



**PlugIn for DpuScan**

**Index**

**Indexing**

**Supplement to the DpuScan Reference Manual**

## Copyrights

© 1997 - 2006 J&K Imaging, Marietta/USA and Janich & Klass, D-Wuppertal.

All rights reserved. Printed in Germany.

The information contained in this documentation is the property of J&K Imaging, Marietta and Janich & Klass, Wuppertal. Neither receipt nor possession hereof confers or transfers any right to reproduce or disclose any part of the contents hereof, without the prior written consent of J&K Imaging, Marietta and Janich & Klass, Wuppertal.

## Trademarks

The DPU logo is a registered trademark of Janich & Klass, Wuppertal. DpuScan is a trademark of J&K Imaging, Marietta/USA. All other product names and logos are trademarks or registered trademarks of their representative companies.

## Disclaimer

The instructions and descriptions in this manual were accurate at the time of this manual's printing. However, we reserve the right to alter the description and/or the product at anytime without prior notice.

J&K Imaging and Janich & Klass assume no liability for damages incurred directly or indirectly from errors, omissions, or discrepancies between this manual and the product.

## Actuality

It may happen that a more recent version of this manual for DpuScan is available for download from the Internet. Therefore, it is recommended that you should compare the version by means of the date printed on this page with the version on the Internet. You should use the most up-to-date version of the manual.

The Internet version of this annex to the DpuScan Reference Manual is found on the Web at the following address:

<http://www.jkimaging.com/pdf/PlugIns/Index-English.pdf>

© 2006 Janich & Klass Computertechnik GmbH, Wuppertal, Germany

September 13, 2006

## Table of Contents

1	Overview .....	4
1.1	The PlugIn Interface .....	4
1.1.1	Installation and Registration .....	4
1.1.2	Usage in a Class .....	5
1.1.3	PlugIn Configuration (Data Exchange) .....	7
2	Using the Index PlugIn .....	11
2.1	Before you get Started .....	12
2.2	Creating an Indexing Mask .....	13
2.3	Data Transfer and Keyboard Assignment .....	14
2.4	Presetment already during Scanning .....	15
2.5	Positioning When Leafing Through the Images .....	16
2.6	Double Percent Signs for the Output .....	17
3	Index Definition with the Wizard .....	18

PlugIns for DpuScan are expansions for its functional scope and must be licensed separately.

This documentation describes one such additional module for already existing licenses of DpuScan. The use of this PlugIn is possible only in combination with DpuScan. Therefore, this documentation can also be used only together with the documentation for DpuScan.

## **1 Overview**

Indexing is a process where data are manually entered on an image. Mostly, these are hand-written remarks or other image information that can not be captured electronically, at a reasonable price.

In the following text, "Indexing" always means such manual entries.

Janich & Klass offers, for their capture software DpuScan, the PlugIn called JK\_INDEX. It uses the User-Defined Dialogs (= UDD) that can be created in DpuScan, as an entry mask. For doing so, the features to be indexed are defined as user-defined variables, and a user-defined dialog is made in the Class to later serve as index mask.

The decisive difference to the "normal" User Dialogs that are well-known in DpuScan since long, the "embedded" Indexing dialog will constantly remain on the screen, during an Indexing job, and will not only show once, for example at the start of a batch.

### **1.1 The PlugIn Interface**

Since Version 4.10 of DpuScan, external program parts can be queried via an expanded interface. Contrary to the Broker, to Function-DLL and F3-Function-DLL it is possible here to process or to create image data outside of DpuScan.

PlugIns can, on the one hand, be called in the Task during the scan process, targeted for every image (process mode) or, on the other hand, be operated by the user via his / her display in Pause condition (interactive mode).

#### **1.1.1 Installation and Registration**

A PlugIn usually consists of a program library (DLL file) and eventually of further files which are used by this program library.

During installation, all files of the PlugIn are copied into the EXT sub-folder of DpuScan. DpuScan can offer only those PlugIns that exist in this sub-folder.

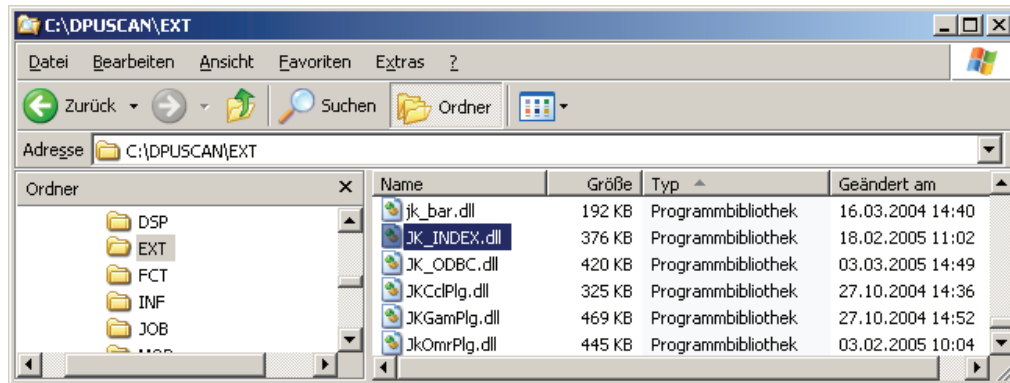


Illustration 1 – Storage Place for PlugIns

In order to use a PlugIn, it must also be *registered*, what means: The user will get a key from J&K that must be entered during the registration procedure:

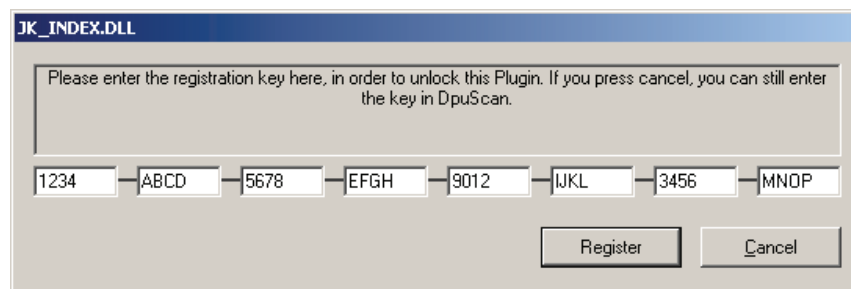


Illustration 2 – Registering a PlugIn

This key can not be transferred to any other station. Different keys exist, also with timely limitation. In this case, the above dialog will also display when the registration has run out.

