Prepared for:

DG Rijkswaterstaat

# ADEPTS technical documentation

May 2000

R3300

WL | delft hydraulics

# ADEPTS technical documentation

A. Hendriks

May 2000

WL | delft hydraulics

# Testplan Garantiewerk ADEPTS

André Hendriks

22 mei 2000

WL | delft hydraulics

| | |
|---|---|
| CLIENT: | Ministry of Transport, Public Works and Water Management<br>Directorate-General of Public Works and Water Management<br>National Institute for Coastal and Marine Management / RIKZ<br>Institute for Inland Water Management and Wastewater Treatment / RIZA |
| TITLE: | Testplan Garantiewerk ADEPTS |

ABSTRACT:

Tests performed to test whether the shortcomings in version 1.00.0004 of ADEPTS that should be remedied in version 1.00.0006 where indeed fixed.

REFERENCES:

| VER. | ORIGINATOR | | DATE | REMARKS | REVIEW | | APPROVED BY | |
|---|---|---|---|---|---|---|---|---|
| 1.00.06 | A. Hendriks | | may 22, 2000 | - | P. Gijsbers | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PROJECT IDENTIFICATION: | | | | | | | | |
| KEYWORDS: | | | | | | | | |
| CONTENTS: | TEXT PAGES | 8 | TABLES | 1 | FIGURES | 0 | APPENDICES | 0 |

STATUS: □ PRELIMINARY □ DRAFT ☒ FINAL

# I Inleiding

Na de eerste oplevering van ADEPTS 1.00.0004 bleken er nog enige onvolkomenheden in de database en programmatuur te zitten die binnen de kaders van het geldende contract opgelost moesten worden. Dit testplan omschrijft de tests die zijn uitgevoerd om te testen of deze onvolkomenheden zijn opgelost.

# 2 Testen

Waar bij deze testen een probleemnummer wordt vermeld, betreft dit het probleemnummer waaronder een afwijking is opgenomen is de "ADEPTS Probleemmeldingen database".

## 2.1 Performance

Volgens probleemnummer 4 is de performance op een aantal gebieden onvoldoende.

De performance testen zijn allen uitgevoerd op de PC van de ontwikkelaar, een Pentium Pro 450 MHz met 128 MByte RAM geheugen, draaiend onder Windows 95 (build 1212 B). De volgende andere programma's waren actief tijdens het testen:

- McAfee VShield 4.0.2
- Powerdesk Explorer 3.03a.

De gebruikte testselectie omvatten de stoffen *carbamaten* en *dioxinen*, de parameters *biodegradatie* en *log Kow* en alle omgevingssystemen (162 rijen). De weergegeven tijden zijn het gemiddelde van 3 runs, waarbij de tijd in hele seconden is gemeten. Elke run werd het programma opnieuw gestart. Het percentage is berekend op basis van het gemiddelde, afgerond op één cijfer achter de komma.

| | **ADEPTS 1.00.0004** | **ADEPTS 1.00.0005** | | **ADEPTS 1.00.0006** | |
|---|---|---|---|---|---|
| **Omschrijving test** | **tijd in seconden** | **tijd in seconden** | **percentage t.o.v. versie 1.00.0004** | **tijd in seconden** | **percentage t.o.v. versie 1.00.0004** |
| Tijd vanaf opstarten in explorer (d.m.v. druk op [Enter]) tot het hoofdscherm goed zichtbaar is, zonder selectie. | 4.0 | 5.0 | 125 % | 4.7 | 117 % |
| Tijd vanaf opstarten in explorer (d.m.v. het op de executable slepen van de testselectie) tot het hoofdscherm goed zichtbaar is, met testselectie. | 4.0 | 4.7 | 117 % | 4.3 | 107 % |
| Selectie van alle klassen in het linker stofselectiescherm van de testselectie. Van druk op knop tot klassen scherm in beeld. | 46.3 | 1.7 | 4 % | 1.7 | 4 % |

| | | | | | |
|---|---|---|---|---|---|
| Selectie van alle stoffen behorende tot de klasse *furans* in het linker stofselectiescherm van de testselectie. Van druk op knop, met *furans* geselecteerd, tot het voortgangscherm weg is. | 47.0 | 1.0 | 2 % | 1.0 | 2 % |
| Overbrengen van de furans naar selectie. Van druk op knop tot laatste stof in selectie. | <1 | <1 | - | <1 | - |
| Selectie van alle CAS Nummers in het linker stofselectiescherm van de testselectie (in de selectie de furans plus de oorspronkelijke stoffen). Van druk op knop tot CAS scherm in beeld. | 48.7 | 1.3 | 3 % | 1.3 | 3 % |
| Selectie van alle klassen in het rechter stofselectiescherm van de testselectie. Van druk op knop tot klassen scherm in beeld. | 2.0 | <1 | - | <1 | - |
| Selectie van alle stoffen behorende tot de klasse *furans* in het rechter stofselectiescherm van de testselectie. Van druk op knop, met *furans* geselecteerd, tot het voortgangscherm weg is. | 2.7 | 0.7 | 26 % | 0.7 | 26 % |
| Eerste keer tonen testselectie (zonder *furans*), vanaf druk op knop *show* tot de toolbar weer enabled is. | 25.3 | 13.0 | 51 % | 16.7 | 66 % |
| Tonen testselectie (zonder *furans*) nadat alle *Exp ID* kolommen op onzichtbaar zijn gezet in het constraintscherm. Vanaf druk op OK tot toolbar weer enabled is. | 29.7 | 1.0 | 3 % | 1.0 | 3 % |
| Tonen testselectie (zonder *furans*) nadat het format voor *Minimum Biodegradation* op 0.0 gezet is in het constraintscherm. Vanaf druk op OK tot toolbar weer enabled is. | 30.0 | 18.0 | 60 % | 26.0 | 87 % |
| Tonen testselectie (zonder *furans*) nadat de constraint voor *Half time Biodegradation* op >100 is gezet in het constraintscherm. Vanaf druk op OK tot toolbar weer enabled is. | 22.3 | 16.0 | 72 % | 21.3 | 96 % |

## 2.2 Foutieve referenties Henry Constante

Dit betreft probleemnummers 7. Ten onrechte was de referentie voor log Kow bij import uit de BKH tabellen toegekend aan alle fysische parameters. Na hernieuwde import geldt de referentie uitsluitend voor log Kow.

