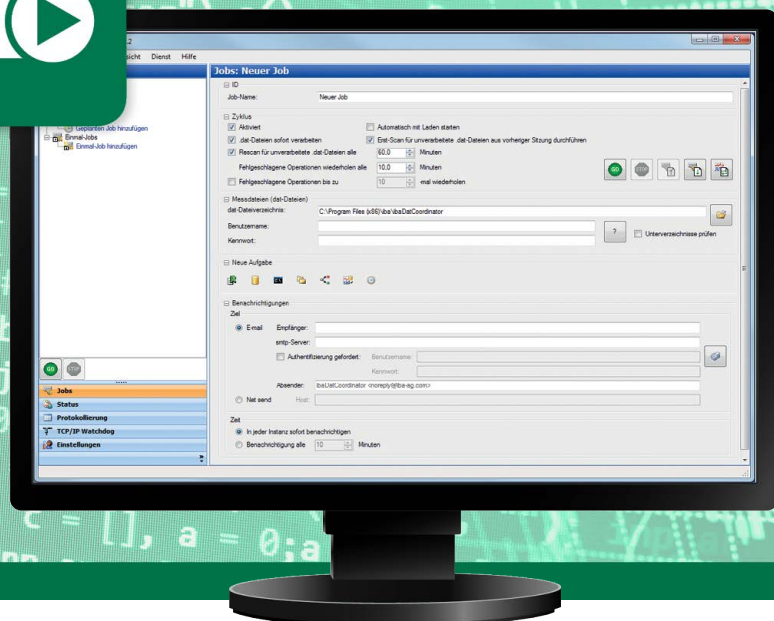




Das Ganze sehen



# ibaDatCoordinator

Automatisierte Analyse mit mehreren Messdateien

Technische Notiz

Ausgabe 2.0

**Messsysteme für  
Industrie und Energie**

---

## Hersteller

iba AG  
Königswarterstr. 44  
90762 Fürth  
Deutschland

## Kontakte

Zentrale	+49 911 97282-0
Telefax	+49 911 97282-33
Support	+49 911 97282-14
Technik	+49 911 97282-13
E-Mail	iba@iba-ag.com
Web	www.iba-ag.com

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz.

© iba AG 2018, alle Rechte vorbehalten.

Der Inhalt dieser Druckschrift wurde auf Übereinstimmung mit der beschriebenen Hard- und Software überprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Garantie übernommen werden kann. Die Angaben in dieser Druckschrift werden jedoch regelmäßig aktualisiert. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten oder können über das Internet heruntergeladen werden.

Die aktuelle Version liegt auf unserer Website [www.iba-ag.com](http://www.iba-ag.com) zum Download bereit.

Version	Datum	Revision - Kapitel / Seite	Autor	Version SW
2.0	09-2018	Überarbeitete Ausgabe, Musterskripte	RM	2.0

Windows® ist eine Marke und eingetragenes Warenzeichen der Microsoft Corporation. Andere in diesem Handbuch erwähnte Produkt- und Firmennamen können Marken oder Handelsnamen der jeweiligen Eigentümer sein.

## Inhalt

<b>1</b>	<b>Zu diesem Dokument .....</b>	<b>4</b>
1.1	Zielgruppe und Vorkenntnisse .....	4
1.2	Schreibweisen .....	4
1.3	Verwendete Symbole .....	5
<b>2</b>	<b>Problemstellung .....</b>	<b>6</b>
<b>3</b>	<b>Lösungsansatz .....</b>	<b>7</b>
<b>4</b>	<b>Voraussetzungen .....</b>	<b>8</b>
<b>5</b>	<b>Konfiguration .....</b>	<b>9</b>
<b>6</b>	<b>Anwendungsbeispiele .....</b>	<b>11</b>
6.1	Produktorientierte Auswertung im Warmwalzwerk .....	11
6.2	Zeitorientierte Auswertung an einer HGÜ-Anlage .....	13
<b>7</b>	<b>Support und Kontakt .....</b>	<b>15</b>

# 1 Zu diesem Dokument

Dieses Dokument beschreibt die Konfiguration spezieller Jobs und Aufgaben im Programm *ibaDatCoordinator*. Dieses Dokument liefert zusätzliche Informationen zum Standardhandbuch von *ibaDatCoordinator*.

