

DIM – Device Instrument Manager

Max Malhotra, University of Michigan Survey Research Center

1. Abstract

Using Blaise 5 for interviewer-administrated surveys pose unique technical challenges, some of these technical challenges include running offline distributed surveys. To address these challenges we developed an in-house application called Device Instrument Manager (DIM).

We required a mechanism that was robust where an interviewer could synchronize their laptop with a central repository and receive the components that are pertinent to them and be confident they could conduct offline interviews in a seamless manner and once their tasks were completed and they had internet access they can synchronize their data back to the main repository.

This paper will discuss two major components of the DIM and some of the subcommands.

2. Introduction to DIM

Prior to the development of DIM, we did not have a mechanism in place that could be coupled with an in house sample management system to provide a succinct user experience and to address the delivery and subsequent data collection which worked for both single-mode or mixed-mode environments.

This paper will discuss two major components of the DIM and some of the subcommands this application consists of such as.

1. Blaise Sync:
 - a. Check Version
 - i. Download Instruments
 - ii. Perform Data Model Migrations
 - b. Upload Cases
 - c. Download Cases
2. Run Survey
 - a. Run Pre App
 - b. Start Survey
 - c. Run Post App
 - d. Pull Values

This paper will delve into how and why the above commands were required and will talk briefly about each command's implementation strategy, as well as go into our future plans for utilizing the write and download interceptors more extensively. We will also touch on some of the trials and tribulations we faced along the way until we got to a successful outcome.

There are also other subcommands that we will discuss in this paper that complement and help to facilitate a smooth flow and experience, those subcommands are not cupped with major commands such as Blaise Sync and Run Survey.

3. Background

Prior to the creation of DIM, we evaluated a number of design options that would best fit our organizational long term needs to determine what direction the design and development process will adhere too. Some of the Implementation Options and Top Concerns we had were:

- 1) Leverage native Blaise upload/download services.
 - a) CBS is willing to provide instruction, but if SRO's needs do not align with the design of Blaise's services, CBS may not be willing and will not be able to customize their services to SRO's exact needs on SRO's timeline.
 - b) Does this option push SRO toward long-term reliance on a single vendor?
- 2) Leverage native Microsoft Sync Framework (SQL Server synchronization utilities).
 - 1) The Sync Framework is a robust toolkit, but extensive technical design and development are required to tailor the tools to SRO's needs, which may prevent us from achieving full functionality on our timeline and within our requested budget.
- 3) Create SRO Blaise Sync Service from custom components, extending approach used with Blaise 4 and Blaise 5 with SurveyTrak.
 - 1) These technologies may not be the most efficient way to implement mixed-mode data transfer.
 - 2) The estimated effort may prevent us from achieving full functionality on our timeline and within our requested budget.

During the evaluation period we kept the following project scope in mind:

- 1) Start the survey on the server before downloading it.
- 2) Download new instruments to laptop.
- 3) Download a new case.
- 4) Resume survey on laptop after download.
- 5) Start survey on laptop after download.
- 6) Upload case data for a single case.
- 7) Upload case data for multiple cases.
- 8) Resume survey on the server after upload.
- 9) Download an existing instrument.
- 10) Download an existing case.
- 11) Delete case from the laptop (end of protocol).
- 12) Delete instrument from the laptop (end of project).