## 2.3 Run-time errors

Dit betreft probleemnummer 8. Op een 'schone' test PC (uitsluitend Windows 95 geinstalleerd) waren de problemen in versie 1.00.0006 niet meer aanwezig.

## 2.4 Foutieve preview

Dit betreft probleemnummer 10. Het probleem is helaas niet op te lossen zonder over te gaan op een nieuwere versie van de vsPrinter control uit de VideoSoft VsView component (VSVIEW3.OCX). De nieuwe control heeft echter niet dezelfde werking als de gebruikte versie 3, zodat geheel nieuwe code nodig zou zijn om het probleem te verhelpen. Het probleem wordt niet voldoende ernstig geacht om deze werkzaamheden in garantiekader te verrichten.

## 2.5 Ontbrekende/foutieve eenheden

Dit betreft probleemnummer 12. De meta-data tabel is gecorrigeerd volgens de gegevens aangeleverd door RIZA en BKH.

## 2.6 Tox-water uitgezet tegen Tox-terrestrisch.

Dit betreft probleemnummer 20. Omdat de begeleidingsgroep verdeeld was over het nut van deze mogelijkheid is de oude situatie gehandhaafd.

## 2.7 Vreemde weergave kolombreedte.

Dit betreft probleemnummer 21. Het probleem is niet te reproduceren op diverse testconfiguraties bij WL | Delft Hydraulics.

## 2.8 Compartiment toekenning.

Dit betreft probleemnummer 23. De toekenning is op een aantal punten veranderd. Het resultaat is meer in overeenstemming met wat de gebruikers verwachten.

## 2.9 Afgeleide kolommen worden niet als zodanig herkend.

Dit betreft probleemnummer 24. Door aanpassing van de meta-data tabel is dit verholpen.

## 2.10 Uitlijning kolommen

Dit betreft probleemnummer 25. Het probleem is niet te reproduceren op diverse testconfiguraties bij WL | Delft Hydraulics.

## 2.11 Scheidingsteken export

Dit betreft probleemnummer 26. Bij export wordt nu gebruik gemaakt van het teken dat de gebruiker ingesteld heeft via 'Landinstellingen' in het Windows configuratiescherm.

# 3 Conclusie

De performance van versie 1.00.0006 ten opzichte van versie 1.00.0004 is flink verbeterd, vooral voor de selectie van de stoffen via *klassen* of *CAS Nummer*. Voor het **weergeven** van de *Selectie* is de verbetering minder spectaculair, maar gezien de complexiteit van het opzoeken van de gegevens mocht dit ook verwacht worden. Wel is grote winst behaald bij het opnieuw weergeven van een *Selectie* als er geen noodzaak is de gegevens opnieuw op te zoeken of te formatteren (dus als er geen veranderingen zijn in de *Constraints* of de opmaak): dit neemt nog maar ca. 3 % van de tijd in beslag die eerst nodig was. De verbetering van probleemnummers 12 en 24 die plaatsvond tussen versie 1.00.0005 en versie 1.00.0006 had, zoals uit de toename van het aantal raadplegingen van de meta-data tabel was te verwachten, een gering negatief effect op de performanceverbetering.

De overige problemen die onder de garantiebepalingen vielen zijn verholpen, met uitzondering van de Print Preview (probleemnummer 10, hoofdstuk 2.4), waar de problemen intern in de gebruikte component zitten (VideoSoft VsView OCX) en de problemen die niet reproduceerbaar waren op de WL | Delft Hydraulics testmachine (probleemnummers 21 en 25, hoofdstukken 2.7 en 2.10).

# ADEPTS

## Database Management Manual

André Hendriks

May, 2000

WL | delft hydraulics

| | |
|---|---|
| CLIENT: | Ministry of Transport, Public Works and Water Management<br>Directorate-General of Public Works and Water Management<br>National Institute for Coastal and Marine Management / RIKZ<br>Institute for Inland Water Management and Wastewater Treatment / RIZA |
| TITLE: | ADEPTS Database Management Manual |

ABSTRACT:

This manual discusses two aspects of the management of the ADEPTS database: security and addition of new experiment tables.

REFERENCES:

| VER. | ORIGINATOR | | DATE | REMARKS | REVIEW | | APPROVED BY | |
|---|---|---|---|---|---|---|---|---|
| 1.00 | A. Hendriks | | november 23, 1999 | | | | Peter Glas | |
| 1.00.06 | A. Hendriks | | may 24, 2000 | | P. Gijsbers | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PROJECT IDENTIFICATION: | | R3300 | | | | | | |
| KEYWORDS: | | | | | | | | |
| CONTENTS: | TEXT PAGES | 6 | TABLES | 0 | FIGURES | 0 | APPENDICES | |
| STATUS: | □ PRELIMINARY | | □ DRAFT | | ☒ FINAL | | | |

# 1 Introduction

ADEPTS, an acronym for "**A** **D**atabase for **E**nvironmental **P**roperties of **T**oxic **S**ubstances", is an application and database for access to information about Physico-Chemical and Ecotoxicologic parameters and substances. This manual is written for the ADEPTS database manager, who has to add end-users to the database system or has to add information to the database.

The ADEPTS database, `ADEPTSDB.mdb`, contains the data for the ADEPTS application. It is normally situated in the folder `C:\Program Files\ADEPTS\`, but the end-user may change this during the installation. This manual discusses two aspects of the management of this database: security and addition of new experiment tables.