**With DpuScan QSI, this registration will be void. Here, the Index PlugIn belongs to the scope of supply and will be automatically released by the QSI dongle.**

## 1.1.2 Usage in a Class

Once that all technical and legal conditions are fulfilled, the PlugIn can be selected in DpuScan in its Class Configuration and can be fitted with the necessary parameters. To reach this dialog, click **Class | Configuration | Process | PlugIns**

Use the **Add** button to load the DLL to be used. This dialog will display:

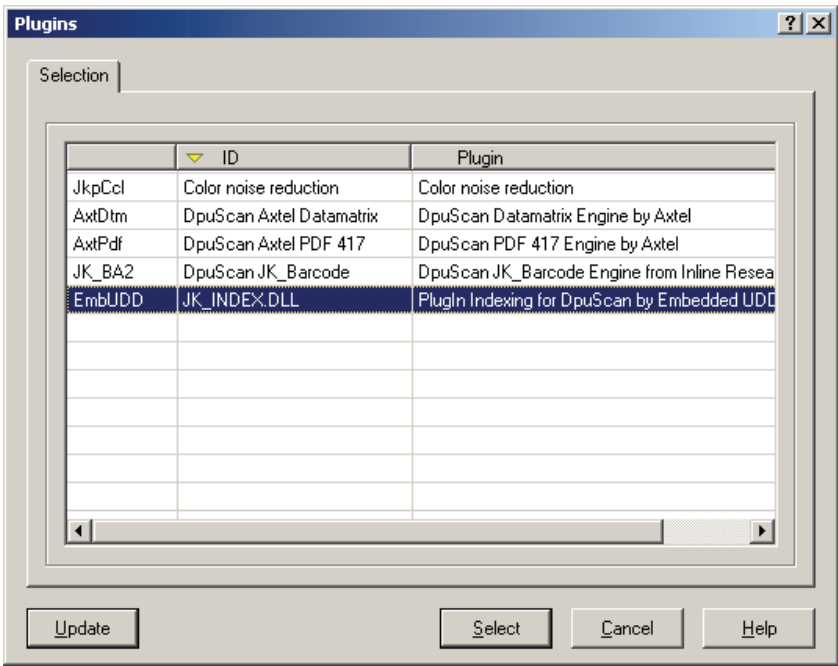


Illustration 3 – Loading the Plugin

When the DLL is loaded for the first time, the dialog for entering the registration key will display, see [Illustration 2 – Registering a Plugin](#) on page 5. Afterwards, the Plugin will appear in the list:

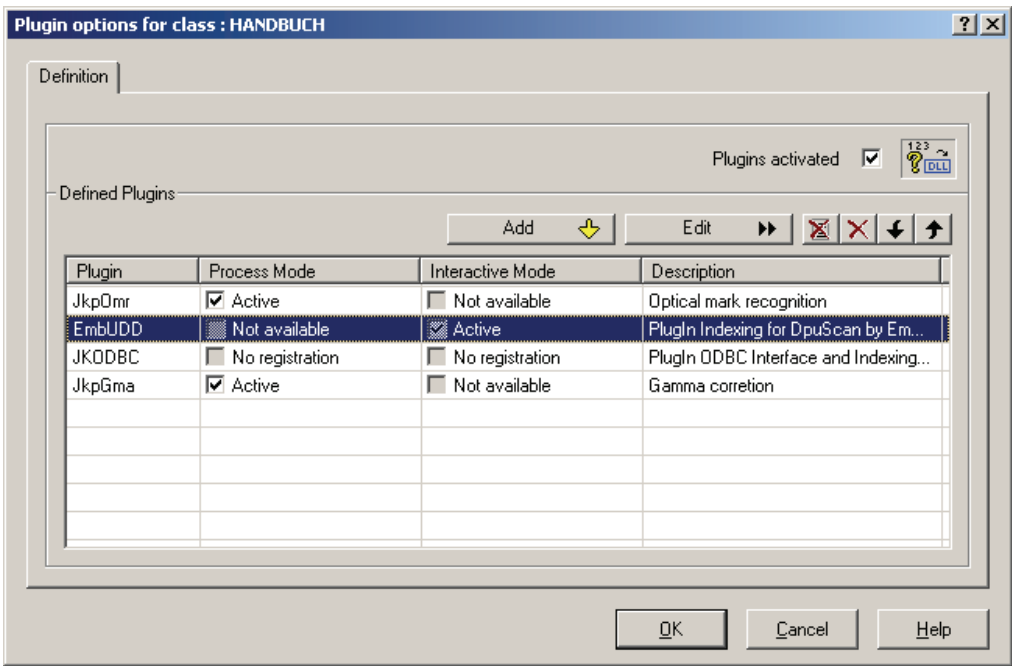


Illustration 4 – Plugins in the Class

While loading the PlugIn, DpuScan and the PlugIn will specify which data shall be transferred as standard and when DpuScan shall put available which parts of the display to the module.

The table displays, in its first column, the PlugIns with their "Shortnames". This shortname will later be used, for example, in the configuration of the display arrangement.

The columns **Process Mode** and **Interactive Mode** indicate whether and how the PlugIn is used in DpuScan.

The **Description** details the name of the DLL where the PlugIn is stored.

The entries in this list can be processed as follows:



Deletes the entire list.



Deletes the selected entry.



Moves the selected entry upwards.



Moves the selected entry downwards.

The **Edit** button opens a dialog for further configuration of the PlugIn interface. Then you can indicate which images and variables are sent to the PlugIn or are returned from it.

## 1.1.3 PlugIn Configuration (Data Exchange)

This dialog rules the data exchange between DpuScan and the PlugIn. You can also click the **External Settings** button to call the dialog for the specific PlugIn.

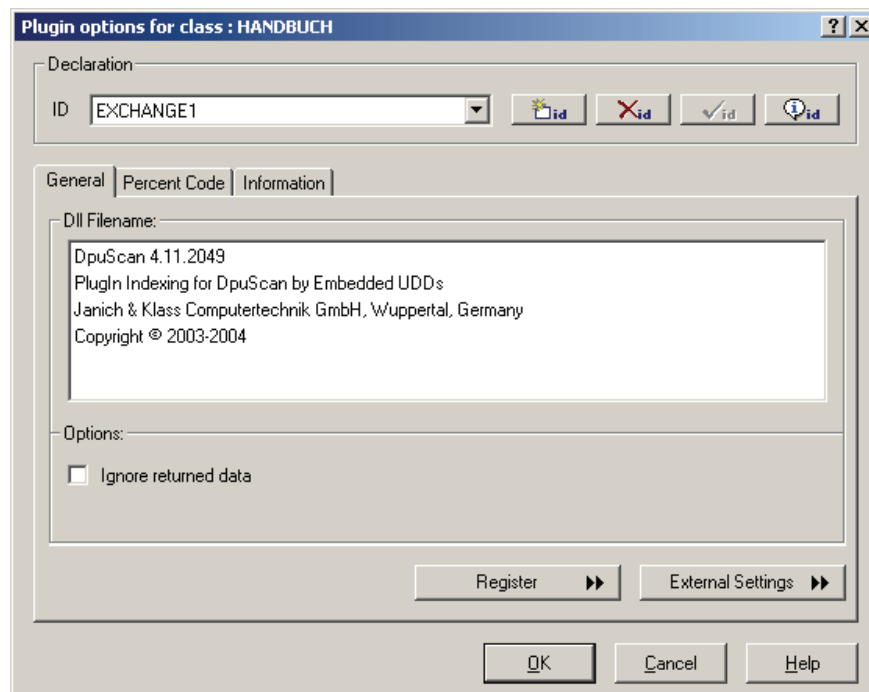


Illustration 5 – PlugIn Configuration, General Page

The PlugIn configuration dialog has the usual button as known in DpuScan for administering configurations:



Selects an existing configuration and arranges it to this Class.



Starts defining a new configuration.



