

NICCI 3.2

DATA EXCHANGE

NICCI 3.2 ASYNCHRONOUS PROTOCOL TECHNICAL SPECIFICATION

Version of the documentation 1.7 dated 2024-05-10

Document status

Internal development

Keywords

Economic Information Bureau, BIG, Nicci, Tranche

Copyright © Krajowy Rejestr Długów, 2006-2024

Please send any corrections, comments and notes to pomocit@krd.pl



Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

Document attributes

	Attribute A	Value B
1	Project	Nicci 3.2
2	Title	Data exchange
3	Subtitle	Nicci 3.2 asynchronous protocol technical specification
4	Documentation version	1.7
5	Version time	2024-05-10
6	File	Nicci 3.2 - Technical specification.docx / pdf
7	Authors	Rafał Stramski, Piotr Piosik, Maciej Łukasik, Marcin Dawidziński, Katarzyna Spirovska, Robert Woszczyk
8	Supervision	Sebastian Tkocz, Maciej Łukasik
9	Copyright	Copyright © Krajowy Rejestr Długów, 2006-2024
10	Comments	

History of the document

	Attribute A	Value B	Date C
1	Documentation version	1.0	2021-07-13
2	Author	Piotr Piosik	
3	Content proofread by	Maciej Łukasik	
4	Form proofread by	Maciej Łukasik	
5	Approved by	Rafał Stramski	
6	Description	1. Correspondence address and residence address added for the positive information subject of the consumer type (Extension of the paidObligationConsumerType type)	
1	Documentation version	1.1	2021-07-22
2	Author	Katarzyna Spirovska	
3	Content proofread by	Maciej Łukasik	2021-08-20
4	Form proofread by	Maciej Łukasik	2021-08-20
5	Approved by	Maciej Łukasik	2021-08-20
6	Description	1. Change type of the positive information subject of the consumer type (Change the subjectType for the consumer to paidObligationConsumerType)	
1	Documentation version	1.2	2021-09-02
2	Author	Leysan Khamidullina	
3	Content proofread by	Maciej Łukasik	2021-09-13
4	Form proofread by	Maciej Łukasik	2021-09-13
5	Approved by	Maciej Łukasik	2021-09-13
6	Description	1. Change type of the positive information subject of the entrepreneur type (Change the subjectType for the entrepreneur to paidObligationEntrepreneurType)	
1	Documentation version	1.3	2022-01-17
2	Author	Aleksander Naporowski	
3	Content proofread by	Maciej Łukasik	2022-01-19
4	Form proofread by	Maciej Łukasik	2022-01-19
5	Approved by	Maciej Łukasik	2022-01-19
6	Description	1. Add additional properties to alimony obligation	
1	Documentation version	1.4	2022-07-05
2	Author	Maciej Łukasik	
3	Content proofread by	Maciej Łukasik	2022-07-05
4	Form proofread by	Maciej Łukasik	2022-07-05
5	Approved by	Maciej Łukasik	2022-07-05
6	Description	1. Change / Correction: first page, new description of the contractor – payer (positive information), description of the debtor (negative information), layout of the text, typos, layout and content of tables, description of the negative obligation (e.g. alimony, executive title), description of XSD schemas, description "monitorNotificationEnum type", description "monitorEventTypeEnum type", description of values with "MonitorCondition", description "processSnapshotType", description "searchManagementOrderType type", description "2.1. Management of orders referring to economic information" regarding the updating of economic information (adding information about converting types of obligations) etc.	

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

1	Documentation version	1.5	2023-04-19
2	Author	Maciej Łukasik	
3	Content proofread by	Maciej Łukasik	2023-04-19
4	Form proofread by	Maciej Łukasik	2023-04-19
5	Approved by	Maciej Łukasik	2023-04-19
6	Description	1. Change / Correction: description of monitoring (e.g. "2.3. Orders referring to monitoring").	
1	Documentation version	1.6	2023-09-20
2	Author	Marcin Dawidziński	
3	Content proofread by	Marcin Dawidziński	2023-09-20
4	Form proofread by	Marcin Dawidziński	2023-09-20
5	Approved by	Marcin Dawidziński	2023-09-20
6	Description	<p>1. Added a new type in section 3.4.3 "Definitions of types", i.e.: 3.4.3.30. "Type: debtExceedReportType" with a detailed description and connection of all elements such as Number, Created, SearchCriterion, Requester and IsDebtExceeded with places in the documentation where their types are described.</p> <p>2. Added a new "DebtExceedReport" element and its description in subsection 3.4.3.25. regarding the "Type: genericDisclosureReportType" type.</p>	
1	Documentation version	1.7	2024-05-10
2	Author	Marcin Dawidziński	
3	Content proofread by	Marcin Dawidziński	2024-05-10
4	Form proofread by	Marcin Dawidziński	2024-05-10
5	Approved by	Marcin Dawidziński	2024-05-10
6	Description	<p>1. In subsection 1.1.3.3. "Alimony obligation" has been changed in the content related to the description of the type of obligation and the possibility of transferring data. The required fields in "Table 9" have also been changed.</p> <p>2. In subsection 3.2.4.38. regarding the type "AlimonyObligationType", the mandatory nature of the "Date", "Signature" and "DecidingAuthority" elements has been removed.</p>	

Table of Contents

TABLE OF CONTENTS	4
TABLE LIST	5
INTRODUCTION	6
1. DATA SUBMITTED TO THE BUREAU	7
1.1. CASES.....	7
1.1.1. Debtor's data – negative information	8
1.1.2. Subject's (contractor's – payer's) data – positive information.....	11
1.1.3. Data of the unpaid obligations – negative information	13
1.1.4. Data of the paid obligation – positive information	19
2. ORDERS	20
2.1. MANAGEMENT OF ORDERS REFERRING TO ECONOMIC INFORMATION	20
2.2. ORDERS REFERRING TO ECONOMIC INFORMATION DISCLOSURE	21
2.3. ORDERS REFERRING TO MONITORING	21
2.4. ORDERS TO DISCLOSE BUSINESS INFORMATION IN INCOGNITO MODE.....	22
2.5. ORDERS TO DISCLOSE INFORMATION FROM THE INQUIRY REGISTER IN INCOGNITO MODE	22
3. XML SCHEMA DESCRIPTION	23
3.1. GENERAL TYPES FROM THE "XS" NAMESPACE	23
3.1.1. Simple types.....	23
3.2. NICCICOMMON.XSD FILE.....	24
3.2.1. Simple types.....	24
3.2.2. Definitions of attributes.....	31
3.2.3. Groups of elements.....	33
3.2.4. Complex types	37
3.3. NICCIINPUT.XSD FILE	55
3.3.1. Definitions of types.....	55
3.3.2. The main Input element	65
3.4. NICCIOUTPUT.XSD FILE	66
3.4.1. Definitions attribute	66
3.4.2. Simple types.....	66
3.4.3. Definitions of types.....	67

Table list

Table 1: Natural person data (debtor) – negative information	8
Table 2: Business entity data (debtor) – negative information.....	9
Table 3: One-man business data (debtor) – negative information.....	10
Table 4: Natural person data (subject – contractor – payer) – positive information.....	11
Table 5: Business entity data (subject – contractor – payer) – positive information	12
Table 6: One-man business data (subject – contractor – payer) – positive information	12
Table 7: Data of an ordinary obligation – negative information.....	14
Table 8: Data of an executive obligation – negative information.....	16
Table 9: Data of an alimony obligation – negative information	17
Table 10: Data of a paid obligation – positive information	19

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

Introduction

The KRD system provides the services of the Economic Information Bureau pursuant to the Act on Disclosure of Economic Information and Exchange of Economic Data, dated 9 April, 2010. It enables adding, updating and deleting economic information and sharing it with third parties.

The economic information may be entered into the KRD system with one of the two available methods: interaction on the website of the bureau – KRD WWW Customer Panel (manually or via CSV files) and with XML files (with the appropriate orders).

This document describes the structure of the data files transferred to the bureau in tranches and the structure of the report on the implementation of ordered services. The scheme of the XML files containing this data is also described in the present document. Finally, the modes of swapping tranches along with the authorization methods are described.

NOTE!

Please send us one aggregate tranche of Nicci 3.2 / several aggregate tranches of Nicci 3.2 at the maximum.

Sending many individual tranches with an order, e.g. adding only one case, means that their waiting for processing / processing on our side takes longer (sometimes much longer) than the processing of one aggregate tranche with multiple orders, such as case additions.

That is why, the tutorial which you receive from us in the implementation process, shows how multiple cases and related obligations are added through one aggregate tranche. Update, removal, suspension, unsuspension and the like orders should be handled in a similar way. All orders may, and even should, be included in one aggregate tranche.

Additionally, we ask for:

- aggregation of negative obligations of the same debtor in one negative case instead of e.g. several negative cases,
- aggregation of positive obligations of the same contractor in one positive case instead of e.g. several positive cases.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

1. Data submitted to the bureau

The data which may be submitted to the Economic Information Bureau is specified by the Act on Disclosure of Economic Information and Exchange of Economic Data, dated 9 April, 2010. It defines the minimum and maximum data range and also additional conditions to be fulfilled for the bureau to accept such data.

The bureau can accept three types of economic information.

The first type is information about the unpaid obligations, referring to both business entities and natural persons. They are grouped for each debtor in cases. A case contains information on the debtor, creditor (who may reserve the disclosure of their data) and on the unpaid obligations of the debtor to the creditor.

The concept of a case is visible only to the Client submitting this case and to the KR D. For third parties who have granted access to economic information of a debtor, the obligations of one case will be visible as not linked to economic information referring to unpaid obligations (apart from the creditor's data if the disclosure of these is not reserved).

The second type of economic information is the information on paid obligations (before the due date, on the due date or up to the 29th day after the due date). Payments settled no later than 29 days after the due date based on the contractor's consent (it is possible to add positive information about the settled obligation within 14 days from the due date) or the application received from him (within 14 days of receiving the application, the creditor is obliged to provide positive information about the obligations settled by the contractor in the last 12 months) can be added to the BIG database.

Important! There must be no more than 12 months between the paid (repayment) date and the addition date, and no more than 29 days between the due date and the paid (repayment) date. The system will not accept such cases (in which at least one obligation does not meet the above requirements) and will return an error in the result tranche Nicci 3.2.

The third type of economic information is the information about using a forged or someone else's document in reference to the Client.

1.1. Cases

A case is a group of economic negative / positive information referring to the obligations of one debtor / contractor – payer towards the Client submitting the information. The case is reported to the bureau as a whole, i.e., it must contain all obligations. After submitting the case to the bureau, it can be removed or updated. Updating a case consists in modifying the debtor's / contractor's – payer's data or, for example, modifying / removing one of the obligation in this case.

The case contains an optional user identifier (any sequence of a maximum length of 128 characters) given by the creditor (provider), the debtor's / contractor's – payer's data, and the data of the obligations (at least one).

The concept of a case refers to information about both unpaid (negative information) and paid (positive information) obligations, however, those types of obligations cannot be mixed in one case.

1.1.1. Debtor's data – negative information

A debtor can be a natural person (consumer), an entity with legal personality (legal person) or a one-man business (entrepreneur). Each type of debtor has specified requirements as to the submitted data.

1.1.1.1. A natural person (a consumer)

The required data of a natural person are: first name and surname, and the Polish Resident Identification Number PESEL (for the Polish citizens) or a different identification number (for foreigners). Other optional data include the middle name, residence address, mailing address, and the data of an additional identity document.

The address contains from 2 to 4 lines, including, for example, street and house number, as well as postal code and city. For foreign addresses, it is recommended to input the name or the code of the country in the last line. Preferred filling of the Nicci 3.2 tranche: first line: street and number, second line: postal code and city.

Table 1: Natural person data (debtor) – negative information

	Field name	Description	Notes	Required
	A	B	C	D
1	FirstName	First name		YES
2	Surname	Surname		YES
3	IdentityNumber	Identity number (PESEL for Polish citizens)		YES
4	SecondName	Middle name		NO
5	Address	Residence address		NO
6	MailingAddress	Mailing address		NO
7	DocumentNumber	Additional identity document		NO

1.1.1.2. Business entity (legal personality)

For this type of entity, the required data are designation (name), tax identification number for entities based in the European Union or any other identification number for other entities, and seat address. Optional data are the number under which the entity is entered into a relevant register (along with the denomination of the registry court), the REGON number (Polish National Business Registry Number) and the main object of business activity.

The tax identification number for entities from the European Union should be submitted as a sequence of digits in the format relevant for the given country, preceded by a two-character country code. For the Polish numbers, the country code can be omitted. In the case of entities that do not have a European tax identification number, a different identification number should be submitted along with its type. We accept the REGON number in the 9-digit version.

The address contains from 2 to 4 lines, including, for example, street and house number, as well as postal code and city. For foreign addresses, it is recommended to input the name or the code of the country in the last line. Preferred filling of the Nicci 3.2 tranche: first line: street and number, second line: postal code and city.

Additional data may be also submitted, such as first names, surnames, and identification numbers (PESEL for the Polish citizens) of the partners, board members, proxies, and first names and surnames of the authorized representatives of this entity.

Table 2: Business entity data (debtor) – negative information

	Field name	Description	Notes	Required
	A	B	C	D
1	Name	Name (designation) of the company		YES
2	IdentityNumber	Tax identification number (EU tax identification number for entities from the European Union)		YES
3	SeatAddress	Seat address		YES
4	RegistrationNumber	Registry number	Must appear together	NO
5	RegistryName	The name of the registry containing the above number		
6	Regon	The REGON number of the company		NO
7	Ekd	EKD (NACE) / PKD number (European / Polish classification of economic activities)		NO
8	Persons	List of partners, board members, proxies		NO
9	Representatives	List of authorized representatives		NO

1.1.1.3. One-man business

The required data for this type of entity include the designation (name), the tax identification number for entities from European Union or a different identification number for other entities, the first and last name of the owner and one of the three addresses; the address of pursuing economic activity, residence address, mailing address. Optionally, this data can be supplemented with the full details of the business entity and the data, analogous to in the case of a consumer (concerning the owner).

The address contains from 2 to 4 lines, including, for example, street and house number, as well as postal code and city. For foreign addresses, it is recommended to input the name or the code of the country in the last line. Preferred filling of the Nicci 3.2 tranche: first line: street and number, second line: postal code and city.

Table 3: One-man business data (debtor) – negative information

	Field name	Description	Notes	Required
	A	B	C	D
1	Name	Name (designation) of the company	Must include the first and last name of the owner	YES
2	NonConsumerIdentityNumber	Tax identification number (EU tax identification number for entities from the European Union)		YES
3	FirstName	Owner's first name		YES
4	Surname	Owner's surname		YES
5	SecondName	Owner's middle name		NO
6	MailingAddress	Mailing address		YES / NO
7	ResidenceAddress	Residence address		YES / NO
8	SeatAddress	The address of pursuing economic activity		YES / NO
9	ConsumerIdentityNumber	Identity Owner's number (PESEL for Polish citizens)		NO
10	DocumentNumber	Additional identity document of the owner		NO
11	RegistrationNumber	Registry number	Must appear together	NO
12	RegistryName	The name of the registry containing the above number		
13	Regon	The REGON number of the company		NO
14	Ekd	EKD (NACE) / PKD number (European / Polish classification of economic activities)		NO
15	Representatives	List of authorized representatives		NO

1.1.2. Subject's (contractor's – payer's) data – positive information

A subject can be a natural person (consumer), an entity with legal personality (legal person) or a one-man business (entrepreneur). Each type of subject has specified requirements as to the submitted data.

1.1.2.1. A natural person (a consumer)

The required data of a natural person are: first name and surname, and the Polish Resident Identification Number PESEL (for the Polish citizens) or a different identification number (for foreigners). Other optional data include the middle name, residence address and mailing address.

The address contains from 2 to 4 lines, including, for example, street and house number, as well as postal code and city. For foreign addresses, it is recommended to input the name or the code of the country in the last line. Preferred filling of the Nicci 3.2 tranche: first line: street and number, second line: postal code and city.

Table 4: Natural person data (subject – contractor – payer) – positive information

	Field name	Description	Notes	Required
	A	B	C	D
1	FirstName	First name		YES
2	Surname	Surname		YES
3	IdentityNumber	Identity number (PESEL for Polish citizens)		YES
4	SecondName	Middle name		NO
5	Address	Residence address		NO
6	MailingAddress	Mailing address		NO

1.1.2.2. Business entity (legal personality)

For this type of entity, the required data are designation (name), tax identification number for entities based in the European Union or any other identification number for other entities, and seat address.

The tax identification number for entities from the European Union should be submitted as a sequence of digits in the format relevant for the given country, preceded by a two-character country code. For the Polish numbers, the country code can be omitted. In the case of entities that do not have a European tax identification number, a different identification number should be submitted along with its type.

The address contains from 2 to 4 lines, including, for example, street and house number, as well as postal code and city. For foreign addresses, it is recommended to input the name or the code of the country in the last line. Preferred filling of the Nicci 3.2 tranche: first line: street and number, second line: postal code and city.

Table 5: Business entity data (subject – contractor – payer) – positive information

	Field name	Description	Notes	Required
	A	B	C	D
1	Name	Name (designation) of the company		YES
2	IdentityNumber	Tax identification number (EU tax identification number for entities from the European Union)		YES
3	SeatAddress	Seat address		YES

1.1.2.3. One-man business

The required data for this type of entity include the designation (name), the tax identification number for entities from European Union or a different identification number for other entities and one of the three addresses; the address of pursuing economic activity, residence address, mailing address.

The address contains from 2 to 4 lines, including, for example, street and house number, as well as postal code and city. For foreign addresses, it is recommended to input the name or the code of the country in the last line. Preferred filling of the Nicci 3.2 tranche: first line: street and number, second line: postal code and city.

Table 6: One-man business data (subject – contractor – payer) – positive information

	Field name	Description	Notes	Required
	A	B	C	D
1	Name	Name (designation) of the company	Must include the first and last name of the owner	YES
2	NonConsumerIdentityNumber	Tax identification number (EU tax identification number for entities from the European Union)		YES
3	MailingAddress	Mailing address		YES / NO
4	ResidenceAddress	Residence address		YES / NO
5	SeatAddress	The address of pursuing economic activity		YES / NO

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

1.1.3. Data of the unpaid obligations – negative information

An obligation is defined with at least three elements: the legal title of the obligation, the amount and the currency of the arrears, and the date of origin of the arrears.

The obligation data may also be supplemented with objections regarding the obligation. Objections are represented by two dictionary values encoded in "Objections" field. This information is encoded in format: X-Y where X is a digit representing dictionary value of objections of limitation regarding obligation, while Y is a digit representing dictionary value of objections of questioning obligation by the debtor.

X can be assigned values:

- 1 – No objection of limitation regarding obligation
- 2 – Objection of limitation regarding part of the obligation
- 3 – Objection of limitation regarding entirety of the obligation.

Y can be assigned values:

- 1 – No objection of questioning the obligation
- 2 – Objection of questioning part of the obligation
- 3 – Objection of questioning entirety of the obligation

E.g., value 3-1 send in "Objections" field will be interpreted as: Objection of limitation regarding entirety of the obligation and No objection of questioning the obligation.

Other text values inserted into "Objections" field will be interpreted as objection of questioning the entirety or a part of obligation („Kwestionowanie całości lub części zobowiązania”). Value 1-1 inserted in "Objections" field is invalid. If there are no objections regarding obligation, the element "NoObjections" should be used instead (with the "true" value).

The obligation data may also be supplemented with information about the proceedings regarding the obligation in the "Proceedings" field. This field takes values of "string" type, and the information is encoded as a dictionary value represented by a single digit placed in this field or sent as any text value (any text value will be saved in the system but will not be displayed in KRD reports, as opposed to the dictionary values given below).

Field can be assigned values:

- 1 – Civil court proceeding
- 2 – Arbitration court proceeding
- 3 – Administrative proceeding
- 4 – Administrative court proceeding
- 5 – Criminal proceeding
- 6 – Enforcement proceeding

Obligations are divided into three types: ordinary obligations, declared with an executive title, and alimony.

When submitting the obligation data to the bureau, the Client must specify the date of sending a call for payment of this obligation to the debtor. The call for payment must include a warning of the intention to submit the data to the economic data bureau, with the name and address of this bureau included.

1.1.3.1. Ordinary obligation

For the ordinary obligation information to be accepted by the bureau, the submitted data must satisfy following 4 conditions (apart from the delivery of the minimum data):

- The obligation is due from at least 30 days (which means that "PaymentDate" was 30 days ago).
- At least 30 days have passed from the date on which the call for payment was sent / delivered to the debtor with the warning about the intention to add the obligation to the KRD (meaning from the "CallSent" date).
- The total amount of the debtor's obligations ("Arrears") to the creditor exceeds 200 PLN if the debtor is a consumer and 500 PLN if the debtor is an economic entity (legal person / entrepreneur).
- For ordinary obligations, the date of debt occurrence ("PaymentDate") cannot be older than 6 years from the date of sending the information to the bureau in the case of consumers, this requirement does not apply to an economic entity (legal persons / entrepreneurs) (the Act does not indicate a maximum payment date).

Of course, the other conditions specified in the Act and not specified in the data must also be satisfied (e.g. holding a permission to deliver economic information about consumers).

