



BSI Standards Publication

Reporting in support of supervision of online gambling services by the gambling regulatory authorities of the Member States

National foreword

This British Standard is the UK implementation of EN 17531:2021+A1:2023. It supersedes BS EN 17531:2021, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/33/-/7, Reporting in support of online gambling supervision.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 28932 9

ICS 97.200.99; 35.240.99; 03.160

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 January 2022.

Amendments/corrigenda issued since publication

Date	Text affected
31 October 2023	Implementation of CEN amendment A1:2023

English Version

Reporting in support of supervision of online gambling services by the gambling regulatory authorities of the Member States

Remontées d'informations à l'appui de la surveillance des services de jeux d'argent et de hasard en ligne par les autorités de régulation des jeux en ligne des États membres

Berichterstattung zur Unterstützung der Aufsicht über Online-Glücksspiele durch die Glücksspielaufsichtsbehörden der Mitgliedstaaten

This European Standard was approved by CEN on 25 July 2021 and includes Amendment approved by CEN on 2 August 2023.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Contents

Page

European foreword	6
Introduction	7
1 Scope.....	8
2 Normative references.....	8
3 Terms and definitions.....	8
4 Symbols and Abbreviated Terms	8
5 Clause title, e.g. Paragraphs and Lists	9
5.1 General.....	9
5.2 Reporting Model.....	9
5.3 Data Transport Security	9
5.4 Model for Funds Movement.....	10
5.5 Periodic and Near-Real-Time Reports	11
5.6 Reporting Requirements.....	13
6 Common Data Types	14
6.1 Complex Data Types.....	14
6.2 Simple Data Types	19
7 Constructing Document Files for Reporting.....	26
7.1 Introduction.....	26
7.2 File Naming Convention.....	27
7.3 XML Document Structure	27
7.4 Document Types.....	29
7.5 Simple Data Types	34
8 Reporting on Player Registration	35
8.1 Introduction.....	35
8.2 Periodic & Near-Real-Time Reports.....	35
8.3 Complex Data Types.....	37
8.4 Simple Data Types	47
9 Reporting on Player Wallets	53
9.1 Introduction.....	53
9.2 Funds Movement Model.....	53
9.3 Periodic & Near-Real-Time Reports.....	54
9.4 Player Activities.....	56
9.5 Complex Data Types.....	59
9.6 Simple Data Types	60
10 Reporting on Jackpots.....	62
10.1 Introduction.....	62
10.2 Funds Movement Model.....	62
10.3 Periodic & Near-Real-Time Reports.....	63
10.4 Jackpot Activities	70
10.5 Complex Data Types.....	72
10.6 Simple Data Types	74
11 Reporting on Online Games.....	76

11.1	Introduction.....	76
11.2	Game Organization.....	78
11.3	Periodic & Near-Real-Time Reports	78
11.4	Complex Data Types	95
11.5	Simple Data Types.....	97
12	Reporting on Online Casino Games.....	98
12.1	Overview.....	98
12.2	Game Organization.....	98
12.3	Usage Conventions	99
12.4	Funds Movement Model	99
12.5	Other Metrics.....	100
12.6	Game Configuration Detail (gameConfigDetail)	100
12.7	Game Activity Detail (gameActivityDetail).....	102
12.8	Game Results Detail (gameResultsDetail)	104
12.9	Game Summary Detail (gameSummaryDetail)	106
12.10	Game Funds-In-Play Detail (gameInPlayDetail).....	107
13	Reporting on Online Casino Tournaments.....	108
13.1	Overview.....	108
13.2	Game Organization.....	109
13.3	Usage Conventions	109
13.4	Funds Movement Model	110
13.5	Other Metrics.....	110
13.6	Game Configuration Detail (gameConfigDetail)	111
13.7	Game Activity Detail (gameActivityDetail).....	111
13.8	Game Results Detail (gameResultsDetail)	116
13.9	Game Summary Detail (gameSummaryDetail)	117
13.10	Game Funds-In-Play Detail (gameInPlayDetail).....	118
14	Reporting on Online Fixed-odds Betting.....	119
14.1	Overview.....	119
14.2	Game Organization.....	120
14.3	Usage Conventions	120
14.4	Funds Movement Model	120
14.5	Other Metrics.....	121
14.6	Game Configuration Detail (gameConfigDetail)	121
14.7	Game Activity Detail (gameActivityDetail).....	121
14.8	Game Results Detail (gameResultsDetail)	135
14.9	Game Summary Detail (gameSummaryDetail)	136
14.10	Game Funds-In-Play Detail (gameInPlayDetail).....	138
14.11	Complex Data Types	139
14.12	Simple Data Types.....	139
15	Reporting on Online Betting Exchanges.....	141
15.1	Overview.....	141
15.2	Game Organization.....	142
15.3	Usage Conventions	142
15.4	Funds Movement Model	142
15.5	Other Metrics.....	143
15.6	Game Configuration Detail (gameConfigDetail)	144
15.7	Game Activity Detail (gameActivityDetail).....	144
15.8	Game Results Detail (gameResultsDetail)	153
15.9	Game Summary Detail (gameSummaryDetail)	155
15.10	Game Funds-In-Play Detail (gameInPlayDetail).....	156

15.11	Complex Data Types.....	157
15.12	Simple Data Types	157
16	Reporting on Pari-Mutuel Betting.....	158
16.1	Overview	158
16.2	Game Organization.....	159
16.3	Usage Conventions.....	159
16.4	Funds Movement Model.....	160
16.5	Other Metrics.....	160
16.6	Game Configuration Detail (gameConfigDetail)	161
16.7	Game Activity Detail (gameActivityDetail)	161
16.8	Game Results Detail (gameResultsDetail)	176
16.9	Game Summary Detail (gameSummaryDetail)	177
16.10	Game Funds-In-Play Detail (gameInPlayDetail)	179
16.11	Complex Data Types.....	180
16.12	Simple Data Types	180
17	Reporting on Online Poker Cash Games	181
17.1	Overview	181
17.2	Game Organization.....	182
17.3	Usage Conventions.....	182
17.4	Funds Movement Model.....	182
17.5	Other Metrics.....	183
17.6	Game Configuration Detail (gameConfigDetail)	184
17.7	Game Activity Detail (gameActivityDetail)	184
17.8	Game Results Detail (gameResultsDetail)	193
17.9	Game Summary Detail (gameSummaryDetail)	194
17.10	Game Funds-In-Play Detail (gameInPlayDetail)	196
17.11	Complex Data Types.....	197
17.12	Simple Data Types	197
18	Reporting on Online Poker Tournaments	199
18.1	Overview	199
18.2	Game Organization.....	199
18.3	Usage Conventions.....	200
18.4	Funds Movement Model.....	200
18.5	Other Metrics.....	201
18.6	Game Configuration Detail (gameConfigDetail)	202
18.7	Game Activity Detail (gameActivityDetail)	202
18.8	Game Results Detail (gameResultsDetail)	209
18.9	Game Summary Detail (gameSummaryDetail)	210
18.10	Game Funds-In-Play Detail (gameInPlayDetail)	212
18.11	Complex Data Types.....	213
18.12	Simple Data Types	214
19	Reporting on Online Fantasy (Manager) Games.....	214
19.1	Overview	214
19.2	Game Organization.....	215
19.3	Usage Conventions.....	215
19.4	Funds Movement Model.....	215
19.5	Other Metrics.....	216
19.6	Game Configuration Detail (gameConfigDetail)	216
19.7	Game Activity Detail (gameActivityDetail)	217
19.8	Game Results Detail (gameResultsDetail)	222
19.9	Game Summary Detail (gameSummaryDetail)	223

19.10	Game Funds-In-Play Detail (gameInPlayDetail).....	224
20	Reporting on Online Bingo Games.....	226
20.1	Overview.....	226
20.2	Game Organization.....	226
20.3	Usage Conventions	227
20.4	Funds Movement Model	227
20.5	Other Metrics.....	228
20.6	Game Configuration Detail (gameConfigDetail)	229
20.7	Game Activity Detail (gameActivityDetail).....	229
20.8	Game Results Detail (gameResultsDetail)	236
20.9	Game Summary Detail (gameSummaryDetail)	238
20.10	Game Funds-In-Play Detail (gameInPlayDetail).....	239
21	Reporting on Online Numbers Games.....	241
21.1	Overview.....	241
21.2	Game Organization.....	242
21.3	Usage Conventions	242
21.4	Funds Movement Model	243
21.5	Other Metrics.....	243
21.6	Game Configuration Detail (gameConfigDetail)	244
21.7	Game Activity Detail (gameActivityDetail).....	244
21.8	Game Results Detail (gameResultsDetail)	257
21.9	Game Summary Detail (gameSummaryDetail)	258
21.10	Game Funds-In-Play Detail (gameInPlayDetail).....	260
22	Reporting on Other Activities.....	261
22.1	Introduction.....	261
22.2	Periodic & Near-Real-Time Reports	261
22.3	Simple Data Types	269
	Bibliography	271

European foreword

This document (EN 17531:2021+A1:2023) has been prepared by Technical Committee CEN/TC 456 “Reporting in support of online gambling supervision”, the secretariat of which is held by AFNOR.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by April 2024, and conflicting national standards shall be withdrawn at the latest by April 2024.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

This document supersedes EN 17531:2021.

This document includes Amendment 1 approved by CEN on 02 August 2023.

The purpose of this consolidated amendment is the circulation of the additional files which are referred to in 5.6.

This document has been prepared under a mandate given to CEN by the European Commission and the European Free Trade Association.

Any feedback and questions on this document should be directed to the users’ national standards body. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organisations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Introduction

This document was developed in response to a European Commission standardisation request as regards a European standard on reporting in support of supervision of online gambling services by the gambling regulatory authorities of the Member States of 4 April 2018.

This document seeks to identify the data elements needed by Member States' regulatory authorities for online gambling supervision purposes. However, it recognizes that some Member States **may** need to add data elements that are unique to, or required in, their regulation of online gambling. Further, in some Member States, regulatory authorities **may** not be permitted to collect certain data, such as player personal information. Therefore, some clauses in this standard **may** not be applicable to some Member States. It is the responsibility of the operators and suppliers to ensure compliance with the requirements of each Member State.

Additionally, the document seeks to provide regulatory authorities with access to data reports that will support the achievement of their objectives of public policy, in particular consumer protection, operational transparency, game fairness, and the detection and prevention of fraud and betting-related match-fixing.

Given the divergent regulation of all forms of gambling and in the absence of harmonisation, the document aims at minimising administrative burden for regulatory authorities, operators and suppliers resulting from compliance with different regulatory reporting requirements through a voluntary standard.

1 Scope

The development of (a) European standard(s) on reporting by online gambling service operators and suppliers to the gambling regulatory authorities in the Member States for the purpose of supervision of online gambling services standard (referred to within the specification as OGR or Online Gambling Reporting).

It will provide a voluntary tool to the gambling regulatory authorities in the Member States, without prejudice to the competence of Member States in the regulation of online gambling. It does not impose any obligation on them to introduce or alter reporting requirements or to authorise or deny authorisation to any operators or suppliers for example where the national gambling legislation imposes other rules. Member States remain competent to define for which games reporting should take place. The scope of reporting is in accordance with the applicable legislation in the Member State where the operator is licensed and where services are offered to consumers.

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/IEC 9834-8:2014, *Information technology — Procedures for the operation of object identifier registration authorities — Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers*

ISO 20022 (Series), *Financial services — Universal financial industry message scheme*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country code*

ISO 3166-2, *Codes for the representation of names of countries and their subdivisions — Part 2: Country subdivision code*

ISO 4217, *Codes for the representation of currencies*

ISO 5218, *Information Interchange — Representation of Human Sexes*

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

RFC 2822, *Internet Message Format*

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

4 Symbols and Abbreviated Terms

The following abbreviated term is used within this document.

