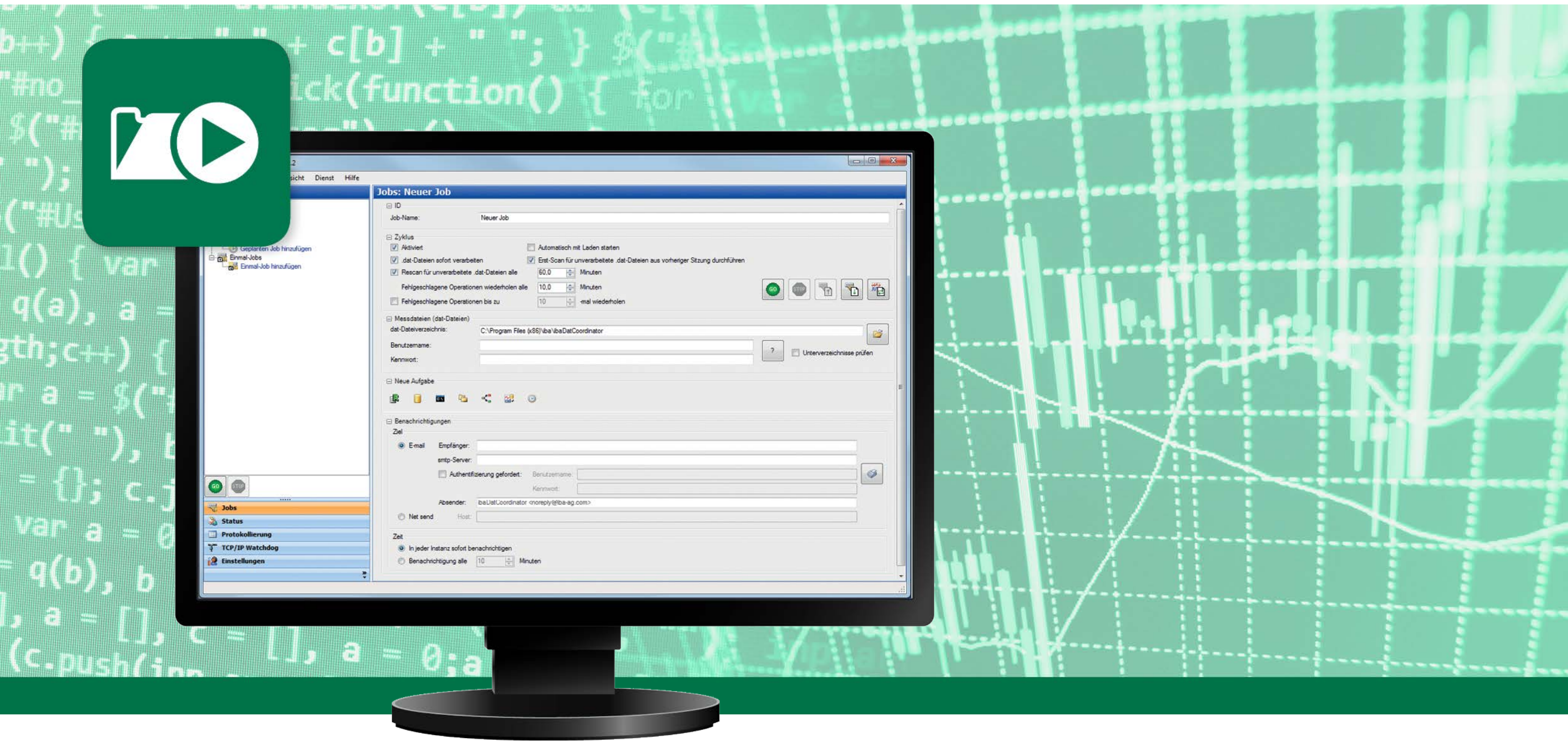
See the Big Picture



Jobs: Neuer Job

ID

Job-Name: Neuer Job

Zyklus

☑ Aktiviert

☐ Automatisch mit Laden starten

☑ .dat-Dateien sofort verarbeiten

☑ Erst-Scan für unverarbeitete .dat-Dateien aus vorheriger Sitzung durchführen

☑ Rescan für unverarbeitete .dat-Dateien alle 60,0 Minuten

Fehlgeschlagene Operationen wiederholen alle 10,0 Minuten

☐ Fehlgeschlagene Operationen bis zu 10 -mal wiederholen

Messdateien (dat-Dateien)

dat-Dateiverzeichnis: C:\Program Files (x86)\iba\ibaDatCoordinator

Benutzername:

Kennwort:

☐ Unterverzeichnisse prüfen

Neue Aufgabe

Benachrichtigungen

Ziel

● E-mail Empfänger:

smtp-Server:

☐ Authentifizierung gefordert: Benutzername: Kennwort:

Absender: ibaDatCoordinator <noreply@iba-ag.com>

○ Net send Host:

Zeit

● In jeder Instanz sofort benachrichtigen

○ Benachrichtigung alle 10 Minuten

ibaDatCoordinator

Automated Analysis with Multiple Data Files

Technical note

Edition 2.0

Measurement Systems for Industry and Energy

Manufacturer

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Contacts

Main office	+49 911 97282-0
Fax	+49 911 97282-33
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out, and we do not provide guarantee for complete conformity. However, the information furnished in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.

