enterprise SOA Object & Service Operation Design

Part IV of VI

Global Data Type (GDT) Design Guideline

Version 2.7
February 27th, 2009
## Authors

<table>
<thead>
<tr>
<th>Name</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seubert, Michael</td>
<td>Author</td>
</tr>
<tr>
<td>Richsteiger, Dirk</td>
<td>Author</td>
</tr>
<tr>
<td>Brossler, Andreas</td>
<td>Contributing Author</td>
</tr>
<tr>
<td>Krebs, Friedhelm</td>
<td>Contributing Author</td>
</tr>
</tbody>
</table>
# Table of Contents

## 1 Global Data Type (GDT) Design

Guide to Readers ......................................................... 5

### 1 Introduction .......................................................... 7

### 2 Global Data Types ..................................................... 9

#### 2.1 Basics ........................................................................ 9

#### 2.2 Typing of Business Semantic ...................................... 10

#### 2.3 Meta Model of Data Types ......................................... 12

#### 2.4 “Build-up” of Global Data Types ................................. 17

#### 2.5 Derivation of Context-specific Data Types ..................... 17

#### 2.5.1 Projection ............................................................. 18

#### 2.5.2 Context ............................................................... 18

#### 2.6 Naming ..................................................................... 19

#### 2.7 Documentation Pattern .............................................. 21

#### 2.8 Global Data Type Taxonomy ....................................... 24

#### 2.8.1 Taxonomy According to ISO 11179’s Object Class Term ......................................................... 24

#### 2.8.2 Taxonomy According to Context ............................... 25

### 3 Global Data Type Modeling ........................................... 27

#### 3.1 Design Steps for Global Data Types ............................ 27

#### 3.2 GDT Determination .................................................. 28

#### 3.3 GDT Definition ....................................................... 29

#### 3.4 GDT Name .............................................................. 30

#### 3.4.1 Naming Rules for basic GDTs, Elements, and Attributes ......................................................... 33

#### 3.4.2 Naming Rules for aggregated GDTs .......................... 39

#### 3.5 GDT Structure ......................................................... 44

#### 3.5.1 Object Class, Property, and Representation ................. 45

#### 3.5.2 Core Data Type .................................................. 47

#### 3.5.3 Basic Global Data Type ........................................ 47

#### 3.5.4 Aggregated Global Data Type ................................. 49

#### 3.5.5 Attribute ........................................................... 51

#### 3.5.6 Template and Projection ........................................ 54

#### 3.6 GDT Value Range, Integrity Conditions, Use, and Notes ......................................................... 60

#### 3.7 GDT Appendix ......................................................... 61
3.8 Changes on Existing GDTs ................................................................. 62
3.8.1 Compatibility of Changes ................................................................. 62
3.8.2 Incompatible Changes Force new GDT Versions ......................... 63
3.9 Versioning of GDTs ........................................................................... 64

4 Additional Concepts for Global Data Types ........................................ 65
4.1 Attributes.............................................................................................. 65
4.2 Restrictions on CDTs / GDTs ............................................................... 66
4.2.1 Semantic Restriction – Concepts ....................................................... 67
4.2.2 Structural Restriction – Variable Concept ........................................... 67
4.2.3 Local Restriction (exceptional) ......................................................... 69
4.3 Code Lists ............................................................................................ 70
4.3.1 Basics .............................................................................................. 70
4.3.2 Implementation ................................................................................ 72

5 Architecture Data Types ....................................................................... 75
5.1 GDTs for Object Model Meta Structure and Object Identification .... 78
5.1.1 ObjectTypeCode and ObjectNodeTypeCode in Backend ................... 81
5.1.2 Applying for new ObjectTypeCodes .................................................. 81
5.1.3 Applying for new ObjectNodeTypeCodes .......................................... 81
5.2 Measure and Quantity ......................................................................... 82

6 Quick Reference to Global Data Type Modeling .................................... 83
6.1 Basics ................................................................................................ 83
6.1.1 GDT Determination ......................................................................... 84
6.1.2 GDT Definition ................................................................................ 84
6.1.3 GDT Name ....................................................................................... 85
6.1.4 GDT Structure ................................................................................ 86
6.1.5 GDT Value Range ........................................................................... 90
6.1.6 GDT Integrity Conditions ................................................................. 90
6.1.7 GDT Use and Notes ....................................................................... 91
6.1.8 Code list .......................................................................................... 91
6.1.9 Qualifier list ..................................................................................... 91
6.2 Supporting Resources for Creating GDTs ........................................... 92
6.3 Applying for new GDTs (or Changes to Existing GDTs) ..................... 92

7 Index .................................................................................................... 93

8 Change History ..................................................................................... 94
Guide to Readers

The following part describes the basics to the design of Global Data Types. GDTs are introduced with their definition and placement within the meta model of business objects and service operations.

In the subsequent chapters a detailed description of the basic design rules for GDTs is given. These chapters are followed by a section on several selected topics regarding the design of GDTs and an overview of the architecture data types.

This part concludes with a quick guide to the creation of a GDT that gives an overview on the main steps necessary to model a data type.
1 Introduction

The semantic integration of components is an essential key differentiator in the competition with other software providers. Along with the tools required for this, SAP also provides the business content (process integration content) necessary for the communication between systems and their components. This content is designed in line with international standards. This outside-in approach ensures that SAP (“inside”) is speaking the language of the business world (“outside”).

The basis for this is an integrated Business Object Model (iBOM) that is harmonized across industries and business areas. The iBOM describes the business-relevant concepts in one central location. This means that it reflects all the design decisions that were made during the modeling of the business entities from the real world. It consists of the individual business objects and their relationships to one another. A business object is a capsule with an inner hierarchical structure, an object behavior specified by its operations, and constraints. Business objects are semantically disjoint, which means that a business information unit is represented exactly once. The internal structure of a business object is represented by hierarchically arranged nodes with their elements. Operations are ultimately derived from the iBOM, and this ensures their overall consistency.

The SAP-wide normed data types play a significant role in the harmonization of the structures of business object nodes in the iBOM across industries and business areas. Data types type the elements of business objects and their operations with a specified structure. A data type represents a specific business-related subject matter, and elements that reflect a particular subject matter are always typed by the same data type. By this harmonization across the business objects is achieved.
2 Global Data Types

2.1 Basics

Global data types (GDTs) are SAP-wide normed and reconciled data types that represent business-related subject matters as they occur in standards or would correspondingly be defined there. By them harmonization of the hierarchical signature structures of operations and the structure of the business object nodes is achieved.

The basic idea of uniform typing and hence harmonization is expressed by the following key sentence:

"… the same subject matter is always described (typed) by the same data type …“ (uniform typing)

Global data types represent a business-related subject matter that is described by a specified structure. If this semantic subject matter occurs in a business object node or in a B2B- or A2A operation, it is always typed by the same global data type.

This leads to uniform typing across all business objects, interfaces, and operations.

Global data types are, with regard to a subject matter, maximally defined data types that contain all elements required for the subject matter in different contexts.

GDTs can be provided context-specifically and when used in operations (or business objects) with more semantics with regard to a business-related context ("Context Data Types"). The elements used can also be restricted and integrity constraints made stricter. Thus, based on a maximal GDT further data types can be projected as context-specific restrictions.

Global data types can be defined as follows:

Global Data Types (GDTs)

... are reusable semantic building blocks for service operations and business objects

... are based on the rules described in the international standard UN/CEFACT CCTS (Core Component Technical Specification)

... have been approved SAP-wide by the Governance Process for Business Content (embedded in the SAP standard ‘Application Integration & Interfaces’)

... have been defined in the ESI Repository and are described by XML schema

... have been documented in accordance with the documentation templates
2.2 Typing of Business Semantic

The hierarchical structure of the SAP Business Objects and B2B / A2A operations is always described at least at the lowest level (“leaf” elements) by a GDT. That means that sometimes aggregated GDTs can be used to type more complex subject matters. These data types in turn may also have a hierarchical structure (see Figure 1).

![Hierarchical Node Structure with Global Data Types](image)

**Figure 1: Hierarchical Node Structure with Global Data Types**

This leads to a cross-component harmonization of the Business Objects, i.e. their node signature, and of the derived service operations, because these are now - according to the construction kit principle (“Lego bricks”) – typed by standardized building blocks.

By doing this, the same subject matter is always typed identically (that is, by the same data type or by a view of this type).

GDTs do, according to their nature, carry business semantics, but are defined usage-neutral.

By using them (in typing of business object nodes and service operation signatures) they get their usage-specific semantics (see Figure 2).

There are both elementary (basic) and composed (aggregate) data types.
Figure 2: Usage-Specific Composition of Operations from GDTs
2.3 Meta Model of Data Types

The business content is described by a conceptual mode which shows the relation of the various entities: The business objects with their nodes and the assigned service interfaces / operations with their related message types. Their signatures are then typed by GDTs.

![Meta Model of Data Types Diagram](image)

Business Object Nodes and Message Types expose complex business-related subject matters. This business-related subject matter is semantically structured by elements. Hence, each element represents a fraction of that business-related subject matter.

Business Object Nodes and Message Types are each typed by their respective entity-specific data types: Node Data Types and Message Data Types. These data types provide the inner element structure for the Business Object Nodes and Message Types and therefore are designed for a specific use.

As mentioned previously already: there is always a business-related subject matter that is typed by a data type. In other words, there is always an element which represents the business semantic and a data type which gives the element its technical features.
SOA Object & Service Operation Design

Figure 4: Elements are typed by Data Types that in turn may contain elements

Of course, a data type can represent itself a more complex business-related subject matter. Then the data type is described by elements (that structure the business-related subject matter).

Figure 5: Meta Model of Usage-neutral Data Types

Data types can contain elements or attributes:

- An element represents a business-related subject matter. Its technical representation (for example: value ranges, length) are specified by data types that type each element. The data type in turn can contain further elements.

- Attributes define an elementary feature of a data type that is there is a strong dependency between an attribute and its data type. The attribute’s value is needed to interpret the data type’s value. A change of the attribute’s value affects directly the value of the data type. An attribute is typed by a data type (either a GDT or XSD) that specifies the representation of data.
There are entity-specific and global data types.

Entity-specific data types have always a specific business semantic and are built with regard to their usage in Business Objects and Service Operations (see Guideline, Part II and III). They group the elements of Business Object Nodes and Message Types.
Entity-specific data types can only type entity specific elements, but never elements of different entities.
These data types rely on CDTs and Basic / Aggregated GDTs that means their entity specific elements at the deepest level of their inner element hierarchy are typed by either a CDT or Basic / Aggregated GDT.

Entity-specific data types are distinguished in

- **Intermediate Data Type (IDT)**
  An intermediate data type is a grouping of elements that have a strong semantic relationship. Intermediate data types are categorized into
  - **Message IDT (MIDT)**
    An MIDT is a grouping of message elements. It is used to type elements of messages.
  - **Form Message IDT (FMIDT)**
    A FMIDT is a grouping of form message elements. It is used to type elements of form messages.
  - **Business Object IDT (BOIDT)**
    A BOIDT is a grouping of BO node elements. It is used to type elements of Business Object Nodes.
  - **Query IDT (QIDT)**
    A QIDT is a grouping of query elements. It is used to type elements of queries.

- **Node Data Type (NDT)**
  A node data type is a grouping of all elements of a node and provides the typing for them.

- **Message Data Type (MDT)**
  A message data type is a grouping of all elements of a message type and provides the typing for them.

- **Key Data Type (KDT)**
  A key data type is a grouping of elements that make up a key of a Business Object (Node) and provides the typing for them.

- **Query Data Type (QDT)**
  A query data type is a grouping of elements that make up the signature of a core query and provides the typing for them.

- **Action Data Type (ADT)**
  An action data type is a grouping of elements that make up the signature of an action and provides the typing for them.

- **Filter Data Type (FDT)**
  A filter data type is a grouping of elements that are parameters for a filter on a relationship and provides the typing for them.

Global data types are SAP-wide normed and reconciled data types. GDTs carry business semantics; they are defined usage neutral and can be used globally. They are distinguished in Basic GDTs and Aggregated GDTs:

- **Core Data Type**
  Core data types (CDT) are internationally standardized data types (refer to the UN/CEFACT Core Component Technical Specification (CCTS)), based on W3C data types without any business semantic. They may have attributes. Attributes of CDTs are called Supplementary Components (example:
“currencyCode” at CDT “Amount”). The CDTs implemented at SAP may have structural restrictions (for example number of digits at Quantity and Amount).

**Basic Global Data Type**

A basic global data type (BGDT) is derived directly from a CDT or a basic GDT. Hence, each basic GDT is derived – at least indirect – from a CDT. A basic GDT may have Supplementary Components, but no elements.

A basic GDT that can be used for typing elements of messages only (due to its semantic) is called **Message BGDT (MBGDT)**.

A basic GDT represents a simple business-related subject matter.

**Aggregated Global Data Type**

An aggregated global data type (AGDT) always contains elements. An aggregated GDT may have attributes.

An aggregated GDT that can be used for typing elements of messages only (due to its specific structure or semantic) is called **Message AGDT (MAGDT)**.

An aggregated GDT represents a complex business-related subject matter.
2.4 “Build-up” of Global Data Types

All data types are either a restriction (based on) or an aggregation of other data types. The basis for all GDTs are the CDTs provided by the UN/CEFACT CCTS.

Directly built upon the CDTs are the basic GDTs. A basic GDT represents a simple business-related business subject matter and its structure is limited to the maximum structure as defined at the CDT it is built upon. Therefore the structure of a GDT built upon a CDT is always a restriction on the CDT’s structure. The restriction may be on the length, the value ranges, or the omission of Supplementary Components.

A basic GDT may base on another basic GDT. In this case the business-related subject matter of the basic GDT must be a semantic refinement of the basic GDT it bases on. Its structure must be a restriction of the one defined by the underlying GDT.

More complex business-related subject matters are represented by aggregated GDTs. They are described by elements. Their elements may have a simple or again a complex business semantic. Hence, they are typed by CDTs or GDT which may either be basic or aggregated. By this very complex business-related subject matters can be represented and structured by aggregated GDTs with elements that are typed in turn by aggregated GDTs.

An aggregated GDT may base on another aggregated GDT. In this case the business-related subject matter of the aggregated GDT must be a semantic refinement of the aggregated GDT it bases on. Its structure must be a restriction of the one defined by the base GDT. The restriction may include the omission of elements and attributes or strengthening of their cardinalities.

