

# Automatic Upgrade of Production Blaise Instruments

*Lilia Filippenko Joseph Nofziger & Gilbert Rodriguez, RTI International*

## 1. Introduction

Sometimes it is necessary to upgrade instruments when data collection has begun. But for modifications that alter the structure of a Blaise data model, it is difficult to do: an existing production Blaise data files must be upgraded with the new data model when the data model has changed. Although the Manipula setup for this procedure is very simple, many steps need to be taken to avoid loss of data: creation of backup folders, copying of a number of files, the preparation of Manipula setups for the upgrade and so forth. It can be an extremely complex process, especially when instruments reside on multiple computers, possibly with different versions.

At RTI International we have projects that use Blaise data models for CAPI, CATI, and ACASI instruments. A Master database resides at RTI, but individual interview cases are assigned to different Field Interviewers who use laptops to conduct the interviews. The cases are therefore distributed among laptops. When an interview for a case is completed, its data file will be sent back for loading into the Master database. When a case is transferred to another interviewer, its data files are transferred between laptops. An upgrade can of course occur at any point. So applying the upgrade means upgrading the Master database, data files on a number of laptops (it could be hundreds), and - posing a particular challenge - many cases that are in transit.

A Visual Basic program was written to perform these tasks. A key piece of data for the program is the version number of the instrument. Every time the data model is changed, it should be prepared using the corresponding Blaise project with a new version number.

Several additional requirements drove the design of the program:

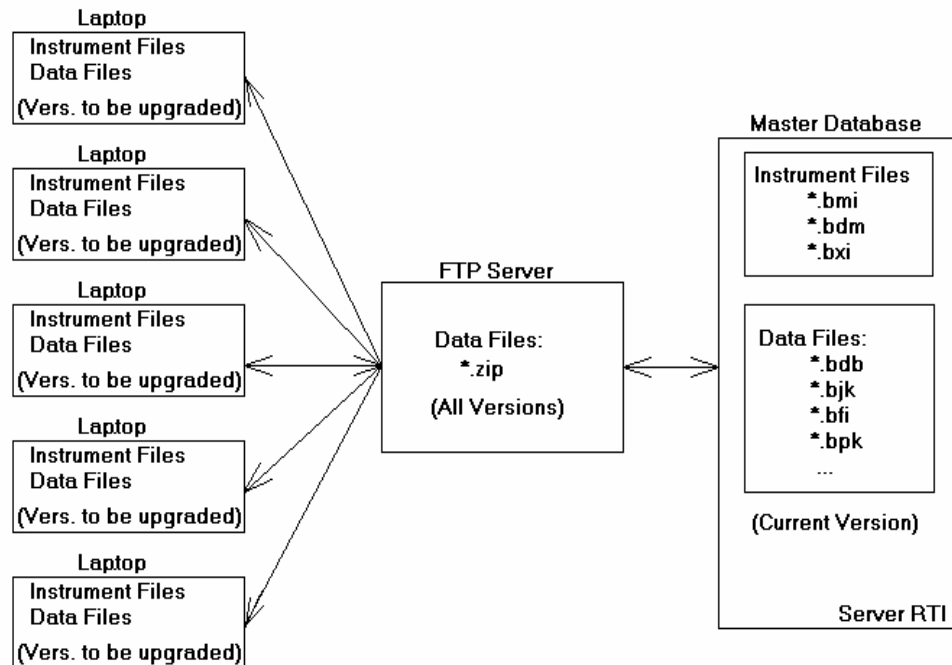
- Upgrades must be reliable - all data and other Blaise files are backed up before upgrading, and restored on failure.
- Manipula setups that depend on the data model must be prepared whenever the Master database is successfully upgraded.
- It must be possible to upgrade from any previously released version.
- And finally, we must be able to monitor whether field laptops have been upgraded successfully.

## 2. General Approach

The usual structure of the project is shown in Figure 1. The Master database resides on the server at RTI and contains a current version of the project: the instrument and the data files (all interview cases). Before project is started, all needed software and instrument files are loaded on laptops at RTI. Laptops are then issued to Field Interviewers. The cases are initialized in the Master database, placed on an FTP server in individual files, and sent to laptops when the project is started. The same server is used to transfer data back to the Master database during data collection.

