CMA – Case Management Application

2.6

A (mainly) technical description

DepApp – Manipula

Statistics Netherlands
c/o Blaise
P.O. Box 4481
6401 CZ Heerlen
NETHERLANDS

blaise@cbs.nl

www.blaise.com

Table of Contents

Introduction

This document describes how you can install CMA (**C**ase **M**anagement **A**pplication) on a client device and how to get surveys and cases for those surveys on CMA.

As the basis for the implementation, the specification document from Gina Cheung (SRC University of Michigan) was used. This specification has not been applied strictly-as-provided. There are several reasons for this, the most important ones being:

1. Additional specifications by others:
    a. CMA should do things automatically when possible. For instance, installation / de-installation of a survey should be triggered by the cases for the interviewer using CMA.
    b. It should be possible to see all cases for all surveys in one list in CMA.
2. The strong wish to keep the implementation as agnostic as possible from the surveys and to shift as much as possible of the organisational specific wishes to CMA-specific data models.

Based on the known specification and additional requirements, a design of the architecture was made and CMA implemented.

The first production release of CMA was shipped as version 1.03 with Blaise 5.8.2 on July 16th, 2020. Appendix J contains the version history up to this version

Most important changes compared to version 2.5

This version brings only a limited number of extensions / improvements. The most important are:

1. CMA can now write a trace file of downloaded / uploaded launcher cases. The trace will be written when the CMA option **sWriteSyncTraceFile**=*yes.*
2. A retry has been added for downloading topic data to the client machine. CMA will try to download topic data up to 3 times when the error code -2 is returned by the download.
3. Support has been added for a custom use field in the survey settings XML: **Survey_CustomUse**. The content of this field is accessible in the templates by using $Survey_CustomUse (comparable to $CMA_CustomUse).
4. From now on CMA suppresses the receipt page of a survey by using the command line option *NoReceiptPage:True*.
5. Support has been added for a Manipula setup that needs to be executed during synchronization. This offer you the possibility to do additional downloads / uploads for downloaded / uploaded launcher cases. The setup has access to the trace file written during the synchronization.
6. Support has been added for using the end date in the filter displaying cases in the CMA cases datagrid. When the option **sUseEndDateInFilter**=*yes* a case will no longer be visible in the grid after the value specified in **CMA_EndDate** plus **sDaysToShowAfter** has been passed.
7. The following file related instructions have been added: GET_FILES, PUT_FILES, DELETE_FILE and LIST_FILES.
8. Support has been added for not resetting the **CMA_ForWhom** field when releasing a case. This can be achieved by setting **sKeepForWhomOnRelease**=*yes*. Not resetting allows for better overview of the workload of an interviewer in the CMA admin tool.

Upgrading from CMA 1.0 to CMA 2.6

CMA 2.6 is shipped with Blaise 5.12 and to run it you will need at least Blaise 5.12.1 on the server and for the DepApp (for all three platforms; Windows, Android and iOS).

In CMA 2.6, the launcher database has some changes and some additional fields compared to CMA 1.0. All of the changes and additions are done in a **harmless** way. If you want to upgrade CMA from version 1.0 to version 2.6 you only need to install the multi package **CMA.mbpkg** on top of the existing CMA installation on the server. This can be done while interviewers are actively using CMA on a client device. When an interviewer starts DepApp again or presses the refresh button on the first DepApp screen, DepApp will update CMA on the interviewer's device. CMA will then automatically handle all necessary changes to the local databases, e.g. all databases with *_loc* in the file name that need to be converted because of a changed checksum of the new datamodel will be converted.

Note that once CMA is upgraded it is not easily possible to downgrade.

Upgrading from CMA 1.5 / 2.0 / 2.5 to CMA 2.6

CMA 2.6 is fully compatible with CMA 1.5, 2.0 and 2.5

CMA – The pieces

CMA is based on a Blaise Manipula dialog application which is installed in the Blaise Data Entry app (for short: DepApp) on the client device (Windows, iOS or Android).

The project for this application has the option 'Is app user interface' checked. The result of this is that in the DepApp all surveys are not visible and the DepApp only displays the CMA-application for download and install. Download and install is done by pressing the download button in the DepApp.

After downloading CMA, it is started automatically. It connects to the server and downloads and installs four additional CMA-specific surveys: **CMA_Launcher, CMA_Attempts, CMA_Contactinfo** and **CMA_Logging**.

The survey packages **CMA**, **CMA_Launcher, CMA_Contactinfo** and **CMA_Attempts** need to be installed on a server park that uses the App Run mode setting 'Disconnected'. The package **CMA_Logging** is optional. When installed it will be used by CMA to upload the CMA log file (a maximum of once a day).

By pressing the Sync button in CMA, cases can be downloaded to the client device. Completed cases will be uploaded during a Sync. During Sync all changes to non-completed cases will also be uploaded.

When cases are downloaded for a survey that is not yet installed on the client device, CMA will handle the download and the installation of that survey automatically. Hence, after downloading cases, the interviewer will be able to start interviewing and register attempts.

The Launcher datamodel

The **CMA_Launcher** survey is used to provide cases to CMA. The **Launcher** datamodel has a number of fields that have certain roles in CMA. CMA will use the value of the field **CMA_ContactData** on the lower part of the screen and the value of the field **ContactInfoShort** in the data grid. The content of these fields originates from the **CMA_ContactInfo** datamodel. It is allowed to change contact information for a selected case from within CMA (update button on cases screen). This is handled by starting a data entry session using the **CMA_ContactInfo** datamodel.

Some important fields in the **Launcher** datamodel and their role in CMA:

Name	Description
MainSurveyID	Part of the primary key of the **Launcher** datamodel. The GUID of the survey that the interviewer needs to handle.
ID	Part of the primary key of the **Launcher** datamodel. The primary key of the case. It must be a correctly formatted key for the command line of the Data Entry client. If the primary key of the survey instrument key contains multiple fields use a comma (',') to separate the key values. For detail on the formatting of the key fields see the Blaise Help.
SurveyDisplayName	Name of the survey to display in the survey column in the cases data grid. This field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=ICT Survey,NL=ICT onderzoek. May remain empty.
CMA_StartDate	The first day the interviewer may see the case and start working on it. May remain empty.
CMA_EndDate	The last day the interviewer is allowed to work on this case. May remain empty.

<table>
<tr><td>CMA_CmdLineForEdit</td><td>This field can be used to pass command line parameters to the survey, for instance what CARI settings to use and which layout set to use</td></tr>
<tr><td>CMA_PreLoadForEdit</td><td>This field can be used to pass field values to the topic instrument at start. Note that this will be done using -AssignMode:Always</td></tr>
<tr><td>CMA_ForWhom</td><td>This field contains the identifier of the interviewer for which the case is intended</td></tr>
<tr><td>CMA_Status</td><td>String field that keeps track of the status of the case on the client. The value of this field can be:

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><empty></td><td>Nothing happened yet</td></tr>
<tr><td>Started</td><td>An attempt has been added, interview not started</td></tr>
<tr><td>Interrupted</td><td>Interview started and not completed, attempt added</td></tr>
<tr><td>Completed</td><td>Interview started and completed</td></tr>
<tr><td>Finalized</td><td>Attempt indicates it is final. Partial interview can be present</td></tr>
<tr><td>Added</td><td>The case was added by the interviewer</td></tr>
<tr><td>Reopened</td><td>The case has been reopened for the interviewer</td></tr>
<tr><td>Closed</td><td>The case has been closed because of a close request</td></tr>
<tr><td>Accepted</td><td>The case has been accepted in the CMA_AdminTool</td></tr>
</table>

It is allowed to pre-populate this field with an initial text as long as this text differs from the value mentioned above. In CMA this is handled as ‘nothing happened yet’.</td></tr>
<tr><td>CMA_CaseClosed</td><td>When this field has the value Yes (1) the case is no longer accessible in CMA.</td></tr>
<tr><td>CMA_Location</td><td>String field that is used to keep track of where the case is. It is also used to trigger certain actions for the case, for instance whether the case should be moved to another interviewer, and to report the result of the action (if applicable).

The values of this field can be:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><empty></td><td>Initial value</td></tr>
<tr><td>ONHOLD</td><td>Intended to be used for new cases. This disables the download to the client machine when the CMA_ForWhom has already been set.</td></tr>
<tr><td>CLIENT</td><td>Set by client after download of a case to client device</td></tr>
<tr><td>TRANSFER_REQ</td><td>Request the transfer of a case back to the server. Such a case will be downloaded to the client during synchronize when the field CMA_ForWhom is not equal to the field CMA_InPossession.
CMA_ForWhom = empty can be used to recall a case without assigning it to another interviewer. Note that in case of a group, a transfer request for the parent case triggers a transfer of all cases in the group.</td></tr>
<tr><td>TRANSFER_OK</td><td>Transfer of a case is allowed. Set by client after ‘releasing’ case</td></tr>
<tr><td>SERVER</td><td>Set by client after upload of a completed / finalized case</td></tr>
</table>
</td></tr>
</table>

	DOWNLOAD_REQ	Set by download interceptor but only for cases for which this field is empty. Will later be replaced by client with an applicable value (like TRANSFER_OK or CLIENT)
	RESTORE_REQ	Request to download the case again to the client. This can for instance be used to download the last known data for cases, attempts and topics to a device. Intended to be used in case the previous device got lost / damaged.
	RELEASE_REQ	Request to release the case on the client. This can for instance be used to remove completed cases that no longer need to be present on the client machine. Note that in case of a group, a release request for the parent case triggers a release of all cases in the group. Note that a case can also be released automatically on the client machine by a setting in the CMA settings file x days after the **CMA_EndDate**.
	RELEASED	Indicates that the RELEASE_REQ was successful.
	RELEASE_FAILED	Indicates that the RELEASE_REQ was NOT successful.
	REOPEN_REQ	Request to reopen the case on the client. The value will be reset to CLIENT when successful. Note that in case of a group, a reopen request for the parent case triggers a reopen of all cases in the group.
	CLOSE_REQ	Request to close the case on the client. The value will be reset to CLIENT when successful. Note that in case of a group, a close request for the parent case triggers a close of all cases in the group.
	REFRESHCASE_REQ	Request to refresh fields in the case on the client, The following fields are handled: CMA_StartDate, CMA_EndDate, CMA_Supervisor, CMA_CmdLineForEdit, CMA_PreLoadForEdit, CMA_TimeZone,CMA_AttemptsRoute, CMA_ContactImage, CMA_DetailsTemplate, CMA_CustomUse, CaseNote, CMA_ContactData, ContactInfoShort. Note that the last 3 fields will only be refreshed when not already changed on the client. The value of the **CMA_Location** field will be reset to CLIENT when handled.
	REFRESHDATA_REQ	Request to download the case for the topic database. Note that this will only be done when the case has not yet been changed on the client. The value will be reset to CLIENT when handled.
	REFRESHBOTH_REQ	Does a REFRESHCASE_REQ and REFRESHDATA_REQ at the same time
	HANDLED	To report back that a null-guid file instruction has been handled.
CMA_InPossession	This field contains the identifier of the interviewer that downloaded the case. Initial value is <empty>. Value will be set by the client once it has the case in possession. When **CMA_InPossession** has a value and this value is not equal to **CMA_ForWhom**, a transfer of a case can be requested by populating the **CMA_Location** field with value TRANSFER_REQ.	

CMA_GeoLocation	String field that contains the geolocation at the moment of pressing the *determine geo location* button.
CMA_HappeningsLbl	String field that contains the label of the enumeration code of the field **Happenings** in the last registered **attempt**
CMA_HappeningsStr	String field that contains the text (using the active language) of the enumeration code of the field **Happenings** in the last registered **attempt**
CMA_HappeningsCod	String field that contains the enumeration code of the field **Happenings** in the last registered **attempt**
CMA_AllowSpawning	When set to *yes* CMA allows the user to create additional cases by pressing the create case button. The new ID is based on the original **ID**. When the primary key of the survey has multiple fields and the last field in the **ID** of the case is a numeric value each create adds 1 to the original value. If ID has no comma's then the new ID is the original key plus '-' plus the number of created cases for the case. The number of created cases is tracked in the field **CMA_SpawnCount**. Example: original ID = 123,1 → new ID = 123,2 for first create, 123,3 for second etc. Original ID = X123-45 → new ID = X123-45-1 for first created, X123-45-2 for second etc.
CMA_IsDonorCase	When set to *yes* CMA will not allow interviewing. The case is only used to create new cases when **CMA_AllowSpawning=Yes**
CMA_GroupType	Can be empty or have the value g*Parent* or g*Child*. It is used to handle the cases as a group. It is expected that the ID of the child cases start with the ID of the parent case.
CMA_GroupID	An identifier that is used by CMA to determine what cases are part of the group. The identifier is used to populate the cases group view.
CMA_GroupSort	Used to sort the cases in a group.
CMA_GroupStatus	String field to keep track of the status of the group = *gParent* and all related Child cases. The group status is computed automatically when the content of the field **CMA_Status** of one of the cases in the group is changed. See further for details how the group status is computed. The possible values are the possible values of the field **CMA_Status** plus the status *Done*. This status indicates that all cases of the group have been handled and have as value in **CMA_Status** *Completed* or *Finalized*.
CMA_GroupSummary	String field that contains a summary of the statuses of the group. See further.
CMA_Process	This block field contains information about what happened to the case like the date/time of first download. See below for more details.
CMA_Data	This block contains a blob with the backup of the case data as XML-file and it contains a blob with the backup of the attempts data as XML-file.
CMA_AttemptRoute	String field passed to the attempt datamodel. It can be used to set up different routes in the attempt datamodel.
CMA_AttemptsGUID	GUID of the datamodel to be used to enter information for the attempt. This datamodel needs to be installed in the same server park where CMA is installed. When empty the default attempts datamodel will be used.
CMA_ContactDataGUID	GUID of the datamodel to be used to edit the contact information and the case note. This datamodel needs to be installed in the same server park where CMA is installed. When empty the default cma_contactinfo datamodel will be used.
CMA_ContactImage	This field contains a blob that contains an optional image file that can be displayed in CMA in a separate dialog.

CMA_DetailsTemplate	The identifier for the custom table layout that needs to be used in the details part of the cases screen. In the contact datamodel package you can include text files (using a .bcps file) with the definition of the table using the following name: detailstemplate _< cma_ detailstemplate>_<language_id>.txt. You are also allowed to include them in the CMA package. Example: *detailstemplate_psu_en.txt*. When empty or when the template file is not present, the default template file detailstemplate_cma_<language_id>.txt will be used.
CMA_ContactData	This field contains field-value pairs with data from the contact info datamodel applicable to the case. By default the field is assumed to contain field<TAB>value<TAB> pairs (a TAB character, ASCII 9, as separator between the field and the value and a separator between the field-value pairs). Both separators can be changed through the **sContactDataSep** setting in the CMA settings file.
CMA_CustomUse	This string field does not play a role in CMA but can be used to store additional data for custom use. The content of the field or part of the content of the field can be displayed on the screen as a column in the data grid or in the details part. By default the field is assumed to contain field;value; pairs (a semicolon as separator between the field and the value and as separator between the field-value pairs). Both separators can be changed through the **sCustomDataSep** setting in the CMA settings file.

Block CMA_Process data

This block contains the fields listed below. The fields receive a value in the download interceptor or in CMA.

Name	Description
CreatedDT	When the record was first created
LastChangedDT	When the record was last changed
GeoLocation	Geo-location during the last attempt
FirstDownloaded	When the record was first downloaded and by whom
FirstUploaded	When the record was first uploaded and by whom
LastDownloaded	When the record was last downloaded and by whom
LastUploaded	When the record was last uploaded and by whom
LastAttempt	When the last attempt was made and by whom
FirstInterviewTime	When the interview was started the first time and by whom
LastInterviewTime	When the interview was started the last time and by whom
LastInterviewEndTime	When the interview was ended the last time
TotalInterviewTimeUsed	Total interview time in seconds

There is in general no reason to make any changes to the launcher datamodel.

The CMA_GroupStatus field

The field **CMA_GroupStatus** is computed based on all cases that have an equal **CMA_GroupID** value.

When a subset cases of the group have a value in the **CMA_Status** field: CMA_GroupStatus:= ‘Started’.

When they all have a value in **CMA_Status** field the following algorithm applies:
When there is one case with **CMA_Status** is 'Reopened': CMA_GroupStatus:= 'Reopened', else when there is one case with **CMA_Status** is 'Added' or 'Started': CMA_GroupStatus:= 'Started', else when there is one case with **CMA_Status** is 'Interrupted' and **CMA_CaseClosed** <> *yes*: CMA_GroupStatus:= 'Interrupted', else CMA_GroupStatus:= 'Done'

CMA keeps track of a summary of the status of all cases in the group in the CMA_GroupStatus field. The content is a comma separated list of <char>=<number> pairs. The char can be: #=Total, A= *Added*, S=*Started*, I=*Interrupted*, D=*Interrupted Done*, C=*Completed*, F=*Finalized*, R=*Reopened*, L=*Closed*.

The Attempts datamodel

The attempts survey is used to allow registration of attempts for a specific case by the interviewer. The attempts datamodel has a number of fields that have a certain role in CMA. The attempts datamodel is started automatically after an exit of the topic instrument or it is started directly by the user of CMA to register an attempt. Some important fields in the Attempts datamodel and their role in CMA:

Name	Description
MainSurveyID	The GUID of the survey that this attempt belongs to
ID	The primary key of the case
SeqNr	The sequence number of the attempt
When	String field that contains a date-time stamp of when the attempt was made
GeoLocation	String field that contains the geolocation at the moment of registering the attempt (optional)
Happenings	This field contains the enumeration that gives the possible operational codes. The label of this field is stored by CMA in the cases database in the string field **HappeningsLbl**
HappeningsStr	This field contains the text to be displayed in CMA for this attempt. When not filled, CMA uses the meta-data text (using language EN) of the value of the field **Happenings**
CMA_CaseClosed	When **CMA_CaseClosed=*yes*** the case is ready and can be uploaded to the server when syncing

When starting the attempts datamodel in CMA a number of fields are passed to the data entry session:

- **AfterInterview**. This field has the value Yes when the attempts model is started because a topic instrument is closed.
- **CaseStatus**. This field can have the value None, Partial or Complete.
- **User**. The identifier of the current user of CMA.
- **InterviewStart**. The date/time when the interview was started (if applicable)
- **InterviewEnd**. The date/time when the interview ended (if applicable)
- **Happenings**. The value of the (optional) Happenings field in the topic instrument (if applicable).
- **CMA_AttemptRoute**. This field can be used to support different routes in the attempt data model.
- **CMA_Status**. The value of this field will only be passed to the data entry session when the field is also defined in the attempt datamodel (by default it is not; expected definition a string field of sufficient width). This will allow you to influence what the **CMA_Status** field

becomes when the attempt is finished rather than leaving this up to the logic programmed in CMA. Note that when the outcome in the attempts model of this field is changed, CMA will assign that value to **CMA_Status**. Note also that you should only assign a valid value, so one of the following (case sensitive!): *<empty>, Started, Interrupted, Completed, Finalized, Added, Reopened*.

- **PrevHappeningsCod.** The value of this field will only be passed to the data entry session when the field is also defined in the attempt datamodel (by default it is not; expected definition is *THappenings*). This field will be given the value of **CMA_HappeningsCod** as stored in the launcher case.

For CMA, although the actual value of field **Happenings** does not play a role the field must have a value when the attempt is finished. Only then will the attempt be processed by CMA.
For CMA two things are important:

1. Whether the case is complete or not. This is communicated by the attempt to CMA through the field **CMA_CaseClosed**. If **CMA_CaseClosed** equals ***yes*,** then the case is ready and will be uploaded to the server when a sync is requested.
2. A non-empty value of the **Happenings** field. The code, the label and the string that describes the Happenings choice for the last attempt are stored in the launcher case (fields **CMA_HappeningsCod**, **CMA_HappeningsLbl** and **CMA_HappeningsStr**).

The organisation that uses CMA is allowed to customize the datamodel while making sure that the two points mentioned above are handled correctly.

CMA supports multiple attempts datamodels. Each model needs to comply with the rules stated above. The trigger to use a different attempts datamodel for a case is the content of the field **CMA_AttemptsGUID**.

The Launcher Download Interceptor

The cases which are available for download are determined by the download interceptor that is part of the CMA_Launcher project. Based on the identifier of the user of CMA that is requesting the download, the appropriate cases are provided to CMA. The download interceptor also informs CMA of cases that need to be transferred to another interviewer and of cases that need to be reopened.

The download interceptor updates a log file in which the result of an execution of the setup is written.

The download interceptor receives the current user name from the App and uses that user name in the filter for the records in the launcher database: only records that have that user name in the **CMA_ForWhom** field or in the **CMA_InPossession** field will be made available in a record set. This record set can (optionally) be further restricted to cases for surveys that are intended to run on the specific client device type that is requesting the download. Each record in the then resulting record set will be inspected and can result in a record that will be in the download record set for the client device.

The following records from the resulting record set will be downloaded:

- Records for which **CMA_InPossession** is empty. This will be a new case for the user requestion the download.
- Records for which the value in **CMA_ForWhom** is not equal to the user. Because it is in the record set, it is in possession of the user requesting the download. Such a record will be

downloaded to the client and transferred if the field **CMA_Location** contains the value TRANSFER_REQ.