Classification of Global Data Types

GDTs can be classified according to criteria like representation, functional area, corresponding business-related subject matters.

The classification of GDTs is currently in preparation.

2.5 Derivation of Context-specific Data Types

The GDTs represents the maximum business matter of fact which can be covered. Each context is therefore a view onto this GDT (Basic or Aggregated GDT) and a reduction of the business-related subject matter according to specific context dimensions. Context-specific data types can be derived from a maximum set of GDTs. The derivation is realized by projection, that is, a GDTs structure may get restricted, but additions are not permitted.

![Figure 7: Global Data Types and Context-Specific Derivation](image-url)
Business objects and service operations have always a specific business semantic and are built with regard to a usage. GDTs are used to type elements of Business Objects and service operations. Hence, GDTs are used in Business Objects and Service Operations in a specific context.

Within a specific context a GDT’s semantic and structure may be restricted. For this a context-specific data type is derived from the GDT by means of a projection.

The projection of GDTs ensures a harmonization of the context-specific derived data types. Every context-specific derived data type is semantically a specialization and its structure a subset of the GDT it is projected from.

Regarding GDTs, distinctions must be made between the following aspects:

- **Definition** of the GDT
- **Context-specific "refinement"** of the GDT (view forming / derivation (projection) )
- **Usage** of the GDT in operations or objects (assembly)

### 2.5.1 Projection

A projection of a data type forms a view on a data type by

- specialization of the business-related subject matter
- omitting elements and attributes
- limiting cardinalities of elements and attributes
- restricting value ranges and lengths

A valid projection of a GDT is specified for a specific context.

### 2.5.2 Context

Context is divided into eight context categories ("context dimensions") according to the UN/CEFACT CCTS:

- Business Process
- Product
- Industry
- Geopolitical
- Official Constraints
- Business Process Role
- Supporting Role
- System Capabilities

A specific context is defined by a combination of context category instances.
2.6 Naming

GDTs type the properties (= elements) of business objects and their operations with a specified structure. A GDT represents a specific business-related subject matter, and properties that reflect a particular subject are always typed by the same data type.

GDTs are always defined and unique in their semantics in the context of an object. They represent ("type") either properties of an object or the object itself.

Each property has a standardized representation which is based on standardized core data types.

GDTs are given semantically meaningful names. The name is derived from the definition of the business-related subject matter the GDT represents. The business-related subject matter can either have an object-like quality that means it is described by further properties, or it has just a quality of a property of an object. In rare cases the business-related subject matter is a generic property that means it can occur in different objects as a simple property.

From the definition of the business-related subject matter the terms representing an object, a property, and the representation are identified. These terms are used to form the name of a GDT.

This naming follows the naming rules specified in the ebXML CCTS, which are based on ISO standard 11179 (5).

An ISO 11179 data element is made up of three parts:

- The "object class": A set of concepts, abstractions or things in the real world which can be identified within clear boundaries and meanings, and whose characteristics and behavior follow the same rules (examples: automobile, person, household, order ...).

- The "property": A characteristic feature shared by all the instances of an object class (examples: color, age, income, address ...).

- The "representation": Describes how the data is represented, meaning the data type and its value range (examples: a date can be represented with xsd:date or xsd:datetime).

Each of these three name components can be described in more detail by a qualifier.
For example: **BusinessTransactionDocumentTypeCode**

**ObjectClass**: BusinessTransactionDocument; **Property**: Type; **Representation**: Code

---

**Figure 8: Naming Rules in Accordance with ISO 11179**

<table>
<thead>
<tr>
<th>Object Class</th>
<th>Property</th>
<th>Representation/Association</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Qualifier</strong></td>
<td><strong>Term</strong></td>
<td><strong>Qualifier</strong></td>
</tr>
<tr>
<td>Business Transaction Document</td>
<td></td>
<td>Type</td>
</tr>
</tbody>
</table>

---
2.7 Documentation Pattern

A GDT specification is documented in a particular format. The format is specified by a pattern. This ensures a uniform documentation structure for all GDTs, that is, definition, name, structure, and further specifications of a GDT are always documented in the same manner and order. This documentation pattern allows to clearly arranging the GDTs in catalog form. You can browse the GDT catalog in Corporate Portal.

The GDT documentation pattern is divided into the following sections:

1.1 <GDT Name> (header)
1.1.1 Status / Owner
1.1.2 Definition
1.1.2.1 Comment
1.1.2.2 Dictionary Entry Name
1.1.3 Example (Instance)
1.1.4 Structure
1.1.5 Detailed Description and Value Ranges
1.1.6 Integrity Conditions
1.1.7 Use
1.1.8 Notes
1.1.9 Appendix – Code Lists
1.1.10 Appendix – Qualifier List

Status / Owner section
This is internal organizational information for organizing the PIC governance process.

Definition section
This section contains the definition of the GDT (see chapter 3.3).
It has two sub-sections:

- Comment – contains definitions of terms used in the GDT’s definition
- Dictionary Entry Name – a formal representation of the GDT name components according to ISO 11179 (see chapter 3.4).

Example (Instance) section
This section gives an example instance of the GDT in XML code.

---

1 https://portal.wdf.sap.corp/irj/servlet/prt/portal/prtroot/com.sap.km.cm.docs/guid/f0783985-4a5b-2810-4c9c-b8df0cdafe0
Structure section
This section contains the structure definition of the GDT in table form (see chapter 3.5).

Detailed Description and Value Ranges section
This section describes the attributes, elements and other details of the GDT if necessary. The valid value ranges are also defined here (see chapter 3.6).

Integrity conditions section
This section describes the integrity conditions of the GDT (see chapter 3.6).

Use section
This section describes the common use of a GDT (see chapter 3.6).

Notes section
This section contains general information that is not to be listed in the other sections (see chapter 3.6).

Code Lists section
For code GDTs this section describes the valid code lists if code list content is supplied (see chapter 4.3).

Qualifier List section
This section contains the list of allowed qualifiers for a CDT or GDT.

Special GDT Documentation Patterns
Special documentation patterns have been created for the most common types of GDTs or qualified CDTs / GDTs.

• Identifier
• Indicator
• Name
• Description
• Code

The patterns are available on the Corporate Portal:


The GDT documentation is created in accordance with one of the patterns provided.
2.8 Global Data Type Taxonomy

The GDT taxonomy is a classification of all GDTs according to certain criteria. The GDT taxonomy ensures an overall integrity of all GDTs and enables a guided lookup for existing GDTs.

Currently the GDT taxonomy is set up according to

- The generalized object terms of the GDTs’ ISO 11179 names (“Dictionary Entry Name”)
- The context a GDT is valid in

2.8.1 Taxonomy According to ISO 11179's Object Class Term

Each GDT has to have an ISO 11179 compliant name made up of object class term, property term, and representation term (see chapter 2.6).

The object is a set of concepts, abstractions, or things in the real world; hence it is a central part of the modeling methodology and therefore a good basis for GDT taxonomy.

GDTs with the same generalized object class have similar semantics and can be classified in the same way. According to this the classification is done in the following steps:

1. For all GDTs the object class term of a GDT is analyzed for its generalization
2. A list of generalized object class terms is the result
3. Each GDT gets assigned to the appropriate generalized object class term

Object Class Terms
- Purchase Order
- Sales Order
- Production Order
- Engineering Change Order
- Maintenance Order

Generalized Object Class Term: “Order”

<table>
<thead>
<tr>
<th>Generalized Object Class Term</th>
<th>Assigned Global Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order</td>
<td>AnalyticalViewOfTradingOrderID</td>
</tr>
<tr>
<td></td>
<td>BankPaymentOrderLifeCycleStatusCode</td>
</tr>
<tr>
<td></td>
<td>ClearingHousePaymentOrderLifeCycleStatusCode</td>
</tr>
<tr>
<td></td>
<td>EngineeringChangeOrderID</td>
</tr>
<tr>
<td></td>
<td>EngineeringChangeOrderLifeCycleStatusCode</td>
</tr>
<tr>
<td></td>
<td>EngineeringChangeOrderTypeCode</td>
</tr>
<tr>
<td></td>
<td>EngineeringDesignChangeOrderID</td>
</tr>
<tr>
<td></td>
<td>OrderRejectionReasonCode</td>
</tr>
<tr>
<td></td>
<td>PaymentOrderLifeCycleStatusCode</td>
</tr>
<tr>
<td></td>
<td>PaymentOrderRejectionReasonCode</td>
</tr>
<tr>
<td></td>
<td>ProcurementPlanningOrderID</td>
</tr>
<tr>
<td></td>
<td>ProcurementPlanningOrderLifeCycleStatusID</td>
</tr>
<tr>
<td></td>
<td>ProductionPlanningOrderLifeCycleStatusID</td>
</tr>
<tr>
<td></td>
<td>ProductionPlanningOrderMaterialDataOriginTypeCode</td>
</tr>
<tr>
<td></td>
<td>PurchaseOrderDeliveryStatusID</td>
</tr>
<tr>
<td></td>
<td>PurchaseOrderLifeCycleStatusCode</td>
</tr>
<tr>
<td></td>
<td>StockTransportPlanningOrderID</td>
</tr>
</tbody>
</table>

Figure 9: Example of GDT classification according to generalized object class term
2.8.2 Taxonomy According to Context

A GDT has specific semantics. These semantics may not be valid in every context. A context is specified by context dimensions. In the context based GDT taxonomy the GDTs are classified according to these context dimensions.

Currently the context dimensions are the ones defined by UN/CEFACT ebXML CCTS (see chapter 2.5.2) and additional dimension defined by SAP:

- Architectural Concepts
- Processing
- Elementary Properties
- Planning, Execution & Simulation
- Price, Tax & Financial Valuation
- Business Functional Area
- Technical Infrastructure

Each context dimension is divided up into disjoint scales. A vector of scale values from the dimensions define a specific context to which GDTs get assigned.

![Figure 10: Assignment of GDTs to a context](image)

Each GDT and qualified GDT is classified in this manner. The classification is not integrated into the development tools, but realized with an Excel-based tool.

<table>
<thead>
<tr>
<th>Context Specification</th>
<th>Assigned GDTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geopolitical Context = &quot;USA&quot;</td>
<td>EarnedIncomeCreditEmployeeTaxationMaritalStatusCode</td>
</tr>
<tr>
<td>Legal Constraints Context = &quot;Employee Taxation&quot;</td>
<td>MedicareEmployeeTaxationExemptionMethodCode</td>
</tr>
<tr>
<td>Price And Tax And Financial Valuation = &quot;Tax&quot;</td>
<td></td>
</tr>
<tr>
<td>Business Functional Area = &quot;Human Capital Management&quot;</td>
<td></td>
</tr>
<tr>
<td>Business Process Context = &quot;Payment Processing Of Incoming Cheque With Lockbox&quot;</td>
<td>ChequeID ChequeStoragePartyID LockboxBatchID LockboxBatchItemID</td>
</tr>
</tbody>
</table>

![Figure 11: Example of classified GDTs for a specific context](image)
3 Global Data Type Modeling

3.1 Design Steps for Global Data Types

1. GDT Determination
   - Business subject and use case
   - Lookup of a “suitable” GDT and qualifier from the GDT catalog

2. GDT Definition
   - Standard (ISO 11179) and guideline compliance
   - Business standard and pattern compliance

3. GDT Name
   - Derivation of GDT name from the definition
   - Standard (ISO 11179) and guideline compliance

4. GDT Structure
   - Object Class, Property, and Representation
   - Basic GDTs and Aggregated GDTs

5. GDT Value Range
   - Permissible value range for GDT and its elements

6. GDT Integrity Conditions
   - External integrity conditions of GDT
   - Internal integrity conditions of GDT between elements and attributes

7. GDT Use and Notes
   - General use or example use cases
   - Further information about GDT

8. Code List
   - Type of code list
   - Code definition, name, and value

9. Qualifier List
   - Definition of qualified GDT

Figure 12: Steps for designing a GDT

Global Data Types (GDTs) are used to type business object node elements and service operation signature elements. Each GDT represents exactly one business subject in a structured form. The design of a GDT is divided into nine steps. In each step the designs of specific modeling topics have to be specified, for example in step two the definition of the business-related subject matter represented by the GDT. After finishing the last step (the last two steps are optional) a GDT specification is complete. New GDTs have to be approved by the Process Integration Council.
3.2 GDT Determination

New GDTs are created if the business subject is not covered by one of the existing GDTs. The creation of a GDT is motivated by the modeling of business object nodes. A business object node represents a complex business subject. This subject is represented by the elements of the node in a structured manner. Each of these elements must be typed by a GDT.

A subject is represented by exactly one GDT

First the business subject to be represented by the node element is defined, and then the definitions of the approved GDTs in the GDT catalog are scanned to see whether one of these GDTs covers this subject. The GDT catalog is to be found on the Corporate Portal at AP engineering. If you do not find anything there, you check whether there is a suitable GDT in ARIS that is already in the “PIC Governance Process”.

- If a GDT is identified which exactly covers the business subject to be represented, then the node element can be typed with this GDT. If this is the case, the name of the node element corresponds exactly to the name of the typing GDT (however, the name may deviate due to abbreviation and replacement rules; see section 3.4).

- If an existing GDT is identified which represents in a generalized form the business subject to be represented, then the node element can be typed with this GDT, but the semantics of the GDT is restricted in relation to the context. This means that the name of the node element is made up of a semantic qualifier and the GDT name.

For example:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>GDT</th>
<th>1st Qualifier</th>
<th>2nd Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrivalDateTime</td>
<td>DateTime</td>
<td>Arrival</td>
<td></td>
</tr>
<tr>
<td>YardArrivalDateTime</td>
<td>DateTime</td>
<td>Arrival</td>
<td>Yard</td>
</tr>
</tbody>
</table>

Note: If there is also a technical restriction in addition to the semantic one – for example, if a certain standard that is to be supported specifies a different length for a character string – then you must create a new GDT.

Before applying for a new qualifier the qualifier should be checked against the list of all existing qualifiers.

- If no GDT is identified which covers the business subject to be represented, then you must create a new GDT.

[R 1] A new GDT is only created if the business subject to be represented is not already covered by an existing GDT, or if an existing GDT must be specialized, and this specialization includes a technical restriction.

[R 2] All the qualifiers of the first level of a GDT are listed in its documentation with the owner and a description. If the subject is a special subject that can be used again, in exceptional cases the second qualifier can also be included.