The structure of the data files is defined in the project design stage and normally could not be modified after data collection has begun. However, such modifications are often desirable. An upgrade of the project could be a result of the customer's late request, hardware modification, or bug found in one of the project components. To do so we have to upgrade the instrument files at each laptop as well as the data files, which already contain collected data.

Figure 1



However, the Blaise software provides a tool to make such an upgrade possible and easy. The main idea is to utilize the version number, which exists for each project, and which is assigned to every element of the project: instrument and data files. When the Master database is upgraded, each component inside the database receives a new version number, and this new number is the key to a successful upgrade of the project components on the laptops even after the data collection has begun.

To process the upgrade a special Visual Basic application, UBI (Upgrade Blaise Instrument) is created and distributed to all laptops. The UBI first handles the upgrade of the Master database components. During the upgrade of the Master database the UBI prepares Manipula setups to be used later in upgrading the laptop's data files. We should, however, remember that versions of the case data files on the FTP server could differ from the current Master database version, as well as from the laptop's files. This is because the previous upgrades (if any) upgraded the data files on the laptops and in the Master database, but not on the FTP server. Therefore, the UBI requires special Manipula setups to handle conversion from any old version to the current one for individual case data files.

### 3. Description of the UBI

#### 3.1 Naming convention for directories and Manipula setups

An important part of the upgrade is maintaining the upgrade history and backup files on each machine. It will be used as a backup copy if the upgrade fails. Also, we need to have all previous instrument files in case we need to upgrade case data files that are significantly older than the current version.

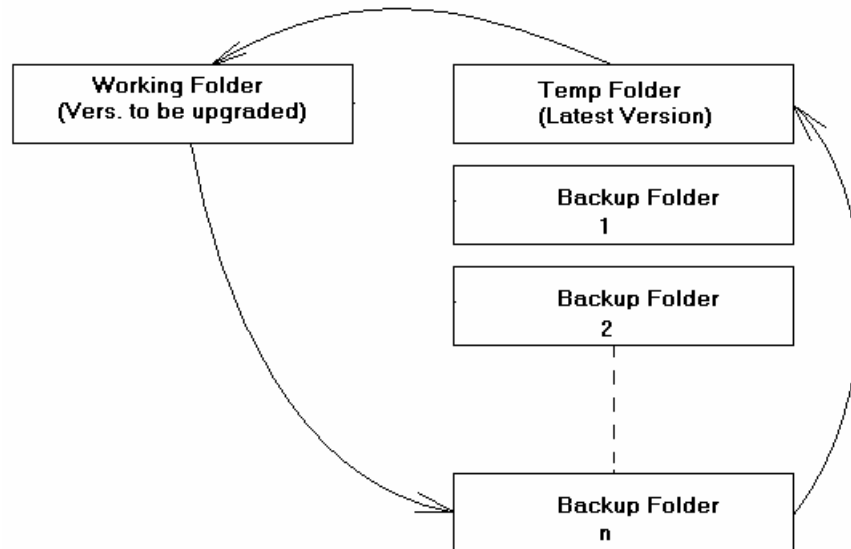
A special rule on the naming of backup directories names provides a way to keep records on the history of upgrades: the name consists of the instrument name and the version number separated by an underscore:

<inst>\_<version>

This allows us to provide the UBI with a simple way to identify the location of the files that are necessary for handling an upgrade from any older version.

A typical directory structure is shown in Figure 2.

**Figure 2**



Names for the Manipula setups that perform the upgrade include both version numbers: the old version and the latest one. This is used to determine what Manipula setup should be used to do the upgrade:

<SetupName><OldVersion>To<NewVersion>

### **3.2 Modes used with UBI**

The UBI is used in three modes depending on the kind of data files that need to be upgraded: Master database, database on laptops, and data files for individual cases.

### **3.2.1 Upgrade of Master database**

When the decision has been made to upgrade an instrument that is already in the production stage, the upgrade should be applied first to the Master database.

Our approach for upgrading to the Master database is as follows:

- Determine the version of the current instrument in the working folder
- Create backup folder with all current instrument files
- Upgrade data files from the backup folder to the temp folder
- Copy new instrument files and upgraded data files from the temp folder to the working folder

To perform these major steps, the latest version of instrument files is placed in the temp folder. When the UBI is started, the current version – prior to upgrade - all instrument files, data files, and Manipula setups are copied from the working folder to a new backup folder. This will be the base version for the upgrade, and will remain for future upgrades of individual cases. It also serves as a backup in case the upgrade fails.