- The field **CMA_Location** contains either of the following values: RESTORE, RESTORE_REQ, RELEASE_REQ, REOPEN_REQ, CLOSE_REQ, REFRESHCASE_REQ, REFRESHDATA_REQ, REFRESHBOTH_REQ

The records that are downloaded that have an empty **CMA_Location** field will receive the temporary value DOWNLOAD_REQ in the field **CMA_Location** in the **server** launcher database. When the download succeeds this value will be updated by any of the following values: CLIENT, TRANSFER_OK, RELEASED, RELEASE_FAILED, CLOSED

The CMA XML-settings file

CMA reads / writes an XML settings file named settings.xml. This file can be used to specify parameters that influence specific behaviour of CMA. It is also used to set the initial language and the initial style. The selected language, selected style and selected survey in the user interface are stored in this file. In the settings file CMA keeps track of the last change in the data, the last synchronization and when the log file was uploaded the last time. Initial behaviour can be set by providing the CMA_settings.xml or the settings.xml file in the CMA package.

See Appendix A for an overview of all available settings.

Setting a field filter for the cases data grid

A field filter is a comma separated list of field names. The order of the field names in the list is also the order in which they will be displayed on the screen. The following field names are available:

Name	Description
tContactInfoShort	The content of the **ContactInfoShort** field
tSurveyName	The Survey name
tCaseStatus	The content of the **CMA_Status** field
tGroupStatus	The content of the **CMA_GroupStatus** field. Note that this value is only available in a case with **CMA_GroupType**=*gParent*
tGroupSort	The content of the **CMA_GroupSort** field
tStatus	When **CMA_GroupType**=*gParent,* this field has the value of the **CMA_GroupStatus** field, else the value of the **CMA_Status** field
tAttemptsCount	The content of the **CMA_Data.AttemptsCount** field
tLastResult	The last registered result
tAppointment	The appointment registered for the case
tIdentifier	The content of the **ID** field
tStartDate	The content of the **CMA_StartDate** field
tEndDate	The content of the **CMA_EndDate** field
tSpawnCount	The content of the **CMA_SpawnCount** field
tGroupID	The content of the **CMA_GroupID** field
tGeoLocation	The content of the **CMA_GeoLocation** field
tLastAttempt	The content of the **CMA_Process.LastAttempt.When** field
tCaseNote	The content of the **CaseNote** field
tCustomUse	The content of the **CMA_CustomUse** field
tCustomCol<x>	The content of the custom column tCustomCol<x> where <x> can be 1..10. For example **tCustomCol1**.

There are 3 default field filters used by CMA, They are as follows:

1. On the cases screen when showing all surveys:
   tContactInfoShort, tSurveyName, tStatus, tAttemptsCount, tLastResult, tAppointment, tIdentifier, tLastAttempt.
2. On the cases screen when showing the cases of one specific survey:
   tContactInfoShort,tStatus,tAttemptsCount,tLastResult,tAppointment,tIdentifier,tLastAttempt
3. On the group screen:
   tSurveyName,tStatus,tAttemptsCount,tLastResult,tAppointment,tIdentifier,tSpawnCount,tLastAttempt

See Appendix B for an example of creating a CMA settings XML file.

Defining custom columns

It is possible to show data that is stored in the **CMA_ContactData** field or in the **CMA_CustomUse** field in separate columns on the cases or group screen. This can be done through the CMA settings XML file. Note that when using the **CMA_CustomUse** field CMA assumes that this fields contains name-value pairs. By default <name>;<value>;<name>;<value> etc., but it is possible to specify a different pattern using the **sCustomDataSep** setting. This allows for instance for <name>=<value>;

See Appendix B for an example of creating a CMA settings XML file with custom columns.

The cases display order

By default the cases are displayed in the order of a combination of a number of fields of the launcher case. On the cases screen the order is based on the following fields from the launcher: **ContactInfoShort**,**MainSurveyID**,**ID**. On the group screen the order is based on the following fields: **CMA_GroupSort**, **ContactInfoShort**,**MainSurveyID**,**ID.**

You can define another order and / or allow the interviewer to display the cases in a number of different orders. That can be done through the CMA settings XML file. You can do this for the cases screen and for the group screen.

A sort order definition is based on one or more fields. As field names, the field names that are also allowed for a field filter can be used.

See Appendix B for an example of creating a CMA settings XML file with sort definitions.

The Survey XML settings file

CMA supports settings on a survey level. Survey settings can be provided in an optional XML file in the survey project installation package (by using a .bcps file in the project). This file contains a number of settings that either overrule a setting in the CMA settings file or are default values for fields in the launcher case. The name of the settings file is ***cma_surveysettings.xml*** for a non-wave project and **cma_surveysettings_<wavename>.xml** for a wave project. In case of a wave project survey package without a survey settings file, the survey settings file of the non-wave project will be used when present. **Note that the file name needs to be in all lowercase if you are planning to run CMA on Android or iOS devices because in both operating systems the file names are case sensitive.**

See Appendix C for an overview of all available settings.

Survey_SetupAtEnd

It is possible to instruct CMA to run a Manipula setup at the end of a survey. This setup receives access to some of the files used by CMA and it allows you to add cases and / or change information in a case. The setup receives the following information on the command line:

- The primary key of the case just handled. This is available in the setup in the auxfield *cmdID*.
- GUID of the survey just handled. This is available in the setup in the auxfield *cmdGuid*.
- The value of the CMA_Status field before starting the case. This is available in the setup in the auxfield *cmdPrevStatus.*
- In what situation the setup is called. This is available in the setup in the auxfield *cmdAfterInterview*. When this field has the value 1, the interview program was executed before executing the At End setup.
- The complete datamodel file name of the survey just handled using the meta identifier *TopicMeta*.
- The complete file name of the launcher.bmix datamodel using the meta identifier *Launcher*.
- The complete file name of the used attempts datamodel using the meta identifier *Attempts.*
- The complete file name of the logmeta.bmix datamodel using the meta identifier *LogMeta*.
- The complete file name of the topic survey data file using the file identifier *topic*. You are allowed to read and write to this database.
- The complete file name of the local launcher cases data file using the file identifier *ufLocalCases*. You are allowed to read and write to this database.
- The complete file name of the local attempts data file using the file identifier *ufLocalAttempts*. You are allowed to read and write to this database.
- The complete file name of the CMA log file using the file identifier *oLog*.

A framework for such a setup is available in the file *Survey_SetupAtEnd.manx* that is part of the CMA distribution. See also Appendix E.

Survey_SetupAtSpawn

It is possible to instruct CMA to run a Manipula setup when creating a case. A case is created by pressing the *Spawn Case* action button  . This gives you complete control over the content of the new case. This setup receives access to some of the files used by CMA and it allows you to add cases and / or change information in a case. The setup receives the following information on the command line:

- The primary key of the focussed case in the data grid. This is available in the setup in the auxfield *cmdID*.
- GUID of the focussed case in the data grid. This is available in the setup in the auxfield *cmdGuid*.
- The complete datamodel file name of the survey just handled using the meta identifier *TopicMeta*.
- The complete file name of the launcher.bmix datamodel using the meta identifier *Launcher*.
- The complete file name of the logmeta.bmix datamodel using the meta identifier *LogMeta*.
- The complete file name of the topic survey data file using the file identifier *topic*. You are allowed to read and write to this database.
- The complete file name of the local launcher cases data file using the file identifier *ufLocalCases*. You are allowed to read and write to this database.

- The complete file name of the CMA log file using the file identifier *oLog*.

The setup allows for returning the ID of the created case back to CMA. This is handled through the temporary file *tSpawnID*

A framework for such a setup is available in the file *Survey_SetupAtSpawn.manx* that is part of the CMA distribution. See also Appendix F.

Survey_SetupAtSync

It is possible to instruct CMA to run a Manipula setup during synchronization. The setup will be executed as last part of the synchronization, after processing the downloads and handling the upload of launcher cases and topic data. CMA will do this for each survey that has such a setup defined and for which launcher cases have been download / uploaded during the synchronization.

The setup allows you to handle additional downloads / uploads for a survey from/to other sources. The setup receives access to some of the files used by CMA. The setup receives the following information on the command line:

- GUID of the survey. This is available in the setup in the auxfield *cmdGuid*.
- The complete file name of the launcher.bmix datamodel using the meta identifier *Launcher*.
- The complete file name of the logmeta.bmix datamodel using the meta identifier *LogMeta*.
- The complete file name of the synctrace.bmix datamodel using the file identifier *SyncTrace*.
- The complete file name of the local launcher cases data file using the file identifier *ufLocalCases*. You are allowed to read and write to this database.
- The complete file name of the sync trace file using the file identifier *iSyncTrace*. You can only read from this file. The sync trace file will be available when **sWriteSyncTraceFile**=*yes.* By processing the sync trace file you are able to find out which cases were handled during the synchronization.
- The complete file name of the CMA log file using the file identifier *oLog*.

A framework for such a setup is available in the file *Survey_SetupAtSync.manx* that is part of the CMA distribution. See also Appendix G.

Creating cases

CMA offers ways to create cases on the interviewer's device. Created cases will be handled by CMA the same way as cases that were downloaded to the device. Created cases will automatically be uploaded to the server while syncing. A case can be created by the interviewer by clicking the 'create case' button or it can be created in a Manipula script that runs after the interview.

By setting the value Yes in the **CMA_AllowSpawning** field, the create case button becomes visible and enabled on the screen when the case is focused by the interviewer. By setting the value to No the button will be visible but not enabled. When clicked, a new case will be created that is partly a copy of the currently focused case. The original case and the new case will both have the same value in the field **CMA_GroupID**. The new case will receive a value in **ID** that is derived from the original key and the content of the **CMA_SpawnCount** of the original case. The system also supports creating a case through a Manipula setup that is executed when the button is clicked, the *RunAtSpawn* setup.

A case can also be created in the *Survey_SetupAtEnd* setup.

The CMA_ContactInfo datamodel

An industry best practice is to keep Personally Identifiable Information (PII) separate from survey data. However, practically speaking, the interviewer must be able to see some PII (e.g., name and address), and even be able to change PII. Therefore, by default, the CMA_ContactInfo datamodel offers support for a standardised block with PII fields. To allow for other kinds of identifiable case information CMA supports a separate datamodel that is used when the edit case button is clicked. The following needs to be done:

- Create a separate datamodel in which the contact information fields are defined and can be edited. This datamodel must contain the field **ContactInfoShort** on the top level of the datamodel. The value of this field must be computed in the datamodel. Other required fields are **MainSurveyID**, **CaseNote**, **LastChangedCI** and **LastChangedNote** (see the launcher datamodel as shipped with CMA for their exact definitions).
- Populate the field **CMA_ContactDataGUID** with the GUID of the datamodel created.

Note that this datamodel is only for editing the contact data of a launcher case. The contact data needs to be preloaded in the **CMA_ContactData** field using field-value pairs with TAB (ASCII 9) as separator. When the Edit button is pressed, CMA populates the fields of the datamodel using the content of **CMA_ContactData**. After the edit CMA extracts all data fields from the edit datamodel and puts them in the field **CMA_ContactData** using again field-value pairs with the TAB (ASCII 9) as separator.
The content of the field **CMA_ContactData** is merged with the table template to create the details part of the cases screen. CMA will search for the table template text file in the contact info survey folder. When not present there it searches for the template in the CMA folder.

Case details

Details of the case are displayed by CMA using a table definition in a text file. The definition is based on the <TABLE> that can be defined in a text in a datamodel. In a cell of the table a reference can be made to a field in the applicable contact info datamodel using a $ followed by a fully qualified fieldname referencing a field in the contact info datamodel.

It is also possible to use some predefined variables in the definition:

Name	Description
$CMA_Status	Status of the current case
$CMA_GroupStatus	The status of the parent-child group
$CaseOrGroupStatus	When on the main cases screen it is replaced by the value of **CMA_GroupStatus** for a parent case. In all other situations it is the value of **CMA_Status**.
$CMA_LastAttempt	Date and time of last attempt
$CMA_LastResult	The last registered result of the case
$CMA_Geolocation	The value of field **CMA_GeoLocation**
$CMA_Appointment	The last known appointment made in the attempts
$CMA_AttemptsCount	The number of registered attempts for the case
$CMA_ContactImage	To be used to access a preload image for the case. When image is available, a button will appear here and when clicked the display image dialog will be activated.
$CMA_SpawnCount	The number of cases created by the case. When a created case the number represents the number when being created.
$CMA_GroupSummary	The value of the group summary field
$CMA_GroupSum_<char>	To extract one specific value from the group summary field you can use **$CMA_GroupSum_<char>**, where **<char>** can be #, A, S, I, D, C, F, R or L. For instance **$CMA_GroupSum_S**
$CMA_StartDate	The start date of the case, displayed using the active culture
$CMA_EndDate	The end date of the case, displayed using the active culture
$CMA_CustomUse	The value of the **CMA_CustomUse** field
$CMA_CustomUse(<name>)	Can only be used when **CMA_CustomUse** contains name-value pairs. **$CMA_CustomUse(<name>)** is replaced by the <value> of <name> from **CMA_CustomUse.** For instance when the field contains *ID1;ABC;ID2;KLM;ID3;XYZ* then $CMA_CustomUse(ID2) is *KLM*.
$Survey_CustomUse	The value of the **Survey_CustomUse** field
$Survey_CustomUse(<name>)	Can only be used when **Survey_CustomUse** contains name-value pairs. **$Survey_CustomUse(<name>)** is replaced by the <value> of <name> from **Survey_CustomUse.** For instance when the field contains *period=5;projectID=202233* then $Survey_CustomUse(projectID) is *202233*.
$ContactInfoShort	The value of the **ContactInfoShort** field of the case

CMA supports separate detail templates for both portrait and landscape orientation. Both templates need to be defined in one template file. The portrait mode template needs to start with *<!--Portrait-->*. The landscape mode template may start with *<!--Landscape-->*.

The naming convention of the details text file is:
detailstemplate_<cma_ detailstemplate>_<language_id>.txt.
Example: *detailstemplate_psu_en.txt*.

When empty or when the template file is not present, the default template file *detailstemplate_cma_<language_id>.txt* will be used.

See Appendix H for the default case details template shipped with CMA.

Attempt details

Details of the focussed attempt in the attempts dialog are displayed by CMA using a table definition in a text file. The definition is based on the <TABLE> that can be defined in a text in a datamodel. In a cell of the table a reference can be made to a field in the attempt. The following additional references are allowed besides the ones mentioned under *case details*:

Name	Description
$Att_SeqNr	The sequence number of the current attempt
$Att_When	Date and time of the attempt
$Att_Result	The result of the attempt
$Att_Note	The note made in the attempt
$Att_Appointment	The appointment made in the attempt

The naming convention of the attempt details text file is comparable to the naming convention of the case details text file: *attemptstemplate _< cma_ detailstemplate>_<language_id>.txt* or *attemptstemplate _cma_<language_id>.txt*. When no template text file can be located the system default as defined in the CMA.bitt file for the constant *cAttemptsTemplate* will be used.

See Appendix H for the default attempts definition template.

Running a Manipula dialog application as a survey in CMA

CMA supports running a survey that is based on a Manipula dialog setup package. In this case the field **ManiSurveyID** of the launcher case refers to the GUID of a Manipula dialog setup package that has been installed on the server. For such a case, CMA will not start an edit session but it will start the Manipula dialog application as a sub-process in CMA.

When started, the setup receives access to some of the files used by CMA and it allows you to add cases and / or change information in a case. The setup receives the following information on the command line:

- The primary key of the case. This is available in the setup in the auxfield *cmdID*.
- GUID of the setup. This is available in the setup in the auxfield *cmdGuid*.
- The current style of CMA. This is available in the setup in the auxfield *cmdStyle*.
- The complete file name of the launcher.bmix datamodel using the meta identifier *Launcher*.
- The complete file name of the used attempts datamodel using the meta identifier *Attempts.*
- The complete file name of the logmeta.bmix datamodel using the meta identifier *LogMeta*.
- The complete file name of the local launcher cases data file using the file identifier *ufLocalCases*. You are allowed to read and write to this database.
- The complete file name of the local attempts data file using the file identifier *ufLocalAttempts*. You are allowed to read and write to this database.
- The complete file name of the CMA log file using the file identifier *oLog*.

You can add a survey settings XML file to the package. If present, CMA will handle the download and install of child surveys. The child surveys can be used in the setup to collect data. Uploading/downloading of collected data can for instance be handled through child launcher cases.

A framework for such a setup is available in the file *Survey_DialogApplication.manx* that is part of the CMA distribution. See also Appendix I.

Special instructions

CMA supports a number of special instructions that are handled through so-called NULL-GUID cases in the launcher database. The special instructions are handled as a first action during synchronization, so before processing downloaded cases and processing local cases that could result in uploads.

To use a special instruction, a specific case needs to be added to the launcher data base. The case needs to have the null GUID in the field **MainSurveyID**. The null GUID is either ***00000000-0000-0000-0000-000000000000*** or ***NULL***. The field **ID** needs to be given a value also, but this can be any value, for instance the identifier of the interviewer, a generated GUID, a date-time stamp, as long as it makes the case unique in the launcher database. In the launcher case the field **CMA_ForWhom** needs to receive the identifier of the Interviewer. The field **CMA_Location** needs to be set to the special instruction you want to use.

The following case related special instructions are available:

Instruction	Description
RELEASE_ALL	This instruction can be used to remove all cases on a client machine. It can for instance be used if a transfer of cases is not possible and cases need to be re-assigned to another interviewer immediately. Using this functionality prevents overwriting data when cases are now owned by someone else but are also still on the device of someone else.
RELEASE_SOME	This instruction can be used to remove specific cases on a client machine. CMA expects a list of the cases to be released in the **CMA_CustomUse** specified as *<guid>;<id>*; pairs. RELEASE_SOME also handles the release of all cases with the same value for **CMA_GroupID**. This special instruction can for instance be used if a transfer of cases is not possible and cases need to be re-assigned to another interviewer immediately. Using this functionality prevents overwriting data when cases are now owned by someone else but are also still on the device of someone else.
RELEASE_SURVEYS	This instruction can be used to remove all cases from one or more specific surveys on a client machine. CMA expects a list of the surveys to be released in the **CMA_CustomUse** specified as semi-colon separated list, so *<guid>; <guid>;<guid>;…*
RELEASE_ORPHANS	This instruction can be used to remove all orphaned cases on a client machine. An orphaned case is a closed case (**CMA_CaseClosed**=*yes*) for which no survey is present on the server.

When processed successfully the value RELEASE_DONE in the **CMA_Location** field will be sent back to the server.

The following file related special instructions are available:

GET_FILES	This instruction can be used to get specific files from the client which are all located in the same folder. The names of the files needs to be specified in the **CMA_CustomUse** field using a semicolon as separator. The file names all need to be relative to the Surveys folder in the deploy path. For instance: to retrieve the files *CMA_synctraceAll.csv* and *settings.xml* from the *cma* folder you need to specify *cma\CMA_synctraceAll.csv;cma\settings.xml.* You are allowed to use wildcards. For instance: to retrieve all files in the folder *mysurvey* you need to specify *mysurvey*.** The files will be returned in a zip file called *getfiles.zip* in the **CMA_Data.Survey** blob in the launcher case. Note that on Android/iOS file names are case sensitive.
PUT_FILES	This instruction can be used to put files into a specic folder on the client. The name of the folder needs to be specified in the **CMA_CustomUse** field. The files to put on the client need to be loaded in a zip file in the **CMA_Data.Survey** blob in the launcher case. Note that on Android/iOS file names are case sensitive.
DELETE_FILE	This instruction can be used to delete a specific file on the client. The name of the file needs to be specified in the **CMA_CustomUse** field. The file name needs to be relative to the *Surveys* deploy folder. For instance: to delete the file *CMA_synctraceAll.csv* in the subfolder *cma* you need to specify *cma/ CMA_synctraceAll.csv*
LIST_FILES	This instruction can be used to get a list of all files located in the subfolders of the *Surveys* deploy folder. The result will be stored in the **CMA_CustomUse** field as a '|' (pipe symbol) separated list.

When processed the value HANDLED in the **CMA_Location** field will be sent back to the server.

Running a survey on a specific device

It is possible for an interviewer to have multiple client devices each running a specific set of surveys using the same login account. To enable this you need to install the optional project CMA_ClientInfo on the server and set the CMA setting **sUseDeviceType** to *yes*. In the ClientInfo database on the server you need to enter the survey's GUID along with the specific device type it needs to run on. The following device types are currently supported by CMA: *WindowsPC*, *WindowsTablet*, *iPad*, *AndroidTablet*. Possible future devices are: *AndroidPhone* and *iPhone*.

CMA will download cases for all surveys mentioned in the database for the specific device type plus cases for surveys that are not mentioned in the database for any specific device type.

It is possible to run a certain survey on more than one device type, for instance on an iPad and on an Android tablet. This will only work when a specific interviewer has only one of those devices.

For example, suppose you want to run survey <x> on an Android tablet and all other surveys on a Windows PC. The database has to contain only the entry <x> with DeviceType set to *AndroidTablet*.

If you want to make sure that all of your surveys can only be used on a specific device type you need to enter all survey GUIDs in the database.

Note that this functionality is only available when the conditional define *MultiDevice* is active in the sources of the CMA project and the CMA_Launcher project. Note that at least Blaise 5.10.6 is required on the client and at least Blaise 5.10.7 is required on the server. If you want to use CMA 2.0 on an earlier version of 5.10 then the conditional define needs to be disabled.

Activating CMA for Multi-mode case management

In case you want to use CMA in conjunction with the Blaise multi-mode management system (Blaise version 5.14 or higher) it is recommended that you activate the MultiMode conditional define in the sources of the CMA_Launcher project. This will ensure that the CMA_HappeningsCod field of the CMA_Launcher will be included in the event messages that are processed by the case management service.

