

Blaise



Data



Overview

- Blaise Data Provider
- Blaise SQL versus Native SQL
- Record Filters
- Optimizing query performance
- Data access methods in Manipula and API
- Data Conversion methods during install
- New features added to BDIX files from 5.11 on
- History table
- Changing data outside Blaise
- Hospital



Blaise Data Provider

- Blaise 5 uses BDIX files as data files
 - Data Interface
 - DataSource type
 - RDBMS, Text, JSON, XAML, Text
 - Connection information
 - Connection string
 - In case of text files
 - Separator, String Delimiter, et cetera
 - In case of RDBMS
 - Contains definitions for the available objects in the database
 - Tables, Indexes, Triggers, Sequences



Blaise Data Provider

- .NET Data Provider
- Is used to access data in Blaise data files (*.BDIX)
 - Accesses data in BDBX (Sqlite), RDBMS, Json, Text and XML files
- Internally used by Blaise applications, like Data Service, Data Viewer, Manipula and DataLink API
- StatNeth.Blaise.Data.Provider.dll



Blaise Data Provider

- Has objects that are common to a .NET Data Provider
- BlaiseConnection
 - Can be used to open connection to data interface (*.BDIX) and general data interface files (*.BSDI, *.BCDI, *.BADI, etc.)
- BlaiseCommand
 - CommandText
 - Can be used to execute statements against a BlaiseConnection
 - Select statements
 - Native SQL statements
 - Update statement (added in 5.14)



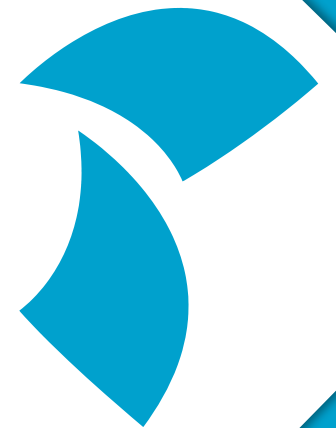
Blaise Data Provider

- BlaiseDataAdapter
 - Can be used to fill a System.Data.DataTable table based on the CommandText that has been specified in a Blaise Command
 - Can be used to fill a Blaise dataset based on the CommandText that has been specified in a Blaise Command
- BlaiseDataReader
 - Can be used to fill a System.Data.DataReader based on the CommandText that has been specified in a BlaiseCommand



Accessing data

Blaise SQL versus Native SQL



Blaise SQL versus Native SQL

Blaise SQL

- SQL understood by Blaise Data Provider
- Is used by all Blaise applications and DataLink API to access BDIX data
- Knows the fields in the associated Blaise data model
- Can access columns that are present in a bdix
- Is data source and table structure independent

Native SQL

- Is native to the underlying database that a BDIX is targetting
- Is not limited to the tables which are present in the bdix



Blaise SQL

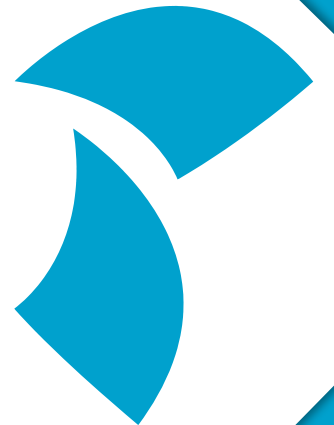
- Supports Select and Update (in 5.14) statements
- Syntax rules:
 - Blaise field names must be fully qualified field names
 - Names can be delimited by ` to escape SQL reserved words
 - Special columns must be surrounded by square brackets
 - [FormID], [ValidationStatus], [SaveStatus]

Using SQL – Methods and functions

- Methods will use the connection as specified in the bdix
- DataLink API
 - IDataLink7.ExecuteNonQuery(string commandText, bool isNative = true)
- Manipula
 - QueryFile.Open(selectStatement)
 - ExecuteQuery(selectStatement, QueryFile)
 - Can be used with InputFile, OutputFile, UpdateFile and SurveyDataFile
 - Blaise datamodel is dynamically generated for QueryFile
 - ExecuteNonQuery(dmlStatement)
 - Can be used in an ActionSetup during a data entry session!

