



Reporting of establishment information under the Seveso Directive (eSPIRS)

Quality Controls Documentation



Cover design: EEA

Cover photo: Photo by Victor on [Unsplash](#)

Layout: EEA

Acknowledgments

This data model documentation was prepared by the [European Environment Agency](#) (EEA) in cooperation with the [European Topic Center for Health and Environment](#) (ETC/HE). The data model was produced using Unified Modelling Language (UML). Please contact seveso.helpdesk@eea.europa.eu for any questions, clarifications or comments.

Version control

Version ID	Description	Date
0.1	Internal draft for discussion	May 2025
0.2	Internal draft for discussion	September 2025
0.3	Version for sharing with the Member State Reporters	October 2025
0.4	Final version for 2026 eSPIRS reporting cycle • Added field length checks for reporting authority. • Added field length checks for inspireId sub components. • Added custom checks on URL values allowing multiple, comma separated entries. • Removed checks on site confidentiality following simplification. • Added cross-delivery QC descriptions.	June 2026

Contents

Acknowledgments	2
Version control	2
Contents	3
1 Introduction	4
2 Types of quality checks	5
3 Quality checks severity levels	6
4 Validation process	7
5 Quality checks description	9
5.1 Custom checks on text field lengths	11
5.2 Custom checks on InspireId format	13
5.3 Custom checks on productionSite data consistency	14
5.4 Custom Spatial Checks	15
5.5 Custom checks on linked tables and general data consistency	16
5.6 Custom checks on URL formats	18
5.7 Information Level checks	19
5.8 Offline cross-delivery checks	20
Annex 1 – Glossary of Terms	22

1 Introduction

Quality control is a key component of effective data management, ensuring that reported data meets fundamental quality criteria of accuracy, completeness, consistency, and uniqueness. Providing timely and comprehensive feedback on data quality issues helps minimize the number of correction cycles and reduces the overall time required to achieve high-quality data submission.

The Reportnet 3 platform includes an integrated automatic quality control framework that delivers immediate feedback to users upon data submission. This document outlines the general principles of the quality control mechanisms implemented in Reportnet 3. It explains how reporters can use the system to verify their data and interpret quality checks (QCs) feedback to improve the quality of their submissions.

In addition, this document provides a detailed description of the specific quality controls developed for the eSPIRS dataflow. These include so-called Tier 2 checks, which are designed to assess the consistency of the reported data against previously reported information. Tier 2 checks are currently performed by the EEA outside the Reportnet 3 platform. However, feedback on the outcomes of these checks is still provided through the system. To complement this document, the following related documents are available to address other aspects of the dataflow:

- The Data Model Documentation describing the data elements to be reported for eSPIRS.
- The Data Format Guide, describing the structure and content of the files to upload in Reportnet 3.
- A Manual for Reporters, which will guide reporters on the practicalities of the reporting.

All of the documents concerning the reporting of Seveso Directive data flows can be found at: <https://industry.eea.europa.eu/seveso/about>

2 Types of quality checks

Quality checks in Reportnet 3 can be categorized in several ways to help users understand their purpose and impact. Two primary dimensions of classification are type of implementation (automated vs. custom) and level of data structure (field, record, or table level).

When looking at implementation types, we have **Automated** quality checks that are generated automatically by the system based on the structure and metadata of the dataset.

Examples of automated checks are:

- Ensuring that values in a column match the expected data type (e.g., numeric fields contain only numbers).
- Enforcing the provision of values in the mandatory fields.
- Verifying the uniqueness of Primary Key values in a table.

Automated checks are integral to the validation process and help identify basic structural issues early in the submission workflow.

Custom quality checks, on the other hand, are used for more complex, rule-based validations defined by domain experts at the dataset design stage. Manual checks are tailored to address specific aspects of the data.

Examples of manual checks are:

- Verifying that a numeric value falls within a valid range, possibly dependent on the content of other columns.
- Implementing logical rules that go beyond simple structural validation.