Synchronization

When starting CMA on the client and when pressing the Sync button on the cases screen, a synchronization with the server will be executed. During the synchronization various things happen:

1. CMA downloads data for the launcher survey. The download interceptor delivers the cases specific to the user that did the login in the DepApp. The download can result in 0 or more cases. When not zero cases, CMA first checks what surveys are needed to handle the cases and downloads and installs the surveys that are not yet present on the device. All downloaded cases for which an installed survey is available on the device are stored in a local database in the CMA_Launcher folder on the device. For each case survey data and attempts are also downloaded from the server, if available. Downloaded cases for which the survey could not be installed are skipped and returned. Downloaded cases that are already present on the client and have no value in the **CMA_InPossession** field or the value RESTORE_REQ are skipped and the case as stored on the local machine is then uploaded to the server.
2. A downloaded case can contain the request to transfer a case. In this case the case is uploaded, including the collected data and attempts. The case is then removed from the client because it no longer belongs to the current interviewer.
3. A downloaded case can contain the request to reopen a case. In this case the **CMA_CaseClosed** field is reset to empty and the **CMA_Status** is set to *Reopened*. The case is now available again to the interviewer.
4. A downloaded case can contain the request to release a case. In this case the case is deleted, including the collected data and attempts.
5. A downloaded case can contain the request to restore a case. In this case the case is restored to the last known situation at the server, including the data in the attempts and topic database.
6. Cases that are labelled as Final are uploaded, including the collected survey data and attempts. The case and the attempts remain available on the client, but they are read-only. They will be released based on a setting or upon request during a synchronization by populating the field **CMA_Location** on the server with the value RELEASE_REQ.
7. CMA checks for missing surveys (a missing survey can occur when cases have been downloaded to the device in a previous session and at that moment the survey was not yet installed on the server) and it also checks for surveys that are installed on the device but for which a more recent version is available on the server. If present, needed surveys are downloaded and installed on the device.
8. CMA checks for linked surveys and installs those.
9. Upload of the logging of CMA when not already done on the current day.
10. Optional: When all cases of a survey have been handled and synchronized, a survey will automatically be removed when **sRemoveSurveyWhenNoCaseLeft** is *yes* in the settings file.

11. Optional: All data of cases that have a value in the field **CMA_Process.LastChangedDT** will be uploaded. This optional upload is activated by setting the field **sUploadChanges** to *yes* in the settings file.
12. Optional: Run the Survey_SetupAtSync Manipula setup.

When the setting **sSyncWhenConnected** = *yes*, CMA will try to synchronize with the hosting server after an attempt has been added and when closing CMA but only if an internet connection is available. When starting CMA, a synchronization will be done, but again only if an internet connection is available.

During the synchronization CMA keeps track of what cases were downloaded / uploaded. This information is written to the so called sync trace file. A description of that file can be found in the next section.

The CMA sync trace files

This ASCII files are produced during the synchronization. Each file contain for each line information on a specific launcher case that was handled during the synchronization. The datamodel that describes the data in a trace file is as follows:

```
DATAMODEL mSyncTrace
PRIMARY MainSurveyID,ID
FIELDS
  TimeStamp: STRING[21]
  MainSurveyID: STRING[36]
  ID: STRING[219]
  SurveyDisplayName: string[200],EMPTY
  CMA_GroupType: (gParent,gChild),EMPTY
  //download info
  CMA_ForWhomD: STRING[20]      //if applicable: value of CMA_ForWhom during download
  CMA_InPossessionD: STRING[20] //if applicable: value of CMA_InPossession during download
  CMA_LocationD: STRINg[20]     //if applicable: value of CMA_Location during download
  TopicDownloaded: (yes),EMPTY  //if applicable: was topic data downloaded?
  //upload info
  CMA_ForWhomU: STRING[20]      //value of CMA_ForWhom during upload
  CMA_InPossessionU: STRING[20] //value of CMA_InPossession during upload
  CMA_LocationU: STRINg[20]     //value of CMA_Location during upload
  ReasonU: STRING[15]           //a string to describe the reason for the upload
  TopicUploaded: (yes,no),EMPTY
  CaseUploadFailed: (yes),EMPTY
ENDMODEL
```

For each launcher case that has been downloaded, an adapted launcher case is also written back to the server. For those cases the download info in the sync trace file contains the values as received by CMA. The upload info in the sync trace file contains the values that were present in the cases that were uploaded to the server.

The field **ReasonU** contains a string with a reason why a launcher case has been uploaded to the server. It can be any of the following strings:

Value	Description
Confirmed	The download case was handled
Changed	The case has been uploaded because it was changed on the client since the last synchronization
Closed	The case has been uploaded and now closed on the client
Present	The downloaded case was not accepted because it was already present on the client
Unhandled	The downloaded case did not meet any of the conditions needed to handle the case
Error	A RELEASE_REQ/REOPEN_REQ request could not be handled because the case is not present on the client

Skipped	The case was skipped because the survey has not been installed on the client
Failed	The upload of topic data failed
Missing	The topic data could not be uploaded because the topic survey is missing on the server
Group	The parent case of a group has been uploaded because something in the group changed

CMA can produce two sync trace files. The file *CMA_synctrace.csv* contains the information of the last synchronization session. The file *CMA_synctraceAll.csv* contains the information of all synchronization sessions, including the last one.
As example some lines from a sync trace file were some new cases are downloaded to the client and a closed case is uploaded to the server:

```
"20221031 11:11:51.733";"a764d1a1-5277-4b1c-94ad-cf356dd2fba3";"12";"ICT
 survey";;"CMA_Ann";"";"";;"CMA_Ann";"cma_ann";"CLIENT";"Confirmed";;
"20221031 11:11:51.787";"a764d1a1-5277-4b1c-94ad-cf356dd2fba3";"15";"ICT
 survey";;"CMA_Ann";"";"";;"CMA_Ann";"cma_ann";"CLIENT";"Confirmed";;
"20221031 11:11:51.792";"a764d1a1-5277-4b1c-94ad-cf356dd2fba3";"18";"ICT
 survey";;"";"";"";;"CMA_Ann";"cma_ann";"SERVER";"Closed";;
```

Using CMA for synchronization only

It is possible to call the synchronization of CMA from an external Manipula setup. This can be done by calling the *HandleSynchronization* procedure in CMA.msux using the -&: parameter on the command line. This procedure has parameters that are needed to enable CMA to make a connection with the server that contains the launcher database. As example:

```
PROCEDURE CallCMASynchroniation
INSTRUCTIONS
  res:= CALL(LocalInstallPath+'cma\cma.msux '+
          ' -W:'+LocalInstallPath+'cma -Q:true -&:HandleSynchronization('+Credentials+')')
ENDPROCEDURE
```

In the example the variable *Credentials* contains a comma separated string with the connection parameters, the variable *LocalInstallPath* contains the deploy path on the client.

The CMA Log file

CMA writes a log file on the client in the folder of CMA. This file keeps track of what happened in CMA. Each line has a time stamp and a log text with a ';' as separator. The name of the log file is *cma_logging.txt*.

Example:

```
20211128,15:38:24+01:00;Start of CMA 1.924 by user cma_ann
20211128,15:38:25+01:00;Blaise run-time version: 5.10.6.2875
20211128,15:38:25+01:00;Blaise setup version: 5.10.6.2875
20211128,15:38:25+01:00;CMA location: C:\Blaise5\DepApp\localhost\cma_ann\Surveys\cma\
20211128,15:38:25+01:00;Operation system: Windows, 10 pro
20211128,15:38:25+01:00;Current culture: nl-NL
20211128,15:38:25+01:00;Active language: EN
20211128,15:38:25+01:00;Serverpark: cma
20211128,15:38:25+01:00;Auto sync: Yes
20211128,15:38:25+01:00;Settings read
20211128,15:38:25+01:00;Geo location will NOT be determined
20211128,15:38:25+01:00;LauncherServerpark: cma
20211128,15:38:25+01:00;Connected and authenticated
20211128,15:38:25+01:00;Display: Initializing...
20211128,15:38:27+01:00;Display: Initializing...
20211128,15:38:28+01:00;Display: Syncing...
20211128,15:38:31+01:00;Number of cases downloaded: 33
```

The log file will be uploaded to the server when the survey Logging has been installed. At most once a day the logging is uploaded during synchronizing or when the Manipula message file CMA_messages.txt exists. The message file is included in the upload. After the upload the logging file is rewritten and the message file erased (if applicable). Note that on the day of installing CMA the logging is not uploaded.

It is possible to view the content of the current log file on the client machine by pressing the Blaise logo on the top left of the cases screen. That screen also shows the current version of CMA and the date/time of installing CMA. By pressing the synchronize button  the logging will be uploaded immediately.

The loggings from users in the logging database on the server can be combined to one text file for each user by running the setup *ExtractLogging.manx* that is shipped with CMA. The setup can be found in sub-folder ExtractBlobs.

The CMA user interface

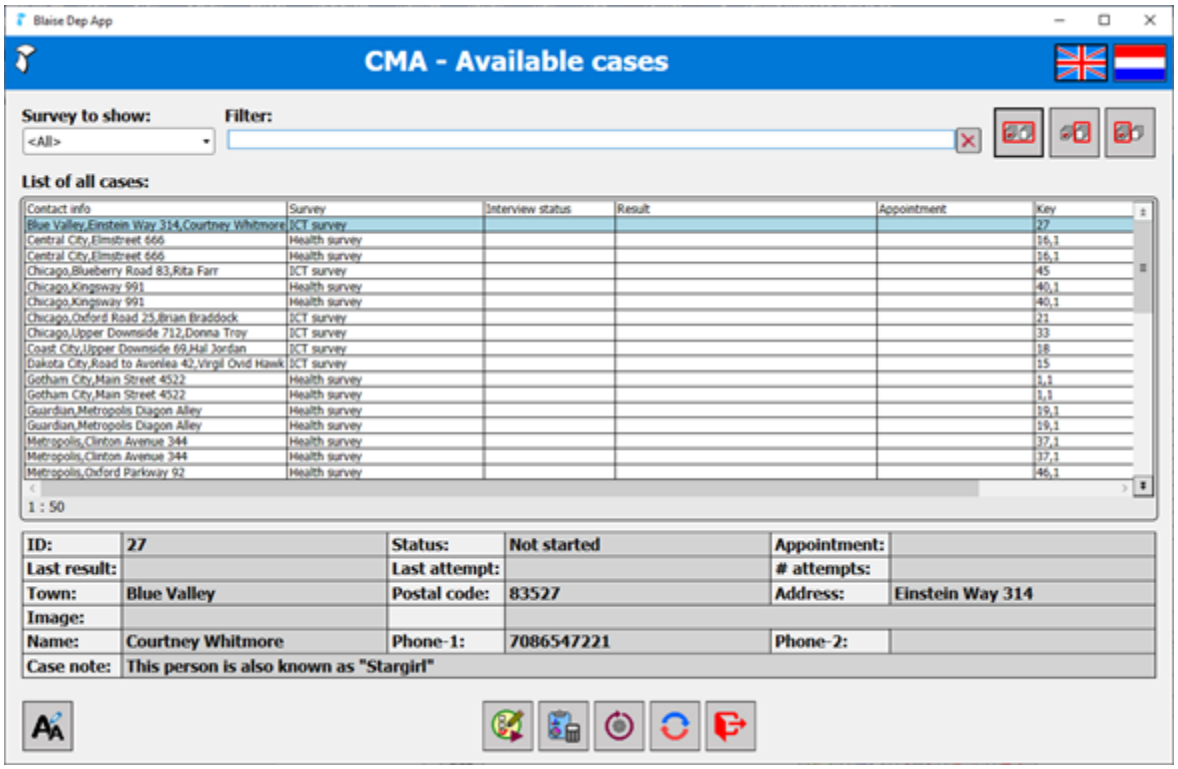
The Cases screen

The central screen of CMA is the cases screen. This screen displays the cases for the current user.

Various views are supported. The views can be set by clicking one of the 3 buttons on the right top of the screen, by selecting an entry in the *Survey to show* dropdown box on the top left of the screen and by entering a value in the filter text box. The bottom part of the screen displays a number of action buttons.

The cases lister

The cases lister shows the cases in the current view. The cases are sorted on the first column (displays the field **ContactInfoShort** of the case). The last appointment registered in an attempt for the case is displayed in the *Appointment* column when still current (for today or in the future). The column *Interview Status* shows the value of the field **CMA_Status** of the case.



Case details

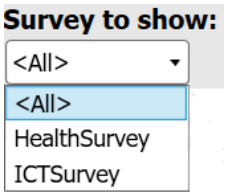
Below the data grid the details for the currently selected case are displayed on a panel. If needed a vertical scroll is visible for the panel.

The 3 buttons for case selection

When pressing the right button all handled cases will be displayed, when pressing the middle button all not-handled cases will be displayed and when pressing the left button all cases will be displayed. The middle button is the default, but this can be changed through the CMA settings file. A text that

describes the currently active selection is displayed above the list with cases. The selected button has a thick black border.

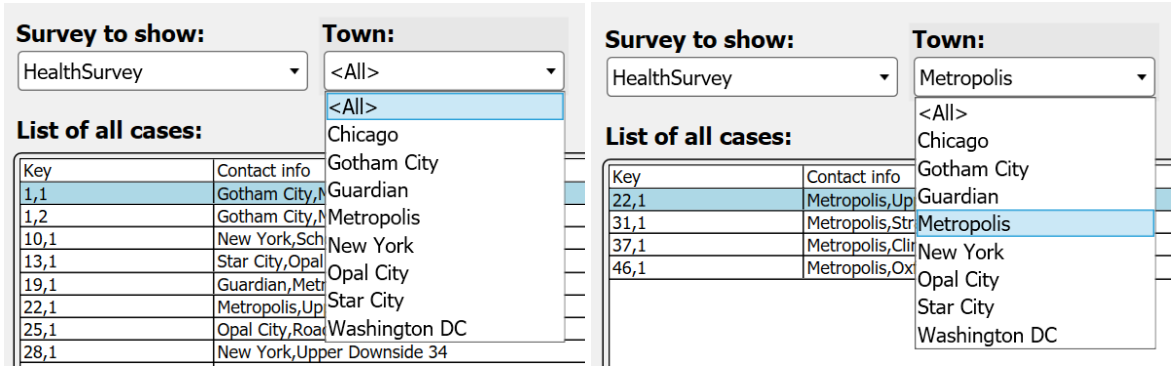
The dropdown survey box for survey selection



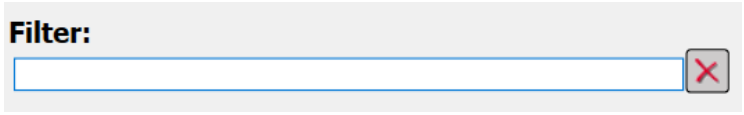
This dropdown box shows all available surveys installed on the device for all cases on the device. By selecting the top-most entry *<All>*, the cases for all surveys for the current user will be displayed.

The custom dropdown filter

It is possible to define a dropdown filter that can be used to filter cases that have a certain value in a certain field in the launcher case. This filter is displayed after the survey selection drop down box. For example:

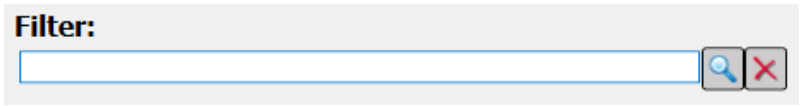


The cases filter



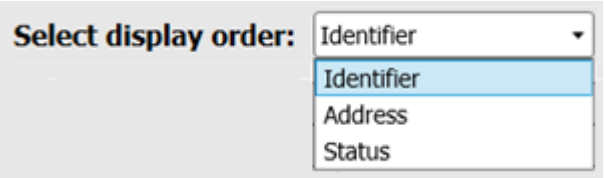
It is possible to filter the cases by entering a text in the filter text box. By default the filter text is used on the **ContactInfoShort** field and the **ID** field. Only the cases are displayed that contain the filter text as substring in the mentioned fields. It is possible to define what fields need to be used through the CMA settings file. Pressing  clears the text in the filter text box. Filtering is done on the active set of cases in the data grid.

By default the filtering is activated by a timer that is started at the moment the user stops entering text in the filter text box. By setting **sSearchButton** to *yes* the timer will be disabled and to activate the filtering the user must now press the search button  :



The select display order dropdown

It is possible to define a dropdown for selecting the order in which the cases must be displayed.



The action buttons

The following buttons are available (from left to right) in the main cases screen:

- ***Show case group***. This button is only visible if the currently focussed case has the value *gParent* in the field **CMA_GroupType**.
- ***Spawn case***. This button is only visible if the currently focussed case has a value in the field **CMA_AllowSpawning** and **CMA_GroupType** is empty. When the value is *no,* the button is disabled (grey button), when the value is *yes,* the button is enabled.
- ***Start interview***. This button is visible when **CMA_GroupType** is empty. When visible, it is disabled (grey) for a handled case and for a case with **CMA_AllowSpawning** set to *yes* and **CMA_IsDonorCase** set to *yes*.
- ***Show attempts***. This button is visible when **CMA_GroupType** is empty. When visible it is always enabled. When clicked the show attempts screen is displayed.
- ***Edit case***. This button is visible when **CMA_GroupType** is empty and the CMA setting **sDoNotAllowEditCase** is empty. When visible, it is disabled (grey) for a handled case and for a case with **CMA_AllowSpawning** set to *yes* and **CMA_IsDonorCase** set to *yes*. When clicked the datamodel for editing the case is started. That model can be used to allow editing of the contact data and to enter / edit the case note.
- ***Determine Geo location***. This button is visible when the CMA setting **sShowGeoPositionBtn** is *yes.* When clicked the current geo location is determined.
- ***Synchronize data***. This button is always visible and enabled. When clicked a synchronization with the office server is performed.
- ***Quit application***. When clicked CMA is stopped. The Quit button is only available under Windows. Under Android the back button of the device can be used and under iOS the back button of the App on the top of the screen.

The Switch Style button

It is possible to show this button on the screen to switch the style of CMA. This button will be displayed when there are at least two styles where the interviewer can choose from. When the button is clicked CMA will switch to the next style in the provided list. When the last of the list is active CMA will switch to the first one. The resource database shipped with CMA contains 3 styles: *Dialogs* (system default as displayed on the screen shot), *DialogsMedium* and *DialogsLarge*. The main differences between these styles are the used font sizes and buttons sizes.

The select language buttons

When CMA has been prepared with multiple languages you can choose to display buttons to switch between the supported languages. You can switch between at maximum 3 languages.

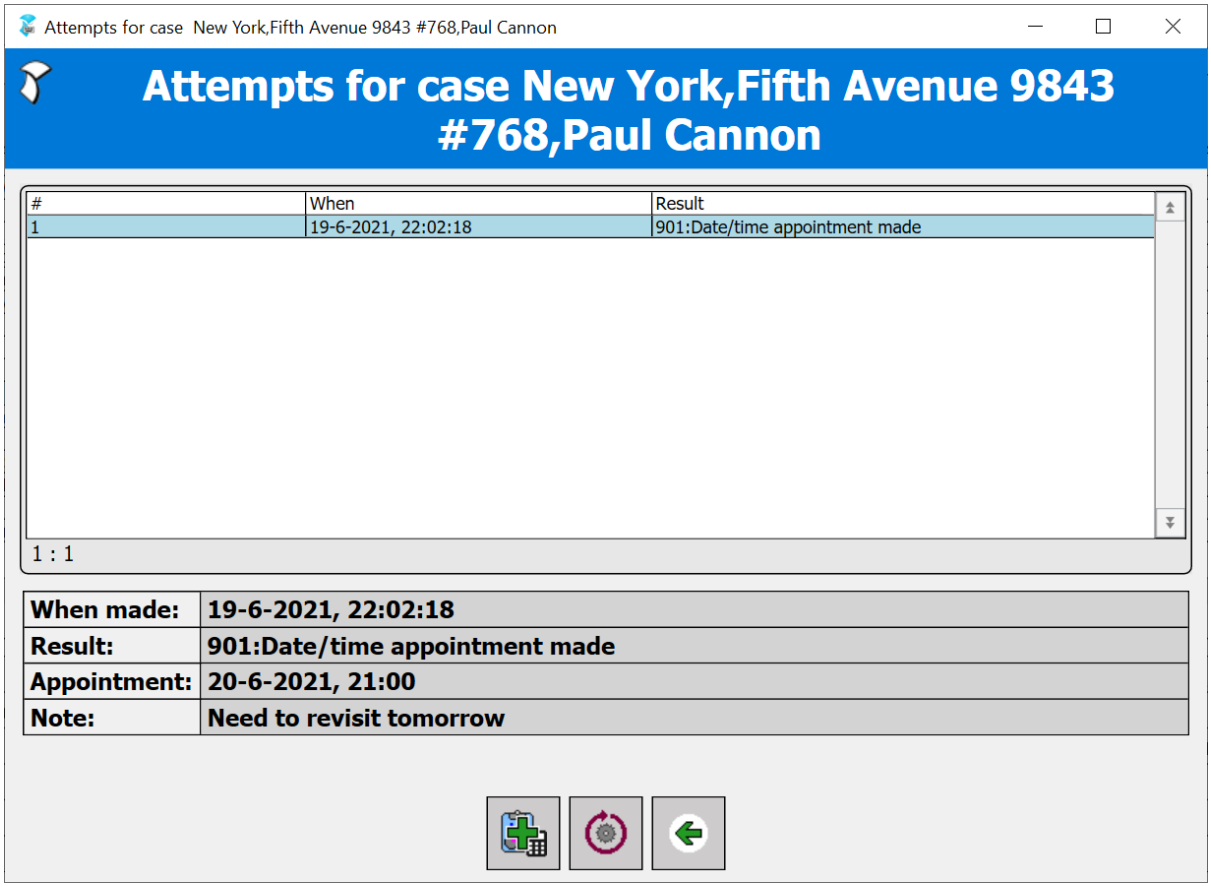
The Case Group screen

When the Show case group button is clicked the Case group screen appears. It contains all cases that have the same value in the **CMA_GroupID** field as the case focussed on the main case screen when the button was clicked. The following action buttons are possible in this screen:

- ***Spawn case***. This button is only visible if the currently focussed case has a value in the field **CMA_AllowSpawning**. When the value is *no,* the button is disabled (grey button), when the value is *yes,* the button is enabled.
- ***Start interview***. This button is always visible. It is disabled (grey) for a handled case and for a case with **CMA_AllowSpawning** set to *yes* and **CMA_IsDonorCase** set to *yes*.
- ***Add attempt***. It is only visible when the CMA setting **Survey_AddAttemptBtnOnGroupScr** is *yes* for the parent of the group. Use this button to register a new attempt. This button is enabled only when the case is not yet completed.
- ***Show attempts***. It is always enabled. When clicked the show attempts screen is displayed.
- ***Edit case***. This button is visible when the CMA setting **sDoNotAllowEditCase** is empty. It is disabled (grey) for a handled case and for a case with **CMA_AllowSpawning** set to *yes* and **CMA_IsDonorCase** set to *yes*. When clicked the datamodel for editing the case is started. That model can be used to allow editing of the contact data and to enter / edit the case note.
- ***Set Group done***. Only visible when the condition to set the group to done is *true* and the setting **Survey_GroupDoneBtn** is *yes* for the parent of the group. When pressed a confirmation message will appear. After confirming the button will no longer be visible.
- ***Determine Geo location***. This button is visible when the CMA setting **sShowGeoPositionBtn** is *yes.* When clicked the current geo location is determined.
- ***Synchronize data***. This button is always visible and enabled. When clicked a synchronization with the office server is performed.
- ***Back***. When clicked CMA returns to the main cases screen.