## 1.1 Zielgruppe und Vorkenntnisse

Diese Dokumentation wendet sich insbesondere an Personen, die mit der Auswertung von Mess- und Prozessdaten befasst sind. Da die Bereitstellung der Daten sowie deren Weiterverarbeitung mit anderen iba-Produkten erfolgt, sind für die hier beschriebenen Arbeiten folgende Vorkenntnisse erforderlich bzw. hilfreich:

- Betriebssystem Windows
- *ibaPDA* (Entstehung und Struktur der Messdateien)
- *ibaAnalyzer* (Erstellung von Analysevorschriften)
- Konfiguration von *ibaDatCoordinator*
- Skript-Programmierung (für die Nutzung von Skript-Aufgaben)

## 1.2 Schreibweisen

In dieser Dokumentation werden folgende Schreibweisen verwendet:

Aktion	Schreibweise
Menübefehle	Menü <i>Funktionsplan</i>
Aufruf von Menübefehlen	<i>Schritt 1 – Schritt 2 – Schritt 3 – Schritt x</i> Beispiel: Wählen Sie Menü <i>Funktionsplan – Hinzufügen – Neuer Funktionsblock</i>
Tastaturtasten	<Tastename> Beispiel: <Alt>; <F1>
Tastaturtasten gleichzeitig drücken	<Tastename> + <Tastename> Beispiel: <Alt> + <Strg>
Grafische Tasten (Buttons)	<Tastename> Beispiel: <OK>; <Abbrechen>
Dateinamen, Pfade	"Dateiname", "Pfad" Beispiel: "Test.doc"

## 1.3 Verwendete Symbole

Wenn in dieser Dokumentation Sicherheitshinweise oder andere Hinweise verwendet werden, dann bedeuten diese:

---

### Gefahr!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die unmittelbare Gefahr des Todes oder der schweren Körperverletzung!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Warnung!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr des Todes oder schwerer Körperverletzung!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Vorsicht!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr der Körperverletzung oder des Sachschadens!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Hinweis



Hinweis, wenn es etwas Besonderes zu beachten gibt, wie z. B. Ausnahmen von der Regel usw.

---

### Tipp



Tipp oder Beispiel als hilfreicher Hinweis oder Griff in die Trickkiste, um sich die Arbeit ein wenig zu erleichtern.

---

### Andere Dokumentation



Verweis auf ergänzende Dokumentation oder weiterführende Literatur.

---

## 2 Problemstellung

Das Programm *ibaDatCoordinator* ist standardmäßig nur in der Lage, pro Aufruf von *ibaAnalyzer* eine Analyse- und eine einzelne Messdatei zu übergeben. Werden aber in einer Analyse mehrere Messdateien verwendet, z. B. von mehreren, parallel laufenden *ibaPDA*-Systemen aus unterschiedlichen Anlagenabschnitten, dann ist eine automatische Verarbeitung durch *ibaDatCoordinator* nicht ohne weiteres möglich.

### 3 Lösungsansatz

Um diese Beschränkung von *ibaDatCoordinator* zu umgehen, wird auf eine so genannte Skript-Aufgabe zurückgegriffen. Diese ermöglicht dem Programm *ibaDatCoordinator* die Ausführung beliebiger Skripte, die als \*.bat, \*.vbs oder \*.js vorliegen. In den folgenden Beispielen wird die Lösung anhand von „Windows Scripting Host“-Dateien demonstriert. Diese kann das Programm *ibaAnalyzer* mit einer größeren Anzahl von Messdateien in der richtigen Reihenfolge aufrufen.

Die Kriterien, nach denen passende Messdateien zusammengeführt werden, können dabei je nach Anwendungsfall unterschiedlich sein.