OGR Online Gambling Reporting a shortened version of the standard title Online Gambling: Reporting in Support of Supervision

5 Clause title, e.g. Paragraphs and Lists

5.1 General

In alignment with the European Commission mandate, this document's specification uses XML as the extensible mark-up language. The specification is developed using a set of rules and conventions that apply throughout. For ease of implementation by all types of enterprises and regulatory authorities an XSD is provided.

Within this document, many elements are identified as “optional”. Similarly, 0 (zero) instances of many elements are allowed. This simply means that the specific elements are not required within reports for the reports to be syntactically correct.

Regulators **may** require that operators report some or all these elements to meet the semantic requirements of the jurisdiction. Operators should be sure to ask their local regulators which of these elements are required within the regulator’s jurisdiction. Similarly, regulators **may** specify default values for the optional elements – that is, the semantic meaning of the optional element if it is omitted.

As a convention any data element that does not have a value **shall** be <empty> or omitted from the reporting.

5.2 Reporting Model

Two modes of reporting are supported by this standard: periodic reporting and near-real-time reporting.

- **Periodic** reporting is the primary mode of reporting within this standard. It **may** be the only mode of reporting used in many jurisdictions. In this mode, gaming activity is summarized and reported after the fact, typically on a daily and/or monthly basis. The information reported in this mode includes game-play summaries, funds-in-play summaries, player account activity summaries, and jackpot activity summaries.
- **Near-real-time** reporting is the secondary mode of reporting. It **may** be used when a greater level of detail is required. In this mode, gaming activity is also reported after the fact, but at a much greater level of detail and at a much greater frequency than with periodic reporting – for example, reporting sports bets immediately after they have been placed. Near-real-time reports are designed to provide the detail behind the periodic reports. The types of information reported in this mode include game-play results, player account movements, and jackpot movements. Even though these reports are referred to as near-real-time, the frequency can be much longer depending on the needs of the jurisdiction – for example, sports bets can be reported hourly or daily if that is the preferred frequency.

All reports are designed to be internally consistent with one another so that the reported data itself can be audited and reconciled. Near-real-time reports can be reconciled against periodic reports. Different periodic reports can be reconciled against one another. Ending balances can be reconciled against opening balances.

Not all reports **may** be needed by a specific jurisdiction – for example, if a jurisdiction does not audit player registration information, the jurisdiction might not need reports containing that information. OGR is designed so that jurisdictions can select the modes, as well as the types of data, most appropriate to their needs.

5.3 Data Transport Security

It is expected that standard computing data delivery security mechanisms, for example HTTPS, VPNs, and standard encryption methodologies, for example TLS, PGP or AES, will be used for data in transit, that is for the data files being transferred between Operator and Regulator systems. However, data at rest, i.e. data stored within the Operator’s and Regulator’s systems and how it is protected, is outside the scope of

this document. To address data integrity, a standard mechanism, for example digital signatures, may be used.

5.4 Model for Funds Movement

The primary funds movement model used in this document is similar to double-entry bookkeeping (see Figure 1). For every debit, there is an offsetting credit. For example, if there is a debit (deduction) to a player account for a wager, there is also an offsetting credit to the game for the wager. Player accounts, games, and jackpots are viewed as bookkeeping accounts. Funds move from one account to another. The total debits always equal the total credits. Within the specification, credits are represented as positive values; debits are represented as negative values. Positive values increase the value of an account; negative values decrease the value of an account.

The only exception to this rule is funds movements to/from external sources – for example, deposits to player wallets or adjustments to jackpots. Within this document, there are no offsetting entries for those types of funds movements. Thus, those types of funds movements cannot be reconciled against other offsetting funds movements reported with this document. Instead, those types of funds movements must be reconciled against external sources of information – for example, statements from banks or payments processors.

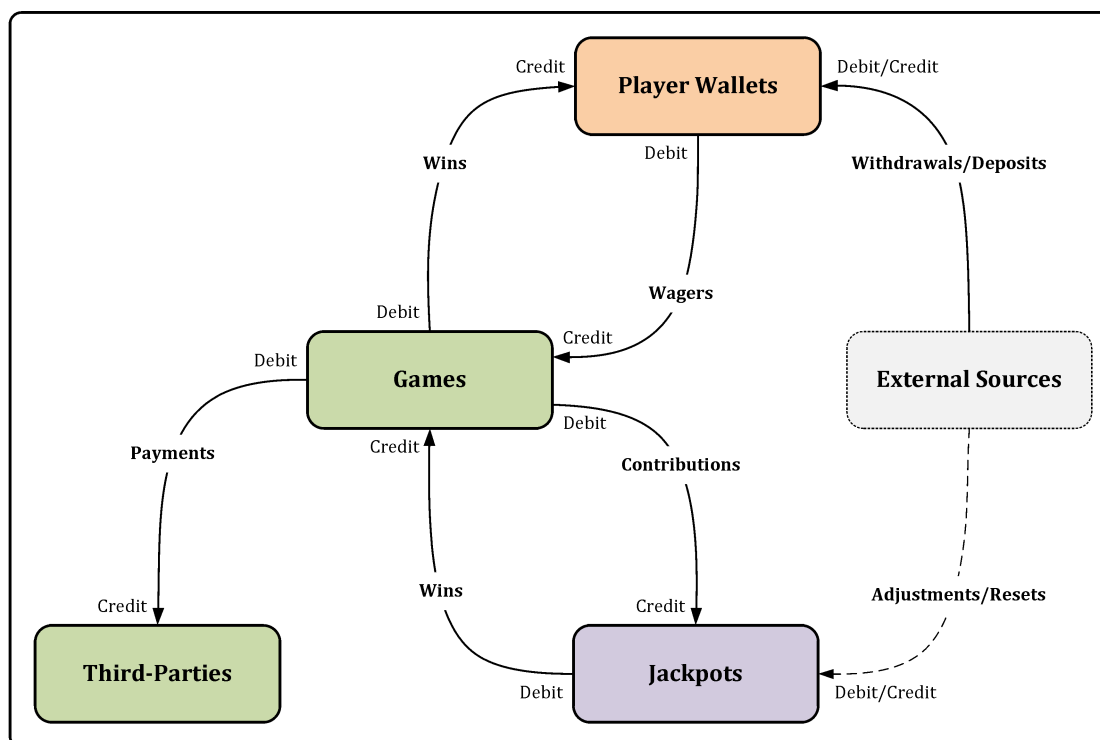


Figure 1

This model is referred to as the Online Reporting Model within this document.

The sections of the specification that cover specific game categories – that is, fixed-odds betting, pari-mutuel betting, poker cash games, poker tournaments, etc. – include information about the recommended accounting models for the game categories. This information includes the list of transaction types (also known as metrics or meters) used within the accounting model.

The sections of the specification related to player accounts and jackpots contain the recommended accounting models for external funds movements to/from player accounts and jackpots. Funds movements associated with game play are addressed in the pertinent sections related to specific game

categories; external funds movements, which are not associated with game play, are addressed in the sections related to player accounts and jackpots.

5.5 Periodic and Near-Real-Time Reports

The following table provides a summary of the primary periodic and near-real-time (NRT) reports supported by the standard. The data structures for the reports, as well as the near real-time transactions that support them, are fully described in the pertinent sections of the specification. Other secondary reports **may** also be described in those sections.

A common set of headers is used with all reports defined within the specification. The headers act as the outer wrapper for the reports. They identify the contents of the reports. The headers are fully described in the section of the specification on constructing reports.

Some reports can be sent as near-real-time reports or as periodic reports – for example, Player Registration reports can be sent in near-real-time as players are being registered or on a periodic basis to report all players with changes to their registration information for the period or to report all players who were active during the period. The descriptions of the individual reports (see Table 1) indicate whether a report can be sent in near-real-time, periodically, or both. The exact requirements will be determined by the jurisdictions requiring the reports.

Table 1 — Periodic and Near-Real-Time Reports

Report	Section	Mode	Description
Player Registration	Player Registration	NRT or Periodic	Contains new player registration information as well as updates to that information.
Player Activity	Player Accounts	NRT or Periodic	Contains changes to player account balances from sources other than game-play – for example, deposits, adjustments, etc.
Player Balance	Player Accounts	Periodic	Contains opening balances, closing balances, and a summary of transactions affecting those balances for the period grouped by player (playerId).
Jackpot Activity	Jackpots	NRT or Periodic	Contains records of changes to jackpot balances from sources other than game-play – for example, adjustments, jackpot resets, etc.
Jackpot Balance	Jackpots	Periodic	Contains opening balances, closing balances, and a summary of transactions affecting those balances for the period grouped by jackpot (jackpotControllerId).
Jackpot Interval	Jackpots	Periodic	Contains jackpot balances captured at intervals during the period.
Game Activity	Online Games	NRT or Periodic	Contains records of individual game-play activities – for example, wagers, winnings, cancellations, etc.
Game Results	Online Games	NRT or Periodic	Contains records of completed games including wagers, wins, jackpot contributions, and third-party payments.
Game Summary	Online Games	Periodic	Contains a summary of funds movements by game for the period.
Player Summary	Online Games	Periodic	Contains a summary of funds movements by player for the period.
Jackpot Summary	Online Games	Periodic	Contains a summary of funds movements by jackpot for the period.
Payment Summary	Online Games	Periodic	Contains a summary of funds movements by payee for the period.

Report	Section	Mode	Description
Game Funds-In-Play	Online Games	Periodic	Contains a summary of wagers, wins, jackpot contributions, and third-party payments for games that were not completed at the end of the period grouped by game (gameId).
Player Funds-In-Play	Online Games	Periodic	Contains a summary of wagers and wins for games that were not completed at the end of the period grouped by player (playerId).
Jackpot Funds-In-Play	Online Games	Periodic	Contains a summary of jackpot contributions and wins for games that were not completed at the end of the period grouped by jackpot (jackpotControllerId).
Payee Funds-In-Play	Online Games	Periodic	Contains a summary of third-party payments for games that were not completed at the end of the period grouped by payee (payeeId).

5.6 Reporting Requirements

5.6.1 XML Encoding

Reports created following this document **shall** be encoded using Extensible Markup Language (XML). To aid developers in generating programs that create or consume the reports, XML schemas accompany this document. The schemas conform to the XML Schema Definition (XSD) standard. All reports created following this standard specification **shall** be able to pass validation against the XML schemas. If there is a discrepancy between this document and the XML schemas, the XML schemas take precedence.

The XML schemas are intended to be extensible so that jurisdiction-specific requirements can be easily met while still working within the framework of this document without requiring the submission of elements that would be unused by the jurisdiction. XML Namespaces

The XML schemas that accompany this document use XML namespaces to help organize the schemas into logical units. These namespaces are an integral part of the XML schemas as well as this document. The XML namespaces in the XML schemas **shall** NOT be modified. To avoid ambiguity, XML namespaces and prefixes **shall** be used in reports that are intended to be compliant with this document.

The XML namespaces are constructed as Uniform Resource Names (URN) using the following identifiers (see Table 2) separated by colons (:). When extending the XML schemas, the same methodology should be used for creating new namespaces.

Table 2 — Namespace Identifiers

Identifier	Description	Example
URN	Indicates that the XML namespace value is a URN.	urn
Domain	Identifies the overall domain to which the XML namespace belongs.	CEN
Protocol	Identifies the protocol with which the XML namespace is used.	OGR
Version	Identifies the version of the protocol in which the namespace was introduced.	1.0
Author	Identifies the author or creator of the XML namespace.	CEN
Section	Identifies section of the protocol or extension with which the namespace is associated.	ogr

The following example demonstrates the construction of an XML namespace using this methodology.

urn:CEN:OGR:1.0:CEN:ogr

The first three identifiers – URN, Domain, and Protocol – **shall** be the same for all XML namespaces used with this document. The remaining identifiers – Version, Author and Section – will vary between XML namespaces.