The Attempts screen

The attempts screen shows what attempts have been registered for the case. The screen shows the detail of the focussed attempt below the list.



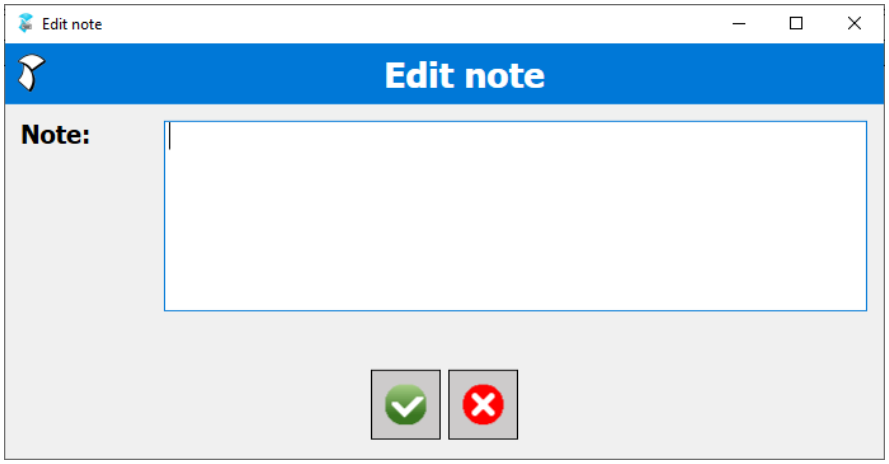
There are four possible action buttons:

- **Add attempt**. Use this button to register a new attempt. This button is enabled only when the case is not yet completed. It can be removed through a CMA setting.
- **Edit note**. Use this button the edit the note of the attempt. The button is only enabled for the last attempt and only when the case is not yet completed. It can be removed through a CMA setting or a survey setting.
- **View interview**. This button allows read-only access to the interview data of the currently selected case. The button is only visible if the survey data entry settings allow for read-only access of a case. It will be enabled when the current case has interview data.
- **Back to cases or group screen**.

The details information shows the appointment if one has been specified when registering the attempt.

The Edit note screen

The edit note screen shows the note of the selected attempt in an edit box.



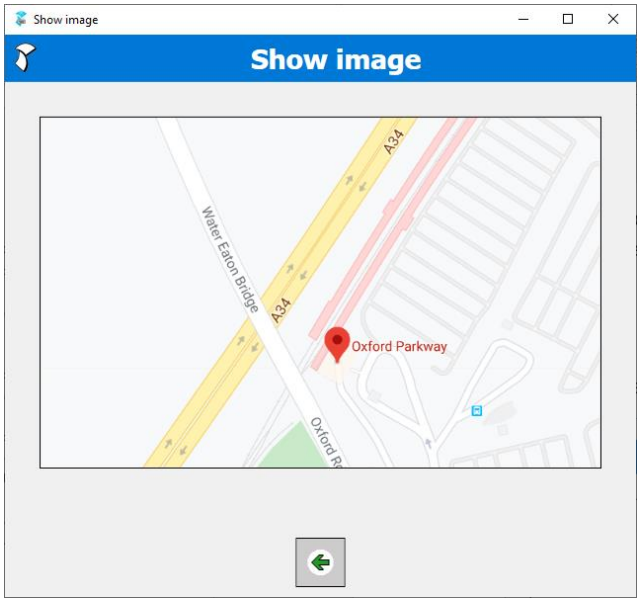
The left button is for saving the changes in the note, the right button to cancel.

Show Image

When a case has an image attached to it, this can be made visible in the details part of the cases screen. For example:

ID:	46,1	Status:	Not started	Appointment:	
Last result:		Last attempt:		# attempts:	
Town:	Metropolis	Postal code:	92745	Address:	Oxford Parkway 92
Image:	Click to show image...				
Name:		Phone-1:		Phone-2:	
Case note:	This is the address of "Cyclops". You need to interview everyone present at this address				

The focussed case has an image and because of this a button is visible with the text ‘Click to show image…’. When clicked the image is displayed in a separate dialog. The next screen shows the image attached to the currently focussed case on the cases screen:

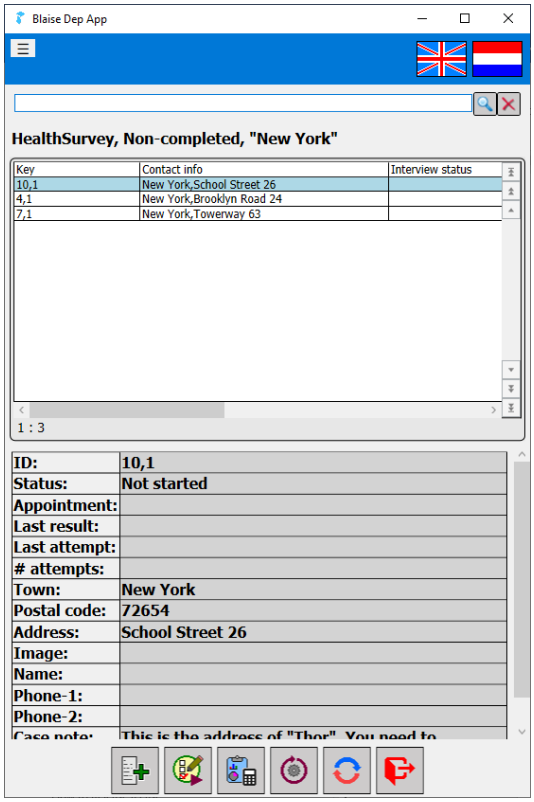


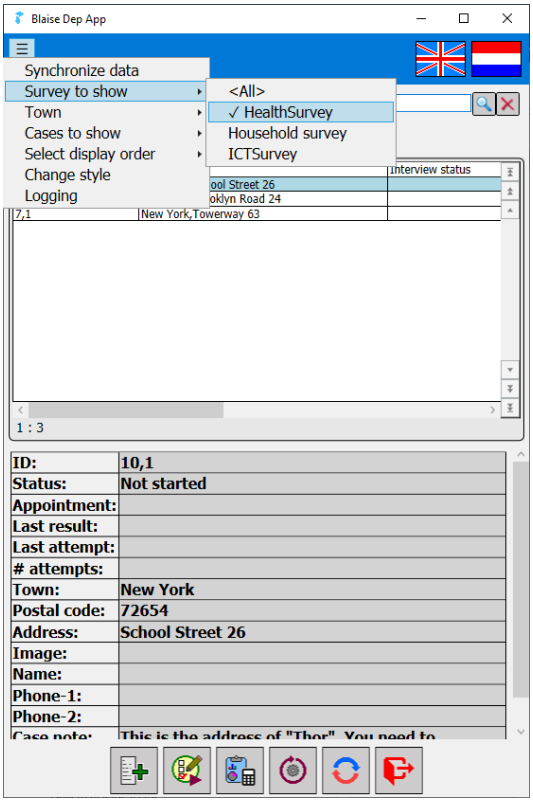
CMA in portrait mode

The screen is adapted when the client device is in portrait mode. The following changes are applied:

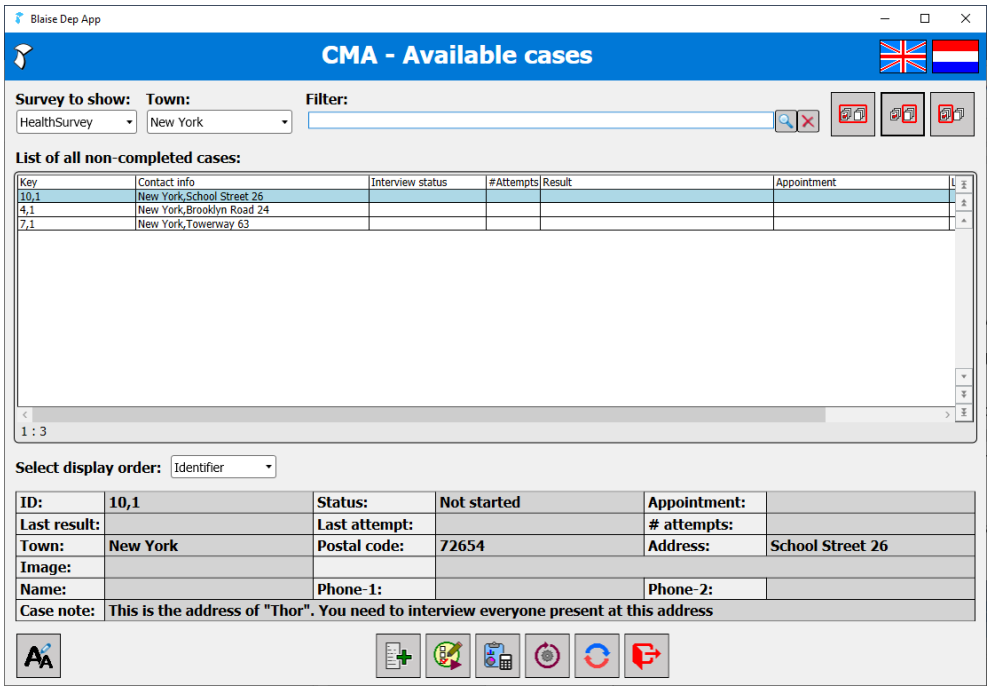
- The following controls are collapsed (hidden) on the screen:
  - The survey to show dropdown.
  - The cases to show selection buttons.
  - The custom dropdown filter.
  - The select display order dropdown.
  - The change style button.
  - The language dropdown (when used; language buttons remain visible).
  - The icon for accessing the logging overview.
- The title of the main screen is collapsed.
- The details below the data grid uses the portrait version of the template.
- The text above the data grid that indicates what cases are currently displayed is replaced by a text the lists the currently selected survey (or <All Surveys>), the current case selection (Non-Completed / Completed) and currently selected value of the custom filter (between double quotes).
- The details get the same height on the screen as the data grid.
- A menu control ☰ is added on the top left of the screen. When clicked the menu opens. In the menu the current active option is indicated by the ✓ symbol (if applicable).

Two example screen shots under Windows are displayed below. The first one shows the menu and the second one shows the menu with the second option selected (*Survey to show*). The current selected survey is *HealthSurvey*. The details below the data grid uses 2 columns.





The same screen in landscape mode is displayed below. Here the height of the data grid is twice the height of the details. The details uses 6 columns in landscape mode.



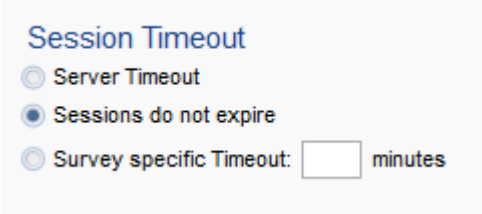
Checklist for a datamodel that you want to use in CMA

In principle every datamodel that you create can run in DepApp and as such also in CMA. But there are some things that you need to be aware of and some things you might give some consideration.

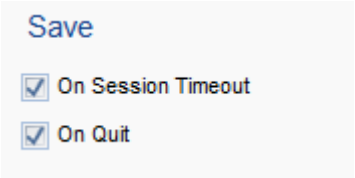
1. In the Data model project, you must select the value *On* in the 'App Support' dropdown box.

2. In order to be able to run in DepApp, the package of the datamodel that you install on the server needs to be prepared WITH generated pages. Only then will the server manager prepare a special package for download to the device. When CMA tries to install a survey that does **not** have generated pages **error -3** will be reported.

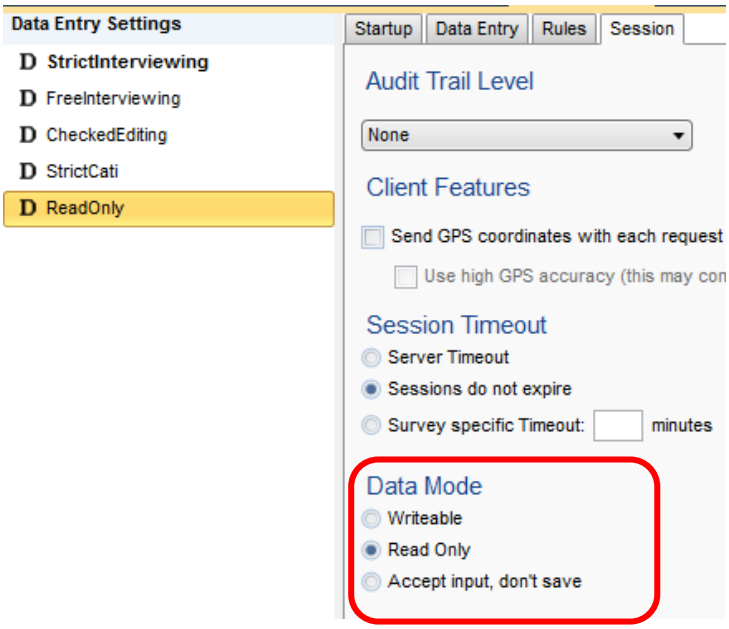
3. Consider setting the Session Timeout in the data entry settings you want to use in CMA to *Sessions do not expire.* When not set, the questionnaire will be terminated after some time of non-activity.



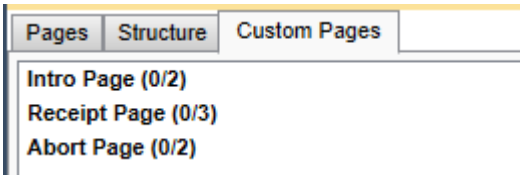
4. Set the Save options in the data entry settings you want to use in CMA to the following:



5. Consider adding a *Data Entry Settings* set with Data Mode set *Read Only.*



6. Consider removing the receipt page and the abort page from the layout set you want to use in CMA. You can do this under the Custom Pages tab.



7. Consider allowing a way to break-off the survey. This can for instance be done by setting the ShowExit parameter to true on the master page (in case you are using the default resource database shipped with Blaise). When set the page will show the exit button on the top right.

8. Check out if the defaults for the following command line option are okay:
   1. Mode
   2. LayoutSet
   3. DataEntrySettings

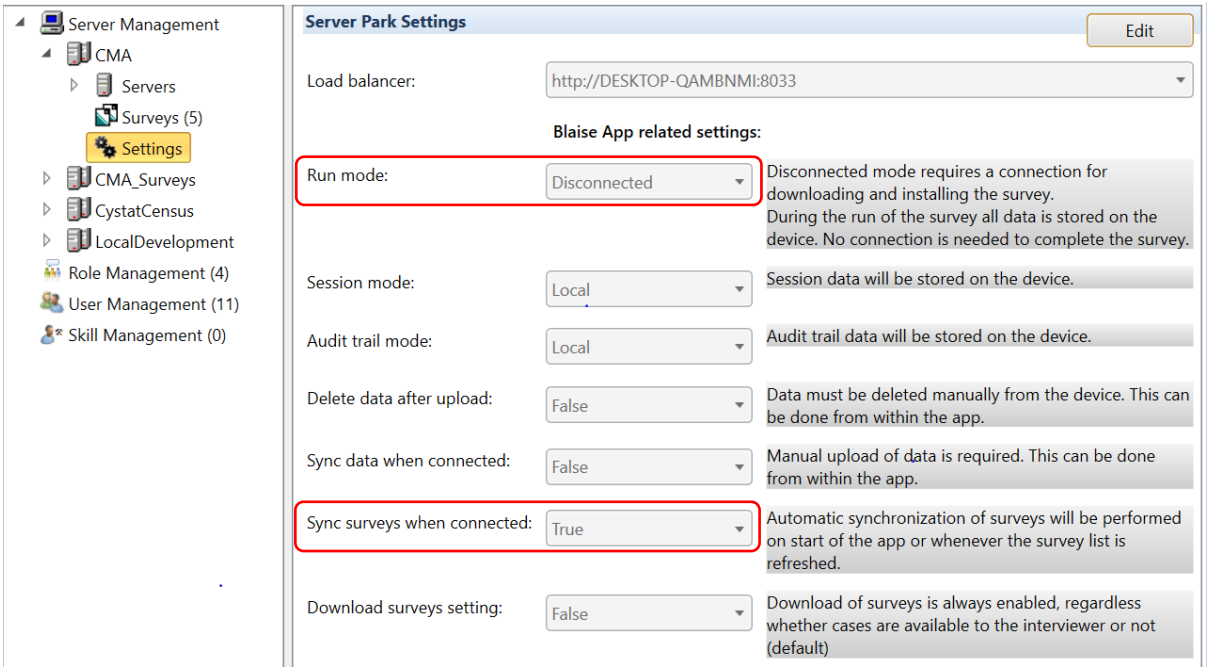
   If you want to change the default you can specify them in the field **Survey_CmdLineForEdit** in the Survey specification file or in the field **CMA_CmdLineForEdit** in the case in the launcher database. For Mode you can also set it through **sDefaultMode** in the CMA settings file or through **Survey_DefaultMode** in the Survey specification file. For LayoutSet you can also set it through **sDefaultLayoutSet** in the CMA settings file or through **Survey_DefaultLayoutSet** in the Survey specification file. For DataEntrySettings you can also set it through **sDefaultDataEntrySettings** in the CMA settings file or through **Survey_DefaultDataEntrySettings** in the Survey specification file.

9. Make sure that the value in the **ID** field in the launcher database fits the primary key definition of the datamodel. If the primary key is composed of multiple fields then the **ID** field in the launcher database is expected to have a comma between the values of the individual keys fields.

10. Make sure that you install the survey in a server park that all interviewers that will receive cases through the launcher database have access to.

Instructions how to set up CMA

1. Set up a server park for installing the CMA multipackage. You need to set the option *Run mode* to *Disconnected* and the option *Sync surveys when connected* to *True*. **All other Blaise App related settings must not be changed.**

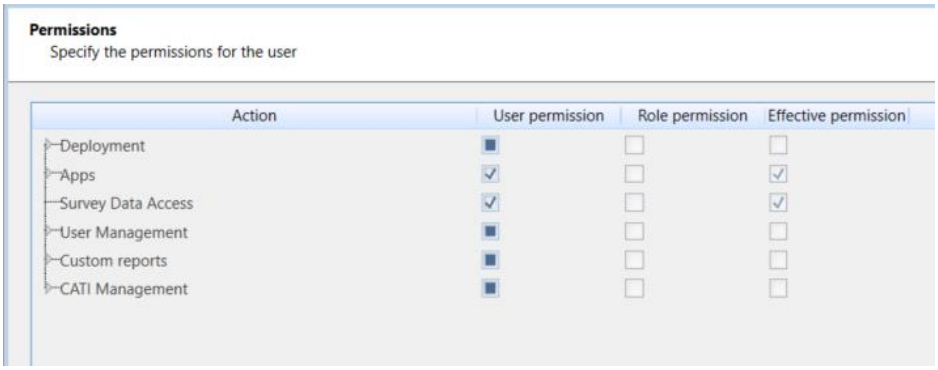


In this server park you need to install the CMA multipackage and **nothing else**. Note that CMA and all the 4 CMA datamodels will automatically be downloaded to the client devices.

