



PlugIn for DpuScan

ODBC

Open Database Connectivity

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/ODBC-English.pdf>

© 2006 Janich & Klass Computertechnik GmbH, Wuppertal, Germany

September 13, 2006

Table of Contents

| | | |
|-------|---|----|
| 1 | Overview | 4 |
| 1.1 | Setting Up a Data Source | 4 |
| 2 | The PlugIn in the Task Definition | 7 |
| 3 | The PlugIn in the Class Definition | 9 |
| 3.1 | Configuration of the PlugIn in DpuScan | 10 |
| 3.1.1 | Property Page General | 11 |
| 3.1.2 | Property Page Percent Code | 12 |
| 3.1.3 | Property Page Information | 13 |
| 4 | Configuration of the Database Query | 14 |
| 5 | Using the ODBC PlugIn in the Interactive Mode | 17 |
| 5.1 | Before you get started | 18 |
| 5.2 | Creating an Indexing Mask | 19 |
| 5.3 | Data Transfer and Keyboard Assignment | 21 |
| 5.4 | Presetting already during Scanning | 21 |
| 5.5 | How to Query Data while Indexing | 22 |
| 5.6 | Positioning When Leafing Through the Images | 22 |
| 5.7 | Double Percent Signs for the Output | 23 |

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

The ODBC Plugin for DpuScan enables a reading access to a so-called ODBC data source.

By means of index data, matching data sets are found and the found values are copied in DpuScan variables. For example, a Client ID could be found via barcode, and the matching address is then retrieved via ODBC.

Data can be queried either in Process mode during the scan process, or later when the data are displayed in an embedded dialog for indexing.

Actually, writing access to databases is not possible.

1.1 Setting Up a Data Source

ODBC data sources can be created in the operating system. You have to select a database, an Excel sheet, or a folder with tables of text files. When the matching ODBC driver is used, access to the data is granted in form of tables and columns. The details of this process strongly depend on the kind of database and can therefore be described here only in an example.

We suppose that the folder C:\DB holds a Microsoft Access database named JKTEST.MDB. This database contains a table PERSONS with the key column ID and some more columns like FIRSTNAME, NAME, ADDRESS, etc.