When looking at data structure level, we have **Field** level checks that validate individual data elements, such as ensuring a value complies with the expected type or is not missing in a mandatory field. **Record** level checks evaluate conditions across multiple fields within the same row (record). For instance, if the value in one column imposes constraints on another column in the same row, a record-level check would apply. Finally, **Table** level checks involve the entire content of a table. They are typically used to confirm the completeness of the dataset, for example, verifying that all required records are present.

The level at which a QC is applied affects how messages are displayed in the Reportnet 3 user interface, which will be illustrated later in this document.

3 Quality checks severity levels

Each quality check in Reportnet 3 is assigned to a severity level, which determines the significance of the issue and its impact on the reporting process. Figure 1 shows the symbology associated to the error severity levels. There are currently four severity levels:

INFO level checks are used for Informational messages that highlight noteworthy aspects of the data. They may include summaries, report unusual values, or significant changes compared to previous submissions.

- *No action is required by the reporter.*

WARNING level checks Indicate a potential issue that cannot be conclusively flagged as an error due to system limitations or ambiguous dataset definitions. For example, a field might be conditionally mandatory under complex rules that are difficult to evaluate programmatically.

- *It is up to the reporter to assess the warning and decide whether corrections are needed.*

ERROR level checks signal a clear issue that, while not critical, still requires correction. An example could be a missing mandatory field that is not used in downstream processes or regulatory assessments.

- *Reporters are expected to correct these issues. Data requesters may follow up to ensure resolution.*

BLOCKER level check represents critical problems that prevent the data from being processed or understood. This can be caused by incorrect data types (e.g., text instead of numbers) or invalid date formats.

- *BLOCKER errors must be resolved by the reporter before the dataset can be submitted. A single blocker will prevent dataset release.*

Figure 1. Error level severity errors and their representations,



4 Validation process

The validation process is automatically performed during the release of the data; however, the reporter can trigger validation at any time during the reporting process. Validations are performed separately for each dataset. The reporter needs to separately validate the **ReportInfo** and **Establishments** datasets to perform a full validation of the dataflow content. Validation of a dataset starts by clicking on the **Validate** button [A] in the Reportnet 3 user interface, indicated in Figure 2. The Mapping dataset does not feature any validation as it is intended only to provide read-only content mapping of the identifiers created by the EEA for data publication to the inspireIds reported by the countries. The validation functionality is only available when the table is not in edit mode, thus the user needs to first exit edit mode and then start the validation. Edit mode can be enabled and disabled using the rightmost button in position [C] in Figure 2.

Figure 2. Reportnet 3 user interface. Controls relevant to dataset validation.



The execution of the quality checks for the eSPIRS dataflow normally takes a few minutes, depending on the size of the dataset and the global workload the platform is experiencing. Upon completion of the process, the reporter can access the results of the validation by clicking on the **Show validations** button [B] which will show the validations dialog. The validations dialog presents a table listing all the QC error messages together with their severity level, their identification code, the level of data structure they belong to, the table and field (if any) they refer to, and the number of records affected.

The user can navigate the validation table by using the controls placed at the bottom **[D]** (see Figure 3), sort the columns by clicking on the header **[E]**, or filter the content of the table using the controls at the top of the table. After applying the filters **[F]**, the original content of the table can be restored by clicking the **Reset** button **[G]**. Clicking on a specific line of the validation table **[H]** will show the records in the table affected by the error.

Figure 3. Validations dialog.