The current version is available for download on our web site www.iba-ag.com.

Version	Date	Revision - Chapter / Page	Author	Version SW
2.0	09-2018	Revised issue, sample scripts	RM	2.0

Content

1 About this document

This document describes the configuration of special jobs and tasks in the software *ibaDatCoordinator*. This document provides supplementary information in addition to the standard manual of *ibaDatCoordinator*.

1.1 Target group and previous knowledge

This documentation especially addresses persons who are responsible for analyzing measurement and process data. As the data are provided and processed using other iba products, the following prior knowledge is required or helpful for the described tasks:

- Windows operating system
- *ibaPDA* (collection and structure of data files).
- *ibaAnalyzer* (creating analysis files)
- Configuration of *ibaDatCoordinator*
- Script programming (for using script tasks)

1.2 Notations

In this manual, the following notations are used:

Action	Notation
Menu command	Menu *Logic diagram*
Calling the menu command	*Step 1 – Step 2 – Step 3 – Step x* Example: Select the menu *Logic diagram - Add - New function block*.
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
File names, paths	"Filename", "Path" Example: "Test.doc"

1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

Danger!

The non-observance of this safety information may result in an imminent risk of death or severe injury:

- Observe the specified measures.

Warning!

The non-observance of this safety information may result in a potential risk of death or severe injury!

- Observe the specified measures.

Caution!

The non-observance of this safety information may result in a potential risk of injury or material damage!

- Observe the specified measures

Note

A note specifies special requirements or actions to be observed.

Tip

Tip or example as a helpful note or insider tip to make the work a little bit easier.

Other documentation

Reference to additional documentation or further reading.

2 Definition of task

The *ibaDatCoordinator* program is generally only capable of forwarding one single analysis and one single .dat file per launch of *ibaAnalyzer*. If in an analysis more than one .dat files are used, e.g. by several *ibaPDA* systems belonging to different areas of a plant running in parallel, these files cannot be processed just like that.

3 Approach for the solution

For avoiding this restriction of *ibaDatCoordinator*, you have access to a so called ExtractScript task. By means of this task, the *ibaDatCoordinator* program can execute any scripts that are available as *.bat, *.vbs or *.js. The following examples show the solution by means of "Windows Scripting Host" files. The application *ibaAnalyzer* can call these files with a greater number of .dat files in the right sequence.

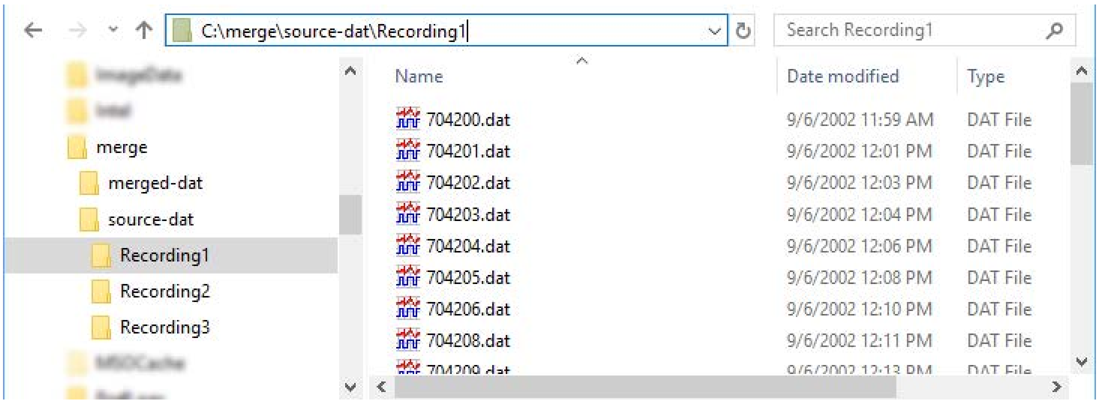
The criteria for merging matching data files may be different, depending on the the application case.