Table 7: Data of an ordinary obligation – negative information

	Field name	Description	Notes	Required
	A	B	C	D
1	Title	Legal title of the obligation, e.g. "Invoice 145/A/02"	A group of elements depending on the type of obligation	YES / NO
2	Type	Type of obligation The legal title consists of two fields: "Title" and "Type"; at least one of them must be non-empty.	Type of obligation	YES / NO
3	CustomType	Another type of obligation	Any text (max. 128 characters)	NO
4	CallSent	Date of sending or delivering a call for payment to the debtor with the warning about the intention to add the obligation to the KRD	See Type: date	YES
5	PaymentDate	Payment date (date by which the obligation should be repaid)	See Type: date	YES

6	Debt	Amount and currency of the obligation	See moneyType	NO
7	Arrears	Amount and currency of the arrears	See moneyType	YES
8	Proceedings	Information about current proceedings regarding the obligation. Can be encoded by a single digit representing dictionary value .	If there are no pending proceedings regarding the obligation - this field in the tranche can be omitted	YES / NO
9	NoObjections	Obligatory when the debtor does not dispute the obligation. Then it takes the "true" value. When the debtor disputes the obligation, the element is absent and the "Objections" element is obligatory.	The field is absent when the completed "Objections" element appears	YES / NO
10	Objections	Obligatory when the debtor disputes the obligation (no "NoObjections" element). This field accepts values of "string" type. This information is encoded in format: X-Y where X and Y are digits representing dictionary value of objections of limitation regarding obligation and value of objections of questioning obligation.	The field is absent when "NoObjections" is "true"	YES / NO
11	Note	An optional note [contains additional information about the obligation (not disclosed in KRD reports)]	Can be empty	NO

1.1.3.2. Executive obligation

This type of obligation is intended for the creditors who have an executive title to the obligation, such as a court ruling. When submitting such an obligation to the KRD, the following should be provided: the type of the executive title ("Type"), the date of issue ("Date"), signature ("Signature"), the name of the issuing authority ("DecidingAuthority"). It is also necessary to provide the date of sending or delivering a warning to the debtor about the intention to add the obligation to the KRD ("CallSent") and the date by which the obligation should be repaid (payment date - "PaymentDate" element).

For the executive obligation information to be accepted by the bureau, the submitted data must satisfy following 3 conditions (apart from the delivery of the minimum data):

- At least 14 days have passed from the date of sending or delivering to the debtor a warning about the intention to add the obligation to the KRD (meaning from the "CallSent" date).
- The total amount of executive obligations ("Arrears") must be only greater than zero.
- The date of establishing the executive title ("Date") cannot be older than 6 years from the date of sending the information to the bureau in the case of entrepreneurs / legal persons and consumers.

Of course, the other conditions specified in the Act and not specified in the data must also be satisfied (e.g. holding a permission to deliver economic information about consumers).

Table 8: Data of an executive obligation – negative information

	Field name	Description	Notes	Required
	A	B	C	D
1	Type	Type of executive obligation	Type of executive obligation	YES
2	CustomType	Another type of obligation	Any text (max. 128 characters)	NO
3	Date	Date of establishing the executive title	See Type: date	YES
4	Signature	Signature	See text	YES
5	DecidingAuthority	Deciding authority	See text	YES
6	CallSent	Date of sending or delivering to the debtor a warning about the intention to add the obligation to the KRD	See Type: date	YES
7	PaymentDate	Payment date (date by which the obligation should be repaid)	See Type: date	YES
8	Debt	Amount and currency of the obligation	See moneyType	NO
9	Arrears	Amount and currency of the arrears	See moneyType	YES
10	Proceedings	Information about current proceedings regarding the obligation. Can be encoded by a single digit representing dictionary value .	If there are no pending proceedings regarding the obligation - this field in the tranche can be omitted	YES / NO
11	NoObjections	Obligatory when the debtor does not dispute the obligation. Then it takes the "true" value. When the debtor disputes the obligation, the element is absent and the "Objections" element is obligatory.	The field is absent when the completed "Objections" element appears	YES / NO
12	Objections	Obligatory when the debtor disputes the obligation (no "NoObjections" element). This field accepts values of "string" type. This information is encoded in format: X-Y where X and Y are digits representing dictionary value of objections of limitation regarding obligation and value of objections of questioning obligation.	The field is absent when "NoObjections" is "true"	YES / NO
13	Note	An optional note [contains additional information about the obligation (not disclosed in KRD reports)]	Can be empty	NO

1.1.3.3. Alimony obligation

Alimony obligations can be only linked to a debtor being a consumer ("Consumer").

This type of obligation is intended for the creditors who have an executive title to the obligation. When submitting such an obligation to the KRD, it is possible to enter additional data such as: the date of establishing the executive title ("Date"), signature ("Signature"), the name of the issuing authority ("DecidingAuthority"). It is also possible to indicate the date of sending or delivering a warning to the debtor about the intention to add the obligation to the KRD ("CallSent") and the date by which the obligation should be repaid (payment date – "PaymentDate" element).

For the alimony obligation information to be accepted by the bureau, the submitted data must satisfy following 2 conditions (apart from the delivery of the minimum data):

- At least 14 days have passed from the date of sending or delivering to the debtor a warning about the intention to add the obligation to the KRD (meaning from the "CallSent" date).
- The total amount of alimony obligation ("Arrears") must be only greater than zero.

Of course, the other conditions specified in the Act and not specified in the data must also be satisfied (e.g. holding a permission to deliver economic information about consumers).

Table 9: Data of an alimony obligation – negative information

	Field name	Description	Notes	Required
	A	B	C	D
1	Date	Date of establishing the executive title	See Type: date	NO
2	Signature	Signature	See text	NO
3	DecidingAuthority	Deciding authority	See text	NO
4	CallSent	Date of sending or delivering to the debtor a warning about the intention to add the obligation to the KRD	See Type: date	NO
5	PaymentDate	Payment date (date by which the obligation should be repaid)	See Type: date	YES
6	Debt	Amount and currency of the obligation	See moneyType	NO
7	Arrears	Amount and currency of the arrears	See moneyType	YES

8	Proceedings	<p>Information about current proceedings regarding the obligation.</p> <p>Can be encoded by a single digit representing dictionary value.</p>	<p>If there are no pending proceedings regarding the obligation - this field in the tranche can be omitted</p>	YES / NO
9	NoObjections	<p>Obligatory when the debtor does not dispute the obligation. Then it takes the "true" value.</p> <p>When the debtor disputes the obligation, the element is absent and the "Objections" element is obligatory.</p>	<p>The field is absent when the completed "Objections" element appears</p>	YES / NO
10	Objections	<p>Obligatory when the debtor disputes the obligation (no "NoObjections" element). This field accepts values of "string" type.</p> <p>This information is encoded in format: X-Y where X and Y are digits representing dictionary value of objections of limitation regarding obligation and value of objections of questioning obligation.</p>	<p>The field is absent when "NoObjections" is "true"</p>	YES / NO
11	Note	<p>An optional note [contains additional information about the obligation (not disclosed in KRD reports)]</p>	<p>Can be empty</p>	NO

1.1.4. Data of the paid obligation – positive information

When submitting information on an obligation of this type it is required to include the information about the reason of it (legal title of the obligation), the amount and currency of repayment, the amount and currency of the due, due date, paid date, whether it is the sum of smaller repayments. There must be no more than 12 months between the paid (repayment) date and the addition date, and no more than 29 days between the due date and the paid (repayment) date.

Table 10: Data of a paid obligation – positive information

	Field name	Description	Notes	Required
	A	B	C	D
1	Reason	Reason (legal title of the obligation) – what the obligation referred to	Amounts: Bill Contract Invoice	NO
2	OtherReason	Other reason (legal title of the obligation) – what the obligation referred to	Any text (max. 32 characters)	NO
3	PaidDebt	Amount and currency repaid	See moneyType	YES
4	TotalDebt	Amount and currency due	See moneyType	YES
5	DueDate	Repayment due date (date by which the obligation should be repaid)	See Type: date	YES
6	PaidDate	Repayment date (date of repayment of the obligation)	See Type: date	YES
7	IsSumOfInstallments	A flag defining if the information refers to a sum of obligations amounts (whether the obligation has been repaid in installments)	See Type: boolean	YES

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

2. Orders

2.1. *Management of orders referring to economic information*

Orders for providing services can be submitted to an economic information bureau. These services refer to economic information management: addition, modification, deletion (negative / positive information), suspending and restoring the publication of negative economic information.

Addition or modification of economic information consists in submitting the data of this information (new or modified) with the use of "AddInformation" or "UpdateInformation" element.

As part of the "UpdateInformation" element, it is also possible to convert the type of obligation from *Obligation* (ordinary unpaid obligation – negative information) to *ExecutiveObligation* (unpaid obligation with an executive title – negative information) or from *ExecutiveObligation* to *Obligation*. The described update method works by updating the whole case (*UpdateCase*) and updating the case obligation (*UpdateObligation* / *UpdateExecutiveObligation*). The change of the type of obligation does not affect its original date of adding to the KRD.

Deleting, suspending or restoring negative economic information consists in submitting the agreed (while delivering this information) information identifier within the relevant elements identifying the orders. These are the following elements: "RemoveInformation", "SuspendInformation" and "UnsuspendInformation". Additionally, in the case of suspending the publication of economic information, it is required to specify the date until which the publication will be suspended. Specifying this date is obligatory, however, it is of course possible to set it as a very distant date, e.g. 2048-12-31.

It is possible for the Client to obtain a list of economic information added by him / her. This operation might be useful in order to verify the published data or to check their validity. The list of information can be obtained with the "GetInformations" element. Moreover, it is possible to order verification of the correctness of the submitted negative data, e.g. with the KRD database using "GetInformationsVerificationEvents".

Every piece of information has its owner that is the login who manages this piece of information. The Client can change the owner at any time using the "ChangeInformationOwner" function.

All orders referring to economic information require specifying the identifier of the given information. The identifier can be specified by the user or generated by the system. The type of the used identifier must be included in the "IDType" attribute, which takes the value "UserId" (user-specified) and "SystemId" (system-generated). If this attribute is not specified, the system will assume the "UserId" value.

The second attribute necessary for identifying the information is "informationType". It can include the following values: "Case" (a case containing a set of unpaid obligations – negative informations), "ObligationInformation" (unpaid obligation – negative information), "ExecutiveObligationInformation" (unpaid executive obligation – negative information), "AlimonyObligationInformation" (unpaid alimony obligation – negative information), "PaidObligationCase" (a case containing a set of paid obligations – positive informations) or "PaidObligationInformation" (paid obligation – positive information).

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

2.2. Orders referring to economic information disclosure

The Nicci 3.2 protocol, allows the Clients to obtain economic information about other entities (KRD reports).

In order to do this, a request for disclosure of economic information must be sent in the first place using the "SearchNonConsumer" order. In this element, the search criteria are specified, and they determine the type of information searched for by the system. It is possible to perform a search according to tax identification number for entities based in European Union or a different number for foreign entities.

In order to obtain economic information of consumers, "SearchConsumer" order must be used with the PESEL number for the Polish citizens or a different number for foreigners as a search criterion. Additionally, searching for the information about consumers is also possible only after obtaining a relevant authorization (only institutions such as Supreme Audit Office or public prosecutor's office do not need it). The information about the possession of such authorizations is declared by the Client, who must specify the date of obtaining the authorization. Such authorization is valid for 60 days.

A report with economic information satisfying the requirements of the Client is saved in the system and is available for the Client also later (for a specified period of time). The Client can download the list of his / her reports with economic information ordering the "GetDisclosureReports" task or view the details of a specific report with "GetDisclosureReport".

2.3. Orders referring to monitoring

The bureau allows its Clients to monitor events, which are related to an appropriate Tax Identification Number (Tax ID). When somebody performs certain operation connected with monitored Tax ID (e.g. adding to KRD), Client will be notified about it. The method of notification is chosen by the Client, and it can be done by sending an e-mail to an adequate address and / or by sending a short SMS message (the latter is configured in the KRD WWW Customer Panel in tab: Monitoring -> Monitoring settings).

Adding and updating monitoring conditions (relates to monitored tax identification numbers) is executed by means of orders described by elements "AddMonitorCondition" and "UpdateMonitorCondition", respectively. It is possible to stop (remove) further monitoring at any time by using "StopMonitorCondition".

List of monitoring conditions that the Client has added can be downloaded by ordering the "GetMonitorConditions" task.

For retrieving the list of events from the system, the "GetEvents" element is used. In response, you get a list of events generated by the system on the basis of active monitoring conditions. Such a list usually contains events from the previous day, but can also contain events from "today" (the event-generating component starts at around 03:00, and if there are changes in economic information by that hour, they will be included). Informatively – the "Date" element presents the date of processing / generating of the monitoring event (not the date of the event itself). The same events are also placed in KRD WWW Customer Panel (tab: Monitoring -> Monitoring events list) and can be delivered in the aforementioned e-mail.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

2.4. Orders to disclose business information in incognito mode

Using the Nicci 3.2 protocol, Clients who are authorized entities can obtain economic information about other entities (KRD reports) in incognito mode. Authorized entities are defined in Art. 25 of the Act on Disclosure of Economic Information and Exchange of Economic Data, dated 9 April, 2010.

Incognito mode ensures the anonymity of the inquiry (request), which means that the inquiry will not be saved in the inquiry register (requests history) of the investigated entity.

To download economic information about a business entity, use the "IncognitoSearchNonConsumer" order. Here, you enter the information search criteria, which determine the type of information the system will look for. You can search by tax identification number for European Union entities or other number for foreign entities.

To download economic information about consumers, use the "IncognitoSearchConsumer" order and enter the search criterion as a PESEL number for Polish citizens or other identification number for a foreign consumer.

Unlike in the disclosure of economic information in the normal mode, in incognito mode it is impossible to re-download the report by its number (no "GetDisclosureReport" operation).

2.5. Orders to disclose information from the inquiry register in incognito mode

Using the Nicci 3.2 protocol, Clients who are authorized entities can obtain information from the inquiry register (requests history) for business entities and consumers. This operation consists in disclosing information about to whom and what economic information was disclosed on a given business entity or consumer. Authorized entities are defined in Art. 25 of the Act on Disclosure of Economic Information and Exchange of Economic Data, dated 9 April, 2010.

Incognito mode ensures the anonymity of the inquiry (request), which means that the inquiry will not be saved in the inquiry register (requests history) of the investigated entity.

To download information from the business entity inquiry register (requests history), use the "IncognitoSearchRegistryReportNonConsumer" order. Here, you enter the information search criteria, which determine the type of information the system will look for. You can search by tax identification number for European Union entities or other number for foreign entities.

To download information from the consumer inquiry register (requests history), use the "IncognitoSearchRegistryReportConsumer" order. Here, you enter the consumer search criterion as the PESEL number for Polish citizens or other identification number for a foreign consumer.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

3. XML schema description

This part of the document contains the description of XSD files elements that make up the XML schema definition of the data exchange files.

Item [3.1](#) contains a description of types and groups used in data exchange. Item [3.2](#) contains a description of the definition of the file with orders to be fulfilled sent by the Clients to the bureau. Item [3.3](#) contains the definitions of the types and elements which are used by the Clients when creating the import files i.e. the files sent to the bureau. The definition of the reply sent by the bureau to the Clients is described in item [3.4](#).

3.1. General types from the "xs" namespace

3.1.1. Simple types

3.1.1.1. Type: string

Any string of characters

3.1.1.2. Type: int

Any integer in the range from -2147483648 to 2147483648

3.1.1.3. Type: positiveInteger

Any integer in the range from 1 to 2147483648

3.1.1.4. Type: nonNegativeInteger

Any integer greater than or equal 0 and less than 2147483648

3.1.1.5. Type: decimal

Floating point number. As a separator, always use the "." symbol, e.g. 2424.2343

3.1.1.6. Type: boolean

Logical value. Accepted values are "true" or "1" (true) and "false" or "0" (false)

3.1.1.7. Type: date

Date in the ISO 8601 format ([-]CCYY-MM-DD[Z](+|-)hh:mm]), e.g. 2001-10-26

3.1.1.8. Type: dateTime

Date and time in the ISO 8601 format ([-]CCYY-MM-DDThh:mm:ss[Z](+|-)hh:mm]), e.g. 2001-10-26T21:32:52

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

3.2. *nicciCommon.xsd* file

The file contains definitions of the basic types and all elements and attributes which are common both for the input file and the response file.

3.2.1. Simple types

The simple types define the requirements for single values, such as a tax identification number or a date. The requirements include the minimum and maximum length of the value and also its format. The format is described with regular expressions ("regex").

3.2.1.1. Type: nonEmptyString

```
<xs:simpleType name="nonEmptyString">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>
```

A type indicating a non-empty string.

3.2.1.2. Type: nonEmptyStringXXXX

```
<xs:simpleType name="nonEmptyString16">
  <xs:restriction base="nonEmptyString"/>
    <xs:maxLength value="16"/>
  </xs:restriction>
</xs:simpleType>
```

Types indicating a non-empty string of specified maximum length, where XXXX is the maximum length of the string.

3.2.1.3. Type: orderIDType

```
<xs:simpleType name="orderIDType">
  <xs:restriction base="nonEmptyString128"/>
</xs:simpleType>
```

The elements of this type contain a unique identifier of the tasks ordered by the Client of the bureau. The type is compliant with [nonEmptyString128](#).

3.2.1.4. Type: loginNameType

```
<xs:simpleType name="loginNameType">
  <xs:restriction base="nonEmptyString128"/>
</xs:simpleType>
```

The elements of this type contain the system user identifier (login). In terms of value, the type is compliant with [nonEmptyString128](#).

3.2.1.5. Type: peselType

```
<xs:simpleType name="peselType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{11}"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type contain the PESEL number. The value of this element is not checked in terms of correctness (apart from confirming it contains only digits and apart from confirming it contains the same digits, eg. 11111111111 – the second verification takes place during the processing of the Nicci tranche), but only in terms of length. A similar rule applies to Tax ID.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

3.2.1.6. Type: regonType

```
<xs:simpleType name="regonType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{9}|\d{7}"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type contain the REGON number. The value of this element is not checked in terms of correctness (apart from confirming it contains only digits), but only in terms of length.

3.2.1.7. Type: ekdType

```
<xs:simpleType name="ekdType">
  <xs:restriction base="nonEmptyString16"/>
</xs:simpleType>
```

The elements of this type contain the EKD (NACE) / PKD number (European / Polish classification of economic activities). The type is compliant with [nonEmptyString16](#).

3.2.1.8. Type: decimalType

```
<xs:simpleType name="decimalType">
  <xs:restriction base="xs:decimal">
    <xs:pattern value="-?\d+[.]?\d*" />
  </xs:restriction>
</xs:simpleType>
```

The elements of this contain a floating point number in which the fractional part (if any) is divided by a point. Additionally, the number can be preceded with the '-' symbol (ASCII 45 code).

3.2.1.9. Type: emailType

```
<xs:simpleType name="emailType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="64"/>
    <xs:pattern value="[a-zA-Z0-9][-_a-zA-Z0-9\.\.]+@[a-zA-Z0-9][-_a-zA-Z0-9\.\.]*[a-zA-Z]"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type contain the e-mail address.

3.2.1.10. Type: phoneNumberType

```
<xs:simpleType name="phoneNumberType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="32"/>
    <xs:pattern value="(\+\d+)?(\(\d+\))?[\d -]*/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type contain the phone number.

3.2.1.11. Type: versionType

```
<xs:simpleType name="versionType">
  <xs:restriction base="xs:string">
    <xs:pattern value="3\.\d"/>
  </xs:restriction>
</xs:simpleType>
```

The type is used to specify the version of the XML schema. In this version of the document, the only possible value is '3.2'.

3.2.1.12. Type: intMaxType

```
<xs:simpleType name="intMaxType">
  <xs:union memberTypes="xs:positiveInteger">
    <xs:simpleType>
      <xs:restriction base="nonEmptyString8">
        <xs:enumeration value="max"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

The elements of this type contain integer greater than or equal 0. Additionally, instead of a specific number you can enter the "max", which means an order to download the full list of data elements.

3.2.1.13. Type: addressTypeEnum

```
<xs:simpleType name="addressTypeEnum">
  <xs:restriction base="nonEmptyString32">
    <xs:enumeration value="Regular"/>
    <xs:enumeration value="Registered"/>
  </xs:restriction>
</xs:simpleType>
```

This element is used to specify the type of mailing address. There are two available values:

- *Regular* – mailing address to which a notification about adding the debtor to KRD is to be sent by regular mail (for information – currently we do not send letters abroad),
- *Registered* – mailing address to which a notification about adding the debtor to KRD is to be sent by registered mail (for information – currently we do not send letters abroad).

3.2.1.14. Type: idTypeEnum

```
<xs:simpleType name="idTypeEnum">
  <xs:restriction base="nonEmptyString32">
    <xs:enumeration value="UserId"/>
    <xs:enumeration value="SystemId"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type contain the case / obligation identifier type. There are two available values:

- *UserId* – identifier given by the user,
- *SystemId* – identifier generated by the system.

3.2.1.15. Type: personRoleEnum

```
<xs:simpleType name="personRoleEnum">
  <xs:restriction base="nonEmptyString64">
    <xs:enumeration value="BoardMember"/>
    <xs:enumeration value="Partner"/>
    <xs:enumeration value="Proxy"/>
  </xs:restriction>
</xs:simpleType>
```

This element is used to specify the type of person related to the debtor. Accepted values:

- *BoardMember* – member of the board,
- *Partner* – partner,
- *Proxy* – proxy.

