|  |
| --- |
| Florida Department of Transportation |
| Naming Conventions |
| Attributes, Columns and Elements |
|  |
| **Data Administration** |
| **2/1/2013** |

|  |
| --- |
| Document the naming conventions used by the Florida Department of Transportation (FDOT) for naming Repository Elements.  |

Table of Contents

[Purpose 3](#_Toc347472470)

[References 3](#_Toc347472471)

[Definitions 3](#_Toc347472472)

[Background 4](#_Toc347472473)

[Procedure 4](#_Toc347472474)

[Business Name 4](#_Toc347472475)

[Naming Rules 4](#_Toc347472476)

[Name Construction 5](#_Toc347472477)

[Elements and Columns 6](#_Toc347472478)

[Naming Rules 6](#_Toc347472479)

[Naming Guidelines 8](#_Toc347472480)

[Glossary 9](#_Toc347472481)

[Creating Glossary Abbreviations 9](#_Toc347472482)

[Creating Glossary Abbreviations 10](#_Toc347472483)

[Change History 10](#_Toc347472484)

[Appendix 1 11](#_Toc347472485)

[Class Word/Domain Data Type Crosswalk 11](#_Toc347472486)

# Purpose

To define and document the standards and conventions used by the Florida Department of Transportation (FDOT) for naming entity attributes, data elements, and database table columns.

# References

Naming Conventions – Physical Object Naming

# Definitions

**Abbreviation –** A shortened form of a word or phrase created by leaving out or substituting letters.

**Acronym** – a word formed from the first, or first few, letters of a series of words, e.g. radar = radio detecting and ranging.

**Attribute** – A characteristic or property associated with a business entity (person, place, event, and so on). An attribute is the logical equivalent to a column.

**Business Name –** A name recognized and understood by the business user for a discrete item of data.

**CASE**– Computer Aided Software Engineering. A category of tools used to develop and deploy computer application software. All aspects of the software development process are documented in the tool from initial requirements to code development, testing, and deployment.

**Class Term or Class Word** – the domain category to which a data element belongs. In FDOT usage, this will be the last node of an attribute or data element name.

**Column –** The physical implementation of an attribute in a database.

**Element** – Also known as “data element,” the smallest unit of information that can be perceived or understood by a computer application user. Element is an abstract item that forms the basis for attributes and columns. An attribute is a relationship between an element and an entity. A column is a relationship between an element and a table.

**Entity** – A concept, object, event, etc. of concern to the business.

**ERwin** – The CA-ERwin data modeling tool used by FDOT to define and document data models and entity relationship diagrams.

**Gen** – The CA-Gen CASE tool. This tool is used to define, produce, and deliver application systems.

**Glossary** – A list of words or phrases with definitions and abbreviations.

**Initialism –** an abbreviation formed from the initial letters of words forming a phrase, each letter specifically pronounced, e.g. CD for Compact Disk, DNA for Deoxyribonucleic Acid, and UHF for Ultra High Frequency.

**Key Unification** – the practice of combining like foreign key columns resulting from multiple relationships from a common origin to a single table.

**Metadata** – information that provides meaning and context to the business data of the organization.

**Repository** – the CA Repository, a software application designed to store, organize and make available electronically FDOT’s enterprise metadata.

**Repository Glossary** – A list of business terms, words and phrases, with their abbreviations stored in the Repository.

**Webstation Option –** A Web interface to the Repository. This will allow access to FDOT enterprise metadata related to tables, columns, data elements, the glossary, and other information.

# Background

This procedure was developed to document the naming conventions used to construct names for Repository Elements, Entity Attributes and Table Columns.

# Procedure

## Business Name

For the purposes of this document an attribute name and a business name are the same.