5.6.2 Case Sensitivity

Unless specified differently in an underlying specification, all protocol-defined constructs described within this document are case-sensitive including element names, sub-element names, enumeration values, code values, identifier values, data values, etc. Uppercase letters **shall** NOT be considered equivalent to lowercase letters. For example, "ABC" **shall** not be considered equivalent to "abc".

6 Common Data Types

6.1 Complex Data Types

6.1.1 Introduction

The following complex data types are used in multiple sections of the specification. Other complex data types, which are not described in this section of the specification, are described in those sections.

6.1.2 Amount Currency (c_amountCurrency)

The `c_amountCurrency` data type (see Table 3) is used to report both the value of a transaction and the currency in which the transaction took place.

Table 3 — c_amountCurrency Element

Element	Restrictions	Description
amount	type: <code>t_money</code> minOcc: 1 maxOcc: 1	Monetary amount.
currency	type: <code>t_currencyCode</code> minOcc: 0 maxOcc: 1	Currency code.

6.1.3 Funds Movement (c_fundsMovement)

The `c_fundsMovement` data type (see Table 4), is used to report funds movements amongst player wallets, games, jackpots, and third-party payees as well as external funds movements to/from player wallets and jackpots. This set of elements includes the Movement Type, the value of the movement, and the Currency Code.

Players, games, jackpots, and third-party payees are considered accounts. A Player account **may** be comprised of multiple wallets. Funds move in and out of the accounts and, in the case of a Player's account, the wallet(s). Credits increase the value of the accounts; debits decrease the value. When funds movements are reported, for every credit there is an offsetting debit; the total credits always equal the total debits. Within the specification, credits are represented as positive values; debits are represented as negative values. Positive values increase the value of an account; negative values decrease the value of an account.

When funds are converted from a foreign currency to the base currency of an account as part of a funds movement (or vice versa), two additional elements are used – the value of the movement in the foreign currency and the Currency Code of the foreign currency. When needed, the exchange rate from the foreign currency to the base currency of the account can be calculated by dividing the value in the base currency by the value in the foreign currency.

Table 4 — c_fundsMovement Elements

Elements	Data Type	Description
walletId	type: t_description minOcc: 0 maxOcc: 1	Wallet ID; unique identifier issued to each of a player's wallets
transactionId	type: t_description minOcc: 1 maxOcc: 1	Transaction ID; unique identifier assigned to each specific transaction.
transactionClass	type: t_transactionClass minOcc: 0 maxOcc: 1 default: OGR_online	Transaction Class; the domain to which the Transaction Type belongs.
transactionType	type: t_transactionType minOcc: 1 maxOcc: 1	Transaction Type; the type of funds movement; for example, stake, win, deposit, etc.
fundsType	type: t_fundsType minOcc: 1 maxOcc: 1	Funds Type; data type is used to identify the type of funds in an account.
transactionAmount	type: t_money minOcc: 1 maxOcc: 1	Transaction Amount; the value of the funds movement; debits shall be reported as negative values; credits shall be reported as positive values.
transactionDescription	type: t_description minOcc: 0 maxOcc: 1	Transaction Description; field to capture the nature of the award given to the player.
currency	type: t_currencyCode minOcc: 0 maxOcc: 1	Currency Code; currency in which the funds movement was recorded in the account.
foreignAmount	type: t_money minOcc: 0 maxOcc: 1	Foreign Amount; the value of the funds movement in the Foreign Currency.
foreignCode	type: t_currencyCode minOcc: 0 maxOcc: 1 default: XXX	Foreign Currency; foreign currency associated with the funds movement; XXX indicates that no foreign exchange took place.

6.1.4 Funds Movements (c_fundsMovementList)

The `c_fundsMovementList` data type (see Table 5) is used to report funds movements. Because there **may** be multiple funds movements as the result of a single reported event, funds movements are reported as lists.

Table 5 — c_fundsMovementList Element

Element	Data Type	Description
fundsMovement	type: <code>c_fundsMovement</code> minOcc: 0 maxOcc: ∞	Contains a funds movement; see Funds Movement (<code>c_fundsMovement</code>) for more details.

6.1.5 Game (c_game)

The `c_game` data type (see Table 6), is used to identify a specific Game offered by an operator. This set of elements includes the Operator Identifier, and Game Category as well as a series of other elements that describe the general characteristics of the game.

Table 6 — c_game Elements

Elements	Restrictions	Description
jurisdictionCode	type: <code>t_jurisdictionCode</code> minOcc: 0 maxOcc: 1	Jurisdiction Code; jurisdiction that assigned the Operator Identifier.
operatorId	type: <code>t_description</code> minOcc: 1 maxOcc: 1	Operator Identifier; operator that assigned the Game Identifier.
gameId	type: <code>t_description</code> minOcc: 0 maxOcc: 1	Game Identifier. Typically required unless it is being used for summarization by game category in specific reports required by a regulator.
gameName	type: <code>t_description</code> minOcc: 0 maxOcc: 1	Game Name (title).
gameCategory	type: <code>t_gameCategory</code> minOcc: 1 maxOcc: 1	Game Category; category of the Game.
gameType	type: <code>t_gameType</code> minOcc: 0 maxOcc: 1	Game Type; type of game within the Game Category.

6.1.6 Jackpot Controller (c_jackpotController)

The `c_jackpotController` complex data type (see Table 7) is used throughout this document to identify a specific Jackpot Controller. This set of elements includes the Operator Identifier, Jackpot Server Identifier, and Jackpot Controller Identifier. This combination is intended to be globally unique.

Table 7 — c_jackpotController Elements

Elements	Restrictions	Description
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Jurisdiction Code; jurisdiction that assigned the Operator Identifier.
operatorId	type: t_description minOcc: 1 maxOcc: 1	Operator Identifier; operator that assigned the Jackpot Server Identifier and Jackpot Controller Identifier.
jackpotServerId	type: t_description minOcc: 1 maxOcc: 1	Jackpot Server Identifier.
jackpotControllerId	type: t_description minOcc: 1 maxOcc: 1	Jackpot Controller Identifier.

6.1.7 Operator (c_operator)

The `c_operator` data type (see Table 8) is used to identify a specific operator. The Operator Identifier is intended to be jurisdictionally unique.

Table 8 — c_operator Elements

Elements	Restrictions	Description
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Jurisdiction Code; jurisdiction that issued the Operator Identifier. In gaming jurisdictions that use this element, the combination of Jurisdiction Code and Operator Identifier is intended to be unique.
operatorId	type: t_description minOcc: 1 maxOcc: 1	Operator Identifier; intended to be unique within a gaming jurisdiction

6.1.8 Payee (c_payee)

The `c_payee` data type (see Table 9) is used to identify a specific third-party payee of an operator. It includes the Operator Identifier, Payee Identifier, and the name of the Payee. The combination of Operator Identifier, and Payee Identifier is intended to be jurisdictionally unique.

Table 9 — c_payee Elements

Elements	Restrictions	Description
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Jurisdiction Code; jurisdiction that issued the Operator Identifier. In jurisdictions that use this element, the combination of jurisdictionCode, operatorId and payeeId is intended to be unique.
operatorId	type: t_description minOcc: 1 maxOcc: 1	Operator Identifier; operator that assigned the Payee Identifier.
payeeId	type: t_description minOcc: 1 maxOcc: 1	Payee Identifier; intended to be unique within each Operator's system.
payeeName	type: t_description minOcc: 0 maxOcc: 1	Payee Name.

6.1.9 Player (c_player)

The `c_player` complex data type (see Table 10) is used to identify a specific player. The set of elements includes the Operator Identifier, and the Player Identifier. This combination is intended to be globally unique.

Table 10 — c_player Elements

Elements	Data Type	Description
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Player Jurisdiction Code; jurisdiction that issued the Player Identifier.
operatorId	type: t_description minOcc: 1 maxOcc: 1	Operator Identifier; operator that issued the Player Identifier.
playerId	type: t_description minOcc: 1 maxOcc: 1	Player Identifier.

6.1.10 Supplier (c_supplier)

The `c_supplier` data type (see Table 11) is used to identify a specific Supplier (manufacturer). It includes the Operator Identifier, Supplier Identifier, and the name of the Supplier. The combination of Jurisdiction Code, Operator Identifier, and Supplier Identifier is intended to be globally unique.

Table 11 — c_supplier Elements

Elements	Restrictions	Description
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Jurisdiction Code; jurisdiction that assigned the Operator Identifier.
operatorId	type: t_description minOcc: 1 maxOcc: 1	Operator Identifier; operator that assigned the Supplier Identifier.
supplierId	type: t_description minOcc: 1 maxOcc: 1	Supplier Identifier.
supplierName	type: t_description minOcc: 0 maxOcc: 1	Supplier Name.

6.2 Simple Data Types

6.2.1 Introduction

The following simple data types (see Table 12) are used in multiple sections of the OGR specification. Other simple data types, which are not described in this section of the OGR specification, are described in those sections.

6.2.2 Simple Data Type Table

Table 12 — Simple Data Types

Data Type	Restrictions	Description
t_channelType	type: t_uniqueId	Channel Types; the distribution channels through which a website was accessed. See Channel Type (t_channelType) for more details.
t_countryCode	type: xs:string maxLen: 2	Country code; shall conform to ISO 3166-1 alpha-2 standard (2 characters); for example, ES.
t_currencyCode	type: xs:string maxLen: 3	Currency Code; shall conform to ISO 4217; for example, EUR; see Currency Code (t_currencyCode) for more details.
t_date	type: xs:date	Date; shall comply with the full-date format specified in RFC 3339.
t_dateTime	type: xs:dateTime	Date/Time; shall comply with the date-time format specified in RFC 3339, which requires that time be in UTC ("Z") or that a time-zone offset be included.
t_description	type: xs:string maxLen: 128	Description; 128-characters.

Data Type	Restrictions	Description
t_extraInfo	type: xs:string maxLen: 1024	Extra Information; 1024-characters.
t_fundsType	type: t_uniqueId	Funds Type; see Funds Type (t_fundsType) for more details.
t_gameCategory	type: t_uniqueId	Game Category; see Game Organisation (t_gameCategory and t_gameType) for more details.
t_gameType	type: t_uniqueId	Game Type; see Game Organisation (t_gameCategory, t_gameType) for more details.
t_jurisdictionCode	type: xs:string maxLen: 8	Jurisdiction code; see Jurisdiction Code (t_jurisdictionCode) for more details.
t_liquidityType	type: t_uniqueId	Liquidity Type; see Liquidity Type (t_liquidityType) for more details.
t_money	type: xs:decimal fracDig: 7	Monetary Value; see Monetary Values (t_money) for more details.
t_operatorType	Type: t_uniqueId	Operator Type; The type of operator being reported for; see Operator Type (t_operatorType) for more details.
t_percent	type: xs:decimal fracDig: 6	Percentage; expressed as a percentage; for example, 87.45 % is expressed as 87.45.
t_periodType	type: t_uniqueId maxLen: 1	Period Type.
t_quantity	type: xs:long	Quantity.
t_time	type: xs:time	Time; shall comply with the partial-time format specified in RFC 3339.
t_transactionClass	type: t_uniqueId	Transaction Class; see Transaction Class (t_transactionClass) for more details.
t_transactionType	type: t_uniqueId	Transaction Type; see Transaction Type (t_transactionType) for more details.
t_uniqueId	type: xs:string pattern: OGR_[a-zA-Z0-9]{1,39}	Unique Identifier; 43-characters.
t_UUID	type: xs:string maxLen: 36	Universally unique identifier; may comply with ISO/IEC 9834-8:2014, or other methods may be used.

6.2.3 Channel Type (t_channelType)

The `t_channelType` data type (see Table 13) identifies the type of distribution channel used by the player to access the operator's website – for example, tablet, mobile, etc. The following table contains the Channel Types supported within this document. Other Channel Types **may** be added in future versions of this document.

Table 13 — t_channelType Enumerations

Enumeration	Description
OGR_desktop	Desktop or laptop devices.
OGR_tablet	Tablet devices.
OGR_mobile	Mobile phones.
OGR_terminal	Terminal devices.
OGR_other	Other channel.