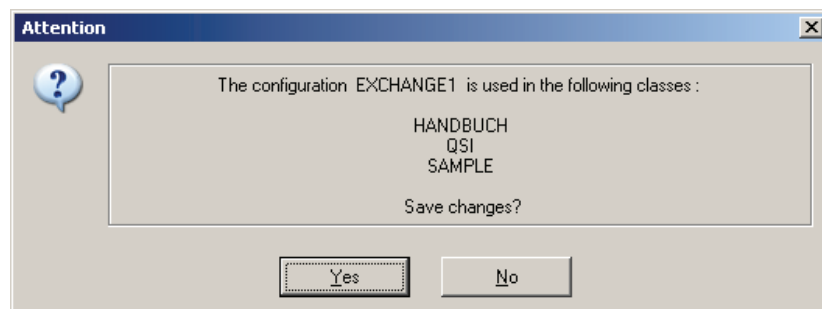
Deletes the actual configuration.



Applies the modifications for the actual configuration.



Opens a dialog that displays in which Classes the actual configuration is also used.



*Illustration 6 – Warning before Saving*

### 1.1.3.1 PlugIn Configuration: General

The dialog for PlugIn configuration on the DpuScan side serves less for setting parameters within DpuScan, but details the information that is exchanged during initialization; so there is only little to set.

On the first page, refer [Illustration 5 – PlugIn Configuration, General Page](#) on page 7, you can generally determine whether the data that the PlugIn returns shall be ignored, or not. This option should, however, only be used for test purposes.

The **Registration** button opens the registration dialog for entering the registration key, refer [Illustration 2 – Registering a PlugIn](#) on page 5.

Click to **External Setting** you reach the dialog that the PlugIn offers to make further settings. This dialog differs from PlugIn to PlugIn and will be described in the following Chapters.

### 1.1.3.2 PlugIn Configuration: Percent Codes

This page shows which variables are sent to the PlugIn and must be returned from there again. This way, for example, the user-defined variable %(I.WIZARD4) can be modified in the Index PlugIn.



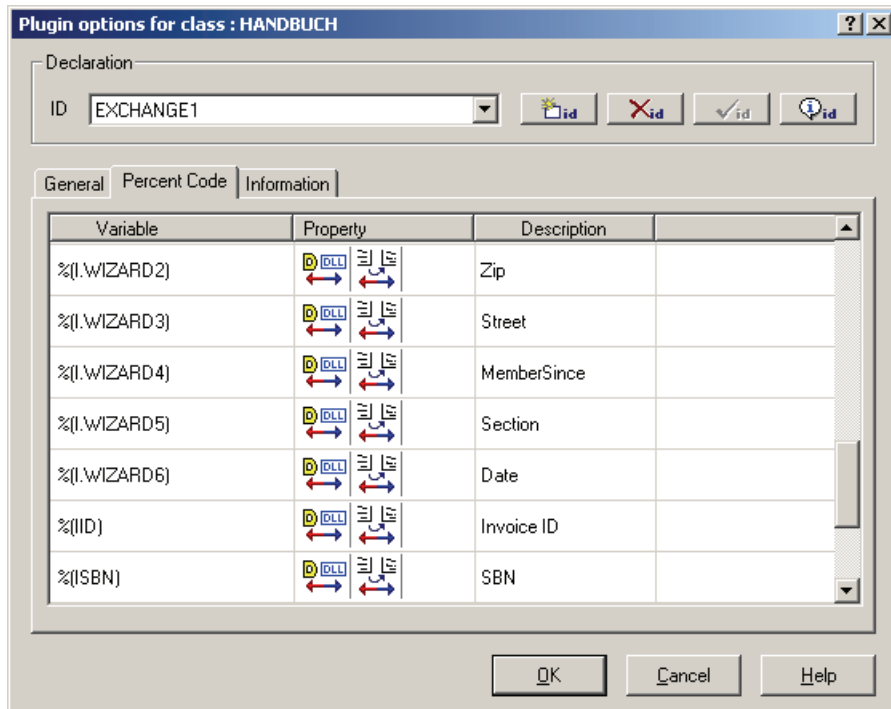


Illustration 7 – Plugin Configuration, Percent Code Tab

While loading the Plugin, DpuScan and the Plugin will specify which data shall be exchanged.

Variables can have these features:



Data are sent from the PlugIn to DpuScan.



DpuScan sends data to the PlugIn.



DpuScan sends data to the PlugIn, the modified data are returned by the PlugIn.



In the **Process Mode**, the data are sent as Task step.



In the **Interactive Mode**, the data are exchanged **before** the new selection (like going to the next image) is displayed.



In the **Interactive Mode**, the data are exchanged **after** the new selection has been executed and the new image is displayed.

Examples:

The PlugIn JK\_INDEX knows all variables that can be set in the Task as Task step, that are set in the Event Rules, and all user-defined variables without a specific validity range

### 1.1.3.3 PlugIn Configuration: Information

Here, a summary of all information about the PlugIn is displayed:

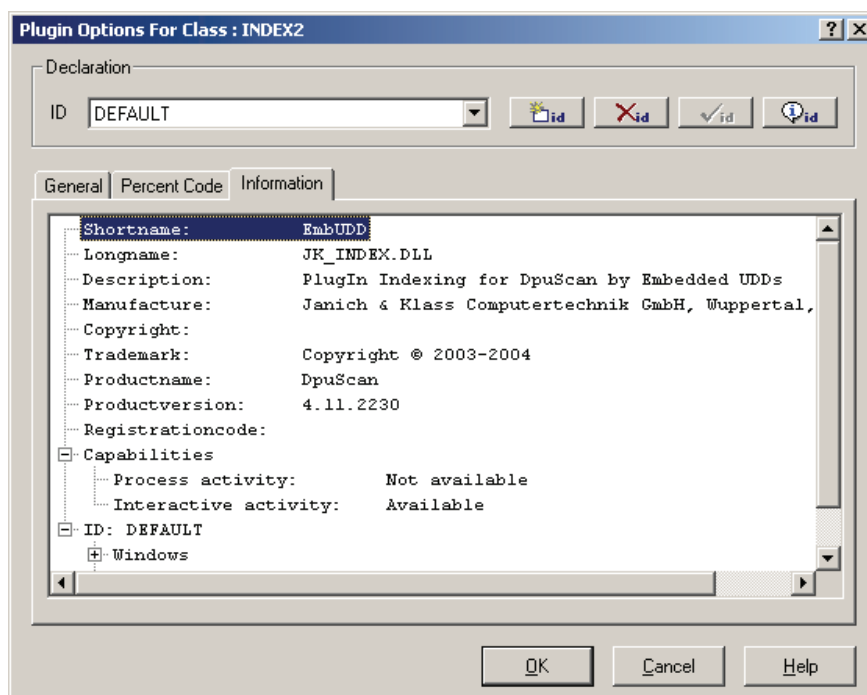


Illustration 8 – PlugIn Configuration: Information

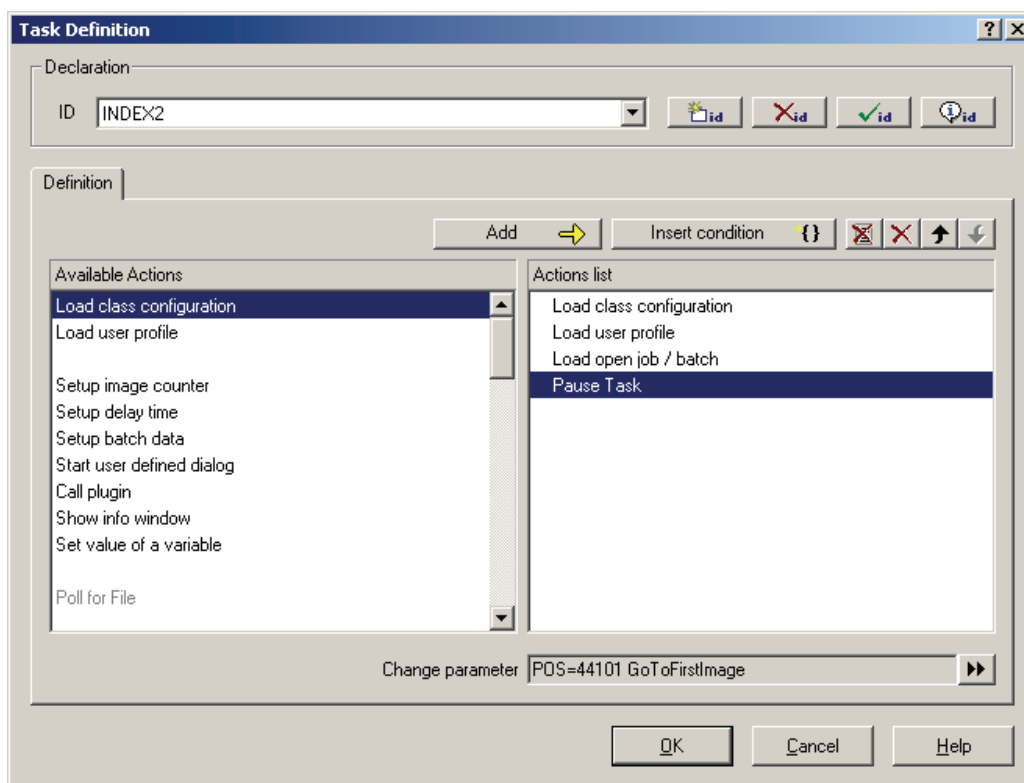
It displays in which configurations the PlugIn will have which windows, variables, or images.

## 2 Using the Index Plugin

The Index Plugin must exclusively be used in the interactive mode and never in the process mode, even if this offered in the Task list, by mistake.

**Attention: Please bear in mind that the Index Plugin will not be loaded in the Class if you have a mere scanning Task.**

The Index Plugin is a good example for using a Plugin in the interactive mode. After scanning, the job is opened again on one, or on several indexing stations in order to manually add missing data. In this case, the Task is accordingly rather short:



*Illustration 9 – Task for Plugin in the Interactive Mode*

The Configuration itself is done in the Class while defining the screen layout. When Plugins that work in the interactive mode are loaded in the Class, the screen layout will offer their exported windows:

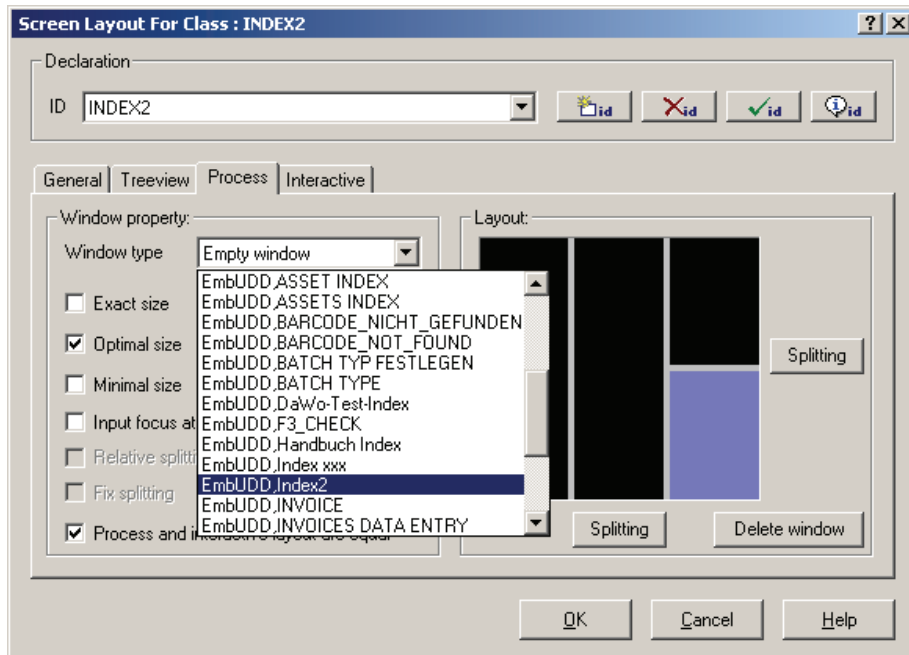


Illustration 10 – Screen Layout for PlugIn in the Interactive Mode

In this example, the selected (blue) sub-window is assigned with the INDEXING window of the EmbUDD PlugIn.

## 2.1 Before you get Started ...

Indexing is usually not done on the scan station but in a later process, on one or several indexing stations. Data are transferred exclusively via Open Jobs.

You should arrange your scan and indexing station in a manner that the completed scan job is exported from the scan station, or imported by the indexing station.

If several stations have access to a common folder, the jobs being under procession by another station will accordingly be blocked.

As opening and closing of a jobs requires a certain amount of time, the individual jobs should not be too large.

**Actually, there is no possibility in DpuScan to divide a large job before its indexing into several smaller jobs and to put them together again, later on.**

But still, it is certainly possible to load images from any different folders from the hard disk, to create Open Jobs from them, and to later finalize them again into one common folder.

Before your start scanning, you should also determine all the features that you intend to import in your database. Later modifications at this backbone mostly cause the need to alter, on different stations, different Classes or Tasks (preferably simultaneously).

If your want to capture features that shall remain the same for several images, like a parts number for all following detail drawings, for example, you should secure that all such images stay in one only Open Job, when you split up the jobs. So: Never split a job in the middle of a document.

## 2.2 Creating an Indexing Mask

A detailed description of the User-Defined Dialogs is found in the according Chapter of the Reference Manual. Please observe the Chapters about spreading **Frames** and the **Validity** of variables. Generally, every UDD might be used as an indexing dialog, but there are some practical restrictions:

**In the Embedded Mode, no standard buttons exist like OK, Cancel, or Help.**

**Index-UDDs should not contain lines of type Button, as pressing such buttons will have no effect while indexing.**

The remaining Plugin configuration is as easy as can be and restricts itself mainly to which UDD shall display.

How to select the dialog that shall display is described in Chapter [2 Using the Index Plugin](#) on page [12](#). All available UDDs are cited in the drop-down list of window types, with the prefix **EmbUDD**, refer [Illustration 10 – Screen Layout for Plugin in the Interactive Mode](#) on page [12](#).