### Naming Rules

* 1. Order of terms from left to right:
		1. Prime business term for the object being referenced. A noun.
		2. Qualifier or modifier terms – usually adjectives or other nouns that qualify the context of the prime business term.
		3. Class term - the domain category to which a data element belongs, for example, for the data element Contract Letting Date, date will be the class term.
	2. Terms in a name should be limited to nouns and adjectives. These identify things and satisfy a primary concern of data modeling which is to answer the “What is it?” and “What does it look like?” questions. Verbs and adverbs, which represent actions or the “How is it done?” question, should be carefully scrutinized when they show up in a name. A process definition, code table or list of values may be lurking nearby.
	3. Business names are limited to no more than 32 characters in length.
	4. Business name terms are separated by a space.
	5. When the business name is longer than 32 characters then the order of actions to reduce the number of characters in the name is:
		1. Eliminate name terms that do not contribute meaning to the name.
		2. Translate name terms to their respective abbreviations working from right to left until the name is not longer than 32 characters.
		3. Look for another name that can substitute for the requested name.
1. FDOT does not support key unification. Role name attribute names that enter an entity (as foreign keys from multiple incoming relationships) to prevent key unification. The role name is based on the parent entity contributing the attribute to the child entity and is generally a name prefix.
2. Class terms or class words are determined by consulting the Class Word/Domain/Data Type Crosswalk. (See Appendix 1 below).

### Name Construction

A common and recommended method for generating a business name is to determine a name using the IBM “OF” language syntax and then reverse the order of the name terms and eliminate the connector terms.

Examples:

* 1. “The name of the manager of the project.”
		1. Eliminate the connectors to produce Name Manager Project.
		2. Reverse the term order to produce Project Manager Name.
		3. Project is the prime term, Manager is a qualifier for Name, Name is the class term.
	2. “The code that is the status of the contract.
		1. Eliminate the connectors to produce Code Status Contract.
		2. Reverse the term order to produce Contract Status Code.
		3. Contract is the prime term, Status is a qualifier for Code, Code is the class term.
	3. “The date that is the beginning of the contract.”
		1. Eliminate the connectors to produce Date Beginning Contract.
		2. Reverse the term order to produce Contract Beginning Date.
		3. Contract is the prime term, Beginning is a modifier for Date, Date is the class term.

## Elements and Columns

Data model columns, Repository elements and Repository columns all have the same naming rules.

### Naming Rules

1. Element names are based on abbreviations for the terms that contained in the business or attribute name.
2. The order of terms is the same as for the business or attribute name.
3. Business words or phrases are translated to element name terms using the Repository glossary. The glossary abbreviation will be substituted for each business word or phrase in the element business name to form the element name.
4. Element names are limited to a maximum of 18 characters to correctly generate COBOL copybooks in the Department’s z/OS environment.
5. Terms are separated by an underscore. The underscore between the class term and the term to its left is mandatory and shall not be removed.
6. When the direct translation of the attribute name results in a column name longer than 18 characters, then the order of actions to reduce the number of characters in the name is:
	* 1. Remove underscore characters except for the underscore which precedes the class term.
		2. Remove characters from an abbreviation term as long as meaning is not sacrificed. Any vowels that are part of the abbreviation are candidates. This is usually not a good idea, but evaluate the possibility anyway.
		3. Remove abbreviated terms as long as overall meaning is not sacrificed. This is also not usually a good idea, but evaluate the possibility anyway.
		4. Look for another attribute name with a shorter abbreviation.
7. Maximum length of an element name of data type VARCHAR or VARCHAR2 is 17 characters for all environments in order to correctly generate COBOL copybooks in our z/OS environment.
8. Term abbreviations in the element name are separated by underscore characters. This is done by replacing space characters in the business name with underscore characters in the element name.
9. FDOT does not support key unification. Role name identical element names entering a table as foreign key columns from multiple incoming relationships to prevent key unification. The role name should be based on the parent entity contributing the element to the child entity.
10. Elements created using the current rules shall have a status value of “PRIME” when created in the Repository.
11. Element names consist of alphanumeric characters and underscores only.
12. Examples:
	1. “Contract Letting Date”
		1. Substitute the abbreviation for each word in the element business name, resulting in CNTRT\_LET\_DT.
	2. “Average Annual Daily Traffic Yr” (Year is abbreviated to avoid violating the business name limitation of 32 characters.)
		1. Substitute the standard initialism AADT from the Repository glossary for the phrase “Average Annual Daily Traffic,” and use the abbreviation for year, resulting in AADT\_YR.
	3. “Outsider Location Contract Status Code “
		1. The translated name Out\_Loc\_Cntrt\_Stat\_Cd is 21 characters.
		2. Remove the underscores, except the one preceding the class term, and you get OutLocCntrtStat\_Cd which is 18 characters.
	4. “Vendor Location Contract Status Code”
		1. The translated name Vend\_Loc\_Cntrt\_Stat\_Cd is 22 characters.
		2. Remove the underscores and you get the 19 character name VendLocCntrtStat\_Cd.
		3. Remove the vowel from Vend and you get the 18 character name VndLocCntrtStat\_Cd.