6.2.4 Currency Codes (t_currencyCode)

The `t_currencyCode` data type identifies the currency in which a monetary value is represented. Currency Codes **shall** be selected from the set of currency codes defined in ISO 4217. That standard describes the 3-character codes associated with each currency as well as the number of digits to the right of the decimal place that are used for the minor unit of the currency. For example, the Euro is identified using the currency code EUR and uses two digits to the right of the decimal place for cents.

6.2.5 Funds Type (t_fundsType)

The `t_fundsType` data type (see Table 14) is used to identify the type of funds in a wallet. Within a wallet, a player **may** have more than one balance. For example, the player **may** have a real-money balance containing funds deposited and won by the player as well as a bonus-money balance containing promotional funds awarded by the operator. Each balance is identified by a Funds Type.

There **may** be more than one balance for each Funds Type within a wallet. For example, some real-money funds might be available for play while other real-money funds might be blocked waiting for a deposit to be approved.

Typically, wagers are recorded as real-money or bonus-money; wins are recorded as real-money or in-kind. Other scenarios are possible.

This information **may** be critical to the calculation of taxable revenue. For example, a jurisdiction **may** allow the operator to exclude bonus-money wagers from the calculation of taxable revenues. The information **may** also be critical to the calculation of player loss in responsible gaming applications. For example, a jurisdiction **may** choose to exclude bonus-money wagers from the calculation of player loss.

The following table contains the Funds Types supported within this document. Other Funds Types **may** be added in future versions of this document.

Table 14 — t_fundsType Enumerations

Enumerations	Description
OGR_cashable	Real-Money; funds that are equivalent to cash.
OGR_noncashable	Real-Money – funds that cannot be cashed out but which must be wagered.
OGR_promotional	Bonus-Money; funds that were awarded for promotional purposes.
OGR_inKind	In-Kind Prize; winnings that were awarded as in-kind prizes – for example, a car or a boat; typically, excluded from the balance in a player’s wallet.
OGR_other	Other funds type.

6.2.6 Game Organisation (t_gameCategory, t_gameType)

Within this document, games are organised by Game Category and Game Type.

- The Game Category (`t_gameCategory`) identifies the broad category of games to which a specific game belongs – for example, poker cash games, casino games, fixed-odds betting, etc. This identifier is critical to interoperability when using this document. In many reports, it determines which sub-elements are included in the reports. Different sub-elements are included for different Game Categories. It is also used to organize games into broad categories for summarization and reporting.
- The Game Type (`t_gameType`) data type identifies a specific type of game within a Game Category – for example, roulette, blackjack, baccarat, etc. This identifier is important for cross-operator uses.

The enumerations for Game Category and Game Type are contained in the sections of the OGR specification that address online game play.

There are substantial dependencies on Game Category within this document. Other Game Categories should not be added or substituted without carefully considering the implications of the additions or substitutions on this document.

6.2.7 Jackpot Server Identifier (t_description)

The `jackpotServerId` data type is used to identify a specific progressive jackpot server within the gaming network. See Jackpot Controller Identifier (`t_description`) for more details. The administrator of the gaming network is responsible for assigning a Jackpot Server Identifier to each jackpot server. To avoid conflicts, the administrator should assign a unique Jackpot Server Identifier to each jackpot server.

The following Figure 2 illustrates the relationships between jackpot servers, jackpot controllers, and jackpots levels.

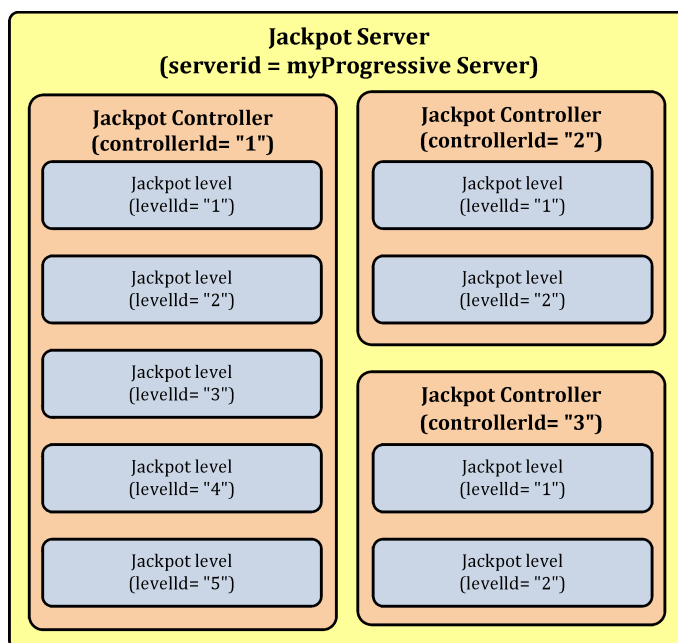


Figure 2

6.2.8 Jackpot Controller Identifier (t_description)

The `jackpotControllerId` data type is used to identify a specific set of jackpots that is being managed by a jackpot server. A jackpot server **may** support more than one logical controller. Each controller **may** contain one or more jackpot levels (sometimes referred to as pools or, simply, jackpots). For example, a controller **may** have a top prize and a smaller secondary prize. The jackpot server is responsible for assigning a unique Jackpot Controller Identifier to each logical controller that it manages.

Contributions to the jackpot levels are based on qualifying wagers from games that are linked to the controller. The method for calculating the contributions and distributing the contributions to individual jackpot levels is determined by the jackpot server. For example, the jackpot server **may** contribute 1.5 % of all qualifying wagers to the top prize and 0.5% to a secondary prize. The qualifying wagers are determined by game servers based on the specific games, win levels, and denominations that are linked to the controller.

6.2.9 Jackpot Level Identifier (t_description)

The `jackpotLevelId` data type is used to identify a specific jackpot level (pool) within a jackpot controller. See Jackpot Controller Identifier (t_description) for more details. The jackpot server is responsible for assigning a unique Jackpot Level Identifier to each level within the jackpot controller. To provide complete accounting for jackpot controllers, visible jackpot levels as well as overflow and hidden pools **shall** be included and reported in the Jackpot Level.

6.2.10 Jurisdiction Code (t_jurisdictionCode)

The `t_jurisdictionCode` data type is used to identify a specific gaming jurisdiction. If the gaming jurisdiction matches one of the country codes defined in ISO 3166-1 alpha-2, the country code **shall** be used for the jurisdiction. For example, MT for Malta. If the gaming jurisdiction matches one of the sub-division codes defined in ISO 3166-2, the relevant country code plus the sub-division code, separated by a hyphen, **shall** be used for the jurisdiction. For example, ES-AN for Andalusia. If the gaming jurisdiction does not match one of the country or sub-division codes defined in ISO 3166-1 alpha-2 or ISO-3166-2, a unique jurisdiction code should be assigned by the jurisdiction needing it; For example, GG-AL for Alderney.

6.2.11 Liquidity Type (t_liquidityType)

The `t_liquidityType` data type (see Table 15) identifies the type of liquidity for a game or jackpot. The following table contains the Liquidity Types supported within this document. Other Liquidity Types **may** be added in future versions of this document.

Table 15 — t_liquidityType Enumerations

Enumeration	Description
OGR_operator	The game or jackpot is played between players of the same operator
OGR_jurisdiction	The game or jackpot is played between players of operators of the same jurisdiction
OGR_multijurisdiction	The game or jackpot is played between players of operators of several jurisdictions
OGR_other	Other liquidity type.

6.2.12 Monetary Values (t_money)

The `t_money` data type is used to report monetary values. Monetary values **shall** be expressed in the major unit of the currency and the minor unit of the currency separated by a period (.). For example, EUR 1258.75 is expressed as 1258.75.

This requires that monetary values be converted to their native display format by implementations. The native display format of the currency is not communicated through this protocol, only the Currency Code is communicated. Implementations are expected to know the proper native display formats for the currencies and jurisdictions that they support. For example, in Slovenia, EUR 1258.75 should be displayed as 1.258,75.

Monetary values **may** be carried to 7 (seven) decimal places. Typically, this feature is only used with monetary values associated with jackpot contributions, but it **may** be used when reporting other monetary values as well.

Values, such as quantities and time, **may** be communicated using the `t_money` data type. In such cases, the units in which the other values are communicated is specified in the description of the element or sub-element used to communicate the values.

Implementations should support a minimum of 15 (fifteen) digits to the left of the decimal point. Support for additional digits **may** be required for some currencies.

6.2.13 Operator Identifier (t_description)

The `operatorId` data type is used to uniquely identify a specific operator within a jurisdiction. The jurisdiction is responsible for assigning Operator Identifiers. The exact usage depends on the needs of the jurisdiction. To avoid conflicts, each Operator Identifier should be unique.

When Operator Identifiers are used as part of file identifiers, then characters that would cause errors in file identifiers in target operating systems **shall** NOT be used.

Due to the regulatory framework in certain jurisdictions, there **may** be cases where an operator is not actually an operator within a jurisdiction but is still required to provide reports described within the OGR specification to the regulator. In such cases, the regulator should treat the operator as if it were an actual operator and issue the operator a unique Operator Identifier.

6.2.14 Operator Type (t_operatorType)

The `t_operatorType` data type, (see Table 16) identifies the type of operator for which a report is being created.

Table 16 — t_operatorType Enumerations

Enumeration	Description
OGR_b2B	Report is being generated for a Business to Business operator type.
OGR_b2C	Report is being generated for a Business to Customer operator type.

6.2.15 Period Type (t_periodType)

The `t_periodType` data type (see Table 17) identifies the reporting period for which a report was created. All data in the report **shall** pertain to the same period. The Period Type is used in conjunction with the `periodStart` element of reports. The Period Type identifies the duration of the reporting period – day, week, month, etc. The `periodStart` element identifies the first day of the reporting period. The following table contains the Period Types supported within this document and further describes how the `periodStart` element is used with each Period Type. Other Period Types **may** be added in future versions of this document.

Table 17 — t_periodType Enumerations

Enumeration	Period	Period Start
OGR_day	Day	The date of the day; only includes data for the specified date; for example, 20190408 indicates data for April 8, 2019. A day starts at 00:00:00 and ends at 23:59:59 of the relevant time zone.
OGR_week	Week	The date of the first day of the week; shall be a Monday; includes data for seven days starting with the specified date; for example, 20190408 indicates data for April 8, 2019 through April 14, 2019.
OGR_month	Month	The date of the first day of the month; shall be the first day of a calendar month; includes data for one month starting with the specified date; for example, 20190401 indicates data for April 1, 2019 through April 30, 2019.
OGR_quarter	Quarter	The date of the first day of the quarter; shall be the first day of a calendar quarter – that is, January, April, July, or October; includes data for three months starting with the specified date; for example, 20190401 indicates data for April 1, 2019 through June 30, 2019.
OGR_semiAnnual	Semi-Annual	The date of the first day of the semi-annual period; shall be the first day of a calendar semi-annual period – that is, January or July. includes data for six months starting with the specified date; for example, 20190101 indicates data for January 1, 2019 through June 30, 2019.
OGR_year	Year	The date of the first day of the yearly period; shall be the first day of a calendar year – that is, January; includes data for twelve months starting with the specifies date; for example, 20190101 indicates data for January 1, 2019 through December 31, 2019.

6.2.16 Transaction Class (t_transactionClass)

The `t_transactionClass` data type is used to identify the domain to which a Transaction Type belongs. It allows different sets of Transaction Types to be used within OGR without creating conflicts. The definition and usage requirements for the Transactions Types is dependent on the Transaction Class to which they belong.

The default value for the Transaction Class is `OGR_online`. This indicates that the Transaction Types belong to the Online Reporting Model. Other values are used to identify Transaction Types that belong to Traditional Reporting Models. These values are beyond the scope of the OGR specification. They **may** be defined in other specifications that are intended to be compatible with the OGR specification.