3.2.1.16. Type: informationTypeEnum

```
<xs:simpleType name="informationTypeEnum">
  <xs:restriction base="nonEmptyString32">
    <xs:enumeration value="Case"/>
    <xs:enumeration value="ObligationInformation"/>
    <xs:enumeration value="PaidObligationInformation"/>
    <xs:enumeration value="PaidObligationCase"/>
    <xs:enumeration value="DocumentInformation"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type specify the type of the submitted information. Accepted values:

- *Case* – a case containing a set of unpaid obligations (negative informations),
- *ObligationInformation* – unpaid obligation (negative information),
- *PaidObligationInformation* – paid obligation (positive information),
- *PaidObligationCase* – a case containing a set of paid obligations (positive informations),
- *DocumentInformation* – information about using a forged / someone else's document.

3.2.1.17. Type: removeReasonType

```
<xs:simpleType name="removeReasonType">
  <xs:restriction base="nonEmptyString16">
    <xs:enumeration value="Paid"/>
    <xs:enumeration value="Cession"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

This element is used to specify the reason for deleting the unpaid obligation (negative information). Accepted values:

- *Paid* – paid obligation,
- *Cession* – cession,
- *Other* – a different reason.

3.2.1.18. Type: obligationTypeEnum

```
<xs:simpleType name="obligationTypeEnum">
  <xs:annotation>
    <xs:documentation>Predefined list of obligation types</xs:documentation>
  </xs:annotation>
  <xs:restriction base="nonEmptyString32">
    <xs:enumeration value="Invoice"/>
    <xs:enumeration value="Bill"/>
    <xs:enumeration value="Contract"/>
    <xs:enumeration value="StipulatedPenalty"/>
    <xs:enumeration value="InterestNote"/>
    <xs:enumeration value="Bond"/>
    <xs:enumeration value="DebtAcknowledgment"/>
    <xs:enumeration value="Agreement"/>
    <xs:enumeration value="CourtDecision"/>
    <xs:enumeration value="WritOfExecution"/>
    <xs:enumeration value="LegalCosts"/>
    <xs:enumeration value="ExecutionCosts"/>
  </xs:restriction>
</xs:simpleType>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

This element is used to specify the type of the unpaid obligation (negative information). Accepted values:

- *Invoice* – a VAT invoice,
- *Bill* – a bill,
- *Contract* – a contract,
- *StipulatedPenalty* – a stipulated penalty,
- *InterestNote* – an interest note,
- *Bond* – a bond,
- *DebtAcknowledgment* – a debt acknowledgement,
- *Agreement* – an agreement,
- *CourtDecision* – a court decision,
- *WritOfExecution* – an executive title,
- *LegalCosts* – court fees,
- *ExecutionCosts* – execution costs.

3.2.1.19. Type: executiveObligationTypeEnum

```
<xs:simpleType name="executiveObligationTypeEnum">
  <xs:restriction base="nonEmptyString32">
    <xs:enumeration value="Sentence"/>
    <xs:enumeration value="OrderToPay"/>
    <xs:enumeration value="BankEnforceableTitle"/>
    <xs:enumeration value="NotarialDeed"/>
    <xs:enumeration value="CaseOfTheArbitralTribunal"/>
    <xs:enumeration value="CourtOfArbitrationSettlement"/>
    <xs:enumeration value="AgreementBeforeMediator"/>
  </xs:restriction>
</xs:simpleType>
```

This element is used to specify the type of the unpaid obligation declared by an executive title (negative information). Accepted values:

- *Sentence* – a sentence,
- *OrderToPay* – an order to pay,
- *BankEnforceableTitle* – a bank enforcement (executive) title,
- *NotarialDeed* – a notarial deed,
- *CaseOfTheArbitralTribunal* – an arbitration award,
- *CourtOfArbitrationSettlement* – a settlement concluded before an arbitration court,
- *AgreementBeforeMediator* – settlement concluded before a mediator.

3.2.1.20. Type: currencyEnum

```
<xs:simpleType name="currencyEnum">
  <xs:restriction base="nonEmptyString8">
    <xs:pattern value="[A-Z]{3}">
  </xs:restriction>
</xs:simpleType>
```

This type is used to indicate the currency code being a 3-letter (capital letters) contraction.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

3.2.1.21. Type: countryCodeEnum

```
<xs:simpleType name="countryCodeEnum">
  <xs:restriction base="nonEmptyString3">
    <xs:pattern value="[A-Z]{2}">
  </xs:restriction>
</xs:simpleType>
```

A 2-letter contraction used to indicate the country code.

3.2.1.22. Type: monitorNotificationEnum

```
<xs:simpleType name="monitorNotificationEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Nobody"/>
    <xs:enumeration value="All"/>
    <xs:enumeration value="Main"/>
    <xs:enumeration value="Owner"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type contain information about the e-mail address to which the notification about an event monitored (this is an event about the monitored Tax ID) with a monitoring condition is to be sent. The element can assume only one of the accepted values:

- *Nobody* – an e-mail is not to be sent at all,
- *All* – the notification sent to the e-mail address of the main login and to the e-mail addresses of other logins,
- *Main* – the notification sent only to the e-mail address of the main login,
- *Owner* – the notification sent only to the e-mail address of the login who is the owner of the monitoring condition (monitored Tax ID).

3.2.1.23. Type: monitorEventTypeEnum

```
<xs:simpleType name="monitorEventTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Add"/>
    <xs:enumeration value="Update"/>
    <xs:enumeration value="Remove"/>
    <xs:enumeration value="Search"/>
    <xs:enumeration value="AddPositiveInformation"/>
    <xs:enumeration value="UpdatePositiveInformation"/>
    <xs:enumeration value="RemovePositiveInformation"/>
    <xs:enumeration value="AddQuicklyPaidPositiveInformation"/>
    <xs:enumeration value="UpdateQuicklyPaidPositiveInformation"/>
    <xs:enumeration value="RemoveQuicklyPaidPositiveInformation"/>
  </xs:restriction>
</xs:simpleType>
```

This element contains a string indicating the type of event which occurred during monitoring Tax ID. The element can assume 10 allowed values:

- *Add* – company [the monitored tax identification number (Tax ID)] has been added to the list of debtors (KRD) by one of the providers,
- *Update* – debt amount of monitored company [the monitored tax identification number (Tax ID)] has been increased or decreased, or other details of this company have been edited (e.g. address),
- *Remove* – company has been removed (deleted) from the list of debtors (KRD) by one of the providers,

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

- *Search* – applies to automonitoring only. If someone checks your Tax ID (Tax ID) in KRD, and you have automonitoring enabled in KRD WWW Customer Panel (usually it is enabled by default) – this event will appear,
- *AddPositiveInformation* – adding of positive economic information on obligation paid 8 to 29 days after maturity date (DueDate),
- *UpdatePositiveInformation* – update of positive economic information on obligation paid 8 to 29 days after maturity date (DueDate),
- *RemovePositiveInformation* – removal of positive economic information on obligation paid 8 to 29 days after maturity date (DueDate),
- *AddQuicklyPaidPositiveInformation* – adding of positive economic information on obligation paid by the 7th day after maturity date (DueDate),
- *UpdateQuicklyPaidPositiveInformation* – update of positive economic information on obligation paid by the 7th day after maturity date (DueDate),
- *RemoveQuicklyPaidPositiveInformation* – removal of positive economic information on obligation paid by the 7th day after maturity date (DueDate).

3.2.1.24. Type: documentNumberEnum

```
<xs:simpleType name="documentNumberEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DrivingLicense"/>
    <xs:enumeration value="IdCard"/>
    <xs:enumeration value="Passport"/>
    <xs:enumeration value="SocialInsurance"/>
  </xs:restriction>
</xs:simpleType>
```

Contains a list of predefined list of identity documents:

- *DrivingLicense* – a driving license,
- *IdCard* – an identity card,
- *Passport* – a passport,
- *SocialInsurance* – an insurance.

3.2.1.25. Type: showProviderTypeEnum

```
<xs:simpleType name="showProviderTypeEnum">
  <xs:restriction base="nonEmptyString16">
    <xs:enumeration value="All"/>
    <xs:enumeration value="Ekd"/>
    <xs:enumeration value="Sector"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type contain the information about hiding the creditor's data. The element with a positive case can assume one of the accepted values:

- *All* – all creditor's data are visible,
- *Ekd* – only the creditor's EKD (NACE) / PKD number is visible,
- *Sector* – only the information about the sector of business is visible.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

In the negative case it takes a boolean value (true, false, 1, 0).
By default, provider's data are visible.

3.2.1.26. Type: paidObligationAddReasonEnum

```
<xs:simpleType name="paidObligationAddReasonEnum">
  <xs:restriction base="nonEmptyString32">
    <xs:enumeration value="Bill"/>
    <xs:enumeration value="Contract"/>
    <xs:enumeration value="Invoice"/>
  </xs:restriction>
</xs:simpleType>
```

The elements of this type include the information about the reason (legal title of the obligation) for paying the obligation (positive obligation). The element can assume only one of the accepted values:

- *Bill* – a bill,
- *Contract* – a contract,
- *Invoice* – an invoice.

3.2.2. Definitions of attributes

Attributes are used to define extra characteristics which are not the component of economic information in terms of the Act.

3.2.2.1. Attribute: showProvider

```
<xs:attribute name="showProvider" type="showProviderTypeEnum" default="All"/>
```

It specifies if the economic information KRD report will include all the creditor's data.

3.2.2.2. Attribute: verifyResult

```
<xs:attribute name="verifyResult" type="xs:boolean"/>
```

It is used to specify the response to the order. Value "true" means that the response will contain the data from economic information after fulfilling the given order. E.g. an update order for the case with this option will give in response the updated data of the case.

3.2.2.3. Group of attributes: inputAttributes

```
<xs:attributeGroup name="inputAttributes">
  <xs:attribute name="version" type="versionType" use="required" fixed="3.2"/>
  <xs:attribute name="generator" type="nonEmptyString128" use="optional"/>
  <xs:attribute name="fileName" type="nonEmptyString256" use="optional"/>
  <xs:attribute name="timeStamp" type="xs:dateTime" use="optional"/>
</xs:attributeGroup>
```

A group of attributes used to describe the input file parameters.

3.2.2.4. Group of attributes: idAttributeGroup

```
<xs:attributeGroup name="idAttributeGroup">
  <xs:attribute name="ID" type="nonEmptyString128" use="required"/>
  <xs:attribute name="IDType" type="idTypeEnum" use="optional" default="UserId"/>
</xs:attributeGroup>
```

A group of attributes used to identify the given economic information (positive case, negative case, positive obligation, negative obligation):

- ID – ID of the economic information. This may be the system identifier or identifier generated by the user,
- IDType – required if the system identifier was used.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

3.2.2.5. Group of attributes: nonRequiredIdAttributeGroup

```
<xs:attributeGroup name="nonRequiredIdAttributeGroup">
  <xs:attribute name="ID" type="nonEmptyString128" use="optional"/>
  <xs:attribute name="IDType" type="idTypeEnum" use="optional" default="UserId"/>
</xs:attributeGroup>
```

A group of attributes used to identify the given economic information (positive case, negative case, positive obligation, negative obligation). Element ID is not required.

3.2.2.6. Group of attributes: suspendedAttributes

```
<xs:attributeGroup name="suspendedAttributes">
  <xs:attribute name="suspendedFrom" type="xs:date" use="optional"/>
  <xs:attribute name="suspendedTo" type="xs:date" use="optional"/>
  <xs:attribute name="suspendedComment" type="nonEmptyString128" use="optional"/>
</xs:attributeGroup>
```

This group is used to communicate the notification about the suspension of the negative case. It consists of the following elements:

- *suspendedFrom* – the beginning date,
- *suspendedTo* – the ending date,
- *suspendedComment* – a description of the suspension.

3.2.2.7. Group of attributes: entityAttributes

```
<xs:attributeGroup name="entityAttributes">
  <xs:attribute name="deliveryDate" type="xs:dateTime" use="optional"/>
  <xs:attribute name="modified" type="xs:dateTime" use="optional"/>
</xs:attributeGroup>
```

It includes the information about the date of submitting the last modification of the economic information.

3.2.2.8. Group of attributes: outputIdAttributes

```
<xs:attributeGroup name="outputIdAttributes">
  <xs:attribute name="systemID" type="nonEmptyString128" use="required"/>
  <xs:attribute name="userID" type="nonEmptyString128" use="optional"/>
</xs:attributeGroup>
```

This group is used in return tranches and communicates information about the economic information system identifier (applied automatically after adding information) and own identifier (if it was given during submitting the information).

3.2.2.9. Group of attributes: addIdAttributeGroup

```
<xs:attributeGroup name="addIdAttributeGroup">
  <xs:attribute name="userID" type="nonEmptyString128" use="optional"/>
</xs:attributeGroup>
```

It contains attribute "userID" used to define the own identifier of the economic information. Used only in addition orders.

3.2.2.10. Group of attributes: pagerAttributeGroup

```
<xs:attributeGroup name="pagerAttributeGroup">
  <xs:attribute name="totalCount" type="xs:nonNegativeInteger" use="required"/>
</xs:attributeGroup>
```

It consists of one attribute "totalCount" containing the total number of items on the Client's account (or on the provided login, if he / she has limited rights and cannot see the cases and obligations of other logins).

3.2.3. Groups of elements

Groups of elements define the set of elements which in contrast to the complex types do not create a new type, but only make the concept which we use to further define other types. Creating groups can impose restrictions on the common occurrence of certain elements without creating an additional type (element).

3.2.3.1. Group: registrationGroup

```
<xs:group name="registrationGroup">
  <xs:sequence>
    <xs:element name="RegistrationNumber" type="nonEmptyString128"/>
    <xs:element name="RegistryName" type="nonEmptyString128"/>
  </xs:sequence>
</xs:group>
```

A group of this type is used to provide information about the registration authority. It consists of the elements:

- *RegistrationNumber* – number of registry organ,
- *RegistryName* – name of registry organ.

3.2.3.2. Group: personGroup

```
<xs:group name="personGroup">
  <xs:sequence>
    <xs:element name="FirstName" type="nonEmptyString32"/>
    <xs:element name="SecondName" type="nonEmptyString32" minOccurs="0"/>
    <xs:element name="Surname" type="nonEmptyString64"/>
  </xs:sequence>
</xs:group>
```

A group of this type is used to provide information about related persons with the debtor (partners, board members, proxies, authorized representatives). It consists of the elements:

- *FirstName* – first name,
- *SecondName* – middle name,
- *Surname* – surname.

3.2.3.3. Group: requiredEntrepreneurDataGroup

```
<xs:group name="requiredEntrepreneurDataGroup">
  <xs:sequence>
    <xs:element name="Name" type="nonEmptyString128"/>
    <xs:element name="NonConsumerIdentityNumber"
      type="nonConsumerIdentityNumberType"/>
    <xs:element name="FirstName" type="nonEmptyString32"/>
    <xs:element name="Surname" type="nonEmptyString64"/>
  </xs:sequence>
</xs:group>
```

A group of this type is used to provide information about the person (debtor) leading a one-man business, which are required. It consists of the elements:

- *Name* – identification (name) of the entity. It should contain name and surname of the owner,
- *NonConsumerIdentityNumber* – identification number of the entity *nonConsumerIdentityNumberType* type,
- *FirstName* – owner name,
- *Surname* – owner surname.

3.2.3.4. Group: nonRequiredEntrepreneurDataGroup

```
<xs:group name="nonRequiredEntrepreneurDataGroup">
  <xs:sequence>
    <xs:element name="SecondName" type="nonEmptyString32" minOccurs="0"/>
    <xs:element name="MailingAddress" type="addressType" minOccurs="0"/>
    <xs:element name="ResidenceAddress" type="addressType" minOccurs="0"/>
    <xs:element name="SeatAddress" type="addressType" minOccurs="0"/>
    <xs:element name="ConsumerIdentityNumber" type="consumerIdentityNumberType"
minOccurs="0"/>
    <xs:element name="DocumentNumber" type="documentNumberType" minOccurs="0"/>
    <xs:group ref="registrationGroup" minOccurs="0"/>
    <xs:element name="Ekd" type="ekdType" minOccurs="0"/>
    <xs:element name="Regon" type="regonType" minOccurs="0"/>
    <xs:element name="Representatives" type="representativesType" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

A group of this type is used to provide information about the person (debtor) leading a one-man business, which are not required. It consists of the elements:

- *SecondName* – owner second name,
- *MailingAddress* – mailing address,
- *ResidenceAddress* – residence address,
- *SeatAddress* – the address of pursuing economic activity,
- *ConsumerIdentityNumber* – identification number,
- *DocumentNumber* – document identifying the owner,
- *registrationGroup* – group of elements responsible for the registration authority,
- *Ekd* – EKD (NACE) / PKD,
- *Regon* – REGON,
- *Representatives* – persons representing the debtor (authorized representatives).

3.2.3.5. Group: requiredConsumerDataGroup

```
<xs:group name="requiredConsumerDataGroup">
  <xs:sequence>
    <xs:element name="FirstName" type="nonEmptyString32"/>
    <xs:element name="Surname" type="nonEmptyString64"/>
    <xs:element name="IdentityNumber" type="consumerIdentityNumberType"/>
  </xs:sequence>
</xs:group>
```

A group of this type is used to provide information about the consumer (debtor) which are required. It consists of the elements:

- *FirstName* – name,
- *Surname* – surname,
- *IdentityNumber* – identification number (PESEL for Polish citizens).

3.2.3.6. Group: nonRequiredConsumerDataGroup

```
<xs:group name="nonRequiredConsumerDataGroup">
  <xs:sequence>
    <xs:element name="SecondName" type="nonEmptyString32" minOccurs="0"/>
    <xs:element name="Address" type="addressType" minOccurs="0"/>
    <xs:element name="MailingAddress" type="addressType" minOccurs="0"/>
    <xs:element name="DocumentNumber" type="documentNumberType" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

```
</xs:sequence>
</xs:group>
```

A group of this type is used to provide information about the consumer (debtor), which are not required. It consists of the elements:

- *SecondName* – second name,
- *Address* – residence address,
- *MailingAddress* – mailing address,
- *DocumentNumber* – document identifying the person.

3.2.3.7. Group: obligationBaseGroup

```
<xs:group name="obligationBaseGroup">
  <xs:sequence>
    <xs:element name="PaymentDate" type="xs:date"/>
    <xs:element name="Debt" type="moneyType" minOccurs="0"/>
    <xs:element name="Arrears" type="moneyType"/>
    <xs:element name="Proceedings" type="nonEmptyString1024" minOccurs="0"/>
    <xs:choice>
      <xs:element name="NoObjections" type="xs:boolean" fixed="true"/>
      <xs:element name="Objections" type="nonEmptyString1024"/>
    </xs:choice>
    <xs:element name="Note" type="nonEmptyString1024" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

Group consists of the elements identifying negative obligation:

- *PaymentDate* – contains the date by which the obligation should be repaid (payment date),
- *Debt* – including the amount of the obligation,
- *Arrears* – including the amount of the arrears,
- *Proceedings* – including information about current proceedings regarding the obligation (it can be encoded in the form of a digit corresponding to the values from the dictionary of proceedings, details [here](#)),
- *NoObjections* or *Objections* (it can be encoded in the form of a digit corresponding to the values from the dictionary of objections, details [here](#)) – determining whether the debtor disputes the obligation,
- *Note* – an optional note [contains additional information about the obligation (not disclosed in KRD reports)].

3.2.3.8. Group: documentSeriesNumberGroup

```
<xs:group name="documentSeriesNumberGroup">
  <xs:sequence>
    <xs:element name="Series" type="nonEmptyString128" minOccurs="0"/>
    <xs:element name="Number" type="nonEmptyString128"/>
  </xs:sequence>
</xs:group>
```

Group consists of the elements identifying debtor's identity document:

- *Series* – contains the series number of the document,
- *Number* – contains document number.