Entity	Table	Field	Code	Level error	Message	Number of records
FIELD	SevesoEstablishment	inspireId	02_FC_inspireId_Format	BLOCKER	MT-SEVESO/00007 Establishment is not a valid inspireId. It should contain at least one single / separating the namespace from the localid. The / should be single and not placed at the beginning or at the end.	1
RECORD	SevesoEstablishment		02_S5_establishmentLocationCountry	BLOCKER	The establishment <inspireId> is over 1km outside the country boundaries.	7
TABLE	SevesoEstablishment		02_TR_EstablishmentCountByTier	INFO	Current submission contains 153 Tier I and 117 Tier II Establishments.	1
TABLE	SevesoEstablishment		02_TR_SitesCount	INFO	Current submission contains 265 ProductionSites.	1
RECORD	SevesoEstablishment		03_S5_EstablishmentSiteLocation	BLOCKER	The isolated establishment MT-SEVESO/000005 Establishment ; MT-SEVESO/000007 Establishment ; MT-SEVESO/000006 Establishment ; BE-SEVESO/000137 Establishment ; MT-SEVESO/000004 Establishment ; BE-SEVESO/000061 Establishment ; BE-SEVESO/000206 Establishment ; BE-SEVESO/000150 Establishment is too far away from the site.	1
TABLE	SevesoEstablishment		03_TR_EstablishmentCountByStatus	INFO	Current submission contains 0 Decommissioned, 57 Disused and 213 Functional Establishments.	1
FIELD	SevesoEstablishment	siteLongitude	04_CC_siteLongitude	BLOCKER	Different values of siteLongitude (14.498518 ; 15.36082) are reported for the same productionSite in different records	4
RECORD	SevesoEstablishment		04_S5_NonIsolatedEstablishmentSiteLocation	BLOCKER	The non isolated establishment MT-SEVESO/000003 Establishment ; BE-SEVESO/000045 Establishment ; BE-SEVESO/000001 Establishment ; BE-SEVESO/000002 Establishment is too far away from the site.	1
TABLE	EstablishmentConfidentiality		04_TR_EstablishmentCountByConfidentialityFields	INFO	Reported confidentiality counts for Establishments, Name: 3, ParentCompany: 4, Location 5, Address: 5, SevesoTier: 1, Status: 2, IndustryType: 7, LastInspection: 8, Comments: 9.	1
FIELD	SevesoEstablishment	siteLatitude	05_CC_siteLatitude	BLOCKER	Different values of siteLatitude (35.886224 ; 44.77403) are reported for the same productionSite in different records	4

Rows per page: 10 | 1 2 3 4 | Go to: 2 of 4 | Total: 1277 records (total errors: 422)

After completion of the dataset validation process, error messages and icons are shown in the tab of any table where a QC rule fails. When viewing the table content, Field level errors are marked by icons in the specific field **[J]**, while Table level errors are shown by an icon placed on top of the table **[K]**. Record level QC errors in a record are highlighted by an icon, placed in the **Validations** column **[I]** (see Figure 4). The icon displayed in the **Validations** column represents the highest severity level of the errors observed in the record. For example, a **BLOCKER** icon will be shown when two Fields of the same Record have errors of **BLOCKER** and **ERROR** severity. Hovering on the icon shows a tooltip bubble presenting the combined error messages of all the QC failing for the row

Figure 4. Validation errors representation in tables,

Validations	inspireId	identifierScheme	identifier	sevesoEstablishmentTier	status	establishment
[I]	BE-SEVESO/000045. Establishment	EstablishmentCode	BE-000045	upperTier	functional	TIBI sc - Usine

5 Quality checks description

In the eSPIRS dataflow, the **Establishments** dataset currently features over a hundred QCs, while the **ReportInfo** dataset has just around twenty. Most of these QCs are automated, and the vast majority have a BLOCKER severity level. Error messages associated with automatic QCs are self-explanatory and usually not customized. For additional clarity, in this document we provide, for each group of automated checks, a description of the underlying problem. In the following section, each group of checks is identified by the first part of its code and, when needed, by its name.

FT (Field Type): checks that the value in the field is a valid representation of the field data type (INTEGER, DECIMAL, URL, DATE). An error is reported when the textual representation of the value cannot be converted to the underlying data type.

All FT QCs are BLOCKERs.

Examples:

- A double decimal place in a decimal number 14..3444 instead of 14.3444
- A date provided in a different format 03-APR-2025 instead of 03/04/2025

Solution: Correct the data to comply with the format specification. If the error persists, contact the seveso helpdesk.

FC (Field Cardinality): checks that a value is provided for a field marked as mandatory.

All FC QCs are BLOCKERs except the ones checking the presence of mandatory Establishment Address components (streetName, buildingNumber, city, postCode), and those checking the mainIndustryTypeNace, publicInformationURL that are classified as ERRORs. FC QC related to substance quantity declaration (quantityTNE) is also classified as ERROR.