Since the default value for the Transaction Class is `OGR_online`, it can be omitted from reports when the Online Reporting Model is being used; the recipient will assume that the value for the Transaction Class is `OGR_online`.

There are limited dependencies on Transaction Class within this document. Other Transaction Classes **may** be added without affecting the integrity of this document. However, other Transaction Classes should not be substituted without carefully considering the implications of the substitutions on this document.

6.2.17 Transaction Type (`t_transactionType`)

The `t_transactionType` data type is used to identify the type of transaction that took place in a gaming system – for example, wager, base game win, jackpot win, deposit, withdrawal, etc. Transaction Types are used with games and jackpots as well as player wallets and third-party payments.

For game play, the exact usage of the Transaction Types is dependent on the Game Category. The sections of the OGR specification that address individual Game Categories describe which Transaction Types should be used with the different Game Categories.

For external movements to/from player wallets and jackpots, the sections of the OGR specification that address the external movements describe which Transaction Types should be used when reporting the different types of external movements.

7 Constructing Document Files for Reporting

7.1 Introduction

This section describes the general structure and design of the document files, using the defined XML within the specification. This version of the standard only contains specifications for the creation of document files to report periodic and near-real-time information from operators. However, the structure has been designed so that it can be extended to other types of information exchange.

The details about the contents of specific reports are contained in other sections of the specification. It is expected that each jurisdiction will determine which type of report and which frequency is best to meet its needs.

It is expected that periodic and near-real-time reports will be delivered as static electronic files. The standardized file identifier **shall** reflect the type of the report it contains along with other meta information as described in this section. If the static file is compressed or otherwise encrypted during the delivery process, the name of the file **shall** be amended to identify the compression or encryption methodology. It is expected that common data processing methodologies, such as FTP, SFTP, or HTTP REST, will be used to deliver the files; other methodologies **may** be used.

7.2 File Naming Convention

This section describes the file naming convention.

When storing gaming activity reports as static files in file systems, the following naming convention should be used.

Filename.xml

File identifiers **shall** use the [a-zA-Z0-9_\-+] encoding standard and **shall** observe letter case preservation. File identifiers **shall** have a maximum length of 255 characters and **shall** be constructed using several data elements separated by a hashtag (#), as follows:

jurisdictionCode#operatorId#submitterId#partitionId#periodType#periodStart#
reportId#sequenceNumber#specialId

Section 7.3 defines the data elements.

The seqNumber **shall** be zero-padded left to six digits and **shall** always start from 000001.

The report name segments **shall** be separated by a hashtag character. Except for the specialId segment, the hashtag character shall be included even if the value of a segment is <empty>.

If the report is not the result of a Special Request – that is the specialId value is equal to <empty>, then the specialId and the preceding hashtag **shall** be omitted from the file identifier. The periodStart **shall** be represented in the YYYYMMDD format. The filename's extension **shall** be xml.

7.3 XML Document Structure

The Standard allows for different types of data structures to be created and sent to regulatory authorities, however, in the current version of the Standard only one type of data structure which is reporting, has been defined.

This section describes the structure of the reports defined within this standard. The document structure has been designed to be simple and uniform while simultaneously providing flexibility for adaption by users.

Data types and data elements defined by the standard are arranged into data domains (see Figure 3). The domains are defined as Level 0 through 5, where Level 0 is the base level and Level 5 is the top level. Starting from the top level, each domain includes the preceding one. Level 0 has no inclusions.

Level	Description
5	Defines the Standard CEN TC/456 Document
4	Defines the report structure and type of reports
3	Defines the Simple and Complex Types used in the reports defined in level 4
2	Fundamental Simple and Complex Types
1	Fundamental List Types
0	Fundamental General data Types

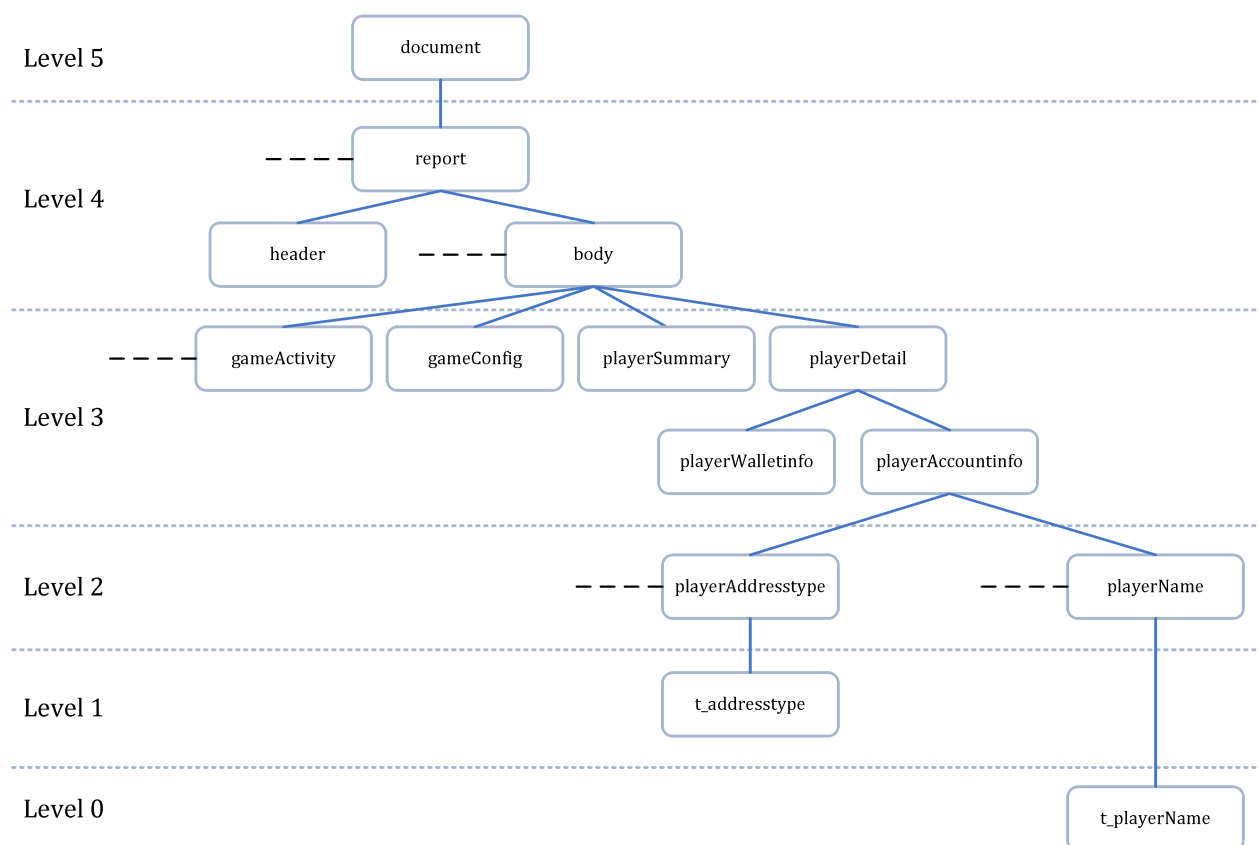


Figure 3

The data domains defined in the standard are part of the “cen 456” namespace (prefix “cen”). Users of the standard **may** create their own data domains. This can be done in Levels 3 and 4 only and **shall** use a separate namespace. The standard itself **shall** not be altered and any changes **shall** be additions.

It is recommended that the namespace reflect the jurisdiction code such as “ES”, “DE”, etc. Any new data elements **shall** only use the data types in Levels 0-2.