Criterium	Application
Date and time	Temporal precise comparison of simultaneously created data files on synchronized *ibaPDA* systems, e.g. In energy distribution plants
Product-ID/ Number	Collecting measured values from different production plants, passed by the same product one after the other, which have their own *ibaPDA* systems.
Event / trigger	Comparison of measured values (time- and product-independent) at a certain process event (triggered data files with pre- and post-trigger time)

Table 1: Matching criteria and application cases

4 Prerequisites

This approach works because the data files can be assigned clearly either by Technostring information or by time stamp. The structure in the file system looks like the structure shown in the following figure:

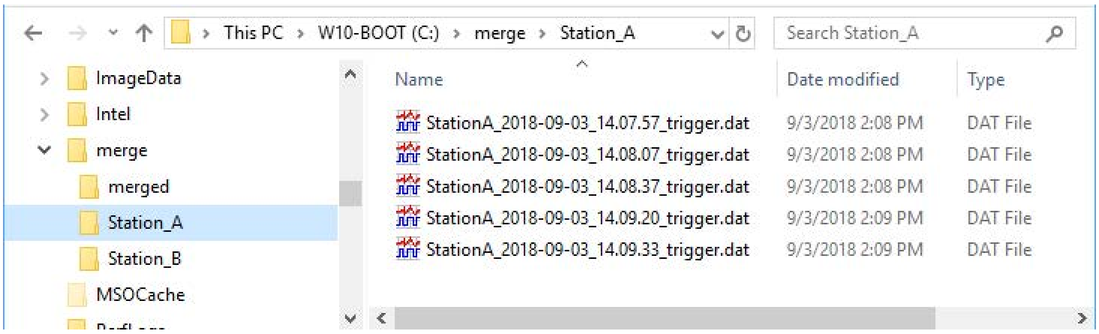


Name	Date modified	Type
704200.dat	9/6/2002 11:59 AM	DAT File
704201.dat	9/6/2002 12:01 PM	DAT File
704202.dat	9/6/2002 12:03 PM	DAT File
704203.dat	9/6/2002 12:04 PM	DAT File
704204.dat	9/6/2002 12:06 PM	DAT File
704205.dat	9/6/2002 12:08 PM	DAT File
704206.dat	9/6/2002 12:10 PM	DAT File
704208.dat	9/6/2002 12:11 PM	DAT File

Fig. 1: Example for file system of data files for product-oriented analysis

The folders "Recording1", "Recording2" and "Recording3" correspond to the different *ibaPDA* systems or production plants respectively and contain the corresponding files "*Product number*.dat" (1 data file per product). The product numbers in the filenames are used for the assignment of the data files from different stations later.

For data files, whose measured values should be compared precisely along the time axis, the assignment can be done with consideration of date/time.



Name	Date modified	Type
StationA_2018-09-03_14.07.57_trigger.dat	9/3/2018 2:08 PM	DAT File
StationA_2018-09-03_14.08.07_trigger.dat	9/3/2018 2:08 PM	DAT File
StationA_2018-09-03_14.08.37_trigger.dat	9/3/2018 2:08 PM	DAT File
StationA_2018-09-03_14.09.20_trigger.dat	9/3/2018 2:09 PM	DAT File
StationA_2018-09-03_14.09.33_trigger.dat	9/3/2018 2:09 PM	DAT File

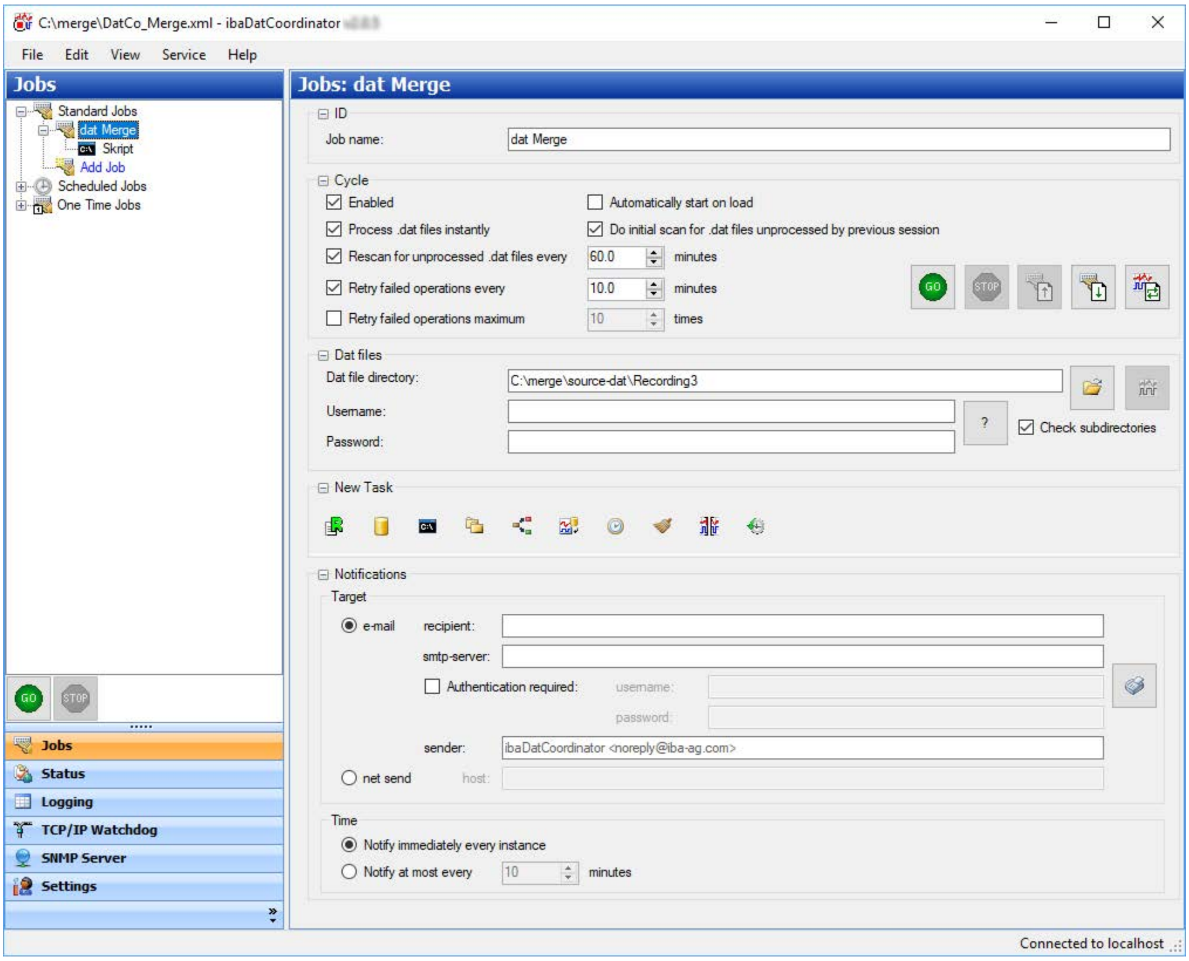
Fig. 2: Example for file system of data files for time-oriented analysis