---

3 \dwdff029\pic-coaching\06_Global_Data_Types\10_Work_Consolidated\QualifierTracking\Qualifier_Overview.xls
3.3 GDT Definition

The starting point in the creation of a new GDT is the definition of the business subject.

[R 1] The definition must be composed in the singular form.

[R 2] The definition must specify what the business subject is, and not only what it is not.

[R 3] The definition must be formulated in according to the pattern “A(n) …<is / specifies / …> …”.

[R 4] The definition must be described in generally understandable terms.

[R 5] The definition may not contain other definitions of other terms or concepts (only permissible as comments).

[R 6] The definition may not contain any technical implementation details.

[R 7] The definition may not describe the use.
(in accordance with ISO 11179)

[R 8] Start every definition with the super ordinate term to which it belongs.
Use an article at the start of the definition.
If the definition contains a term that requires further explanation, then an explanatory comment may be added.

Examples:

GDT DatePeriod:

“A period limited by two points in time. These points in time are expressed in calendar days. This period is specified by a starting point and a finishing point, or a starting point and a duration, or a duration and a finishing point.”

GDT CostCentreID:

“An identifier for a cost center.

Comment:
A CostCentre is an organizational unit that represents a clearly defined location at which costs arise, and for which costs are entered separately. The definition can be based on functional, accounting, spatial or cost-responsibility factors.”

The definition is entered in the GDT Documentation, under “Definition”.

© 2009 SAP AG
Dietmar-Hopp-Allee 16,
D-69190 Walldorf

Part IV of VI: Global Data Type (GDT) Design
Version 2.7 Date: February 27th, 2009
Authors: Michael Seubert, Dirk Richsteiger

Intellectual Property Rights and
Copyrights protected
3.4 GDT Name

The semantically meaningful name of a GDT is derived from its definition. This naming follows the ISO standard 11176 (5).

![Diagram](image)

**Figure 13: Name derivation according to ISO 11179 (5)**

The GDTs definition is analyzed for main terms. These main terms represent either an object, property, or the representation:

- **The “object class”:** A set of concepts, abstractions or things in the real world which can be identified within clear boundaries and meanings, and whose characteristics and behavior follow the same rules (examples: automobile, person, household, order ...).

- **The “property”:** A characteristic feature shared by all the instances of an object class (examples: color, age, income, address ...).

- **The “representation”:** Describes how the data is represented, meaning the data type and its value range (examples: a date can be represented with xsd:date or xsd:datetime).

Each of these three name components can be described in more detail by a qualifier.

To decide whether there is one object class with several qualifiers or there are several object classes the following rules of thumb can be used:

- If the same instance of an object may be used in different ways there is one object class with qualifiers.

  *Example:* The same instance of a party may be used as “Buyer Party Details” and as “Ship To Party Details” so “Buyer” and “Ship To” are qualifiers of the object class “Party”.

- If different instances of an object may be used in different ways but the way how an instance is used has no impact on the instance itself there is one object class with qualifiers.
Example: Different instances of a price may be used as “Gross Price. Details” and as “Net Price. Details” but the distinction between gross price and net price has no impact on the price itself. So “Gross” and “Net” are qualifiers of the object class “Price”.

- If different instances of an object may have different integrity constraints on how the elements are used but all instances have the same structure there is one object class with qualifiers.

Example: A “Person Address. Details” has different integrity constraints on how the elements are used than a “Company Address. Details” but both addresses have the same structure so “Person” and “Company” are qualifiers of the object class “Address”.

- If different instances of an object have different structures there are several object classes.

Example: A “Purchase Order Item. Details” has a different structure than a “Supplier Invoice Item. Details”, so “Purchase Order Item” and “Supplier Invoice Item” are different object classes (and “Purchase Order” and “Supplier Invoice” are no qualifiers of the object class “Item”).

The semantic name is derived mainly by concatenating

- qualifier(s) of object class term
- object class term
- qualifier(s) of property term
- property term
- representation term

Further shortening rules may apply on the derived semantic name.

Example: BankAccountTypeCode
Definition: A coded representation of a type of a bank account.
Object class term: bank account, Property: type, Representation: code
(the name “BankAccountCode” would be wrong as the definition “is the coded bank account” would be wrong).

[R 1] Commonly accepted business term may be used instead of names directly derived word by word from the definition.

Example:
A Price is the exchange value, expressed in a monetary unit, of a product in relation to a basic amount (and not: A BasicAmountProductExchangeValue is the exchange value, expressed in a monetary unit, of a product in relation to a basic amount).

[R 2] A name consists of one or several words in Oxford British English.
Example:
CatalogueReference (not CatalogReference)

[R 3] Every word starts with an upper case letter except the first word of an attribute name starting
with a lower case letter.

Example:
GDT name “ExchangeRate”
Element name “GivenName”
Attribute name “actionCode”

[R 4] A name uses singular if no plural semantic is explicitly required.

Example:
CompanyLegalFormCode (not CompanyLegalFormCodes)
CreditRiskClassCode (not CreditsRiskClassCode)

The semantic division of the GDT’s name is described in a formal manner by the Dictionary Entry
Name (DEN) following the pattern:

\((([^\(<\text{object class qualifier}\>_\_\_])^*<\text{object class term}>. )?\)
\((([^\(<\text{property qualifier}\>_\_\_])^*<\text{property term}>. )?<\text{representation term}>\)

3.4.1 Naming Rules for basic GDTs, Elements, and Attributes

For basic GDTs, elements and attributes the representation term determined by the amount of valid representation categories (each having a correlated CDT):

- **Amount**
  Amount with currency unit

- **BinaryObject**
  Data flow of random binary-represented characters

- **Code**
  Abbreviated version of a value, a method or a property

- **Date**
  Specification of an exact day in the Gregorian calendar

- **DateTime**
  Time stamp for a calendar day, accurate to the second

- **Duration**
  Period of time of a particular length without a fixed start or end time. This period of time is expressed in years, months, days, hours, minutes, seconds, and fractions of a second

- **Graphic**
  Finite data stream of diagram, graph, mathematical curves, or similar vector based representation in a specific notation, which is expressed in based 64 encoding.

- **Identifier**
  Unique identifier for an object

- **Indicator**
  Binary-coded indicator for a subject (‘0’ / ‘1’ or “true” / “false”)

- **Measure**
  Physical measurement of the related measuring unit

- **Name**
  word or combination of words used to name or define an object

- **Numeric**
  Decimal value

- **Percent**
  number that relates to the comparison figure 100

- **Picture**
  visual representation of a person, object, or scene in binary notation (octets)

- **Quantity**
  Quantity, with the related unit of measurement

- **Ratio**
  quotient of two figures („numerator divided by denominator“)

- **Sound**
  can be used for all kinds of audio files. This includes files such as audio recordings in binary notation (octets).

- **Text**
  Character string with optional language

- **Time**
  Time since begin of day in a 24 hour day.
• Value
  expresses the concept of numeric worth in general.

• Video
  relating to the recording, reproducing or broadcasting of visual images on magnetic tape or digitally in binary notation (octets)

Example:
Name: BankAccountTypeCode  DEN: BankAccount. Type. Code
Name: ChangedIndicator     DEN: Changed. Indicator
Name: TimeZoneDifferenceValue  DEN: Time Zone. Difference. Value

[R 6] For basic GDTs, elements and attributes the object class term is the part of the dictionary entry name describing the object a GDT, element or attribute is related to.

Example:
In the GDT BankStandardID (DEN: Bank. Standard. Identification. Identifier) “Bank” is the object the GDT is related to.

In the element AddressChangedIndicator (DEN: Address. Changed. Indicator) “Address” is the object the element is related to.

Some basic GDTs are not related to a specific object and have no object class term at all. For example the GDT ActionCode (DEN: Action. Code) with “Action” being the property term and “Code” being the representation term is not related to a specific object.
**Naming Rules for Codes**

[R 7] For codes the property term is what is coded.

**Example:**
A ContactPersonFunctionTypeCode (DEN: Contact Person. Function. Type. Code) is the coded type of the function a contact person has.

A CostCentreTypeCode (DEN: Cost Centre. Type. Code) is the coded type of a cost center.

A CurrencyCode (DEN: Currency. Code) is the coded currency.

[R 8] Type, Category, Class, Status, Reason, Rating, Rule, Method or Purpose are to be treated as property terms for codes.

[R 9] For codes the property qualifier is the qualification of what is coded.

**Example:**
A ContactPersonFunctionTypeCode (DEN: Contact Person. Function. Type. Code) is the coded type of the function a contact person has.

[R 10] For codes the object class term is the object, the code is related to.

**Example:**
A ContactPersonFunctionTypeCode (DEN: Contact Person. Function. Type. Code) is the coded type of the function a contact person has.

[R 11] Groups of objects (in contrast to Types, Categories and Classes) are in general considered to be an object class term and not a property term.

**Example:**
Naming Rules for Date Times, Dates, Times

[R 12] For date times the property term is what happens.

Example:
A MeetingScheduledStartDateTime (DEN: Meeting. Scheduled_ Start. Date Time) is the scheduled point in time, when a meeting starts.
A DeliveryDateTime (DEN: Delivery. Date Time) is the point in time, when something is delivered.

[R 13] For date times the property qualifier is the qualification of what happens.

Example:
A MeetingScheduledStartDateTime (DEN: Meeting. Scheduled_ Start. Date Time) is the scheduled point in time, when a meeting starts.

[R 14] For date times the object class term is the object that is related to what happens.

Example:
A MeetingScheduledStartDateTime (DEN: Meeting. Scheduled_ Start. Date Time) is the scheduled point in time, when a meeting starts.

Naming Rules for Identifiers

[R 15] For identifiers the property term is “Identification” or “Universally Unique Identification”.

[R 16] For names of identifiers the property term and representation term “Identification. Identifier” is replaced by “ID”.

[R 17] For names of identifiers the property term and representation term “Universally Unique Identification. Identifier” is replaced with “UUID”.

Example:
A ProductStandardID (DEN: Product. Standard_ Identification. Identifier) is the standardized identification for a product.
The element with object class term “Product”, property term: “Universally Unique Identification”, representation term “Identifier” has the semantic name “ProductUUID”.

[R 18] For identifiers the property qualifier is “Standard”, “Party” or “Internal” if a property qualifier is needed. “Standard” may be replaced in element names, if only one specific standard is possible.
Example:

A **ProductStandardID** (DEN: Product. Standard. Identification. Identifier) is a standardized identification for a product.

A **ProductGTINID** (DEN: Product. GTIN. Identification. Identifier) is a GTIN standardized identification for a product.


A **ProductInternalID** (DEN: Product. Internal. Identification. Identifier) is an internally assigned identification for a product.

[R 19] For identifiers the object class term is the object to be identified.

Example:

A **ProductID** (DEN: Product. Identification. Identifier) is an identifier for a product.

**Naming Rules for Indicators**

[R 20] For indicators the property term is what is indicated.

Example:

A **CancelledIndicator** (DEN: Cancelled. Indicator) indicates whether something is cancelled or not.


[R 21] For indicators the property qualifier is the qualification of what is indicated.

Example:


[R 22] For indicators the object class term is the object the indicator is related to.

Example:

Many indicators are not related to a specific object and have no explicitly given object class term.

**Example:**
A CancelledIndicator (DEN: Cancelled. Indicator) indicates whether something (any object) is cancelled or not.

**Naming Rules for Texts**

[R 23] For names of texts the representation term “Text” is omitted.

[R 24] For texts the property term is the kind of the text.

**Example:**
The GDT Description (DEN: Description. Text)
The element IdentifierIssuingAgencyName (DEN: Identifier Issuing_Agency. Name. Name)
The GDT SearchText (DEN: Search_Text. Text)
Most texts have the property term “Description”, “Name” or sometimes “Text” but other property terms like “Note” and “Prefix” are also possible.

[R 25] For texts the property qualifier is the qualification for the kind of the text.

**Example:**
A SearchText (DEN: Search_Text. Text) is a text that is searched for.
Most texts have no qualification for the kind of the text.

[R 26] For texts with a property term “Description” or “Name” the object class term is the object that is described or named.

**Example:**
The element ProductDescription (DEN: Product. Description. Text)
The element IdentifierIssuingAgencyName (DEN: Identifier Issuing_Agency. Name. Name)
### 3.4.2 Naming Rules for aggregated GDTs

[R 27] For aggregated GDTs and elements the representation term is “Details”.

[R 28] For names of aggregated GDTs and elements the representation term “Details” is omitted.

**Example:**

GDT Address (DEN: Address. Details)

[R 29] For aggregated GDTs and elements the object class term is the part of the dictionary entry name describing the object represented by the GDT or element.

**Example:**

In the GDT Address (DEN: Address. Details) “Address” is the object represented by the GDT.

In the element ContactPersonAddress (DEN: Contact Person_Address. Details) “Address” is the object represented by the element.

### Naming Rules for Date Time, Date, Time Periods

[R 30] For names of periods the property term “Date Time Period”, “Date Period” and “Time Period” is replaced with “Period”.

**Example:**

The element name DeliveryDateTimePeriod (DEN: Delivery. Date Time Period. Details) becomes DeliveryPeriod.
Naming Rules for Elements and Attributes

[R 31] For elements and attributes the name includes the object class term and the property term of the GDT’s name that is used for typing of the element or attribute.

Example:
The name for an element of the type Address includes the object class term “Address” and the representation term “Details”.

The name for an attribute of the type ActionCode includes the property term “Action” and the representation term “Code”.

[R 32] For elements and attributes of a type having no object class term the name includes the object class qualifiers and the object class term of the GDT the element or attribute is defined in as object class qualifiers and object class term.

Example:
According to [R 31] and [R 32] the country code element in the GDT Bank (DEN: Bank. Details) has the name BankCountryCode with “Bank” being the object class, “Country” being the property and “Code” being the representation term.

[R 33] For elements and attributes of a type having an object class term the name includes the object class qualifiers and the object class term of the GDT the element or attribute is defined in as object class qualifiers.

Example:
According to [R 31] and [R 33] the address element in the GDT Bank (DEN: Bank. Details) has the name BankAddress (DEN: Bank. Address. Details) with “Bank” being the object class qualifier, “Address” being the object class and “Details” being the representation term.

[R 34] For elements and attributes redundant object class qualifiers are skipped.

Example:
According to [R 31] and [R 34] the internal ID element in the GDT Bank has the full name BankBankID (DEN: Bank. Bank. Internal. Identification. Identifier). The redundant object class qualifier “Bank” is skipped so the correct name is BankID.