3.2.3.9. Group: monitorConditionGroup

```
<xs:group name="monitorConditionGroup">
  <xs:sequence>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

```
<xs:element name="Number" type="nonConsumerIdentityNumberType"/>
<xs:element name="DateFrom" type="xs:date"/>
<xs:element name="DateTo" type="xs:date" minOccurs="0"/>
<xs:element name="Description" type="nonEmptyString256" minOccurs="0"/>
<xs:element name="MonitorNotificationType" type="monitorNotificationEnum"/>
</xs:sequence>
</xs:group>
```

Group consists of the elements identifying condition monitoring (applies to Tax ID monitoring). The group consists of the following elements:

- *Number* – contains monitored Tax ID,
- *DateFrom* – meaning from when to start monitoring,
- *DateTo* – meaning from when the monitoring should start. If the element is omitted, this means that the monitoring is to operate until further notice,
- *Description* – contains additional description of condition monitoring (appears on the list of monitoring events: [description 1](#), [description 2](#)),
- *MonitorNotificationType* – of the *monitorNotificationEnum* type, contains information to whom should be sent the e-mail notifications (details [here](#)).

3.2.3.10. Group: requiredLegalPersonDataGroup

```
<xs:group name="requiredLegalPersonDataGroup">
  <xs:sequence>
    <xs:element name="Name" type="nonEmptyString128"/>
    <xs:element name="IdentityNumber" type="nonConsumerIdentityNumberType"/>
    <xs:element name="SeatAddress" type="addressType"/>
  </xs:sequence>
</xs:group>
```

A group of this type is used to provide information about the companies and institutions (debtors) which are required. It consists of the elements:

- *Name* – company name,
- *IdentityNumber* – identification number,
- *SeatAddress* – seat address.

3.2.3.11. Group: nonRequiredLegalPersonDataGroup

```
<xs:group name="nonRequiredLegalPersonDataGroup">
  <xs:sequence>
    <xs:group ref="registrationGroup" minOccurs="0"/>
    <xs:element name="Regon" type="regonType" minOccurs="0"/>
    <xs:element name="Persons" type="personsType" minOccurs="0"/>
    <xs:element name="Representatives" type="representativesType" minOccurs="0"/>
    <xs:element name="Ekd" type="ekdType" minOccurs="0"/>
  </xs:sequence>
</xs:group>
```

A group of this type is used to provide information about the companies and institutions (debtors) which are not required. It consists of the elements:

- *registrationGroup* – a group of elements responsible for the registration authority,
- *Regon* – REGON,
- *Persons* – persons related to the debtor (partners, board members, proxies),

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

- *Representatives* – persons representing the debtor (authorized representatives),
- *Ekd* – EKD / PKD.

3.2.3.12. Group: nonRequiredIdAttributeGroup

```
<xs:attributeGroup name="nonRequiredIdAttributeGroup">
  <xs:attribute name="ID" type="nonEmptyString128" use="optional"/>
  <xs:attribute name="IDType" type="idTypeEnum" use="optional" default="UserId"/>
</xs:attributeGroup>
```

Group of this type is used to provide unrequired information about the identifier. It consists of the elements:

- *ID* – ID value, which is a non-empty string with a maximum length of 128 characters,
- *IDType* – identifier type in accordance with the values from *idTypeEnum*; optional element; the default is *UserId*.

3.2.4. Complex types

The complex types define the data structures composed of more than one element. They indicate the required multiplexing and the type or format of the elements.

3.2.4.1. Type: orderBaseType

```
<xs:complexType name="orderBaseType">
  <xs:attribute name="ID" type="nonEmptyString128" use="required"/>
</xs:complexType>
```

The basic type for all tasks ordered by the Client in the bureau. The elements of this type contain an attribute *ID* defined as a string of maximum length 128. The value of the element is specified by the Client, and it is very important for the identifier not to be repeated in the whole import file. It is allowed to use the same identifiers in the following import files.

The identifier is then returned in reply, and it is used to link the ordered tasks with the bureau's reply.

3.2.4.2. Type: moneyType

```
<xs:complexType name="moneyType">
  <xs:simpleContent>
    <xs:extension base="decimalType">
      <xs:attribute name="currency" type="currencyEnum" default="PLN"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

This type is used to define the elements containing information about amounts of money. It has an extra attribute *currency*, defining the used currency.

3.2.4.3. Type: customNumberType

```
<xs:complexType name="customNumberType">
  <xs:simpleContent>
    <xs:extension base="nonEmptyString32">
      <xs:attribute name="type" type="nonEmptyString64" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

It is used for submitting one's own identification number by entities or persons without a Polish identifier (PESEL, Tax ID or EU Tax ID).

3.2.4.4. Type: moneyFilterType

```
<xs:complexType name="moneyFilter">
  <xs:sequence>
    <xs:element name="AmountFrom" type="decimalType" minOccurs="0"/>
    <xs:element name="AmountTo" type="decimalType" minOccurs="0"/>
    <xs:element name="Currency" type="currencyEnum" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

An element filtering the returned information after the amount of money. It consists of 3 elements:

- *AmountFrom* – minimum amount,
- *AmountTo* – maximum amount,
- *Currency* – currency.

3.2.4.5. Type: dateFilterType

```
<xs:complexType name="dateFilter">
  <xs:sequence>
    <xs:element name="From" type="xs:date" minOccurs="0"/>
    <xs:element name="To" type="xs:date" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

An element filtering the returned information after the date. It consists of 2 elements:

- *From* – the earliest date,
- *To* – the latest date.

3.2.4.6. Type: caseFilterType

```
<xs:complexType name="caseFilterType">
  <xs:choice>
    <xs:element name="DebtorIdentityNumber" type="nonEmptyString64" minOccurs="0"/>
    <xs:element name="Arrears" type="moneyFilterType" minOccurs="0"/>
    <xs:element name="DeliveryDate" type="dateFilterType" minOccurs="0"/>
    <xs:element name="LoginName" type="nonEmptyString128" minOccurs="0"/>
  </xs:choice>
</xs:complexType>
```

An element filtering the returned negative cases. It is possible to submit one of the following elements:

- *DebtorIdentityNumber* – debtor's identifier (eg. PESEL / Tax ID),
- *Arrears* – the amount of the arrears,
- *DeliveryDate* – the date of delivery,
- *LoginName* – the login of the user who is the owner of the given case.

3.2.4.7. Type: monitorConditionFilterType

```
<xs:complexType name="monitorConditionFilterType">
  <xs:sequence>
    <xs:element name="Active" type="xs:boolean" default="true" minOccurs="0"/>
    <xs:element name="Inactive" type="xs:boolean" minOccurs="0"/>
    <xs:element name="LoginName" type="nonEmptyString128" minOccurs="0"/>
    <xs:element name="Number" type="nonEmptyString64" minOccurs="0"/>
    <xs:element name="Date" type="dateFilterType" minOccurs="0"/>
  </xs:sequence>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

</xs:complexType>

An element of this type is used to define a filter for the list of the applied monitors (monitored Tax ID's). The following values are accepted:

- *Active* – determines whether the list should contain the active monitors,
- *Inactive* – determines whether the list should contain the inactive monitors (expired or future),
- *LoginName* – the login of the user who applied the given monitor,
- *Number* – identity number of monitored entity (Tax ID),
- *Date* – date range operation condition monitoring.

3.2.4.8. Type: getMonitorEventsFilterType

```
<xs:complexType name="getMonitorEventsFilterType">
  <xs:sequence>
    <xs:element name="Date" type="dateFilterType" minOccurs="0"/>
    <xs:element name="EventType" type="monitorEventTypeEnum" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="LoginName" type="nonEmptyString128" minOccurs="0"/>
    <xs:element name="Number" type="nonEmptyString64" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to define a filter for the list of monitoring events (applies to monitored Tax ID's). The following values are accepted:

- *Date* – date range that interests us,
- *EventType* – determines the type of event. The omission of this element is synonymous with the choice of all events,
- *LoginName* – the login of the user who applied the given monitor,
- *Number* – identity number of monitored entity (Tax ID).

3.2.4.9. Type: pagerType

```
<xs:complexType name="pagerType">
  <xs:sequence>
    <xs:element name="StartIndex" type="xs:nonNegativeInteger" default="0"/>
    <xs:element name="Count" type="intMaxType" default="100" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Contains information on the range of the returned data. It is possible to define the initial value of the range and the number of returned records. By default, the first 100 records are returned. When you want to obtain all the records, the beginning of the range (*StartIndex*) should be "0" and the number of records (*Count*) should be "max" value. Such values of the range of data are recommended only for lists containing small numbers of records.

3.2.4.10. Type: getInformationsBaseType

```
<xs:complexType name="getInformationsBaseType">
  <xs:sequence>
    <xs:element name="Pager" type="pagerType"/>
  </xs:sequence>
</xs:complexType>
```

This type is used to define orders for obtaining lists of economic information.

3.2.4.11. Type: getMonitorConditionsType

```
<xs:complexType name="getMonitorConditionsType">
  <xs:complexContent>
    <xs:extension base="getInformationsBaseType">
      <xs:sequence>
        <xs:element name="Filter" type="monitorConditionFilterType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to obtain a list of the applied monitors (applies to monitored Tax ID's).

3.2.4.12. Type: getMonitorEventsType

```
<xs:complexType name="getMonitorEventsType">
  <xs:complexContent>
    <xs:extension base="getInformationsBaseType">
      <xs:sequence>
        <xs:element name="Filter" type="getMonitorEventsFilterType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to obtain a list of the monitoring events (applies to monitored Tax ID's).

3.2.4.13. Type: getCasesType

```
<xs:complexType name="getCasesType">
  <xs:complexContent>
    <xs:extension base="getInformationsBaseType">
      <xs:sequence>
        <xs:element name="Filter" type="caseFilterType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

A type used to receive the list of negative cases. It allows for defining range and filter.

3.2.4.14. Type: getPaidObligationCasesType

```
<xs:complexType name="getPaidObligationCasesType">
  <xs:complexContent>
    <xs:extension base="getInformationsBaseType"/>
  </xs:complexContent>
</xs:complexType>
```

A type used to receive the list of positive cases.

3.2.4.15. Type: getInformationsType

```
<xs:complexType name="getInformationsType">
  <xs:choice>
    <xs:element name="Cases" type="getCasesType"/>
    <xs:element name="PaidObligationCases" type="getPaidObligationCasesType"/>
  </xs:choice>
</xs:complexType>
```

This element defines choices for the order of economic information (from own account).

3.2.4.16. Type: nonConsumerIdentityNumberType

```
<xs:complexType name="nonConsumerIdentityNumberType">
  <xs:choice>
    <xs:element name="TaxId" type="nonEmptyString128"/>
    <xs:element name="Custom" type="customNumberType"/>
  </xs:choice>
</xs:complexType>
```

Elements of this type include information about the type of the identifier not being a consumer (applies to the negative / positive information, monitoring). Accepted values:

- *TaxId* – tax identification number (Tax ID),
- *Custom* – any identifier.

3.2.4.17. Type: consumerIdentityNumberType

```
<xs:complexType name="consumerIdentityNumberType">
  <xs:choice>
    <xs:element name="Pesel" type="peselType"/>
    <xs:element name="Custom" type="customNumberType"/>
  </xs:choice>
</xs:complexType>
```

Elements of this type include information about the type of the identifier of a consumer (applies to the negative / positive information). Accepted values:

- *Pesel* – PESEL number,
- *Custom* – any identifier.

3.2.4.18. Type: documentNumberType

```
<xs:complexType name="documentNumberType">
  <xs:choice>
    <xs:element name="Number">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="nonEmptyString128">
            <xs:attribute name="type" type="documentNumberEnum" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Custom" type="customNumberType"/>
  </xs:choice>
</xs:complexType>
```

An element of this type contains the information about the number of the debtor's identification document. Accepted values:

- *Number* – series and number of the document containing the attribute with document *type* named type of the *documentNumberEnum* type,
- *Custom* – a different number of the *customNumberType* type.

3.2.4.19. Type: propertyType

```
<xs:complexType name="propertyType">
  <xs:simpleContent>
    <xs:extension base="nonEmptyString128">
      <xs:attribute name="name" type="nonEmptyString64" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

This element is used to pass the dictionary's own properties.

3.2.4.20. Type: `propertiesType`

```
<xs:complexType name="propertiesType">
  <xs:sequence>
    <xs:element name="Property" type="propertyType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

It contains a collection of properties.

3.2.4.21. Type: `notificationAddressType`

```
<xs:complexType name="notificationAddressType">
  <xs:complexContent>
    <xs:extension base="addressType">
      <xs:attribute name="type" type="addressTypeEnum" default="Regular"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Elements of this type include information about the address for sending the notification about adding the debtor to the KRD (for information – currently we do not send letters abroad). They are an extension of the `addressType` type with an extra attribute `type` specifying the mailing address type.

3.2.4.22. Type: `addressType`

```
<xs:complexType name="addressType">
  <xs:sequence>
    <xs:element name="Line" type="nonEmptyString128" minOccurs="2" maxOccurs="4"/>
  </xs:sequence>
  <xs:attribute name="countryCode" type="countryCodeEnum" use="optional" default="PL"/>
</xs:complexType>
```

An element of this type contains the address information. It allows 2-4 *line* elements specifying the address. This element contains information about the country code.

3.2.4.23. Type: `personsType`

```
<xs:complexType name="personsType">
  <xs:sequence>
    <xs:element name="Person" type="personType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to inform about the persons related to the debtor (partners, board members, proxies). This type includes a list of *personType* elements.

3.2.4.24. Type: `representativesType`

```
<xs:complexType name="representativesType">
  <xs:sequence>
    <xs:element name="Representative" type="representativeType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to inform about the persons representing the debtor (authorized representatives). This type includes a list of *representativeType* elements.

3.2.4.25. Type: `personType`

```
<xs:complexType name="personType">
  <xs:sequence>
    <xs:group ref="personGroup"/>
  </xs:sequence>
</xs:complexType>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

```
<xs:element name="Role" type="personRoleEnum"/>
<xs:element name="OtherRole" type="nonEmptyString128"/>
</xs:choice>
<xs:element name="IdentityNumber" type="consumerIdentityNumberType"/>
</xs:sequence>
</xs:complexType>
```

An element of this type is used to inform about the persons related to the debtor (partners, board members, proxies). This type contains the basic data of a person (*personGroup* group), role of the person and the identification number.

3.2.4.26. Type: representativeType

```
<xs:complexType name="representativeType">
  <xs:sequence>
    <xs:group ref="personGroup"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to inform about the persons representing the debtor (authorised representatives). This type includes a list of *personGroup* groups.

3.2.4.27. Type: entrepreneurType

```
<xs:complexType name="entrepreneurType">
  <xs:complexContent>
    <xs:extension base="entrepreneurMinType">
      <xs:group ref="nonRequiredEntrepreneurDataGroup" minOccurs="0"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to present the data of a person running a one-man business (debtor). It contains the *entrepreneurMinType* type and the optional *nonRequiredEntrepreneurDataGroup* group.

3.2.4.28. Type: entrepreneurMinType

```
<xs:complexType name="entrepreneurMinType">
  <xs:sequence>
    <xs:group ref="requiredEntrepreneurDataGroup"/>
  </xs:sequence>
  <xs:attribute name="isPolishCitizen" type="xs:boolean" use="optional" default="true"/>
</xs:complexType>
```

An element of this type is used to present the required data of a person running a one-man business [related with the debtor, contractor – payer, the person downloading the KRD report (requester)]. It consists of the *requiredEntrepreneurDataGroup* group and the optional *isPolishCitizen* attribute specifying the citizenship.

3.2.4.29. Type: consumerType

```
<xs:complexType name="consumerType">
  <xs:complexContent>
    <xs:extension base="consumerMinType">
      <xs:group ref="nonRequiredConsumerDataGroup" minOccurs="0"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to present the consumer's data (debtor). It contains the *consumerMinType* type and the optional group *nonRequiredConsumerDataGroup*.

3.2.4.30. Type: consumerMinType

```
<xs:complexType name="consumerMinType">
  <xs:sequence>
    <xs:group ref="requiredConsumerDataGroup"/>
  </xs:sequence>
  <xs:attribute name="isPolishCitizen" type="xs:boolean" use="optional" default="true"/>
</xs:complexType>
```

An element of this type is used to present the required consumer's data [related with the debtor, contractor – payer, the person downloading the KRD report (requester)]. It consists of the *requiredConsumerDataGroup* group and the optional *isPolishCitizen* attribute specifying that the person has the Polish citizenship.

3.2.4.31. Type: legalPersonType

```
<xs:complexType name="legalPersonType">
  <xs:complexContent>
    <xs:extension base="legalPersonMinType">
      <xs:sequence>
        <xs:group ref="nonRequiredLegalPersonDataGroup" minOccurs="0"/>
        <xs:element name="Stakeholders" type="stakeholdersType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to present the entrepreneurs' and institutions' data (debtors). It contains the *legalPersonMinType* type and optionally the *nonRequiredLegalPersonDataGroup* group.

3.2.4.32. Type: legalPersonMinType

```
<xs:complexType name="legalPersonMinType">
  <xs:sequence>
    <xs:group ref="requiredLegalPersonDataGroup"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to present the entrepreneurs' and institutions' data [related with the debtor, contractor – payer, the person downloading the KRD report (requester)]. It consists of the *requiredLegalPersonDataGroup* group.

3.2.4.33. Type: subjectType

```
<xs:complexType name="subjectType">
  <xs:choice>
    <xs:element name="Consumer" type="paidObligationConsumerType"/>
    <xs:element name="LegalPerson" type="legalPersonMinType"/>
    <xs:element name="Entrepreneur" type="paidObligationEntrepreneurType"/>
  </xs:choice>
</xs:complexType>
```

This element defines the type and the data of contractor – payer (positive information).

3.2.4.34. Type: debtorType

```
<xs:complexType name="debtorType">
  <xs:choice>
    <xs:element name="Consumer" type="consumerType"/>
    <xs:element name="LegalPerson" type="legalPersonType"/>
    <xs:element name="Entrepreneur" type="entrepreneurType"/>
  </xs:choice>
</xs:complexType>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

This element defines the type and the data of a debtor (negative information).

3.2.4.35. Type: stakeholderType

```
<xs:complexType name="stakeholderType">
  <xs:sequence>
    <xs:element name="Name" type="nonEmptyString256" minOccurs="0"/>
    <xs:element name="NonConsumerIdentityNumber" type="nonConsumerIdentityNumberType"
minOccurs="0"/>
    <xs:element name="FirstName" type="nonEmptyString32" minOccurs="0"/>
    <xs:element name="SecondName" type="nonEmptyString32" minOccurs="0"/>
    <xs:element name="Surname" type="nonEmptyString64" minOccurs="0"/>
    <xs:element name="ConsumerIdentityNumber" type="consumerIdentityNumberType"
minOccurs="0"/>
    <xs:element name="DocumentNumber" type="documentNumberType" minOccurs="0"/>
    <xs:group ref="registrationGroup" minOccurs="0"/>
    <xs:element name="Ekd" type="ekdType" minOccurs="0"/>
    <xs:element name="Regon" type="regonType" minOccurs="0"/>
    <xs:element name="MailingAddress" type="addressType" minOccurs="0"/>
    <xs:element name="Address" type="addressType" minOccurs="0"/>
    <xs:element name="SeatAddress" type="addressType" minOccurs="0"/>
    <xs:element name="Persons" type="personsType" minOccurs="0"/>
    <xs:element name="Representatives" type="representativesType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to store the data of partners, board members, proxies, authorized representatives, etc. of the debtor. It consists of the following elements:

- *Name* – name,
- *NonConsumerIdentityNumber* – an identifier of a non-consumer,
- *FirstName* – first name,
- *SecondName* – middle name,
- *Surname* – surname,
- *ConsumerIdentityNumber* – the consumer's identity number,
- *DocumentNumber* – identity document,
- *registrationGroup* – a group of elements responsible for the registering body data,
- *Ekd* – EKD (NACE) / PKD number,
- *Regon* – Polish National Business Registry Number (REGON),
- *MailingAddress* – mailing address,
- *Address* – residence address,
- *SeatAddress* – seat address / the address of pursuing economic activity,
- *Persons* – persons related to the debtor (partners, board members, proxies),
- *Representatives* – persons representing the debtor (authorized representatives).

3.2.4.36. Type: stakeholdersType

```
<xs:complexType name="stakeholdersType">
  <xs:sequence>
    <xs:element name="Stakeholder" type="stakeholderType" maxOccurs="unbounded"/>
  </xs:sequence>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

</xs:complexType>

An element of this type is used to store the data of partners, board members, proxies, authorized representatives, etc. of the debtor. It can contain a list of *stakeholderType* elements.

3.2.4.37. Type: obligationType

```
<xs:complexType name="obligationType">
  <xs:sequence>
    <xs:choice>
      <xs:sequence>
        <xs:element name="Title" type="nonEmptyString128"/>
        <xs:sequence minOccurs="0">
          <xs:choice>
            <xs:element name="Type" type="obligationTypeEnum"/>
            <xs:element name="CustomType" type="nonEmptyString128"/>
          </xs:choice>
        </xs:sequence>
      </xs:sequence>
      <xs:element name="Type" type="obligationTypeEnum"/>
      <xs:element name="CustomType" type="nonEmptyString128"/>
    </xs:choice>
    <xs:element name="CallSent" type="xs:date"/>
    <xs:group ref="obligationBaseGroup"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to store data about the ordinary obligations (negative informations). It consists of the following elements:

- *Title* – Legal title of the obligation,
- *Type* – type of obligation,
- *CustomType* – another type of obligation,
- *CallSent* – date of sending or delivering a call for payment to the debtor with the warning to add the obligation to the KRD,
- *obligationBaseGroup* – a group of elements describing the obligation.