Note

There may occur problems if umlauts or special characters are used in filenames. Therefore, name the files without blanks and special characters.

5 Configuration

The application *ibaDatCoordinator* has to be configured for a script task.



C:\merge\DatCo_Merge.xml - ibaDatCoordinator

Jobs: dat Merge

ID

Job name: dat Merge

Cycle

☑ Enabled

☐ Automatically start on load

☑ Process .dat files instantly

☑ Do initial scan for .dat files unprocessed by previous session

☑ Rescan for unprocessed .dat files every 60.0 minutes

☑ Retry failed operations every 10.0 minutes

☐ Retry failed operations maximum 10 times

Dat files

Dat file directory: C:\merge\source-dat\Recording3

Username:

Password:

☑ Check subdirectories

New Task

Notifications

Target

● e-mail recipient:

smtp-server:

☐ Authentication required: username: password:

sender: ibaDatCoordinator <noreply@iba-ag.com>

○ net send host:

Time

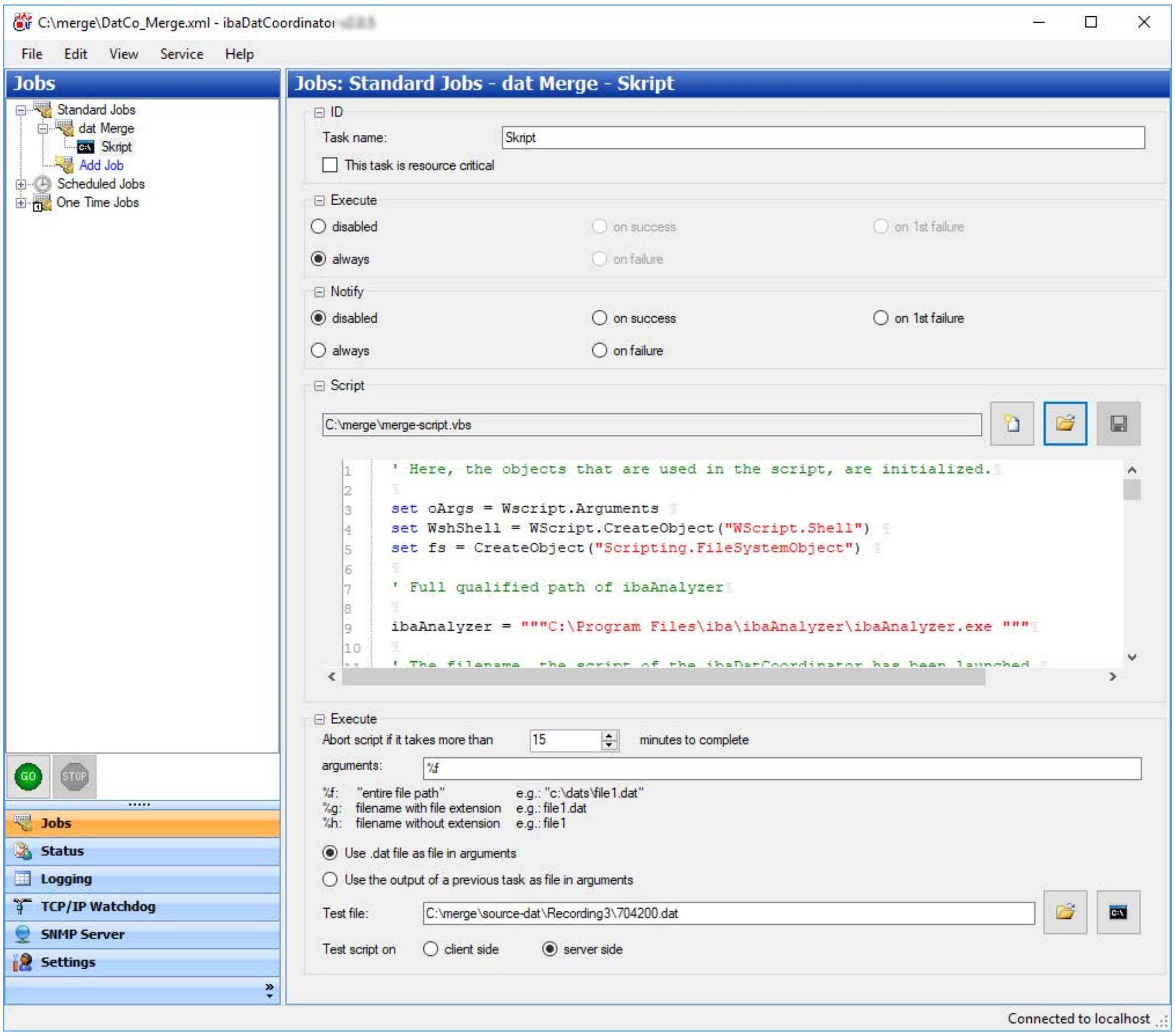
● Notify immediately every instance

○ Notify at most every 10 minutes

Connected to localhost

Fig. 3: Screenshot of the project configuration for the job “dat.Merge”

As shown in Fig. 3, the project in *ibaDatCoordinator* is adapted to the specific requirements. As Dat file directory, you specify the directory of the .dat files whose plant section is processed last (in time). This is how it is guaranteed that the files of the other recordings have been processed as well. If we have e.g. a look at a hot strip mill, where data are recorded on the roughing mill, finishing line and the coiler, you would have to define the directory with the coiler´s data as Dat file directory as it is the last process in the row.



Jobs: Standard Jobs - dat Merge - Skript

ID

Task name: Skript

☐ This task is resource critical

Execute

○ disabled ○ on success ○ on 1st failure

● always ○ on failure

Notify

● disabled ○ on success ○ on 1st failure