For individual GDTs there may be specific rules described with the GDT documentation for similar object class qualifiers to be skipped in element or attribute dictionary entry names.

Example:
For the GDT BusinessTransactionDocumentParty there is a specific rule that allows an object class qualifier Business Transaction Document Item to be skipped for element names.
[R 35] For elements and attributes a property term may be added if there is no property term given by the GDT used as type of the element or attribute.

Example:
The element BaseQuantity (DEN: Base_Quantity) of the type GDT Quantity in the GDT PriceDetails
The element StartDateTime (DEN: Start_DateTime) of the type GDT DateTime in the GDT DateTimePeriod

[R 36] For elements additional object class and property qualifiers may be added.

Example:
In the GDT PriceDetails the element PriceBase_Quantity uses the additional representation qualifier “Base” to qualify the property term “Quantity”.
In the GDT Bank the element BranchAddress uses the additional object class qualifier “Branch” to qualify the object class “Address”.

[R 38] For elements object class terms naming a super class may be replaced with names of a sub class if the element is restricted to the sub class by the context where it is used. The following replacements are defined:

<table>
<thead>
<tr>
<th>Super Class</th>
<th>Sub Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generalized term from iBOM</td>
<td>Specialization term from iBOM</td>
</tr>
<tr>
<td>Business Transaction Document</td>
<td>&lt;Name of a code value as defined in the GDT BusinessTransactionTypeCode&gt;</td>
</tr>
<tr>
<td>Definition:</td>
<td>Example:</td>
</tr>
<tr>
<td>Refer to GDT</td>
<td>Purchase Order</td>
</tr>
<tr>
<td>BusinessTransactionDocumentID</td>
<td>Sales Order</td>
</tr>
<tr>
<td></td>
<td>Supplier Invoice</td>
</tr>
<tr>
<td>Logistics Branching</td>
<td>Execution Production Data Structure Production Segment Branching</td>
</tr>
<tr>
<td>Definition:</td>
<td>Production Order Branching</td>
</tr>
<tr>
<td>Refer to GDT</td>
<td>Site Logistics Data Structure Process Segment Branching</td>
</tr>
<tr>
<td>LogisticsBranchingID</td>
<td>Site Logistics Order Branching</td>
</tr>
<tr>
<td>Logistics Confirmation</td>
<td>Execution Production Data Structure Production Segment Operation</td>
</tr>
<tr>
<td>Definition:</td>
<td>Confirmation</td>
</tr>
<tr>
<td>A logistic confirmation records the progress of a logistic process</td>
<td>Execution Production Data Structure Production Segment Operation Confirmation</td>
</tr>
<tr>
<td></td>
<td>Execution Production Data Structure Production Segment Operation Confirmation</td>
</tr>
</tbody>
</table>
ess, for example, goods movement or component consumption.

<table>
<thead>
<tr>
<th>Cash Location</th>
<th>House Bank Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Cash Account</td>
</tr>
<tr>
<td></td>
<td>Check Storage</td>
</tr>
<tr>
<td></td>
<td>Bill of Exchange Book</td>
</tr>
<tr>
<td></td>
<td>Payment Card Receivables Account</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Price Specification Element</th>
<th>Price Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Price Specification</td>
</tr>
<tr>
<td>A PriceSpecificationElement specification of a price, a discount, a surcharge, or a tax that depends on a combination of properties, and that is valid for a specific period of time.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Financial Reporting Structure</th>
<th>Balance Sheet Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td>Income Statement Structure</td>
</tr>
<tr>
<td>A FinancialReportingStructure-Code is the coded representation of a financial reporting structure.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contribution Margin Schema</td>
</tr>
<tr>
<td></td>
<td>Cost Revenue Reporting Structure</td>
</tr>
<tr>
<td></td>
<td>Cash Flow Statement Structure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customs Declaration</th>
<th>Export Declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition:</td>
<td></td>
</tr>
<tr>
<td>A customs declaration is a declaration to the customs authority for moving goods or delivering services across territory borders according to legal requirements.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

An element of the type BusinessTransactionDocumentReference may be named PurchaseOrder-Reference. Details if it is restricted to reference only purchase orders by the context where it is used.
An element of the type ProductID may be named ServiceID if it is restricted to the identification of services by the context where it is used.

[R 39] For identifier elements “Party” may be replaced by the role of the party assigning the ID as defined in the GDT PartyRoleCategoryCode if the element is restricted to this role by the context where it is used.

Example:
An element of the type ProductPartyID may be named ProductSellerID, if the party issuing the ID is restricted to the “Seller” role by the context where the element is used.

[R 40] For element and attribute names the parts of the name (object class term, property term, or qualifiers) are removed which are given by the GDT the element or attribute is defined in.

Refer to [R 31], [R 32] and [R 33].

Example:
3.5 GDT Structure

A GDT represents a business-related subject matter that is described by a specific structure. This section describes how to specify such a structure.

The GDT structure provides information about:
- whether the GDT describes an object or a property
- attributes (supplementary components) a GDT contains
- elements that go to make up an GDT
- how data is represented
- the CDT or GDT the GDT is projected from

The structure is described in the GDT documentation in a table contained in the GDT Documentation, at "Structure".

<table>
<thead>
<tr>
<th>GDT</th>
<th>Cat.</th>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
<th>Type</th>
<th>Len.</th>
<th>Card.</th>
<th>Name</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The main table structure is setup according to ISO 11179. There a data element is made up of three parts:
- The "object class": A set of concepts, abstractions or things in the real world which can be identified within clear boundaries and meanings, and whose characteristics and behavior follow the same rules (examples: automobile, person, household, order ...).
- The "property": A characteristic feature shared by all the instances of an object class (examples: color, age, income, address ...).
- The "representation": Describes how the data is represented, meaning the data type and its value range (examples: a date can be represented with xsd:date or xsd:datetime).

The object class and property name components can be described in more detail by a qualifier.

The last columns of the structure table describe how the data is represented. There typing of attributes and attributes is specified as well as length and number of digits.

The first rows in the structure table describe the GDT itself and further rows attributes and elements belonging to the GDT.

[R 1] The GDT name is entered in column “GDT” in the first row.

**Example:** GDT DemandInfluencingEventStatusCode

<table>
<thead>
<tr>
<th>GDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DemandInfluencingEventStatusCode</td>
</tr>
</tbody>
</table>
3.5.1 Object Class, Property, and Representation

GDTs type the properties (= elements) of business objects and their operations with a specified structure. A GDT represents a specific business-related subject matter, and properties that reflect a particular subject are always typed by the same data type.

The business-related subject matter is described by the definition of a GDT. From the definition the name of the GDT is derived.
- A component of a GDT’s name may identify an object type (or object class). This is the case, if the corresponding part of the subject matter is a complex one that means it is described by further properties.
- In contrast, another component of a GDT’s name identifies a property, if the corresponding part of the business-related subject matter is a simple one.
- Each name component object type or property may be described in more detail by a qualifier. The qualified name component is a semantic specialization of the name component.

A business-related subject matter is represented in a certain structure. The possible representations are defined by representation categories (each has a correlated CDT) – see 3.4.1.

The name of the GDT is analyzed with the information provided by the definition to determine the object class term, property term, and representation term (see chapter 3.4). This analysis may lead to changes of the GDT’s definition and as a result to a new name:

[R 2] The Dictionary Entry Name has to match the object class term, property term, and representation term of the GDT specified in the structure table (first line of table).

Object class term and property term documentation in the structure table:

[R 3] The object class term is entered in column “Object Class Term”, its qualifier(s) into column “Object Class Qualifier”.
   The property term is entered in column “Property Term”, its qualifier(s) into column “Property Qualifier”.

Examples:

GDT PriorityCode

<table>
<thead>
<tr>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualifier</td>
<td>Term</td>
<td>Qualifier</td>
</tr>
<tr>
<td>Priority</td>
<td>Code</td>
<td></td>
</tr>
</tbody>
</table>

“Priority” is a property and may type priority properties of arbitrary business objects / operations.
GDT ProductID

<table>
<thead>
<tr>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualifier</td>
<td>Term</td>
<td>Qualifier</td>
</tr>
<tr>
<td>Product</td>
<td>Identification</td>
<td>Identifier</td>
</tr>
</tbody>
</table>

“Product” is an object type, “Identification” a property of this object. This GDT may type properties of business objects / operations that identify products.

GDT: ProductDemandInfluencingEventStatusCode

<table>
<thead>
<tr>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualifier</td>
<td>Term</td>
<td>Qualifier</td>
</tr>
<tr>
<td>Product_Demand_</td>
<td>Influencing Event</td>
<td>Status</td>
</tr>
</tbody>
</table>

“Influencing Event” is an object type. “Product Demand” qualifies the “Influencing Event” (“Product Demand Influencing Event” is a specialization of “Influencing Event”). “Status” is a property of this object. This GDT may type properties of business objects / operations that represent the status of a product demand influencing event.
3.5.2 Core Data Type

CDTs are the smallest reusable building blocks without any business semantic. They represent data according to the structure defined by W3C data types they're based on. For each representation category (listed in 3.5.1) a correlated CDT exists.

CDTs define the maximum structure for a certain category of representation. The CDTs form the basic quantity of the possible data types.

The CDTs are specified by the UN/CEFACT and described in the Core Component Technical Specification (see [http://xml.coverpages.org/UBL-CoreComponentTypes-xsd-20020314.html](http://xml.coverpages.org/UBL-CoreComponentTypes-xsd-20020314.html)).

3.5.3 Basic Global Data Type

A basic GDT represents a simple business-related subject matter. A basic GDT is the smallest reusable building block reflecting business semantic. Basic GDTs are based on CDTs that means at first level a basic GDT is derived from a CDT.

Besides the basic representation specified by the CDT the basic GDT is derived from a certain length can be specified for text-based or numeric GDTs to control the representation of data.

The length can be fixed, or it can be a range, and it is entered as follows:

- A number ‘n’ for a fixed length.
- “n..m” for a range, where ‘n’ and ‘m’ are the lower and upper limits respectively for the length.
- “n.m” for a decimal number, where ‘n’ is the number of digits before the decimal place and ‘m’ is the number of digits after it.

The following information is permissible for GDTs:

<table>
<thead>
<tr>
<th>Representation Term</th>
<th>Permitted Length Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>“1..m” (in exceptional cases a fix length ‘n’)</td>
</tr>
<tr>
<td>Identifier</td>
<td>“1..m” (in exceptional cases a fix length ‘n’)</td>
</tr>
<tr>
<td>Text (Name, Description, Note)</td>
<td>“1..m” (standardized lengths available!)</td>
</tr>
<tr>
<td>Amount</td>
<td>“n.m” (standardized lengths available!)</td>
</tr>
<tr>
<td>Quantity</td>
<td>“n.m” (standardized lengths available!)</td>
</tr>
<tr>
<td>Measure</td>
<td>“n.m” (standardized lengths available!)</td>
</tr>
<tr>
<td>Numeric (Value, Rate, Percent)</td>
<td>“n.m”</td>
</tr>
</tbody>
</table>
Length of Identifiers exceeding 60 characters:

There is a potential risk that Business Intelligence integration does not work (max. length of IDs in BI is 60 characters. Please contact a BI expert to clarify this potential issue.


**Example:** GDT UUID
UUIDs are 36 characters long

<table>
<thead>
<tr>
<th>Len.</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
</tr>
</tbody>
</table>

**Example:** GDT LocationID
LocationIDs can be 1 to 32 characters long

<table>
<thead>
<tr>
<th>Len.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1..32</td>
</tr>
</tbody>
</table>

**Example:** GDT Amount
Amounts can have up to 22 places before the decimal and 6 places after the decimal

<table>
<thead>
<tr>
<th>Len.</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.6</td>
</tr>
</tbody>
</table>
3.5.4 Aggregated Global Data Type

An aggregated GDT represents a complex business-related subject matter that is semantically structured by elements.

An element represents a part of a complex business-related subject matter exposed by the aggregated GDT the element belongs to. This part of a complex business-related subject matter is defined and its name derived from this definition following the same rules as for a GDT’s definition and name.

A group of elements of an aggregated GDT represent the full business-related subject matter of the GDT.

[R 5] In aggregated GDTs, the elements of the aggregate are listed in the “GDT” column, indented by one column. “E” (element) is entered in the column “Cat.” (category).

Example: GDT PaymentCard

<table>
<thead>
<tr>
<th>GDT</th>
<th>Cat.</th>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaymentCard</td>
<td></td>
<td>Payment Card</td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>E</td>
<td>Payment Card</td>
<td>Identification</td>
<td>Identifier</td>
</tr>
<tr>
<td>CategoryCode</td>
<td>E</td>
<td>Payment Card</td>
<td>Category</td>
<td>Code</td>
</tr>
<tr>
<td>...</td>
<td>E</td>
<td>Payment Card</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Description</td>
<td>E</td>
<td>Payment Card</td>
<td>Description</td>
<td>Description</td>
</tr>
<tr>
<td>...</td>
<td>E</td>
<td>Payment Card</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The structure of aggregated GDTs should be uniform throughout. This means that the semantically sequence of the elements should always be the same.

[R 6] In aggregated GDTs, the respective identifier, if it exists, should be defined as the first element.

[R 7] Element groups that occur in several GDTs have always to be structured in the same order.

Typing and cardinality

The data of an element is structured according to the structure definition of a CDT / GDT that types the attribute or element. A restriction on the structure definition during typing is currently not possible.

For correct typing the business-related subject matter of the element and the typing CDT / GDT have to fit. Hence, the structure of the element name (object class term, property term, and representation term) and the CDT / GDT have to fit. The element may only have additional qualifiers for the object class and / or property term. This means an element may be semantically more specific as the subject matter represented by the typing CDT / GDT (the subject matter represented by an element is a specialization of the subject matter represented by the CDT / GDT).

The cardinality specifies the multiplicity of an element.
[R 8] Elements must be typed with a GDT or a CDT.

[R 9] The cardinality of the elements of a GDT is entered in the “Card.” (Cardinality) column.

[R 10] Only the cardinalities “0..1” and “1” are permitted if the GDT is used in business objects.
3.5.5 Attribute

An attribute (called “Supplementary Component” at CDTs from UN/CEFACT CCTS) defines an elementary feature of a data type. Attributes and a GDT form a semantic unit in which the GDT is depending on the attributes. This dependency can be described as a function:

$$ \text{GDT} := f(x,a_1,...,a_n) $$

If an attribute changes its value, the GDTs value itself changes. This change can be just a different interpretation of the GDT’s value or a change of the content).