Kriterium	Anwendung
Datum und Uhrzeit	Zeitlich exakte Gegenüberstellung von gleichzeitig entstandenen Messdateien zeitsynchronisierter <i>ibaPDA</i> -Systeme, z. B. bei Energieversorgungsanlagen
Produkt-ID/lfd. Nummer	Zusammenfassung der Messwerte aus verschiedenen, von einem Produkt nacheinander durchlaufenden Produktionsanlagen, die jeweils ein eigenes <i>ibaPDA</i> -System nutzen.
Ereignis/Trigger	Vergleich der Messwerte (zeit- und produktunabhängig) zu einem bestimmten Prozessereignis (getriggerte Messdateien mit Pre- und Post-Trigger)

Tab. 1: Zuordnungskriterien nach Anwendungsfall

## 4 Voraussetzungen

Dieser Ansatz funktioniert, da sich die Messdateien entweder anhand von Technostring-Informationen oder dem Zeitstempel eindeutig einander zuordnen lassen. Die Struktur im Dateisystem sieht dabei z. B. so aus wie in der folgenden Abbildung:

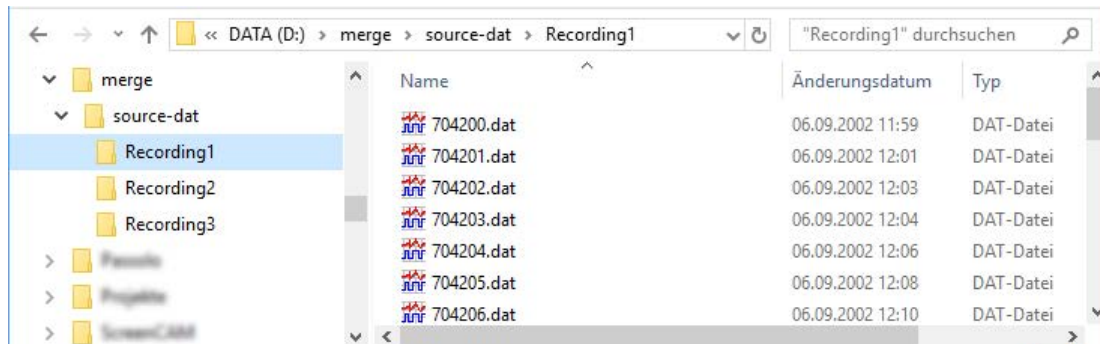


Abb. 1: Beispiel Dateisystem der Messdateien für produktorientierte Auswertung

Die Verzeichnisse "Recording1", "Recording2" und "Recording3" entsprechen den verschiedenen *ibaPDA*-Systemen bzw. Produktionsanlagen und enthalten dementsprechend Dateien "*Produktnummer.dat*" (1 Messdatei pro Produkt). Die Produktnummer im Dateinamen dient hier später der Zuordnung der Messdateien aus den verschiedenen Stationen.

Bei gleichzeitig erzeugten Messdateien, deren Messwerte bezüglich der Zeitachse exakt verglichen werden sollen, kann die Zuordnung anhand von Datum/Uhrzeit erfolgen.

