or several design files.

A structural model of road or street design may contain:

— individual alignments and their purpose, defined by type coding (geometric alignments and line strings);

— stringline models (line string alignments);

— surface meshes (TIN surface);

— cross-section models.

The superelevation of the road may be described alongside the cross-section parameters. Cross-section parameters are described at the transition points, when the transition of the value of a parameter begins or ends. Superelevation encompasses the cross slope of the roadway and, in case, of a street also the sidewalk.

L.7.2 Geometry

The central alignments such as the centerline and left and right edges of a road or the centerline and edges of a sidewalk for a street are typically described as geometric alignments while the rest are described in terms of line strings.

L.7.3 String line model

A detailed description of the stringline model can be found in Subclause L.6.5.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved

L.7.4 Cross-sections

L.7.4.1 General

A detailed description of the cross-section model can be found in Subclause L.6.6.

L.7.4.2 Cross-section parameters

The cross-section parameters are set for the stationing reference alignment cross-sections *<Alignment>.<CrossSects>.<CrossSect>* in the optional ISO15143-4_CrossSect *<Feature>* extension. The first cross-section of the alignment is defined by describing all parameters of the cross-section. The parameters of the following cross-sections are only described if a value begins or stops changing.

The following parameters may be defined in the ISO15143-4_CrossSect *<Feature>* extension for each individual cross-section:

— Pavement class.

— Pavement thickness.

— Load capacity (kN/m2).

More details regarding the ISO15143-4_CrossSect *<Feature>* extension can be found in Subclause L.15.3.

L.7.4.3 Transitions in superelevation

The superelevation is indicating the transition points, when a transition in the superelevation either begins or ends. The cross-slopes are defined along with the cross-section parameters. Figure L.10 illustrates the process in a road design environment.

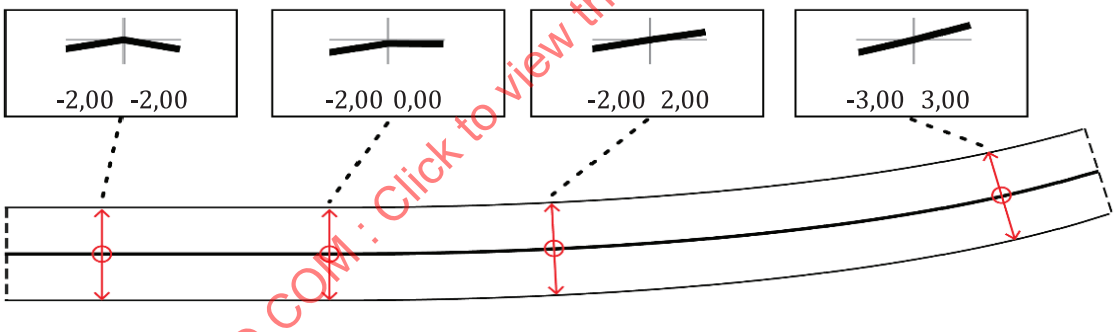


Figure L.10 — Transitions in superelevation

L.7.5 Structural model

The process of constructing a structural model of a road or street is described in detail in Subclause L.6.8.

Figures L.11 and L.12 demonstrate the composition of structural models (as surface meshes) in road and street design.

STANDARDSISO.COM : Click to view the full PDF of ISO/TS 15143-4:2025

© ISO 2025 – All rights reserved