Record Filters

Record Filters and Query Performance



Record Filters

- Way to filter the data and to only get the filtered data back
- Must be specified in Blaise SQL syntax
- Will be translated into an SQL where clause



Record Filter support in Blaise Tools

- DataLink API
 - Delete method
 - Read method to retrieve datasets
- DataEntry API / Apps
 - DownloadCases
 - UploadData
- DataInterface
 - RecordFilter property
- Data Viewer
- Manipula
 - File Settings and SetRecordFilter method
 - DownloadData and UploadData functions



Using Blaise Field Names

- Must be fully qualified field names:

```
Address.Street = 'Kerkweg'
```

```
Person[1].Name = 'John'
```

```
NrOfPeople > 2 and Town = 'Kerkrade'
```

```
(NrOfPeople > 2) or (Town like 'Ams%' and IntervNo in (1,2))
```

- Names can be delimited by `

```
`NrOfPeople` > 4 and `Town` = 'Kerkrade'
```



Using special columns and null / not null

- Special Blaise columns must be delimited by [] :

```
[ValidationStatus] in (0,1)
```

```
[FormID] > 1000
```

```
[SaveStatus] = 'Completed'
```

```
[Mode] in ('CAWI', 'CAPI')
```

- Filtering on null and not null values

- Street is null
- Town is not null
- [Mode] is null



Record Filter: Filtering Key Values

- Always specify which key to filter via [KEYNAME]
- [KeyValue] is a string value and must be delimited by ' '
- Key Values can have several formats:

```
[KEYNAME] = PRIMARY AND [KEYVALUE] = '12'           // Filter on primary key
[KEYNAME] = SECONDARY AND [KEYVALUE] >= '12'        // Filter on secondary key
[KEYNAME] = <keyname> AND [KEYVALUE] >= 'Dennis'    // Filter on key with name <keyname>
[KEYNAME] = PRIMARY AND [KEYVALUE] = '12 14 '       // formatted length keyvalue
[KEYNAME] = PRIMARY AND [KEYVALUE] = '12,14'        // comma separated key value
[KEYNAME] = PRIMARY AND [KEYVALUE] = '12;14'        // key value separated by ;
[KEYNAME] = PRIMARY AND [KEYVALUE] = '12,"Hello world"' // second key value delimited by "
```



Optimizing query performance

- **Good: All items that are in record filter have a column in the database**
 - Corresponding columns will be used in where clause
 - Filter can be applied in the database directly
- **Bad: One or more of the items in record filter do not have a column**
 - Data to be filtered is stored in the data stream only
 - All records must be read because we cannot filter the data in the database itself!
 - Requested record data will be loaded into a ADO.NET DataTable
 - DataTable will be filtered with the specified filter using by using a ADO.NET DataView
 - Filtered data in DataView will be returned



Optimizing query performance

Data Partition Type	Filtering key fields	Filtering non-key fields	Comment
Stream	😊	😞	Key fields have dedicated columns, non-key fields are stored in data stream
Flat, Blocks	😊	😊	Every field has its own column
Flat, No blocks	😊	😊	Every field has its own column
In Depth	😊	😐	Non-key fields must be filtered by using the in-depth data table; can have many rows
Generic Stream	😞	😞	No way to filter based on individual field values; key values are stored as a concatenating string
Generic In Depth	😐	😐	Fields must be filtered by using the in depth data table; can have many rows
Single Table	😊	😊	Every field has its own column



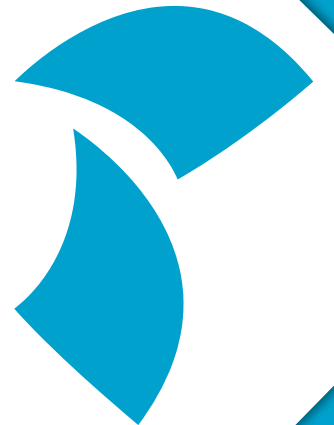
Optimizing query performance

- Recommendation
 - Investigate which record filters that you want to use
 - Add a flat data table to your bdix that contains columns for items that don't have a dedicated column in the database or are stored in a in-depth way only
 - Create indexes on columns to optimize performance even further
 - Blaise Data Provider will use these columns automatically when they are present
 - Populate flat table when the other tables have already data
 - Will be done by automatically by Hospital in 5.14

Demo: add flat table to BDIX



Data Conversion



Install Survey - Data Update Options

- None
- Harmless change update
- Data Conversion Setup
 - Installation package contains an incompatible datamodel and a Manipula data conversion setup
 - Setup is executed to perform the data update
- New option in 5.14: Implicit data conversion
- Last two options have a two step installation process
 - The data conversion is executed on the first install. Once completed, the survey is still running with the old meta and data files.
 - At this point you can look whether the conversion was executed successfully
 - The second install will update the survey and data files at the deploy location with the converted data.