○ always ○ on failure

Script

C:\merge\merge-script.vbs

```
' Here, the objects that are used in the script, are initialized.

set oArgs = Wscript.Arguments
set WshShell = WScript.CreateObject("WScript.Shell")
set fs = CreateObject("Scripting.FileSystemObject")

' Full qualified path of ibaAnalyzer

ibaAnalyzer = """C:\Program Files\iba\ibaAnalyzer\ibaAnalyzer.exe """

' The filename, the script of the ibaDatCoordinator has been launched
```

Execute

Abort script if it takes more than 15 minutes to complete

arguments: %f

%f: "entire file path" e.g.: "c:\dats\file1.dat"
%g: filename with file extension e.g.: file1.dat
%h: filename without extension e.g.: file1

● Use .dat file as file in arguments

○ Use the output of a previous task as file in arguments

Test file: C:\merge\source-dat\Recording3\704200.dat

Test script on ○ client side ● server side

Fig. 4: Configuration of the script task

In Fig. 4, you can see the required settings for the script task. In the „Script” field, you can either directly program a script or edit an existing script. You can also upload an existing file to the project by clicking on the “Open file” button. In the “Execute” field, fill in “%f” as argument. This way, the script for each .dat file is launched with the complete path of the file.

6 Application examples

In this chapter you will find several examples for different applications.

Tip

The sample scripts and sample analysis files as well as some testing data of the examples can be found on the DVD "iba Software & Manuals", which is part of each software delivery.

6.1 Product-oriented analysis in a hot rolling mill

Application

The process chain consists of the plants Roughing Mill, Finishing Line and Coiler, each using its own *ibaPDA* system. The data files of the three *ibaPDA* systems are stored in the folders Recording 1, Recording2 and Recording3.