Security is of importance for several reasons:

- The collection of the data in the database was a time-consuming and thus expensive task. So this data represents a certain value, which must be protected from illegal use.
- If users make changes to the data without proper knowledge of the database structure, they may cause corruption of the database and the application.

Security is discussed in chapter 2.

Addition of new parameters or experiments is possible without altering or recompiling the ADEPTS executable. However, some rules must be followed to maintain the integrity of the application. These rules are discussed in chapter 3.

# 2 Security

In release 1.00 of ADEPTS the database `ADEPTSDB.mdb` is **not** secured by a workgroup file. The workgroup file `system.mdw`, which contains information about users of the database, must however be present in the same folder as the database file. Version 1.00 of the application does not use the information in this file, but does check the presence of this file.

The database **is** secured by a password, which is `BKH3WL2` (case-sensitive).

# 3 Adding experiment tables

To add a new parameter to ADEPTS the following steps must be taken:

1. All basetables that the experiment uses, and that are not already present for other experiments, must be imported into the database file. Basetables must have the key (the field the relation is based on) as the **first field** in the table definition.
2. New information for existing base tables is added to these tables. This includes new substances.
3. The new experiment table must be imported into the database file. The new table must follow some guidelines explained in paragraph 3.1 below.
4. The relations with existing and new basetables must be added to the database.
5. The table parameters must be updated with information about this new experiment. This is explained in paragraph 3.2.
6. The table metadata must be updated with information about the fields in the new experiment table. This is explained in paragraph 3.3.

## 3.1 Guidelines for experiment tables

Each experiment table must follow the next guidelines.

- The **first** field in the table contains a long integer with the unique experiment number (primary key). The name of the field is free, but the preferred name is 'Experiment ID'. This field may not be left empty (or null).
- A field 'Substance ID' of type long integer is present, and this field contains the identifier of the substance in the record (foreign key). The 'Substance ID's are as given in the table 'Substances'. New substances **first** have to be added to this table (see point 2 in the 6-step list at the start of this chapter).
- A field 'EnvSystem ID' must be present, and filled with the codes as defined in the table 'EnvSystems' or null values (foreign key). New ID's may be added if needed (see point 2 in the 6-step list at the start of this chapter).

## 3.2 The parameters table

For **each** EnvSystem present in the new table, one or more records in the table 'Parameters' have to be added. Two cases can be distinguished:

1. An experiment table is added for a new EnvSystem, but the parameter already existed.
2. A whole new parameter is added.

In the first case records have to be added to the parameter table, that use the same value for the field 'Parameter ID' as the already existing tables, in the second case records have to be added that have a new, **unique**, identifier for this parameter.

The field 'EnvSystem ID' has to contain an ID as defined in the table 'EnvSystems'. 'Parameter ID' and 'EnvSystem ID' together form the unique primary key in the parameters table. This means that there can be only one table that contains information about a specific parameter for a specific compartment (EnvSystem).
The field 'Description' gives the name of the parameter as presented to the user. If the addition concerns an already existing parameter, the new name should be exactly the same as the existing name.
The field 'Table ID' gives a unique integer to identify the table with. As long as it is unique within this table, the value is not important.
The field 'Parameter table' gives the name of the table in the database.

## 3.3 The metadata table

In the table 'Metadata' records may be added for the new table. A new record should be added for each field in the new tables that has

1. a unit which you want to show to the user
   and/or
2. contains a pointer to a value in a lookup table instead of just a value **and** you want the end-user to see the looked-up value (and not the integer). For the field 'Substance ID' this is **not** necessary, the application will always replace the 'Substance ID' by the complete information about the substance. For example, the new table contains the field 'EnvSystem ID' which contains a long integer, a foreign key into the table 'EnvSystems'. In the table 'EnvSystems' there are fields 'EnvSystem ID' and 'Description'. The meaning of the integer in the field 'EnvSystem ID' in the new table is looked up in the table 'EnvSystems' and the description is displayed, not the integer.

The field 'Tablename' should contain the name of the new table.
The field 'Fieldname' contains the name of the field.
If you want to specify a unit (case 1 above) the field 'Unit' contains the unit.
In case you want to indicate the field is looked-up in a basetable (case 2 above), field 'Lookup' contains 'yes' and field 'LookupTable' contains the name of the basetable.

# ADEPTS Application Design

André Hendriks

May, 2000

WL | delft hydraulics

CLIENT:

Ministry of Transport, Public Works and Water Management
Directorate-General of Public Works and Water Management
National Institute for Coastal and Marine Management / RIKZ
Institute for Inland Water Management and Wastewater Treatment / RIZA

TITLE:

ADEPTS Application Design

ABSTRACT:

Technical design document for ADEPTS

REFERENCES:

| VER. | ORIGINATOR | | DATE | REMARKS | REVIEW | | APPROVED BY | |
|---|---|---|---|---|---|---|---|---|
| 1.00 | A. Hendriks | | november 18, 1999 | | | | | |
| 1.00.06 | A. Hendriks | | may 24, 2000 | | P. Gijsbers | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

| | |
|---|---|
| PROJECT IDENTIFICATION: | R3300 |
| KEYWORDS: | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| CONTENTS: | TEXT PAGES | | TABLES | | FIGURES | | APPENDICES | |

STATUS: □ PRELIMINARY □ DRAFT ☒ FINAL

# 1 Introduction

This document gives some insight in the design of the ADEPTS application. It is not meant to be a complete functional or technical design document, but covers some topics that might be relevant to a programmer who has to make changes to the application or has to solve problems that occur during the use of the application. During the start of the development, some aspects were covered in "Datamodel and Basic Design DAPHNE" [WL 1999], but as the application and the database developed the original ideas were adapted. This document extends and partly succeeds that document. This document covers Adepts 1.00.04 of November 1999. Until october 1999 the Adepts project was called by its working name, Daphne.