Install survey - Configure installation

Configure installation
Set installation parameters.

Package: C:\Users\g7\Documents\IBUC 2023\Apps\ADB\Flight\Flight.bpkg

Survey: Flight

Installation: Overwrite data : No
 Only when Data is Incompatible
 Yes

Clear sessions : Yes
 No

Data update : None
 Harmless change update. Will only work if your installation package contains a datamodel that has harmless changes as opposed to the installed datamodel. Session data will be preserved if the harmless change update is possible or if the datamodel in the package is compatible with the installed datamodel.
 Manipula Data Conversion Setup
Run an implicit conversion. The existing data will be imported into the Data Interface file (*.bdix) that is present in the package. Session data will be preserved only if the datamodel in the package is compatible with the already installed datamodel.

Help Cancel < Back Next > Install



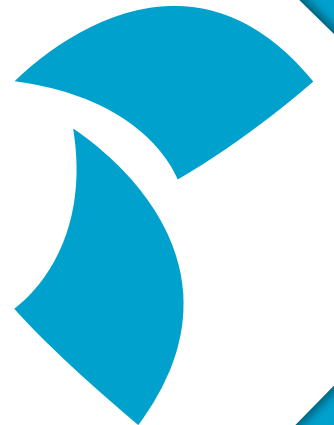
Data Conversion Options

- Implicit data conversion
 - Can be performed regardless whether the update is compatible, harmless or harmful
 - Data will be copied to the new BDIX based on field name
 - Can be used to switch data source
 - For example: from BDBX to SQL Server
 - Can be used to switch data partition type
 - For example: from Stream to SingleTable
 - Session data will be preserved if:
 - the new datamodel is compatible with the old one
 - the new datamodel has harmless changes only
- Can also be executed via DataLink API: `DataLinkManager.ConvertData`



BDIX New Features

Data related new features



New columns

- Goal: to store information about save actions
 - Background:
 - In the past we couldn't tell who/which process saved data
 - Who did what at what time to the data?
 - In order to address this we have added additional columns to the FormInfo table
 - TimeCreated
 - Time that the record was saved for the first time
 - LastModification
 - Time stamp last save
 - IdentityName
 - Name of the user/identity who saved the record
 - SourceInfo
 - Application/service that saved the data
 - Setup name if data is stored by Manipula
 - Columns can also be added to SingleTable bdix in 5.14



Optional feature: add a History Table

- Data record that is going to be saved is also in a history table on each save
 - Allows you to keep track of changes that are made to a record
- Can be set when table definitions are going to be created
 - Adds a history table to the bdix
 - Same structure as FormInfo table + HistoryID and UpdateKind columns
 - <datamodel>_History / Blaise_History
 - DataStream contains complete record data
- Implemented by using database triggers
 - Triggers are created on FormInfo table
 - After Insert, After Update, After Delete
 - SQLite/BDBX, SQL Server, Oracle, PostgreSQL and MySQL
 - Requires Create Trigger/ Drop Trigger privileges

Demo: History table



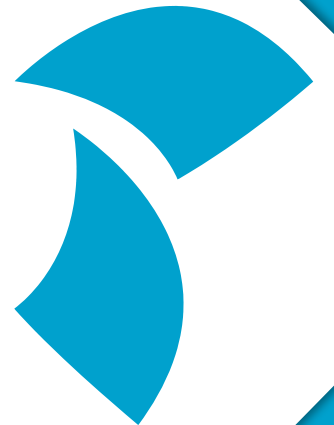
ReadDataStream setting

- Can improve read performance when a non-stream data partition type is being used
 - Reads data stream instead of table data
- Options:
 - At design time:
 - Setting bdix; always read data stream; default = false
 - At runtime:
 - Manipula file setting
 - ReadDataStream = yes
 - DataLink API: ReadDataStream parameter in Read methods