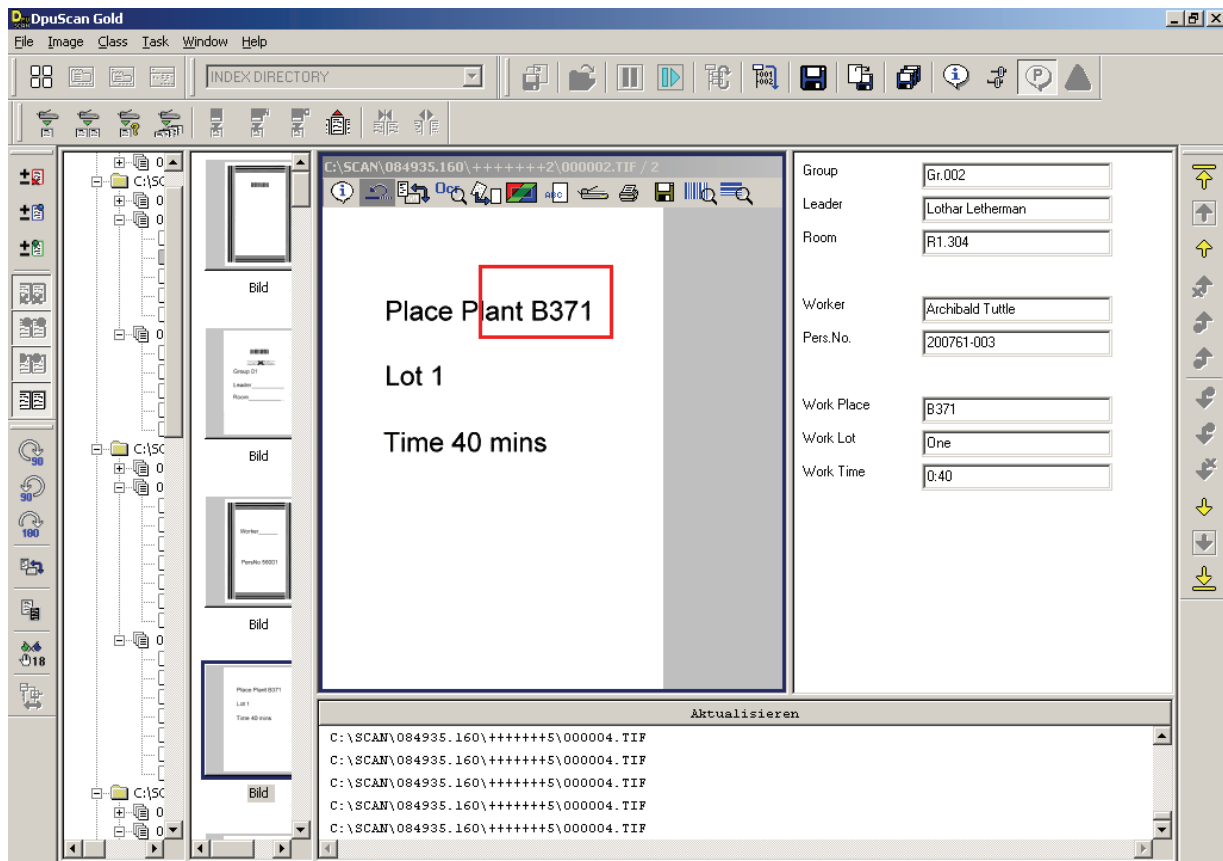


Illustration 11 – JK\_INDEX Embedded Dialog

In the above image, a UDD is displayed in "Embedded Mode" in a sub-window on the right-hand side of the screen.

It is absolutely possible to use several indexing dialogs at one time, for example one to capture only such features that hold for an entire folder, another for capturing features that belong to a multi-image file and one to capture the image features. This way, dialogs can be used also for other demands.

2.3 Data Transfer and Keyboard Assignment

Data are transferred every time when you change from one window to another. Thus, the keyboard should be configured accordingly, for example the keys for PageUp and PageDown must cause the according actions.

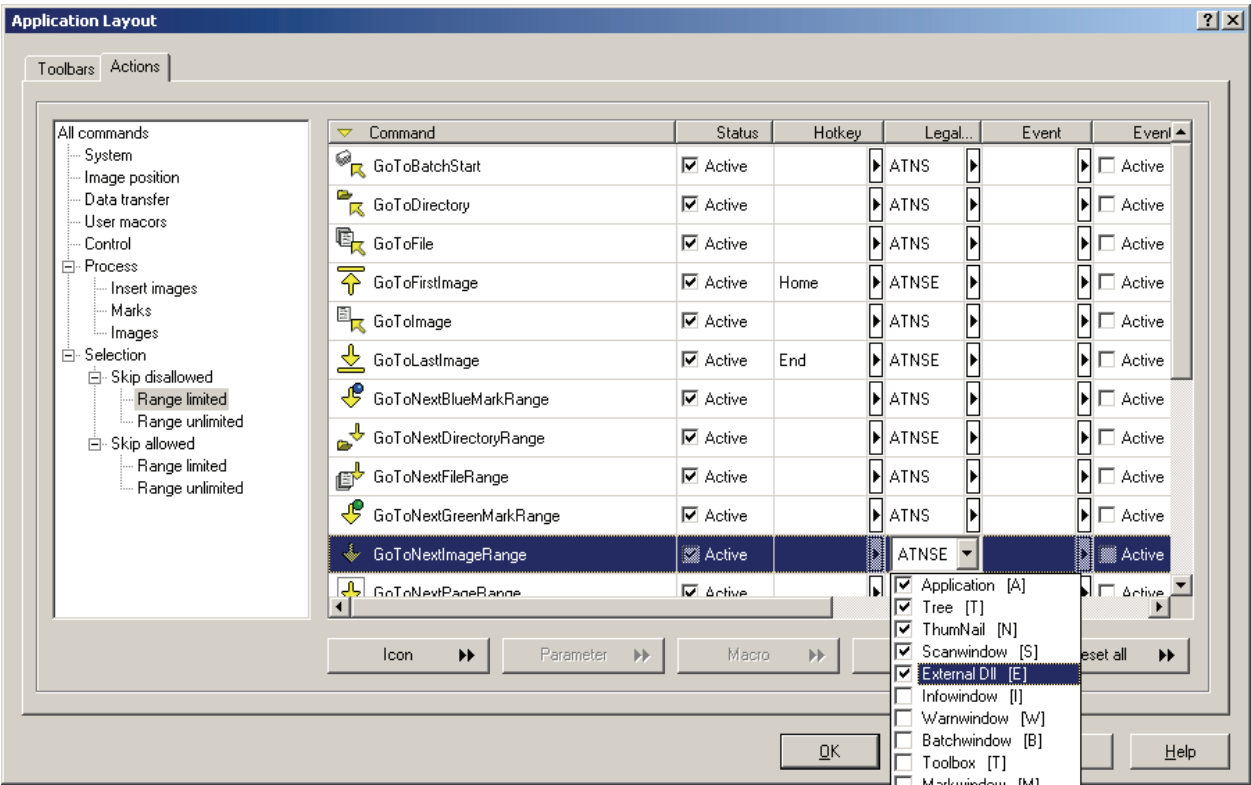


Illustration 12 –Keyboard Configuration

Here it is most important to select, in the Legal column, the **External DLL** option for the Legal area of the hotkey. You will find a detailed description for the keyboard configuration in the Reference Manual in Chapter Application Layout.

## 2.4 Presettings already during Scanning

The variables that shall be manually indexed later should already be preset during scanning (or loading).

It certainly makes sense to use OCR results for such presettings, or to preset them at least with valid values.

**Invalid preset values will be corrected when the data display in the embedded UDD, according to the prescriptions of the relative field.**

If for example a variable is pre-assigned with any OCR result, and the according line of the UDD allows only numbers, all non-numbers will be removed from the presettings.

If no preset is made, DpuScan will globally define the variables for this job, so that these variables exist for the image for which the dialog was opened, and for every other image of the batch that DpuScan will still "touch", but not for the images which were created before the first access.

During indexing, in this case, all variables as defined in the index dialog are assigned directly for the first image, and the relative last value is taken over for all following images of the batch.

In most cases, such a behavior is not desired, we therefore again recommend to preset all variables already during scanning.