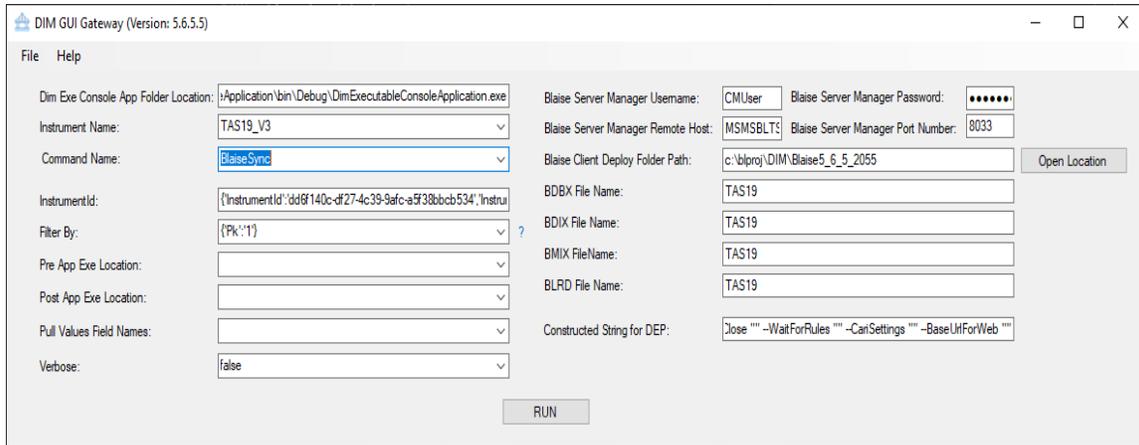
After we carefully evaluated each option it was decided that option 1 (Leverage native Blaise upload/download services) met our needs and timeline best.

At this point, DIM also leveraged the Start Kit Project and the Dep App and utilized them as a baseline during the development process.

4. Overview

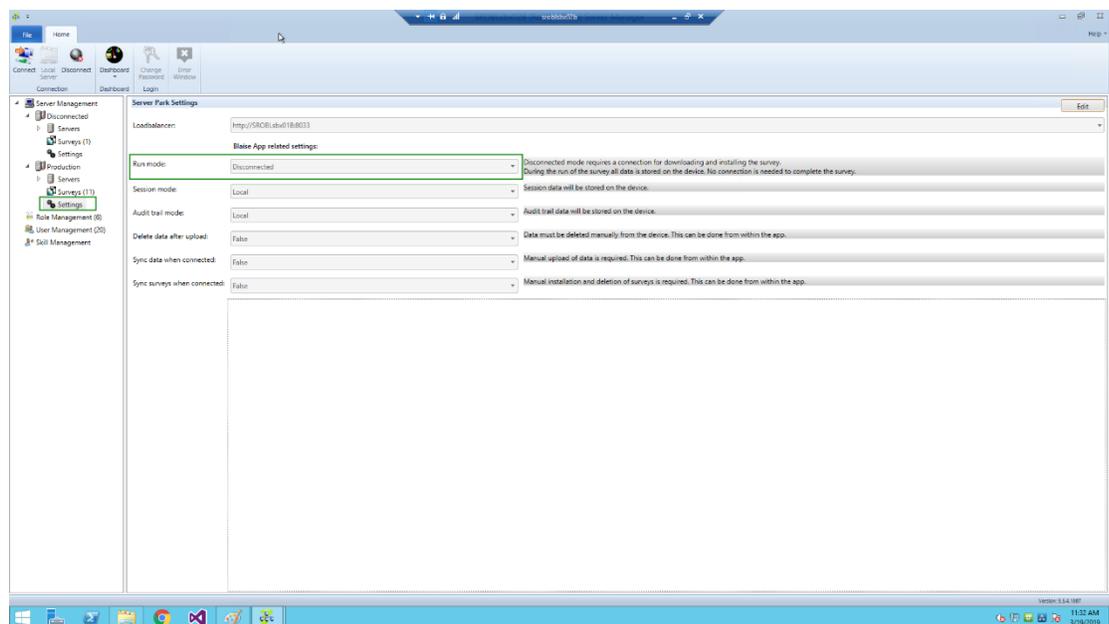
DIM is a C# .NET program that is the underlying core of how offline single-mode and mixed-mode cases are delivered to interviewers in our organization. DIM can be called completely independently for standalone testing purposes by utilizing a command-line interface and supplying the appropriate parameters for the command being executed.