# 2 Design philosophy

## 2.1 Packages

Adepts 1.00 consist of one executable. This however contains three packages which may be compiled into separate executables and/or Dynamic Link Libraries (active-X components):

- The Adepts User Interface (AdeptsUI)
- The Data Centric Classes (AdeptsDC)
- The User Centric Classes (AdeptsUC)

In the development environment and the Visual SourceSafe database, these packages are stored in separate folders. Halfway during the development phase a test was conducted where the packages were compiled separately and the `AdeptsDC`[1] (Active-X exe) package was run on an Windows NT server and the `AdeptsUC` (Active-X DLL) and `AdeptsUI.exe` were run on a Windows 95 PC. This test was successful, but the performance left something to be desired. For performance reasons therefore from that moment on all three packages were developed inside one executable: `AdeptsA.exe`. A separate VB project AdeptsA was made. The other projects are saved in the SourceSafe Database, but might not be up-to-date with respect to the classes used in the application. A compile-time constant `ADEPTSALL` is used to call `DCMain` and `UCMain` if one executable is generated, otherwise this constant has to be undefined and `DCMain` and `UCMain` have to be renamed to `Main` (it is unfortunately impossible in VB to define another start-up routine than *`Main`*.)

The packaging of ADEPTS follows the principals as described in [Lhotka, 1997].

### 2.1.1 AdeptsDC package

This package contains all classes that communicate directly with the database. This is the only package that 'knows' which database ADEPTS is connected to[2]. It uses it's own initialisation file `AdeptsDC.ini` for this purpose. The classes in the `AdeptsDC` package are described in chapter 3.1. The `Buffer` class is shared with `AdeptsUC`, as this is used to share data between `AdeptsDC` and `AdeptsUC`. The modules which define 'object properties' are also shared because the persistence classes in `AdeptsDC` and the normal classes in `AdeptsUC` have to use the same definitions.

[1] *Then still called DaphneDC*

[2] *Other packages generate SQL statements to send to DaphneDC, if the database is changed to one using another SQL dialect, these might need slight modification.*

### 2.1.2 AdeptsUC package

This package contains the *business logic* for ADEPTS. It contains the classes that are not part of AdeptsUI (forms) or the AdeptsDC. The Buffer class is shared with AdeptsDC, as this is used to share data between AdeptsDC and AdeptsUC. The modules which define 'object properties' are also shared because the persistence classes in AdeptsDC and the normal classes in AdeptsUC have to use the same definitions.

### 2.1.3 AdeptsUI package

This package contains the forms used in the user interface. It does **not** directly use objects from AdeptsDC.

## 2.2 Application flow

The central object in the application is a Selection.

When the application starts, frmMain creates a Selection, and a reference to this object is passed to other forms that view or edit (parts of) the Selection. Most forms have a *Property Set* for this purpose. The Selection object does **not** 'store' Substance or Experiment objects, because if it did, it would have to store large amounts of data. Instead it maintains a number of NameIDCollection objects that contain the names and ID's of substances in the Selection, experiments in the Selection and envsystems in the Selection. Furthermore frmMain creates three **global** NameIDCollections that are passed to the Selection it creates, one with a list of available substances in the database, one with available parameters and one with available envsystems. These are used by a Selection to find the names associated with an ID (for substance, parameter or envsystem). If these global NameIDCollections are not present the Selection uses the ID as name, but this will not be appreciated by the user.

FrmMain also creates a **global** ExperimentInfoCollection, that is passed to all Experiments that are created in frmSelectionEdit. This collection contains the information about all experiments in the database and their fields. It **must** be present before an Experiment can be loaded.

From the Main Window (frmMain) four other forms are called:

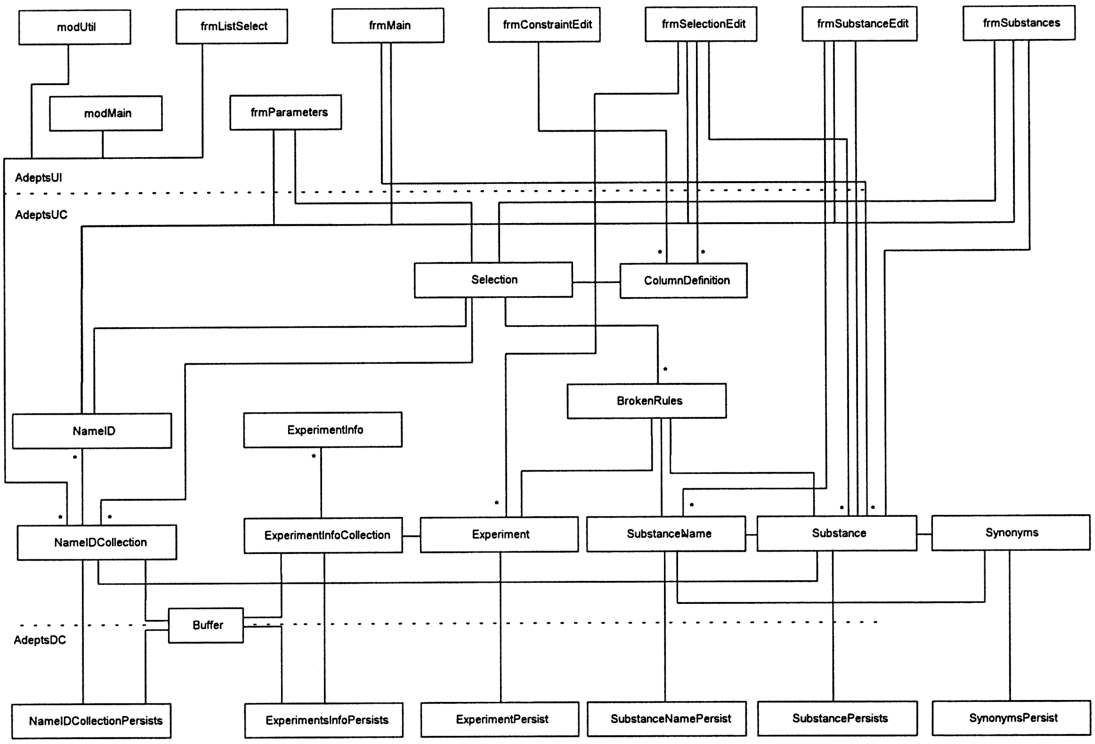
- frmSubstances to edit the list of substances in the Selection;
- frmParameters to edit the list of parameters in the Selection;
- frmSelectionEdit to view and edit the Selection and
- frmAbout to display information about the application.

