

Rekursive Datensicherung

Zeigt wie eine Funktion rekursiv aufgerufen werden kann, um z.B. eine Datensicherung durchzuführen.

Code

```
Ausgabe:
##### 1. Vorbereitung
<!-- Zielpfad erstellen falls nicht vorhanden -->
<PathExists Path="{@ZielBackupOrdner}" Variable="{@Existiert}"
<ConditionTrue
<PathCreate Path="{@ZielBackupOrdner}" Variable="{@Existiert}"
</ConditionTrue>

<!-- Sicherheitskopie des Ordners -->
<PathExists Path="{@ZielBackupOrdnerSicherheitskopie}" Variable="{@Existiert}"
<ConditionTrue
<PathCreate Path="{@ZielBackupOrdnerSicherheitskopie}" Variable="{@Existiert}"
</ConditionTrue>

<!-- Sicherheitskopie des letzten Backups erstellen -->
<PathExists Path="{@ZielBackupOrdnerSicherheitskopie}" Variable="{@Existiert}"
<ConditionTrue
<PathCreate Path="{@ZielBackupOrdnerSicherheitskopie}" Variable="{@Existiert}"
</ConditionTrue>

<!-- Variablen fuer den Backup Vorgang initialisieren -->
<Set Variable="{@AktuellesVerzeichnisVollerPfad}" Value="{@QuelleBackupOrdner}"
<Set Variable="{@AktuellesVerzeichnis}" Value="{@QuelleBackupOrdner}"

##### 2. Backup ausfuehren #####
<!-- Alle Dateien und Verzeichnisse rekursiv sichern -->
<CallFunction Name="{@backup}" Variable="{@AktuellesVerzeichnis}"

<!-- Verzeichnis anlegen -->
<PathExists Path="{@ZielBackupOrdnerSicherheitskopie}" Variable="{@Existiert}"
<ConditionTrue
<PathCreate Path="{@ZielBackupOrdnerSicherheitskopie}" Variable="{@Existiert}"
</ConditionTrue>

</ConditionTrue>
</ForEach>
</Function>
</Batch>
```

Alles anzeigen

Das Skript soll den Umgang mit [Funktionen](#) demonstrieren und zeigen wie rekursiv die Ordner und [Dateien](#) aus einem [Dateisystem](#) ausgelesen werden können.

Zur Datensicherung gibt es effizientere Methoden wie die BatchPad Aktion *FileMirror*. Mehr Informationen zu *FileMirror* gibt es unter der Rubrik "Aktionen und [Parameter](#)".