At the highest level, the standard schema for the XML file document is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
  xmlns:cen="urn:CEN:OGR:1.0:CEN:ogr" targetNamespace="urn:CEN:OGR:1.0:CEN:ogr"
  elementFormDefault="qualified">
  <xs:include schemaLocation="tc456_level14.xsd"/>
  <xs:element name="document" type="cen:c_document"/>
  <xs:complexType name="c_document">
    <xs:choice>
      <xs:element ref="cen:report"/>
    </xs:choice>
  </xs:complexType>
</xs:schema>
```

7.4 Document Types

7.4.1 Introduction

While several document types (see Table 18) have been identified as possible document types for use within this standard, only the `report` document type has been fully developed and deemed within this standard’s current scope. Thus, only the `report` document type, as well as its subsidiary set of regulatory reports, are defined in this document and are included within the XML schemas that accompany the specification.

Users of this standard should not create their own document types as these are fundamental to the standard’s file structure and defined in the highest level (Level 4) of the XML schema.

Figure 4 shows the document types have been created within this standard.

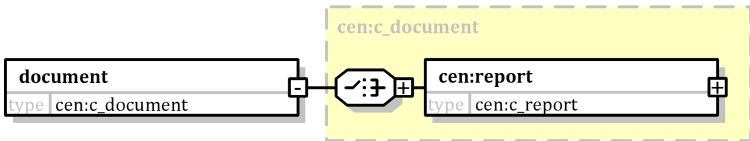


Figure 4

Table 18 — Document Types

c_report	Document (file) contains a report.
----------	------------------------------------

7.4.2 Report Data Element (Report)

All periodic and near-real-time reports share a common root element (`report`). The contents of this element describe the report’s structure, which is as follows:

The report element is comprised of a header (`header`) and one of the report types (see Figure 5).

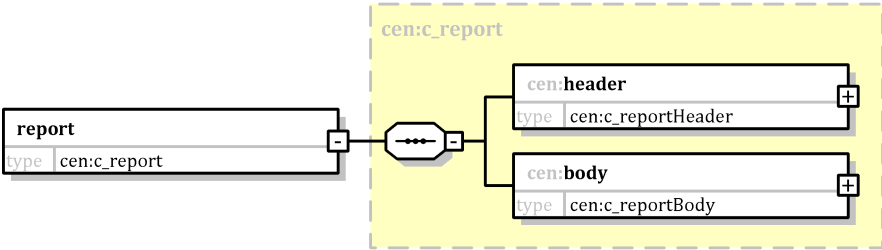


Figure 5

7.4.2.1 Header Element (header)

7.4.2.1.1 General

The header is a mandatory standard data element and of type c_reportHeader (see Table 19). Its structure is shown in Figure 6.

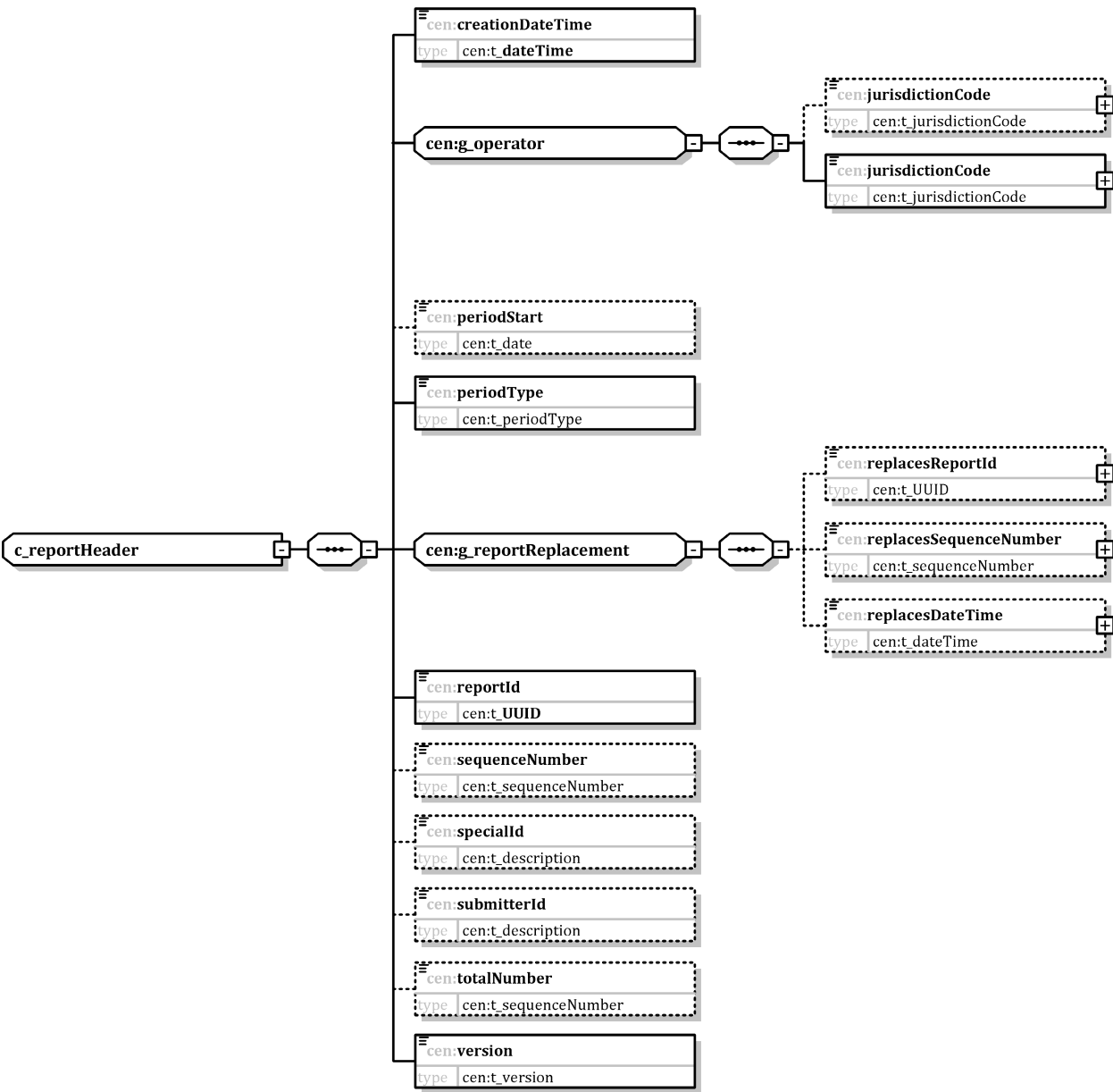


Figure 6

Table 19 — reportHeader Elements

Elements	Restrictions	Description
creationdateTime	type: t_dateTime minOcc: 1 maxOcc: 1	Date/time that the report was created.
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Jurisdiction Code; the jurisdiction to which the report is being submitted.
operatorId	type: t_description minOcc: 1 maxOcc: 1	Operator Identifier; the operator submitting the report.
partitionId	type: t_description minOcc: 0 maxOcc: 1	Partition identifier. See Partition Identifier (partitionId) for more details.
periodStart	type: t_date minOcc: 0 maxOcc: 1	Period Start Date; identifies the date on which the period started; when required, more specific date/time ranges may be contained within individual reports. See Period Type (t_periodType) for more details.
periodType	type: t_periodType minOcc: 1 maxOcc: 1	Period Type; identifies the accounting period for which the report was created. See Period Type (t_periodType) for more details.
replacesReportId	type: t_UUID minOcc: 0 maxOcc: 1	Replaces Report Identifier; uniquely identifies the original set of data being replaced. See Replaces Report Identifier (replacesReportId) for more details.
replacesSequenceNumber	type: t_sequenceNumber minOcc: 0 maxOcc: 1	Replaces Sequence Number; uniquely identifies the subset of the original data being replaced. See Replaces Report Identifier (replacesReportId) for more details.
replacesDateTime	type: t_dateTime minOcc: 0 maxOcc: 1	Replaces Date/Time; the date/time the original report was created – that is, the <code>dateTime</code> of the original report. See Replaces Report Identifier (replacesReportId) for more details.
reportId	type: t_UUID minOcc: 1 maxOcc: 1	Report Identifier; uniquely identifies the set of data included in the report. See Report Identifier (reportId) for more details.
sequenceNumber	type: t_sequenceNumber minOcc: 0 maxOcc: 1	Sequence Number. See Sequence Number (sequenceNumber) for more details.
specialId	type: t_description minOcc: 0 maxOcc: 1	Special Request Identifier; shall be set to <empty> or omitted for standard reports; shall be included and set to a value other than <empty> for Special Requests. See Special Request Identifier (specialId) for more details.
submitterId	type: t_description minOcc: 0 maxOcc: 1	Submitter Identifier. See Submitter Identifier (submitterId) for more details.
totalNumber	type: t_sequenceNumber minOcc: 0 maxOcc: 1	Total Number of reports in the sequence. See Sequence Number (sequenceNumber) for more details.
version	type: t_version minOcc: 1 maxOcc: 1	Identifies the version of the OGR specification used to create the report.

7.4.2.1.2 Version Identifier (version)

The `version` element is used to identify the version of the specification used to create a report. A dot-delimited format **shall** be used when constructing the version number. For example, "version 1.0" **shall** be represented as the character string "1.0". Superfluous leading and trailing characters **shall** be removed. For example, "v1.0" **shall** be represented as "1.0".

7.4.2.1.3 Submitter Identifier (submitterId)

The `submitterId` element is used to uniquely identify a specific submitter of data for an operator. The exact usage depends on the needs of the operator and the jurisdiction. The Submitter Identifier could be used to identify different systems operated by the operator or different suppliers working for the operator. The operator is responsible for assigning Submitter Identifiers. To avoid conflicts, each Submitter Identifier should be unique.

When Submitter Identifiers are used as part of file identifiers then, characters that would cause errors in file identifiers in target operating systems shall NOT be used.

7.4.2.1.4 Partition Identifier (partitionId)

The `partitionId` element is used to uniquely identify a specific partition or subset of data from a submitter. The exact usage depends on the needs of the operator and the jurisdiction. The Partition Identifier could represent the intended destination for a report, such as a data vault. Or, it could be used to identify different series of reports generated by different systems operated by the same submitter. The operator is responsible for assigning a Partition Identifiers. To avoid conflicts, each Partition Identifier should be unique.

When Partition Identifiers are used as part of file identifiers then, characters that would cause errors in file identifiers in target operating systems **shall** NOT be used.

7.4.2.1.5 Report Identifier (reportId)

The `reportId` element uniquely identifies the set of data being reported to a jurisdiction. The Report Identifier **shall** be constructed as a Universally Unique Identifier as specified in ISO/IEC 9834-8:2014.

The set of data associated with a specific Report Identifier **may** be communicated to a jurisdiction in more than one report. For example, a large set of data, such as all the player registration information for an operator, **may** be communicated to a jurisdiction in multiple reports. In such cases, the same Report Identifier **shall** be used in all reports associated with the set of data. The Sequence Number element is used to uniquely identify a specific report associated with a Report Identifier. See Sequence Number Element (`sequenceNumber`) for more details.

7.4.2.1.6 Sequence Number (sequenceNumber)

The Sequence Number element is used when a large set of data needs to be broken into multiple subsets for reporting purposes. It identifies a specific subset of the large set data. The Report Identifier identifies the whole set of data; the Sequence Number identifies a subset within that set of data. When sending the data, the source system **shall** send the number of subsets that comprise the entire data set with each subset. This would equate to the last `sequenceNumber` that will be used. For example, if the data set is going to be broken down into five (5) subsets, then the last `sequenceNumber` value **shall** equal five (5).

The Sequence Number **shall** be constructed from a strictly increasing series that starts at 1 and increments by 1. That is, the first report associated with a Report Identifier **shall** be assigned sequence number 1, the second report **shall** be assigned sequence number 2, etc. A new series **shall** be used for each new Report Identifier. If the whole set of data is reported in a single report, the Sequence Number **shall** simply be set to 1.

The `totalNumber` is the value of the last Sequence Number used with a specific Report Identifier. If the whole set of data is reported in a single report, the Total Number **shall** simply be set to 1 like the Sequence Number.

7.4.2.1.7 Special Request Identifier (`specialId`)

The `specialId` element is used to indicate that a report was the result of a Special Request from a jurisdiction. For example, a special request from a jurisdiction for all deposits and withdrawals for a player for a specific time period. The method for making Special Requests is not described within the specification; only the mechanism for reporting the results of the Special Requests is described.

Like standard reports, Special Requests are organized by accounting period. If the Special Request spans multiple accounting periods, multiple reports will be generated. Similarly, Special Requests use the format of one of the Report Types described within the specification. If the Special Request requires multiple Report Types, multiple reports will be generated.

To differentiate Special Requests from standard reports, the Special Request Identifier is included in the `gamingReport` element of the resulting reports. It should also be appended to the standardized name of those reports. See File Naming Conventions for more details.

In most other aspects, Special Requests are the same as standard reports. The primary difference is that Special Requests only contain the data specifically requested by the jurisdiction; standard requests contain the full set of data specified within this document. The data requested by the jurisdiction **may** be a subset of the full set of data normally reported.

The jurisdiction and/or operator is responsible for assigning a Special Request Identifiers. To avoid conflicts, each Special Request Identifier should be unique.

When Special Request Identifiers are included within the file identifiers of the resulting reports then, characters that would cause errors in file identifiers in target operating systems **shall** NOT be used.

7.4.2.1.8 Replaces Report Identifier (`replacesReportId`)

If a report is intended to replace a previously submitted report, the `replacesReportId` element **shall** contain the `reportId` of the previously submitted report; otherwise, the `replacesReportId` element **shall** be omitted.

If a report is intended to replace a single report in a series, the `replacesSequenceNumber` element **shall** be set to the `sequenceNumber` of the original report; otherwise, the `replacesSequenceNumber` element **shall** be omitted, implying that all reports in the series are being replaced.

If the `replacesReportId` element is included, the `replacesDateTime` element **shall** also be included; otherwise, the `replacesDateTime` element **shall** be omitted.

If the `replacesDateTime` element is included, the `replacesDateTime` element **shall** be set to the `dateTime` used when the original report was submitted. When the `replacesSeqNumber` is included, the `replacesDateTime` **shall** be the `dateTime` associated with the original `sequenceNumber`; otherwise, the `replacesDateTime` **shall** be the `dateTime` of the first `sequenceNumber` associated with the original series of reports.

A previously submitted report can be cancelled by replacing it with a new report containing no sub-elements.

7.4.2.2 Report Data Element (body)

The `header` element provides a description and context for the report being generated. This data includes the period covered by the report, the operator generating it, and the report's name, in addition to other elements. The `body` element contains the report data itself as sub-elements. The names of the sub-elements identify the type of report data being submitted. All sub-elements within a `body` element should be of the same type – for example, all `gameSummary` sub-elements or all `playerRegistration` sub-elements. However, sub-elements of a `body` element **may** be of different types.

The standard reports(see Table 21), which are detailed in the individual chapters within this document, are as follows:

Table 20 — Reports List

Report Name
Player Registration (playerRegistration)
Player Activity (playerActivity)
Player Balance (playerBalance)
Jackpot Activity (jackpotActivity)
Jackpot Balance (jackpotBalance)
Jackpot Interval (jackpotInterval)