The UBI automatically creates the Manipula setup and calls the B4Cpars.exe utility to prepare the Manipula setup. It then invokes the prepared Manipula setup to perform the upgrade. Only after successful completion of this process will the instrument and data files in the working folder be replaced with the new instrument and data files from the temporary folder. This allows us to perform any necessary tests on the new data files before committing the upgrade to the working folder.

Then the UBI creates and prepares Manipula setups to do an upgrade for individual case data files that will be used later to load cases in the Master database or on a laptop.

During execution of the UBI for upgrading the Master database, messages will be displayed and a log file will be created with additional information for future review.

Flowchart 1 presents a diagram for this mode of the UBI in Appendix A.

### **3.2.2 Upgrade of individual cases**

For transport of individual cases among laptops or between a laptop and a server, we extract case data to a Blaise data file containing only the data for the case. When it reaches its destination, the data file may be an older version than the one into which it is to be loaded. For such cases we will upgrade the data files directly to the working folder rather than to a temporary folder. But we should bear in mind that some cases could be a few versions behind before they reach their destination. Perhaps they are en route from a laptop that had not yet successfully applied the upgrade at the time the case was extracted.

Our approach is as follows:

- Determine the version of the case data file in the working folder
- Copy the file to the appropriate version folder
- Upgrade from the version folder to the working folder

During execution of the UBI for upgrading case data files, information will be added to the log file for future review. On laptops a special upgrade status flag will be created. If the status is “Failure”, information about this particular case is

written to an Access database to be sent later to RTI for review. Such cases will be re-sent to the field laptop.

Flowchart 2 presents a diagram for this mode of the UBI in Appendix A.

### **3.2.3 Upgrade of database on field laptops.**

After successfully upgrading the Master database we will have the new instrument files, new Manipula setups for upgrading database on laptops, and Manipula setups for upgrading cases while loading them into the databases.

If laptops contain additional Manipula setups that refer to the instrument files, those must also be prepared with the new instrument. We prepare these in house. Finally, we create a command script to invoke the UBI on the laptop with the proper parameters.

All these files are uploaded to laptops, where the command script is invoked automatically using existing mechanisms. Most of the same steps described previously for upgrading the Master database will be performed to upgrade the database on the field laptop.

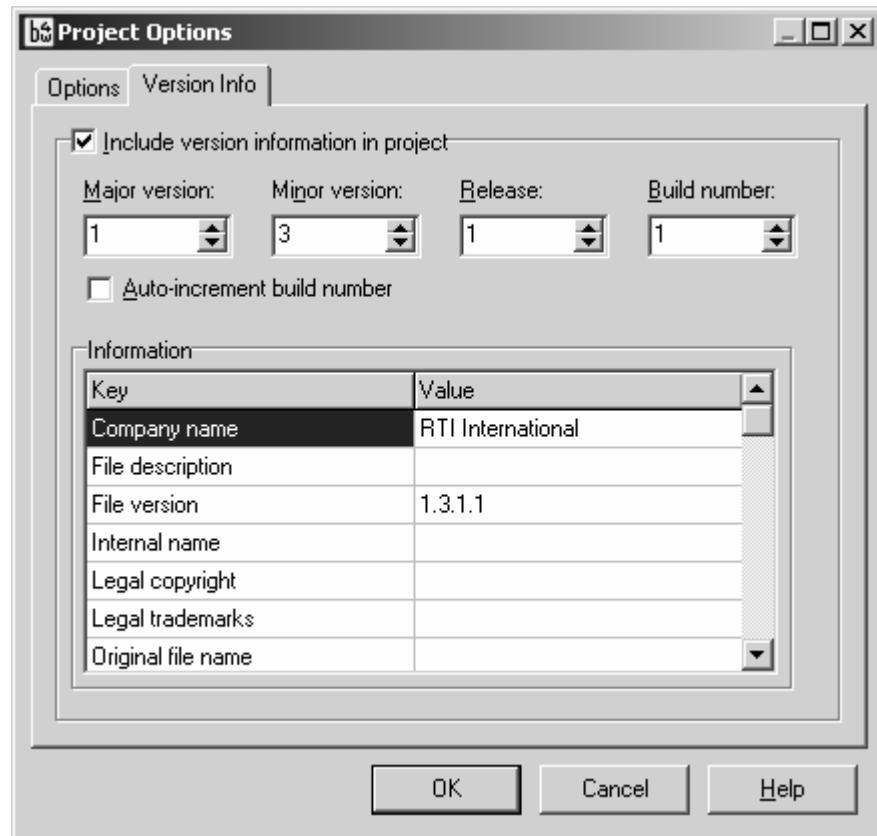
In the laptop environment there will be no user interaction; all messages will be logged. The next time the laptop connects to RTI, the more critical messages will be sent to a server log, including the new instrument version on the laptop. If errors occur and the upgrade fails but the information in the server log is not sufficient to determine what occurred, we can set up a request to retrieve the log file from the laptop for review at RTI.