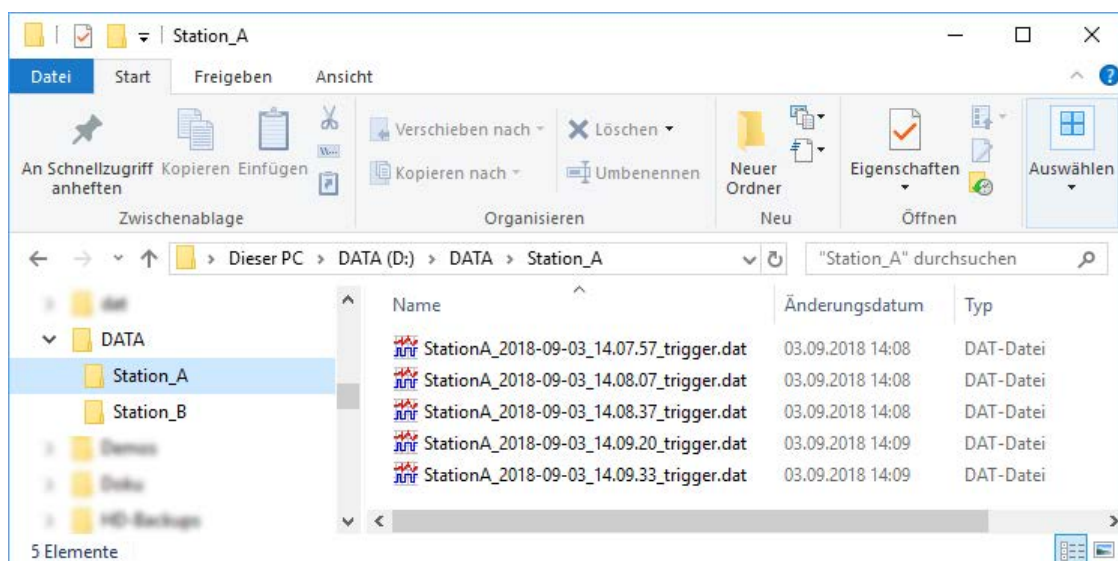


Abb. 2: Beispiel Dateisystem der Messdateien für eine zeitorientierte Auswertung

### Hinweis



Es können Probleme mit Umlauten in Dateinamen auftreten. Die Benennung der Dateien ohne Leer- und Sonderzeichen ist daher zu bevorzugen.



## 5 Konfiguration

Die Anwendung *ibaDatCoordinator* wird für eine Skript-Aufgabe konfiguriert.

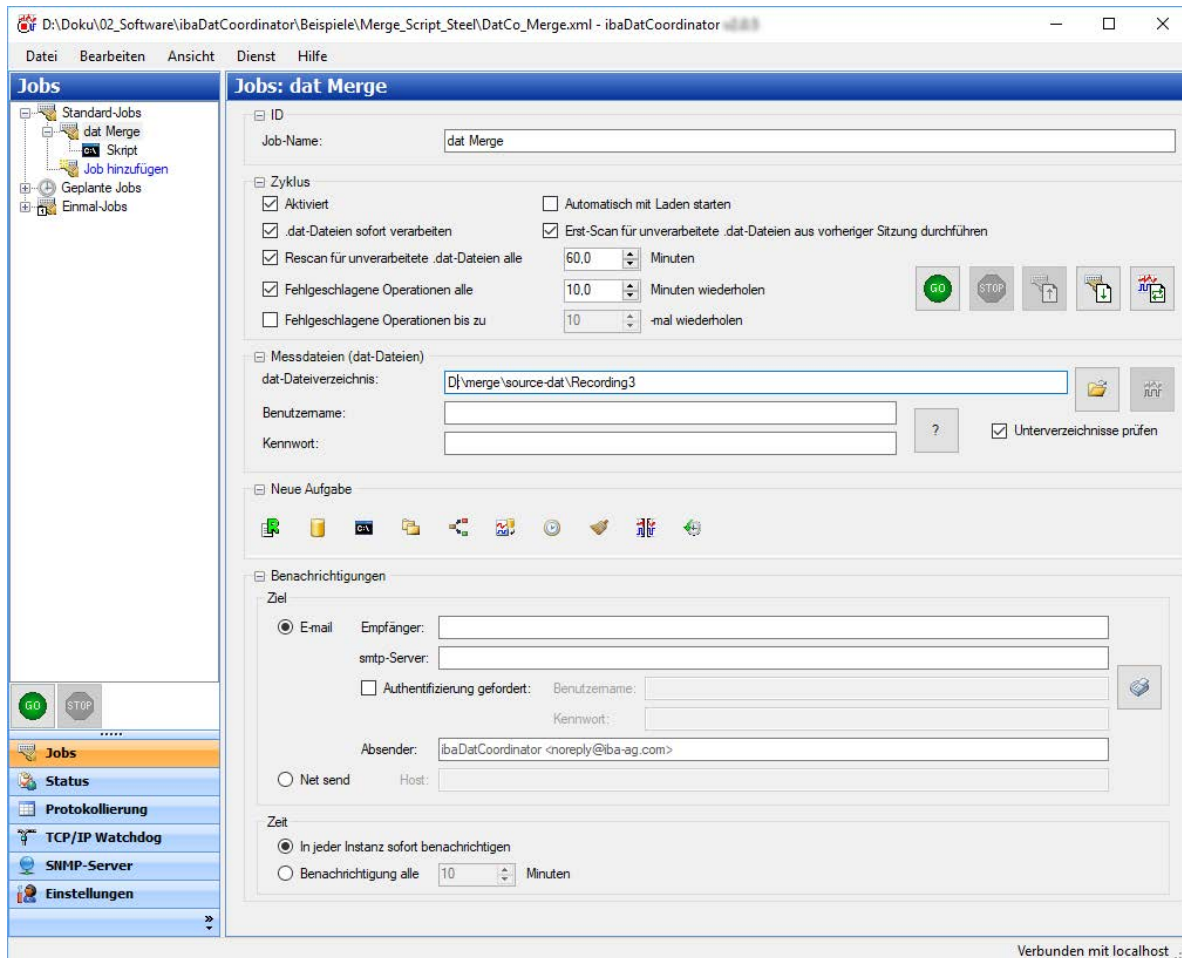


Abb. 3: Ansicht der Projektkonfiguration für den Job "dat.Merge"