FrmSelectionEdit also creates/fills other Selection properties, like the ColumnDefinition objects. From frmSelectionEdit the user may either go to frmPrintPreview for previewing the Selection as it would be printed, or to frmConstraintEdit to edit the properties of individual ColumnDefinition items from the Selection.

By cloning, Selections may also be created in frmSelectionEdit (a new frmSelectionEdit with its own Selection is created when a clone is made). They may be passed back to frmMain (*Property Set* Object) to allow their substances, parameters and envsystems to be edited.

# 3 Classes

In the next Class Diagram all the classes are depicted:



The diagram is simplified, in that classes may have several associations with the same other class, each with different cardinalities. E.g. the `Selection` class has an association 'Available Substances' with the `NameIDCollection` class, where each `Selection` object has 0 or 1 'Available Substances' `NameIDCollection` object associated with it. This same object may be associated with 0 or more `Selections`. Another association of a `Selection` object with the `NameIDCollection` class is for the 'actual substances' in the `Selection`. This instance is only associated with one `Selection`, and each `Selection` has exactly one such `NameIDCollection` associated with it.

Various attributes of objects are defined as *String*, but stored in *User Defined Types* of type *String * N,* where *N* is a positive integer. When setting these attributes an error is generated when you attempt to store a string longer than *N* characters. The length of the strings in these *UDT's* is defined in the 'Properties'-modules. When these fixed length strings are returned by the *Property Get* the leading and trailing spaces are removed using `Trim$()`.

The next paragraphs describe all classes and their public attributes and operations.

## 3.1 Classes in the AdeptsDC package

ADEPTS 1.00 is an application that only reads from the database. Some objects have operations for saving (as described in [Lhotka, 1997]), but these were not tested extensively. For information about the design of the objects, especially the `Fetch`, `GetState` and `SetState` methods, we refer to [Lhotka, 1997].

### 3.1.1 Buffer

This class supplies the buffer for communication between the `AdeptsDC` objects `NameIDCollectionPersist` and `ExperimentsPersist` and the `AdeptsUC` objects `NameIDCollection` and `ExperimentInfoCollection` respectively.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Item | GL | String | lngIndex | Sets/Gets the lngIndex-th item from the buffer. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Add | - | strData | Adds strData to the buffer. |
| Count | Long | - | The number of items in the buffer. |
| GetState | String | - | Gets the state of the internal datastructure. |
| Initialize | - | intLength, lngEst | Initialise the buffer for use with app. lngEst items of length intLength |
| Length | Long | - | Return the length of an element. |
| SetState | - | strBuffer | Sets the state of the internal datastructure. |

### 3.1.2 ExperimentPersist

Deals with the persistence of the Experiment object.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| SubstanceIDField | G | Long | - | The column position of the SubstanceID field. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| DeleteObject | - | strTable, lngID | Delete an experiment from the table. |
| Fetch | - | strTable, lngID, objExperiment-InfoCollection | Fetch the data for an experiment. Use objExperiment-InfoCollection to determine the 'columns' for this experiment. |
| GetState | Variant | - | Gets the state of the internal datastructure. |
| Save | - | - | Save the data. |
| SetState | - | varBuffer | Sets the state of the internal datastructure. |

### 3.1.3 ExperimentsInfoPersist

Deals with the persistence of the ExperimentInfoCollection object.

**Attributes**

none

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Fetch | - | - | Fetch the data for all experiments. |
| GetState | String | - | Gets the state of the internal datastructure. |

### 3.1.4 NameIDCollectionPersist

Deals with the persistence of the NameIDCollection object.

**Attributes**

none

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Fetch | - | strSQL, strFieldID, strFieldName. | Fetch the collection from database. StrSQL contains the SQL query that gives the correct set of names and IDs, the field containing the ID is given in strFieldID and the Field containing the name in strFieldName. Both fieldnames may be empty. If strFieldID is empty, the first field in the table is taken, if strFieldName is empty, Str$(strFieldID) is returned as the name. |
| GetState | String | - | Gets the state of the internal datastructure. |

### 3.1.5 SubstNamePersist

Deals with the persistence of the SubstName object.

**Attributes**

none

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| DeleteObject | - | lngID | Delete an object from the table. |
| Fetch | - | lngID | Fetch the data for an name with SubstName ID lngID. |
| FetchPrimary | - | lngID | Fetch the primary name for the substance with ID lngID. |
| GetState | String | - | Gets the state of the internal datastructure. |
| Save | - | - | Save the data. |
| SetState | - | strBuffer | Sets the state of the internal datastructure. |

### 3.1.6 SubstancePersist

Deals with the persistence of the Substance object.

**Attributes**
none

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| DeleteObject | - | lngID | Delete a substance from the table. |
| Fetch | - | lngID | Fetch the data for a substance. |
| GetState | String | - | Gets the state of the internal datastructure. |
| Save | - | - | Save the data. |
| SetState | - | strBuffer | Sets the state of the internal datastructure. |

### 3.1.7 SynonymsPersist

Deals with the persistence of the Synonyms object.