Solution: Provide the missing data. If the error persists, contact the seveso helpdesk.

TB (Mandatory Table records check): checks that at least one record is provided in a table that is marked as mandatory

All TB QCs are BLOCKERs.

Solution: Provide the missing data if the data is not missing contact the seveso helpdesk.

TC (Field Type LINK): checks that LINK¹ fields, when not empty, contain values matching an entry in the linked table. To solve these kinds of errors, the reporter should verify that the value inserted in the field corresponds to a value in the linked field. Please note that in some cases, in addition to the actual value used in the link, an additional element of the linked table is shown (as a label). However, only the value used in the link should be entered in the field.

All TC QCs are BLOCKERs.

¹ A LINK field is used when a column of a certain table should only contain values that belong to column in a different table (possibly in a separate dataset). For example, all the fields containing values selected from a pre-defined list (substances, Seveso tier classification, etc.) are defined as LINK fields referencing tables in the Vocabularies dataset.

Solution: Verify that the value in the linked table exists and that there are no spelling errors in the specified value. If a value is missing in a reference dataset table or if the error persists even if the value seems to be correct, contact the seveso helpdesk.

In addition to the automatic checks, several custom checks are also present for both the **Establishments** and **ReportInfo** datasets. The custom checks are designed to identify issues specific to the nature of the reported data. Custom checks are identified by a code that begins with the class of verification they perform (for example, length check is labelled FL: Field Length, while format check code is FF: Field Format) followed by a numeric index unique within each class and by the name of the field(s) that is(are) checked. Additional textual description could be placed at the end to better identify the check purpose. In the rest of the section, we describe each one of the manual checks and provide instructions on how to solve the underlying issues when an error is detected.

5.1 Custom checks on text field lengths

A first group of manual checks is designed to verify lengths of textual fields where there is a field length (FL) constraint.

FL_01_establishmentName. Verifies that **establishmentName** does not exceed 255 characters.

FL_02_parentCompany. Verifies that **parentCompany** does not exceed 255 characters.

FL_03_streetName. Verifies that **streetName** does not exceed 255 characters.

FL_04_buildingNumber. Verifies that **buildingNumber** does not exceed 100 characters.

FL_05_city. Verifies that **city** does not exceed 60 characters.

FL_06_postCode. Verifies that **postCode** does not exceed 30 characters.

FL_07_publicInformationURL. Verifies that **publicInformationURL** does not exceed 2000 characters.

FL_08_generalURL. Verifies that **generalURL** does not exceed 2000 characters.

FL_09_lastInspectionURL. Verifies that **lastInspectionURL** does not exceed 2000 characters.

FL_10_comments. Verifies that **comments** does not exceed 800 characters.

FL_11_identifierScheme. Verifies that **identifierScheme** does not exceed 80 characters.

FL_12_identifier. Verifies that **identifier** does not exceed 80 characters.

FL_13_siteIdentifierScheme. Verifies that **siteIdentifierScheme** does not exceed 80 characters.

FL_14_siteIdentifier. Verifies that **siteIdentifier** does not exceed 80 characters.

FL_21_physicalProperties. Verifies that **physicalProperties** does not exceed 500 characters.

FL_22_substanceComments. Verifies that **substanceComments** does not exceed 500 characters.

FL_23_authorityName. Verifies that reporting authority **authorityName** does not exceed 255 characters.

FL_24_auth_streetName. Verifies that reporting authority **streetName** does not exceed 255 characters.

FL_25_auth_buildingNumber. Verifies that reporting authority **buildingNumber** does not exceed 100 characters.

FL_26_auth_city. Verifies that reporting authority **city** does not exceed 60 characters.

FL_27_auth_postCode. Verifies that reporting authority **postCode** does not exceed 30 characters.

FL_28_authorityPhone. Verifies that reporting authority **authorityPhone** does not exceed 30 characters.

FL_29_authorityEmail. Verifies that reporting authority **authorityEmail** does not exceed 255 characters.