3.2.4.38. Type: alimonyObligationType

```
<xs:complexType name="alimonyObligationType">
  <xs:sequence>
    <xs:element name="Date" type="xs:date" minOccurs="0"/>
    <xs:element name="Signature" type="nonEmptyString128" minOccurs="0"/>
    <xs:element name="DecidingAuthority" type="nonEmptyString128" minOccurs="0"/>
    <xs:element name="CallSent" type="xs:date" minOccurs="0"/>
    <xs:group ref="obligationBaseGroup"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to store data about the alimony obligations (negative informations). It consists of the following elements:

- *Date* – date of establishing the executive title,
- *Signature* – signature,
- *DecidingAuthority* – deciding authority,
- *CallSent* – date of sending or delivering to the debtor a warning about the intention to add the obligation to the KRD,
- *obligationBaseGroup* – a group of elements describing the obligation.

3.2.4.39. Type: executiveObligationType

```
<xs:complexType name="executiveObligationType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Type" type="executiveObligationTypeEnum"/>
      <xs:element name="CustomType" type="nonEmptyString128"/>
    </xs:choice>
    <xs:element name="Date" type="xs:date"/>
    <xs:element name="Signature" type="nonEmptyString128"/>
    <xs:element name="DecidingAuthority" type="nonEmptyString128"/>
    <xs:element name="CallSent" type="xs:date"/>
    <xs:group ref="obligationBaseGroup"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to store data about the obligations declared by an executive title (negative informations). It consists of the following elements:

- *Type* – type of obligation,
- *CustomType* – another type of obligation,
- *Date* – date of establishing the executive title,
- *Signature* – signature,
- *DecidingAuthority* – deciding authority,
- *CallSent* – date of sending to the debtor a warning about the intention to add the obligation to the KRD,
- *obligationBaseGroup* – a group of elements describing the obligation.

3.2.4.40. Type: addCaseType

```
<xs:complexType name="addCaseType">
  <xs:sequence>
    <xs:element name="Debtor" type="debtorType"/>
    <xs:choice>
      <xs:sequence>
        <xs:element name="Obligations" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Obligation" type="addObligationType"
maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="ExecutiveObligations" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ExecutiveObligation" type="addExecutiveObligationType"
maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:element name="AlimonyObligations">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="AlimonyObligation" type="addAlimonyObligationType"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
    <xs:element name="Note" type="nonEmptyString1024" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```



```
<xs:element name="Notify" type="notifyDebtorType" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="showProvider" type="xs:boolean" use="optional" default="true"/>
</xs:complexType>
```

An element of this type defines all data of the debtor's case, i.e. debtor's data, information about the obligation, mailing address for the delivery of the notification about adding to the KRD (for information – currently we do not send letters abroad) and optionally the identifier of the case and a note.

This element can contain the *showProvider*, attribute specifying if and what data of an entity disclosing the economic information (provider) will be visible here. By default, all data are visible.

3.2.4.41. Type: monitorConditionType

```
<xs:complexType name="monitorConditionType">
  <xs:group ref="monitorConditionGroup"/>
</xs:complexType>
```

An element of this type is used to submit a collection of monitoring conditions (applies to monitored Tax ID's). It consists of elements belonging to the *monitorConditionGroup* group.

3.2.4.42. Type: updateCaseType

```
<xs:complexType name="updateCaseType">
  <xs:sequence>
    <xs:element name="Debtor" type="debtorType"/>
    <xs:choice>
      <xs:sequence>
        <xs:element name="Obligations" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Obligation" type="updateObligationType"
maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="ExecutiveObligations" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ExecutiveObligation" type="updateExecutiveObligationType"
maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="AlimonyObligations" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="AlimonyObligation" type="updateAlimonyObligationType"
minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    <xs:element name="Note" type="nonEmptyString1024" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="showProvider" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="obligationRemoveReason" type="removeReasonType" use="optional"
default="Paid"/>
</xs:complexType>
```

An element of this type contains a set of data necessary for debtor's case (negative informations) update. It contains the following elements:

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

- *Debtor* – debtor's data,
- *Obligation* or *ExecutiveObligation*, or *AlimonyObligation* – a collection of obligations of a given type,
- *Note* – an optional note [contains additional information about the obligation (not disclosed in KRD reports)].

Additionally, it is required to submit attributes allowing to identify the case in the system, defining if the creditor's data are published and presenting the reason for deleting obligations from the case (only when some obligations are deleted during editing).

3.2.4.43. Type: addObligationType

```
<xs:complexType name="addObligationType">
  <xs:complexContent>
    <xs:extension base="obligationType">
      <xs:attributeGroup ref="addIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to add ordinary obligations (negative informations). It is an extension of the *obligationType* type with attributes from the *addIdAttributeGroup* group.

3.2.4.44. Type: updateObligationType

```
<xs:complexType name="updateObligationType">
  <xs:complexContent>
    <xs:extension base="obligationType">
      <xs:attributeGroup ref="nonRequiredIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to update ordinary obligations (negative informations). It is an extension of the *obligationType* type with attributes from the *nonRequiredIdAttributeGroup* group.

3.2.4.45. Type: addAlimonyObligationType

```
<xs:complexType name="addAlimonyObligationType">
  <xs:complexContent>
    <xs:extension base="alimonyObligationType">
      <xs:attributeGroup ref="addIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to add alimony obligations (negative informations). It is an extension of the *alimonyObligationType* type with attributes from the *addIdAttributeGroup* group.

3.2.4.46. Type: updateAlimonyObligationType

```
<xs:complexType name="updateAlimonyObligationType">
  <xs:complexContent>
    <xs:extension base="alimonyObligationType">
      <xs:attributeGroup ref="nonRequiredIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to update alimony obligations (negative informations). It is an extension of the *alimonyObligationType* type with attributes from the *nonRequiredIdAttributeGroup* group.

3.2.4.47. Type: addExecutiveObligationType

```
<xs:complexType name="addExecutiveObligationType">
  <xs:complexContent>
    <xs:extension base="executiveObligationType">
      <xs:attributeGroup ref="addIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to add obligations declared with an executive title (negative informations). It is an extension of the *executiveObligationType* type with attributes from the *addIdAttributeGroup* group.

3.2.4.48. Type: updateExecutiveObligationType

```
<xs:complexType name="updateExecutiveObligationType">
  <xs:complexContent>
    <xs:extension base="executiveObligationType">
      <xs:attributeGroup ref="nonRequiredIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An element of this type is used to update obligations declared with an executive title (negative informations). It is an extension of the *executiveObligationType* type with attributes from the *nonRequiredIdAttributeGroup* group.

3.2.4.49. Type: notifyDebtorType

```
<xs:complexType name="notifyDebtorType">
  <xs:sequence>
    <xs:element name="Address" type="notificationAddressType" minOccurs="0"/>
    <xs:element name="Email" type="emailType" minOccurs="0"/>
    <xs:element name="ContactDetails" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="FullName" type="nonEmptyString64"
minOccurs="0"/>
          <xs:element name="Role" type="nonEmptyString64"
minOccurs="0"/>
          <xs:element name="PhoneNumber" type="nonEmptyString128"
minOccurs="0"/>
          <xs:element name="Email" type="nonEmptyString128"
minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="CustomProperties" type="propertiesType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="notificationLanguage" type="xs:string" use="optional"/>
</xs:complexType>
```

This element defines the range of data needed to send the notification to the debtor on adding to KRD (for information – currently we do not send letters abroad). It contains information about shipping postal address, e-mail address to send a copy of the notification [to the provider, the system will send an email when it finds the indicated email address in the Client's account (on one of the logins)], contact information, which will appear in print (notification) and language notice.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

In addition, to the *CustomProperties* field, it is possible to define additional properties, such as bank account details to which the debtor should repay the debt (arrears). These details will be included in the notification.

To add an account number to the notification, complete the *CustomProperties* field as shown below:

```
<CustomProperties>
  <Property name="BANKACCOUNTNUMBER">PL81124063511880473391550000</Property>
  <Property name="BANKACCOUNTOWNERNAME">Account owner name</Property>
  <Property name="BANKACCOUNTOWNERADDRESSLINE1">First line of the address of the
account owner</Property>
  <Property name="BANKACCOUNTOWNERADDRESSLINE2">Second line of the address of the
account owner</Property>
  <Property name="BANKACCOUNTOWNERADDRESSLINE3">Third line of the address of the
account owner</Property>
  <Property name="BANKACCOUNTOWNERADDRESSLINE4">Fourth line of the address of the
account owner</Property>
</CustomProperties>
```

3.2.4.50. Type: *entityManagementType*

```
<xs:complexType name="entityManagementType">
  <xs:attributeGroup ref="idAttributeGroup"/>
  <xs:attribute name="informationType" type="informationTypeEnum" use="required"/>
</xs:complexType>
```

A basic type used to perform operations on economic information.

3.2.4.51. Type: *entityManagementWithVerificationType*

```
<xs:complexType name="entityManagementWithVerificationType">
  <xs:complexContent>
    <xs:extension base="entityManagementType">
      <xs:attribute ref="verifyResult" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

A basic type used to perform operations on economic information with the possibility to download verification data.

3.2.4.52. Type: *removeInformationType*

```
<xs:complexType name="removeInformationType">
  <xs:complexContent>
    <xs:extension base="entityManagementWithVerificationType">
      <xs:attribute name="removeReason" type="removeReasonType" use="optional"
default="Paid"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Elements of this type are used in economic information deleting orders. It extends the basic type with an attribute allowing to define the reason for deleting a given information (refers to unpaid negative obligations).

3.2.4.53. Type: *addPaidObligationCaseType*

```
<xs:complexType name="addPaidObligationCaseType">
  <xs:sequence>
    <xs:element name="Debtor" type="paidObligationDebtorType"/>
    <xs:element name="Obligations" type="addPaidObligationsType"/>
  </xs:sequence>
  <xs:attribute ref="showProvider" use="optional" default="All"/>
</xs:complexType>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

Elements of this type are used in adding new paid (positive) obligations cases. It consists of the following elements:

- *Debtor* – containing the contractor’s – payer’s data, the *paidObligationDebtorType* type,
- *Obligations* – containing the paid (positive) obligations data, the *addPaidObligationsType* type.

This element can contain the *showProvider*, attribute specifying if and what data of an entity disclosing the economic information (provider) will be visible here. By default, all data are visible.

3.2.4.54. Type: *paidObligationType*

```
<xs:complexType name="paidObligationType">
  <xs:sequence>
    <xs:choice minOccurs="0">
      <xs:element name="Reason" type="paidObligationAddReasonEnum"/>
      <xs:element name="OtherReason" type="nonEmptyString32"/>
    </xs:choice>
    <xs:element name="PaidDebt" type="moneyType"/>
    <xs:element name="TotalDebt" type="moneyType"/>
    <xs:element name="DueDate" type="xs:date"/>
    <xs:element name="PaidDate" type="xs:date"/>
    <xs:element name="IsSumOfInstallments" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type contains the basic information referring to economic information about a paid obligation (positive information). It consists of the following elements:

- *Reason / OtherReason* – containing the reason for paying the obligation (legal title of the obligation). Depending on the needs, it is possible to use the *Reason* node including one of the predefined reasons of the *paidObligationAddReasonEnum* type or in the *OtherReason* node specify the reason as a string of not more than 32 characters,
- *PaidDebt* – specifying the amount and currency repaid, the *moneyType* type,
- *TotalDebt* – specifying the total amount and currency due, the *moneyType* type,
- *DueDate* – specifying the repayment due date (date by which the obligation should be repaid),
- *PaidDate* – specifying the repayment date (date of repayment of the obligation),
- *IsSumOfInstallments* – a flag defining if the information refers to a sum of obligations amounts (whether the obligation has been repaid in installments).

3.2.4.55. Type: *paidObligationDebtorType*

```
<xs:complexType name="paidObligationDebtorType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Consumer" type="paidObligationConsumerType"/>
      <xs:element name="LegalPerson" type="paidObligationLegalPersonType"/>
      <xs:element name="Entrepreneur" type="paidObligationEntrepreneurType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Elements of this type are used to provide information about the type of entity (contractor – payer, positive information) who the economic information about the paid obligations refers to. The element can assume one of the three accepted values:

- *Consumer* – when the information refers to a consumer, the *paidObligationConsumerType* type,
- *LegalPerson* – when the information refers to a legal person or an institution, the *paidObligationLegalPersonType* type,
- *Entrepreneur* – when the information refers to a person running a one-man business, the *paidObligationEntrepreneurType* type.

3.2.4.56. Type: addPaidObligationsType

```
<xs:complexType name="addPaidObligationsType">
  <xs:sequence>
    <xs:element name="Obligation" type="addPaidObligationType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Element of this type are used to group the economic information about the paid (positive) obligations. It contains an unlimited number of nodes of the *addPaidObligationType* type named *Obligation*.

3.2.4.57. Type: addPaidObligationType

```
<xs:complexType name="addPaidObligationType">
  <xs:complexContent>
    <xs:extension base="paidObligationType">
      <xs:attributeGroup ref="addIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

This type is used to extend the *paidObligationType* type with information allowing to submit the identifier for the added information about the paid (positive) obligation.

3.2.4.58. Type: paidObligationConsumerType

```
<xs:complexType name="paidObligationConsumerType">
  <xs:complexContent>
    <xs:extension base="consumerMinType">
      <xs:sequence>
        <xs:element name="SecondName" type="nonEmptyString32" minOccurs="0"/>
        <xs:element name="Address" type="addressType" minOccurs="0"/>
        <xs:element name="MailingAddress" type="addressType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

This type (applies to positive information) is used to extend the element of the *consumerMinType* type with the following nodes:

- *SecondName* – allowing to submit the information about the middle name of the submitted entity. This field is optional and can hold a non-empty string of maximum length of 32 characters,
- *Address* – residence address, *addressType* type,
- *MailingAddress* – mailing address, *addressType* type.

3.2.4.59. Type: `paidObligationLegalPersonType`

```
<xs:complexType name="paidObligationLegalPersonType">
  <xs:complexContent>
    <xs:extension base="legalPersonMinType"/>
  </xs:complexContent>
</xs:complexType>
```

This type (it concerns positive information) includes a *legalPersonMinType* element.

3.2.4.60. Type: `paidObligationEntrepreneurType`

```
<xs:complexType name="paidObligationEntrepreneurType">
  <xs:sequence>
    <xs:element name="Name" type="nonEmptyString128"/>
    <xs:element name="NonConsumerIdentityNumber" type="nonConsumerIdentityNumberType"/>
    <xs:element name="MailingAddress" type="addressType" minOccurs="0"/>
    <xs:element name="ResidenceAddress" type="addressType" minOccurs="0"/>
    <xs:element name="SeatAddress" type="addressType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="isPolishCitizen" type="xs:boolean" use="optional" default="true"/>
</xs:complexType>
```

An element of this type (applies to positive information) contains information about an entity conducting business activity (one-man business). It consists of the following elements:

- *Name* – name of the entity being a non-empty string of maximum length of 128 characters,
- *NonConsumerIdentityNumber* – the entity's identity number, *nonConsumerIdentityNumberType* type,
- *MailingAddress* – mailing address, *addressType* type,
- *ResidenceAddress* – residence address, *addressType* type,
- *SeatAddress* – the address of pursuing economic activity, *addressType* type.

Optionally, the element may contain the *isPolishCitizen* attribute specifying if the entity has the Polish citizenship. By default, this attribute is set to the positive value (*true*).

3.2.4.61. Type: `updatePaidObligationCaseType`

```
<xs:complexType name="updatePaidObligationCaseType">
  <xs:sequence>
    <xs:element name="Debtor" type="paidObligationDebtorType"/>
    <xs:element name="Obligations" type="updatePaidObligationsType"/>
  </xs:sequence>
  <xs:attribute ref="showProvider" use="optional" default="All"/>
</xs:complexType>
```

An element of this type is used to update economic information about a paid (positive) obligation. It consists of the following nodes:

- *Debtor* – containing contractor's – payer's data, *paidObligationDebtorType* type,
- *Obligations* – containing the data of the updated obligations, *updatePaidObligationsType* type.

Optionally, this element may be supplemented with the *showProvider* attribute specifying what information of the person submitting the information are to be disclosed. By default, all data are visible.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

3.2.4.62. Type: updatePaidObligationsType

```
<xs:complexType name="updatePaidObligationsType">
  <xs:sequence>
    <xs:element name="Obligation" type="updatePaidObligationType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type contains an unlimited list of *Obligation* nodes of the *updatePaidObligationType* type and are used to submit the data of the obligations (positive) which are to be updated.

3.2.4.63. Type: updatePaidObligationType

```
<xs:complexType name="updatePaidObligationType">
  <xs:complexContent>
    <xs:extension base="paidObligationType">
      <xs:attributeGroup ref="nonRequiredIdAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

This type is an extension of the *paidObligationType* type with a group of attributes of the *nonRequiredIdAttributeGroup* type. An element of this type is used to submit the data for updating a single economic information about the paid (positive) obligation.

3.2.4.64. Type: processSnapshotPaidObligationCaseType

```
<xs:complexType name="processSnapshotPaidObligationCaseType">
  <xs:sequence>
    <xs:element name="Debtor" type="paidObligationDebtorType"/>
    <xs:element name="Obligations" type="updatePaidObligationsType"/>
  </xs:sequence>
  <xs:attributeGroup ref="idAttributeGroup"/>
</xs:complexType>
```

An element of this type is used to process the so-called "snapshot" of the data with paid (positive) economic information. It is necessary to send all the economic information in the range and form valid at the moment to the system. The system will perform an update if the data already exist, will add new information or delete them. It consists of the following nodes:

- *Debtor* – containing contractor's – payer's data, *paidObligationDebtorType* type,
- *Obligations* – containing the data of the updated obligations, *updatePaidObligationsType* type.

3.3. *nicciInput.xsd* file

The file contains the definitions of the types and elements which are used by the Clients when creating the import files, i.e. the files sent to the bureau.

3.3.1. Definitions of types

3.3.1.1. Type: informationManagementOrdersType

```
<xs:complexType name="informationManagementOrdersType">
  <xs:sequence>
    <xs:element name="Order" type="informationManagementOrderType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```


Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

The *informationManagementOrdersType* type is used to manage economic information (e.g. adding, deleting, updating). An element of this type can contain the following nodes:

- *Order* – of the *informationManagementOrderType* type, storing the information about the action which is to be performed.

3.3.1.2. Type: informationManagementOrderType

```
<xs:complexType name="informationManagementOrderType">
  <xs:complexContent>
    <xs:extension base="orderBaseType">
      <xs:choice>
        <xs:element name="AddInformation" type="addInformationType"/>
        <xs:element name="UpdateInformation" type="updateInformationType"/>
        <xs:element name="RemoveInformation" type="removeInformationType"/>
        <xs:element name="GetInformation" type="entityManagementType"/>
        <xs:element name="SuspendInformation" type="suspendType"/>
        <xs:element name="UnsuspendInformation"
type="entityManagementWithVerificationType"/>
        <xs:element name="SendNotification" type="notifyDebtorOrderType"/>
        <xs:element name="ChangeInformationOwner" type="changeOwnerType"/>
        <xs:element name="GetInformations" type="getInformationsType"/>
        <xs:element name="ProcessSnapshot" type="processSnapshotType"/>
        <xs:element name="GetInformationsVerificationEvents" type="xs:anySimpleType"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *informationManagementOrderType* type is used to define the order for managing economic information (e.g. adding, deleting, updating). It is an extension of the *orderBaseType* type. An element of this type can contain one of the following nodes:

- *AddInformation* – of the *addInformationType* type storing information about adding economic information (a case or an obligation),
- *UpdateInformation* – of the *updateInformationType* type storing information about updating economic information (a case or an obligation),
- *RemoveInformation* – of the *removeInformationType* type storing information about the economic information (a case or an obligation) which is to be deleted,
- *GetInformation* – of the *entityManagementType* type storing information about economic information (a case or an obligation) that one wants to get from our account,
- *SuspendInformation* – of the *suspendType* type storing information about the economic negative information which is intended for suspension together with the data of the term of suspension,
- *UnsuspendInformation* – of the *entityManagementWithVerificationType* type storing information about the economic negative information which is intended for being published again (unsuspend),
- *SendNotification* – of the *notifyDebtorOrderType* type storing information about sending the notification to the debtor on adding to KRD (for information – currently we do not send letters abroad),
- *ChangeInformationOwner* – of the *changeOwnerType* type storing the data of the new owner of the economic information,

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

- *GetInformations* – of the *getInformationsType* type storing the data allowing to get much economic information added by the Client on his account,
- *ProcessSnapshot* – of the *processSnapshotType* type allowing for a snapshot data synchronization,
- *GetInformationsVerificationEvents* – allowing to get the events of negative data verification.

3.3.1.3. Type: monitorManagementOrdersType

```
<xs:complexType name="monitorManagementOrdersType">
  <xs:sequence>
    <xs:element name="Order" type="monitorManagementOrderType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The *monitorManagementOrdersType* type is used to manage the economic information monitors (applies to Tax ID monitoring). An element of this type can contain the following nodes:

- *Order* – of the *monitorManagementOrderType* type storing the information about an action which is to be performed.