Flowchart 3 presents a diagram for this mode of the UBI in Appendix A.

### **3.3 Example of Master database upgrade**

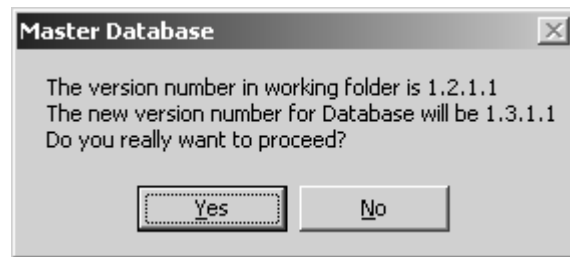
As an example we will use the instrument with the name “hope”. The current version for the instrument “hope” is 1.2.1.1.

When changes to the instrument files require upgrading data files, the new version should be specified for the instrument. This is done in the Control Centre under Project options menu. For this example the new version for the “hope” instrument became 1.3.1.1



New instrument files hope.bmi, hope.bdm, and hope.bxi are copied in the “temp” folder created under the folder where the current (1.2.1.1) version of the “hope” instrument exist.

The UBI accepts the instrument name, working folder name with a full path, and a name for the temporary folder as parameters. When the UBI starts, it will determine the versions of the new instrument in the temporary folder (“temp”) and the old instrument in the working folder.



When we answer “Yes” to start the upgrade, a new backup folder is created under the working folder with the name “hope\_1.2.1.1”. Files from this folder are used for upgrade Master database.

The UBI creates the Manipula setup in the working folder.

**BlzUpdateBlz.man:**

```

USES
Old 'Hope_1.2.1.1\Hope'
New 'temp\Hope'
INPUTFILE InFile : Old ('Hope', BLAISE)
OUTPUTFILE Outfile : New ('Hope', BLAISE)
MANIPULATE
Outfile.WRITE

```

Note that upgraded data files are not created in the working folder but in the “temp” folder. The prepared Manipula setup (BlzUpdateBlz.msu) is created by invoking the B4Cpars.exe utility in the “temp” directory.

The UBI runs the upgrade Manipula setup BlzUpdateBlz.msu and the upgraded data files are created in the “temp” folder. The version number assigned to the new data files should become 1.3.1.1. When the UBI determines that the Master database has been upgraded successfully, all instrument files: “hope.b\*” are copied to the working folder. The working folder now contains version 1.3.1.1 of the “hope” instrument.

New Manipula setups are created for upgrade of individual cases. In this example the “hope” instrument is upgraded for the second time and folder “hope\_1.1.1.1” has instrument files for version 1.1.1.1. Two Manipula setups are created and prepared:

#### **BlzUpdCaseBlz1\_1\_1\_1To1\_3\_1\_1.msu**

```
USES
Old 'Hope_1.1.1.1\Hope'
New 'Hope'
INPUTFILE InFile : Old ('Hope', BLAISE)
OUTPUTFILE Outfile : New ('Hope', BLAISE)
MANIPULATE
Outfile.WRITE
```

#### **BlzUpdCaseBlz1\_2\_1\_1To1\_3\_1\_1.msu**

```
USES
Old 'Hope_1.2.1.1\Hope'
New 'Hope'
INPUTFILE InFile : Old ('Hope', BLAISE)
OUTPUTFILE Outfile : New ('Hope', BLAISE)
MANIPULATE
Outfile.WRITE
```

The last step performed by the program is the deletion of old prepared Manipula setups that are used by other processes, such as loading any finalized cases into the Master database. Those Manipula setups will be prepared again for the new instrument files by other programs.