FL_30_auth_comments. Verifies that reporting authority **comments** does not exceed 500 characters.

.

All FL QCs are classified as ERRORS except those concerning the inspireId of Establishments or Production Site that are BLOCKERS. FL QC concerning InspireIds are described in the following section.

Solution: Verify that the field length does not exceed the maximum allowed size, reduce the size of the text to fit the maximum size. If the error persists even if the value seems to be correct, contact the seveso helpdesk.

5.2 Custom checks on InspireId format

The following manual checks verify that the format of the InspireId identifier is correct, they belong to field length (FL) and Field Format (FF) categories

FL_15_inspireId. Checks that the total length of the establishment inspireId does not exceed the limit of 161 characters given by the sum of the maximum length of its components: **namespace** (80 characters), **localId** (80 characters) plus one separation character (forward slash '/').

FL_16_namespace. Checks that the length of the **namespace** component of the establishment inspireId does not exceed the limit of 80 characters.

FL_17_localId. Checks that the length of the **localId** component of the establishment **inspireId** does not exceed the limit of 80 characters.

FF_01_inspireId_characters. Checks that only valid characters are used. The only allowed characters are uppercase and lowercase letters of the latin alphabet, digits, colon [:], forward slash [/], dash [-] and underscore [_]. No spaces, special characters or tabs are allowed.

FF_02_inspireId_components. Checks that the **inspireId** can be separated into its underlying components namespace and localId. The inspireId is built by concatenating the **namespace** and **localId** placing a forward slash between them.

InspireIds should contain at least one isolated forward slash; the text at the left of the last forward slash is considered the **namespace**, the text at the right is the **localId**. No forward slash can be used as the last character of the **inspireId** and the forward slash cannot appear next to another forward slash.

Similar checks are performed on **siteInspireId**:

- **FL_18_siteInspireId,**
- **FL_19_site_namespace,**
- **FL_20_site_localId**
- **FF_03_siteInspireId_characters,**
- **FF_04_siteInspireId_components.**

All InspireId QCs including those on Field Length are classified as BLOCKERS.

Solution: check that only allowed characters are used in the inspireId and that a single forward slash is used to separate the namespace from the localId. Verify the total length of the inspireId and the length of its components. If the error persists even if the value seems to be correct, contact the 13seveso helpdesk.

Example of InspireId composition:

Namespace: IT.CAED **LocalId:** EST_00134234

InspireId: IT.CAED/EST_00134234

5.3 Custom checks on productionSite data consistency

As explained in more detail in the data format guide, the eSPIRS data model was simplified in Reportnet 3 by combining the **establishment** and **productionSite** information in one table. When more than one establishment shares the same **productionSite**, the productionSite specific information (name, coordinates, thematic identifier, confidentiality classification) is repeated more than once, leading to possible inconsistencies. A set of custom QCs was introduced to detect when the records with the same **siteInspireId** contain different values of site related information. All the custom checks on data consistency are identified by the code 'CC' (Consistency Check).

CC_01_siteName. Checks that the **siteName** field has the same values for records in **SevesoEstablishment** table having the same **siteInspireId**.

CC_02_siteIdentifierScheme. Checks that the **siteIdentifierScheme** field has the same values for records in the **SevesoEstablishment** table having the same **siteInspireId**.

CC_03_siteIdentifier. Checks that the **siteIdentifier** field has the same values for records in the **SevesoEstablishment** table having the same **siteInspireId**.

CC_04_siteLongitude. Checks that the **siteLongitude** field has the same values for records in the **SevesoEstablishment** table having the same **siteInspireId**.

CC_05_siteLatitude. Checks that the **siteLatitude** field has the same values for records in the **SevesoEstablishment** table having the same **siteInspireId**.

All consistency checks on productionSite are classified as BLOCKERS.