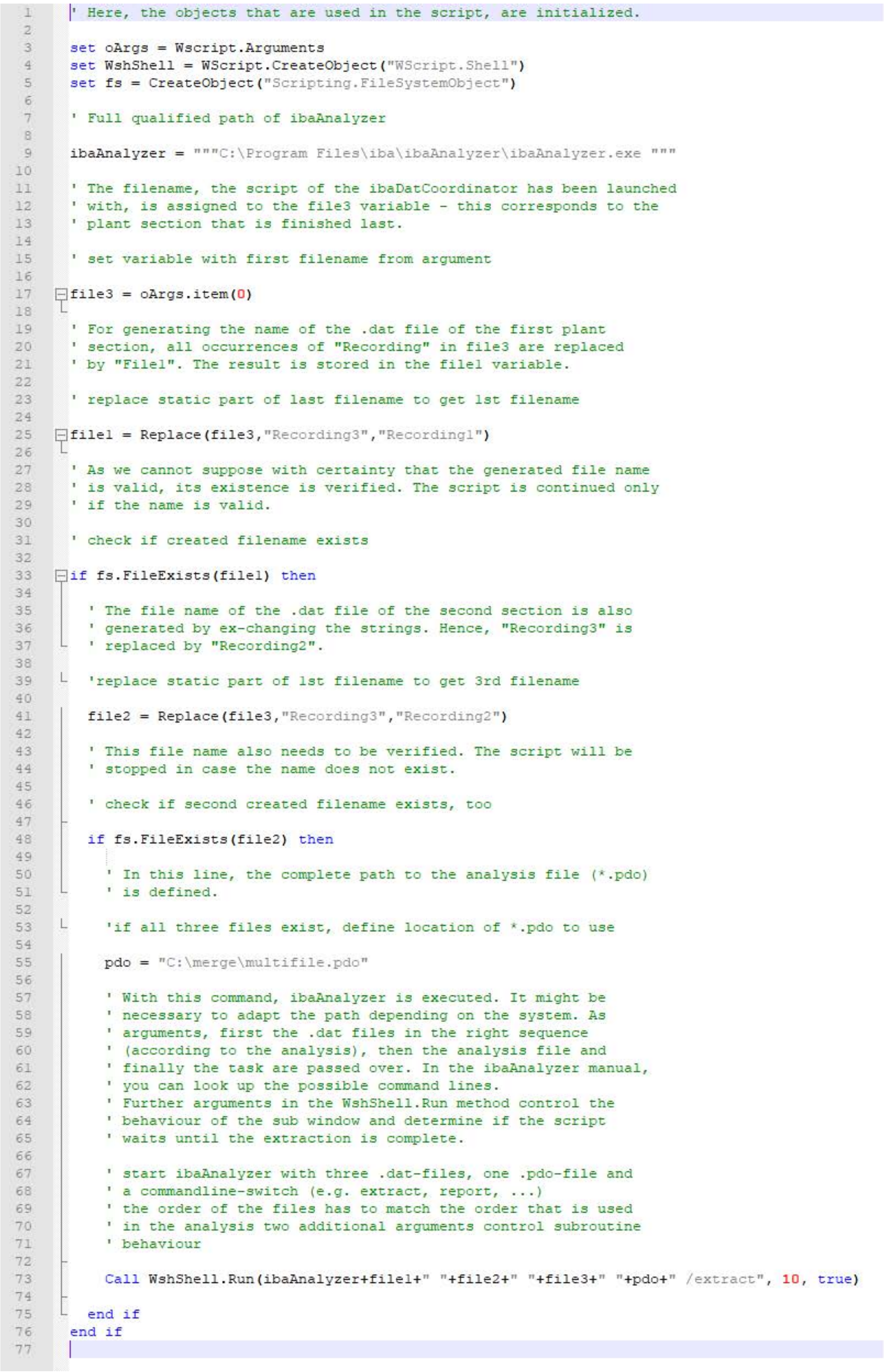
The product ID (coil number) remains the same throughout the entire process chain. The material tracking system provides for transmission of the correct product ID to each *ibaPDA* system, enabling each *ibaPDA* system to write the product ID into the data filenames.

The purpose of this application is to merge the measured values of the entire process chain into one new data file. Therefore, all data files belonging to one product will be loaded into *ibaAnalyzer* together with the analysis file "multifile.pdo" and extracted into one new data file. In order to avoid duplicated module numbers in the resulting data file, a module offset is applied during extraction.

Sample data on DVD

04_Libraries_and_Examples	Basic folder
___110_ibaDatCoordinator-Multifile	ibaDatCoordinator Multifile application samples
______01_Merge_HotRollingMill	Application hot rolling mill
_________merge	**Copy folder to c:\ for testing**
____________merged-dat	Folder for storing the resulting files
____________source-dat	Source data folder
_______________Recording1	Data files from roughing mill
_______________Recording2	Data files from finishing line
_______________Recording3	Data filed from coiler
____________DatCo_Merge.xml	ibaDatCoordinator configuration file
____________merge-script.vbs	VB script
____________multifile.pdo	Analysis file for loading and extracting