### Naming Guidelines

1. Element names shall be reused where reasonable and appropriate. An element represents a business idea, concept, characteristic or property. When an element is used in multiple places the same element name should be used each time. Conversely, an element name is atomic in the Repository and means the same thing regardless of the entity or table in which it resides. FDOT provides a Web interface, known as Webstation Option (WSO), to the Repository. Development teams may use WSO to research elements for possible reuse, and to find possible name clashes with existing elements.
2. Code elements require special handling. FDOT has an enterprise table for storing and making available codes and descriptions. This table allows only a single set of code values and descriptions for a data element name. In order to avoid name clashes in this code table, code element names must be reserved in the Repository as soon as possible. Delaying to reserve the names may require renaming of these elements at a late stage in design, possibly causing rework for other deliverables dependent on these names.
3. If a business or element name acquires specialized characteristics because of the entity in which it resides then it may be appropriate to include the entity name, or abbreviation, in the business and element names. If the business name Work Activity Code has one set of values applied consistently throughout the organization *except* when it is used in the Maintenance business area then an appropriate application of this guideline would call for the creation of a new business name, and element, Maintenance Work Activity Code to represent the set of values used with the Maintenance business area.
4. Use of numerals, Roman numerals or words.

Several factors are taken into consideration when developing element names that contain numbers:

* 1. Length – if the generated name is too long when using words then consider replacing words for numerals with the numerals, i.e. replace “Four” with “4”, “MCCCIVIX” with “1349”.
	2. Consistency and Sensibility – be consistent when creating a series of elements with numbers in them. Use all numerals rather than the first 10 as words and the remainder as numerals.
	3. Size – always use numerals for numbers greater than 10.
	4. Ease of Reporting – if you are creating a series of elements with numbers that cross a precision boundary then use the number of digits needed to represent the larger precision boundary. If there are 11 elements then use “01”, “02”, “03” … “11” instead of “1”, “2”, … “9”, “10”, “11”. In the second example “1” and “11” will sort together, before “2”, when you probably want “11” to sort after “2”.

## Glossary

### Creating Glossary Abbreviations

#### Rules

1. For new abbreviations, contact FDOT Data Administration with the words or phrases that need abbreviating. Data Administration will create the new abbreviation and make it available to development teams who require it.
2. All business terms used in the Repository, an ERwin model or a Gen model, and their abbreviations, will be stored in the Repository Glossary. This information can be accessed through Webstation Option, a Web interface to the Repository. This software is accessible within the FDOT network.
3. The established abbreviations for terms will be used, i.e. the established abbreviation for the term “Administration” is “ADM”. “ADMIN” is not an acceptable abbreviation for this term and shall not be used for this term.
4. All terms will use the abbreviation established for the root form of the term, i.e. Accept, Accepts, Accepted, Accepting and Acceptance all use the abbreviation for Accept (ACPT).
5. The root form of a term will always be the singular form of the term.
6. All terms that are not word forms will have their own unique abbreviations.

### Creating Glossary Abbreviations

# Change History

|  |  |
| --- | --- |
| **Effective Date:** December 1, 2011 |  |
| **Create Date**: November 21, 2011  | **Created By:** Sarah Close |

|  |  |  |
| --- | --- | --- |
| **Last Update Date** | **Last Updated By** | **Reason for Change** |
| November 21, 2011 | Sarah Close | Convert procedure to new format |
| December 8, 2011 | Sarah Close | Updates based on received comments. |
| February 8, 2012 | Sarah Close | Updates based on received comments. |
| February 15, 2012 | Sarah Close | Minor typographical corrections and re-ordering. |
| February 28, 2012 | Sarah Close | Updates based on received comments. |
| January 29, 2013 | Morgan Bunch | Modified for distribution as part of RFQ process. |
| January 31, 2013 | Morgan Bunch | Added class word/domain/data type crosswalk as an appendix. Incorporated Jolene Jobe’s suggestions into this document. |
| February 1, 2013 | Morgan Bunch | Spell/grammar check. |

# Appendix 1

## Class Word/Domain Data Type Crosswalk

Those data types that are in bold are the preferred data types for that platform and class word.