During our development phase of the process, we took this a step further and developed a supplemental application called the DIM GUI Interface to serve as a bridge to help facilitate rapid testing and project integration while the in house sample management system, Michigan Sample Management System (MSMS) was being developed and downstream phases which allowed for a coupling between MSMS and DIM were being planned for implementation with DIM. During this phase the DIM GUI Application provided a Graphical User Interface for the DIM Executable Console Application this greatly aided in the development process and also helped to isolate any downstream bugs to either the DIM application (with the underlying Application Programming Interface (APIs) being utilized) or to the calling components of MSMS, this way we know where to allocate our resources to quickly address the issues being discovered.

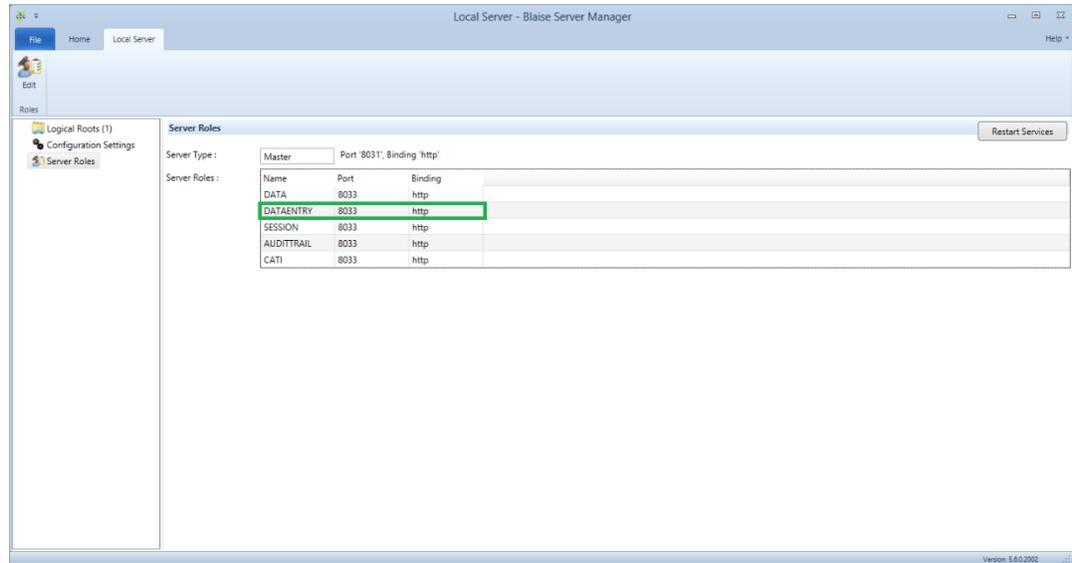


5. Prerequisites

- Run Mode on the Server Park in question MUST be set to Disconnected for DIM application to work. There is NO harm in setting it to Disconnected VS Client Server as Disconnected simply creates more files such as app.bpkg that DIM can utilize.



- On the **01** (Web server) > Right-click Blaise Server Manager > Run as Administrator > Click cancel on the prompt to log in > Click on Logical Servers > Click on Server Roles > Verify **Data Entry** role is present if not Edit and add one. Without this you cannot log in to the remote host of ***01 server.



- .NET Framework 4.7.2 or higher needs to be installed on the client machine
- Ideally, the setup project (Data Model) should be built in the version being tested and MATCH the NuGet **packages** the version of DIM is utilizing.
- Check that the latest version of SerialNumber.txt is present in the root directory of the DIM application otherwise DIM command calls such as Remove Cases Command that initiates the API to call datalink.Delete() will fail due to "Operation is only allowed with a valid license."
- Make sure that you have unchecked the Prefer 32-bit checkbox (for Both Debug and Release Configuration) at the project's Build tab (In Visual Studios).
 - Otherwise, you will get an SQL lite exception and will not be able to use AnyCpu and then forced to use x64.

6. DIM Commands – Inputs and Outputs

This section describes what comprises each of the individual DIM commands and how to initiate the command and what a typical JSON output result will look like for that command.