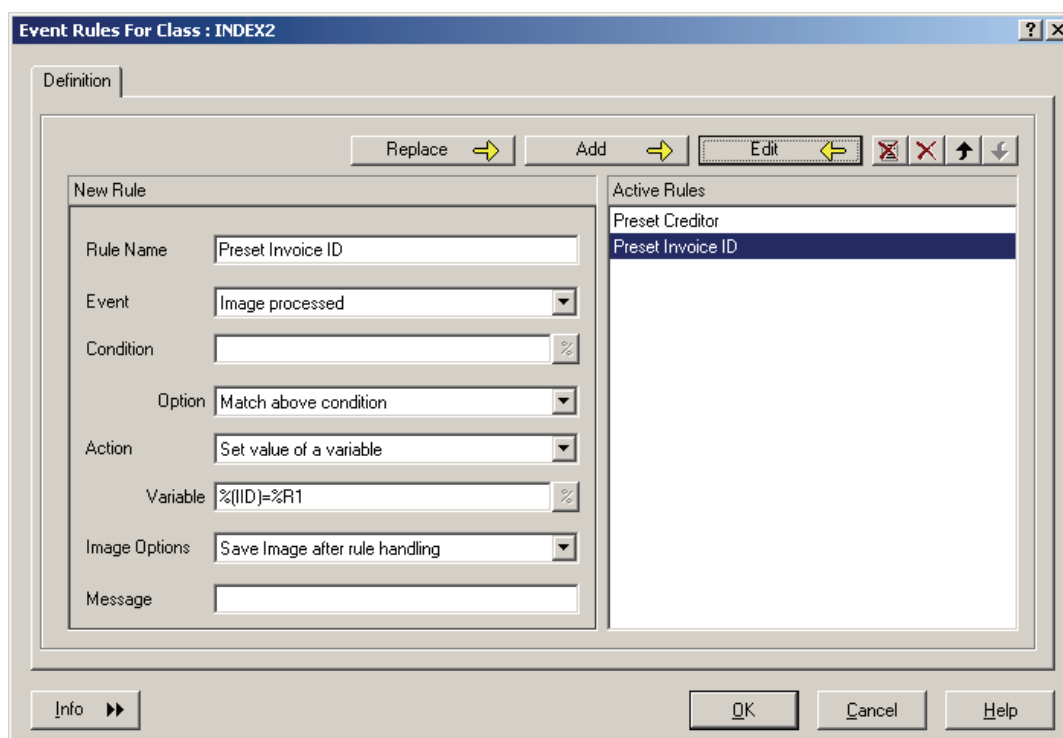


Illustration 13 – Presetting a Variable during Scanning

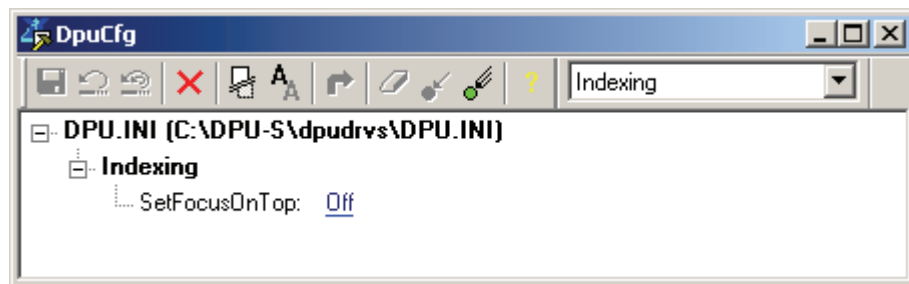
In this example, the user-defined variable %(IID) is preset with the first recognized barcode.

## 2.5 Positioning When Leafing Through the Images

JK\_INDEX is set in a way that when leafing through the images the input focus always stops at the same line. This behavior is based on the assumption that the fields are already preset and that thus indexing is more a correction for these values.

If however you always want to enter data for every image "every line from top to bottom", it makes more sense to restart from the top, on the next image.

Actually, there is no setting dialog as yet, therefore the SetFocusOnTop flag can be changed in the JK\_INDEX\_INIT only manually, or using the DpuCfg tool:



*Illustration 14 – JK\_INDEX Positioning the Cursor for the Next Image*

The tool displayed above is placed in the DpuScan folder and can be reached via entry in the DpuScan group in the Start menu.



## 2.6 Double Percent Signs for the Output

Usually, any imports into a database or into a DMS are made via the Batchfile of DpuScan. As the information is forwarded via several stations inside DpuScan, the variables will once more be populated with new values at the moment of finalizing the Index job.

Therefore it is necessary to enter those variables which will be altered during Indexing with **double percent signs** for their output:

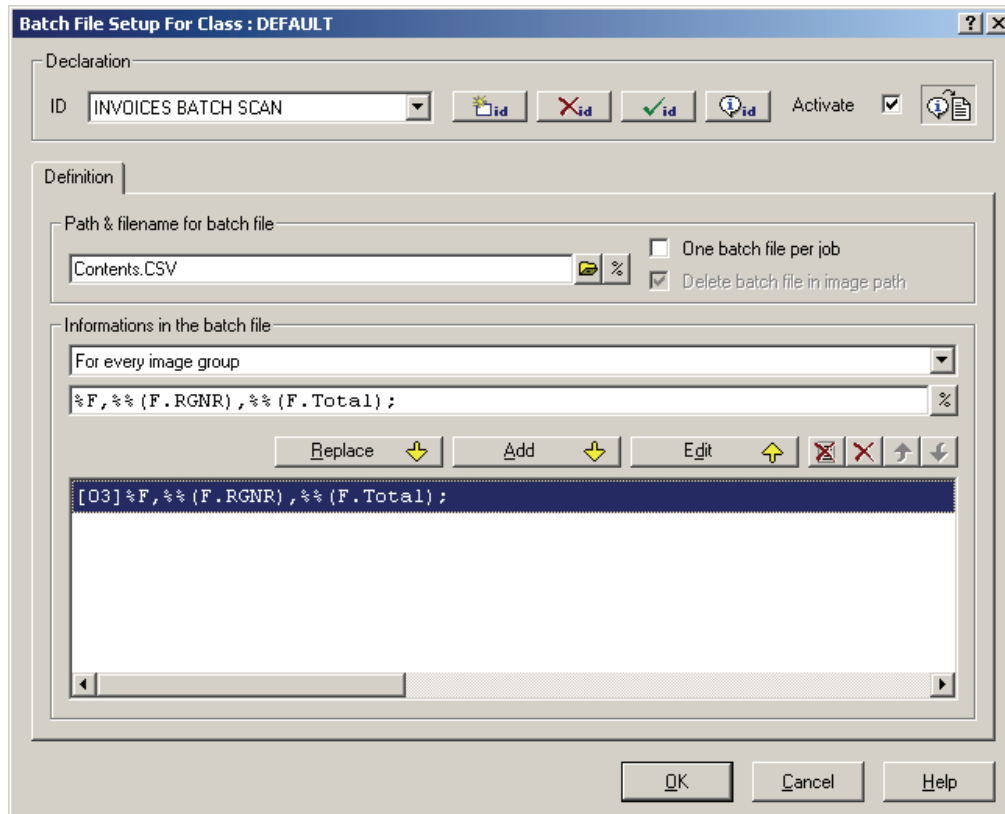


Illustration 15 – JK\_INDEX Double Percent Signs for the Output

In this example, the file name and the variables "RGNR" and "Total" will be put out for every image. As the file name "F" will not change while indexing, there is just one single percent sign before the wildcard for the file name.