*Solution: Correct the submission, making sure that the information on the same **ProductionSite** provided in different rows of the **Establishment** or **EstablishmentConfidentiality** table is the same. If the error persists even if the value seems to be correct, contact the 14seveso helpdesk.*

5.4 Custom Spatial Checks

Approximate coordinates of the centre of the areas belonging to establishments and production sites need to be reported in the **Establishments** dataflow. A number of QCs are in place to make sure that the reported coordinates are within the boundaries of the reporting country and that the locations of establishments and related production sites are compatible in terms of distance. Spatial checks are identified by the code 'SC' (Spatial Consistency).

SC_01_siteLocationCountry. Checks that the location of the reported **ProductionSite** is within 1 km from the country border. A 1 km buffer zone is used to introduce tolerance avoid errors due to different representations of complex boundary geometry.

SC_02_establishmentLocationCountry. Checks that the location of the reported **SevesoEstablishment** is within 1 km from the country border.

SC_03_IsolatedEstablishmentSiteLocation. Checks that the distance between **Establishment** and the **ProductionSite** does not exceed 500 m. This rule applies to 'isolated' Establishments where the **ProductionSite** is linked to a single **Establishment**. In this case, it is assumed that the boundaries of the Site and of the Establishment almost overlap and distances above 500 m could indicate an error in the specification of the coordinates.

SC_04_NonIsolatedEstablishmentSiteLocation. Checks that the distance between **Establishment** and the **ProductionSite** does not exceed 2000 m. The larger threshold value applies to 'non-isolated' Establishments where multiple entities belong to the same **ProductionSite**. In these cases, it is expected that the point representing the site could be further away from the point representing the **Establishment**.

SC_05_longitude_accuracy. Checks that the **Establishment** longitude is provided with at least 5 digits after the decimal separator.

SC_06_latitude_accuracy. Checks that the **Establishment** latitude is provided with at least 5 digits after the decimal separator.

SC_07_siteLongitude_accuracy. Checks that the **ProductionSite** longitude is provided with at least 5 digits after the decimal separator.

SC_08_siteLatitude_accuracy. Checks that the **ProductionSite** latitude is provided with at least 5 digits after the decimal separator.

SC_09_longitude_range. Checks that the **Establishment** longitude falls within the range [-180, 180] degrees.

SC_10_latitude_range. Checks that the **Establishment** latitude falls within the range [-90, 90] degrees.

SC_11_siteLongitude_range. Checks that the **ProductionSite** longitude falls within the range [-180, 180] degrees.

SC_12_siteLatitude_range. Checks that the **ProductionSite** latitude falls within the range [-90, 90] degrees.

All spatial checks are classified as BLOCKERS.

Solution: verify and correct ProductionSite and/or Establishment coordinates. If the error persists even if the values seem to be correct, contact the seveso helpdesk.

5.5 Custom checks on linked tables and general data consistency

This group of custom checks verifies the consistency of entries reported in one or more tables. They belong to the group of consistency checks (CC), table unicity check (TU) or table relationship check (TR).

TR_01_ConfidentialityDeclarations. Checks that each record in **SevesoEstablishment** table has a corresponding record in **EstablishmentConfidentiality** with the same establishmentId. Confidentiality declarations should be provided for every Establishment even if all the confidentiality fields are empty to avoid unintended release of confidential information.

CC_06_otherIndustryTypeSPIRSNACE. Checks that at least one value is provided either for SPIRS or NACE secondary industry types in each row of **EstablishmentIndustryType** table.

CC_07_otherIndustryTypeNACE. Checks that the NACE code used as main industry type is not repeated as a secondary one.

CC_08_otherIndustryTypeSPIRS. Checks that SPIRS code used as main industry type is not repeated as a secondary one.

CC_09_mainIndustryTypeSPIRS. Checks that main SPIRS code is provided, even if not mandatory.

CC_10_mainIndustryTypeNACE. Checks that main NACE code belongs to the latest version (rev. 2.1).

CC_11_substanceDirective. Checks that substance code belongs to the list included in the Seveso III Directive. Seveso II Directive codes are not allowed.

CC_12_mainIndustryTypeSPIRS. Checks that when secondary SPIRS values are provided the main SPIRS is also entered.

CC_13_mainIndustryTypeNACE. Checks that when secondary NACE values are provided the main NACE is also entered.