Jackpot Configuration (jackpotConfig)
Game Results Report (gameResults)
Game Activity Report (gameActivity)
Game Summary Report (gameSummary)
Player Summary Report (playerSummary)
Jackpot Summary Report (jackpotSummary)
Payee Summary Report (payeeSummary)
Game Funds-in-Play Report (gameFundsInPlay)
Player Funds-in-Play Report (playerFundsInPlay)
Jackpot Funds-in-Play Report (jackpotFundsInPlay)
Payee Funds-in-Play Report (payeeFundsInPlay)
Game Configuration Report (gameConfig)
Metrics Report (metricsReport)
Exchange Rate Report (exchangeRate)
Split Report (_splitsReport)
Game Funds-in-play Recap (gameInPlayRecap)
Business to Business Report (b2b)

7.5 Simple Data Types

7.5.1 Introduction

This section of the specification **may** reference simple data types that are also referenced in other sections of the specification. Those simple data types are described in detail in the section of the specification that addresses common data types. Only simple data types that are not described in that section are described in this section (see Table 21).

7.5.2 Simple Data Type Table

Table 21 — Simple Data Types

Data Type	Restrictions	Description
t_sequenceNumber	type: xs:int minIncl: 1	Sequence Number.
t_version	type: xs:string maxLen: 8	Version; identifies the version of the OGR specification being used.

8 Reporting on Player Registration

8.1 Introduction

This section describes the reports that can be used to monitor player registration information gathered by operators through the specification. The information reported about a player includes the player's name, address, phone number, identification, nicknames, status, etc. These reports can be used to meet various regulatory reporting needs. For example,

- monthly, player registration information can be reported for all players registered with the operator except players with closed accounts. Players with closed accounts can be reported in the month that the account was closed and then not reported again while the account remains closed;
- daily, player registration information can be reported for players who registered during the day or had changes to their registration information during the day;
- in near-real-time, new player registrations as well as updates to existing registration information can be reported.

8.2 Periodic & Near-Real-Time Reports

8.2.1 Player Registration Report (playerRegistration)

8.2.1.1 Introduction

The `playerRegistration` report is used to report player registration information. When using this report, the sub-elements of the `body` element **shall** be `playerRegistration` sub-elements. The `playerRegistration` report **may** be sent as a near-real-time report or as a periodic report.

To the extent that the player registration information is required by the regulator, player registration information **shall** always be reported in its entirety. Player registration information not required by the regulator should be omitted by the operator. The most recent information reported by the operator supersedes any information previously reported for the player.

8.2.1.2 Player Registration Element (playerRegistration)

The following Table 22 describes the contents of the `playerRegistration` element.

Table 22 — playerRegistration Elements

Element	Restrictions	Description
<code>lastUpdate</code>	type: <code>t_dateTime</code> minOcc: 1 maxOcc: 1	Date/time that the player registration information was last updated.
<code>player</code>	type: <code>c_player</code> minOcc: 1 maxOcc: 1	Identifies the player. See Player (<code>c_player</code>) for more details.
<code>playerInfo</code> ¹	type: <code>c_playerInfo</code> minOcc: 0 maxOcc: 1	Contains the name and other identifying information for a player who is a natural person. See Player Information (<code>c_playerInfo</code>) for more details.
<code>entityInfo</code> ¹	type: <code>c_entityInfo</code> minOcc: 0 maxOcc: 1	Contains the name and other identifying information for a player who is not a natural person – that is, another type of legal entity. See Entity Information (<code>c_entityInfo</code>) for more details.
<code>playerStatus</code>	type: <code>c_playerStatus</code> minOcc: 1 maxOcc: 1	Contains the status of the player. See Player Status (<code>c_playerStatus</code>) for more details.
<code>playerAddresses</code>	type: <code>c_addressList</code> minOcc: 0 maxOcc: 1	Contains a list of addresses for the player. See Player Addresses (<code>c_playerAddresses</code>) for more details.
<code>playerPhones</code>	type: <code>c_phoneList</code> minOcc: 0 maxOcc: 1	Contains a list of phone numbers for the player. See Player Phones (<code>c_playerPhones</code>) for more details.
<code>playerIdentifications</code>	type: <code>c_playerIdentificationList</code> minOcc: 0 maxOcc: 1	Contains a list of government-issued IDs for the player. See Player Identifications (<code>c_playerIdentificationList</code>) for more details.
<code>playerNicknames</code>	type: <code>c_playerNicknameList</code> minOcc: 0 maxOcc: 1	Contains a list of nicknames for the player. See Player Nicknames (<code>c_playerNicknameList</code>) for more details.
<code>playerLimits</code>	type: <code>c_playerLimitList</code> minOcc: 0 maxOcc: 1	Contains a list of limits for the player. See Player Limits (<code>c_playerLimits</code>) for more details.
<code>playerExclusions</code>	type: <code>c_playerExclusionList</code> minOcc: 0 maxOcc: 1	Contains a list of exclusions for the player. See Player Exclusions (<code>c_playerExclusions</code>) for more details.
¹ Choice: one element may be included – <code>playerInfo</code> or <code>entityInfo</code> .		

8.3 Complex Data Types

8.3.1 Introduction

This section of the OGR specification **may** reference complex data types that are also referenced in other sections of the OGR specification. Those complex data types are described in detail in the section of the OGR specification that addresses common data types. Only complex data types that are not described in that section are described in this section.

8.3.2 Player Information (c_playerInfo)

The `c_playerInfo` data type (see Table 23) includes general descriptive information about a player who is a natural person including the player's name, date of birth, and gender.

Table 23 — c_playerInfo Elements

Element	Restrictions	Description
givenName	type: t_description minOcc: 0 maxOcc: ∞	Player's given name(s); recursive, meaning as many as required. According to ISO 20022 (Series).
middleName	type: t_description minOcc: 0 maxOcc: ∞	Player's middle name(s); recursive, meaning as many as required. According to ISO 20022 (Series). Shall be entered in the order that they appear on legal documents.
familyName	type: t_description minOcc: 0 maxOcc: ∞	Player's family (last) name(s); recursive, meaning as many as required. According to ISO 20022 (Series).
nationality	type: t_countryCode minOcc: 0 maxOcc: 1	Player's nationality. According to ISO 3166-1 alpha-2.
residence	type: t_countryCode minOcc: 0 maxOcc: 1	Player's country of residence. According to ISO 3166-1 alpha-2.
taxSubJurisdiction	type: t_countrySubDivision minOcc: 0 maxOcc: 1	Player's tax jurisdiction; jurisdiction to which gaming taxes should be allocated. According to ISO 3166-1 alpha-2.
personalIdCode	type: t_description minOcc: 0 maxOcc: 1	Government issued Identification Code for a person.
birthDate	type: t_date minOcc: 0 maxOcc: 1	Player's date of birth. According to RFC 3339.
birthCountry	type: t_countryCode minOcc: 0 maxOcc: 1	Player's country of birth. According to ISO 3166-1 alpha-2.

Element	Restrictions	Description
birthSubDivision	type: t_countrySubDivision minOcc: 0 maxOcc: 1	Player's country sub-division (state/province) of birth. According to ISO 3166-1 alpha-2.
birthCity	type: t_description minOcc: 0 maxOcc: 1	Player's city of birth. According to ISO 20022 (Series).
deathDate	type: t_date minOcc: 0 maxOcc: 1	Player's date of death. According to RFC 3339.
gender	type: t_genderType minOcc: 0 maxOcc: 1	Player's gender. According to modified ISO 5218 with an 'OGR_' prefix; the values are (unkown - 0, male - 1, female - 2, other - 3, nonperson - 9).
emailAddress	type: t_emailAddress minOcc: 0 maxOcc: 1	Player's email address. According to RFC 2822: Internet Message Format
userLogin	type: t_description minOcc: 1 maxOcc: 1	The login or profile name used by a Player to log on to this Operator's site.
testAccount	type: xs:boolean minOcc: 0 maxOcc: 1	Identifies whether this is a player test account
adminAccount	type: xs:boolean minOcc: 0 maxOcc: 1	Identifies whether this is a player admin account
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.

8.3.3 Entity Information (c_entityInfo)

The `c_entityInfo` data type (see Table 24) includes general descriptive information about a player who is not a natural person – that is, another type of legal entity such as a corporation. The information includes the entities's name, date of formation, etc.

Table 24 — c_entityInfo Elements

Elements	Restrictions	Description
entityName	type: t_description minOcc: 0 maxOcc: 1	Entity's name.
nationality	type: t_countryCode minOcc: 0 maxOcc: 1	Entity's nationality. According to ISO 3166-1 alpha-2.

Elements	Restrictions	Description
residence	type: t_countryCode minOcc: 0 maxOcc: 1	Entity's country of residence – that is, the country from which the entity is operating. According to ISO 3166-1 alpha-2.
taxSubJurisdiction	type: t_countrySubDivision minOcc: 0 maxOcc: 1	Entity's tax jurisdiction state/province code. According to ISO 3166-1 alpha-2.
formationDate	type: t_date minOcc: 0 maxOcc: 1	Entity's date of formation.
formationCountry	type: t_countryCode minOcc: 0 maxOcc: 1	Entity's country of formation. According to ISO 3166-1 alpha-2.
formationSubDivision	type: t_countrySubDivision minOcc: 0 maxOcc: 1	Entity's country sub-division (state/province) of formation. According to ISO 3166-1 alpha-2.
dissolutionDate	type: t_date minOcc: 0 maxOcc: 1	Entity's date of dissolution.
entityType	type: t_entityType minOcc: 0 maxOcc: 1	Entity Type; for example, corporation or partnership.
emailAddress	type: t_emailAddress minOcc: 0 maxOcc: 1	Entity's email address.
testAccount	type: xs:boolean minOcc: 0 maxOcc: 1	Identifies whether this is a entity test account
adminAccount	type: xs:boolean minOcc: 0 maxOcc: 1	Identifies whether this is a entity admin account
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.

8.3.4 Player Status (c_playerStatus)

The `c_playerStatus` data type (see Tables 25, 26 and 27) contains information about the player's status with the operator including the date/time that the status was changed.

Table 25 — c_playerStatus Elements

Elements	Restrictions	Description
playerCurrentStatus	type: c_playerStatusChange minOcc: 1 maxOcc: 1	
playerStatusChangeHistory	type: c_playerStatusChangeList minOcc: 0 maxOcc: 1	
verificationDateTime	type: t_dateTime minOcc: 0 maxOcc: 1	Verification Date/Time; date/time that the entity's registration information was verified.

Table 26 — playerStatusChangeList Elements

Elements	Restrictions	Description
playerStatusChange	type: c_playerStatusChange minOcc: 0 maxOcc: ∞	

Table 27 — playerStatusChange Sub-Elements

Sub-Elements	Restrictions	Description
playerOperatorStatus	type: t_playerStatus minOcc: 0 maxOcc: 1	Player's status with the operator.
playerJurisdictionStatus	type: t_playerStatus minOcc: 0 maxOcc: 1	Player's status with the regulator.
playerStatusChangedOn	type: t_dateTime minOcc: 0 maxOcc: 1	Date/time that the player's status was last changed.

8.3.5 Player Addresses (c_addressList)

The `c_addressList` complex data type (see Tables 28 and 29) contains a list of physical addresses for a player. Each `address` sub-element contains information about one physical address including street, city, country, sub-divison (state/province), and postal code.

Table 28 — c_addressList Element

Element	Restrictions	Description
address	type: c_address minOcc: 0 maxOcc: ∞	Contains an address for the player.

Table 29 — c_address Sub-Element

Sub-Element	Restrictions	Description
addressType	type: t_addressType minOcc: 0 maxOcc: 1	Address Type; for example, home or work.
addressLine	type: t_description minOcc: 0 maxOcc: ∞	The address field is recursive. According to ISO 20022 (Series). This field is recursive. Meaning that additional fields will be created to capture all of the information necessary for a complete address, excepting that information for which specific fields (such as addressCity, postalCode, etc.) exist.
addressCity	type: t_description minOcc: 0 maxOcc: 1	City name of the address. According to ISO 20022 (Series).
postCode	type: t_postCode minOcc: 0 maxOcc: 1	Postal code of the address. According to ISO 20022 (Series).
subDivision	type: t_countrySubDivision minOcc: 0 maxOcc: 1	Country Sub-Division (State/Province) of the address. According to ISO 3166-1 alpha-2.