In contrast to an element the binding of a single attribute to an GDT is stronger. A change of an element’s value can – but does not have to – lead to a change of values of other elements or to their interpretation. A change of an attribute’s value leads to a change of values of a GDT or to their interpretation.

Example: CDT Amount

```xml
<Amount currencyCode="USD">10</Amount>
```

A change of the currency “USD” to “EUR” would imply a currency conversion, if the represented monetary value should be stable:

```xml
<Amount currencyCode="EUR">0.7</Amount>
```

Example: GDT PaymentCard

```xml
<PaymentCard>
  <ID>4511</ID>
  <CategoryCode>1</CategoryCode>
  ...
  <Description>Card for travel expenses</Description>
  ...
</PaymentCard>
```

A change of element “Description” to “Card for intra-country travel expenses” does not affect the other element’s values or the interpretation of the GDT. The payment card stays the same.

An attribute represent a specific elementary business-related subject matter. This subject matter is defined and its name derived from this definition following the same rules as for a GDT’s definition and name. The only difference in naming is:

[R 11] All the names of attributes must be in LowerCamelCase (LCC) notation. Only the first word begins with a lower-case letter, and all subsequent words with an upper-case letter; there are no spaces or separators between the words.

Attributes can occur in Basic Global Data Types and Aggregated Global Data Types.

Attribute documentation in the structure table:

[R 12] The attributes of a GDT are listed in the “GDT” column, indented by one column, under the name of the GDT. “A” (attribute) is entered in the column “Cat.” (category).

[R 13] The object class term is entered in column “Object Class Term”, its qualifier(s) into column “Object Class Qualifier”.

© 2009 SAP AG
Dietmar-Hopp-Allee 16,
D-69190 Walldorf

Part IV of VI: Global Data Type (GDT) Design
Version 2.7 Date: February 27th, 2009
Authors: Michael Seubert, Dirk Richsteiger

Intellectual Property Rights and
Copyrights protected
The property term is entered in column “Property Term”, its qualifier(s) into column “Property Qualifier”.

**Example:** GDT LocationID

<table>
<thead>
<tr>
<th>GDT</th>
<th>Cat.</th>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocationID</td>
<td></td>
<td>Location</td>
<td>Identification</td>
<td>Identifier</td>
</tr>
<tr>
<td>schemeID</td>
<td>A</td>
<td>Identification Scheme</td>
<td>Identification</td>
<td>Identifier</td>
</tr>
<tr>
<td>schemeAgencyID</td>
<td>A</td>
<td>Identification Scheme Agency</td>
<td>Identification</td>
<td>Identifier</td>
</tr>
</tbody>
</table>

<LocationID schemeAgencyID="9">4012345678910</LocationID>

“Location 4012345678910 according to location ID scheme issued by EAN.UCC (International Article Numbering association)”

**Attributes in aggregated GDTs**

Aggregated GDTs may have attributes on aggregate level.

If an aggregated GDT is made up of a number of elements of the same type having attributes, you can substitute the attributes by a default attribute if the element attributes always have the same value. For example, if a number of amounts (GDT Amount) are always used with the same currency in an aggregated GDT.

[R 14] In Aggregated GDTs attributes are always on top of the data type structure.

[R 15] If all the elements in an aggregated GDT have the same attribute, containing the same values at runtime, the attribute can be assigned to the aggregated GDT as an obligatory default attribute and be omitted from the elements.

**Example:** GDT ProductDimensions

<table>
<thead>
<tr>
<th>GDT</th>
<th>Cat.</th>
<th>Object Class</th>
<th>Property</th>
<th>Representation/ Type Association</th>
<th>Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductDimensions</td>
<td></td>
<td>Product Dimensions</td>
<td>Details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>measureUnitCode</td>
<td>A</td>
<td>Product Dimensions</td>
<td>Measure Unit Code</td>
<td>GDT MeasureUnitCode</td>
<td></td>
</tr>
<tr>
<td>LengthMeasure</td>
<td>E</td>
<td>Product Dimension</td>
<td>Length Measure</td>
<td>GDT Measure</td>
<td></td>
</tr>
<tr>
<td>WidthMeasure</td>
<td>E</td>
<td>Product Dimension</td>
<td>Width Measure</td>
<td>GDT Measure</td>
<td></td>
</tr>
</tbody>
</table>
For attributes the same rules for typing and cardinality specification apply as for elements (see 3.5.4)

Please refer to the prior section and to section 4.1 for design guidelines on attributes.
3.5.6 Template and Projection

Template
A template is introduced for Global Data Types representing similar subject matters

- Generalized term expresses the common semantics of the Global Data Types

A template Global Data Type specifies the union of all elements and attributes of the Global Data Types, without any redundancy.

![Diagram](image)

**Figure 14: Correlation between GDT and Template GDT**

A template Global Data Type

- Is an Global Data Type described by a Global Data Type definition
- Has no implementation (can’t be used for typing elements outside of templates)

Global Data Types are projected from a template Global Data Type

- Global Data Type template ensures consistency of the projected Global Data Types

The template Global Data Type Model contains all components that are part of the projected Global Data Types. These are:

- All SAP attributes / Supplementary Components (including their typing)
- All elements (including their typing)
From the template Global Data Type, Global Data Types can be projected:

- The “Projection”- operator relates the GDT template and its projected GDTs

Figure 15: Template Global Data Type Model

Figure 16: Projected GDT and its Relation to the Template GDT
Definition Pattern and Naming Rule for Template Global Data Type

Definition Pattern

Definition
A template that comprises the maximal possible set of attributes and elements for the <generalized term>s projected from the template.

A <generalized term> is a …

Example: BusinessTransactionDocumentLocation_Template

Definition:
A template that comprises the maximal possible set of attributes and elements for the BusinessTransactionDocumentLocations projected from the template.

A BusinessTransactionDocumentLocation contains the information that is exchanged in business documents about a location relevant for business transactions. This information identifies the location and its address. The identification may be a company-internal ID, a standardized ID, or one or several partner-specific IDs. If the location is an address only, it may be identified via reference to this address. A location is a logical or a physical place.

Naming Rule

The name of a template Global Data Type is:

Syntax: <generalized term>_Template

<generalized term>: Term expressing common semantics of projected Global Data Types

Examples:
- BusinessTransactionDocumentLocation_Template
- BusinessTransactionDocumentParty_Template
Projection
A projection defines a view of a CDT / GDT for
• the business-related subject matter (= semantical specialization of the subject matter)
• the structure
• the value range

GDTs are, with regard to the represented subject matter, maximally defined data type that contain all attributes / elements or have the maximum length and value range (including code lists) required for the representation of that subject matter. Hence, the structure of GDTs that semantically specializes other existing GDTs must always be a projection of these GDTs.

Projection Rules for Basic Global Data Types
A Basic GDT is a projection of
• either a Core Data Type
• or another Basic GDT

A Basic GDT may specialize semantically another Basic GDT. This is reflected first by the definition and finally by a qualification of the object class term and / or property term.
A Basic GDT projected from a Core Data Type or another Basic GDT may define restrictions on
• length
• Supplementary Components
• value range (including code lists)
in the limits defined by the base GDT.

Example: GDT LocationID
LocationID is based on the CDT identifier

<table>
<thead>
<tr>
<th>Type</th>
<th>Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDT</td>
<td>Identifier</td>
</tr>
</tbody>
</table>

Projection Rules for Aggregated Global Data Types
An aggregated GDT may specialize semantically another aggregated GDT. This is reflected first by the definition and finally by a qualification of the object class term and / or property term.
An aggregated GDT projected from another aggregated GDT may define restrictions on
• attributes
• element structure
in the limits defined by the base aggregated GDT.
• Attributes / elements may be completely omitted
• Cardinalities may be stricter (optional attributes / elements may become mandatory) and the max-
  imum multiplicity may get lowered
• Semantic of attributes / elements may be refined
• Typing data type may be substituted by a projected GDT of the original GDT that fits to the se-
  mantic specialization

[R 16] If a GDT is projected from another GDT, then “GDT” (Global Data Type) is entered in the “Type”
  column. If, on the other hand, the GDT is based on a CDT, then “CDT” (Core Data Type) is en-
  tered in the “Type” column. The name of the type on which the GDT is based is entered in the
  “Type Name” column.

Note: If the GDT is an aggregated GDT and not projected, then the “Type” and “Type Name”
columns are not filled.

[R 17] In Basic GDTs only Supplementary Components of the base GDT from which the Basic GDT is
projected are permitted.

[R 18] A GDT projection’s structure is restricted with regard to the CDT or GDT on which it is based,
the term “Restricted” is to be entered in the “Remarks”.

Complete Examples of Structure Definitions

GDT LocationID – restricted with attributes

<table>
<thead>
<tr>
<th>GDT</th>
<th>Cat.</th>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
<th>Type</th>
<th>Type Name</th>
<th>Len.</th>
<th>Card.</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| Loca-
  tionID  |      | Location     | Identification | Identifier     | CDT  | Identifier| 1..32 |       | restricted |
| schemeID  | A    | Identification Scheme | Identification | Identifier | XSD Token | 1..60 | 0..1   |          |
| sche-
  meAgencyID | A    | Identification Scheme Agency | Identification | Identifier | XSD Token | 1..60 | 0..1   |          |

GDT GeoCoordinates - aggregated GDT with two elements and not projected

<table>
<thead>
<tr>
<th>GDT</th>
<th>Cat.</th>
<th>Object Class</th>
<th>Property</th>
<th>Representation</th>
<th>Type</th>
<th>Type Name</th>
<th>Len.</th>
<th>Card.</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| GeoCoor-
  dinates |      | Geo Coordinates | Details |                |      |           |      |       |         |
| Latitude-
  Measure  | E    | Geo Coordinates | Latitude | Measure | GDT Measure | 1    |        |         |
| Longitude-
  Measure  | E    | Geo Coordinates | Longitude | Measure | GDT Measure | 1    |        |         |
GDT ProductDimensions - aggregated GDT with two elements and a default attribute and not projected

<table>
<thead>
<tr>
<th>GDT</th>
<th>Cat.</th>
<th>Object Class</th>
<th>Property</th>
<th>Representation/Association</th>
<th>Type</th>
<th>Type Name</th>
<th>Len.</th>
<th>Card.</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dimensions</td>
<td></td>
<td></td>
<td>Product</td>
<td>Details</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>measure-UnitCode</td>
<td>A</td>
<td>Product</td>
<td>Measure UnitCode</td>
<td>GDT Measure-UnitCode</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LengthMeasure</td>
<td>E</td>
<td>Product</td>
<td>Length Measure</td>
<td>GDT Measure</td>
<td>0..1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WidthMeasure</td>
<td>E</td>
<td>Product</td>
<td>Width Measure</td>
<td>GDT Measure</td>
<td>0..1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.6 GDT Value Range, Integrity Conditions, Use, and Notes

For a GDT, the value range is specified (which values are permitted), which integrity conditions are valid, and how the GDT is generally used. In aggregated GDTs and GDTs with attributes, the elements and attributes must also be described.

[R 1] If uppercase or alpha conversion is used for ID- or Code-GDTs, the permitted value range is restricted. This has to be documented in section "Detailed Description and Value Ranges":
- Alpha conversion at ID-GDTs: “Leading zeros are not significant for IDs consisting of digits only.”
- Uppercase conversion: “Letters have to be in uppercase.”

The value range and the description of the attributes and elements are entered in the GDT Documentation, under “Detailed Description and Value Ranges”.

Note: Code lists are a special case. These are not entered in the GDT Documentation, under “Detailed Description and Value Ranges”, but in “Appendix – Code List”.

The integrity conditions are entered in the GDT Documentation, under “Integrity conditions”.

The use is described in the GDT Documentation, under “Use”.

[R 2] The description of the use of a GDT should describe the common use. Concrete examples of the use can be entered for explanatory purposes.

Example: GDT ProductWeights

1.1.7 Use

The ProductWeights can be used to calculate the total weight of a number of products.

Examples:
- The gross weight is important for selecting the method of transport since goods are normally transported in their packaging.
- The net weight is important for the maximum load bearing of a floor since the product is normally installed without its packaging.
- The packaging weight can be specified instead of the gross weight, for example, to save weighing the product once it is packed; the gross weight can then be calculated.

[R 3] The “Notes” section must not contain information that belongs to the other section. Only additional information is permitted (example: reference to DDIC)
3.7  GDT Appendix

Section is under construction. For further information please contact the coaching team.
3.8 Changes on Existing GDTs

A Global Data Type defines

- semantics – given by the Global Data Type’s definition
- structure –
  Aggregated GDT: specifies set, cardinality, and typing of elements / attributes
  Basic GDT: specifies set, cardinality and typing of attributes (Supplementary Components)
  and of how many digits or characters the content of a Basic GDT consists
- value range – specifies the valid values for elements / attributes or the content of a Basic GDT
- integrity conditions – specify the inner integrity conditions between elements and attributes of a Global Data Type (example: elements ID and UUID have to identify the very same object) and the outer integrity conditions (example: an internal ID may only be exchanged, if sender and recipient have access to shared master data)

for a specific business-related subject matter.

Each Global Data Type definition represents a contract users of Global Data Types rely on (and have to adhere to)!

If any of the semantic, structural, value range or integrity conditions specification is changed in an incompatible way, this contract is violated!

Every change of a Global Data Type has to be analyzed for its compatibility – considering all aspects of the contract: semantic, structure, value range, and integrity conditions.

Any change of an existing Global Data Type has implications on existing Business Object and Message Type models and implementations – at each element the Global Data Type types. A change can result in serious malfunctions of components of SAP solutions, SAP partner extensions, or SAP adapters of third-party software.

Example for an incompatible change violating the GDT’s contract:

GDT BusinessDocumentMessageHeader is extended by an optional element “TestDataIndicator”, that changes the integrity conditions in an incompatible way.

Sender: submits a request with test data (TestDataIndicator = true)
Recipient: uses still old version of the BusinessDocumentMessageHeader, hence ignores the TestDataIndicator
Result: Test data is processed as real data with all consequences!

3.8.1 Compatibility of Changes

A change of an existing Global Data Type is incompatible (and a new version of this Global Data Type is required; see 3.8.2), if the change impacts the processing logic of applications.
On the other hand a change is compatible, if it has no impact on the processing logic. This is the case, if only optional elements, attributes, or codes (to a code list) are added that have the character of only being additional information.