Script



```
' Here, the objects that are used in the script, are initialized.

set oArgs = Wscript.Arguments
set WshShell = WScript.CreateObject("WScript.Shell")
set fs = CreateObject("Scripting.FileSystemObject")

' Full qualified path of ibaAnalyzer

ibaAnalyzer = """C:\Program Files\iba\ibaAnalyzer\ibaAnalyzer.exe """

' The filename, the script of the ibaDatCoordinator has been launched
' with, is assigned to the file3 variable - this corresponds to the
' plant section that is finished last.

' set variable with first filename from argument

file3 = oArgs.item(0)

' For generating the name of the .dat file of the first plant
' section, all occurrences of "Recording" in file3 are replaced
' by "File1". The result is stored in the file1 variable.

' replace static part of last filename to get 1st filename

file1 = Replace(file3,"Recording3","Recording1")

' As we cannot suppose with certainty that the generated file name
' is valid, its existence is verified. The script is continued only
' if the name is valid.

' check if created filename exists

if fs.FileExists(file1) then

 ' The file name of the .dat file of the second section is also
 ' generated by ex-changing the strings. Hence, "Recording3" is
 ' replaced by "Recording2".

 'replace static part of 1st filename to get 3rd filename

 file2 = Replace(file3,"Recording3","Recording2")

 ' This file name also needs to be verified. The script will be
 ' stopped in case the name does not exist.

 ' check if second created filename exists, too

 if fs.FileExists(file2) then

  ' In this line, the complete path to the analysis file (*.pdo)
  ' is defined.

  'if all three files exist, define location of *.pdo to use

  pdo = "C:\merge\multifile.pdo"

  ' With this command, ibaAnalyzer is executed. It might be
  ' necessary to adapt the path depending on the system. As
  ' arguments, first the .dat files in the right sequence
  ' (according to the analysis), then the analysis file and
  ' finally the task are passed over. In the ibaAnalyzer manual,
  ' you can look up the possible command lines.
  ' Further arguments in the WshShell.Run method control the
  ' behaviour of the sub window and determine if the script
  ' waits until the extraction is complete.

  ' start ibaAnalyzer with three .dat-files, one .pdo-file and
  ' a commandline-switch (e.g. extract, report, ...)
  ' the order of the files has to match the order that is used
  ' in the analysis two additional arguments control subroutine
  ' behaviour

  Call WshShell.Run(ibaAnalyzer+file1+" "+file2+" "+file3+" "+pdo+" /extract", 10, true)

 end if
end if
```

Fig. 5: Sample script for product-oriented analysis

6.2 Time-oriented analysis in an HVDC plant

Application

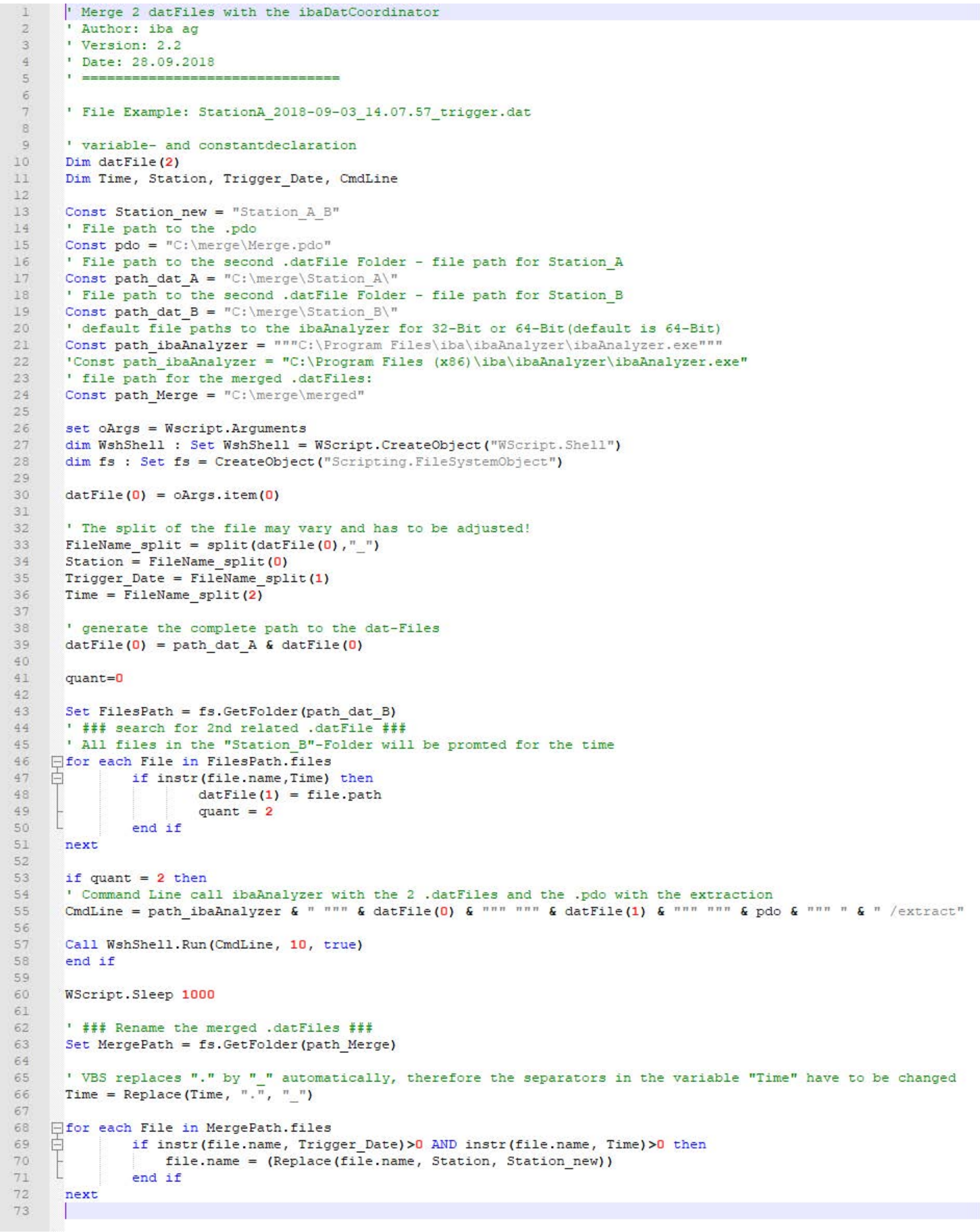
In this case, two data files from different stations, which were created at the same time by triggered recording, should be merged into one file. By that, one is not urged to search for two matching files in different folders.