The items starred (\*) below are the maximum length. They may be defined as smaller if necessary.

| **Class****Word** | **Abbrev** | **Domain****Key** | **Default****Data Type** | **Default****Length** | **Allowed Data Types** |
| --- | --- | --- | --- | --- | --- |
| **DB2** | **Oracle** | **SQL Server** | **Flat File** |
| AMOUNT | AMT | MONEY | DECIMAL | 15.2 | **DECIMAL** | **NUMBER** | **MONEY**DECIMAL | **NUMERIC**DECIMAL |
| AREA | AREA | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| BINARY | BIN | BINARY | BLOB | 0 | **BLOB** | **BLOB** | **BLOB** |  |
| CIRCUMFERENCE | CRCM | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| CODE | CD | CODE DOT | CHAR | 8\* | **CHAR** | **CHAR** | **CHAR** | **CHAR** |
| COMMENT | CM | TEXT | VARCHAR | 255 | **VARCHAR**CLOB<255, CHAR | **VARCHAR2**CLOB | **VARCHAR** | **CHAR** |
| COUNT | CNT | INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| DATE | DT | DATE | DATE | 0 | **DATE**CHAR | **DATE**CHARVARCHAR2 | **DATE**CHARVARCHAR | **CHAR** |
| DEPTH | DPTH | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| DESCRIPTION | DS | DESC DOT | CHAR | 30\* | **CHAR** | **CHAR** | **CHAR** | **CHAR** |
| DESC TBL | CHAR | 30 | **CHAR**VARCHAR | **VARCHAR2**CHAR | **VARCHAR**CHAR | **CHAR** |
| TEXT | VARCHAR | 255 | **VARCHAR**CLOB<255, CHAR | **VARCHAR2**CLOB | **VARCHAR** | **CHAR** |
| DIAMETER | DIAM | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| DISTANCE | DSTNC | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| HEIGHT | HT | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| IDENTIFIER | ID | ID | CHAR | 8 | **CHAR**INTEGER | **VARCHAR2**NUMBER | **VARCHAR**INTEGERCHAR | **CHAR**NUMERIC |
| LENGTH | LNGTH | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| NAME | NM | TEXT | VARCHAR | 255 | **VARCHAR**<255, CHAR | **VARCHAR2** | **VARCHAR** | **CHAR** |
| NUMBER | NUM | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| PERCENT | PCT | DECIMAL | DECIMAL | 7.4 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| PHONE | PH | PHONE | CHAR | 10 | **CHAR** | **VARCHAR2**CHAR | **VARCHAR**CHAR | **CHAR** |
| RADIUS | RAD | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| RATE | RATE | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| RATIO | RTIO | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| SEQUENCE | SQ | INTEGER | INTEGER | 0 | **INTEGER** | **NUMBER** | **INTEGER** | **NUMERIC** |
| TEXT | TXT | TEXT | VARCHAR | 255 | **VARCHAR**CLOB<255 CHAR | **VARCHAR2**CLOB | **VARCHAR** | **CHAR** |
| TIME | TM | TIME | TIME | 0 | **TIME**CHAR | **DATE**CHARVARCHAR2 | **TIME**CHARVARCHAR | **CHAR** |
| TIMESTAMP | TMS | DATETIME | TIMESTMP | 0 | **TIMESTMP** | **DATE**TIMESTMP | **DATETIME2** | **CHAR** |
| WEIGHT | WT | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| WIDTH | WDTH | DECIMAL | DECIMAL | 7.2 | **DECIMAL** | **NUMBER** | **DECIMAL** | **NUMERIC**DECIMAL |
| INTEGER | INTEGER | 0 | **INTEGER**SMALLINT | **NUMBER** | **INTEGER (INT)** | **NUMERIC** |
| YEAR | YR | YEAR | SMALLINT | 0 | **SMALLINT** | **NUMBER(4)** | **SMALLINT** | **NUMERIC(4)** |
| ZIP | ZIP | ZIP | CHAR | 9 | **CHAR** | **VARCHAR2**CHAR | **VARCHAR**CHAR | **CHAR** |