Changing Table Data Outside Blaise

Consequences and actions needed

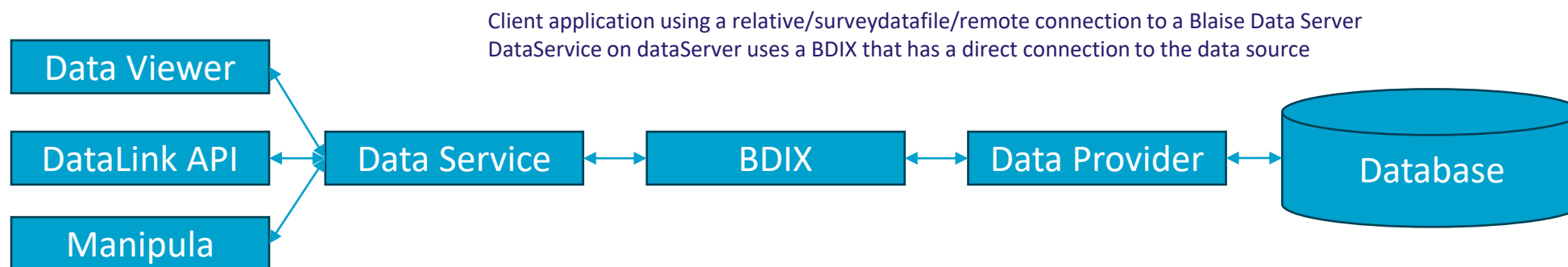
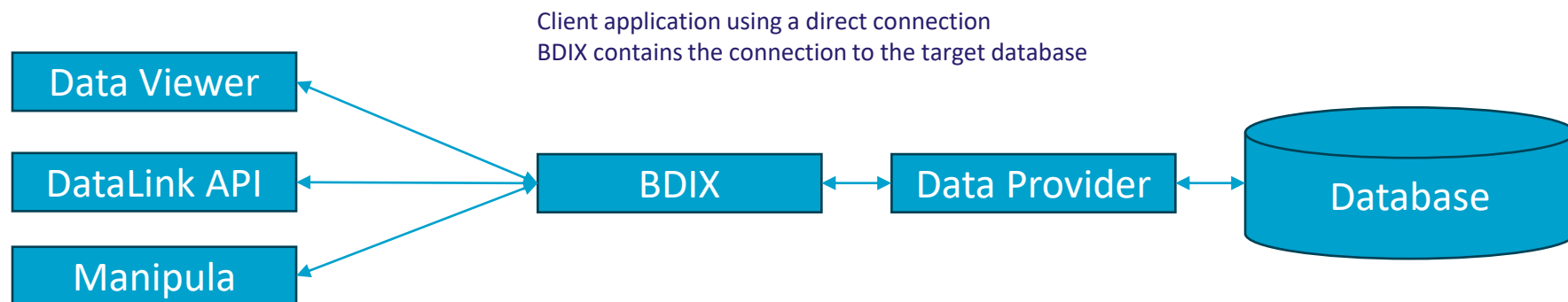


Changing data outside Blaise

- Changes are not always visible in Blaise tools after a change
- Reason: record data is also stored as a byte array in datastream column
- Distinguish between direct access and data service access
 - Direct access: bdix that targets the database directly
 - Has connection string to SQL, MySQL, BDBX, et cetera
 - Data service access: relative bdix files, Remote Datalink and surveydatafiles
 - Have a connection to a Blaise data server



Direct versus DataService Access



Changing data outside Blaise

Direct access

- ReadRecord
 - Data Partition Type is leading and determines how record data is read
- DataSets
 - Data Partition Type is leading and determines how record data is read

Access via Data Service

- ReadRecord
 - Data Partition Type is leading and determines how record data is read
- DataSets
 - 5.13 and earlier always reads binary record stream
 - 5.14 and later will respect Data Partition Type
 - **Fix: Binary record stream has to be synchronized with the changed table data**



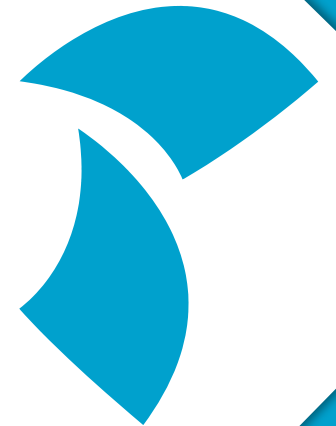
Changing data outside Blaise

- How to fix?
 - Record data in DataStream column must be synchronized with table data
 - Possible ways to do this
 - Manipula with an updatefile which reads and writes all records
 - Hospital tool (5.14)