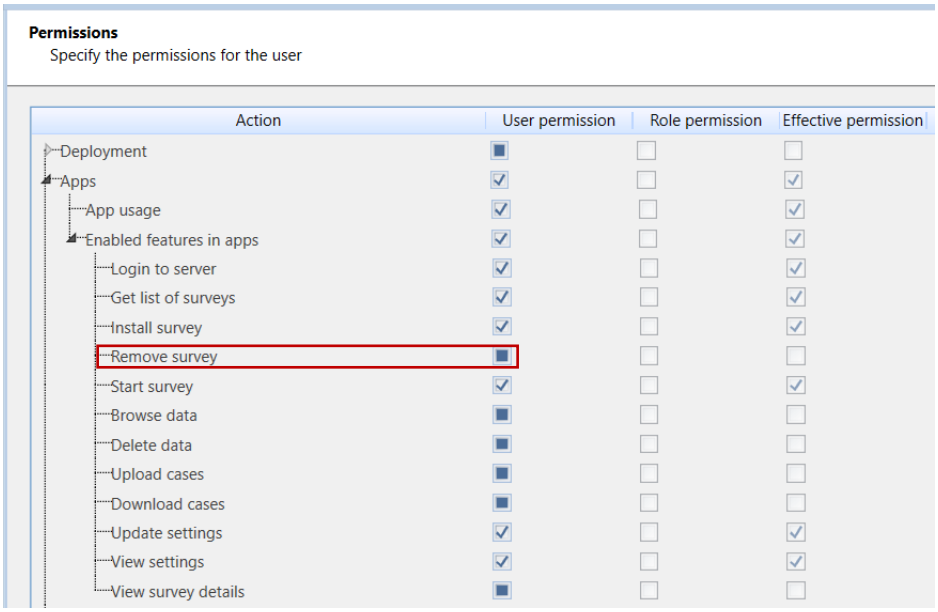
2. Make sure you have at least one server park for installing the topic surveys. For such a server park you need to set the option *Run mode* to *Disconnected*. **Note that all other Blaise App related settings must not be changed.**

3. In the server manager add users and give the users the rights for the CMA server park and for the server park(s) in which the topic surveys will be installed (in the screen shot below that server park has as name *Disconnected* but it can be any valid name):

The following permissions are needed : Apps and Survey Data Access

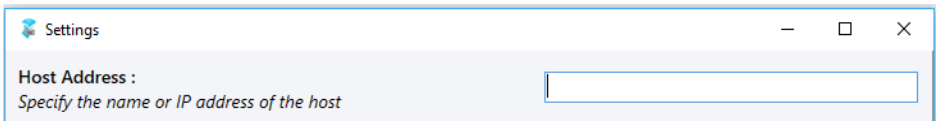


Note that it possible to remove buttons from the main screen of the App (so on the screen that appears before CMA is started). For this you need to adapt the permissions under Apps. The following permissions are needed for CMA to be able to download and run in the App:



We strongly advise to **uncheck** the 'Remove survey' User permission under *Enabled features in apps* as shown in the image above. When this setting is checked the user will be allowed to remove CMA when the DepApp main page is displayed. This page is for instance displayed when closing CMA in DepApp. **Removing CMA in DepApp can lead to undesired behaviour once CMA is installed again**.

4. On the device: Start the DepApp, login with the correct user credentials, select CMA and press the Install button. Make sure that the right host address has been specified on the Settings screen.



Once CMA has been installed it will automatically be started by DepApp and CMA is then initialized for first time use. This can take some time. During the initialization CMA will synchronize data with head office.

It is advised to use the command line option **-Highlander** when starting DepApp on the device. When **-Highlander** is specified there will be only one single instance of DepApp. It will disable an accidental additional start of CMA..

5. In the server park, install survey packages for use on the client devices.

6. In the server park load cases in the launcher database located in the cma_launcher root folder. At least the following fields need to have a value:
    a. **MainSurveyID** = GUID of the survey
    b. **ID** = PRIMARY key of the case within the survey. When the primary key is composed of multiple fields the values of the fields need to be separated by a comma in the ID string field.
    c. Optional: populate the field **CreatedDT** in the block CMA_PROCESS with the date&time when the case was added to the launcher database. The following expression can be used for this: TIMETOSTR(SYSTIME,'yyyyMMdd HH:mm:ss')
    d. Populate the **ContactInfoShort** and **CMA_ContactData** fields with contact information. The **CMA_ContactData** field must be populated with a list of field-value pairs that match the contact info datamodel appliable to the case. The TAB character (ASCII 9) is used as separator, so:
    <field><TAB><value><TAB><field><TAB><value> etc.
    e. Populate the field **CMA_ForWhom** with the login name of the interviewer that needs to get the case.

7. On the client press the Synchronize button to download the cases. The installation of the needed surveys will be done by CMA.

8. Each next time the App is started the App checks if there is an update of CMA available on the server. If so, it will be downloaded and installed. In case any of the supporting datamodels has changed and the changes are harmless the supported datamodels will be updated also.

It is also possible to set up a server park that only has the CMA package installed and another server park that has all packages installed that are required by CMA: CMA_Launcher, CMA_ContactInfo, CMA_Attempts and CMA_Logging. Both server parks need the same settings as mentioned above. Splitting the server parks is important when you want to start using the CMA admin tool.

How to…

In this section short recipes are given for some common CMA related tasks.

How to *Assign a case to an interviewer*

Add a case in the launcher database with the field **MainSurveyID** set to the GUID of the survey project and the field **ID** set to the primary of the case. Make sure that the fields **CMA_ContactData** and **ContactInfoShort** contain the information needed to contact the respondent. Populate the field **CMA_ForWhom** with the identifier of the interviewer that needs to receive the case. During the next synchronization the interviewer will receive the case.

How to *Transfer a case to another interviewer*

Change the value of the **CMA_ForWhom** field to the identifier of the interviewer that needs to receive the case and put the value TRANSFER_REQ in the field **CMA_Location**. Next, the interviewer that has the case in possession needs to synchronize. Then the interviewer that needs to receive the case needs to synchronize. What is described is a two-step procedure and it involves both interviewers. In case the interviewer that has the case in possession is not able to synchronize: consider handling it by using what is described in the section *Remove All or Some Cases*.

How to *Reopen a case*

An already closed case can be reopened for an interviewer by putting the value REOPEN_REQ in the field **CMA_Location**. After synchronizing the case is available again for the interviewer.

How to *Release a case*

A case can be released (= removed from the client device, including all collected data) by putting the value RELEASE_REQ in the field **CMA_Location**. Once released the **CMA_Location** field has the value RELEASED. It is also possible to setup automatic release of cases. This requires the **CMA_EndDate** to be filled and the **sDaysToAutoRelease** field in the CMA settings file to have a value. Completed cases will then automatically be released the number of days specified in **sDaysToAutoRelease** after the value of **CMA_EndDate**. This will happen during synchronizing and will be reported in the **CMA_Location** field as RELEASED. By setting **sAutoReleaseNonCompleted** to *yes* also non-completed cases will automatically be released.

How to *Put a new case on hold*

A new case can be put on hold (= not yet available for download to the interviewer) by putting the value ONHOLD in the field **CMA_Location**.

How to *Restore the content of a device to the last known CMA state*

Cases can be restored (= the last know situation at the server copied to the client machine, including all collected data) by putting the value RESTORE in the field **CMA_Location** in each of the cases that need to be restored.

How to *Inspect the CMA logging of a client device*

The CMA logging of the device is available in the CMA_logging folder on the server in the logging database. The logging is stored in blob fields. It can be extracted to a text file by running the setup ExtractLogging.manx that is shipped with CMA. The setup can be found in sub-folder ExtractBlobs.

How to *Allow for read-only access to an interview in CMA*

Set the field **sAllowReadOnlyAccess** in the CMA settings XML file to *yes* or the field **Survey_AllowReadOnlyAccess** in the survey settings XML file to *yes* and make sure that the topic

survey has a data entry settings set where the property Data Mode is set to *Read Only*. On the attempts dialog the View Interview button will now be visible for a case and enabled if there is an interview registered by CMA.

How to *Remove all cases from the client machine*

This can be done by adding a 'null-GUID' entry in the launcher database with the value RELEASE_ALL in the **CMA_Location** field.

How *to Set the data entry mode for an interview in CMA*

By default the first data entry mode of the data model will be used. This can be overruled various ways:

1. By specifying it as part of the **CMA_CmdLineForEdit** in the launcher case
2. By specifying it as part of the **Survey_CmdLineForEdit** in survey specification file.
3. By specifying the **sDefaultMode** in the CMA specification XML file. When set here it will be used for all surveys unless overruled.
4. By specifying the **Survey_DefaultMode** in the survey specification XML file. When set here it will be used for the specific survey only.

The order listed above is also the order in which the mode that needs to be used will be determined.

How *to Set the layout set for an interview in CMA*

By default the first layout set of the data model will be used. This can be overruled various ways:

1. By specifying it as part of the **CMA_CmdLineForEdit** in the launcher case
2. By specifying it as part of the **Survey_CmdLineForEdit** in survey specification file.
3. By specifying the **sDefaultLayoutSet** in the CMA specification XML file. When set here it will be used for all surveys unless overruled.
4. By specifying the **Survey_DefaultLayoutSet** in the survey specification XML file. When set here it will be used for the specific survey only.

The order listed above is also the order in which the layout set that needs to be used will be determined.

How *to Set the data entry settings for an interview in CMA*

By default the first data entry settings of the data model will be used. This can be overruled various ways:

1. By specifying it as part of the **CMA_CmdLineForEdit** in the launcher case
2. By specifying it as part of the **Survey_CmdLineForEdit** in survey specification file.
3. By specifying the **sDefaultDataEntrySettings** in the CMA specification XML file. When set here it will be used for all surveys unless overruled.
4. By specifying the **Survey_DefaultDataEntrySettings** in the survey specification XML file. When set here it will be used for the specific survey only.

The order listed above is also the order in which the data entry settings that needs to be used will be determined.

The CMA sources

The CMA has one solution: CMA.bsol. This solution has the projects which together form CMA for the client device. The solution has 5 projects, one for the CMA interactive client application and 4 for the supporting CMA datamodels.

Translating the user interface

The CMA sources (the cma.manx, cma_contactinfo.blax and attempts.blax) use as default language English (EN). When a translation to another language is needed then this can be done by using translation text files (extension .bitt). CMA is shipped with some translations. They can be found in the sub-folder Bitt. CMA also has translations for the contact info datamodel and the attempts datamodel (CMA_ContactInfo.bitt, attempts.bitt) but those are currently only available in EN and NL. The details template is available in NL (detailstemplate_cma_nl.txt). Only the texts that are needed are translated in those files to Dutch.

The following needs to be done to enable another language:

1. In the .manx file and .blax files no changes are needed (as of version 5.8 on this is handled automatically during the prepare: all languages defined in the .bitt will be added when not present in the .manx / .blax source and also for all datamodels defined in the .manx source).
2. Add the new language and translations to a CMA .bitt files to the corresponding projects. For details on how to translate a .bitt file see the Blaise help file.
3. You will also need to add the new language to the resource database used by CMA and translate existing texts in that database to the new language. Note that you also have to set the correct culture for the added language because the culture can also influence the user interface of CMA. The name of the resource database is CMA.blrd. For each language a corresponding language button needs to be defined. Best is to use a .png file with size 51 by 36. The identifier needs to be **Flag_<language-ID>**.
4. Make a copy of the file detailstemplate_cma_en.txt to detailstemplate_cma_<id>.txt where <id> is the language identifier used in **lowercase** you want to add (for instance detailstemplate_cma_fr.txt when adding French with identifier FR to CMA). Open the file and translate all texts between `<left>` and `</left>` that do **not** start with a **$** (see as example the texts highlighted yellow in Appendix H in the default template shipped with CMA for EN).
5. Decide on what languages you want to allow in the CMA user interface and set up the correct fields in the CMA settings file. If you do not want to offer the possibility to switch and the language of the user interface needs to correspond with the culture of the device (assuming it is supported by CMA.bitt), then no language settings are required in the CMA settings file.
6. Prepare the CMA solution.

Display Text on action buttons

It is possible to replace the icons on the action buttons by a text or to add a text after the icon. The sizing of the buttons will then need to be change to *Auto*. This can be done by setting **sSetAutoSizeBtns** to *yes*. In the .bitt file you will have to change the entries that refer to an action button.

Instructions how to set up a CMA Demo on a laptop

The CMA demo can be found at <path-to-Samples>\Case Management\CMA\CMADemo. To run the CMA demo you need to go through the following steps:

1. Start the ServerManager.exe program and use Root, Root as credentials.
2. Make sure you have a disconnected server park definition on your laptop. If you do not have one, then create one and call it for instance CMA_Demo. Also make sure you have a second disconnected server park definition to install surveys.
3. Define interviewer CMA_Ann and CMA_Ben. Give them a password that is easy to remember (Ann & Ben). Make sure to set the correct permissions as mentioned in the previous paragraph (allow access to the disconnected server park and permission for Apps and Survey Data Access).
4. Prepare the solution CMADemo.bsol and then install the multipackage CMADemo.mbpkg in the disconnected server park. The multipackage contains the Manipula dialog package CMA.bpkg and 4 datamodel packages, one for each of the 4 datamodels that are used by CMA: CMA_Launcher.bpkg, CMA_ContactInfo.bpkg, CMA_Attempts.bpkg and CMA_Logging.bpkg.
5. Prepare the Surveys solution Surveys.bsol and then install the multipackage Surveys.mbpkg in a disconnected server park that has the setting *Sync surveys when connected* set to *False.* You install this in the second disconnected survey park as mentioned earlier. The multipackage contains 3 datamodel packages: CMA_ICTSurvey.bpkg, CMA_LabourSurvey.bpkg and CMA_HealthSurvey.bpkg.
6. Run the Manipula setup LoadCases.manx. This setup reads the file CMA_Launcher.csv and populates the launcher database on the server. The csv file contains 50 cases for the 2 interviewers for the 3 different surveys. Copy the resulting files launcher.bdbx and launcher.bdix to the folder where the CMA_Launcher package has been installed (for a default Blaise 5 installation: c:\Blaise5\Surveys\CMA_Launcher).
7. Start DepApp.exe (located in the Blaise 5 installation folder). When prompted for user credentials press cancel. Go to the settings dialog by pressing the settings button  and specify *localhost* as host address. Specify as deploy folder *c:\Blaise5\CMA_Client*, as user name *CMA_Ann* and provide the password for user CMA_Ann. Press OK. DepApp.exe now connects to the Blaise service on the laptop and downloads, installs and starts CMA. It takes some time for CMA to install (it installs the 4 CMA datamodel packages and then synchronizes with the server) and start. When successful user CMA_Ann will have 33 cases to handle.
8. To demo transferring of cases from interviewer CMA_Ann to interviewer CMA_Ben run the setup TransferCases.manx on the server. All not yet completed cases are now labelled for transfer in the launcher database on the server. On the client machine of user CMA_Ann, press the synchronize button in CMA. After synchronizing all cases of the ICT survey will no longer be present on the client and the ICT survey has been deinstalled. When user CMA_Ben synchronizes he will receive the transferred cases.
9. To demo restoring of all cases press Synchronize. When finished close CMA and the DepApp. Remove on the client machine the folder c:\Blaise5\CMA_Client (= the deploy folder). Run the setup RestoreCases.manx on the server. Then on the client start DepApp.exe again. Download and install CMA. All cases are now available again on the client.

Demo of the use of a group

It is also possible to include the use of a group in a demo. The sample for this demo can be found in the subfolder GroupDemo. To run this demo you need to do the following additional steps:

1. Open solution GroupDemo.bsol. Prepare the RunAtEnd project and then prepare the solution.
2. Install the GroupDemo.mbpkg in the second disconnected survey park as mentioned earlier. The multipackage contains 4 datamodel packages: HH_Survey.bpkg (for the parent survey of the group), LF_Survey.bpkg, WB_Survey.bpkg (for two child surveys) and Parent_ContactInfo.bpkg (contact info datamodel used by these surveys).
3. Run the Manipula project Group_LoadCase. It will add cases to the launcher database for user CMA_Ann and CMA_Ben.
4. Start the DepApp.exe for user CMA_Ann or CMA_Ben. Additional cases will be downloaded and additional surveys will be installed. By pressing the group button you will enter the group screen for the selected case. Here you can interview the case and when finished additional child cases will be created.

Note that the survey settings files used in the demo have been created by running the Manipula project MakeSurveySettings.

Appendix A - The CMA settings file

An overview of all supported settings in the CMA settings file. The value *yes* in the third column indicates that this setting can be overruled in the Survey settings XML file for a specific survey (see Appendix C)

Name	Description	Survey
sStyleCount	Number of available styles.	
sStyle[x]	The name of a style. The name need to be present in the resource database. x=1..sStyleCount	
sTextFontSize[x]	The text font size to use in the display for sStyle[x]. x=1..sStyleCount.	
sLanguageCount	The number of languages the interviewer can choose from in the CMA user interface. Value can be 0..3, EMPTY. When set to 0 (zero) all languages will be available in the language dropdown control.	
sLanguages	Array that holds the supported languages. Each array element holds the code in the TLanguage type that represents the CMA languages.	
sShowResult	By default the result is displayed as code:string. Set this field to 1 if only the code needs to be displayed and to 2 when only the string value needs to be displayed, to 3 for code:string, to 4 for showing the label and 5 for showing code:label.	
sFieldFilterAll	Field filter to be used when listing the cases of all surveys in the data grid. See Setting a field filter for the cases data grid for further info.	
sConfirmReopenedHandled	When set to *yes* the interviewer, when finishing the interview / attempts, is asked to confirm that the case is now handled. When the interviewer answers *No* the status of the case remains *Reopened*.	
sShowNotStartedTxt	When set to *yes* the data grid will show the 'Not started' text as status for cases that have am empty status.	
sUseIDinConfirmation	When set to *yes* the **ID** will be used in confirmation screens instead of **ContactInfoShort**.	
sTodayAppointmentPreFix	String field. The content of this field will be displayed in front of today's appointment in the cases data grid.	
sTodayAppointmentDetail	String field. The content of this field will be used as a rich text tag around today's appointment in the case details. For example RED.	
sPreserveLogfileOnClient	When set to *yes* CMA will create a file called *cma_logbackup.txt* in which a backup of the uploaded logging will be maintained.	
sWriteSyncTraceFile	When set to *yes* CMA will create a file called *CMA_synctrace.csv* with a trace of the synchronization	
sPreserveSyncTracefileOnClient	When set to *yes* CMA will create a file called *CMA_synctraceAll.csv* with a trace of all synchronizations.	
sSearchCount	The number of CMA launcher field definitions that will be used for filtering.	
sSearchName[x]	Name of a CMA launcher field. x=1..sSearchCount.	
sZipData	When set to *yes* the blobs in **CMA_Data** will contain an encrypted zipped XML file. The password for the encryption is determined by CMA.	

sCheckForMissingTopicData	When set to *yes*, CMA will download survey data of cases that have **sCheckServerForCase** or **Survey_CheckServerForCase** set to *yes* and for which no survey data is present in the local survey database.	
sRefreshTopicData	When set to *yes*, CMA will refresh the survey data for cases that have **sRefreshCasesDuringSync** or **Survey_RefreshCaseDuringSync** set to *yes* and for which no survey data has yet been collected on the client.	
sFieldWidths	Comma separated list that specifies the width in characters of the fields used in the field filter. When not present in the list the width will be determined based on meta and available space in the datagrid. <fieldname>=<width>,<fieldname=width>,...	
sSearchButton	When set to *yes* the timer of the filter text box with be disabled and an additional search button will appear. The interviewer will need to click that button to activate the filter.	
sShowDateFormat	When set all dates in the user interface will be formatted as specified. Example: *yyyyMMdd* will result in a date like '20211028'.	
sSetAutoSizeBtns	When set to *yes* the action buttons will be auto sized. This is needed in case you want to display a text on an action button.	
sUseDeviceType	When set to *yes* CMA will only download cases of surveys that are allowed to run on the device.	
sSyncWhenConnected	When set to *yes*, CMA will initiate a synchronization after registering an attempt. This will be done only when an internet connection is available (either through Wi-Fi, Mobile network or Ethernet).	
sUploadChanges	When set to *yes*, CMA will upload survey data, attempts and spawned cases during synchronization that has been added/changed since the previous synchronization.	
sAskSync	When set to *yes*, CMA will ask permission for performing a synchronization because **sSyncWhenConnected**=*yes*.	
sUseStartDateInFilter	When set to *yes*, CMA will use the **CMA_StartDate** to filter cases for the cases lister.	
sUseEndDateInFilter	When set to *yes*, CMA will use the **CMA_EndDate** to filter cases for the cases lister.	
sDaysToShowBefore	Applies when **CMA_StartDate** is not empty. Can be used to specify the number of days when the cases need to be visible to the interviewer. Default=0, so the case will become visible on the value of **CMA_StartDate**.	
sDaysToShowAfter	Applies when **CMA_EndDate** is not empty. Can be used to specify the number of days when the cases need to remain visible to the interviewer. Default=0, so the case will become invisible on the first day after **CMA_EndDate**.	
sDaysToAutoRelease	Can be used to specify the number of days when the cases need to be released from the device. Default=EMPTY (no auto release).	
sAutoReleaseNonCompleted	When set to *yes* and **sDaysToAutoRelease** is not empty, CMA will also release non-completed cases on the client.	
sDoNotMaximizeEditWindow	Set this value to *yes* when the edit window does not need to start maximized (MS Windows only).	
sCasesToShow	What cases to show at start-up. Can be *All*=0, *NonCompleted*=1 or *Completed*=2. Default is *NonCompleted*.	