- Blaise Sync Flow**

```
Blaise Sync
Check Version
  string InstrumentId
  string InstrumentName
```

int **Order**
 bool **InstalledOnClient**
 bool? **MigrationCompletionFlagFound** (is a nullable bool)

Upgrade Instrument - If needed based on Check Version

Download Instrument - Download All instruments requested first.

Data Migration - If a **MigrationCompleted.txt** file does not exist for that DM

UploadloadCases - On the LAST instrument - Need to Loop through

DownloadCases - On the LAST Instrument - Need to Loop through

- **Blaise Sync Command**

- **Command Prompt:**
- `.\DimExecutableConsoleApplication.exe -c "BlaiseSync" -i "{InstrumentId:'dd6f140c-df27-4c39-9afc-a5f38bbcb534',InstrumentName:'TAS19_V1',Order:'1'},{'InstrumentId':'a9eef537-7109-4a0f-bad4-a59ae5ea35b8',InstrumentName:'TAS19_V2',Order:'2'},{'InstrumentId':'ed139276-b864-4fe2-8d09-a73d95411d90',InstrumentName:'TAS19_V3',Order:'3'}" -f "{Pk:'1'}" -r "MSMSBLTSTNXT02.isr.umich.edu" -u "CMUser" -p "XXXXXXXXXX" -n "8033" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19"`
- **OUTPUT:**

Name	Value
CommandName	"BlaiseSync"
CommandResults	...
CountAffected	1
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"BlaiseSync Ran Successfully."
Success	true

- **CheckVersion Command:**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication. -c "CheckVersion" -i {"InstrumentId':'dd6f140c-df27-4c39-9afc-a5f38bbcb534','InstrumentName':'TAS19_V1','Order':'1'},{'InstrumentId':'a9eef537-7109-4a0f-bad4-a59ae5ea35b8','InstrumentName':'TAS19_V2','Order':'2'},{'InstrumentId':'ed139276-b864-4fe2-8d09-a73d95411d90','InstrumentName':'TAS19_V3','Order':'3'}}" -f "{Pk':'1'}" -v "false" -r "MSMSBLTSTNXT02.isr.umich.edu" -u "CMUser" -p "XXXXXXXXXX" -n "8033" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19"`

- **OUTPUT:**

Name	Value
CheckVersionResults	...
CommandName	"CheckVersion"
CommandResults	...
CountAffected	3
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"CheckVersion Ran Successfully."
Success	true

- **Return Object**

```
string InstrumentId
string InstrumentName
int Order
bool InstalledOnClient
bool? MigrationCompletionFlagFound
```

- **Download Instrument Command:**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication.exe -c "DownloadInstrumentCommand" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -r "MSMSBLTSTNXT02.isr.umich.edu" -u "CMUser" -p "XXXXXXXXXX" -n "8033" -d "c:\blproj\DIM\Blaise5_6_5_2055"`

- **OUTPUT:**

JSON

- CommandName : "DownloadInstrumentCommand"
- InstrumentId : "ed139276-b864-4fe2-8d09-a73d95411d90"
- InstrumentName : "TAS19_V3"
- CountAffected : 1
- Success : true
- Message : "Instrument Name: TAS19_V3 and Instrument ID: ed139276-b864-4fe2-8d09-a73d95411d90"
- FailureReason : 0
- ExitCode : 0
- ExitCodeDescription : "Success"
- CommandResults

Name	Value
CommandName	"DownloadInstrumentCommand"
CommandResults	...
CountAffected	1
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"Instrument Name: TAS19_V3 and Instrument ID: ed139276-b864-4fe2-8d09-a73d95411d90"
Success	true

- **Upload Cases Command**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication.exe -c "UploadCasesCommand" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -f "'Pk':1','Pk':2','Pk':3}" -r "MSMSBLTSTNXT02.isr.umich.edu" -u "CMUser" -p "XXXXXXXXXX" -n "8033" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19"`