**Attributes**
none

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Fetch | - | lngID,<br>lngNameIDs() | Fetch the synonyms for a substance. |

## 3.2 Classes in the AdeptsUC package

ADEPTS 1.00 is an application that only reads from the database. Some objects have operations for saving (as described in [Lhotka, 1997]), but these were not tested extensively. For information about the design of the objects, especially the `ApplyEdit`, `BeginEdit`, `CancelEdit`, `Load`, `GetState` and `SetState` methods and `IsDeleted`, `IsDirty`, `IsNew` and `IsValid` properties, we refer to [Lhotka, 1997].

### 3.2.1 BrokenRules

The BrokenRules object guards if an object is complete. It is described in [Lhotka, 1997].

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Count | G | Integer | - | The number of rules broken. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| RuleBroken | - | strRule, blnBroken | Sets a rule (strRule) to be broken or not. |

**Events**

| name | description |
|---|---|
| BrokenRule | Fires when a rule changes and at least one rule is broken. |
| NoBrokenRules | Fires when a rule changes and no rule is broken. |

### 3.2.2 Buffer

The `Buffer` object is defined in the `AdeptsDC` package **and** the `AdeptsUC` package. See the description in paragraph 3.1.1 for details.

### 3.2.3 ColumnDefinition

The ColumnDefinition object contains information about a column in the display (frmSelectionEdit). ColumnDefinitions are maintained in a Selection if the experiment is not in the collection, so settings are preserved in case the user want to extend his selection with other experiments. Watch the similarities between this object and the ExperimentInfo object.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Active | GL | Boolean | - | Is the definition for a column that is active (present in the current selection)? |
| Constraints | GL | String | - | The constraints for the column. |
| Format | GL | String | - | The formatting (VB string) to apply to the column. |
| Hidden | GL | Boolean | - | Is the column visible? |
| ID | G | String | - | The unique identifier for the column. |
| IsDeleted | G | Boolean | - | See Lhotka. |
| IsDirty | G | Boolean | - | See Lhotka. |
| IsNew | G | Boolean | - | See Lhotka. |
| KeyTyp | GL | String | - | The type of the key-field for the column. |
| PrimaryField | GL | String | - | The primary field for the column. |
| PrimaryTable | GL | String | - | The primary table for the column. |
| PropertyNr | GL | Long | - | The number of the Experiment property this column describes. |
| SecondaryField | GL | String | - | The secondary field (field in look-up table) for the column. |
| SecondaryKey | GL | String | - | The key field in the secondary table (look-up table). |
| SecondaryTable | GL | String | - | The secondary table (look-up table) for the column. |
| Typ | GL | String | - | The type of the column. |
| Unit | GL | String | - | The Unit for the column. |
| Width | GL | Single | - | The width of the column in millimetres. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| ApplyEdit | - | - | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Delete | - | - | See Lhotka. |
| Load | - | intHandle | Load the ColumnDefinition from ASCII file (Selection file) that is open with handle intHandle. |
| Save | - | intHandle | Save the ColumnDefinition to ASCII file (Selection file) that is open with handle intHandle. |

### 3.2.4 Experiment

An actual experiment.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| ExperimentID | G | Long | - | The ID of the experiment. |
| ExperimentInfo Collection | S | Experiment-Info-Collection | - | Set a reference to the information about all experiments. Must be called prior to loading. |
| IsDeleted | G | Boolean | - | See Lhotka. |
| IsDirty | G | Boolean | - | See Lhotka. |
| IsNew | G | Boolean | - | See Lhotka. |
| IsValid | G | Boolean | - | See Lhotka. |
| NrProperties | G | Long | - | The number of properties for this experiment. |
| Property | G | Variant | lngIndex | Property number lngIndex. |
| SubstanceID | G | Long | - | The ID of the substance as given in the database. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| ApplyEdit | - | - | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Delete | - | - | See Lhotka. |
| Load | - | strTable, lngID | Load from database. Set ExperimentInfoCollection first! |

**Events**

| name | description |
|---|---|
| Valid | Fires when the validity of the object changes. |

### 3.2.5 ExperimentInfo

Information about a field in an experiment. Watch the similarities between this object and the ColumnDefinition object.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| ID | G | String | - | The unique identifier for the info. |
| IsNew | G | Boolean | - | See Lhotka. |
| KeyTyp | GL | String | - | The type of the key-field for the column. |
| LookUp | GL | Boolean | - | Is it a field that is looked-up in a look-up table? |
| PrimaryField | GL | String | - | The primary field for the column. |
| PrimaryTable | GL | String | - | The primary table for the column. |
| SecondaryField | GL | String | - | The secondary field (field in look-up table) for the column. |
| SecondaryKey | GL | String | - | The key field in the secondary table (look-up table). |
| SecondaryTable | GL | String | - | The secondary table (look-up table) for the column. |
| Typ | GL | String | - | The type of the column. |
| Unit | GL | String | - | The Unit for the column. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| ApplyEdit | - | - | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |

### 3.2.6 ExperimentInfoCollection

Collection of `ExperimentInfo` objects, contains information about **all** possible fields in **all** possible experiments.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| IsNew | G | Boolean | - | See Lhotka. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Count | Long | - | The number of elements in the collection. |
| Item | Experiment-Info | Index | Return item Index from the collection. |
| Load | - | - | See Lhotka. |
| NewEnum | IUnknown | - | Enumeration function to be able to use For ... Each |

### 3.2.7 NameID

Combination of a unique ID (long) together with a name (max. 300 characters). Is used extensively throughout the application.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| ID | GL | Long | - | The ID part. |
| IsDeleted | G | Boolean | - | See Lhotka. |
| IsNew | G | Boolean | - | See Lhotka. |
| Name | GL | String | - | The name part. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| ApplyEdit | - | lngSubstanceID | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Delete | - | - | See Lhotka. |

### 3.2.8 NameIDCollection

Collection of `NameID` objects. Is used extensively throughout the application to store lists. E.g. the global list of available substances, the list of substances participating in a `Selection`, but also the list of `Selections` that may be lined up are all `NameIDCollections`.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| IsNew | G | Boolean | - | See Lhotka. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Add | NameID | lngID, strName | Add a NameID object to the collection. Returns a reference to the object added. |
| ApplyEdit | - | - | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Count | Long | - | The number of elements in the collection. |
| Delete | - | - | See Lhotka. |
| Item | NameID | Index | Return item Index by ID or index from the collection. |
| Key | NameID | Index | Return item Index by Name or index from the collection. |
| Load | - | strSQL, strFieldID, strFieldName | Load the collection from database. The arguments are passed to the `NameID-CollectionPersist` object. See there for a description of the arguments.. |
| NewEnum | IUnknown | - | See Lhotka. |
| Remove | - | Index | Remove item Index from the collection (by ID or index). |

### 3.2.9 Selection

The central object in the application. See [WL, 1999].