The following changes are incompatible:

- Change of the business-related subject matter (semantic definition) the Global Data Type represents
- Change of the length of a Basic Global Data Type (number of digits or characters)
- Change of the cardinality of attributes or elements
- Change of the valid value ranges
- Change of code lists that control processing logic

The changes that add optional information are compatible, for example:

- Addition of an optional UUID element to an existing mandatory ID element
- Addition of an optional name and/or description element to an ID or code element
- Addition of an optional description element
- Addition of further cancellation reasons to a code list

### 3.8.2 Incompatible Changes Force new GDT Versions

In general any incompatible change of a GDT requires creating a new version of that GDT. The version of a GDT is encoded in the GDT’s name as a suffix:

\[
\text{[name\_of\_GDT]}_{\text{Vx}}
\]

(the “x” is replaced by the current version ID; the version ID is counted from “0” in steps of 1)

The version suffix of the initial GDT version is omitted.

Only in exceptional cases an incompatible change of a GDT is possible. This is the case, if the following conditions are met:

- The GDT has not been shipped to partners or customers yet.
- The incompatible change is aligned with the owner and users of this GDT all over SAP, which includes an aligned plan for the adjustment of all affected implementations to adhere to the changed contract.
- The incompatible change is PICC approved.
3.9 Versioning of GDTs

Section is under construction. For further information please contact the coaching team.
4 Additional Concepts for Global Data Types

4.1 Attributes

An attribute defines an elementary feature of a data type. Attributes can occur in Basic Global Data Types and Aggregated Global Data Types:

![Figure 17: Example for allowed attributes basic and aggregated GDTs](image)

For Basic GDTs the only allowed attributes are the “Supplementary Components” defined by UN/CEFACT’s ebXML CCTS. Basic GDTs are projected from Core Data Types and inherit their “Supplementary Components” (examples are “schemeID” or “unitCode”).

For Aggregated GDTs other attributes as the given “Supplementary Components” are allowed. Aggregated GDTs consist of their elements and may have attributes. An element is a component of an aggregated GDT. All elements make up an Aggregated GDT. An attribute carries information about the content of a GDT. During run-time this information is needed to evaluate the content of an Aggregated GDT’s elements.

An attribute is designed, if a subject matter
- is information about content of a GDT
- is not part of a complex subject matter represented by an Aggregated GDT

**Example:** information for processing of messages

There are subject matters that deal with information needed for handling of messages, but do not belong to the business content of the message itself. This information is evaluated to process the content of a message. Currently the following attributes are defined:

- completeTransmissionIndicator
- actionCode
- renconciliationPeriodCounterValue

For naming rules and rules for attributes in a data type structure please refer to sections 3.4 and 3.5.
4.2 Restrictions on CDTs / GDTs

CDTs and GDTs can be restricted

- Semantically (may imply structural restrictions)
- Structurally (without semantic)

A semantic restriction is a refinement of a business subject matter represented by a data type. These restrictions are defined globally (namespace “SAPGlobal/GDT”).

A structural restriction is a limitation of an element structure, attributes, length, or value ranges of a data type that gets restricted. These restrictions are defined globally or – in exceptional cases – locally.

<table>
<thead>
<tr>
<th></th>
<th>semantic restriction</th>
<th>structural restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>global level</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>local level</td>
<td>-</td>
<td>(X)</td>
</tr>
</tbody>
</table>

Figure 18: Overview of restrictions and where they are defined

[R 1] Restrictions must get a PIC Governance approval.

[R 2] Restrictions must be motivated by standards or common best practices (“outside-in”).
4.2.1 Semantic Restriction – Concepts

A semantic restriction is expressed by a qualifier of a data type name.

There are two possible ways of semantic restrictions:

- **Semantic restriction by projected GDT**
  
  For this an extra data type is created. The semantic restriction can go along with structural restrictions and integrity conditions that are different to the ones given by the data type that gets restricted.

  **Example:** GDT URI → GDT WebURI (“Web” qualifies “URI”)

- **Semantic restriction by additional qualifier within existing CDT / GDT**
  
  This restriction does not require an extra data type, but a new qualifier is added to a list of qualified names for the data type that is restricted. Structure and integrity conditions are unchanged as there is no extra data type.
  
  The list of qualified data type names define valid element names on the 1st – and 2nd if necessary – qualifier level.

  **Example:** one CDT Quantity – qualified names {TotalQuantity, DeliveredQuantity...}

4.2.2 Structural Restriction – Variable Concept

Structural restrictions of CDTs / GDTs are needed on

- Element structure
- Attributes
- Length
- Value Ranges

To define a structural restriction the “Variable Concept” is used:

A variable is a “special qualifier” for a data type name that gets replaced in element names by an approved qualifier. For each variable a restricted data type is created:

<VARIABLE>_<data_type_name>

Only a limited list of variable names is allowed. This list is controlled and approved by the PIC Council.

[R 3] A variable name

- is in capital letters and has an underscore as suffix
- is human readable and understandable giving and not an artificial name like: "AN30_" or "30CHAR_-_"
Variables for Restriction on Data Types – Naming Rules
For variables some naming rules apply. These naming rules build according to the structural element that gets restricted:

[R 4] Data type restriction by leaving out elements or Supplementary Components (SC)

<element / SC name> + “INDEPENDENT”

or

enumeration of remaining elements

Examples: LANGUAGEINDEPENDENT_Name or STARTEND_DateTimePeriod

[R 5] Multiple restrictions on a data type
Each restriction is represented by a restriction variable. Variables are separated by an underscore, whereas the variables are ordered by the order of applied restrictions.

Example: SHORT_Name \rightarrow LANGUAGEINDEPENDENT_SHORT_Name

[R 6] Data type restriction on element and / or Supplementary Components (SC) level
SC and element restriction separated by an underscore:

<SC_restriction>_<element_restriction>_GDT

Example: CURRENCYN/_CNY_MEDIUM_Amount (variable name shortening refer to rule 7)

[R 7] Shortening Rule on variable names
GDT type name itself is not part of the variable

Example: NAMESHORT_Name \rightarrow SHORT_Name

[R 8] Restriction on Code GDTs
Variable name is either

• an enumeration of allowed codes
• or a semantic name of the group of codes (view of code list)

Examples: HOURDAYWEEK_MeasureCode or EUROPEANUNION_CountryCode

4.2.3 Local Restriction (exceptional)

In exceptional cases a local restriction is possible, if there are certain necessities that are approved by the PIC Council. “Local” are all restrictions that are not part of the global namespace “SAP-Global/GDT”.

[R 9] Global restrictions cannot be restricted multiple times locally.

![Figure 19: Invalid definition of a local restriction](image)

[R 10] For a local restriction a CDT / GDT gets copied into a local namespace

• without renaming of the data type
• with reference to the original CDT /GDT

[R 11] The owner takes full responsibility for interoperability issues caused by the local restriction.

![Figure 20: Valid definition of a local restriction](image)
4.3 Code Lists

4.3.1 Basics

Terminology:
A **code GDT** represents a business-related subject matter by one or more code lists.
A **code list** is a consolidated list of codes (value, name, and description).
A **code value** identifies an instance of the business-related subject matter represented by the code GDT.
A **code name** is the name of the instance of the business-related subject matter identified by the code value.
A **code description** is description of the instance of the business-related subject matter identified by the code value.

**Example (simplified):**
For the “clothes size” code type, there are the code lists
- *American clothes sizes* with the code values “S”, “M”, “L”, “XL”, …
- *German clothes sizes* with the code values “48”, “50”, “52”, …

A **code GDT** is the realization of a code type in ESR.

The responsibility for a code list can lie with a standard organization or with SAP, otherwise the code list is a user-specific code list.

To a code GDT at least one code list is assigned. The content of the code list is one of the following types:

<table>
<thead>
<tr>
<th>Content delivered by SAP – not extendable</th>
<th>Content delivered by SAP – extendable</th>
<th>No content delivered by SAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Content defined by standards’ organization or SAP</td>
<td>• Content defined by SAP</td>
<td>• Content defined by users</td>
</tr>
<tr>
<td>• Users cannot extend the content</td>
<td>• Users can extend the content</td>
<td>• No content defined by SAP</td>
</tr>
<tr>
<td>• Code list types: Standard or SAP static</td>
<td>• Code list type: SAP extensible</td>
<td>• Code list type: user specific</td>
</tr>
</tbody>
</table>

In the modeling of a code GDT, the following information must be entered in particular:
- The supported code lists must be entered, together with the organization responsible
[R 1] If possible, standard code lists should be used in code GDTs. If no standard code list exists, an SAP code list should be defined. Only in rare cases SAP does not ship any content or just dummy content (content that gets always deleted during customizing). These cases have to be justified at and approved by the GDT PIC Council.

[R 2] Published code lists are listed and described in a table.

[R 3] Examples of the semantics of the code values which the user can create are entered in the form `<code name>` - `<code description>`. The context which ensures the uniqueness of the code value is entered.

[R 4] If SAP defines the code value, it should be numbered sequentially, starting with “1”. Leading zeros must be avoided.

The following table shows which types of code GDTs are differentiated at present, and how these differences appear in the GDT structure and the documentation. You will find further details on the individual code types and their documentation in the code GDT documentation pattern here.

<table>
<thead>
<tr>
<th>Code List Content Type</th>
<th>GDT Structure</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content delivered by SAP – not extendable</td>
<td>listID, listVersionID, listAgencyID</td>
<td>Attributes can be omitted in the model on permission of the GDT PIC Council. Reasons could be: - Code GDT is intended for typing attributes - Implementation of code list as enumeration in ESR - Code GDT is bound to a specific standard by name of the GDT</td>
</tr>
<tr>
<td>Content delivered by SAP – extendable</td>
<td>listID, listVersionID, listAgencyID</td>
<td>Note: If a standard agency does not provide any listID and this agency defines only one list for the subject matter, the list can get identified uniquely by the “listAgencyID” and the “listID” can be omitted.</td>
</tr>
<tr>
<td>No content delivered by SAP</td>
<td>listID, listVersionID, listAgencyID</td>
<td>A number of code lists can be assigned to a code GDT as long as they have the same semantic. Examples of this would be country-specific code lists or code lists based on alternative standards.</td>
</tr>
</tbody>
</table>

[R 5] The code GDT documentation pattern is used to document a code GDT.

---

4 \dwdf029\pic-coaching\06_Global_Data_Types\90_GDT_Templates
4.3.2 Implementation

This section gives a brief overview of implementing code GDTs. For further details especially on implementation issues refer to the ESI Guidelines.

A code GDT is implemented in ESR by means of two data types with the following naming convention (<xyz> is a placeholder):

- <xyz>Code
- <xyz>CodeContextElements

<xyz>Code

The data type <xyz>Code is a “CDT” with the representation term “Code”. It can have attributes for the identification of the code list used and its administering organization. The attribute can be omitted if the code list used is always obvious, or if it is implicitly known by the sender and the receiver.

The fixed, specified names, lengths, XSD types and cardinality of the attributes are:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>XSD Type</th>
<th>Length</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>listID</td>
<td>xsd:token</td>
<td>1..60</td>
<td>optional</td>
</tr>
<tr>
<td>listVersionID</td>
<td>xsd:token</td>
<td>1..15</td>
<td>optional</td>
</tr>
<tr>
<td>listAgencyID</td>
<td>xsd:token</td>
<td>1..60</td>
<td>optional</td>
</tr>
</tbody>
</table>

This data type is used for typing elements in message structures. It gives you the option of using the attributes to identify the code list.

<xyz>CodeContextElements

The data type <xyz>CodeContextElements is an aggregate data type. It is used to implement the input help in the UI and contains as elements those fields which are required to make the code values unique.

These elements are indicated as mandatory fields in the structure.

Example:

The region “BE” is only unique in the context of a country:

“CH-BE” means Berne
“DE-BE” means Berlin

-------------------

This data type can contain other elements that are required as filter criteria in the search help. These elements are indicated as fields with the cardinality 0..1 in the structure.
5 Architecture Data Types

Architecture GDTs are essential GDTs that

- reflect fundamental design concepts,
- define the structure,
- control processing,

of applications including Business Objects, Service Interfaces / Service Operations.