Wie in Abbildung 3 dargestellt, wird das Projekt in *ibaDatCoordinator* den gegebenen Anforderungen angepasst. Als Messdateiverzeichnis gibt man das Verzeichnis der Messdateien an, deren Anlagenabschnitt zeitlich am Ende der Prozesskette liegt. Dadurch wird sichergestellt, dass die zugehörigen Dateien der anderen Aufzeichnungen ebenfalls abgeschlossen sind. Für das Beispiel eines Warmbandwerks, bei dem Aufzeichnungen an Vorstraße, Fertigstraße und Haspel durchgeführt werden, wäre das Verzeichnis der Haspeldateien als Messdateiverzeichnis anzugeben.

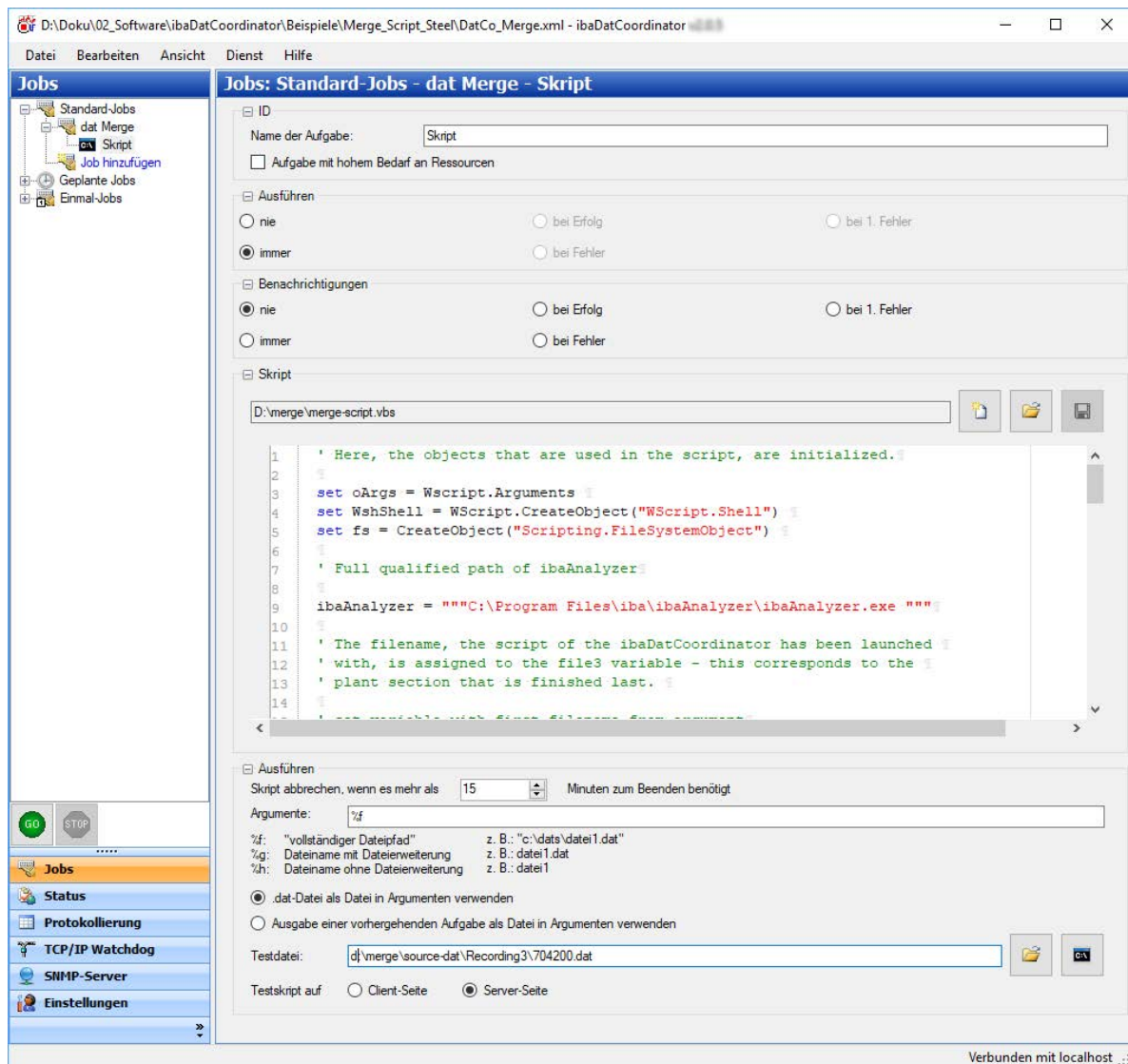


Abb. 4: Konfiguration der Skript-Aufgabe

Die Darstellung in Abbildung 4 zeigt die nötigen Einstellungen für die Skript-Aufgabe. Im Feld „Skript“ kann ein solches direkt erstellt bzw. bearbeitet werden. Alternativ kann man eine bereits vorhandene Datei mit der „Datei öffnen“-Schaltfläche in das Projekt laden. Im Feld „Ausführen“ ist als Argument „%f“ einzutragen, wie im Bild gezeigt. Dadurch wird das Skript für jede Messdatei mit dem vollständigen Pfad der Messdatei aufgerufen.

## 6 Anwendungsbeispiele

In diesem Kapitel finden Sie einige Beispiele für unterschiedliche Anwendungen.

### Tipp



Die Musterskripte und Musteranalysen sowie einige Testdaten der Beispiele finden Sie auf der DVD "iba Software & Manuals", die jeder Softwarelieferung beiliegt.