- **OUTPUT:**

JSON

- UploadedCases : "1,2,3"
- NotUploadedCases : ""
- NotAttemptedUploadedCases : ""
- CommandName : "UploadCasesCommand"
- InstrumentId : "ed139276-b864-4fe2-8d09-a73d95411d90"
- InstrumentName : "TAS19_V3"
- CountAffected : 3
- Success : true
- Message : "Number of Cases Uploaded: 3"
- FailureReason : 0
- ExitCode : 0
- ExitCodeDescription : "Success"
- CommandResults

Name	Value
CommandName	"UploadCasesCommand"
CommandResults	...
CountAffected	3
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"Number of Cases Uploaded: 3"
NotAttemptedUploadedCases	""
NotUploadedCases	""
Success	true
UploadedCases	"1,2,3"

- **Download Cases Command**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication.exe -c "DownloadCasesCommand" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -f "{Pk:'1'};{Pk:'2'};{Pk:'3'}" -r "MSMSBLTSTNXT02.isr.umich.edu" -u "CMUser" -p "XXXXXXXXXX" -n "8033" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19"`

- Note: Download Cases Command requires that -f (FilterArgument) is present to execute.

- **OUTPUT:**

Name	Value
CommandName	"DownloadCasesCommand"
CommandResults	...
CountAffected	3
DownloadedCases	"1,2,3"
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"Number of Cases Downloaded: 3"
NotDownloadedCases	""
Success	true

- **Run Survey Command**

- **NOTE:**

- Commands run in this sequence:

- RunPreApp
 - optional pram of --PreAppExeLocation "C:\Windows\System32\notepad.exe", you do not have to pass it in the below command at all **or** you can pass an --PreAppExeLocation "" if you want to skip it).
 - StartSurveyCommand
 - RunPostApp
 - optional pram of --PostAppExeLocation "C:\Windows\System32\notepad.exe", you do not have to pass it in the below command at all **or** you can pass an --PostAppExeLocation "" if you want to skip it).
 - PullValues
 - If you want to skip this command you can pass an empty string such as: --PullValuesFieldNames "" if that is done then the command will still return a 0 value and return as a success since it presumes you meant to pass nothing and skip it.

- **Command Prompt:**
- `.\DimExecutableConsoleApplication.exe" -c "RunSurvey" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -f "{PK:'1'}" --PreAppExeLocation "C:\Windows\System32\notepad.exe" --PostAppExeLocation "C:\Windows\System32\notepad.exe" --PullValuesFieldNames "{Fn:'Field1'}" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19" --LayoutSetGroup "Interviewing" --LayoutSet "Large" --DataEntrySettings "StrictInterviewing" --Language "" --Fields "xTimeGate=1" --AssignMode "Always" --StartParallel "" --RunMode "ThickClient" --InitialWindowState "Maximized" --EnableResize "False" --EnableClose "False" --WaitForRules "" --CariSettings ""`
- **OUTPUT:**

Name	Value
CommandName	"RunSurvey"
CommandResults	...
CountAffected	0
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"RunSurvey Ran Successfully."
Success	true

- **RunPreApp Command**

- **Command Prompt:**
 - `.\DimExecutableConsoleApplication.exe -c "RunPreApp" -i "ed139276-b864-4fe2-8d09-a73d95411d90" --PreAppExeLocation "C:\Windows\System32\notepad.exe" -d "c:\blproj\DIM\Blaise5_6_5_2055"`
- **OUTPUT:**

JSON

- CommandName : "RunPreApp"
- InstrumentId : "ed139276-b864-4fe2-8d09-a73d95411d90"
- InstrumentName : "TAS19_V3"
- CountAffected : 1
- Success : true
- Message : "RunPreApp Ran Successfully."
- FailureReason : 0
- ExitCode : 0
- ExitCodeDescription : "Success"
- CommandResults

Name ▲	Value
CommandName	"RunPreApp"
CommandResults	...
CountAffected	1
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"RunPreApp Ran Successfully."
Success	true