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| AllEnvSystems | S | NameID-Collection | - | Sets a reference to the collection with all EnvSystems. May be set without calling BeginEdit! |
| AllParameters | S | NameID-Collection | - | Sets a reference to the collection with all Parameters. May be set without calling BeginEdit! |
| AllSubstances | S | NameID-Collection | - | Sets a reference to the collection with all Substances. May be set without calling BeginEdit! |
| Column-Definitions | GS | Collection | - | The column definitions for this selection. |
| Constrained-ExperimentIDs | G | NameID-Collection | strTable, strColMin-Max, strDescMin-Max | Gets a list if IDs and names of experiments that are present in the `Selection` if all constraints are taken into consideration. If the constraints for the table contain 'MIN', 'MEAN' or 'MAX', the arguments strColMinMax and strDescMinMax are filled with the ID of the `ColumnDefinition` that has this constraint and the constraint itself respectively. |
| Constrained-SubstanceIDs | G | NameID-Collection | - | Gets a list if IDs and names of Substances that are present in the Selection if all constraints are taken into consideration. |
| Constraint-Description | G | String | - | One string describing all constraints. |
| EnvSystems | G | NameID-Collection | - | The systems in the Selection. |
| Experiment-Tables | G | NameID-Collection | - | All tables in the Selection. |
| File | GL | String | - | The name of the file where the selection is saved to/read from. |
| IsDeleted | G | Boolean | - | See Lhotka. |
| IsDirty | G | Boolean | - | See Lhotka. |
| IsNew | G | Boolean | - | See Lhotka. |

| | | | | |
|---|---|---|---|---|
| Parameters | G | NameID-Collection | - | The parameters in the selection. |
| ReadOnly | GL | Boolean | - | Is the selection read-only. |
| Substances | G | NameID-Collection | - | The substances in the selection. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| AddEnvSystem | - | lngID, strName | Add system to the collection. |
| AddParameter | NameID | lngID, strName | Add parameter to the collection. |
| AddSubstance | - | lngID, strName | Add substance to the collection. |
| ApplyEdit | - | lngSubstanceID | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Clone | Selection | - | Create a clone of the Selection. |
| Completed | Boolean | - | Is the Selection complete (real selection) or incomplete (profile). |
| Delete | - | - | See Lhotka. |
| Load | - | strFilename | Load a selection from file. |
| MakeDirty | - | - | Sets the IsDirty flag. |
| Remove-EnvSystem | - | lngID | Remove envsystem lngID from the Selection.. |
| Remove-Parameter | - | lngID | Remove parameter lngID from the Selection.. |
| Remove-Substance | - | lngID | Remove substance lngID from the Selection. |
| ResetColumns | - | - | Set all columns in the selection to inactive. |
| Save | - | strFilename | Save the Selection to file. |
| UnMakeDirty | - | - | Resets the IsDirty flag. |
| WellFormed | - | strQuery, Optional lngPoint | Check if query strQuery conforms to the rules defined in Adepts. If not the optional lngPoint points to the first character that is not correct. |

**Events**

| name | description |
|---|---|
| Complete | Fires when the object changes from complete (Selection) to incomplete (Profile). |

### 3.2.10 Substance

An actual substance.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| CAS_Number | GL | String | - | The CAS number |
| Classes | G | NameID-Collection | - | The classes for the substance. |
| EC_Number | GL | String | - | The EC number |
| Formula | GL | String | - | The empirical formula. |
| IsDeleted | G | Boolean | - | See Lhotka. |
| IsDirty | G | Boolean | - | See Lhotka. |
| IsNew | G | Boolean | - | See Lhotka. |
| IsValid | G | Boolean | - | See Lhotka. |
| MolWeight | GL | Double | - | The molecular mass. |
| PrimaryName | GL | String | - | The substance name. |
| PrimaryName-Type | GL | String | - | The type of the name. |
| RiskPhrase1 | GL | String | - | **Not used**. Moved to Norms table. |
| RiskPhrase2 | GL | String | - | **Not used**. Moved to Norms table. |
| SMILES | GL | String | - | The SMILES formula. |
| SubstanceID | G | Long | - | The ID of the substance as given in the database. |
| Synonyms | G | Synonyms | - | The synonyms for the substance name. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| ApplyEdit | - | - | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Delete | - | - | See Lhotka. |
| Load | - | lngID | Load substance with ID lngID. |

**Events**

| name | description |
|---|---|
| Valid | Fires when the validity of the object changes. |

### 3.2.11 SubstName

A substance primary name or synonym.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| IsDeleted | G | Boolean | - | See Lhotka. |
| IsDirty | G | Boolean | - | See Lhotka. |
| IsNew | G | Boolean | - | See Lhotka. |
| IsValid | G | Boolean | - | See Lhotka. |
| Name | GL | String | - | The substance name or synonym. |
| NameType | GL | String | - | The type of the name. |
| Primary | GL | Boolean | - | Is it a primary name? |
| SubstanceID | G | Long | - | The ID of the substance as given in the database. |
| SubstNameID | G | Long | - | The ID of the substance name as given in the database. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| ApplyEdit | - | - | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Delete | - | - | See Lhotka. |

**Events**

| name | description |
|---|---|
| Valid | Fires when the validity of the object changes. |

### 3.2.12 Synonyms

Deals with the persistence of the Synonyms collection object.