### 6.1 Produktorientierte Auswertung im Warmwalzwerk

#### Anwendung

Die Prozesskette besteht aus den Anlagen Vorstraße, Fertigstraße und Haspel, die jeweils mit einem eigenen *ibaPDA*-System ausgestattet sind. Die Messdateien der drei *ibaPDA*-Systeme werden in den Verzeichnissen Recording 1, Recording2 und Recording3 abgelegt.

Die Produkt-ID (Warmbandnummer) bleibt in der gesamten Prozesskette gleich. Die Materialverfolgung sorgt dafür, dass die *ibaPDA*-Systeme stets die korrekte Produkt-ID erhalten, so dass jedes *ibaPDA*-System die Produkt-ID in den Messdateinamen schreiben kann.

Ziel der Anwendung ist es, für jedes Produkt die Messwerte der gesamten Prozesskette in einer neuen Messdatei zu vereinen. Dazu werden die zu einem Produkt gehörenden Messdateien zusammen mit der Analysevorschrift "multifile.pdo" in ibaAnalyzer geladen und in eine neue Messdatei extrahiert. Um doppelte Modulnummern in der resultierenden Datei zu vermeiden, wird bei der Extraktion ein Moduloffset hinzugefügt.

#### Musterdaten auf DVD

04_Libraries_and_Examples	Basisverzeichnis
___110_ibaDatCoordinator-Multifile	ibaDatCoordinator Multifile Anwendungsbeispiele
_____01_Merge_HotRollingMill	Anwendungsbeispiel Warmwalzwerk
_____merge	<b>Verzeichnis zum Testen nach c:\ kopieren</b>
_____merged-dat	Verzeichnis für Ablage der Ergebnisdateien
_____source-dat	Quelldatenverzeichnis
_____Recording1	Messdateien von Vorstraße
_____Recording2	Messdateien von Fertigstraße
_____Recording3	Messdateien von Haspel
_____DatCo_Merge.xml	Konfigurationsdatei für ibaDatCoordinator
_____merge-script.vbs	VB-Skript
_____multifile.pdo	Analysevorschrift zum Laden und Extrahieren