ibaDatCoordinator is set on the folder “StationA” where the data files of the first station are stored. The script is looking for a matching file with the same date in folder “StationB”. Both data files will be loaded into *ibaAnalyzer* and merged by the analysis “Merge.pdo”. Then, the new file will be renamed with a name as set in the script.

Sample data on DVD

04_Libraries_and_Examples	Basic folder
___110_ibaDatCoordinator-Multifile	ibaDatCoordinator Multifile application samples
______02_Merge_Energy	Sample application HVDC plant
_________merge	**Copy folder to c:\ for testing**
____________merged	Folder for storing the resulting files
____________Station_A	Data files station A
____________Station_B	Data files station B
____________Merge.xml	ibaDatCoordinator configuration file
____________merge_dat_V2.2.vbs	VB script
____________Merge.pdo	Analysis file for loading and extracting

Script



```
' Merge 2 datFiles with the ibaDatCoordinator
' Author: iba ag
' Version: 2.2
' Date: 28.09.2018
' ================================

' File Example: StationA_2018-09-03_14.07.57_trigger.dat

' variable- and constantdeclaration
Dim datFile(2)
Dim Time, Station, Trigger_Date, CmdLine

Const Station_new = "Station_A_B"
' File path to the .pdo
Const pdo = "C:\merge\Merge.pdo"
' File path to the second .datFile Folder - file path for Station_A
Const path_dat_A = "C:\merge\Station_A\"
' File path to the second .datFile Folder - file path for Station_B
Const path_dat_B = "C:\merge\Station_B\"
' default file paths to the ibaAnalyzer for 32-Bit or 64-Bit(default is 64-Bit)
Const path_ibaAnalyzer = """C:\Program Files\iba\ibaAnalyzer\ibaAnalyzer.exe"""
'Const path_ibaAnalyzer = "C:\Program Files (x86)\iba\ibaAnalyzer\ibaAnalyzer.exe"
' file path for the merged .datFiles:
Const path_Merge = "C:\merge\merged"

set oArgs = Wscript.Arguments
dim WshShell : Set WshShell = WScript.CreateObject("WScript.Shell")
dim fs : Set fs = CreateObject("Scripting.FileSystemObject")

datFile(0) = oArgs.item(0)

' The split of the file may vary and has to be adjusted!
FileName_split = split(datFile(0),"_")
Station = FileName_split(0)
Trigger_Date = FileName_split(1)
Time = FileName_split(2)

' generate the complete path to the dat-Files
datFile(0) = path_dat_A & datFile(0)

quant=0

Set FilesPath = fs.GetFolder(path_dat_B)
' ### search for 2nd related .datFile ###
' All files in the "Station_B"-Folder will be promted for the time
for each File in FilesPath.files
        if instr(file.name,Time) then
                datFile(1) = file.path
                quant = 2
        end if
next

if quant = 2 then
' Command Line call ibaAnalyzer with the 2 .datFiles and the .pdo with the extraction
CmdLine = path_ibaAnalyzer & " """ & datFile(0) & """ """ & datFile(1) & """ """ & pdo & """ " & " /extract"

Call WshShell.Run(CmdLine, 10, true)
end if

WScript.Sleep 1000

' ### Rename the merged .datFiles ###
Set MergePath = fs.GetFolder(path_Merge)

' VBS replaces "." by "_" automatically, therefore the separators in the variable "Time" have to be changed
Time = Replace(Time, ".", "_")

for each File in MergePath.files
        if instr(file.name, Trigger_Date)>0 AND instr(file.name, Time)>0 then
            file.name = (Replace(file.name, Station, Station_new))
        end if
next
```

Fig. 6: Sample script for time-oriented analysis

7 Support and contact

Support

Phone:	+49 911 97282-14
Fax:	+49 911 97282-33
Email:	support@iba-ag.com

Note

If you require support, indicate the serial number (iba-S/N) of the product.

Contact

Head office

iba AG
Koenigswarterstraße 44
90762 Fuerth
Germany

Phone:	+49 911 97282-0
Fax:	+49 911 97282-33
Email:	iba@iba-ag.com
Contact:	Harald Opel

Regional and Worldwide

For contact data of your regional iba office or representative please refer to our web site

www.iba-ag.com.