- **Start Survey Command**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication.exe -c "StartSurveyCommand" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -f "{PK:'1'}" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19" --LayoutSetGroup "Interviewing" --LayoutSet "Large" --DataEntrySettings "StrictInterviewing" --Language "" --Fields "xTimeGate=1" --AssignMode "Always" --StartParallel "" --RunMode "ThickClient" --InitialWindowState "Maximized" --EnableResize "False" --EnableClose "False" --WaitForRules "" --CariSettings ""`

- **OUTPUT:**

JSON

- CommandName : "StartSurveyCommand"
- InstrumentId : "ed139276-b864-4fe2-8d09-a73d95411d90"
- InstrumentName : "TAS19_V3"
- CountAffected : 1
- Success : true
- Message : "StartSurveyCommand Ran Successfully."
- FailureReason : 0
- ExitCode : 0
- ExitCodeDescription : "Success"
- CommandResults

Name ▲	Value
CommandName	"StartSurveyCommand"
CommandResults	...
CountAffected	1
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"StartSurveyCommand Ran Successfully."
Success	true

- If no value is supplied generally a generic Blaise page will come up which will ask you for your key value. (This is not advisable)
 - Example of Command: `.\DimExecutableConsoleApplication.exe -c "StartSurveyCommand" -i "ce0c7770-c3a1-44c7-b804-7ed0e11d37a7" -f ""`

SRC Screen Design Guidelines (V.1.10)

Preload-Sample ID

- **RunPostApp Command**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication.exe -c "RunPostApp" -i "ed139276-b864-4fe2-8d09-a73d95411d90" --PostAppExeLocation "C:\Windows\System32\notepad.exe" -d "c:\blproj\DIM\Blaise5_6_5_2055"`

- **OUTPUT:**

Name	Value
CommandName	"RunPostApp"
CommandResults	...
CountAffected	1
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"RunPostApp Ran Successfully."
Success	true

JSON
CommandName : "RunPostApp"
InstrumentId : "ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName : "TAS19_V3"
CountAffected : 1
Success : true
Message : "RunPostApp Ran Successfully."
FailureReason : 0
ExitCode : 0
ExitCodeDescription : "Success"
CommandResults

- **Pull Values Command**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication.exe" -c "PullValues" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -f "{Pk:'1'}" --PullValuesFieldNames "{Fn:'Field1'}, {Fn:'Field2'}, {Fn:'Field3'}, {Fn:'Field4'}, {Fn:'Field5'}, {Fn:'Field6'}, {Fn:'Field7'}" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19"`

- **OUTPUT:**

Name	Value
CommandName	"PullValues"
CommandResults	...
CountAffected	7
CountAffectedPullValues...	0
CountAffectedPullValues...	7
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"PullValues Ran Successfully."
PullValueResults	...
Success	true

- **Remove Instrument Command:**

- **Command Prompt:**

- `.\DimExecutableConsoleApplication. -c "RemoveInstrumentCommand" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19"`

- **OUTPUT:**

Name	Value
CommandName	"RemoveInstrumentCommand"
CommandResults	...
CountAffected	1
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"Instrument Name: TAS19_V3 and Instrument ID: ed139276-b864-4fe2-8d09-a73d95411d90"
Success	true

- **Remove Case Command:**

- **Command Prompt:**

- .\DimExecutableConsoleApplication. -c "RemoveCaseCommand" -i "ed139276-b864-4fe2-8d09-a73d95411d90" -f '{"Pk': '1'}" -d "c:\blproj\DIM\Blaise5_6_5_2055" --InstrumentName "TAS19_V3" --BdbxFileName "TAS19" --BdixFileName "TAS19" --BmixFileName "TAS19" --BlrdFileName "TAS19"