For calling the Broker, it is also necessary to use such double percent signs.

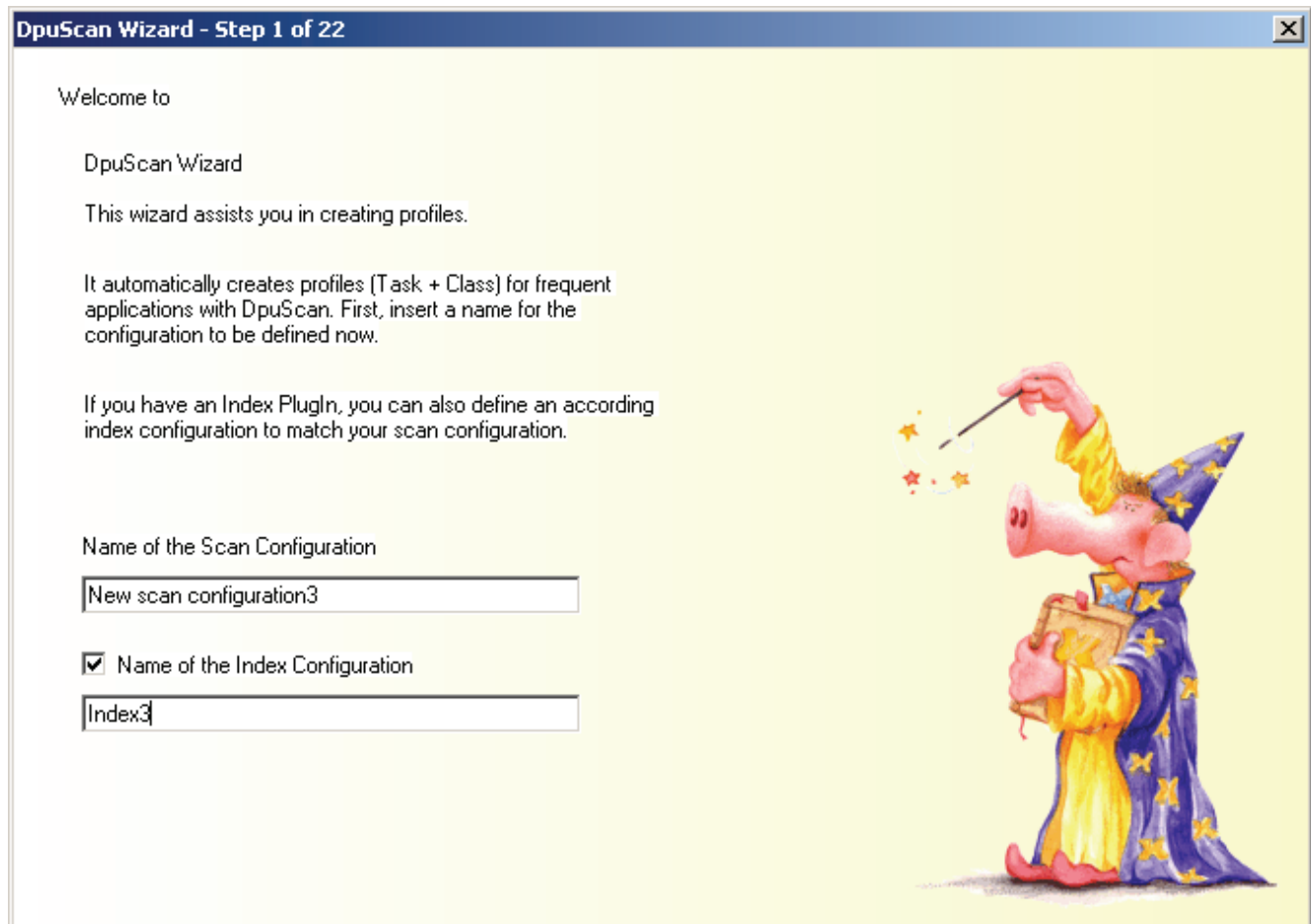
***It must be secured that the utilized variables are defined not only in the Class that is used for scanning, but also in the Class used for Indexing.***

***Also, the definitions of Broker and/or Batch File must match!***

### 3 Index Definition with the Wizard

Probably the most comfortable way to work with the Index Plugin is to create profiles by means of the Wizard. Already in the first step (of a total of 22 steps) you define, in addition to the name of the Scan Configuration, the name of the according Index Configuration:

Check the according box and type the desired name:



*Illustration 16 – Index-Definition with the Wizard, Step 1*

DpuScan will automatically create – further to the Task and the Class for the new scan profile itself – also the configuration for the Indexing Task. This way you can assure that the preconditions as per Chapter [2.6 Double Percent Signs](#) on page [17](#) are obeyed.

In the following steps, the profile is set in detail, as you know it from using the Wizard. Only in Step 18, the next input must be made for Indexing, refer [Illustration 17 – Index-Definition with the Wizard](#), [Step 18](#) on page [19](#):

DpuScan Wizard - Step 18 of 22
✕


### Define index variables

Number

	Name	Validity	Value allocation in	
			Scan job	Index job
#1	Booked	Image	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
#2	Paid	Image	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
#3	Discount	Image	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
#4	Invoice	File	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#5	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#6	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#7	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#8	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
#9	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Tip:

Index variables can be found out internally (date, time, computer name, barcode, patchcode, OCR, for example) or be determined by user inputs. It is also possible to correct internally allocated variables by user input. First, you set the number of index variables to be created, then you define the denomination (name) for every variable. This name will be used as prompt if a value must be allocated to an index variable in a dialog. A variable may be valid for an image, a document, a batch or even for several batches.



*Illustration 17 – Index-Definition with the Wizard, Step 18*

Initially, you must select the number of lines that you require for your desired embedded indexing dialog. Up to nine lines are allowed here.

DpuScan will automatically create the according dialog; in the **Name** column you enter the prompt in clear text, the "question" that shall be answered by a manual input. Next to it, a text field will display during the application where either the preset values appear, or values can be entered manually.

For every variable, the validity is set individually. Mostly, it will be the relative image, but there are more possibilities at choice, as follows:

<b>Application</b>	This variable is valid for all images in all jobs on this station.
<b>Class</b>	This variable is valid for all images in all jobs that base on the actual Class.
<b>Directory</b>	This variable is valid for all images in a folder.
<b>File</b>	This variable is valid for all images in a file.
<b>Image</b>	This variable is valid for the relative image.
<b>Job</b>	This variable is valid for all images in the batch/job.
<b>Task</b>	This variable is valid for all images in all jobs that base on the actual Task.

Use Step 20 to fix the Index variables. You can either preset a constant, or select the matching one from the internal DpuScan variables (% Codes).

**DpuScan Wizard - Step 20 of 22**

**Default in scan job**

	Name	Default	
#1	Booked	%x	%
#2	Paid		%
#3	Discount		%
#4	Invoice	%R1	%
#5			%
#6			%
#7			%
#8			%
#9			%

Tip: =

Here, you can determine the default value for index variables. You may set a constant, or select the suited one from the internal variables (% Codes). If value allocation via dialog is not foreseen for an index variable, the value allocation **MUST** be defined here.