**Attributes**
none

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Add | SubstName | - | Add an empty SubstName object to the collection. |
| ApplyEdit | - | lngSubstanceID | See Lhotka. |
| BeginEdit | - | - | See Lhotka. |
| CancelEdit | - | - | See Lhotka. |
| Count | Long | - | The number of elements in the collection. |
| Delete | - | - | See Lhotka. |
| Item | SubstName | Index | Return item Index from the collection. |
| NewEnum | IUnknown | - | See Lhotka. |

## 3.3 Classes in the AdeptsUI package.

### 3.3.1 frmAbout

Shows information about the application and the operating system.

**Attributes**
none

**Operations**
none

### 3.3.2 frmConstraintEdit

This form edits the constraints and visibility/width of columns for a `Selection`.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Canceled | G | Boolean | - | Was the form closed with OK or Cancel? |
| Object | GS | Selection | - | Set/get the `Selection` the form is operating on. |

**Operations**
none

### 3.3.3 frmListSelect

Edits a list NameIDCollection. The user may select one or more names from this list, non-selected items are removed before returning (OK pressed) or the list is returned unchanged (Cancel pressed).

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| List | S | NameID-Collection | - | The list to work on. |
| RightAlign | L | Integer | - | Right align the strings in the listbox in a string at least this many spaces. |

**Operations**

none

### 3.3.4 frmMain

The main window for the application.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Object | GS | Selection | - | The object that the form is using. |
| Selection | L | String (!) | - | The (file)name of the object that the form is using. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Update-ParameterList | - | - | Used to force update of list from frmSelectionEdit, so the screens stay in sync. |
| Update-SubstancesList | - | - | Used to force update of list from frmSelectionEdit, so the screens stay in sync. |

### 3.3.5 frmParameters

The form to edit which parameters are included in a `Selection`.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Selection | S | Selection | - | The Selection the form is using. |

**Operations**

none

### 3.3.6 frmPrintPreview

The form to preview the grid with a `Selection` as it will be printed.

**Attributes**

| name | Get/<br>Set | type | arguments | description |
|---|---|---|---|---|
| Description | L | String | - | The description to print above the grid in the print. |
| Object | SG | vsFlexArray | - | The grid to print. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| Redraw | - | - | Redraw the preview window. |

### 3.3.7 frmPrintSetup

The form to alter the printer and the page set-up.

**Attributes**

| name | Get/<br>Set | type | arguments | description |
|---|---|---|---|---|
| Canceled | G | Boolean | - | Was the form cancelled? |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| UpdatePreview | - | - | Update the preview. |

### 3.3.8 frmProgress

Shows a progress bar during time-consuming operations.

**Attributes**
none

**Operations**
none

### 3.3.9 frmSelectionEdit

Show a grid with the `Selection` and edit the properties of the `Selection`. Column 0 is invisible, but contains identifiers that are unique for a combination of substance and experiments. These make it possible to line-up `Selections`.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Object | GS | Selection | - | The Selection the form is using. |
| RowIDs | G | Collection | - | The unique identifiers for each row of the Selection in this form. |

**Operations**

| name | returntype | arguments | description |
|---|---|---|---|
| AddEmptyRow | - | strID | Add an empty row with a unique identifier. |
| LineUp | - | - | LineUp this Selection on the unique identifiers. |

### 3.3.10 frmSubstanceEdit

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| EditMode | GL | Boolean | - | Determines if the form allows the user to edit the `Substance` or not. Should always be False in version 1.00. |
| Object | GS | Substance | - | The `Substance` the form is using. |

**Operations**
none

### 3.3.11 frmSubstances

This form edits the list of substances for a `Selection`.

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| Selection | S | Selection | - | The `Selection` the form is using. |

**Operations**

none

### 3.3.12 frmSynonymEdit

**Attributes**

| name | Get/ Set | type | arguments | description |
|---|---|---|---|---|
| EditMode | GL | Boolean | - | Determines if the form allows the user to edit the `SubstName` or not. Should always be False in version 1.00. |
| Object | GS | SubstName | - | The `SubstName` the form is using. |

**Operations**

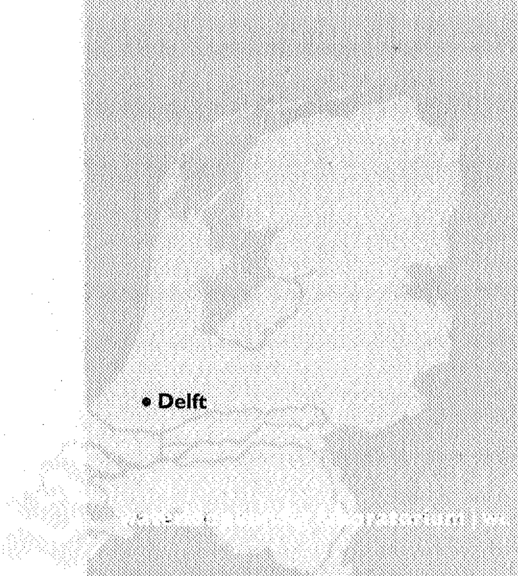
none

# A References

1. Lhotka, Rockford. *Professional Visual Basic 5.0 Business Objects.* Wrox Press Ltd., 1997

2. BKH/WL *Vooronderzoek integratie AQUATOX/AQUAPOL.* BKH Adviesbureau en Waterloopkundig Laboratorium, 1997.

3. WL *Datamodel and Basic Design DAPHNE.* WL | Delft Hydraulics, 1999.



waterloopkundig laboratorium | wl

Rotterdamseweg 185
postbus 177
2600 MH Delft
telefoon 015 285 85 85
telefax 015 285 85 82
telex 38176 hydel-nl
e-mail info@wldelft.nl
internet www.wldelft.nl