○ **OUTPUT:**

JSON

- CountAffectedRemoveCasesOriginalDb : 1
- CountAffectedRemoveCasesUpdatedDb : 0
- CountAffectedRemoveCasesTotal : 1
- CountAffectedRemoveCasesSuccess : 1
- CountAffectedRemoveCasesFail : 0
- CommandName : "RemoveCaseCommand"
- InstrumentId : "ed139276-b864-4fe2-8d09-a73d95411d90"
- InstrumentName : "TAS19_V3"
- CountAffected : 1
- Success : true
- Message : "Instrument Name: TAS19_V3 and Instrument ID: ed139276-b864-4fe2-8d09-a73d95411d90"
- FailureReason : 0
- ExitCode : 0
- ExitCodeDescription : "Success"
- CommandResults

Name	Value
CommandName	"RemoveCaseCommand"
CommandResults	...
CountAffected	1
CountAffectedRemoveCa...	1
CountAffectedRemoveC...	0
CountAffectedRemoveC...	1
CountAffectedRemoveC...	1
CountAffectedRemoveC...	0
ExitCode	0
ExitCodeDescription	"Success"
FailureReason	0
InstrumentId	"ed139276-b864-4fe2-8d09-a73d95411d90"
InstrumentName	"TAS19_V3"
Message	"Instrument Name: TAS19_V3 and Instrument ID: ed139276-b864-4fe2-8d09-a73d95411d90"
Success	true

7. DIM Commands Explained

7.1 Check Version

In the check version command, DIM examines what versions are currently installed on the interviewer laptop machine and compares it to what versions are being requested from the server.

Then check version downloads any instruments in sequence (future versions) that are not currently present on the interviewer laptop.

Finally check version command checks what is the current versions of the Data Model (DM) are installed on the interviewer's laptop, then migrates the data from the current version to each of the new future versions installed on the laptop in the sequence provided a previous migration had not taken place.

7.2 Upload Cases

Uploads the requested cases from the interviewer laptop to the server. When it makes it to the server the Write Interceptor gets triggered server-side and performs the necessary logic to based on business rules and ultimately moves the case to the appropriate database.

The audit data for the cases specified is automatically removed upon successful upload.

7.3 Download Cases

Downloads the requested cases from the server to the interviewer laptop. Currently, we are not utilizing the Download Interceptor but has the potential to be called in future versions where business logic will be run prior to the downloaded cases making it to the appropriate database on the interviewer laptop.

7.4 Run Pre App | Run Post App

The Run Pre App and Run Post App can call external applications prior to and after the survey are conducted. This can be used to create applications that perform complex logic outside of the data model to perhaps spawn new lines based on a preview respondent answer or any other logic that is desired and can be programmed.

The Run Pre App and Run Post app expect an application-level Exit Code of 0 to consider it a success otherwise DIM will break out of the application.

7.5 Start Survey

The Start Survey command launches the custom DEP and launches the instrument.

The custom dep window is designed to check for error handling so if the instrument returns an error internal to the instrument it will return that back to DIM so it can perform the appropriate logic.

The custom dep is also accounting for event handlers such as aborted, completed, and so forth. The custom dep is also set up to handle desired parameters such as always launching maximized and not showing the close button to the interviewer so we can internally control the interviewer experience.

An example of the custom dep can launch an instrument can be seen below.

Respondent: ANNA -- MARTIN

◆ Verify that you are speaking to the correct TAS Respondent before proceeding:

Before I get started, I would like to make sure I am speaking with the right person. You are ANNA - MARTIN? If No, ASK: Do you ever go by that name?

◆ If R never goes by the preloaded name, suspend the interview and seek out the correct Respondent

◆ Minor name corrections can be made in the contact information update screens at the end of the interview

1. Yes
 5. No

7.6 Pull Values

The Pull Values command returns back all the values for the fields being requested for a specific SampleId. It will also tell you the field data type and also let you know if that field is present inside the database.

7.7 Download Instrument

The Download Instrument command is capable of downloading one or many instruments in the order supplied.

7.8 Remove Instrument

The Remove Instrument command removes the specific instrument specified and its associated files.

7.9 Remove Cases

The Remove Cases command removes the specific cases specified and associated session data for that case.