## Skript

```

1  ' Here, the objects that are used in the script, are initialized.
2
3  set oArgs = Wscript.Arguments
4  set WshShell = WScript.CreateObject("WScript.Shell")
5  set fs = CreateObject("Scripting.FileSystemObject")
6
7  ' Full qualified path of ibaAnalyzer
8
9  ibaAnalyzer = ""C:\Program Files\iba\ibaAnalyzer\ibaAnalyzer.exe ""
10
11 ' The filename, the script of the ibaDatCoordinator has been launched
12 ' with, is assigned to the file3 variable - this corresponds to the
13 ' plant section that is finished last.
14
15 ' set variable with first filename from argument
16
17 file3 = oArgs.item(0)
18
19 ' For generating the name of the .dat file of the first plant
20 ' section, all occurrences of "Recording" in file3 are replaced
21 ' by "File1". The result is stored in the file1 variable.
22
23 ' replace static part of last filename to get 1st filename
24
25 file1 = Replace(file3,"Recording3","Recording1")
26
27 ' As we cannot suppose with certainty that the generated file name
28 ' is valid, its existence is verified. The script is continued only
29 ' if the name is valid.
30
31 ' check if created filename exists
32
33 if fs.FileExists(file1) then
34
35 ' The file name of the .dat file of the second section is also
36 ' generated by ex-changing the strings. Hence, "Recording3" is
37 ' replaced by "Recording2".
38
39 'replace static part of 1st filename to get 3rd filename
40
41 file2 = Replace(file3,"Recording3","Recording2")
42
43 ' This file name also needs to be verified. The script will be
44 ' stopped in case the name does not exist.
45
46 ' check if second created filename exists, too
47
48 if fs.FileExists(file2) then
49
50 ' In this line, the complete path to the analysis file (*.pdo)
51 ' is defined.
52
53 'if all three files exist, define location of *.pdo to use
54
55 pdo = "C:\merge\multifile.pdo"
56
57 ' With this command, ibaAnalyzer is executed. It might be
58 ' necessary to adapt the path depending on the system. As
59 ' arguments, first the .dat files in the right sequence
60 ' (according to the analysis), then the analysis file and
61 ' finally the task are passed over. In the ibaAnalyzer manual,
62 ' you can look up the possible command lines.
63 ' Further arguments in the WshShell.Run method control the
64 ' behaviour of the sub window and determine if the script
65 ' waits until the extraction is complete.
66
67 ' start ibaAnalyzer with three .dat-files, one .pdo-file and
68 ' a commandline-switch (e.g. extract, report, ...)
69 ' the order of the files has to match the order that is used
70 ' in the analysis two additional arguments control subroutine
71 ' behaviour
72
73 Call WshShell.Run(ibaAnalyzer+file1+" "+file2+" "+file3+" "+pdo+" /extract", 10, true)
74
75 end if
76 end if
77

```

Abb. 5: Musterskript für produktorientierte Auswertung

## 6.2 Zeitorientierte Auswertung an einer HGÜ-Anlage

### Anwendung

In diesem Fall sollen zwei Messdateien, die von unterschiedlichen Stationen zu gleichen Zeitpunkten getriggert erstellt wurden, zu einer Datei zusammengeführt werden. Somit muss man zur Auswertung nicht erst zwei zusammengehörige Dateien in unterschiedlichen Ordnern suchen.

*ibaDatCoordinator* wird auf das Verzeichnis "StationA" angesetzt, wo die Messdateien des *iba-PDA*-Systems der ersten Station abgelegt werden. Das Skript sucht dann in Ordner "StationB" nach einer passenden Datei mit dem gleichen Datum. Beide Messdateien werden in *ibaAnalyzer* geladen und mit der Analysevorschrift "Merge.pdo" zusammengeführt. Abschließend wird die neu erstellte Datei mit einem neuen Namen (welcher im Skript gesetzt werden kann) umbenannt.