3.3.1.4. Type: monitorManagementOrderType

```
<xs:complexType name="monitorManagementOrderType">
  <xs:complexContent>
    <xs:extension base="orderBaseType">
      <xs:choice>
        <xs:element name="AddMonitorCondition" type="monitorContitionManagementType"/>
        <xs:element name="UpdateMonitorCondition" type="monitorContitionManagementType"/>
        <xs:element name="StopMonitorCondition" type="stopMonitorConditionType"/>
        <xs:element name="GetEvents" type="getMonitorEventsType"/>
        <xs:element name="GetMonitorConditions" type="getMonitorConditionsType"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *monitorManagmentOrderType* type is used to define the order for managing monitoring (applies to Tax ID monitoring). It is an extension of the *orderBaseType* type. An element of this type can contain one of the following nodes:

- *AddMonitorCondition* – of the *monitorContitionManagementType* type storing the information about adding the data of the order to monitor the contractor (the monitor uses the identifying number of the contractor – Tax ID),
- *UpdateMonitorCondition* – of the *monitorContitionManagementType* type storing the information about the update of the contractor monitoring order,
- *StopMonitorCondition* – of the *stopMonitorConditionType* type storing the information about the stop (delete) of the contractor monitoring order,
- *GetEvents* – allowing to get a list of monitoring events from the previous calendar day,
- *GetMonitorConditions* – of the *getMonitorConditionsType* type storing the information on the basis of which it is possible to get a list of the existing monitors.

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

3.3.1.5. Type: *monitorContitionManagementType*

```
<xs:complexType name="monitorContitionManagementType">
  <xs:complexContent>
    <xs:extension base="monitorConditionType">
    </xs:complexContent>
  </xs:complexType>
```

The *monitorContitionManagementType* type extends the *monitorConditionType* type with the *VerifyResult* attribute used to define the type of response returned (applies to Tax ID monitoring).

3.3.1.6. Type: *suspendType*

```
<xs:complexType name="suspendType">
  <xs:complexContent>
    <xs:extension base="entityManagementWithVerificationType">
      <xs:sequence>
        <xs:element name="DateFrom" type="xs:date" minOccurs="0"/>
        <xs:element name="DateTo" type="xs:date"/>
        <xs:element name="Comment" type="nonEmptyString128"
minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *suspendType* type is used for suspending the publication of negative information. It is an extension of the *entityManagementWithVerificationType* type. An element of this type contains the following nodes:

- *DateFrom* – specifies the date from which the publication of the economic information is to be suspended,
- *DateTo* – specifies the date to which the publication of the economic information is to be restored,
- *Comment* – a comment for the operation.

3.3.1.7. Type: *notifyDebtorOrderType*

```
<xs:complexType name="notifyDebtorOrderType">
  <xs:complexContent>
    <xs:extension base="entityManagementType">
      <xs:sequence>
        <xs:element name="Notify" type="notifyDebtorType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *notifyDebtorOrderType* type is used to inform the debtor about their arrears. It is an extension of the *entityManagementType* type. An element of this type can contain the following nodes:

- *Notify* – of the *notifyDebtorType*, type which contains the information about the notification, such as channel and mailing data (e.g. mailing address) or the contact details of the creditor, and it is possible to define the bank account number to which the debtor is to repay (for information – currently we do not send letters abroad).

3.3.1.8. Type: *addObligationInformationType*

```
<xs:complexType name="addObligationInformationType">
  <xs:sequence>
    <xs:element name="Case">
      <xs:complexType>
        <xs:attributeGroup ref="idAttributeGroup"/>
    </xs:element>
  </xs:sequence>
```

```

    </xs:complexType>
  </xs:element>
<xs:choice>
  <xs:element name="Obligation" type="obligationType"/>
  <xs:element name="ExecutiveObligation" type="executiveObligationType"/>
  <xs:element name="AlimonyObligation" type="alimonyObligationType"/>
</xs:choice>
</xs:sequence>
</xs:complexType>

```

The *addObligationInformationType* type is used to add a new negative obligation to an existing case. An element of this type can contain the following nodes:

- *Case* – used to identify the case to which an obligation is to be added,
- *Obligation / ExecutiveObligation / AlimonyObligation* – it is possible to add an ordinary obligation of the *ObligationType* type; an obligation declared with an executive title of the *ExecutiveObligationType* type and an obligation declared with an executive title of the *AlimonyObligationType* type.

3.3.1.9. Type: addPaidObligationInformationType

```

<xs:complexType name="addPaidObligationInformationType">
  <xs:sequence>
    <xs:element name="Case">
      <xs:complexType>
        <xs:attributeGroup ref="idAttributeGroup"/>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:element name="Obligation" type="paidObligationType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

The *addPaidObligationInformationType* type is used to add new information about a paid (positive) obligation to an existing case. An element of this type can contain the following nodes:

- *Case* – used to identify the case to which an obligation is to be added,
- *Obligation* – of the *paidObligationType* type.

3.3.1.10. Type: addInformationType

```

<xs:complexType name="addInformationType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Case" type="addCaseType"/>
      <xs:element name="ObligationInformation" type="addObligationInformationType"/>
      <xs:element name="PaidObligationCase" type="addPaidObligationCaseType"/>
      <xs:element name="PaidObligationInformation"
type="addPaidObligationInformationType"/>
    </xs:choice>
  </xs:sequence>
  <xs:attributeGroup ref="addIdAttributeGroup"/>
  <xs:attribute ref="verifyResult" use="optional"/>
  <xs:attributeGroup ref="suspendedAttributes"/>
</xs:complexType>

```

The *addInformationType* type is used to add a new economic information. An element of this type can contain the following nodes:

- *Case* – of the *addCaseType* type containing the data of a new case about unpaid (negative) obligations,

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

- *ObligationInformation* – of the *addObligationInformationType* type containing the data of a new unpaid (negative) obligation,
- *PaidObligationCase* – of the *addPaidObligationCaseType* type containing the data of a new case about paid (positive) obligations,
- *PaidObligationInformation* – of the *addPaidObligationInformationType* type containing the data of a new paid (positive) obligation.

The *addInformationType* element can have an attribute *verifyResult* and a group of attributes *suspendAttributes*.

3.3.1.11. Type: updateInformationType

```
<xs:complexType name="updateInformationType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Case" type="updateCaseType"/>
      <xs:element name="PaidObligationCase" type="updatePaidObligationCaseType"/>
      <xs:element name="ObligationInformation" type="obligationType"/>
      <xs:element name="ExecutiveObligationInformation"
type="executiveObligationType"/>
      <xs:element name="AlimonyObligationInformation"
type="alimonyObligationType"/>
      <xs:element name="PaidObligationInformation" type="paidObligationType"/>
    </xs:choice>
  </xs:sequence>
  <xs:attributeGroup ref="idAttributeGroup"/>
  <xs:attribute ref="verifyResult" use="optional"/>
</xs:complexType>
```

The *updateInformationType* type is used to update economic information (unpaid – negative or paid – positive) or information about a forged / someone else's document. An element of this type can contain the following nodes:

- *Case* – of the *updateCaseType* type containing the economic information of the case about unpaid (negative) obligations,
- *PaidObligationCase* – of the *updatePaidObligationCaseType* type containing the economic information of the case about paid (positive) obligations,
- *ObligationInformation* – of the *obligationType* type containing the data about the updated unpaid (negative) obligation,
- *ExecutiveObligationInformation* – of the *executiveObligationType* type containing the data about the updated unpaid (negative) obligation declared with an executive title,
- *AlimonyObligationInformation* – of the *alimonyObligationType* type containing the data about the updated unpaid (negative) alimony obligation,
- *PaidObligationInformation* – of the *paidObligationType* type containing the data about the updated paid (positive) obligation.

The *updateInformationType* element can have an attribute *verifyResult*.

3.3.1.12. Type: changeOwnerType

```
<xs:complexType name="changeOwnerType">
  <xs:complexContent>
    <xs:extension base="entityManagementType">
      <xs:sequence>
        <xs:element name="LoginName" type="loginNameType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

        </xs:sequence>
        <xs:attribute ref="verifyResult" use="optional"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The *changeOwnerType* type is used to change the owner of the economic information. It is an extension of the *entityManagementType* type. An element of this type can contain the following nodes:

- *LoginName* – of the *loginNameType* type containing the login of the new economic information owner. The *changeOwnerType* element can contain the *verifyResult* attribute.

3.3.1.13. Type: processSnapshotType

```

<xs:complexType name="processSnapshotType">
  <xs:sequence>
<xs:chocie>
  <xs:element name="Case" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="updateCaseType">
          <xs:sequence>
            <xs:element name="Notify" type="notifyDebtorType" minOccurs="0"/>
          </xs:sequence>
          <xs:attributeGroup ref="nonRequiredIdAttributeGroup" />
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="PaidObligationCase" minOccurs="0" maxOccurs="unbounded"
type="processSnapshotPaidObligationCaseType" /></xs:element>
</xs:chocie>
  </xs:sequence>
</xs:complexType>

```

The *processSnapshotType* type is used to process the so-called "snapshot" of the data with economic information. It is necessary to send all the economic information in the range and form valid at the moment to the system. The system will perform an update if the data already exist, will add new information or delete them.

3.3.1.14. Type: stopMonitorConditionType

```

<xs:complexType name="stopMonitorConditionType">
  <xs:sequence>
    <xs:element name="Number" type="nonConsumerIdentityNumberType" />
    <xs:element name="DateFrom" type="xs:date" minOccurs="0" />
    <xs:element name="LoginName" type="nonEmptyString128" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

```

The *stopMonitorConditionType* type is used to stop (delete) of an active monitor condition (applies to Tax ID's monitoring). An element of this type can contain the following nodes:

- *Number* – of the *nonConsumerIdentityNumberType* type contains the identifying number of the contractor (Tax ID).

3.3.1.15. Type: getDisclosureReportType

```

<xs:complexType name="getDisclosureReportType">
  <xs:sequence>
    <xs:element name="Number" type="nonEmptyString32" />
  </xs:sequence>
</xs:complexType>

```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

The *getDisclosureReportType* type is used to download the KRD report from the search with a specific system number:

- *Number* – system number of the generated report.

3.3.1.16. Type: *getDisclosureReportsType*

```
<xs:complexType name="getDisclosureReportsType">
  <xs:sequence>
    <xs:element name="DateFrom" type="xs:dateTime"/>
    <xs:element name="DateTo" type="xs:dateTime"/>
  </xs:sequence>
</xs:complexType>
```

The *getDisclosureReportsType* type is used to download the list of generated KRD reports from the search in a given time range:

- *DateFrom* – start date,
- *DateTo* – end date.

3.3.1.17. Type: *searchNonConsumerType*

```
<xs:complexType name="searchNonConsumerType">
  <xs:choice>
    <xs:element name="EuropeanTaxId" type="nonEmptyString128"/>
    <xs:element name="NonEuropeanNumber" type="nonEmptyString64"/>
  </xs:choice>
</xs:complexType>
```

The *searchNonConsumerType* type is used to search (KRD report) for economic information relating to a non-consumer. An element of this type contains the following nodes:

- *EuropeanTaxId* – containing the identification number of the entity based in the European Union,
- *NonEuropeanNumber* – containing the identification number of the entity based outside the European Union.

3.3.1.18. Type: *searchConsumerType*

```
<xs:complexType name="searchConsumerType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Pesel" type="peselType"/>
      <xs:element name="NonPolishNumber" type="nonEmptyString128"/>
    </xs:choice>
    <xs:element name="AuthorizationDate" type="xs:date"/>
  </xs:sequence>
</xs:complexType>
```

The *searchConsumerType* type is used to search (KRD report) for economic information relating to a consumer. An element of this type contains the following nodes:

- *Pesel* – containing the PESEL number,
- *NonPolishNumber* – containing a different identification number,
- *AuthorizationDate* – containing the authorization date (consent from the consumer to check in the KRD) of the economic information search.

3.3.1.19. Type: *incognitoSearchNonConsumerType*

```
<xs:complexType name="incognitoSearchNonConsumerType">
  <xs:sequence>
    <xs:choice>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

```
<xs:element name="EuropeanTaxId" type="nonEmptyString128"/>
<xs:element name="NonEuropeanNumber" type="nonEmptyString64"/>
</xs:choice>
</xs:sequence>
</xs:complexType>
```

The *incognitoSearchNonConsumerType* type is used to search (KRD report) for economic information relating to a non-consumer in the incognito mode. An element of this type contains the following nodes:

- *EuropeanTaxId* – containing the identification number of the entity based in the European Union,
- *NonEuropeanNumber* – containing the identification number of the entity based outside the European Union.

3.3.1.20. Type: incognitoSearchConsumerType

```
<xs:complexType name="incognitoSearchConsumerType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Pesel" type="peselType"/>
      <xs:element name="NonPolishNumber" type="nonEmptyString128"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

The *incognitoSearchConsumerType* type is used to search (KRD report) for economic information relating to a consumer in the incognito mode. An element of this type contains the following nodes:

- *Pesel* – containing the PESEL number,
- *NonPolishNumber* – containing a different identification number.

3.3.1.21. Type: searchDocumentType

```
<xs:complexType name="searchDocumentType">
  <xs:sequence>
    <xs:element name="FirstName" type="nonEmptyString32"/>
    <xs:element name="Surname" type="nonEmptyString64"/>
  </xs:sequence>
</xs:complexType>
```

The *searchDocumentType* type is used to search for information relating to using a forged / someone else's documents. An element of this type contains the following nodes:

- *FirstName* – first name from the document,
- *Surname* – surname from the document.

3.3.1.22. Type: searchManagementOrderType

```
<xs:complexType name="searchManagementOrderType">
  <xs:complexContent>
    <xs:extension base="orderBaseType">
      <xs:choice>
        <xs:element name="SearchNonConsumer"
type="searchNonConsumerType"/>
        <xs:element name="SearchConsumer"
type="searchConsumerType"/>
        <xs:element name="SearchDocument"
type="searchDocumentType"/>
        <xs:element name="SearchMe"/>
        <xs:element name="GetDisclosureReports"
type="getDisclosureReportsType"/>
        <xs:element name="GetDisclosureReport"
type="getDisclosureReportType"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```



```

        </xs:choice>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The *searchManagementOrderType* type is used to get a report on economic information search (KRD report). It is an extension of the *orderBaseType* type. An element of this type can contain the following nodes:

- *SearchNonConsumer* – of the *searchNonConsumerType* type containing the identification data of the non-consumer for whom we want to get a report,
- *SearchConsumer* – of the *searchConsumerType* type containing the identification data of the consumer for whom we want to get a report,
- *SearchDocument* – of the *searchDocumentType* type containing the identification data of the forged / someone else's document for which we want to get a report,
- *SearchMe* – allows to get a report of the economic information search of the person performing the search,
- *GetDisclosureReports* – of the *getDisclosureReportsType* type containing the range of dates for a report,
- *GetDisclosureReport* – of the *getDisclosureReportType* type containing the number of a report to be requested.

3.3.1.23. Type: searchManagementOrdersType

```

<xs:complexType name="searchManagementOrdersType">
  <xs:sequence>
    <xs:element name="Order" type="searchManagementOrderType"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

```

The *searchManagementOrdersType* type is used to get a report on economic information search (KRD reports). An element of this type can contain the following nodes:

- *Order* – of the *searchManagementOrderType* type.

3.3.1.24. Type: incognitoSearchManagementOrderType

```

<xs:complexType name="incognitoSearchManagementOrderType">
  <xs:complexContent>
    <xs:extension base="orderBaseType">
      <xs:choice>
        <xs:element name="IncognitoSearchConsumer"
type="incognitoSearchConsumerType" />
        <xs:element name="IncognitoSearchNonConsumer"
type="incognitoSearchNonConsumerType" />
        <xs:element name="IncognitoSearchRegistryReportConsumer"
type="incognitoSearchRegistryReportConsumerType" />
        <xs:element name="IncognitoSearchRegistryReportNonConsumer"
type="incognitoSearchRegistryReportNonConsumerType" />
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The *incognitoSearchManagementOrderType* is used to download a report of economic information search (KRD report) in incognito mode. It is an

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

extension of *orderBaseType*. An element of this type can contain the following nodes:

- *IncognitoSearchConsumer* – of the *incognitoSearchConsumerType* type containing the identification information of the requested consumer,
- *IncognitoSearchNonConsumer* – of the *incognitoSearchNonConsumerType* type containing the identification information of the requested business entity (one-man business, legal persons, entrepreneurs),
- *IncognitoSearchRegistryReportConsumer* – of the *incognitoSearchRegistryReportConsumerType* type containing data from the consumer inquiry register (requests history). They are information on who inquired about the consumer and what economic information about the consumer was provided to the requester,
- *IncognitoSearchRegistryReportNonConsumer* – of the *incognitoSearchRegistryReportNonConsumerType* type containing data from the business entity (one-man business, legal persons, entrepreneurs) inquiry register (requests history). They are information on who inquired about the business entity and what economic information about the business entity was provided to the requester.

3.3.1.25. Type: *incognitoSearchManagementOrdersType*

```
<xs:complexType name="incognitoSearchManagementOrdersType">
  <xs:sequence>
    <xs:element name="Order" type="incognitoSearchManagementOrderType"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

The *incognitoSearchManagementOrdersType* is used to download reports of economic information search (KRD reports) in incognito mode. An element of this type can contain the following nodes:

- *Order* – of the *incognitoSearchManagementOrderType* type.

3.3.2. The main Input element

```
<xs:element name="Input">
  <xs:complexType>
    <xs:choice>
      <xs:element name="InformationManagement" type="informationManagementOrdersType"/>
      <xs:element name="MonitorConditionManagement" type="monitorManagementOrdersType"/>
      <xs:element name="SearchManagement" type="searchManagementOrdersType"/>
      <xs:element name="IncognitoSearchManagement"
type="incognitoSearchManagementOrdersType" />
    </xs:choice>
    <xs:attributeGroup ref="inputAttributes"/>
  </xs:complexType>
</xs:element>
```

The Input type is used to start a protocol of information exchange. An element of this type can contain the following nodes:

- *InformationManagement* – of the *informationManagementOrdersType* type, which is responsible for orders for performing operations on economic information (adding information, updating information, removing information, etc.),
- *MonitorConditionManagement* – of the *monitorManagementOrdersType* type, which is responsible for orders for performing operations (adding,

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

updating, removing) on economic information monitors (applies to Tax ID's monitoring),

- *SearchManagement* – of the *searchManagement OrdersType* type, which is responsible for requesting the performance of the economic information search operation (KRD report),
- *IncognitoSearchManagement* – of the *incognitoSearchManagement OrdersType* type, which is responsible for requesting the performance of the economic information search operation (KRD report) in incognito mode.

3.4. *nicciOutput.xsd* file

The file contains definitions of types and elements that are used by the bureau when creating files with answers to Clients orders.

3.4.1. Definitions attribute

3.4.1.1. Output attributes

```
<xs:attributeGroup ref="inputAttributes"/>
<xs:attribute name="started" type="xs:dateTime" use="required"/>
<xs:attribute name="processed" type="xs:dateTime" use="required"/>
<xs:attribute name="orderCount" type="xs:int" use="required"/>
<xs:attribute name="successCount" type="xs:int" use="required"/>
<xs:attribute name="failCount" type="xs:int" use="required"/>
```

To the output attributes belong a subgroup of input attributes (*inputAttributes*) and additional data describing the results of operations. These data are:

- *started* – required attribute specifying the date and time to start the processing tranche,
- *processed* – required attribute specifying the date and time of completion of processing tranche,
- *orderCount* – required attribute specifies the number of outsourced operations,
- *successCount* – required attribute specifies the number of operations completed successfully,
- *failCount* – required attribute specifies the number of operations ended with an error.

3.4.2. Simple types

3.4.2.1. Type: *responseStatusEnum*

```
<xs:simpleType name="responseStatusEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Success"/>
    <xs:enumeration value="Fail"/>
  </xs:restriction>
</xs:simpleType>
```

An element of this type is used to deliver the information about the result of the operation to the Client. The element can assume one of the two accepted values:

- *Success* – in the case of a positive result of processing,
- *Fail* – in the case of an error.

3.4.2.2. Type: searchCriterionTypeEnum

```
<xs:simpleType name="searchCriterionTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="TaxId"/>
    <xs:enumeration value="Pesel"/>
    <xs:enumeration value="Name"/>
    <xs:enumeration value="Passport"/>
    <xs:enumeration value="IdCard"/>
    <xs:enumeration value="DrivingLicense"/>
    <xs:enumeration value="SocialInsurance"/>
    <xs:enumeration value="Other"/>
    <xs:enumeration value="OtherThanPesel"/>
    <xs:enumeration value="OtherThanTaxId"/>
  </xs:restriction>
</xs:simpleType>
```

An element of this type is used to deliver the information about the type of performed search (KRD report) to the Client. The element can assume one of the following values:

- *TaxId* – search by the Tax ID number,
- *Pesel* – search by the PESEL number,
- *Name* – search by the name,
- *Passport* – search by the passport number,
- *IdCard* – search by the ID card,
- *DrivingLicense* – search by the driving license number,
- *SocialInsurance* – search by the insurance number,
- *Other* – a different search method,
- *OtherThanPesel* – a search method different than by PESEL number,
- *OtherThanTaxId* – a search method different than by Tax ID number.

3.4.2.3. Type: searchType

```
<xs:simpleType name="searchType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="RegularSearch"/>
    <xs:enumeration value="SelfSearch"/>
  </xs:restriction>
</xs:simpleType>
```

An element of this type is used to deliver the information about the type of the performed check (KRD report) to the Client. The element can assume one of the two accepted values:

- *RegularSearch* – in the case of checking the ID number different than one's own,
- *SelfSearch* – in the case of checking one's own ID number.

3.4.3. Definitions of types

3.4.3.1. Type: orderResponseType

```
<xs:complexType name="orderResponseType">
  <xs:complexContent>
    <xs:extension base="orderBaseType">
      <xs:attribute name="status" type="responseStatusEnum" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *orderResponseType* is used to return information about the request sent to be processed. This type contains the base type for *orderBaseType*. Additionally, this type contains the obligatory *status* element, in which it is necessary to submit one of the values of the *responseStatusEnum* type.