## **4. Summary**

It may be necessary to upgrade a Blaise instrument once production has begun and the upgrade may alter the structure of the instrument, making any previously generated data incompatible with the upgrade. This methodology upgrades a Blaise instrument and data files without any interruption to data collection and can be

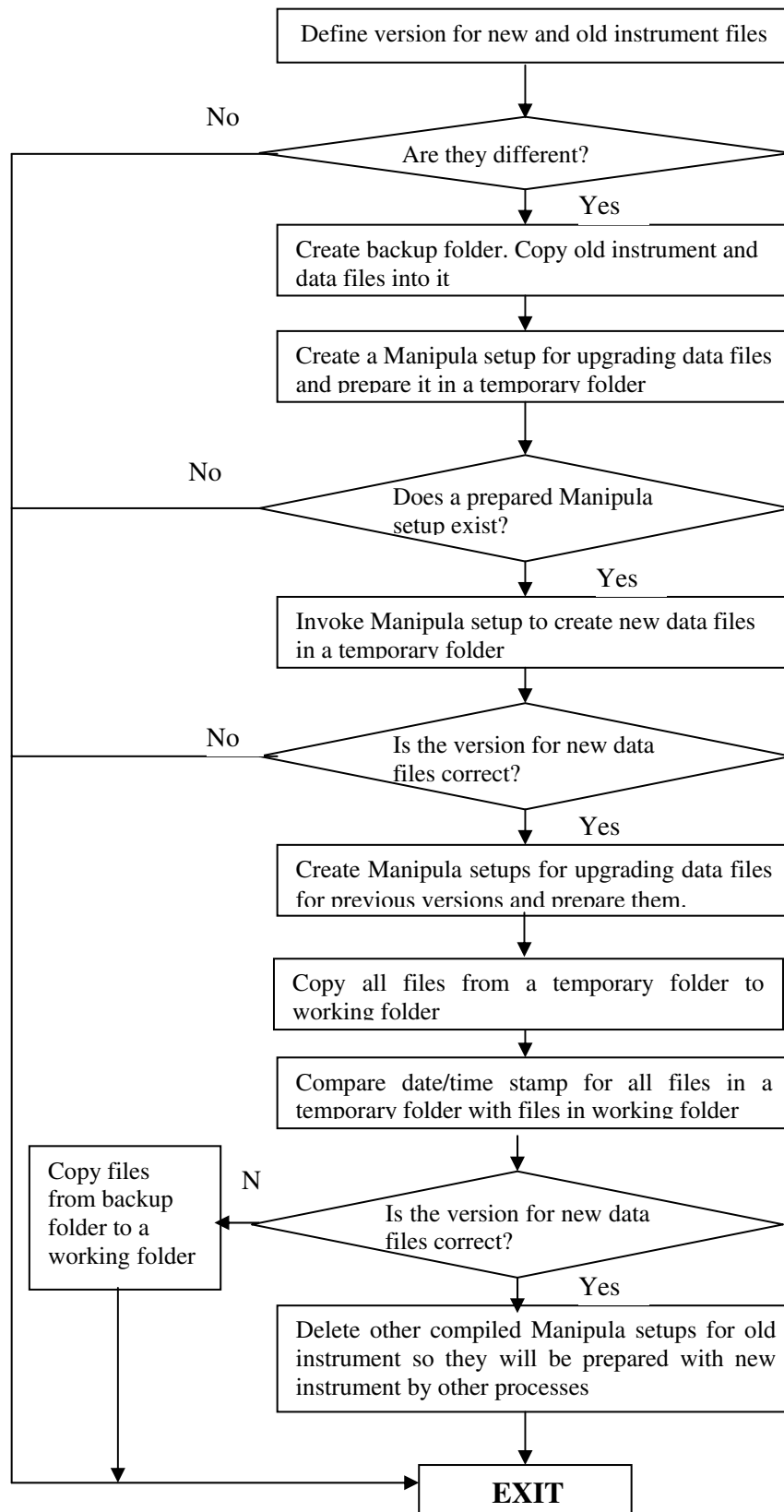
used for different projects that use Blaise data models, resulting in time and cost-effective project maintenance.