sCollapseDetails	When set to *yes* then, when the details template file is empty or not present, the detail below the data grid no longer uses space on the screen (height zero).	
sHideResultOfParent	When set to *yes*, the value of last happening of a parent case (**CMA_GroupType**=*gParent*) will not be displayed as last result on the main cases screen.	
sHonourFormAccessMode	When set to *yes*, CMA is forced to use the Form Access Mode of the used data entry settings of the topic.	
sTrackLocation	When set to *yes*, CMA will register the geo location at the moment of registering an attempt.	yes
sTrackTimeUsed	When set to *yes*, CMA will register the total time spent on a case (in seconds).	yes
sRemoveSurveyWhenNoCasesLeft	When set to *yes*, CMA will remove an installed survey when all cases for the survey have been handled.	yes
sAllowReadOnlyAccess	Set this value to *yes* when you want to allow the user read-only access to interview data of a case.	yes
sOrientation	Set this value to the orientation of the device to be used during the survey data entry session. Possible values: portrait, landscape, both. Default = both.	yes
sFieldFilterTopic	Field filter to be used when listing cases of one topic in the data grid. See Setting a field filter for the cases data grid for further info).	yes
sFieldFilterGroup	Field filter to be used when listing cases of one group in the data grid. See Setting a field filter for the cases data grid for further info.	yes
sDefaultMode	The mode to be used for the data entry session. It will only be used when -Mode: has not been specified for the command line of the data entry session.	yes
sDefaultLayoutSet	The layout set to be used for the data entry session. It will only be used when -LayoutSet: has not been specified for the command line of the data entry session.	yes
sDefaultDataEntrySettings	The data entry settings to be used for the data entry session. It will only be used when -DataEntrySettings: has not been specified for the command line of the data entry session.	yes
sDoNotAllowEditCase	By default the interviewer can access the contact info data by using the *edit case* button. By setting this field to *yes* the *edit case* button will not be visible.	yes
sDoNotAllowManualAttempt	By default the interviewer can add an attempt by using the *add attempt* button on the attempt dialog. By setting this field to *yes* the *Add attempt* button will not be visible.	yes
sDoNotAllowUpdateAttemptNote	By default the interviewer is allowed to edit the note of an attempt. By setting this field to *yes* the *edit note* button will not be visible.	yes
sDoNotAllowShowingAttempts	By default the interviewer is allow to start the attempts dialog. By setting this field to *yes* the show attempts button will not be visible.	yes
sRequireCasePresent	When set to *yes* the interview is only allowed to start when the case for the interview is already in the topic database.	yes
sCheckServerForCase	When set to *yes*, CMA will download survey data during synchronization of cases for which no survey data is present in the local survey database. The setting will only be used when the setting **sCheckForMissingTopicData** has been set to *yes*.	yes
sKeepForWhomOnRelease	When set to *yes* CMA will not empty out the **CMA_ForWhom** field when releasing a case.	

sRefreshCaseDuringSync	When set to *yes*, CMA will refresh the survey data during synchronization for cases for which no survey data has yet been collected. The setting will only be used when the setting **sRefreshTopicData** has been set to *yes*.	yes
sAskDownloadWhenNeeded	When set to *yes* and **sRequireCasePresent** or **Survey_RequireCasePresent** is *yes* and connected to internet then the interviewer will be prompted whether the case needs to be downloaded from the server. When successful the interview will be started.	yes
sShowGeoPositionBtn	By setting this field to yes the *determine geo location* button will be visible on the cases / group screen. Pressing this button determines the value for the **CMA_GeoLocation** field of the currently focussed case. The value can be overruled using a survey settings xml file.	yes
sCustomColCount	The number of custom columns definitions. Maximum allowed is 10.	yes
sCustomCol[x].FieldName	String field. The fully qualified name of the field from where to extract the data. x = 1..sCustomColCount. The data will be stored in the CMA internal field tCustomCol<x>. The custom column will receive the title as defined in the translation file CMA.bitt for tCustomCol<x> or as specified in **sCustomCol[x].Title** (if present).	yes
sCustomCol[x].Origin	Enumerated field. The origin of the data. The value can be oContactInfo (value=1; the **CMA_ContactData** will be used) or oCustomData (value=2; the **CMA_CustomUse** will be used using a semicolon as separator). x=1..sCustomColCount	yes
sCustomCol[x].Width	Number. The column width in characters	yes
sCustomCol[x].Title	The text to be displayed as title for the column. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=Address,NL=Adres. x=1.. sCustomColCount.	yes
sDefSortCount	The number of sort definitions for the cases screen. Maximum allowed is 10.	yes
sDefSort[x].KeyTitle	The text to be displayed in the sort dropdown box. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=Address,NL=Adres. x=1.. sDefSortCount.	yes
sDefSort[x].FieldCount	The number of fields used in the sort definition. Maximum allowed is 10. x=1.. sDefSortCount.	yes
sDefSort[x].FieldName[y]	The field name. It needs to be the name of a field that can also be used in the field filter. x=1..sDefSortCount, y=1.. sDefSort[x].FieldCount.	yes
sDefSortGrpCount	The number of sort definitions for the group screen. Maximum allowed is 10.	yes
sDefSortGrp[x].KeyTitle	The text to be displayed in the sort dropdown box. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=Address,NL=Adres. x=1.. sDefSortGrpCount.	yes
sDefSortGrp[x].FieldCount	The number of fields used in the sort definition. Maximum allowed is 10. x=1.. sDefSortGrpCount	yes
sDefSortGrp[x].FieldName[y]	The field name. It needs to be the name of a field that can also be used in the field filter. x=1.. sDefSortGrpCount, y=1.. sDefSortGrp[x].FieldCount.	yes

sContactDataSep	Separator for the **CMA_ContactData** field. When not specified the default TAB (char 9) will be used. It is possible to change the separator between the field and the value by specifying two separators. For example, if you want 'field=value,' as pattern you specify for **sContactDataSep** '=,'. It is also possible to specify '=TAB'.	yes
sCustomDataSep	Separator for the **CMA_CustomUse** field. When not specified the default ';' (semicolon) will be used. It is possible to change the separator between the field and the value by specifying two separators. For example, if you want 'field=value;' as pattern you specify for **sCustomDataSep** '=;'. It is also possible to specify '=TAB'.	yes
sCustomFilterTxt	The text to be displayed above the custom filter dropdown box. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=City,NL=Plaats	yes
sCustomFilter.FieldName	String field. The fully qualified name of the field from where to extract the data.	yes
sCustomFilter.Origin	Enumerated field. The origin of the data. The value can be oContactInfo (value=1; the **CMA_ContactData** will be used) or oCustomData (value=2; the **CMA_CustomUse** will be used using a semicolon as separator).	yes
sMappingCount	The number of mapping definitions. Maximum allowed is 50.	yes
sMapping [x].Origin	Enumerated field. The origin of the data. The value can be *oContactInfo* (value=1; the **CMA_ContactData** will be used), *oCustomData* (value=2; the **CMA_CustomUse** will be used) or *oLauncher* (value=3; the current launcher case will be used). x=1.. sMappingCount.	yes
sMapping[x].OriginFieldName	String field. The fully qualified name of the field from where to get/put the data. x = 1..sMappingCount. When using *oLauncher* as origin, a field name that is allowed in a field filter is expected.	yes
sMapping [x].Target	Enumerated field. The target of the data. The value can be *tTopic* (value=1; the data will be put in the topic instrument) or *tAttempt* (value=2; the data will be put in the attempt instrument). x=1.. sMappingCount.	yes
sMapping[x].TargetFieldName	String field. The fully qualified name of the field from where to put/get the data. x = 1..sMappingCount.	yes
sMapping[x].Direction	Enumerated field. x = 1..sMappingCount. It can have one of the following values: • *dPut* (value=1): transfer data from the launcher to the topic/attempt before the edit • *dGet* (value=2): transfer data from the topic/attempt to the launcher after the edit *dBoth* (value=3): transfer data from the launcher to the topic/attempt before the edit and transfer data from the topic/attempt to the launcher after the edit	yes

The following variables in the settings.xml file are used to keep track of values used by CMA:

sSelectedLanguage	The language to be used when starting CMA. This field is updated based on the user's choice.
sSelectedStyle	The style to be used when starting CMA. This field is updated based on the user's choice.
sSelectedSurvey	The GUID of the selected survey in the dropdown survey box when closing CMA.
sSelectedSortOn	The selected value of the sort dropdown on the cases screen.
sSelectedSortGrpOn	The selected value of the sort dropdown on the group screen.
sSelectedCasesToShow	The selected *cases to show* button on the cases screen.
sLastLogUpload	To keep track what date the last upload of the logging was done.
sLastChange	To keep track of when the last change to a launcher case was made.
sLastSync	To keep track when the last synchronization was done.

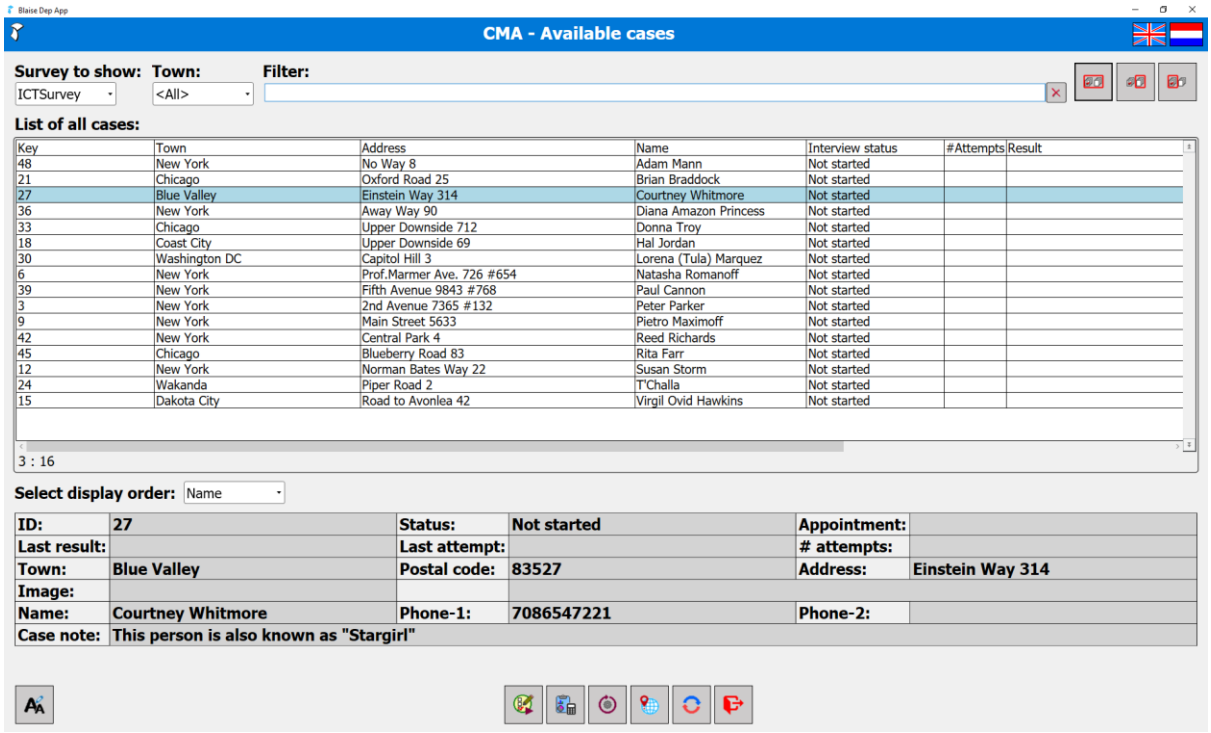
Appendix B – Example of creating a CMA settings XML file

The CMA settings file can best be created using a Manipula setup. As example:

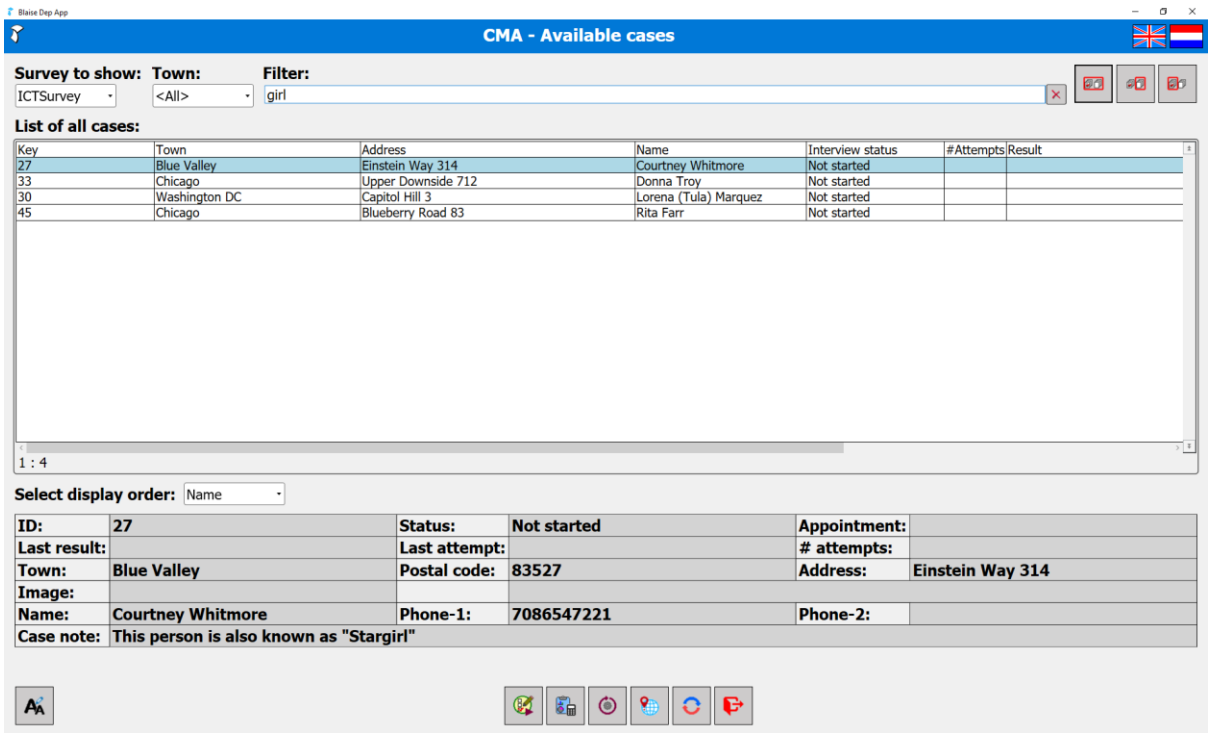
```
SETTINGS AUTOREAD=NO
USES CMA_SettingsMeta

OUTPUTFILE Outp:CMA_SettingsMeta('CMA_settings.xml',XML)

MANIPULATE
  sSelectedLanguage:= 'EN'
  sUploadChanges:= yes
  sTrackTimeUsed:= yes
  sSyncWhenConnected:= yes
  sAskSync:= yes
  sRemoveSurveyWhenNoCasesLeft:= yes
  sCasesToShow:= all
  sConfirmReopenedHandled:= yes
  sAllowReadonlyAccess:= yes
  sShowGeoPositionBtn:= yes
  sShowNotStartedTxt:= yes
  sCustomDataSep:= '=;' //field=value; pattern
  //definition of different styles:
  sStyleCount:= 3
  sStyle[1]:= 'Dialogs' //refers to an existing style in the resource database
  sStyle[2]:= 'DialogsMedium' //refers to an existing style in the resource database
  sStyle[3]:= 'DialogsLarge' //refers to an existing style in the resource database
  sSelectedStyle:= 2
  //defintion of language to be displayed on the screen as button:
  sLanguageCount:= 2
  sLanguages[1]:= 1 //The first language in the CMA.bitt file: EN = English
  sLanguages[2]:= 2 //The second language in the CMA.bitt file: NL = Dutch
  //definition of custom columns:
  sCustomColCount:= 3
  sCustomCol[1].FieldName:= 'PII.Town' //in tCustomCol1
  sCustomCol[1].Origin:= oContactData
  sCustomCol[1].Title:= 'EN=Town,NL=Plaats'
  sCustomCol[1].Width:= 30
  sCustomCol[2].FieldName:= 'PII.Address' //in tCustomCol2
  sCustomCol[2].Origin:= oContactData
  sCustomCol[2].Title:= 'EN=Address,NL=Adres'
  sCustomCol[2].Width:= 40
  sCustomCol[3].FieldName:= 'PII.Name' //in tCustomCol3
  sCustomCol[3].Origin:= oContactData
  sCustomCol[3].Title:= 'EN=Name,NL=Naam'
  sCustomCol[3].Width:= 25
  //field filters to be used:
  sFieldFilterAll:=   'tCustomCol1,tCustomCol2,tCustomCol3,tSurveyName,'+
                      'tIdentifier,tContactInfoShort,tStatus,tLastResult,tAppointment'
  sFieldFilterTopic:= 'tCustomCol1,tCustomCol2,tCustomCol3,tStatus,tAttemptsCount,'+
                      'tLastResult,tAppointment,tIdentifier,tLastAttempt'
  //definition of field width
  sFieldWidths:= 'tIdentifier=20,tStatus=20'
  //definition of sort orders for the cases screen:
  sDefSortCount:= 3
  sDefSort[1].Title:= 'EN=Identifier,NL=Sleutel'
  sDefSort[1].FieldCount:= 1
  sDefSort[1].FieldName[1]:= 'tIdentifier'
  sDefSort[2].Title:= 'EN=Address,NL=Adres'
  sDefSort[2].FieldCount:= 2
  sDefSort[2].FieldName[1]:= 'tCustomCol1'
  sDefSort[2].FieldName[2]:= 'tCustomCol2'
  sDefSort[3].Title:= 'Status'
  sDefSort[3].FieldCount:= 2
  sDefSort[3].FieldName[1]:= 'tStatus'
  sDefSort[3].FieldName[2]:= 'tIdentifier'
  //definition of search field for filter:
  sSearchCount:= 3
  sSearchName[1]:= 'CMA_CustomUse'
  sSearchName[2]:= 'CMA_ContactData'
  sSearchName[3]:= 'ID'
  //custom filter
  sCustomFilterTxt:= 'EN=Town,NL=Plaats'
  sCustomFilter.FieldName:= 'PII.Town' //also defined in tCustomCol1
  sCustomFilter.Origin:= oContactData

  Outp.WRITE
```

Using the CMA_Settings.xml file produced by this setup results in the cases screen below in DepApp. As survey to show *ICTSurvey* has been selected and *Name* has been selected for the display order.

The columns *Town*, *Street* and *Name* show data that is stored in the **CMA_ContactData** field of the case.



Entering the text ‘girl’ on the filter input line results in the next screen. The text ‘girl’ is present in the case note entry in the **CMA_ContactData** field.



Appendix C – The survey settings file

An overview of the settings supported. Note that only non-empty field values are used. All fields are optional and are allowed to be empty.

Name	Description
Survey_ContactInfoGUID	GUID of contact info survey to be used. Can be overruled by the value of **CMA_ContactInfoGUID** in launcher case. **When not** installed in the server park where CMA has been installed, the contact survey will be downloaded and managed by CMA.
Survey_AttemptsGUID	GUID of attempts survey to be used. Can be overruled by the value of **CMA_AttemptsGUID** in launcher case. **When not** installed in the server park where CMA has been installed, the attempts survey will be downloaded and managed by CMA.
Survey_CmdLineForEdit	Command line for the interview session. It can be overruled by the value of **CMA_CmdLineForEdit** in the launcher case
Survey_DetailsTemplate	Name of details template to use. It can be overruled by the value of **CMA_DetailsTemplate** in launcher case
Survey_AttemptsRoute	Value for the Attempts route field. It can be overruled by the value of **CMA_AttemptsRoute** in launcher case
Survey_TrackLocation	Possible values: *yes, no*. It overrules the value of **sTrackLocation** in the CMA settings file.
Survey_TrackTimeUsed	Possible values: *yes, no*. It overrules the value of **sTrackTimeUsed** in the CMA settings file.
Survey_Orientation	Possible values: *portrait, landscape, both*. Overrules value of **sSurveyOrientation** in CMA settings file
Survey_ChildrenCount	Number of child surveys
Survey_Children	Array that contains the GUIDs of additional surveys that need to be installed and managed by CMA
Survey_SetupAtEnd	Name of setup to run at end of a survey. See Survey_SetupAtEnd.
Survey_SetupAtSpawn	Name of setup to run when creating a case by clicking the create case button. See Survey_SetupAtSpawn.
Survey_SetupAtSync	Name of the setup to run during synchronization. See Survey_SetupAtSync.
Survey_RemoveSurveyWhenNoCasesLeft	Possible values: *yes, no*. It overrules the value of **sRemoveSurveyWhenNoCasesLeft** in the CMA settings file.
Survey_FieldFilter	Field filter to be used when displaying case for this survey. It overrules the value of **sFieldFilterTopic** in the CMA settings file.
Survey_DisplayName	Name for the survey to display in the survey column in the data grid. This field will only be used when the field **SurveyDisplayName** in the launcher is empty. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=ICT Survey,NL=ICT onderzoek.
Survey_AllowReadOnlyAccess	Possible values: *yes, no*. It overrules the value of **sAllowReadOnlyAccess** in the CMA settings file.
Survey_RequireCasePresent	Possible values: *yes*, *no*. It overrules the value of **sRequireCasePresent** in the CMA settings file.
Survey_AskDownloadWhenNeeded	Possible values: *yes*, *no*. It overrules the value of **sAskDownloadWhenNeeded** in the CMA settings file.
Survey_DefaultMode	The mode to be used for the data entry session. It overrules the value of **sDefaultMode** in the CMA settings file. It will only be used when -Mode: has not been specified for the command line of the data entry session.