3.4.3.2. Type: simpleResponseInformationType

```
<xs:complexType name="simpleResponseInformationType">
  <xs:sequence>
    <xs:element name="Error" type="errorType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="ID" type="nonEmptyString128" use="required"/>
  <xs:attribute name="IDType" type="idTypeEnum" use="optional" default="UserId"/>
  <xs:attribute name="informationType" type="informationTypeEnum" use="required"/>
</xs:complexType>
```

The *simpleResponseInformationType* type is used to return the basic data referring to the responses related to economic information. It contains an optional element *Error* (of the *errorType* type), attributes informing about the identifier (*ID* and *IDType*) and the type of entity (*informationType*). The *IDType* attribute contains one of the values of the *idTypeEnum* type (by default, the value of the attribute is "*UserId*").

3.4.3.3. Type: errorType

```
<xs:complexType name="errorType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="code" type="xs:int" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The *errorType* type is used to return the information about an error which occurred during processing the request. It is an extension of the basic [string](#) type with the *code* attribute. The *code* attribute is optional and contains the values from basic [int](#) type (integer), which indicate the code of the returned error.

3.4.3.4. Type: responseObligationType

```
<xs:complexType name="responseObligationType">
  <xs:complexContent>
    <xs:extension base="obligationType">
      <xs:attributeGroup ref="outputIdAttributes"/>
      <xs:attributeGroup ref="entityAttributes"/>
      <xs:attributeGroup ref="suspendedAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *responseObligationType* type is used in the case of returning the information about ordinary obligations (negative informations). This type is an extension of the *obligationType* type additionally extended with attribute groups *outputIdAttributes*, *entityAttributes* and *suspendedAttributes*.

3.4.3.5. Type: responseExecutiveObligationType

```
<xs:complexType name="responseExecutiveObligationType">
  <xs:complexContent>
    <xs:extension base="executiveObligationType">
      <xs:attributeGroup ref="outputIdAttributes"/>
      <xs:attributeGroup ref="entityAttributes"/>
      <xs:attributeGroup ref="suspendedAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
</xs:complexContent>
</xs:complexType>
```

The *responseExecutiveObligationType* type is used in the case of returning the information about obligations declared with an executive title (negative informations). This type is an extension of the *executiveObligationType* type additionally extended with attribute groups *outputIdAttributes*, *entityAttributes* and *suspendedAttributes*.

3.4.3.6. Type: responseAlimonyObligationType

```
<xs:complexType name="responseAlimonyObligationType">
  <xs:complexContent>
    <xs:extension base="alimonyObligationType">
      <xs:attributeGroup ref="outputIdAttributes"/>
      <xs:attributeGroup ref="entityAttributes"/>
      <xs:attributeGroup ref="suspendedAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *responseAlimonyObligationType* type is used in the case of returning the information about alimony obligations (negative informations). This type is an extension of the *alimonyObligationType* type additionally extended with attribute groups *outputIdAttributes*, *entityAttributes* and *suspendedAttributes*.

3.4.3.7. Type: responsePaidObligationType

```
<xs:complexType name="responsePaidObligationType">
  <xs:complexContent>
    <xs:extension base="paidObligationType">
      <xs:attributeGroup ref="outputIdAttributes"/>
      <xs:attributeGroup ref="entityAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The *responsePaidObligationType* type is used in the case of returning the information about the paid obligations (positive informations). This type is an extension of the *paidObligationType* type additionally extended with attribute groups *outputIdAttributes* and *entityAttributes*.

3.4.3.8. Type: responseCaseType

```
<xs:complexType name="responseCaseType">
  <xs:sequence>
    <xs:element name="Debtor" type="debtorType"/>
    <xs:element name="Obligations" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Obligation" type="responseObligationType"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="ExecutiveObligations" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ExecutiveObligation" type="responseExecutiveObligationType"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AlimonyObligations" minOccurs="0">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="AlimonyObligation" type="responseAlimonyObligationType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Note" type="nonEmptyString1024" minOccurs="0"/>
<xs:element name="BlockReason" type="nonEmptyString256" minOccurs="0"/>
</xs:sequence>
<xs:attributeGroup ref="outputIdAttributes"/>
<xs:attributeGroup ref="entityAttributes"/>
<xs:attributeGroup ref="suspendedAttributes"/>
<xs:attribute name="loginName" type="nonEmptyString128" use="optional"/>
<xs:attribute name="showProvider" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="blocked" type="xs:boolean" use="optional"/>
<xs:attribute name="wait" type="xs:boolean" use="optional"/>
<xs:attribute name="doubtful" type="xs:boolean" use="optional"/>
<xs:attribute name="showDoubt" type="xs:boolean" use="optional"/>
</xs:complexType>

```

The *responseCaseType* type is used to return the case information (negative information). The main element can contain attributes:

- *loginName* – being a string of maximum length of 128 characters, the attribute contains the information about the login which submitted the case,
- *showProvider* – being a boolean value (true, false, 1, 0), the attribute specifies if the data of the information provider are to be published,
- *blocked* – being a boolean value (true, false, 1, 0), the attribute specifies if the case is blocked,
- *wait* – the case awaits verification,
- *doubtful* – the case marked as doubtful, e.g. there is a suspicion that the provided addresses are incorrect,
- *showDoubt* – specifies if the data related to the doubts are to be visible.

Additionally, the element contains attribute groups of the *outputIdAttributes*, *entityAttributes* and *suspendedAttributes* types.

This type also includes a sequence of elements:

- *Debtor* – element of the *debtorType* type defining the debtor,
- *Obligations* – a list of *Obligation* elements of the *responseObligationType* type, containing the data of the ordinary obligations; the list can be empty,
- *ExecutiveObligations* – a list of *ExecutiveObligation* elements of the *responseExecutiveObligationType* type, containing the data of the obligations declared by an executive title; the list can be empty,
- *AlimonyObligations* – a list of *AlimonyObligation* elements of the *responseAlimonyObligationType* type, containing the data of the alimony obligations; the list can be empty,
- *Note* – an optional note [contains additional information about the obligation (not disclosed in KRD reports)],
- *BlockReason* – the reason for blocking the case.

3.4.3.9. Type: responsePaidObligationCaseType

```
<xs:complexType name="responsePaidObligationCaseType">
  <xs:sequence>
    <xs:element name="Debtor" type="paidObligationDebtorType"/>
    <xs:element name="Obligations" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Obligation" type="responsePaidObligationType"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="outputIdAttributes"/>
  <xs:attributeGroup ref="entityAttributes"/>
  <xs:attribute name="loginName" type="nonEmptyString128" use="optional"/>
  <xs:attribute ref="showProvider" use="optional" default="All"/>
</xs:complexType>
```

The *responsePaidObligationCaseType* type is used to return the paid obligations cases information (positive information). The main element can contain attributes:

- *loginName* – being a string of maximum length of 128 characters, the attribute contains the information about the login which submitted the case,
- *showProvider* – being a boolean value (true, false, 1, 0), the attribute specifies if the data of the information provider are to be published.

Additionally, the element contains attribute groups of the *outputIdAttributes* and *entityAttributes* types.

This type also includes a sequence of elements:

- *Debtor* – element of the *paidObligationDebtorType* type defining the contractor – payer,
- *Obligations* – a list of *Obligation* elements of the *responsePaidObligationType* type, containing the data of the paid obligations; the list can be empty.

3.4.3.10. Type: informationResponseType

```
<xs:complexType name="informationResponseType">
  <xs:complexContent>
    <xs:extension base="simpleResponseInformationType">
      <xs:choice>
        <xs:element name="Case" minOccurs="0">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="responseCaseType">
                <xs:attribute name="notifyDebtor"
type="xs:boolean" use="optional" default="false"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="PaidObligationCase"
type="responsePaidObligationCaseType" minOccurs="0"/>
        <xs:element name="ObligationInformation"
type="responseObligationType" minOccurs="0"/>
        <xs:element name="AlimonyObligationInformation"
type="responseAlimonyObligationType" minOccurs="0"/>
        <xs:element name="ExecutiveObligationInformation"
type="responseExecutiveObligationType" minOccurs="0"/>
        <xs:element name="PaidObligationInformation"
```


Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

```

type="responsePaidObligationType" minOccurs="0"/>
  </xs:choice>
  <xs:attribute ref="verifyResult" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

The *informationResponseType* type is used to return the economic information. This type is an extension of the *simpleResponseInformationType* type. The element can contain the *verifyResult* attribute. An element of this type can contain a list including one of the six types of information:

- *Case* – this element is an extension of the *responseCaseType* type (negative information), additionally the element may contain an attribute of the *notifyDebtor* type defining if the debtor was notified about adding to KRD,
- *PaidObligationCase* – this element is an extension of the *responsePaidObligationCaseType* type (positive information),
- *ObligationInformation* – of the *responseObligationType* type, contains data of the negative obligations,
- *AlimonyObligationInformation* – of the *responseAlimonyObligationType* type, contains data of the negative alimony obligations,
- *ExecutiveObligationInformation* – of the *responseExecutiveObligationType* type, contains data of the negative obligations declared by an executive title,
- *PaidObligationInformation* – of the *responsePaidObligationType* type, contains data of the positive obligations.

3.4.3.11. Type: monitorManagementResponseType

```

<xs:complexType name="monitorManagementResponseType">
  <xs:choice>
    <xs:element name="Error" type="errorType" minOccurs="0"/>
    <xs:element name="MonitorCondition" type="monitorConditionResponseType" minOccurs="0"/>
  </xs:choice>
</xs:complexType>

```

The *monitorManagementResponseType* type is used to return the information about the monitoring conditions (applies to monitored Tax ID's). This type contains one of the two possible elements:

- *Error* – returned in the case of failure, *errorType* type,
- *MonitorCondition* – returned in the case of successful operation, *monitorConditionResponseType* type.

3.4.3.12. Type: monitorConditionResponseType

```

<xs:complexType name="monitorConditionResponseType">
  <xs:complexContent>
    <xs:extension base="monitorConditionType">
      <xs:attribute name="loginName" type="nonEmptyString128" use="required"/>
      <xs:attribute ref="verifyResult" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The *monitorConditionResponseType* type is used to return information about a particular monitoring condition (applies to monitored Tax ID's). This type is an extension of the *monitorConditionType* type. The element must contain the *loginName* attribute, being a string of maximum length of 128 characters

containing the login which created the monitoring condition. Additionally, the element can contain an attribute of the *verifyResult* type.

3.4.3.13. Type: *monitorEventResponseType*

```
<xs:complexType name="monitorEventResponseType">
  <xs:sequence>
    <xs:element name="Number" type="nonConsumerIdentityNumberType"/>
    <xs:element name="Date" type="xs:date"/>
    <xs:element name="EventType" type="monitorEventTypeEnum"/>
  </xs:sequence>
</xs:complexType>
```

The *monitorEventResponseType* type contains information about an event that occurs in the KRD system (applies to monitored Tax ID's). It consists of the elements:

- *Number* – containing the Tax ID of the monitored entity,
- *Date* – containing the date of processing / generating of the monitoring event (not the date of the event itself) [details [here](#) ("GetEvents")],
- *EventType* – containing the type of the event.

3.4.3.14. Type: *getInformationsVerificationEventsResult*

```
<xs:complexType name="getInformationsVerificationEventsResult">
  <xs:sequence>
    <xs:element name="Event" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Status">
            <xs:simpleType>
              <xs:restriction base="nonEmptyString16">
                <xs:enumeration value="Blocked"/>
                <xs:enumeration value="Doubtful"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="Result" type="nonEmptyString64"/>
        </xs:sequence>
        <xs:attributeGroup ref="outputIdAttributes"/>
        <xs:attribute name="informationType" type="informationTypeEnum" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The *getInformationsVerificationEventsResult* type defines the type of event negative data verification in KRD system. It contains a list of *Event* elements [attribute specifying the type of verified information ("Case" – case)] which consists of:

- *Status* – element receiving value „Blocked“ (blocked) or „Doubtful“ (doubtful) – information about doubtfully entered data in the case ("Case"),
- *Result* – element containing information what is in doubt (e.g. „ResidenceAddress_Invalid_Street“),
- *outputIdAttributes* – attributes group containing system and own case identifiers ("Case") (if they were assigned when adding the case),
- *informationType* – attribute specifying the type of verified economic information.

3.4.3.15. Type: disclosureReportsResponseType

```
<xs:complexType name="disclosureReportsResponseType">
  <xs:choice>
    <xs:element name="Error" type="errorType"/>
    <xs:element name="DisclosureReports">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="DisclosureReport" type="disclosureReportType"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

The *disclosureReportsResponseType* type is used to return information about the economic information request result (KRD report). This type contains one of the two possible elements:

- *Error* – returned in the case of failure operation, *errorType* type,
- *DisclosureReport* – returned in the case of successful operation, contains a collection of the elements *disclosureReportType* types.

3.4.3.16. Type: disclosureReportResponseType

```
<xs:complexType name="disclosureReportResponseType">
  <xs:choice>
    <xs:element name="Error" type="errorType"/>
    <xs:element name="DisclosureReport" type="disclosureReportType"/>
  </xs:choice>
</xs:complexType>
```

The *disclosureReportResponseType* type is used to return information about the economic information request result (KRD report). This type contains one of the two possible elements:

- *Error* – returned in the case of failure operation, *errorType* type,
- *DisclosureReport* – returned in the case of successful operation, contains an element of the *disclosureReportType* type.

3.4.3.17. Type: requesterType

```
<xs:complexType name="requesterType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Consumer" type="consumerMinType"/>
      <xs:element name="LegalPerson" type="legalPersonMinType"/>
      <xs:element name="Entrepreneur" type="entrepreneurMinType"/>
    </xs:choice>
    <xs:element name="LoginFullName" type="nonEmptyString64" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

The *requesterType* is used to specify the type of person who made the request (download the KRD report). The element contains subelements:

- *LoginFullName* – specifying the full name of the requester (usually name and surname) and being a string of maximum length of 64 characters.

And one of the following elements:

- *Consumer* – of the *consumerMinType* type, present in the case when a consumer is the requester,

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

- *LegalPerson* – of the *legalPersonMinType* type, present in the case when a legal person is the requester,
- *Entrepreneur* – of the *entrepreneurMinType* type, present in the case when an entrepreneur is the requester.

3.4.3.18. Type: providerType

```
<xs:complexType name="providerType">
  <xs:choice>
    <xs:element name="Consumer" type="consumerProviderType"/>
    <xs:element name="LegalPerson" type="legalPersonProviderType"/>
    <xs:element name="Entrepreneur" type="entrepreneurProviderType"/>
  </xs:choice>
  <xs:attribute name="category" type="xs:int" />
</xs:complexType>
```

The *providerType* type is used to specify the type of the person (provider) providing the economic information. The element contains one of the subelements:

- *Consumer* – of the *consumerProviderType* type, when a consumer is the data provider,
- *LegalPerson* – of the *legalPersonProviderType* type, when a legal person is the data provider,
- *Entrepreneur* – of the *entrepreneurProviderType* type, when an entrepreneur is the data provider.

Additionally, it contains the following attribute / field:

- *category* – specifying the identification number of the provider industry category.

The attribute / field contains specific numerical values for which dictionaries have been developed, which are available at the Client's request. Values in this attribute / field appear after enabling the permissions / dictionary, according to business arrangements.

3.4.3.19. Type: legalPersonProviderType

```
<xs:complexType name="legalPersonProviderType">
  <xs:sequence>
    <xs:group ref="requiredLegalPersonDataGroup" minOccurs="0"/>
    <xs:group ref="nonRequiredLegalPersonDataGroup" minOccurs="0"/>
    <xs:element name="Sector" type="nonEmptyString128" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to present the entrepreneurs' and institutions' (provider) data. It consists of the *requiredLegalPersonDataGroup* type, the non-obligatory *nonRequiredLegalPersonDataGroup* type and the *Sector* field specifying the business sector.

3.4.3.20. Type: entrepreneurProviderType

```
<xs:complexType name="entrepreneurProviderType">
  <xs:sequence>
    <xs:group ref="requiredEntrepreneurDataGroup" minOccurs="0"/>
    <xs:group ref="nonRequiredEntrepreneurDataGroup" minOccurs="0"/>
    <xs:element name="Sector" type="nonEmptyString128" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

An element of this type is used to present the data of a person running a one-man business (provider). It consists of the *requiredEntrepreneurDataGroup* type, the non-obligatory *nonRequiredEntrepreneurDataGroup* type and the *Sector* field specifying the business sector.

3.4.3.21. Type: consumerProviderType

```
<xs:complexType name="consumerProviderType">
  <xs:sequence>
    <xs:group ref="requiredConsumerDataGroup" minOccurs="0"/>
    <xs:group ref="nonRequiredConsumerDataGroup" minOccurs="0"/>
    <xs:element name="Sector" type="nonEmptyString128" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

An element of this type is used to present the consumer's (provider) data. It consists of the *requiredConsumerDataGroup* type, the non-obligatory *nonRequiredConsumerDataGroup* type and the *Sector* field specifying the business sector.

3.4.3.22. Type: searchCriterionType

```
<xs:complexType name="searchCriterionType">
  <xs:sequence>
    <xs:element name="NonConsumerIdentityNumber" type="nonConsumerIdentityNumberType"
minOccurs="0"/>
    <xs:element name="ConsumerIdentityNumber" type="consumerIdentityNumberType"
minOccurs="0"/>
    <xs:element name="Name" type="nonEmptyString256" minOccurs="0"/>
    <xs:element name="AuthorizationDate" type="xs:date" minOccurs="0"/>
    <xs:element name="SearchType" type="searchType" default="RegularSearch" minOccurs="0"/>
    <xs:element name="SearchCriterionType" type="searchCriterionTypeEnum"/>
  </xs:sequence>
</xs:complexType>
```

The *searchCriterionType* type is used to return the search conditions specified in the request (KRD report). An element of this type can contain the following nodes:

- *NonConsumerIdentityNumber* – of the *nonConsumerIdentityNumberType* type, storing the identification number for entities other than a consumer,
- *ConsumerIdentityNumber* – of the *consumerIdentityNumberType* type, storing the identification number for consumers,
- *Name* – storing the name of the entity, being a non-empty string of maximum length of 256 characters,
- *AuthorizationDate* – storing the date of obtaining consent from the consumer to check in the KRD, which is a date,
- *SearchType* – of the *searchType* type, storing the type of the performed search, the default value is *RegularSearch*,
- *SearchCriterionType* – of the *searchCriterionTypeEnum* type, storing the type of the search criteria.

3.4.3.23. Type: disclosureReportType

```
<xs:complexType name="disclosureReportType">
  <xs:sequence>
    <xs:element name="Number" type="nonEmptyString128"/>
    <xs:element name="Created" type="xs:dateTime"/>
    <xs:element name="SearchCriterion" type="searchCriterionType"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:element name="Requester" type="requesterType"/>
<xs:element name="Summary">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="InformationCount" type="xs:int"/>
      <xs:element name="OInformationCount" type="xs:int"/>
      <xs:element name="PInformationCount" type="xs:int"/>
      <xs:element name="DInformationCount" type="xs:int"/>
      <xs:element name="DebtorsCount" type="xs:int"/>
      <xs:element name="CreditorsCount" type="xs:int"/>
      <xs:element name="TotalArrears" type="moneyType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PositiveInformationSummary" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProvidersCount" type="xs:int"/>
      <xs:element name="PaidObligationsCount" type="xs:int"/>
      <xs:element name="AveragePaidDebt" type="decimalType"/>
      <xs:element name="AveragePaymentTimeInDays" type="xs:int"/>
      <xs:element name="SumPaidDebt" type="decimalType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Report" type="reportType" minOccurs="0" />
</xs:sequence>
</xs:complexType>

```

This type contains a full KRD report of the economic information stored in the system according to the specified search criteria.

3.4.3.24. Type: genericDisclosureReportResponseType

```

<xs:complexType name="genericDisclosureReportResponseType">
  <xs:choice>
    <xs:element name="Error" type="errorType"/>
    <xs:element name="GenericDisclosureReport" type="genericDisclosureReportType"/>
  </xs:choice>
</xs:complexType>

```

The *genericDisclosureReportResponseType* type is used to generate result information on the inquiry (request) about a consumer or business entity inquiry register (requests history) in incognito mode. The type contains one of two elements:

- *Error* – generated in case of failure operation, *errorType* type,
- *GenericDisclosureReport* – generated if the operation succeeds, contains a *genericDisclosureReportType* type element.

3.4.3.25. Type: genericDisclosureReportType

```

<xs:complexType name="genericDisclosureReportType">
  <xs:sequence>
    <xs:element name="RegistryReportNumber"
type="nonEmptyString40"/>
    <xs:element name="Created"
type="xs:dateTime"/>
    <xs:choice>
      <xs:element name="IncognitoSearchRegistryReportConsumer"
type="incognitoSearchRegistryReportConsumerType"/>
      <xs:element name="IncognitoSearchRegistryReportNonConsumer"
type="incognitoSearchRegistryReportNonConsumerType"/>
    </xs:choice>
    <xs:element name="RegistryReportRequester"
type="requesterType"/>
  </xs:sequence>
</xs:complexType>

```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

```

<xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element name="DisclosureReport"
type="registryDisclosureReportType" minOccurs="0" maxOccurs="1"/>
  <xs:element name="SimpleDisclosureReport"
type="simpleDisclosureReportType" minOccurs="0" maxOccurs="1"/>
  <xs:element name="DebtExceedReport"
type="debtExceedReportType" minOccurs="0" maxOccurs="1"/>
</xs:choice>
</xs:sequence>
</xs:complexType>

```

This type contains complete information from the consumer or business entity inquiry register (requests history). An element of this type contains the following elements:

- *RegistryReportNumber* – identification number of the generated report from the inquiry register (requests history),
- *Created* – report generation date,
- *RegistryReportRequester* – details of the authorized entity requesting for disclosure of the inquiry register (requests history).