Illustration 18 – Index-Definition with the Wizard, Step 20

If an Index variable shall not receive its value via a dialog, the according value **must** be assigned already here. In our example, this holds for the forth variable – Invoice. We assume that our invoices mainly consist of one only sheet, but they may also consist of several leaves. Every invoice bears a barcode sticker on its first sheet. This barcode will be recognized already during the scan process and is immediately assigned as Index variable. Since its validity was defined not only for the individual image, but for the relative file, this variable will automatically hold also for the following sheets of a multi-page invoice.

In all lines where the value shall (also) be assigned in the Index job, this value can either be captured automatically, or it is entered manually, or the automatically entered value can be corrected manually.

After completing the Wizard configuration, you will first process the scan Task, pause your job, change the Task (dropdown) to the according Indexing Task – then you can start to manually capture the data.

The DpuScan screen will then look as in the example on [Illustration 19 – Embedded Dialog in the Index Job](#) on page 22:

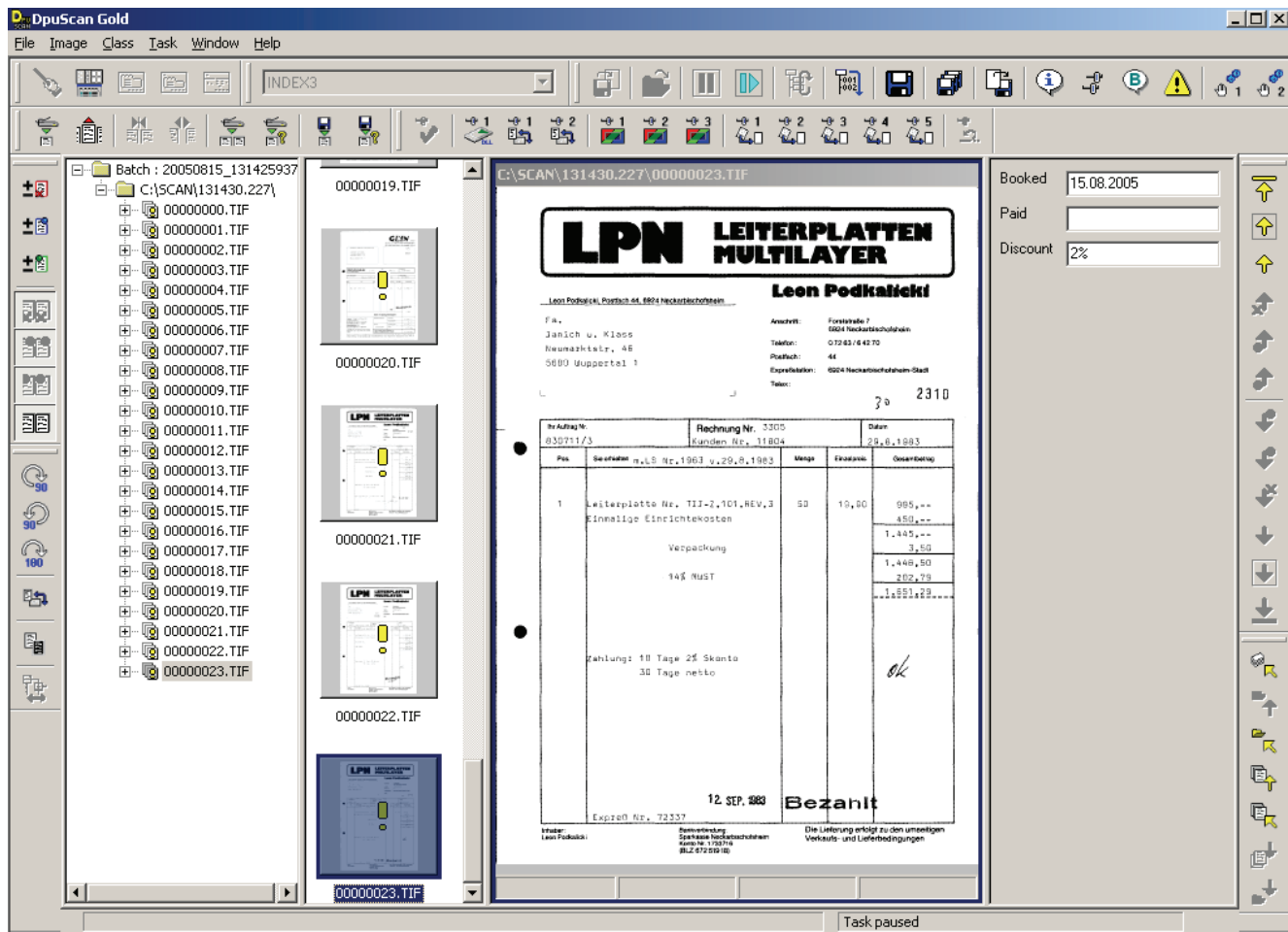
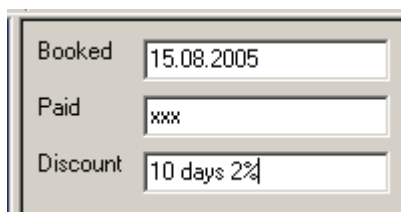


Illustration 19 – Embedded Dialog in the Index Job

Now you can manually select the relative image, make the according entries, or modify or correct the preset values.

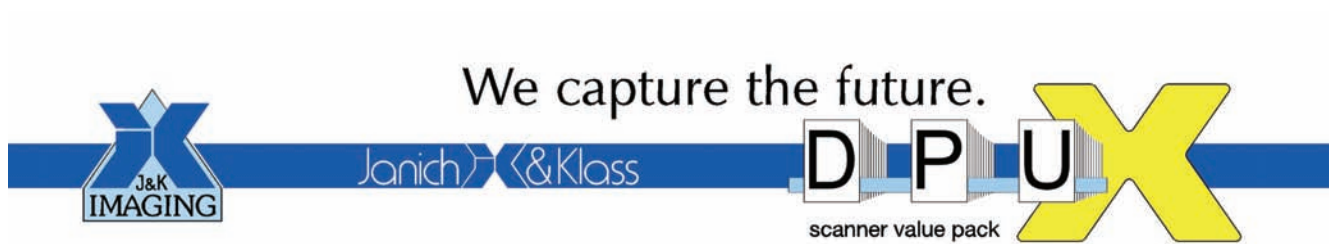


A screenshot of a software dialog box titled 'Index Dialog'. It contains three input fields with labels to their left. The first field is labeled 'Booked' and contains the text '15.08.2005'. The second field is labeled 'Paid' and contains the text 'xxx'. The third field is labeled 'Discount' and contains the text '10 days 2%'. The dialog box has a standard Windows-style border with a title bar and a close button in the top right corner.

Booked	15.08.2005
Paid	xxx
Discount	10 days 2%

*Illustration 20 – Manual Entries in the Index Dialog*

When the values are entered or corrected for the relative image, you select the next image by clicking its thumbnail in order to do Indexing there, until all images are processed.



Janich & Klass Computertechnik GmbH  
Zum Alten Zollhaus 24  
D-42281 Wuppertal  
Germany  
Phone: +49 (0)202 2708-0  
Fax: +49 (0)202 700 625  
<http://www.janichklass.com>

J&K Imaging, L.P.  
1633 Sands Place  
Marietta, GA 30067  
USA  
Phone: (770) 984-1212  
Fax: (770) 953-8399  
<http://www.JKImaging.com>

408.200405.001 9