countryCode	type: t_countryCode minOcc: 0 maxOcc: 1	Country code of the address. According to ISO 3166-1 alpha-2.
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.
isDeleted	type: xs:boolean minOcc: 0 maxOcc: 1	Is Deleted Indicator; indicates whether the record has been logically (hidden) or permanently (erased) deleted.

8.3.6 Player Phones (c_phoneList)

The `c_phoneList` complex data type (see Tables 30 and 31) contains a list of phone numbers for a player. Each `phone` sub-element contains information about one phone number including country code and local number.

Table 30 — c_phoneList Element

Element	Restrictions	Description
phone	type: c_phone minOcc: 0 maxOcc: ∞	Contains a phone number for the player.

Table 31 — c_phone Sub-Elements

Sub-Elements	Restrictions	Description
phoneType	type: t_phoneType minOcc: 0 maxOcc: 1	Phone Type; for example, home or mobile. According to ISO 20022 (Series).
phoneCountry	type: t_ituCountry minOcc: 0 maxOcc: 1	Phone country code. According to ISO 20022 (Series).
phoneNumber	type: t_ituNumber minOcc: 1 maxOcc: 1	Phone number. According to ISO 20022 (Series).
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.
isDeleted	type: xs:boolean minOcc: 0 maxOcc: 1	Is Deleted Indicator; indicates whether the record has been logically (hidden) or permanently (erased) deleted.

8.3.7 Player Identifications (c_playerIdentificationList)

The `c_playerIdentificationList` complex data type (see Tables 32 and 33) contains a list of government-issued IDs for a player. Each `playerIdentification` sub-element contains information about one government-issued ID including the type of ID, country code, sub-division (state/province code), and ID number.

Table 32 — c_playerIdentificationList Element

Element	Restrictions	Description
playerIdentification	type: c_playerIdentification minOcc: 0 maxOcc: ∞	Contains a government-issued ID for the player.

Table 33 — c_playerIdentification Sub-Elements

Sub-Elements	Restrictions	Description
idType	type: t_idType minOcc: 0 maxOcc: 1	Identification type; for example, driver's license or passport.
country	type: t_countryCode minOcc: 0 maxOcc: 1	Country code. According to ISO 3166-1 alpha-2.
countrySubDivision	type: t_countrySubDivision minOcc: 0 maxOcc: 1	State/Province code. According to ISO 3166-1 alpha-2.

Sub-Elements	Restrictions	Description
idDocumentNumber	type: t_description minOcc: 0 maxOcc: 1	Some ID documents have a number that is different from the official personal ID number. This element captures that number assigned by the issuer of the document being used for ID.
personalIdCode	type: t_description minOcc: 0 maxOcc: 1	Government issued Identification Code for a person.
issueDate	type: t_date minOcc: 0 maxOcc: 1	Date of issuance.
expirationDate	type: t_date minOcc: 0 maxOcc: 1	Date of expiration. According to RFC 3339 format.
issueAuthority	type: t_description minOcc: 0 maxOcc: 1	Issuing authority. According to ISO 20022 (Series) format.
issuePlace	type: t_description minOcc: 0 maxOcc: 1	Place of issuance. According to ISO 20022 (Series) format.
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.
isDeleted	type: xs:boolean minOcc: 0 maxOcc: 1	Is Deleted Indicator; indicates whether the record has been logically (hidden) or permanently (erased) deleted.
verificationDateTime	type: t_dateTime minOcc: 0 maxOcc: 1	Verification Date/Time; date/time that the player's registration information was verified.

8.3.8 Player Nicknames (c_playerNicknameList)

The `c_playerNicknameList` complex data type (see Tables 34 and 35) contains a list of nicknames for a player – for example, the login used by the player or some other unique identifier used by an operator to identify the player. Each `playerNickname` sub-element contains information about one nickname used to identify a player.

Table 34 — c_playerNicknameList Element

Element	Restrictions	Description
playerNickname	type: c_playerNickname minOcc: 0 maxOcc: ∞	Contains a nickname for the player.

Table 35 — c_playerNickname Sub-Elements

Sub-Elements	Restrictions	Description
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Jurisdiction Code; jurisdiction that issued the Operator Identifier.
operatorId	type: t_description minOcc: 1 maxOcc: 1	Operator Identifier; identifier of the operator associated with the nickname.
playerNickname	type: t_description minOcc: 1 maxOcc: 1	Nickname (login or other identifier) used by the player with the operator.
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.
isDeleted	type: xs:boolean minOcc: 0 maxOcc: 1	Is Deleted Indicator; indicates whether the record has been logically or permanently deleted.

8.3.9 Player Limits (c_playerLimitList)

The `c_playerLimitList` complex data type (see Tables 36 and 37) contains a list of limits for a player. Limits **may** be self-imposed, regulator-imposed, or operator-imposed (`source`). They may apply to a day, week, month, quarter, or year (`limitPeriod`). And, they may apply to various types of transactions (`limitType`) – for example, deposits, game participation, etc.

Table 36 — c_playerLimits Element

Element	Restrictions	Description
playerLimit	type: c_playerLimit minOcc: 0 maxOcc: ∞	Contains a limit for the player.

Table 37 — c_playerLimit Elements

Sub-Element	Restrictions	Description
source	type: t_limitSource minOcc: 1 maxOcc: 1	Source; for example, self-imposed or regulator-imposed.
period	type: t_limitPeriod minOcc: 1 maxOcc: 1	Period; for example, day, week, month, or year.
type	type: t_limitType minOcc: 1 maxOcc: 1	Type; for example, deposits or stakes.

Sub-Element	Restrictions	Description
amount	type: t_money minOcc: 0 maxOcc: 1	Limit Amount; when the limitType is OGR_time, the value shall be communicated in minutes.
currency	type: t_currencyCode minOcc: 0 maxOcc: 1	Currency Code; XXX indicates that the limit is not a monetary value.
gameCategory	type: t_gameCategory minOcc: 0 maxOcc: 1	The game category type for which this limit was created. <empty> indicates all game categories.
startDateTime	type: t_dateTime minOcc: 0 maxOcc: 1	The date and time the player's limit goes into effect.
limitAction	type: t_limitAction minOcc: 0 maxOcc: 1	Limit Action; for example, warning or exclude from play for the remainder of the period.
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.
isDeleted	type: xs:boolean minOcc: 0 maxOcc: 1	Is Deleted Indicator; indicates whether the record has been logically (hidden) or permanently (erased) deleted.

8.3.10 Player Exclusions (c_playerExclusionList)

The `c_playerExclusionList` complex data type (see Tables 38 and 39) contains a list of exclusions for a player that have been registered with the operator. Exclusions may be self-imposed, regulator-imposed, or operator-imposed (`Source`). They apply to a specific period of time (`startDate` and `endDate`) and to specific hours within that period (`dailyFromTime` and `dailyToTime`). They may apply to a single game category – for example, cash poker – or all game categories (`gameCategory`). And, they may expire automatically at the end of the period or may remain in place until removed by the player (`autoContinuation`).

Table 38 — c_playerExclusionList Element

Element	Restrictions	Description
playerExclusion	type: c_playerExclusion minOcc: 0 maxOcc: ∞	Contains an exclusion for the player.

Table 39 — c_playerExclusion Sub-Elements

Sub-Element	Restrictions	Description
jurisdictionCode	type: t_jurisdictionCode minOcc: 0 maxOcc: 1	Jurisdiction Code; jurisdiction that issued the Operator Identifier.
source	type: t_exclusionSource minOcc: 1 maxOcc: 1	Exclusion Source; for example, self-imposed or regulator imposed.
startDate	type: t_date minOcc: 0 maxOcc: 1	Date that the exclusion starts; first day of the exclusion; local time of the player.
endDate	type: t_date minOcc: 0 maxOcc: 1	Date that the exclusion ends; last day of the exclusion; local time of the player.
dailyFromTime	type: t_time minOcc: 0 maxOcc: 1	Time each day at which the exclusion starts; local time of the player.
dailyToTime	type: t_time minOcc: 0 maxOcc: 1	Time each day at which the exclusion ends; local time of the player.
gameCategory	type: t_gameCategory minOcc: 0 maxOcc: 1	Game Category; <empty> indicates all game categories.
gameType	type: t_gameType minOcc: 0 maxOcc: 1	Game Type; type of game within the Game Category.
gameId	type: t_description minOcc: 0 maxOcc: 1	Game Identifier. Typically required unless it is being used for summarization by game category in specific reports required by a regulator.
autoContinuation	type: xs:boolean minOcc: 0 maxOcc: 1	Auto-Continuation; indicates whether the exclusion continues until removed by the player (true) or is automatically removed at the end of the period (false)

Sub-Element	Restrictions	Description
removalDateTime	type: t_dateTime minOcc: 0 maxOcc: 1	Removal Date/Time; date/time that the player removed the exclusion.
lastChange	type: t_dateTime minOcc: 0 maxOcc: 1	Last Change Date/Time; data time that the record was last changed.
isDeleted	type: xs:boolean minOcc: 0 maxOcc: 1	Is Deleted Indicator; indicates whether the record has been logically (hidden) or permanently (erased) deleted.

8.4 Simple Data Types

8.4.1 Introduction

This section of the specification may reference simple data types that are also referenced in other sections of the specification. Those simple data types (see Table 40) are described in detail in the section of the specification that addresses common data types. Only simple data types that are not described in that section are described in this section.

8.4.2 Simple Data Type Table

Table 40 — Simple Data Types

Data Type	Restrictions	Description
t_addressType	type: t_uniqueId	Address Type; see Address Type (t_addressType) for more details.
t_countrySubDivision	type: xs:string maxLen: 3	Country Sub-Division; for example, state, province, etc.; shall conform to ISO 3166-2 standard; for example, CT.
t_emailAddress	type: xs:string maxLen: 320	Email Address; shall conform to RFC 2822.
t_entityType	type: t_uniqueId	Entity Type; see Entity Type (t_entityType) for more details.
t_exclusionSource	type: t_uniqueId	Exclusion Source; see Exclusion Source (t_exclusionSource) for more details.
t_genderType	type: t_uniqueId	Gender Type; see Gender Type (t_genderType) for more details.
t_idType	type: t_uniqueId	Identification Type; see Identification Type (t_idType) for more details.
t_ituCountry	type: xs:string pattern: [0-9]{0,3}	Phone country code; shall conform to ITU/WTNG conventions.
t_ituNumber	type: xs:string pattern: [0-9]{0,6}[0-9]{0,8}	Phone number; shall conform to ITU/WTNG conventions.
t_limitAction	type: t_uniqueId	Limit Action; see Limit Action (t_limitAction) for more details.
t_limitPeriod	type: t_uniqueId	Limit Period; see Limit Period (t_limitPeriod) for more details.

Data Type	Restrictions	Description
t_limitSource	type: t_uniqueId	Limit Source; see Limit Source (t_limitSource) for more details.
t_limitType	type: t_uniqueId	Limit Type; see Limit Type (t_limitType) for more details.
t_phoneType	type: t_uniqueId	Phone Type; see Phone Type (t_phoneType) for more details.
t_playerStatus	type: t_uniqueId	Player Status; see Player Status (t_playerStatus) for more details.
t_postCode	type: xs:string maxLen: 10	Postal code.

8.4.3 Address Type (t_addressType)

The t_addressType data type identifies the type (see Table 41) of address provided by the player – for example, home, business, etc. The following table contains the Address Types supported within this document based on ISO20022 (Series). Other Address Types may be added in future versions of this document.

There are no dependencies on Address Type within this document. Other Address Types may be added or substituted without affecting the integrity of this document.

Table 41 — t_addressType Enumerations

Enumeration	Description
OGR_home	Home address per ISO 20022 (Series).
OGR_business	Business address per ISO 20022 (Series).
OGR_postOfficeBox	Post office box address per ISO 20022 (Series).
OGR_delivery	Delivery address. per ISO 20022 (Series)
OGR_mailTo	Mail-to address per ISO 20022 (Series).
OGR_other	Other address per ISO 20022 (Series).

8.4.4 Entity Type (t_entityType)

The t_entityType data type (see Table 42) identifies the legal structure of an entity – for example, corporation, partnership, etc. The following table contains the Entity Types supported within this document. Other Entity Types may be added in future versions of this document.

There are no dependencies on Entity Type within this document. Other Entity Types may be added or substituted without affecting the integrity of this document.

Table 42 — t_entityType Enumerations

Enumeration	Description
OGR_corporation	Corporation.
OGR_partnership	Partnership.
OGR_other	Other.