Survey_DefaultLayoutSet	The layout set to be used for the data entry session. It overrules the value of **sDefaultLayoutSet** in the CMA settings file. It will only be used when -LayoutSet: has not been specified for the command line of the data entry session.
Survey_DefaultDataEntrySettings	The data entry settings to be used for the data entry session. It overrules the value of **sDefaultDataEntrySettings** in the CMA settings file. It will only be used when -DataEntrySettings: has not been specified for the command line of the data entry session.
Survey_GroupTitle	Title to display above the data grid on the cases group screen. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=Cases in PSU,NL=Opdrachten in PSU.
Survey_DoNotAllowEditCase	Possible values: *yes*, *no*. It overrules the value of **sDoNotAllowEditCase** in the CMA settings file.
Survey_DoNotAllowManualAttempt	Possible values: *yes*, *no*. It overrules the value of **sDoNotAllowManualAttempt** in the CMA settings file.
Survey_DoNotAllowShowingAttempts	Possible values: *yes*, *no*. It overrules the value of **sDoNotAllowShowingAttempts** in the CMA settings file.
Survey_DoNotAllowUpdateAttemptNote	Possible values: *yes*, *no*. It overrules the value of **sDoNotAllowUpdateAttemptNote** in the CMA settings file.
Survey_CheckServerForCase	Possible values: *yes*, *no*. It overrules the value of **sCheckServerForCase** in the CMA settings file.
Survey_RefreshCaseDuringSync	Possible values: *yes*, *no*. It overrules the value of **sRefreshCaseDuringSync** in the CMA settings file.
Survey_StartInterviewAtSpawn	When set to *yes* the interview will be started immediately after creating the case through a donor case.
Survey_AddAttemptBtnOnGroupScr	When set to *yes* the group screen will show a button to add an attempt.
Survey_DoneValues	A comma separated list of happenings codes.
Survey_GroupDoneBtn	When set to *yes* the group screen can show a button to set the group status to *Done*. The button will be visible when: • All cases of the group have **CMA_Status**: *Completed*, *Finalized* or *Interrupted* • All cases of the group with **CMA_Status** *Interrupted* have a **CMA_HappeningsCod** that is present in the field **Survey_DoneValues** of the current survey or, when **Survey_DoneValues** is empty for the current survey, have a **CMA_HappeningsCod** that is present in the field **Survey_DoneValues** of the parent survey of the group After pressing the button and confirming this action, all cases of the group with **CMA_Status** *Interrupted* will be closed (**CMA_CaseClosed**=*yes*)
Survey_InheritMappings	Enumerated field. The value can be *yes* (value=1) indicating that the mappings defined in the CMA settings need to be applied plus the mappings defined in the survey settings, or *no* (value=2) indicating that the mappings in the CMA_settings must not be applied. When empty the mappings in the CMA settings file will be applied when no mappings have been defined in the survey settings file.
Survey_MappingCount	The number of mapping definitions. Maximum allowed is 50. When set, the mappings defined in the CMA settings file are overruled.

Survey_Mapping [x].Origin	Enumerated field. The origin of the data. The value can be *oContactInfo* (value=1; the **CMA_ContactData** will be used), *oCustomData* (value=2; the **CMA_CustomUse** will be used) or *oLauncher* (value=3; the current launcher case will be used). x=1.. Survey_MappingCount.
Survey_Mapping[x].OriginFieldName	String field. The fully qualified name of the field from where to get/put the data. x = 1..Survey_MappingCount. When using *oLauncher* as origin, a field name that is allowed in a field filter is expected.
Survey_Mapping [x].Target	Enumerated field. The target of the data. The value can be *tTopic* (value=1; the data will be put in the topic instrument) or *tAttempt* (value=2; the data will be put in the attempt instrument). x=1.. Survey_MappingCount.
Survey_Mapping[x].TargetFieldName	String field. The fully qualified name of the field from where to put/get the data. x = 1..Survey_MappingCount.
Survey_Mapping[x].Direction	Enumerated field. x = 1..Survey_MappingCount. It can have one of the following values: • *dPut* (value=1): transfer data from the launcher to the topic/attempt before the edit • *dGet* (value=2): transfer data from the topic/attempt to the launcher after the edit • *dBoth* (value=3): transfer data from the launcher to the topic/attempt before the edit and transfer data from the topic/attempt to the launcher after the edit
Survey_ContactDataSep[1] **Survey_ContactDataSep[2]**	Separator for the **CMA_ContactData** field. When not specified the settings in the CMA settings file will be used. It is possible to change the separator between the field and the value by specifying two separators. For example, if you want 'field=value,' as pattern you specify for **Survey_ContactDataSep[1]** '=' and **Survey_ContactDataSep[2]** ','.
Survey_CustomDataSep[1] **Survey_CustomDataSep[2]**	Separator for the **CMA_CustomUse** field. When not specified the default ';' (semicolon) will be used. It is possible to change the separator between the field and the value by specifying two separators. For example, if you want 'field=value;' as pattern you specify for **Survey_CustomDataSep[1]** '=' and **Survey_CustomDataSep[2]** ';'.
Survey_HasCustomFilter	Possible values: *yes*, *no*. Set to *no* when the CMA settings define a custom filter but the survey does not need one. Set to *yes* when you want to define a custom filter for the survey.
Survey_CustomFilterTxt	The text to be displayed above the custom filter dropdown box. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=City,NL=Plaats
Survey_CustomFilter.FieldName	String field. The fully qualified name of the field from where to extract the data.
Survey_CustomFilter.Origin	Enumerated field. The origin of the data. The value can be oContactInfo (value=1; the **CMA_ContactData** will be used) or oCustomData (value=2; the **CMA_CustomUse** will be used using a semicolon as separator).
Survey_CustomColCount	Integer, 0..10. The number of custom columns definitions. Setting this field to 0 will undo the defined custom columns in the CMA settings file for this survey (so no custom columns).

Survey_CustomCol[x].FieldName	String field. The fully qualified name of the field from where to extract the data. x = 1..sCustomColCount. The data will be stored in the CMA internal field tCustomCol<x>. The custom column will receive the title as defined in the translation file CMA.bitt for tCustomCol<x> or as specified in **sCustomCol[x].Title** (if present).
Survey_CustomCol[x].Origin	Enumerated field. The origin of the data. The value can be oContactInfo (value=1; the **CMA_ContactData** will be used) or oCustomData (value=2; the **CMA_CustomUse** will be used using a semicolon as separator). x=1..sCustomColCount
Survey_CustomCol[x].Width	Number. The column width in characters
Survey_CustomCol[x].Title	The text to be displayed as title for the column. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=Address,NL=Adres. x=1.. sCustomColCount.
Survey_DefSortCount	Integer, 0..10. The number of sort definitions for the cases screen. Setting this field to 0 will undo the definition in the CMA settings file for this survey (so no sort definitions for this survey)
Survey_DefSort[x].KeyTitle	The text to be displayed in the sort dropdown box. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=Address,NL=Adres. x=1.. sDefSortCount.
Survey_DefSort[x].FieldCount	The number of fields used in the sort definition. Maximum allowed is 10. x=1.. sDefSortCount.
Survey_DefSort[x].FieldName[y]	The field name. It needs to be the name of a field that can also be used in the field filter. x=1..sDefSortCount, y=1.. sDefSort[x].FieldCount.
Survey_DefSortGrpCount	Integer, 0..10. The number of sort definitions for the group screen. Setting this field to 0 will undo the definition in the CMA settings file for this survey (so no group sort definitions for this survey).
Survey_DefSortGrp[x].KeyTitle	The text to be displayed in the sort dropdown box. The field supports multiple languages. This is done by specifying a comma separated list of *<languageID>=<survey name>*, for instance: EN=Address,NL=Adres. x=1.. sDefSortGrpCount.
Survey_DefSortGrp[x].FieldCount	The number of fields used in the sort definition. Maximum allowed is 10. x=1.. sDefSortGrpCount
Survey_DefSortGrp[x].FieldName[y]	The field name. It needs to be the name of a field that can also be used in the field filter. x=1.. sDefSortGrpCount, y=1.. sDefSortGrp[x].FieldCount.
Survey_CustomUse	This string field does not play a role in CMA but can be used to store additional data for custom use. The content of the field or part of the content of the field can be displayed on the screen in the details part. By default the field is assumed to contain field;value; pairs (a semicolon as separator between the field and the value and as separator between the field-value pairs). Both separators can be changed through the sCustomDataSep Survey_CustomDataSep settings.

Appendix D – Example of a creating a survey settings XML file

The survey settings file can best be created using a Manipula script.

```
SETTINGS AUTOREAD=NO

USES
  CMA_SurveySettings CMA_SurveySettings'

OUTPUTFILE Outp:CMA_SurveySettings(XML)
SETTINGS
  OPEN=NO
  CREATEBDIX=NO

MANIPULATE
  Outp.OPEN('cma_surveysettings.xml')
  Outp.Survey_ChildrenCount:= 2
  Outp.Survey_Children[1]:= '8695f85c-5543-4f31-bd98-67b33050e362'
  Outp.Survey_Children[2]:= '05bf7a76-5266-4ff3-9b8c-32593ced7260'
  Outp.Survey_SetupAtEnd:= 'RunAtEnd.msux'
  Outp.Survey_RemoveSurveyWhenNoCasesLeft:= no
  Outp.Survey_DisplayName:= 'EN=Household survey,NL=Huishoudenonderzoek'
  Outp.Survey_DetailsTemplate:= 'par'
  Outp.Survey_ContactInfoGUID:= 'ff282825-2051-477b-8578-c4ed0ce41764'
  Outp.Survey_GroupTitle:= 'EN=Household survey, NL=Huishoudenonderzoek'
  Outp.Survey_FieldFilterGroup:=  'tIdentifier,tSurveyName,tCustomCol3,'+
                                  'tStatus,tLastResult,tAppointment'
  Outp.Survey_AllowReadOnlyAccess:= yes
  Outp.Survey_MappingCount:= 2
  Outp.Survey_Mapping[1].Direction:= dBoth
  Outp.Survey_Mapping[1].OriginFieldName:= 'pii.Town'
  Outp.Survey_Mapping[1].TargetFieldName:= 'Municipality'
  Outp.Survey_Mapping[1].Origin:= 1 //contact!
  Outp.Survey_Mapping[1].Target:= tTopic
  Outp.Survey_Mapping[2].Direction:= dPut
  Outp.Survey_Mapping[2].OriginFieldName:= 'tSurveyName'
  Outp.Survey_Mapping[2].TargetFieldName:= 'SurveyName'
  Outp.Survey_Mapping[2].Origin:= 3 //launcher
  Outp.Survey_Mapping[2].Target:= tAttempt

  Outp.WRITE
```

Appendix E – The framework for a Survey_SetupAtEnd setup

```
PROCESS Survey_SetupAtEnd
//CMA 2.6. Framework of AtEnd setup
SETTINGS CONNECT=NO

{.$DEFINE NoTopic}

USES {$IFNDEF NoTopic} topicmeta (VAR) {$ENDIF}
     launcher (VAR)
     attempts (VAR)
     logmeta (VAR) //from CMA appliction

{$IFNDEF NoTopic}
UPDATEFILE topic:topicmeta (BDIX) //name given on cmd line using inp as file id
{$ENDIF}

UPDATEFILE ufLocalCases:launcher (BDIX) //loc launcher name given on cmd line using upd as
file id

UPDATEFILE ufLocalAttempts:attempts (BDIX) //loc launcher name given on cmd line using upd as
file id

TEMPORARYFILE tmp:launcher

OUTPUTFILE oLog:logMeta (ASCII) //log file name set on cmd line using file id oLog
SETTINGS
  MAKENEWFILE=NO
  OPEN=NO
  SEPARATOR=';'
  TRAILINGSPACES=NO
  CREATEBDIX=NO

CONST
  cDateTimeFormat = "yyyyMMdd,HH:mm:sszzz"

FUNCTION NowDT: STRING
INSTRUCTIONS
  RESULT:= TIMETOSTR(SYSTIME,cDateTimeFormat)
ENDFUNCTION

AUXFIELDS
  cmdID: STRING //id of just closed case set through cmd line
  cmdGUID: STRING
  cmdAfterInterview: INTEGER //1=yes, 0=no
  cmdPrevStatus: STRING

PROCEDURE AddToLog
PARAMETERS pTextToLog: STRING
INSTRUCTIONS
  oLog.OPEN
  oLog.PUTVALUE('LogWhen',NowDT)
  oLog.PUTVALUE('LogText',pTextToLog)
  oLog.WRITE
  oLog.RELEASE
ENDPROCEDURE


MANIPULATE
  AddToLog('Start AtEnd setup '+SETUPNAME+':GUID='+cmdGuid+
           ',ID='+cmdID+',AfterInterview='+STR(cmdAfterInterview))
  //
  //Here you do what you want to do...
  //
  AddToLog('End AtEnd setup')
```

Appendix F – The framework for a Survey_SetupAtSpawn setup

```
PROCESS Survey_SetupAtSpawn
//CMA 2.6. Framework of AtSpawn setup
SETTINGS CONNECT=NO

{.$DEFINE NoTopic}

USES {$IFNDEF NoTopic} topicmeta (VAR) {$ENDIF}
     launcher (VAR)
     logmeta (VAR) //from CMA application

  DATAMODEL mId
  FIELDS
    NewID: STRING
  ENDMODEL

{$IFNDEF NoTopic}
UPDATEFILE topic:topicmeta (BDIX) //name given on cmd line using inp as file id
{$ENDIF}

UPDATEFILE ufLocalCases:launcher (BDIX) //loc launcher name given on cmd line

TEMPORARYFILE tSpawnID:mId
SETTINGS
   INTERCHANGE=EXPORT //to return ID of created case

OUTPUTFILE oLog:logMeta (ASCII) //log file name set on cmd line
SETTINGS
  MAKENEWFILE=NO
  OPEN=NO
  SEPARATOR=';'
  TRAILINGSPACES=NO
  CREATEBDIX=NO

CONST
  cDateTimeFormat = "yyyyMMdd,HH:mm:sszzz"

FUNCTION NowDT: STRING
INSTRUCTIONS
  RESULT:= TIMETOSTR(SYSTIME,cDateTimeFormat)
ENDFUNCTION

AUXFIELDS
  cmdID: STRING //id of the selected case in CMA through cmd line
  cmdGUID: STRING //GUID of the selected case set through cmd line

PROCEDURE AddToLog
PARAMETERS pTextToLog: STRING
INSTRUCTIONS
  oLog.OPEN
  oLog.PUTVALUE('LogWhen',NowDT)
  oLog.PUTVALUE('LogText',pTextToLog)
  oLog.WRITE
  oLog.RELEASE
ENDPROCEDURE

MANIPULATE
  AddToLog('Start AtSpawn setup '+SETUPNAME+':  GUID='+cmdGuid+',ID='+cmdID)
  //
  //Here you create an additional case in the launcher
  //and if needed create a new case in the topic database.
  //The id of the created case needs to be returned to CMA
  //through tSpanwnID.NewID:= <id-of-new-case>
  //
  AddToLog('End AtSpwan setup')
```

Appendix G – The framework for a Survey_SetupAtSync setup

```
PROCESS Survey_SetupAtSync
//CMA 2.6. Framework of AtSync setup

SETTINGS CONNECT=NO

{.$DEFINE NoTopic}

USES {$IFNDEF NoTopic} topicmeta (VAR) {$ENDIF}
     launcher (VAR)
     logmeta (VAR)   //from CMA appliction
     synctrace (VAR) //from CMA appliction

INPUTFILE iSyncTrace:synctrace (ASCII) //sync trace file name set on cmd line
SETTINGS
  SEPARATOR=';'
  DELIMITER='"'
  OPEN=NO

{$IFNDEF NoTopic}
UPDATEFILE topic:topicmeta (BDIX) //topic name given on cmd line
{$ENDIF}

UPDATEFILE ufLocalCases:launcher (BDIX) //loc launcher name given on cmd line

TEMPORARYFILE tmp:launcher

OUTPUTFILE oLog:logMeta (ASCII) //log file name set on cmd line
SETTINGS
  MAKENEWFILE=NO
  OPEN=NO
  SEPARATOR=';'
  TRAILINGSPACES=NO
  CREATEBDIX=NO

CONST
  cDateTimeFormat = "yyyyMMdd,HH:mm:sszzz"

FUNCTION NowDT: STRING
INSTRUCTIONS
  RESULT:= TIMETOSTR(SYSTIME,cDateTimeFormat)
ENDFUNCTION

AUXFIELDS
  cmdGUID: STRING

PROCEDURE AddToLog
PARAMETERS pTextToLog: STRING
INSTRUCTIONS
  oLog.OPEN
  oLog.PUTVALUE('LogWhen',NowDT)
  oLog.PUTVALUE('LogText',pTextToLog)
  oLog.WRITE
  oLog.RELEASE
ENDPROCEDURE


MANIPULATE
  AddToLog('Start AtSync setup '+SETUPNAME+': GUID='+cmdGuid)
  //
  //Here you do what you want to do...
  //
  AddToLog('End AtSync setup')
```

Appendix H – Default template for case details and attempts details

Note that text highlighted yellow is text to be translated when adding a new language to CMA.

The case details template:

```
<table borderwidth=1 bordercolor=black grid=both gridwidth=1>
  <column width=auto>
  <column width=*>
  <column width=auto>
  <column width=*>
  <column width=auto>
  <column width=*>
  <row>
    <cell margin="4,2,4,2"><left>ID:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$ID</left></cell>
    <cell margin="4,2,4,2"><left>Status:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_Status</left></cell>
    <cell margin="4,2,4,2"><left>Appointment:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_Appointment</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Last result:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_LastResult</left></cell>
    <cell margin="4,2,4,2"><left>Last attempt:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_LastAttempt</left></cell>
    <cell margin="4,2,4,2"><left># attempts:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_AttemptsCount</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Town:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Town</left></cell>
    <cell margin="4,2,4,2"><left>Postal code:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.PostalCode</left></cell>
    <cell margin="4,2,4,2"><left>Address:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Address</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Image:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_ContactImage</left></cell>
    <cell margin="4,2,4,2"></cell>
    <cell margin="4,2,4,2" colspan=3 background=lightgray></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Name:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Name</left></cell>
    <cell margin="4,2,4,2"><left>Phone-1:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Phone1</left></cell>
    <cell margin="4,2,4,2"><left>Phone-2:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Phone2</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Case note:</left></cell>
    <cell margin="4,2,4,2" colspan=5 background=lightgray><left>$CaseNote</left></cell>
  </row>
</table>
<!--Portrait-->
<table borderwidth=1 bordercolor=black grid=both gridwidth=1>
  <column width=auto>
  <column width=*>
  <row>
    <cell margin="4,2,4,2"><left>ID:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$ID</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Status:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_Status</left></cell>
  </row>
```

```
  <row>
    <cell margin="4,2,4,2"><left>Appointment:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_Appointment</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Last result:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_LastResult</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Last attempt:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_LastAttempt</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left># attempts:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_AttemptsCount</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Town:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Town</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Postal code:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.PostalCode</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Address:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Address</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Image:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$CMA_ContactImage</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Name:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Name</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Phone-1:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Phone1</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Phone-2:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$PII.Phone2</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Case note:</left></cell>
    <cell margin="4,2,4,2"background=lightgray><left>$CaseNote</left></cell>
  </row>
</table>
```

The default attempts details template:

```
<table borderwidth=1 bordercolor=black grid=both gridwidth=1>
  <column width=auto>
  <column width=*>
  <row>
   <cell margin="4,2,4,2"><left>When made:</left></cell>
   <cell margin="4,2,4,2" background=lightgray><left>$Att_When</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Result:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$Att_Result</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Appointment:</left></cell>
    <cell margin="4,2,4,2"background=lightgray><left>$Att_Appointment</left></cell>
  </row>
  <row>
    <cell margin="4,2,4,2"><left>Note:</left></cell>
    <cell margin="4,2,4,2" background=lightgray><left>$Att_Note</left></cell>
  </row>
</table>
```

Appendix I – The framework for a dialog based survey setup

```
PROCESS Survey_DialogApplication

SETTINGS
  CONNECT=NO

LANGUAGES = EN "English"

USES
  launcher (VAR)
  attempts (VAR)
  logmeta (VAR) //from CMA appliction

UPDATEFILE ufLocalCases:launcher (BDIX) //loc launcher name set on cmd line

UPDATEFILE ufLocalAttempts:attempts (BDIX) //loc launcher name set on cmd line

OUTPUTFILE oLog:logMeta (ASCII) //log file name set on cmd line
SETTINGS
  MAKENEWFILE=NO
  OPEN=NO
  SEPARATOR=';'
  TRAILINGSPACES=NO
  CREATEBDIX=NO

CONST
  cDateTimeFormat = "yyyyMMdd,HH:mm:sszzz"

FUNCTION NowDT: STRING
INSTRUCTIONS
  RESULT:= TIMETOSTR(SYSTIME,cDateTimeFormat)
ENDFUNCTION

PROCEDURE AddToLog
PARAMETERS pTextToLog: STRING
INSTRUCTIONS
  oLog.OPEN
  oLog.PUTVALUE('LogWhen',NowDT)
  oLog.PUTVALUE('LogText',pTextToLog)
  oLog.WRITE
  oLog.RELEASE
ENDPROCEDURE

AUXFIELDS
  cmdID: STRING
  cmdGUID: STRING
  cmdStyle: STRING

//Define your dialogs and procedures here...

MANIPULATE
  AddToLog('Start of '+SETUPNAME)
  //
  //Your application...
  //
  AddToLog('End of '+SETUPNAME)
```

Appendix J – Version history

Version 0.7

1. The Cases datamodel has been renamed to Launcher and some key variables have been renamed / moved to a block. This is done to prepare for alignment with the Multi-Mode Management system (Choréo, formerly called M3). Choréo is being developed by Mark Pierzchala with support from the Blaise Team.
2. Introduction of a Restore mode.
3. Support for spawning of cases.
4. Support for filtering cases using the contact information.
5. An appointment made in an attempt is now also registered in the case. The appointment for today and the appointment for the future are listed in the appointment column on the cases screen.
6. An attempt to introduce a generic Attempt datamodel based on Happenings tokens as used by Choréo.