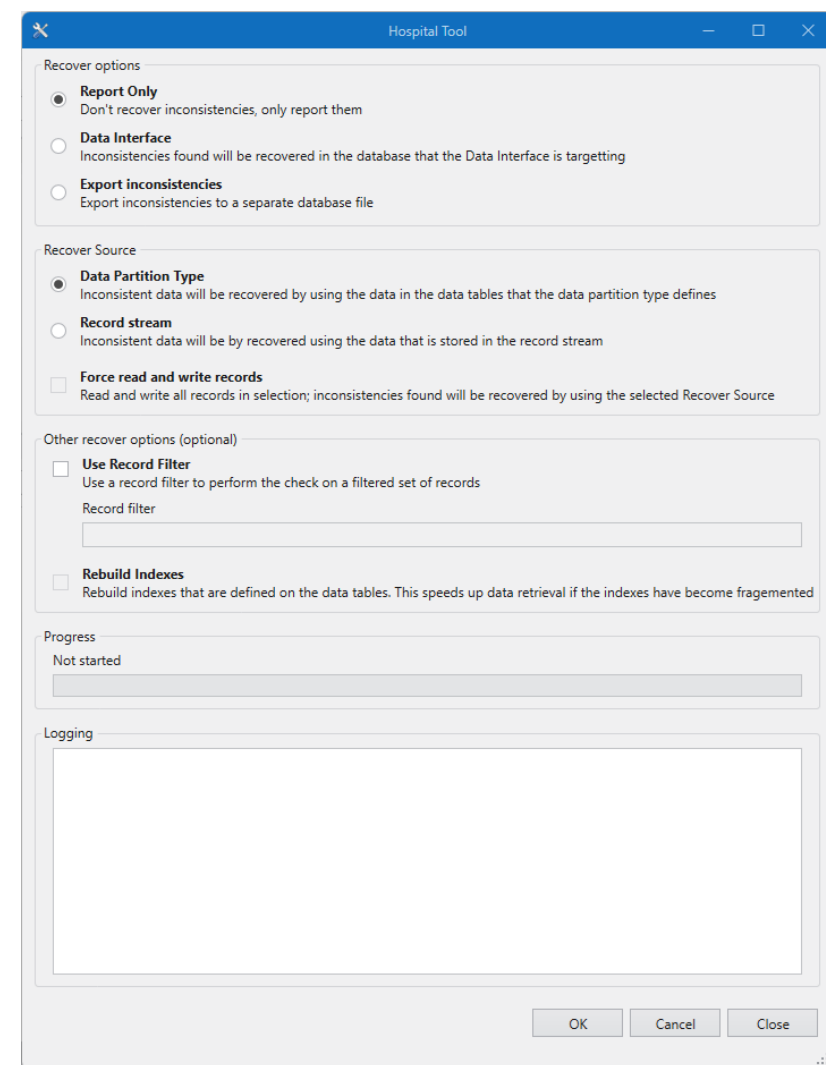
Hospital Tool

Check whether a database is in good condition



Hospital Tool

- Available in ribbon whenever you open a RDBMS based BDIX in Control Centre
- Performs a health check on a BDIX
 - Replaces Data Consistency Check
 - Checks whether the data of a BDIX is consistent
 - Has log and recover options



The screenshot shows the 'Hospital Tool' dialog box with the following sections:

- Recover options:**
 - Report Only**
Don't recover inconsistencies, only report them
 - Data Interface**
Inconsistencies found will be recovered in the database that the Data Interface is targeting
 - Export inconsistencies**
Export inconsistencies to a separate database file
- Recover Source:**
 - Data Partition Type**
Inconsistent data will be recovered by using the data in the data tables that the data partition type defines
 - Record stream**
Inconsistent data will be recovered using the data that is stored in the record stream
 - Force read and write records**
Read and write all records in selection; inconsistencies found will be recovered by using the selected Recover Source
- Other recover options (optional):**
 - Use Record Filter**
Use a record filter to perform the check on a filtered set of records
Record filter:
 - Rebuild indexes**
Rebuild indexes that are defined on the data tables. This speeds up data retrieval if the indexes have become fragmented
- Progress:**
Not started
- Logging:**