<table>
<thead>
<tr>
<th>Architecture Data Types</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Meta Model and Structure – Meta Model</strong></td>
<td></td>
</tr>
<tr>
<td>ObjectCategoryCode</td>
<td></td>
</tr>
<tr>
<td>ObjectTypeCode</td>
<td></td>
</tr>
<tr>
<td>ObjectNodeTypeCode</td>
<td></td>
</tr>
<tr>
<td>ProcessComponentCode</td>
<td></td>
</tr>
<tr>
<td>BusinessProcessVariantTypeCode</td>
<td></td>
</tr>
<tr>
<td>ServiceInterfaceCode</td>
<td></td>
</tr>
<tr>
<td>CompoundServiceOperationCode</td>
<td>5.1</td>
</tr>
<tr>
<td><strong>Meta Model and Structure – Object Instance Identification</strong></td>
<td></td>
</tr>
<tr>
<td>ObjectID</td>
<td></td>
</tr>
<tr>
<td>ObjectNodeReference</td>
<td>5.1</td>
</tr>
<tr>
<td><strong>Meta Model and Structure – Business Document Structure</strong></td>
<td></td>
</tr>
<tr>
<td>BusinessDocumentRelationshipRoleCode</td>
<td></td>
</tr>
<tr>
<td>BusinessDocumentRelationshipTypeCode</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentID</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentItem-related Global Data Types</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentReference</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentRelationshipRoleCode</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentRelationshipTypeCode</td>
<td></td>
</tr>
<tr>
<td><strong>Meta Model and Structure – Service Operation Signature</strong></td>
<td></td>
</tr>
<tr>
<td>ActionCode</td>
<td></td>
</tr>
<tr>
<td>CompleteTransmissionIndicator</td>
<td></td>
</tr>
<tr>
<td>ReconciliationPeriodCounterValue</td>
<td></td>
</tr>
<tr>
<td>ChangeStateID</td>
<td></td>
</tr>
<tr>
<td>BusinessDocumentMessageHeader</td>
<td></td>
</tr>
<tr>
<td>BasicBusinessDocumentMessageHeader</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentLocation</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentParty</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentProduct</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentProductCategory</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>AttachmentFolder</td>
<td></td>
</tr>
<tr>
<td>Architecture Data Types</td>
<td>Section</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>BusinessProcessChainAssignment</td>
<td></td>
</tr>
<tr>
<td>CashDiscountTerms</td>
<td></td>
</tr>
<tr>
<td>MarketSegment</td>
<td></td>
</tr>
<tr>
<td>OrganisationAddress</td>
<td></td>
</tr>
<tr>
<td>PaymentControl</td>
<td></td>
</tr>
<tr>
<td>PersonalAddress</td>
<td></td>
</tr>
<tr>
<td>PhysicalAddress</td>
<td></td>
</tr>
<tr>
<td>TextCollection</td>
<td></td>
</tr>
<tr>
<td>WorkplaceAddress</td>
<td></td>
</tr>
<tr>
<td>Properties – Generic Properties</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td></td>
</tr>
<tr>
<td>PropertyDataType</td>
<td></td>
</tr>
<tr>
<td>PropertyDefinitionClass</td>
<td></td>
</tr>
<tr>
<td>PropertyReference</td>
<td></td>
</tr>
<tr>
<td>PropertyValuation</td>
<td></td>
</tr>
<tr>
<td>PropertyValue</td>
<td></td>
</tr>
<tr>
<td>Properties – Generic Properties</td>
<td></td>
</tr>
<tr>
<td>Amount</td>
<td></td>
</tr>
<tr>
<td>CurrencyCode</td>
<td></td>
</tr>
<tr>
<td>Measure</td>
<td></td>
</tr>
<tr>
<td>Quantity</td>
<td></td>
</tr>
<tr>
<td>MeasureUnitCode</td>
<td></td>
</tr>
<tr>
<td>Measure/QuantityTypeCode</td>
<td></td>
</tr>
<tr>
<td>Measure/QuantityRoleCode</td>
<td></td>
</tr>
<tr>
<td>TimePoint</td>
<td>5.2</td>
</tr>
<tr>
<td>Year, (Year)Quarter, (Year)Month, Year(Week), DayOfMonth</td>
<td></td>
</tr>
<tr>
<td>TimePointPeriod</td>
<td></td>
</tr>
<tr>
<td>TimeZoneCode</td>
<td></td>
</tr>
<tr>
<td>Recurrence</td>
<td></td>
</tr>
<tr>
<td>WeekdayCode</td>
<td></td>
</tr>
<tr>
<td>WeekdaySelection</td>
<td></td>
</tr>
<tr>
<td>CountryCode</td>
<td></td>
</tr>
<tr>
<td>RegionCode</td>
<td></td>
</tr>
<tr>
<td>CartesianCoordinates</td>
<td></td>
</tr>
<tr>
<td>GeoCoordinates</td>
<td></td>
</tr>
<tr>
<td>Properties – Numeric and Statistics Properties</td>
<td></td>
</tr>
<tr>
<td>CounterValue</td>
<td></td>
</tr>
<tr>
<td>DecimalValue</td>
<td></td>
</tr>
<tr>
<td>FloatValue</td>
<td></td>
</tr>
<tr>
<td>IntegerValue</td>
<td></td>
</tr>
<tr>
<td>OrdinalNumberValue</td>
<td></td>
</tr>
<tr>
<td>NumberValue</td>
<td></td>
</tr>
<tr>
<td>Rate</td>
<td></td>
</tr>
<tr>
<td>Numeric</td>
<td></td>
</tr>
<tr>
<td>Percent</td>
<td></td>
</tr>
<tr>
<td>Ratio</td>
<td></td>
</tr>
<tr>
<td>IntervalBoundaryTypeCode</td>
<td></td>
</tr>
<tr>
<td>Properties – Textual Properties</td>
<td></td>
</tr>
<tr>
<td>Architecture Data Types</td>
<td>Section</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>LanguageCode</td>
<td></td>
</tr>
<tr>
<td><strong>Processing and Technical Infrastructure – Processing</strong></td>
<td></td>
</tr>
<tr>
<td>PriorityCode</td>
<td></td>
</tr>
<tr>
<td>PriorityValue</td>
<td></td>
</tr>
<tr>
<td>BusinessScope</td>
<td></td>
</tr>
<tr>
<td>BusinessProcessVariantTypeCode (refer to Meta Model GDTs)</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionTypeCode</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentItemProcessingTypeCode</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionDocumentProcessingTypeCode</td>
<td></td>
</tr>
<tr>
<td>BusinessTransactionExecutionStatusCode</td>
<td></td>
</tr>
<tr>
<td>ProcessingStatusCode</td>
<td></td>
</tr>
<tr>
<td>ProcessingResultCode</td>
<td></td>
</tr>
<tr>
<td>Log</td>
<td></td>
</tr>
<tr>
<td>LogItem</td>
<td></td>
</tr>
<tr>
<td><strong>Processing and Technical Infrastructure – Date and Time Calculation</strong></td>
<td></td>
</tr>
<tr>
<td>DateCalculationFunctionCode</td>
<td></td>
</tr>
<tr>
<td>DateCalculationFunctionGroupCode</td>
<td></td>
</tr>
<tr>
<td>DateCalculationFunctionReference</td>
<td></td>
</tr>
<tr>
<td>ActiveTimePeriodApplicationStrategyCode</td>
<td></td>
</tr>
<tr>
<td>DayProgrammeApplicationStrategyCode</td>
<td></td>
</tr>
<tr>
<td>TimeDirectionCode</td>
<td></td>
</tr>
<tr>
<td>DateFormatCode</td>
<td></td>
</tr>
<tr>
<td>TimeZoneDifferenceValue</td>
<td></td>
</tr>
<tr>
<td>WeekPartitioningCode</td>
<td></td>
</tr>
<tr>
<td>PublicHolidayCalendarCode</td>
<td></td>
</tr>
<tr>
<td>WorkingDayCalendarCode</td>
<td></td>
</tr>
<tr>
<td><strong>Processing and Technical Infrastructure – Technical Infrastructure</strong></td>
<td></td>
</tr>
<tr>
<td>ABAPClassID</td>
<td></td>
</tr>
<tr>
<td>BusinessSystemID</td>
<td></td>
</tr>
<tr>
<td>ObjectNodeTechnicalID</td>
<td></td>
</tr>
<tr>
<td>ObjectNodePartyTechnicalID</td>
<td></td>
</tr>
<tr>
<td>ObjectNodeTechnicalReference</td>
<td></td>
</tr>
<tr>
<td>UUID</td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 21 List of Architecture Data Types**

These GDTs are described in more detail in the following sections.
5.1 GDTs for Object Model Meta Structure and Object Identification

Object categories, types, and node types have their coded representation in the corresponding GDTs:

- ObjectCategoryCode
- ObjectTypeCode
- ObjectNodeTypeCode

For identifying arbitrary object node instances the GDT ObjectID is used. If the root node an object instance is identified by the ObjectID, the object instance itself is identified by it, too.
Currently the following object categories are defined:

- Business Object (BO)
- Master Data Object (MDO)
- Business Transaction Document (BTD)
- Transformed Object (TO)
- Mass Data Run Object (MDRO)
- Dependent Object (DO)
- Technical Object (TecO)
- Template Object
Figure 24: Identifying an Arbitrary Object Node Instance with the GDT ObjectNodeReference

Within the GDT ObjectNodeReference the GDTs ObjectID, ObjectTypeCode, and ObjectNodeTypeCode are used to identify an arbitrary object node instance uniquely.

For this an ObjectID has to be globally unique within the context of an ObjectNodeTypeCode!
## 5.1.1 ObjectTypeCode and ObjectNodeTypeCode in Backend

The code lists of ObjectTypeCode and ObjectNodeTypeCode are implemented in the tables APC_C_OTC and APC_C_ONTC (see Figure 25).

### ObjectTypeCode table
(code table APC_C_OTC & text table APC_C_OTC_T)

<table>
<thead>
<tr>
<th>ObjectTypeCode</th>
<th>Constant Name</th>
<th>Object Category Code</th>
<th>BOProxy Name</th>
<th>Process Component Code</th>
<th>Abstract Class Indicator</th>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code of the object type</td>
<td>Name of ABAP constant</td>
<td>Code of the object category the object type belongs to</td>
<td>Name of ABAP proxy</td>
<td>Code of the process component the object type belongs to</td>
<td>Indicates, if the object is abstract and not implemented</td>
<td>Language</td>
<td>Name of the object type</td>
</tr>
</tbody>
</table>

### ObjectNodeTypeCode table
(code table APC_C_ONTC & text table APC_C_ONTC_T)

<table>
<thead>
<tr>
<th>ObjectNodeTypeCode</th>
<th>Node Proxy Name</th>
<th>Object Type Code</th>
<th>Application Type Code</th>
<th>Node Technical Name</th>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code of the object node type</td>
<td>Name of ABAP proxy</td>
<td>Code of the object type the object node type belongs to</td>
<td>Code of the application that defined the object node type</td>
<td>Full name of a node derived from the object model</td>
<td>Code of the object type</td>
<td>Alternative business term of the object node type</td>
</tr>
</tbody>
</table>

Figure 25: Backend tables of ObjectTypeCode- and ObjectNodeTypeCode-Lists

## 5.1.2 Applying for new ObjectTypeCodes

Object types from the Integrated Object Model get automatically their code values.

Generalizations / specializations of object types that are not reflected as entity in the Integrated Object Model are applied for via an Excel sheet: "ObjectTypeCode_UpdateRequest.xls". These object types have to get a PIC Governance approval!

## 5.1.3 Applying for new ObjectNodeTypeCodes

Object node types from the Integrated Object Model get automatically their code values.

Code values for object node types of dependent objects included in host objects are not generated automatically in all cases. Please contact Friedhelm Krebs in order to get missing code values.

Generalizations / specializations of object node types that are not reflected as entity in the Integrated Object Model are applied for via eMail. These object types have to get a PIC Governance approval!

---

6 `\dwdf029\pic-coaching\10_Architecture_Topic\IdentificationContext\DefiningSchemeCodeList\ObjectTypeCode_UpdateRequest.xls`
5.2 Measure and Quantity

Definition:
- Quantity is a non-monetary amount in a unit of measure
- Measure is a non-monetary amount in a physical unit of measure only

Figure 26: Measure covers only non-monetary amounts in a physical unit of measure

Measure and quantity are represented by corresponding Core Data Types. Both Core Data Types have a Supplementary Component “unitCode” of type “MeasureUnitCode” to specify the unit of measure like “KG”, “piece”…

To specify the type of a quantity or measure like “length” or “volume” the GDTs QuantityTypeCode and MeasureTypeCode can be used.

Measure and Quantity can occur in certain roles like “MaximumQuantity”. These roles are either qualifiers in element names or represented as a code value in QuantityRoleCode.

Overview of dependencies:

<table>
<thead>
<tr>
<th>CDT</th>
<th>Unit</th>
<th>Type</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>MeasureUnitCode</td>
<td>QuantityTypeCode</td>
<td>QuantityRoleCode</td>
</tr>
<tr>
<td>Measure</td>
<td>MeasureUnitCode</td>
<td>MeasureTypeCode</td>
<td>(no requirement)</td>
</tr>
</tbody>
</table>

For further details on quantities and measures see the guidelines of Basic Service Component “Quantity Conversion” for modeling\(^7\) and implementation\(^8\).

\(^7\) `\dwd\0\29\ap\_eng\public\Rollout_Reuse_Components\QuantityConversion\Quantity_Measure_Type_Role.doc`
\(^8\) `\dwd\0\29\ap\_eng\public\Rollout_Reuse_Components\QuantityConversion\Development_Guideline.doc`
6 Quick Reference to Global Data Type Modeling

6.1 Basics

1. GDT Determination
   - Business subject and use case
   - Lookup of “suitable” GDT and qualifier from the GDT catalog

2. GDT Definition
   - Standard (ISO 11179) and guideline compliance
   - Business standard and pattern compliance

3. GDT Name
   - Derivation of GDT name from the definition
   - Standard (ISO 11179) compliance

4. GDT Structure
   - Restriction and length
   - Attributes and elements

5. GDT Value Range
   - Permissible value range for GDT and its elements

6. GDT Integrity Conditions
   - External integrity conditions of GDT
   - Internal integrity conditions of GDT between elements and attributes

7. GDT Use and Notes
   - General use or example use cases
   - Further information about GDT

8. Code List
   - Type of code list
   - Code definition, name, and value

9. Qualifier List
   - Definition of qualified GDT

Figure 27: Steps for Creating a GDT

Global Data Types (GDTs) are used to type business object node elements and interface elements. Each GDT represents exactly one business subject in structured form. The following is a brief introduction on how to select a GDT and create a new GDT. Every GDT has to be documented in a specific structure. Templates and patterns are available for this purpose. New GDTs have to be approved by the Process Integration Council.
6.1.1 GDT Determination

- First you define the internal element structure of a BO node. Each of the individual node elements represents one business subject.

- A GDT represents this subject in structured form.

- For each node element you identify a GDT for typing which appropriately represents this subject in structured form.

  Sources for the GDT search are:
  - GDTs created in ARIS that are still running through the PIC Governance Process.
  - Adopted GDTs in the GDT catalog

- If there is no GDT that appropriately represents this subject, you must adapt an existing GDT or define a new GDT. The following section only describes the creation of a new GDT.

- For more information, see chapter 3.2.

6.1.2 GDT Definition

- Once you have identified the need for a new GDT, the definition of the business subject to be represented is first put into concrete form.

- A definition generally has to be coherent and reflect only the business subject. Technical details of the implementation are irrelevant here.

- A definition is created following the general pattern "A(n) …<is / specifies / …> …". The use or an occurrence may not be described.

- The definition is added to the GDT Documentation, under "Definition".

- Examples:
  "Price: An exchange value (expressed in monetary units) of a product or a service with respect to the base quantity."

  These definitions explain in general terms what is involved in a business sense.

  "PaymentControlData: All the information used for triggering a payment transaction."

  Bad definition! This definition is ineffective because it is not clear what is meant by "information" or "triggering a payment transaction", or what "Control" entails.

- For more information see chapter 3.3.
6.1.3 GDT Name

- The name of the GDT that you are creating is derived from its definition. Here, the key terms of the definition form the name.

- The name of a GDT is formed in accordance with ISO11179. Names must be in British English and in “Upper Camel Case” (“ThisIsUpperCamelCase”).

- The parts of the names are used to generate what is referred to as the Dictionary Entry Name, which is made up of the Object Class Term, Property Term and Representation Term. The Object Class Term and the Property Term can be given qualifier prefixes.