8. DIM Application Usage – End Users

8.1 Intended Audience of the Application

The target audience/end-user of this application would be personnel who administer Computer Assisted Telephone Interviews (CATI) and Computer Assisted Personal Interviewing (CAPI). In the case of our organization, this means both in-house -- our Survey Services Lab (SSL), and field personnel, depending on project need, the majority of the use will be for CAPI.

8.2 Michigan Sample Management System (MSMS)

DIM was designed to be able to be coupled within a house sample management systems, in fact, we built it with this in mind and thus it lends itself well to this purpose as each command and subcommand is designed to provide appropriate notifications back in JSON format to the calling system/application.

The calling system/application will take notifications and perform the necessary business logic that is appropriate for the organization. Time/resources and effort are required to make this coupling happen in a seamless manner and have DIM fully integrate with the in-house sample management system.

9. Trials and Tribulations

We have faced many hurdles along the way and continue to work through issues in conjunction and in cooperation with CBS. Issues such as the Write Interceptor not causing a lockout condition where excessive system memory usage was detected, if this issue cannot be resolved quickly, we will have to reexamine our system workflow for when certain conditions are met such as in the case of mixed-mode interviews.

In the past, we discovered issues with how some APIs did not function the way we desired such as the deploy folder always having a static location and we required dynamic capabilities for that, and other parameters and CBS were able to accommodate in future versions.

We also ran in too much more complex issues to diagnose such as when DIM called the Start Survey Command and launched an instrument the majority of the system instrument looked correct however some lookup tables would behave oddly and not give the desired result. After some back and forth CBS was able to resolve an underlying API problem.

We have countless examples of the above items but the major take away is that it took time and effort and a lot of back and forth but with the help of CBS we were able to resolve the issues as they were discovered.

10. Conclusion

The DIM application was developed to provide a mechanism to conduct single and mixed-mode interviews for offline distributed surveys. The goal was to create a back-end application with the flexibility to serve the needs of many projects without increasing a burden on interviewers and Blaise programmers.

DIM plays an integral role in our organization in facilitating Computer Assisted Interviewing (CAI) studies. It is used to conduct interviews and provide for succinct user experience to the interviewers were

the complexity of synchronization, data migration and data transfer are neutralized from the interviewer viewpoint and the necessary tools are provided to the interviewer to succeed in the task of collecting single-mode and mixed-mode surveys seamlessly. Data validity during the synchronization process is crucial to provide assurance of proper data integrity through the entire system and DIM provides this assurance through intensive code reviews and regression testing conducted.

DIM has been built out using an approach that lends itself well for future development and growth wherein each of its commands and subcommands can be quickly coupled together to produce new commands and has the ability to leverage core Blaise components to keep pace with Blaise development wherein each new version of Blaise is linked to a new version of DIM allowing for DIM to leverage changes made inside the Blaise .dlls through the use of NuGet packages.

In closing, DIM serves as an important utility in our organization for Computer-Assisted Personal Interviewing (CAPI) and on our end we are continuing to put the DIM application through its paces by utilizing new data models as they become available and we continue to try to improve upon it as new feature requests are made.

It has been a long road getting to this point due to running into difficulties and due to project scope design changes such as deciding to implement Blaise Sync using the Record Filter approach as oppose to an more automated Blaise sync using a offline CAPI block in the Blaise database as the driving factor, as well as other various hurdles along the way, but what we can say is that the CBS team has provided excellent support every step of the way and working together we were able to overcome numerous challenges and come out with a viable product that is used in production projects.

In the future, we wish to develop additional functionality that allows for DIM to be further streamlined for deployment. We also wish to work closely with the CBS team to help resolve issues we have encountered with the Write Interceptor as that would be a critical path division if we do not come up with a resolution strategy.

11. References

Blaise 5 Help. (2020). '<http://help.blaise.com>'

Blaise 5 Starter Kit Project. (2019).

Blaise 5 Dep App. (2019).

University of Michigan Survey Research Center. (2019). 'Blaise 5 Integration Documentation'