Version 0.8

1. Support for displaying cases based on the CMA_StartDate.
2. Support for releasing case from the client machine.
3. The cases filter input box now has a clear button.
4. Some changes in the layout and texts.
5. Quit button no longer available under Android and iOS.

Version 0.9

1. This version requires Blaise 5.8 or higher.
2. Manipula error message file added to the logging upload.
3. Revision of the download/upload synchronization to make it more robust in case of error situation. To implement this the topic data and attempt data of a case are now always also added as xml file to blobs in the launcher record when changed.
4. Translation no longer requires changes to the source cma.blax. Preparing the cma project with a valid .bitt file that contains the desired additional language specification is now sufficient.
5. Changing the note of an attempt is now allowed through an additional dialog. This dialog can be activated by pressing the Edit button in the Attempts dialog.
6. Check if the identifier of a case matches the primary key definition of the topic instrument with respect to the number of expected commas. If not, the interviewer gets a message on the screen when trying to start the interview. Should never happen (and will not happen if you use the correct identifiers in the launcher database…).
7. Display message when an already installed survey has been updated with a more recent version on the server.
8. Message added when CMA has been started without authentication while having Internet access.

Version 1.0.

1. ***Important change***: CMA can now update itself when changes have been made in the launcher or the attempt model. This allows for future updates of CMA with possible extensions without the interviewer having to remove CMA first.

2. ***Important change***: The edit of the contact information and the case note is now always handled through a separate datamodel. By default, the datamodel **CMA_ContactInfo** is used. It is installed as part of CMA. CMA picks up the values of all fields from that datamodel and stores them as field-value pairs with a tab = char(9) as separator in the **CMA_ContactData** field. The PII block from the launcher has been moved to the CMA_ContactInfo datamodel. It is no longer allowed to change the launcher datamodel.
3. ***Important change***: It is allowed to add a case specific datamodel for the contact information. The trigger to use this case specific model is the field **CMA_ContactDataGUID.**
4. ***Important change***: The details part of the cases screen is now based on a table definition in a file. The field-value pairs in the **CMA_ContactData** field of the case are merged with this table definition and the result is displayed on the screen. The table definition needs to be provided in a separate ASCII text file, per language one file. The default table definition is based on the file *detailstemplate_cma_en.txt*. The trigger to use a different template is the content of the field **CMA_CustomDetails.**
5. ***Important change***: It is now allowed to have a case specific attempts datamodel. The trigger to use such a model is the content of the field **CMA_AttemptsGUID.**
6. The confirmation text when resuming an interrupted interview has been adapted.
7. The date-time fields now include the UTC time zone offset: "yyyyMMdd,HH:mm:sszzz". For example: 20200623,18:12:45+02:00 (this is the time in CET during summer which is then 2 hours ahead of GMT). When displaying such a field to the interviewer the time zone offset will not be visible.
8. Adding a case has been extended with support for a donor case. On the donor case the user cannot start an interview. After adding a case the interviewer is directed to editing of the contact information.
9. Case create now also can work when primary key of topic has only one field. It must then be a string field. The new key is donor key and '-' plus a sequence number. When the donor key ends with '-000' the new id's will be -001, -002, etc.
10. The string 'Added' for newly created cases has been moved to the status column on the cases screen. It can be pre-populated on the server in the field CMA_Status with an initial text to be displayed in CMA.
11. The launcher datamodel has been extended with a field to store the supervisor identifier. It is not used by CMA but can be used later for reporting on CMA results.
12. The launcher datamodel has been extended with a field to store an image for the case. It can be displayed in CMA in a separate dialog.
13. The launcher datamodel has been extended with a field that is passed on to the attempt datamodel for routing purposes.
14. When adding an attempt, a confirmation dialog is displayed.
15. The attempts count is now displayed in the data grid on the cases screen.
16. The logging datamodel has been extended with a time field. It has been added to the primary key.
17. The code value of the Happenings result field is now also stored as a string in the case in the launcher. The last result now displays by default the code plus the text in the format code:text (was until now only the text). A setting has been introduced to display only the code, display only the text (old behaviour) or display them both.
18. The launcher datamodel has been extended with a field that can be used to pre-load fields in the topic instrument.

19. CMA creates the form for the Topic instrument before executing the .Edit. This done to make sure the interview can be executed even when the *Form Access mode* in the data entry settings would not allow this.

Version 1.05

1. ***Important change***: It is now required to install CMA in a server park where the setting 'Auto sync surveys when connected' has been set to TRUE. CMA will be halted when this setting is not correct on the server park.
2. Support has been added for an optional survey settings file. This file contains a number of settings that either overrule a setting in the CMA settings file or are default values for fields in the launcher case.
3. The Show Attempts button is now disabled for cases with **CMA_IsDonorCase = yes**.
4. Support for a fixed orientation of the survey instrument has been added. You can now opt for always Portrait or always Landscape. Changing the physical orientation of the device will have no effect in this case.
5. All parts of the user interface are now translated to the active language. This included the case status and the resulting Happening text. Note that this is in the user-interface only. The texts stored in the launcher are in ENG (the first language in the Attempts datamodel).
6. The validation status after the interview is now determined based on the validation status stored in the database instead of re-computing it.
7. The Show logging dialog has been extended with a button to upload the logging to the server.
8. Support for showing Happenings result as label and as code label has been added.
9. The launcher is now called CMA_Launcher.

Version 1.5

First shipped with 5.9.6 on June 25th, 2021.

1. CMA now supports switching the used style. Styles are defined in CMA resource database. To enable this, you need to specify some settings in the CMA settings file. When enabled a button will appear on the left bottom corner of the screen. Clicking this button loops over the various styles. In the version that is part of the Blaise installation 3 styles are available: *Dialogs* (the system default), *DialogsMedium* and *DialogsLarge*. CMA keeps track of the last used style and will restore the style when CMA is started.
2. Support has been added for which columns to show in the cases data grid and in what order.
3. The value for the Survey column in the cases data grid can now also be set through the survey settings XML file. The column now also supports multiple languages.
4. It is now possible to run a Manipula setup at the end of a topic survey. The setup has access to the launcher case, the last registered attempt, the case in the topic database and the CMA log file.
5. It is now possible to run a Manipula setup when creating a case. The Manipula setup replaces the code in CMA to create and initialise the new case.
6. Support to re-open a closed case has been added. This can be done by setting the field **CMA_Location** to REOPEN_REQ on the server. This triggers a reset of the field **CMA_CaseClosed** to *EMPTY*.
7. Support has been added for linked surveys. A linked survey is a survey where the 'parent' case has the ability to create one or more 'child' cases. In the user interface the user has the ability to open a separate dialog to show the parent case with all created child cases. Surveys for the child cases will automatically be downloaded based on settings in the survey settings

of the parent survey. Child cases will not be displayed on the main screen. Child cases can be created in the Manipula setup that is executed at the end of the parent case. Four new fields have been introduced in the launcher: **CMA_GroupType**, **CMA_GroupID**, **CMA_GroupSort**, **CMA_GroupStatus**.

8. CMA now keeps track of what surveys it did install and will now only de-install a survey if it was originally installed by CMA.
9. A possibility has been added to remove all cases on a client machine as the first action during synchronization. This can for instance be used if a transfer of cases was not possible and cases were assigned to another interviewer. Using this can prevent overwriting data when cases are now owned by someone else.
10. It is now possible to access an interview in read-only mode. This option is only available for a survey that allows for read-only data access in one of the data entry settings definitions.
11. All sub-dialogs are now started maximized (applies to Windows DepApp only).
12. The survey dropdown box now supports translation.
13. A survey setting has been added to force that a case needs to be present in the topic database before interviewing is allowed to start.
14. A new case in the launcher with **CMA_ForWhom** already set, can be put on hold for download to the client device. This can be done by setting the field **CMA_Location** to ONHOLD on the server.
15. Added support for the mode, layout set and data entry settings to use when starting the topic instrument.
16. CMA now downloads and manages the contact info survey and attempts survey mentioned in a survey specification XML file. So it is no longer needed to install them in the server park where CMA is installed. Note that installing in the CMA server park is (for now) still needed for a contact info survey and for an attempt survey mentioned in the fields **CMA_AttemptsGUID** and **CMA_ContactDataGUID** in the launcher case.
17. The handling of languages and switching of languages has been improved. For this the language identifiers have been changed: ENG is now EN and NED is now NL. The CMA.bitt file now contains all predefined languages (currently EN=English and NL=Dutch) and is by default included in the CMA project. The selection of the language can now be done by clicking a language flag instead of a drop down. It is now possible to specify what languages need to be visible for the interviewer (maximum of 3). It is possible to remove the language buttons for instance in case there is no need to switch language. By default the language used in the user interface corresponds with the culture of the device.
18. To avoid an update of data in the settings.xml file, CMA now supports including the file names CMA_settings.xml (instead of settings.xml) in the CMA package. This will trigger an update of the settings.xml file on the client device while maintaining the value of the fields that are updated by CMA (like the selected style).
19. Added support to change the **CMA_Status** field in the attempt datamodel.
20. While registering a new attempt you now have access to the current value of the field **CMA_HappeningsCod** in the launcher case.
21. Added support for automatic releasing of non-completed cases.
22. Added support for removing some cases on a client machine as first action during synchronization.
23. Added support for refreshing certain fields of the launcher case on the client and refreshing the topic data when interview has not yet been started.
24. Added support to remove the ‘Edit case’ button from the cases / group screen.
25. Added support to remove the ‘Add attempt’ button from the attempts screen.

26. Added support for an additional button on the cases / group screen to determine the geo location. The geo location is stored in the **CMA_GeoLocation** field.
27. Added support to define additional (custom) columns for the cases / group data grid using the content from the **CMA_ContacData** / **CMA_CustomUse** field.
28. Added support to define sort keys for the cases / group data grid. The user can switch the sort key using a drop down control positioned below the data grid.
29. A CMA setting has been added to trigger a confirmation screen for re-opened cases after an interview/attempt.
30. It is now possible to define the fields from the launcher data base that will be used in the filter that the interviewer can specify on the cases screen.
31. It is now possible to ask for a download of a specific case for a topic survey when it is not yet present on the client machine and the interview needs to be started.
32. The attempts dialog detail is now based on a template comparable to the case detail.

Version 2.0

This version brings some major enhancements.

1. **Important change**. When downloading a 'fresh' case (= a case where **CMA_InPossession** is empty or **CMA_Location** is **RESTORE_REQ**) from the server that already exists on the client, then the case on the client will no longer be replaced by the downloaded case. In this situation the case from the client will be uploaded to the server.
2. **Important change**. CMA now also supports using two server parks, one for the CMA application and one for the datamodels used by CMA, for instance CMA_Launcher.
3. A setting has been added to specify the separator of the **CMA_ContactData** field.
4. A setting has been added to specify the separator of the **CMA_CustomUse** field.
5. A setting has been added to trigger the download of topic data for cases already present on the client machine. This will be done during synchronization.
6. It is now possible to define a custom dropdown case filter. The drop down contains all values of a specific custom field of the cases displayed in the case data grid. It is handled through the setting **sCustomFilter** in the CMA XML setting file. The text to be displayed on the screen can be set using the setting **sCustomFilterTxt**. This setting supports the use of different languages. For example, this allows you to create a dropdown list that is populated with the different values of a custom field, for instance all cities that are present. By selecting a specific city in the dropdown list CMA will then only list the cases with that specific city value.
7. The data stored in the XML blobs in the launcher case now also store suppressed error information.
8. It is now possible to define the custom column title in the CMA settings file.
9. Collecting the geo position for a specific case can now be done by pressing the optional geo position button.
10. It is now possible to store the XML survey data in an encrypted ZIP in the CMA_Data.Survey BLOB. Encryption is done using a generated password unique for the launcher case. This is handled through the setting **sZipData** in the CMA settings file.
11. The XML stored in the BLOBs in the launcher case now also contain the check information. Because of this any supressed signals are now maintained in the XML.
12. A case can now be closed on the client using **CLOSE_REQ** for the case on the server. A closed case will remain on the client but cannot be handled anymore by the interviewer. In this case the field **CMA_Status** will be set to *Closed*.

13. Support to change the font size of display messages has been added (Windows only). This is handled through the setting **sTextFontSize** in the CMA settings file.
14. Support to release 'orphan' cases that have been handled has been added. An orphan case is a case for which the survey no longer exists on the server.
15. A setting has been added to refresh topic data during synchronizing.
16. Some optimizations have been implemented to increase performance when many cases are present in the launcher database on the device
17. Using the conditional define *{$DEFINE UseDateTimeType}* CMA will use DATETIMETYPE fields instead of STRING fields (with format *yyyyMMdd,HH:mm:sszzz*) to store date-time values in the launcher database. The conditional is needed in CMA.manx and Launcher.blax.
18. The icon for spawning a case has been change to 
19. It is now possible to add an attempt directly from the group screen.
20. An option has been added to operate the filter input box using a search button instead of the timer. This is handled through the setting **sSearchButton** in the CMA settings file.
21. Support has been added to start an interview immediately after spawning a case using a donor case. This is handled through the survey XML setting **Survey_StartInterviewAtSpawn**.
22. It is now possible to specify the date format to be used when displaying a date in CMA. This is handled through the setting **sShowDateFormat** in the CMA settings file.
23. Locating the attempts details template has been improved. It now also searches for the template in the ContactInfo datamodel folder.
24. Cases that have been downloaded during synchronization with **CMA_GroupType**=*gChild* are now excluded from the download cases counter that is displayed to the interviewer on the screen.
25. Topic data is now only uploaded when the topic data was changed since the last upload.
26. It is now possible to specify the column width for a field displayed in the cases / group data grid in the CMA settings file.
27. Support has been added for running surveys on a specific device type. This allows a user to use multiple devices, each running specific surveys. This functionality is available through the conditional define *{$DEFINE MultiDevice}* in DownloadInterceptor.manx and in CMA.manx. Note that when using this functionality CMA.manx passes a record filter for the download to the download interceptor. The download interceptor combines that record filter with the record filter to select cases for the specific user. This functionality requires Blaise version 5.10.7 or higher on the client and on the server. When using an earlier version the conditional define need to be disabled.
28. Support has been added to declare a specific group as 'Done'. This can only be done if all entries in the group have status *Completed*, *Finalized* or *Interrupted* and all entries with status *Interrupted* have a happenings code in a defined list.
29. The CMA.bitt file is no longer included in the CMA project. The CMA.bitt now only contains the default language (EN=English). Files with additional language translations can be found in the sub folder Bitt. CMA_EN_NL.bitt contains EN and NL (Dutch), CMA_EN_AR.bitt contains EN and AR (Arabic) and CMA_EN_EL.bitt contains EN and EL (Greek).
30. Support has been added to put a string before the appointment in the data grid when the appointment is for today.
31. Support has been added to transfer data from the **CMA_ContactData** / **CMA_CustomUse** field to the ***topic*** case and back. It is also possible to transfer data from any of the fields allowed in the field filter from the ***launcher*** to the ***topic*** instrument. The transfer can be set through field mappings in the survey settings file.

32. Support has been added to transfer data from the **CMA_ContactData** / **CMA_CustomUse** field to the ***attempt*** case and back. It is also possible to transfer data from any of the fields allowed in the field filter from the ***launcher*** to the ***attempt*** instrument. The transfer can be set through field mappings in the survey settings file.
33. Support has been added to define field mappings in the CMA settings file. For a survey it is possible to inherit the mappings defined in the CMA settings file or to ignore them.
34. Support has been added for the use of double quotes as delimiter in the field-value pairs of the **CMA_CustomUse** and **CMA_ContactData** fields. They will be added around the value when there is a conflict between the separator and the value. For example:
    Name="Hekking, Tjolk",Address="Gedoogstraat 2, 1982KB, Juinen",
    Here the double quotes need to be used because the comma is used in the name and in the address value and the comma is also used as the separator between the field-value pairs.
35. Support has been added for putting a string in front of an appointment for today, for instance '>>'. It is also possible to change the color of such an appointment in the details grid.

Version 2.06

1. **Important change.** CMA will now only accept a downloaded case when CMA was able to install the survey for that case on the client. Cases that are not accepted because the survey has not been installed on the client are listed in the log.
2. **Improvement.** The download interceptor of the launcher has been optimised. It is now much more efficient when there are no cases available for download to the client.
3. **Important.** The version fixes numerous reported issues, for instance a synchronization issue resulting in collected topic data not always being uploaded to the server.

Version 2.07

1. CMA now removes the survey session data when releasing / transferring a case.

version 2.5

1. Several fundamental changes were made to the *CMA.manx* source code. Some changes were made to improve maintenance (e.g. CONNECT=NO is now used everywhere) and some changes were made to improve performance. Note that the changes do **not** change the behaviour of CMA.
2. To improve debugging / analysing issues the Manipula message file now uses a date-time stamp (Manipula setting ADDDATETIMETOMESSAGEFILE=YES).
3. Added the special instruction RELEASE_SURVEYS. This special instruction can be used to remove a survey plus all related data from a client device.
4. Added CMA setting **sDoNotAllowUpdateAttemptNote**. This setting can be used to disable making changes to the note of an attempt. When set to *yes* the button to edit the note will not be visible.
5. Added CMA setting **sPreserveLogfileOnClient**. When set to *yes* CMA will create a file called *cma_logbackup.txt* in which a backup of the uploaded logging will be maintained.
6. Support has been added for using the language dropdown control instead of the language flags. This can be enabled by specifying **sLanguageCount** to 0 (zero) in the CMA settings XML.
7. Support has been added for a 'server maintenance mode'. When the status of the CMA_Launcher project has been set to *inactive* (this can be done by pressing the deactivate button in the server manager), CMA will present a message when trying to synchronize. Synchronization will not happen as long as the CMA_Launcher status has not been reset to *active*.

8. Added CMA setting **sCollapseDetails**. When set to *yes* then, when the details template file is empty or not present, the detail below the data grid no longer uses space on the screen (height zero).
9. Support has been added for a survey that uses a Manipula Dialog setup package instead of a Datamodel package. When the interview button is pressed, CMA will start the dialog application instead of the data entry client. In the dialog application you have access to the CMA related databases.
10. The handling of RunAtSpawn setup has been improved. It now supports a return parameter to give CMA access to the ID of the just spawned case.
11. Support has been added to the templates for $CaseOrGroupStatus. When on the main screen it is replaced by the value of **CMA_GroupStatus** for a parent case. In all other situations it is the value of **CMA_Status**.
12. Added CMA setting **sHideResultOfParent**. When set to *yes*, the value of last happening of a *parent* case (**CMA_GroupType**=*gParent*) will not be displayed as last result on the main cases screen.
13. Added CMA setting **sHonourFormAccessMode**. By default CMA creates a new topic case when the topic case is not present on the client when starting the interview. This is true even when the Form Access Mode of the used data entry settings of the topic has been set to '*Forms that already exist in the database*'. By setting **sHonourFormAccessMode** to *yes* CMA is forced to use the Form Access Mode of the used data entry settings of the topic.
14. The attempts data grid now also displays the appointment.
15. Various settings that were only present in the CMA settings file have now been implemented also as settings for a survey.
    a. The following survey settings have been added: **Survey_DoNotAllowManualAttempt**, **Survey_DoNotAllowShowingAttempts**, **Survey_DoNotAllowUpdateAttemptNote, Survey_ShowGeoPositionBtn**. They can be used to overrule the corresponding setting in the CMA setting file.
    b. It is now possible to define custom columns for a specific survey. When present they overrule the definition in the CMA settings file.
    c. It is now possible to define the contact data separator and the custom data separator for a specific survey. When present they overrule the default / the value set in the CMA settings file.
    d. It is now possible to define the sort order and the group sort order for a specific survey. When present they overrule the specification in the CMA settings file when appropriate (= when not showing the cases for all surveys on the main CMA screen).
    e. It is now possible to define the custom field filter for a specific survey. When present it overrules the specification in the CMA settings file when appropriate (= when not showing the cases for all surveys on the main CMA screen).
16. The following CMA settings have been added: **sRequireCasePresent**, **sCheckServerForCase**, **sRefreshCaseDuringSync** and **sAskDownloadWhenNeeded**. They can all be overruled by their corresponding survey setting.
17. CMA has been enabled to run on a phone / small device. The following extensions/improvement are related to this:
    a. Support for detail templates in portrait mode has been added. In a template file you can now define a template for landscape mode and a template for portrait mode. The portrait mode template needs to start with <*!--Portrait-->*. The landscape mode template may start with <*!--Landscape-->*. Existing templates will still work (same template will be used in both orientations)
    b. CMA can now detect the orientation of the device and the change of the orientation.

    c. When in portrait mode (width is less than height), some of the controls on the CMA screen are no longer visible (they are collapsed). The functionality of those controls is then accessible through commands in the menu control that is visible on the left top corner of the screen. When in portrait mode, the data grid and the detail both have the same height (in landscape mode the height of the data grid is twice the height of the details).
18. CMA now logs the ID's of the downloaded topic cases.
19. When receiving a REOPEN_REQ, RELEASE_REQ or CLOSE_REQ for a parent case of a group (**CMA_GroupType**=gParent), then CMA will apply this to all cases of the group.
20. The handling of the **CaseNote** field from the launcher case has been improved. When not empty it will be used in the details template to replace the $CaseNote instead of the value of the CaseNote field in the contact data. When not empty, this value will also be used when editing the contact data of the launcher case.
21. French and German translations of CMA have been added (many thanks to FOD Economie, Belgium for providing them).