- The Dictionary Entry Name is added to the GDT Documentation, under “Dictionary Entry Name”.

- The name and the name parts are added to the GDT document, in the corresponding columns of the structure table at section 1.1.3.

- Examples:

  "ProductDemandInfluencingEventStatusCode is a coded representation for the status of an event which affects the demand for products. This event could be a promotional event, for example."

  The GDT name can be derived directly from the definition.


  "An EmploymentChallengeStatisticExceptionCode is a coded representation of an exception of the statistic entry of a contract for a highly challenged employee."

  Improper name! Here, the GDT name is incorrectly derived from the definition: The term “Employment” is not used in the definition, and important parts of the definition such as the contract and the employees are not reflected in the name.

- For more information see chapter 3.4.
6.1.4 GDT Structure

- Global Data Types (GDTs) are data types that have business semantics. They are based on Core Data Types that do not have business semantics.

<table>
<thead>
<tr>
<th>Global Data Type</th>
<th>Business Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CCTS Core Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W3C Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Figure 28: Between Core Component Types and Generic Data Types

- Core Data Types are based on the XML Schema Definitions (XSD) of the W3C.
- A GDT can have attributes necessary for understanding (such as specifying the currencyCode for Amount).
- A GDT can be aggregated. In other words, it may consist of elements that have to be typed using GDTs.
- The cardinality has to be specified for elements and attributes ("0..1" or '1'; for technical reasons cardinality 'n' is not permitted for use in business objects).
- This structure information is entered in the GDT Documentation in the structure table, under "Structure".
### Basic Data Type (Line 1)

<table>
<thead>
<tr>
<th>Column</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDT Name (Column 1)</td>
<td>Name of the GDT</td>
</tr>
<tr>
<td>Category (Cat.)</td>
<td>No entry</td>
</tr>
<tr>
<td>Object Class Term &amp; Qualifier</td>
<td>Parts of GDT name that represent an object and its qualifier (no entry, where applicable)</td>
</tr>
<tr>
<td>Property Term &amp; Qualifier</td>
<td>Parts of GDT name that represent a property and its qualifier (no entry, where applicable)</td>
</tr>
<tr>
<td>Representation Term</td>
<td>Representation Term of a Core Component Type</td>
</tr>
<tr>
<td>Type</td>
<td>“CDT”, “GDT”, “xsd”</td>
</tr>
<tr>
<td>Type Name</td>
<td>Name of the Core Data Type (for example, “Identifier”), Generic Data Type, or XML Schema Definition</td>
</tr>
<tr>
<td>Length (Len.)</td>
<td>See step 5</td>
</tr>
<tr>
<td>Cardinality (Card.)</td>
<td>No entry</td>
</tr>
<tr>
<td>Remarks</td>
<td>Possibly “restricted”</td>
</tr>
</tbody>
</table>

### Aggregated Data Type (Line 1)

<table>
<thead>
<tr>
<th>Column</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDT Name (Column 1)</td>
<td>Name of the GDT</td>
</tr>
<tr>
<td>Category (Cat.)</td>
<td>No entry</td>
</tr>
<tr>
<td>Object Class Term &amp; Qualifier</td>
<td>Parts of GDT name that represent an object and its qualifier</td>
</tr>
<tr>
<td>Property Term &amp; Qualifier</td>
<td>No entry</td>
</tr>
<tr>
<td>Representation Term</td>
<td>“Details”</td>
</tr>
<tr>
<td>Type</td>
<td>No entry</td>
</tr>
<tr>
<td>Type Name</td>
<td>No entry</td>
</tr>
<tr>
<td>Length (Len.)</td>
<td>No entry</td>
</tr>
<tr>
<td>Cardinality (Card.)</td>
<td>No entry</td>
</tr>
<tr>
<td>Remarks</td>
<td>Possibly “restricted”</td>
</tr>
</tbody>
</table>
### Attribute (other lines)

<table>
<thead>
<tr>
<th>Column</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDT Name (other columns)</td>
<td>Name of the attribute</td>
</tr>
<tr>
<td>Category (Cat.)</td>
<td>&quot;A&quot;</td>
</tr>
<tr>
<td>Object Class Term &amp; Qualifier</td>
<td>Name parts of the property that represent an object and its qualifier (no entry where applicable)</td>
</tr>
<tr>
<td>Property Term &amp; Qualifier</td>
<td>Name parts of the property that represent a property of the object and its qualifier (no entry where applicable)</td>
</tr>
<tr>
<td>Representation Term</td>
<td>Representation Term of the Core Component Type</td>
</tr>
<tr>
<td>Type</td>
<td>“CDT”, “GDT”, “xsd”</td>
</tr>
<tr>
<td>Type Name</td>
<td>Name of the Core Data Type (for example, “Identifier”), Generic Data Type, or XML Schema Definition</td>
</tr>
<tr>
<td>Length (Len.)</td>
<td>See step 5</td>
</tr>
<tr>
<td>Cardinality (Card.)</td>
<td>“0..1” or “1”</td>
</tr>
<tr>
<td>Remarks</td>
<td>Possibly “restricted”</td>
</tr>
</tbody>
</table>

### Element (other lines)

<table>
<thead>
<tr>
<th>Column</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDT Name (other columns)</td>
<td>Name of the element</td>
</tr>
<tr>
<td>Category (Cat.)</td>
<td>&quot;E&quot;</td>
</tr>
<tr>
<td>Object Class Term &amp; Qualifier</td>
<td>Name parts of the element that represent an object and its qualifier (no entry where applicable)</td>
</tr>
<tr>
<td>Property Term &amp; Qualifier</td>
<td>Name parts of element that represent a property and its qualifier (no entry where applicable)</td>
</tr>
<tr>
<td>Representation Term</td>
<td>Representation Term of the Core Component Type</td>
</tr>
<tr>
<td>Type</td>
<td>“CDT”, “GDT”, “xsd”</td>
</tr>
<tr>
<td>Type Name</td>
<td>Name of the Core Data Type (for example, “Identifier”), Generic Data Type, or XML Schema Definition</td>
</tr>
<tr>
<td>Length (Len.)</td>
<td>See step 5</td>
</tr>
<tr>
<td>Cardinality (Card.)</td>
<td>“0..1” or “1”</td>
</tr>
<tr>
<td>Remarks</td>
<td>Possibly “restricted”</td>
</tr>
</tbody>
</table>
Notes:
Names of attributes and elements are listed by their hierarchy level in the structure table under “GDT Name” (compare GDT Address). If there is a technical restriction on the underlying data type it is indicated in the “Remarks” column with the phrase “restricted”.

Examples of basic and aggregated GDTs can be found in the GDT catalog\(^\text{10}\).

For more information see chapter 3.5.

\(^{10}\) [https://portal.wdf.sap.corp/ir/servlet/prt/portal/prtroot/com.sap.km.cm.docs/guid/f0783985-4a5b-2810-4c9c-b8df0ccdafe0](https://portal.wdf.sap.corp/ir/servlet/prt/portal/prtroot/com.sap.km.cm.docs/guid/f0783985-4a5b-2810-4c9c-b8df0ccdafe0)
6.1.5 GDT Value Range

- The valid value range is specified for each GDT if it is not obvious.

- The value range determines the possible values, such as codes, numbers, character strings and so on, and possibly also the length (fixed length – for example, 10; variable length – for example, 1..10; and non-decimal digits, decimal digits – for example, 10.2), which the GDT can represent.

- For aggregated GDTs, this is specified for each element that has a technical restriction with respect to the underlying data type.

- For GDTs with attributes, you specify this for each attribute that has a technical restriction with respect to the underlying data type.

  **Note**: In the section “Detailed Description and Value Ranges”, each attribute, or in the case of aggregated GDTs each element, is described in business terms.

- The length specifications are entered in the GDT Documentation, in the “Len.” column of the structure table in the “Structure” section. The value range is entered in the “Detailed Description and Value Ranges” section. For more information on length specifications, see the GDT Design Guidelines under 3.5.

- For more information see chapter 3.6.

6.1.6 GDT Integrity Conditions

- The internal integrity conditions have to be specified for aggregated GDTs in particular, if elements affect each other.

- The values the GDT can assume may be affected by its environment. These external integrity conditions have to be specified.

- The integrity conditions are entered in the GDT Documentation, under “Integrity conditions”.

- For more information see chapter 3.6.
6.1.7 GDT Use and Notes

- In order that the GDT is reused again correctly, its common use has to be described.
- Common use means that you must not only specify concrete examples of use. Concrete use examples only serve as an example to illustrate common use.
- Further information (for example a reference to DDIC elements) can be provided in “Notes”
- The use is described in the GDT Documentation, under “Use”.
- Further information (for example a reference to DDIC elements) can be provided in “Notes”
- Notes are to be given in the GDT Documentation, under “Notes”.
- For more information see chapter 3.6.

6.1.8 Code list

- Code lists have to be specified for Code GDTs
- Excluded are Code GDTs for which no content can be defined (to be approved by PIC)
- The code lists are contained in the GDT Documentation, under “Appendix – Code lists”.
- For more information see chapter 4.3.

6.1.9 Qualifier list

- Qualifiers are defined for GDTs and CDTs if a GDT or CDT is only semantically specialized without any structural restriction
- A qualifier list a controlled vocabulary that ensures that element names reflecting the same business subject are identically (please refer to naming rules of elements in data types, nodes, and service operation signatures)
- The qualifier list is contained in the GDT Documentation, under “Appendix – Qualifier list”.
- For more information see chapter 3.2.
6.2 Supporting Resources for Creating GDTs

- **GDT catalog**\(^{11}\) with all PIC-approved GDTs
- All the GDTs currently being created in ARIS.
- Click [here]\(^{12}\) for patterns for generating Code, ID, Indicator, Name and Description GDTs.

6.3 Applying for new GDTs (or Changes to Existing GDTs)

- The GDT document must be created in the folder “10 GDT in preparation” in ARIS by the owner.
- To apply a GDT for PICC approval, the integration expert moves the GDT document into folder “20 GDT ready for PIC review”.
- The AP OPS checks the GDT and moves it into the folder “30 GDT in PIC review”.
- The PIC approves the GDT.
- Subsequently, the GDT goes into a two-week, SAP-wide review, and the coaching team enters the GDT into the GDT catalog.
- After the successful SAP-wide review, AP Operations centrally creates the GDT in the Enterprise Service Repository (namespaces: SAPGLOBAL/GDT & COMMON/GDT) and the proxy is generated.
- For more information see Process Integration Content Governance, section PIC Governance Process for Global Data Types (GDTs).

\(^{11}\) https://portal.wdf.sap.corp/irj/servlet/prt/portal/prtroot/com.sap.km.cm.docs/guid/f0783985-4a5b-2810-4c9c-b8df0ccdafe0

\(^{12}\) \dwdf029\AP_eng\06_Global_Data_Types\90_GDT_Templates
7  Index

A

attribute .......................................................... 51, 65

B

building blocks ..................................................... 10
business object model ........................................ 7
business objects ................................................... 12

G

GDT catalog ......................................................... 92
GDT pattern ........................................................ 92
global data type ................................................ 9
aggregated global data type ................................ 15
basic global data type ........................................ 15
core data type .................................................... 15

H

harmonization of business objects ......................... 10

I

ISO standard .................................................. 19

O

outside-in approach ........................................... 7

R

restriction ......................................................... 66
semantic restriction ............................................ 66
structural restriction .......................................... 66

S

service operations ............................................ 12
supplementary components ................................ 65
## 8 Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Name</th>
<th>Changes / Addition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>2007-09-06</td>
<td>Seubert, Michael; Rasch, Jochen</td>
<td>Published enterprise SOA Object &amp; Service Operation Design</td>
</tr>
<tr>
<td>2.0</td>
<td>2007-10-15</td>
<td>Kuehl, Axel</td>
<td>List of Architecture-GDTs added</td>
</tr>
<tr>
<td>2.0</td>
<td>2007-11-22</td>
<td>Richtsteiger, Dirk</td>
<td>CPM design steps updated (3.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;derived GDT&quot; replaced by &quot;projected GDT&quot;</td>
</tr>
<tr>
<td>2.1</td>
<td>2007-11-26</td>
<td>Kühl, Axel</td>
<td>New version of complete eSOA guideline handbook</td>
</tr>
<tr>
<td>2.2</td>
<td>2007-12-14</td>
<td>Richtsteiger, Dirk</td>
<td>Restructured GDT Design Guidelines</td>
</tr>
<tr>
<td>2.3</td>
<td>2008-02-05</td>
<td>Richtsteiger, Dirk</td>
<td>Deletion naming rule #30, Added note for IDs with more than 60 characters</td>
</tr>
<tr>
<td>2.4</td>
<td>2008-04-15</td>
<td>Richtsteiger, Dirk</td>
<td>Links for Object(Node)TypeCode Excels updated</td>
</tr>
<tr>
<td>2.4</td>
<td>2008-05-16</td>
<td>Richtsteiger, Dirk</td>
<td>Data type meta model changed Rule for Supplementary Components of IDs and Codes changed</td>
</tr>
<tr>
<td>2.5</td>
<td>2008-06-27</td>
<td>Kühl, Axel</td>
<td>New version of complete eSOA guideline handbook</td>
</tr>
<tr>
<td>2.5</td>
<td>2008-08-01</td>
<td>Richtsteiger, Dirk</td>
<td>Chapter 4.4 added</td>
</tr>
<tr>
<td>2.5</td>
<td>2008-08-19</td>
<td>Richtsteiger, Dirk</td>
<td>Corrections in chapter 4.4 and moved to 3.7</td>
</tr>
<tr>
<td>2.5</td>
<td>2008-10-08</td>
<td>Richtsteiger, Dirk</td>
<td>Chapter 3.5.6 updated</td>
</tr>
<tr>
<td>2.5</td>
<td>2008-11-26</td>
<td>Richtsteiger, Dirk</td>
<td>Chapter 3.2: link to qualifier overview added</td>
</tr>
<tr>
<td>2.5</td>
<td>2008-12-01</td>
<td>Richtsteiger, Dirk</td>
<td>Meta model: qualifier and code lists added</td>
</tr>
<tr>
<td>2.5</td>
<td>2008-12-10</td>
<td>Richtsteiger, Dirk</td>
<td>Chapter 2.8 added</td>
</tr>
<tr>
<td>2.5</td>
<td>2009-02-25</td>
<td>Richtsteiger, Dirk</td>
<td>Naming rule 38 extended by “Customs Declaration”</td>
</tr>
</tbody>
</table>