### Musterdaten auf DVD

04_Libraries_and_Examples	Basisverzeichnis
___110_ibaDatCoordinator-Multifile	ibaDatCoordinator Multifile Anwendungsbeispiele
_____02_Merge_Energy	Anwendungsbeispiel Energie
_____merge	<b>Verzeichnis zum Testen nach c:\ kopieren</b>
_____merged	Verzeichnis für Ablage der Ergebnisdateien
_____Station_A	Messdateien Station A
_____Station_B	Messdateien Station B
_____Merge.xml	Konfigurationsdatei für ibaDatCoordinator
_____merge_dat_V2.2.vbs	VB-Skript
_____Merge.pdo	Analysevorschrift zum Laden und Extrahieren



## Skript

```

1  ' Merge 2 datFiles with the ibaDatCoordinator
2  ' Author: iba ag
3  ' Version: 2.2
4  ' Date: 28.09.2018
5  ' =====
6
7  ' File Example: StationA_2018-09-03_14.07.57_trigger.dat
8
9  ' variable- and constantdeclaration
10 Dim datFile(2)
11 Dim Time, Station, Trigger_Date, CmdLine
12
13 Const Station_new = "Station_A_B"
14 ' File path to the .pdo
15 Const pdo = "C:\merge\Merge.pdo"
16 ' File path to the second .datFile Folder - file path for Station_A
17 Const path_dat_A = "C:\merge\Station_A\"
18 ' File path to the second .datFile Folder - file path for Station_B
19 Const path_dat_B = "C:\merge\Station_B\"
20 ' default file paths to the ibaAnalyzer for 32-Bit or 64-Bit(default is 64-Bit)
21 Const path_ibaAnalyzer = ""C:\Program Files\iba\ibaAnalyzer\ibaAnalyzer.exe""
22 'Const path_ibaAnalyzer = "C:\Program Files (x86)\iba\ibaAnalyzer\ibaAnalyzer.exe"
23 ' file path for the merged .datFiles:
24 Const path_Merge = "C:\merge\merged"
25
26 set oArgs = Wscript.Arguments
27 dim WshShell : Set WshShell = WScript.CreateObject("WScript.Shell")
28 dim fs : Set fs = CreateObject("Scripting.FileSystemObject")
29
30 datFile(0) = oArgs.item(0)
31
32 ' The split of the file may vary and has to be adjusted!
33 FileName_split = split(datFile(0), "_")
34 Station = FileName_split(0)
35 Trigger_Date = FileName_split(1)
36 Time = FileName_split(2)
37
38 ' generate the complete path to the dat-Files
39 datFile(0) = path_dat_A & datFile(0)
40
41 quant=0
42
43 Set FilesPath = fs.GetFolder(path_dat_B)
44 ' ### search for 2nd related .datFile ###
45 ' All files in the "Station_B"-Folder will be prompted for the time
46 for each File in FilesPath.files
47     if instr(file.name,Time) then
48         datFile(1) = file.path
49         quant = 2
50     end if
51 next
52
53 if quant = 2 then
54 ' Command Line call ibaAnalyzer with the 2 .datFiles and the .pdo with the extraction
55 CmdLine = path_ibaAnalyzer & " " & datFile(0) & " " & datFile(1) & " " & pdo & " " & "/extract"
56
57 Call WshShell.Run(CmdLine, 10, true)
58 end if
59
60 WScript.Sleep 1000
61
62 ' ### Rename the merged .datFiles ###
63 Set MergePath = fs.GetFolder(path_Merge)
64
65 ' VBS replaces "." by "_" automatically, therefore the separators in the variable "Time" have to be changed
66 Time = Replace(Time, ".", "_")
67
68 for each File in MergePath.files
69     if instr(file.name, Trigger_Date)>0 AND instr(file.name, Time)>0 then
70         file.name = (Replace(file.name, Station, Station_new))
71     end if
72 next
73

```

Abb. 6: Musterskript für zeitorientierte Auswertung

## 7 Support und Kontakt

### Support

Tel.: +49 911 97282-14  
Fax: +49 911 97282-33  
E-Mail: [support@iba-ag.com](mailto:support@iba-ag.com)

---

### Hinweis



Wenn Sie Support benötigen, dann geben Sie die Seriennummer (iba-S/N) des Produktes an.

---

### Kontakt

#### Zentrale

iba AG  
Königswarterstraße 44  
90762 Fürth  
Deutschland

Tel.: +49 911 97282-0  
Fax: +49 911 97282-33  
E-Mail: [iba@iba-ag.com](mailto:iba@iba-ag.com)  
Kontakt: Harald Opel

#### Regional und weltweit

Weitere Kontaktadressen unserer regionalen Niederlassungen oder Vertretungen finden Sie auf unserer Webseite

**[www.iba-ag.com](http://www.iba-ag.com).**