CC_14_mainIndustryTypeNACE_level. Checks that the main NACE classification is provided with the 4 digits level of detail required by CID 2022/1979.

CC_15_quantityTNE_range. Checks that substance quantities are not negative..

TU_01_IndustryTypeSPIRS. Checks that the same SPIRS code is not declared multiple times for the same **Establishment** as 'other' industry type.

TU_02_industryTypeNACE. Checks that the same NACE code is not declared multiple times for the same **Establishment** as 'other' industry type.

TU_03_ThematicIdentifier. Checks that the combination of **thematicIdentifierScheme** and **thematicIdentifier** is not repeated in the **SevesoEstablishment** table.

TU_04_siteThematicIdentifier. Checks that the combination of **siteThematicIdentifierScheme** and **siteThematicIdentifier** is not repeated in the **SevesoEstablishment** table.

TU_05_siteInspireIdUniqueness. Checks that inspireId used for establishments are not used for ProductionSites.

All custom checks on linked table consistency are classified as BLOCKERs, with the exception of CC_09_mainIndustryTypeSPIRS and CC_10_mainIndustryTypeNACE) that is classified as WARNING, and CC_14_mainIndustryTypeNACE_level, that is classified as ERROR.

Solution: Check the underlying data. If the error persists contact the seveso helpdesk.

5.6 Custom checks on URL formats

Data fields **generalURL**, **publicInformationURL** and **lastInspectionURL** in the **SevesoEstablishment** table store URL information. Following the Member States request, the data schema was amended to allow submission of multiple URL separated by a semicolon. To verify the correctness of the URL representation custom field format checks were introduced. All the URLs in the sequence should start with **http://** or **https://**. URLs starting with **www** are not considered valid.

FF_05_ publicInformationURL. Checks that the field contains one or more valid URLs separated by a semicolon.

FF_06_ generalURL. Checks that the field contains one or more valid URLs separated by a semicolon.

FF_07_ lastInspectionURL. Checks that the field contains one or more valid URLs separated by a semicolon.

All custom checks on URLs format are classified as ERRORS.

Solution: Check the format of the provided urls and the presence of the semicolon separator in case of multiple entries. Contact seveso helpdesk. if the error persists.

5.7 Information Level checks

Information level checks provide the reporters summary information on the data delivery that could help identify potential issues. These data summary checks are identified by the code DS (Data Summary). DS checks are useful to verify the completeness of the data submitted as well and the correct specification of the confidentiality declarations.

DS_01_EstablishmentCount. Reports the total number of **Establishments** in the current data delivery.

DS_02_EstablishmentCountByTier. Reports the number of **Establishments** declared as Lower Tier and Upper Tier.

DS_03_EstablishmentCountByStatus. Reports the number of **Establishments** declaring different values of operationalStatus.

DS_04_SitesCount. Reports the number of distinct **ProductionSites** in the current data delivery.

DS_05_EstablishmentCountByConfidentialityFields. Reports the number of **Establishments** declaring confidentiality in **EstablishmentConfidentiality** for each confidentiality field.

DS_06_EstablishmentReportingSubstances. Reports the number of **Establishments** having at least one **Substance** declared.

DS_07_ConfidentialSubstanceCount. Reports the number of **Establishments** declaring at least one substance confidential and the number of confidentiality substance declarations.

DS_08_SecondaryIndustryDeclaration. Reports the number of **Establishments** declaring at least one NACE or SPIRS secondary industry type in **EstablishmentIndustryType**.

DS_09_SiteCountByConfidentialityFields. Reports the number of **Sites** having name or location implicitly confidential due to name or location confidentiality declarations made at underlying **Establishments**.

DS_10_SiteNameConfidentialityOrigin. Lists the **Establishments** having name confidentiality declarations making corresponding **Site** name confidential.

DS_11_SiteLocationConfidentialityOrigin. Lists the **Establishments** having location confidentiality declarations making corresponding **Site** location confidential.