Our approach can even be used while an instrument is under development since if the instrument is upgraded and the structure has changed, deleting data files may require re-inputting data, which may be time-consuming.

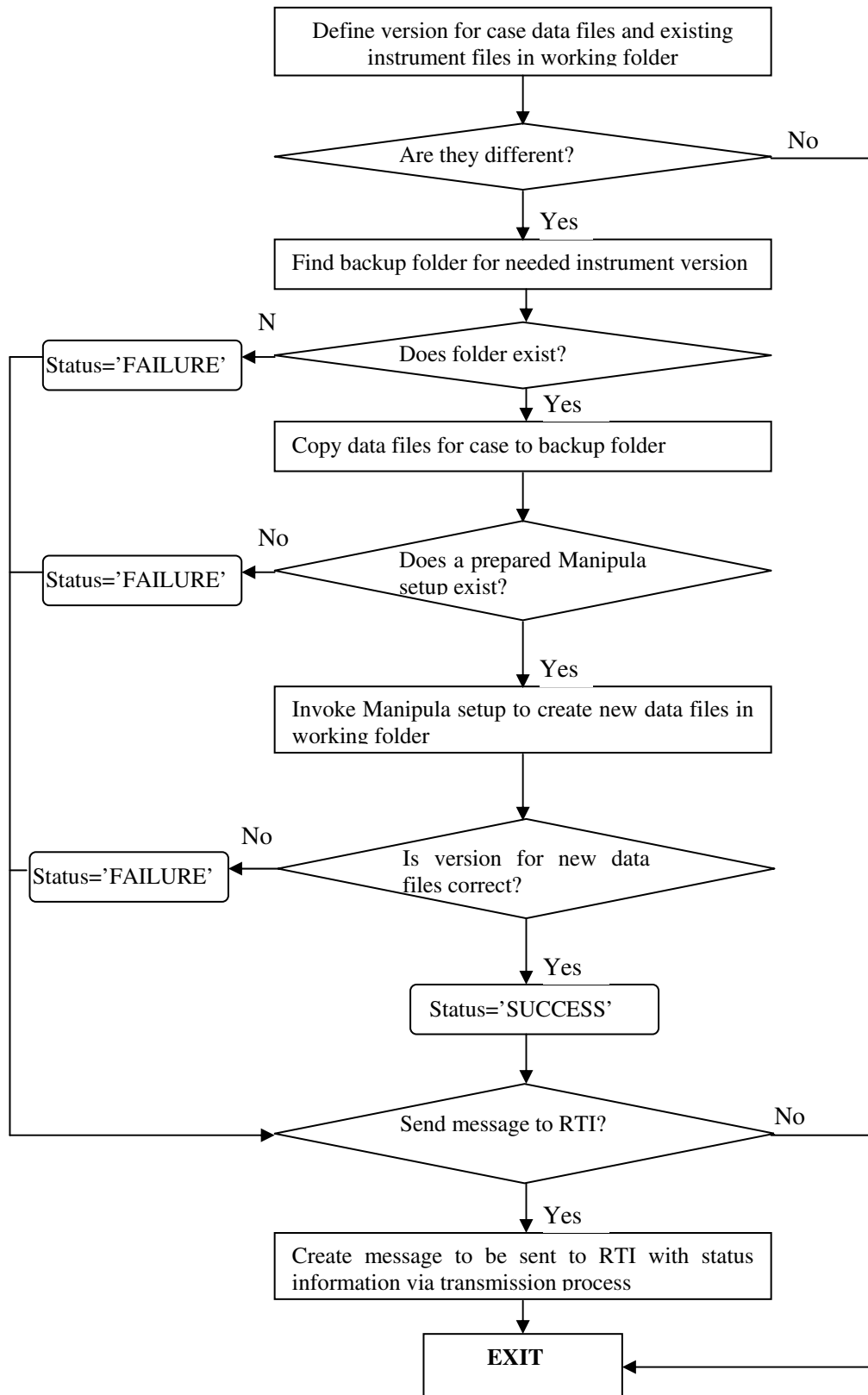


## Appendix A.

Flowchart 1 - Upgrade of Master database



## Flowchart 2 - Upgrade for individual case



### Flowchart 3 - Upgrade database on field laptops