It also contains one of the elements:

- *IncognitoSearchRegistryReportConsumer* – consumer search criteria,
- *IncognitoSearchRegistryReportNonConsumer* – business entity search criteria.

It can contain multiple elements:

- *SimpleDisclosureReport* – disclosed data from the inquiry register (requests history), a simplified version of the report containing economic information, determining whether the entity has unpaid debts,
- *DisclosureReport* – a detailed report containing economic information about the consumer or business entity,
- *DebtExceedReport* – returned economic information on the existence of unpaid liabilities over 2.000 PLN / 0.05% of annual revenue.

3.4.3.26. Type: incognitoSearchRegistryReportConsumerType

```

<xs:complexType name="incognitoSearchRegistryReportConsumerType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="NonPolishNumber" type="nonEmptyString40"/>
      <xs:element name="Pesel" type="nonEmptyString40"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

This type contains information about the consumer search criteria in the order to disclose information from the inquiry register (requests history).

3.4.3.27. Type: incognitoSearchRegistryReportNonConsumerType

```

<xs:complexType name="incognitoSearchRegistryReportNonConsumerType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="EuropeanTaxId" type="nonEmptyString40"/>
      <xs:element name="NonEuropeanNumber" type="nonEmptyString40"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

This type contains information about the business entity search criteria in the order to disclose information from the inquiry register (requests history).

3.4.3.28. Type: simpleDisclosureReportType

```
<xs:complexType name="simpleDisclosureReportType">
  <xs:sequence>
    <xs:element name="Number" type="nonEmptyString40" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="Created" type="xs:dateTime" minOccurs="1"
maxOccurs="1" />
    <xs:element name="SearchCriterion" type="searchCriterionType" minOccurs="0"
maxOccurs="1" />
    <xs:element name="Requester" type="disclosureReportRequesterType" minOccurs="0"
maxOccurs="1" />
    <xs:element name="ObligationsInformationsExists" type="xs:boolean" minOccurs="1"
maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

A simplified version of the report from the inquiry register (requests history) containing economic information, determining whether the entity had unpaid debts. It consists of:

- *Number* – report number,
- *Created* – report creation date,
- *SearchCriterion* – entity's search criteria,
- *Requester* – the entity requesting for economic information,
- *ObligationsInformationsExists* – *true* / *false* value specifying whether the entity had unpaid debts.

3.4.3.29. Type: disclosureReportRequesterType

```
<xs:complexType name="disclosureReportRequesterType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Consumer" type="consumerMinType"/>
      <xs:element name="LegalPerson" type="legalPersonMinType"/>
      <xs:element name="Entrepreneur" type="entrepreneurMinType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

The requesterType type is used to determine the type of person who made the inquiry (downloading the KRD report). The element contains one of the elements:

- *Consumer* – *consumerMinType* type, present when the request is made by a consumer,
- *LegalPerson* – *legalPersonMinType* type, present when the request is made by a company,
- *Entrepreneur* – *entrepreneurMinType* type, present when the request is made by a one-man business.

3.4.3.30. Type: debtExceedReportType

```
<xs:complexType name="debtExceedReportType">
  <xs:sequence>
    <xs:element name="Number" type="nonEmptyString40" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="Created" type="xs:dateTime" minOccurs="1"
maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```


Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

```

<xs:element name="SearchCriterion" type="searchCriterionType" minOccurs="0"
maxOccurs="1"/>
<xs:element name="Requester" type="requesterType" minOccurs="0"
maxOccurs="1"/>
<xs:element name="IsDebtExceeded" type="xs:boolean" minOccurs="1"
maxOccurs="1"/>
</xs:sequence>
</xs:complexType>

```

Economic information on the existence of unpaid liabilities over 2.000 PLN / 0.05% of annual revenue. An element of this type contains the following sub-elements:

- *Number* – report number,
- *Created* – report creation date,
- *SearchCriterion* – search criterion for a consumer or business entity,
- *Requester* – details of the entity requesting for economic information,
- *IsDebtExceeded* – true/false value specifying whether the entity had unpaid liabilities.

3.4.3.31. Type: registryDisclosureReportType

```

<xs:complexType name="registryDisclosureReportType">
  <xs:sequence>
    <xs:element name="Number" type="nonEmptyString40" minOccurs="1" maxOccurs="1"/>
    <xs:element name="Created" type="xs:dateTime" minOccurs="1" maxOccurs="1" />
    <xs:element name="SearchCriterion" type="searchCriterionType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="Requester" type="disclosureReportRequesterType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="Summary">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="InformationCount" type="xs:int"/>
          <xs:element name="OInformationCount" type="xs:int"/>
          <xs:element name="PInformationCount" type="xs:int"/>
          <xs:element name="DInformationCount" type="xs:int"/>
          <xs:element name="DebtorsCount" type="xs:int"/>
          <xs:element name="CreditorsCount" type="xs:int"/>
          <xs:element name="TotalArrears" type="moneyType"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="PositiveInformationSummary" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ProvidersCount" type="xs:int"/>
          <xs:element name="PaidObligationsCount" type="xs:int"/>
          <xs:element name="AveragePaidDebt" type="decimalType"/>
          <xs:element name="AveragePaymentTimeInDays" type="xs:int"/>
          <xs:element name="SumPaidDebt" type="decimalType" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Report" type="reportType" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

The *registryDisclosureReportType* type is a disclosed, detailed report containing economic information about a consumer or business entity. It contains economic information about the entity that is up-to-date when the inquiry was processed. In simple terms, these are data about who inquired

about the entity or consumer and what information they obtained. An element of this type contains the following sub-elements:

- *Number* – report number,
- *Created* – report creation date,
- *SearchCriterion* – search criterion for a consumer or business entity,
- *Requester* – details of the entity requesting for economic information,
- *Summary* – summary of information on unpaid (negative) obligations,
- *PositiveInformationSummary* – summary of information on paid (positive) obligations,
- *Report* – a detailed report on unpaid (negative) obligations.

3.4.3.32. Type: reportType

```
<xs:complexType name="reportType">
  <xs:sequence>
    <xs:element name="ObligationInformations" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ObligationInformation" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Debtor" type="debtorType"/>
                <xs:element name="Provider" type="providerType" minOccurs="0"/>
                <xs:choice>
                  <xs:sequence>
                    <xs:element name="Title" type="nonEmptyString512"/>
                    <xs:element name="Type" type="obligationTypeEnum" minOccurs="0"/>
                  </xs:sequence>
                  <xs:sequence>
                    <xs:element name="Type" type="obligationTypeEnum"/>
                  </xs:sequence>
                </xs:choice>
                <xs:element name="CustomType" type="nonEmptyString128" minOccurs="0"/>
                <xs:element name="PaymentDate" type="xs:dateTime"/>
                <xs:element name="Proceedings" type="nonEmptyString1024" minOccurs="0"/>
                <xs:element name="Objections" type="nonEmptyString1024" minOccurs="0"/>
                <xs:element name="NoObjections" type="xs:boolean" minOccurs="0"/>
                <xs:element name="CallSent" type="xs:dateTime" minOccurs="0"/>
                <xs:element name="Debt" type="moneyType" minOccurs="0"/>
                <xs:element name="Arrears" type="moneyType"/>
                <xs:element name="Category" type="xs:int" minOccurs="0"/>
                <xs:element name="Subcategory" type="xs:int" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="PaidObligationInformations" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="PaidObligationInformation" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Subject" type="subjectType"/>
                <xs:element name="Provider" type="providerType" minOccurs="0"/>
                <xs:element name="PaidObligations" minOccurs="0">
                  <xs:complexType>
                    <xs:sequence>
```

```

        <xs:element name="PaidObligation" minOccurs="0"
maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:choice minOccurs="0">
                        <xs:element name="Reason"
type="paidObligationAddReasonEnum"/>
                        <xs:element name="CustomReason"
type="nonEmptyString32"/>
                    </xs:choice>
                    <xs:element name="PaidDebt" type="moneyType"/>
                    <xs:element name="TotalDebt" type="moneyType"/>
                    <xs:element name="DueDate" type="xs:dateTime"/>
                    <xs:element name="PaidDate" type="xs:dateTime"/>
                    <xs:element name="IsSumOfInstallments" type="xs:boolean"
minOccurs="0"/>
                    <xs:element name="TotalDebtPLN" type="decimalType"
minOccurs="0"/>
                    <xs:element name="PaidDebtPLN" type="decimalType"
minOccurs="0"/>
                    <xs:element name="Category" type="xs:int" minOccurs="0"/>
                    <xs:element name="Subcategory" type="xs:int" minOccurs="0"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DocumentInformations" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="DocumentInformation" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Provider" type="providerType" minOccurs="0"/>
                        <xs:element name="Document" type="documentNumberType"/>
                        <xs:element name="IssueDate" type="xs:date" minOccurs="0"/>
                        <xs:element name="FirstName" type="nonEmptyString32"/>
                        <xs:element name="SecondName" type="nonEmptyString32" minOccurs="0"/>
                        <xs:element name="Surname" type="nonEmptyString64"/>
                        <xs:element name="DrawerName" type="nonEmptyString128" minOccurs="0"/>
                        <xs:element name="DrawerOffice" type="nonEmptyString1024" minOccurs="0"/>
                        <xs:element name="Circumstances" type="nonEmptyString1024" minOccurs="0"/>
                        <xs:element name="Validator" type="nonEmptyString128" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

This type contains a full KRD report of the economic information located in the system according to indicated search criteria.

Additional information for the *category* attribute / field (specifies the identification number of the obligation title category) and *subcategory* (specifies the identification number of the obligation title sub-category):

The attributes / fields contains specific numerical values for which a dictionary has been developed, which are available at the Client's request. Values in these attributes / fields appear after enabling the permissions / dictionary, according to business arrangements.

3.4.3.33. The main Output element

```

<xs:element name="Output">
  <xs:complexType>
    <xs:choice>
      <xs:element name="FatalError" type="errorType"/>
      <xs:sequence>
        <xs:element name="InformationManagement" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Order" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:complexContent>
                    <xs:extension base="orderResponseType">
                      <xs:choice>
                        <xs:element name="AddInformation"
type="informationResponseType"/>
                        <xs:element name="UpdateInformation"
type="informationResponseType"/>
                        <xs:element name="RemoveInformation"
type="informationResponseType"/>
                        <xs:element name="SuspendInformation"
type="informationResponseType"/>
                        <xs:element name="UnsuspendInformation"
type="informationResponseType"/>
                        <xs:element name="GetInformation">
                          <xs:complexType>
                            <xs:choice>
                              <xs:element name="Error"
type="errorType"/>
                              <xs:element name="Case"
type="responseCaseType"/>
                              <xs:element name="Obligation"
type="responseObligationType"/>
                              <xs:element name="AlimonyObligation"
type="responseAlimonyObligationType"/>
                              <xs:element name="ExecutiveObligation"
type="responseExecutiveObligationType"/>
                              <xs:element name="PaidObligationCase"
type="responsePaidObligationCaseType"/>
                              <xs:element name="PaidObligation"
type="responsePaidObligationType"/>
                            </xs:choice>
                          </xs:complexType>
                        </xs:element>
                      <xs:element name="GetInformations">
                        <xs:complexType>
                          <xs:choice>
                            <xs:element name="Error" type="errorType"/>
                            <xs:element name="Cases">
                              <xs:complexType>
                                <xs:complexContent>
                                  <xs:extension base="getCasesType">
                                    <xs:sequence>
                                      <xs:element name="Result">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Case"
type="responseCaseType" minOccurs="0" maxOccurs="unbounded"/>
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                    </xs:sequence>
                                  </xs:extension>
                                </xs:complexContent>
                              </xs:complexType>
                            </xs:element>
                          </xs:choice>
                        </xs:complexType>
                      </xs:element>
                    </xs:extension>
                  </xs:complexContent>
                </xs:complexType>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:choice>
    </xs:complexType>
  </xs:element>

```

```

        </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="pagerAttributeGroup"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="PaidObligationCases">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="getPaidObligationCasesType">
                <xs:sequence>
                    <xs:element name="Result">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Case"
type="responsePaidObligationCaseType" minOccurs="0" maxOccurs="unbounded"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attributeGroup ref="pagerAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="SendNotification"
type="simpleResponseInformationType"/>
<xs:element name="ChangeInformationOwner">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="simpleResponseInformationType">
                <xs:sequence>
                    <xs:element name="Case"
type="responseCaseType" minOccurs="0"/>
                </xs:sequence>
                <xs:attribute name="loginName"
type="nonEmptyString128" use="required"/>
                <xs:attribute ref="verifyResult" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="ProcessSnapshot">
    <xs:complexType>
        <xs:choice>
            <xs:element name="Error" type="errorType"/>
            <xs:element name="InformationManagementOrders">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Order" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:complexContent>
                                    <xs:extension base="orderBaseType">
                                        <xs:choice>
                                            <xs:element name="AddInformation">
                                                <xs:complexType>
                                                    <xs:sequence>
                                                        <xs:choice>
                                                            <xs:element name="Case"
type="addCaseType"/>
                                                            <xs:element name="PaidObligationCase"

```

```

type="addPaidObligationCaseType">
    </xs:choice>
    </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="UpdateInformation">
    <xs:complexType>
        <xs:sequence>
            <xs:choice>
                <xs:element name="Case"
type="updateCaseType">
                    </xs:choice>
                <xs:element name="PaidObligationCase"
type="updatePaidObligationCaseType">
                    </xs:choice>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="RemoveInformation"
type="entityManagementWithVerificationType">
        </xs:choice>
        </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="GetInformationsVerificationEvents">
    <xs:complexType>
        <xs:choice>
            <xs:element name="Error" type="errorType">
            <xs:element name="Result"
type="getInformationsVerificationEventsResult">
                </xs:choice>
            </xs:complexType>
        </xs:element>
    </xs:choice>
    </xs:extension>
    </xs:complexContent>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="MonitorConditionManagement" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Order" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:complexContent>
                        <xs:extension base="orderResponseType">
                            <xs:choice>
                                <xs:element name="AddMonitorCondition"
type="monitorManagementResponseType">
                                    </xs:choice>
                                <xs:element name="UpdateMonitorCondition"
type="monitorManagementResponseType">
                                    </xs:choice>
                                <xs:element name="StopMonitorCondition"
type="monitorManagementResponseType">
                                    </xs:choice>
                                <xs:element name="GetEvents">
                                    <xs:complexType>
                                        <xs:choice>
                                            <xs:element name="Error" type="errorType">
                                            <xs:element name="MonitorEvents">

```

```

        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="getMonitorEventsType">
              <xs:sequence>
                <xs:element name="Result">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="MonitorEvent"
type="monitorEventResponseType" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
              <xs:attributeGroup ref="pagerAttributeGroup"/>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="GetMonitorConditions">
  <xs:complexType>
    <xs:choice>
      <xs:element name="Error" type="errorType"/>
      <xs:element name="MonitorConditons">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="getMonitorConditionsType">
              <xs:sequence>
                <xs:element name="Result">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="MonitorCondition"
type="monitorConditionResponseType" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
              <xs:attributeGroup ref="pagerAttributeGroup"/>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="SearchManagement" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Order" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="orderResponseType">
              <xs:choice>
                <xs:element name="SearchNonConsumer"
type="disclosureReportResponseType"/>
              </xs:choice>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        <xs:element name="SearchConsumer"
type="disclosureReportResponseType"/>
        <xs:element name="SearchDocument"
type="disclosureReportResponseType"/>
        <xs:element name="SearchMe"
type="disclosureReportResponseType"/>
        <xs:element name="GetDisclosureReports"
type="disclosureReportsResponseType"/>
        <xs:element name="GetDisclosureReport"
type="disclosureReportResponseType"/>
    </xs:choice>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="IncognitoSearchManagement" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Order" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:complexContent>
                        <xs:extension base="orderResponseType">
                            <xs:choice>
                                <xs:element name="IncognitoSearchConsumer"
type="disclosureReportResponseType" />
                                <xs:element name="IncognitoSearchNonConsumer"
type="disclosureReportResponseType" />
                                <xs:element name="IncognitoSearchRegistryReportConsumer"
type="genericDisclosureReportResponseType" />
                                <xs:element name="IncognitoSearchRegistryReportNonConsumer"
type="genericDisclosureReportResponseType" />
                            </xs:choice>
                        </xs:extension>
                    </xs:complexContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:sequence>
</xs:choice>
<xs:attributeGroup ref="inputAttributes"/>
<xs:attribute name="started" type="xs:dateTime" use="required"/>
<xs:attribute name="processed" type="xs:dateTime" use="required"/>
<xs:attribute name="orderCount" type="xs:int" use="required"/>
<xs:attribute name="successCount" type="xs:int" use="required"/>
<xs:attribute name="failCount" type="xs:int" use="required"/>
</xs:complexType>
</xs:element>

```

An element of the *Output* type is located in the report file sent to the Client, and its content is used to provide information about fulfilling or not the submitted requests for services.

The element contains a group of attributes describing the input parameters and the performed operations' statistics.

In case of an error blocking the start of processing the request, the component contains the *FatalError* element describing the error which occurred during the request processing.

If the request processing has started, the *Output* component, instead of the *FatalError* element, contains the sequence of elements defining the particular

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

types of requests, any of which can but does not have to occur. These elements are:

- *InformationManagement* – containing the data of the requests for managing economic information,
- *MonitorConditionManagement* – containing the data of the requests for managing monitoring conditions (applies to monitored Tax ID's),
- *SearchManagement* – containing the data of the requests for economic information searches (KRD reports),
- *IncognitoSearchManagement* – containing the data of the requests for economic information searches (KRD report) and information from inquiry register (requests history) in incognito mode.

The *InformationManagement* element can contain:

- *AddInformation* – when the request referred to adding economic information,
- *UpdateInformation* – when the request referred to updating economic information,
- *RemoveInformation* – when the request referred to deleting economic information,
- *SuspendInformation* – when the request referred to suspending negative economic information,
- *UnsuspendInformation* – when the request referred to restoring negative economic information,
- *GetInformation* – when the request referred to downloading one economic information (from own account),
- *GetInformations* – when the request referred to downloading economic information (from own account),
- *SendNotification* – when the request referred to sending a notification about adding the debtor to the KRD (for information – currently we do not send letters abroad),
- *ChangeInformationOwner* – when the request referred to changing the owner of the economic information. Returns information about the status of a performed request for changing the owner of the information along with the data of economic information, if verification was requested,
- *ProcessSnapshot* – when the request referred to the current status of the Client's account. Returns a list of adding, updating and deleting (removing) cases requests as a result of the comparison of the sent "snapshot" with the current state of the system; the returned data can be used to equalize the state of the database to the previously ordered "snapshot",
- *GetInformationsVerificationEvents* – when the request referred to downloading information about economic negative information verification. Returns a list of events generated by the system during verification of the debtors' data; the elements contain status of the event with values "*Blocked*" (blocked), "*Doubtful*" (doubtful), description of the doubt and information identifiers.

The *MonitorConditionManagement* element (applies to Tax ID monitoring) can contain:

Nicci 3.2	Version of the documentation: 1.7
Nicci 3.2 asynchronous protocol technical specification	Date: 2024-05-10

- *AddMonitorCondition* – when the request referred to adding a monitoring condition,
- *UpdateMonitorCondition* – when the request referred to updating a monitoring condition,
- *StopMonitorCondition* – when the request referred to stopping (removing) a monitoring condition,
- *GetEvents* – when the request concerned the verification of events monitoring conditions. Returns the list of events described *MonitorEvent* element containing monitored identification number (Tax ID), date of event and the event type described by *monitorEventTypeEnum*. If the download events fail, the *Error* element will contain the type of problem,
- *GetMonitorConditions* – when the request referred to downloading the monitoring conditions.

The *SearchManagement* element can contain:

- *SearchNonConsumer* – when the search (downloading the KRD report) referred to an entity different than a consumer,
- *SearchConsumer* – when the search (downloading the KRD report) referred to a consumer,
- *SearchDocument* – when the search (downloading the KRD report) referred to a forged / someone else's document,
- *SearchMe* – when the search (downloading the KRD report) referred to the searcher,
- *GetDisclosureReports* – when the search referred to downloading KRD reports with economic information (it's about redownloading already downloaded reports),
- *GetDisclosureReport* – when the search referred to downloading KRD report with economic information (it's about redownloading already downloaded report).

The *IncognitoSearchManagement* element can contain:

- *IncognitoSearchConsumer* – if the purpose of the search (KRD report) in incognito mode is the disclosure of economic information about a consumer,
- *IncognitoSearchNonConsumer* – if the purpose of the search (KRD report) in incognito mode is the disclosure of economic information about a business entity,
- *IncognitoSearchRegistryReportConsumer* – if the purpose of the search in incognito mode is the disclosure of information from a consumer inquiry register (requests history),
- *IncognitoSearchRegistryReportNonConsumer* – if the purpose of the search in incognito mode is the disclosure of information from a business entity inquiry register (requests history).

END OF DOCUMENT