5.8 Offline cross-delivery checks

Offline quality checks do not take place in the Reportnet 3 system and are only performed after the release of the dataflow, before its technical acceptance. Currently, all the checks that require cross-verifying data consistency with past submission are implemented as offline checks, mainly for data security reasons.

XC_01_EstablishmentStatus. Checks that the Establishment status is compatible with previous deliveries. Establishments reported with **decommissioned** status cannot be reported in subsequent years.

XC_02_EstablishmentLocation. Checks that the Establishment location is no more than 200 m from the location declared in the last submission.

XC_03_EstablishmentCity. Checks that the **city** in Establishment address matches the city declared in the last submission.

XC_04_EstablishmentPostCode. Checks that the Establishment address **postCode** matches the one declared in the last submission.

XC_05_EstablishmentNameConfidentiality. Checks that **establishmentName** is marked confidential if it was declared in the last submission.

XC_06_EstablishmentTierConfidentiality. Checks that **establishmentTier** is marked confidential if it was declared in the last submission.

XC_07_StatusConfidentiality. Checks that Establishment **status** is marked confidential if it was declared as such in the last submission.

XC_08_ParentCompanyConfidentiality. Checks that Establishment **parentCompany** is marked confidential if it was declared as such in the last submission.

XC_09_LocationConfidentiality. Checks that Establishment **location** is marked confidential if it was declared as such in the last submission.

XC_10_AddressConfidentiality. Checks that Establishment **address** is marked confidential if it was declared as such in the last submission.

XC_11_IndustryTypeConfidentiality. Checks that Establishment **industryType** is marked confidential if it was declared as such in the last submission.

XC_12_LastInspectionConfidentiality. Checks that Establishment **lastInspection** is marked confidential if it was declared as such in the last submission.

XC_13_CommentsConfidentiality. Checks that Establishment **comments** are marked confidential if they were declared as such in the last submission.

XC_14_NameSimilarity. Checks that current **establishmentName** and the one reported in the last submission do not have small differences that could be related to typos.

XC_15_ParentCompanySimilarity. Checks that current **parentCompany** and the one reported in the last submission do not have small differences that could be related to typos.

XC_16_LastInspectionDate. Checks that **lastInspectionDate** either matched the one reported in the last submission for a given Establishment or is more recent.

XC_17_mainSPIRSConsistency. Checks that the **mainIndustryTypeSPIRS** matches the one reported in the last submission.

XC_18_mainNACEConsistency. Checks that the **mainIndustryTypeNACE** matches the one reported in the last submission.

If any of the offline QCs fails, a document containing detailed description of the errors, reference to the Establishments affected and possible solutions is attached to the technical feedback. The document is uploaded as attachment to the technical feedback. More information on the technical feedback is provided in the Manual for Reporters.

All the cross-delivery checks are treated as **ERRORs**, countries need to confirm that the reported data are correct before the submission can be finalised as technically accepted.

Annex 1 – Glossary of Terms

This section contains explanations of the key terms used in this document.

Boolean attribute – These attributes are evaluated as either true or false.

Code list – a defined set of values which can be used to populate an attribute.

Data type – A data model element that defines both characteristics of data and which operations can be performed on the data.

Feature type – Represents a class of data together with relevant attributes.

Geospatial information – Data about a physical object that can be represented by numeric values in a geographic coordinate system.

INSPIRE – Infrastructure for Spatial Information in the European Community. The INSPIRE Directive (2007/2/EC) aims to establish an infrastructure for the sharing of environmental spatial data within the European Union. This will enable sharing among public sector organisations, facilitate public access to spatial data across Europe, and will aid in cross-boundary policy making.

Multiplicity – A definition of cardinality - i.e. the permitted number of elements - of some collection of elements.

NACE - The 'statistical classification of economic activities' in the European Community, abbreviated as NACE, is the classification of economic activities in the European Union (EU).

Production Site – Represents the geographical location of the Seveso Establishment or a piece of land where the facility was, is, or is intended to be located.

Seveso Establishment – Represents the whole location under the control of an operator where dangerous substances are present in one or more installations, including common or related infrastructures or activities according to DIRECTIVE 2012/18/EU.