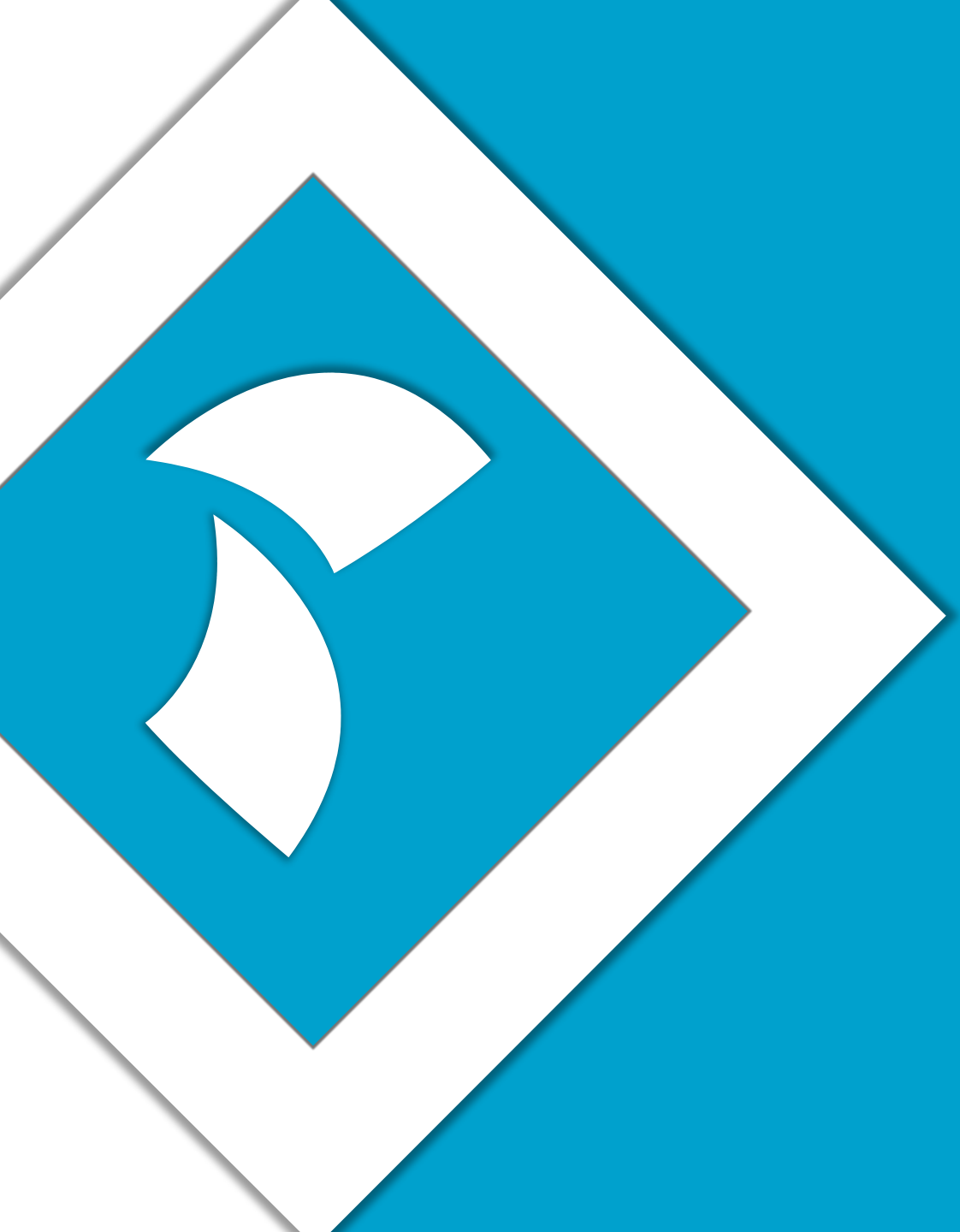
Buttons: OK, Cancel, Close



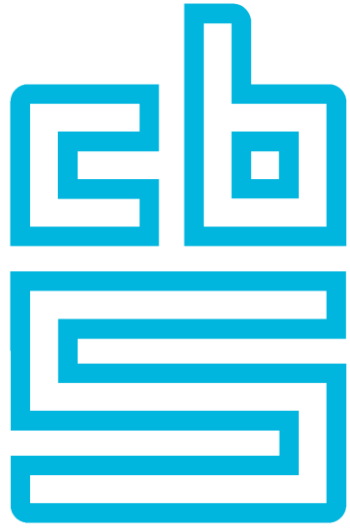
Hospital Tool: Features

- Checks whether data in data tables is consistent with content binary record stream
- Recover options
 - Update record stream with current content of data tables (default)
 - Update data tables with current content of record stream
- Output targets
 - Report only; write inconsistencies found to a log file
 - Recover inconsistencies found directly in target database
 - Export inconsistencies found to a separate BDBX
- Settings
 - Force reading and writing of all records; by default only the records that need a fix will be written
 - Record Filter: perform the check / recover only for a selection of records
- Rebuild indexes option





**Thank you
for your time**



Blaise

Gaining deeper understanding



www.blaise.com



blaise@cbs.nl



[@blaiseCBS](https://twitter.com/blaiseCBS)



[@Blaise5](https://www.youtube.com/@Blaise5)