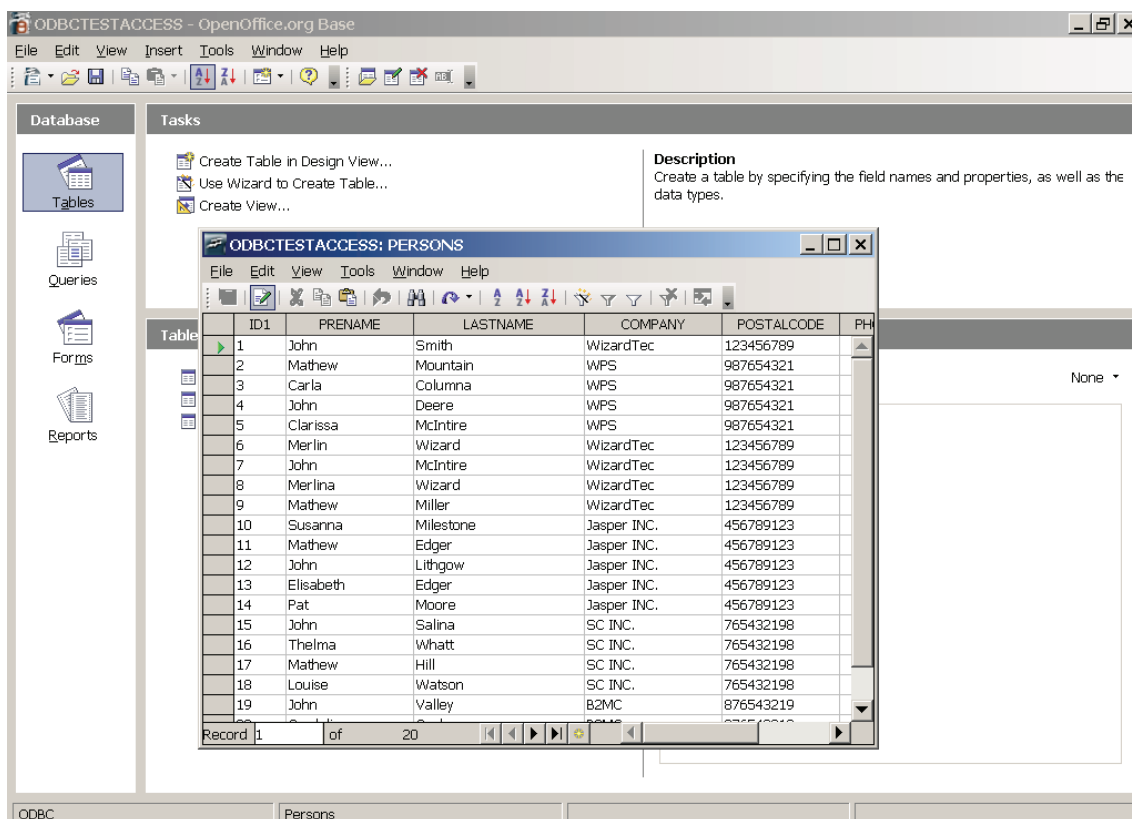


Illustration 1 – Example: Table in an Access Database

Actually, DpuScan can handle only fields of type TEXT.

A source shall be established, named JKTEST. Via the Windows Start menu you reach the system control, and there the administration. There, you will find the item ODBC data sources:

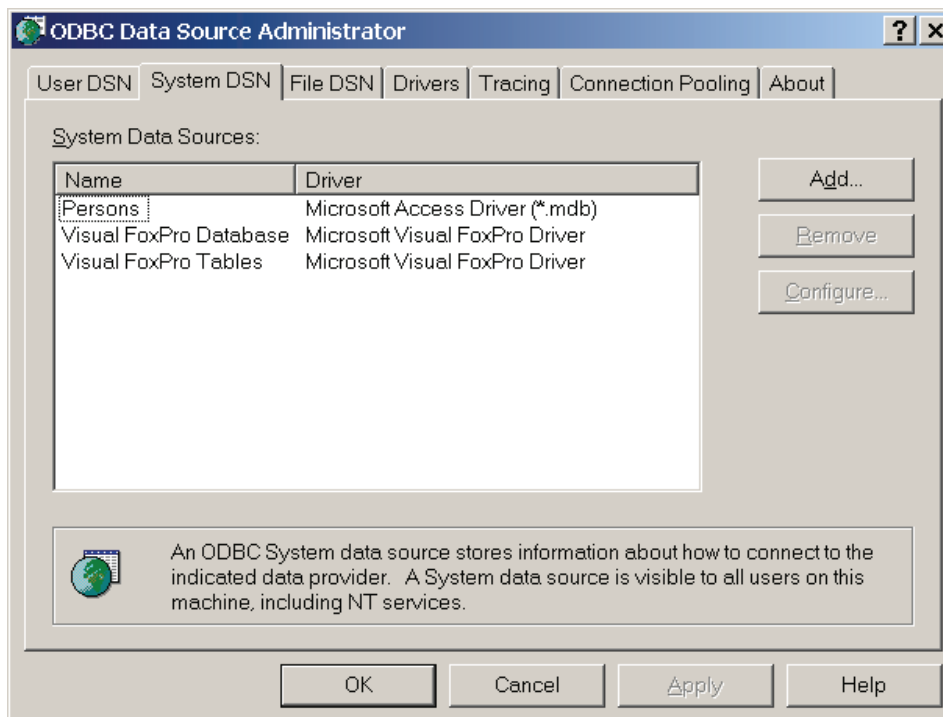


Illustration 2 – ODBC Database Sources Administrator

Click **Add** to reach the dialog for selecting the ODBC driver. In our example, we selected the driver for Microsoft Access:

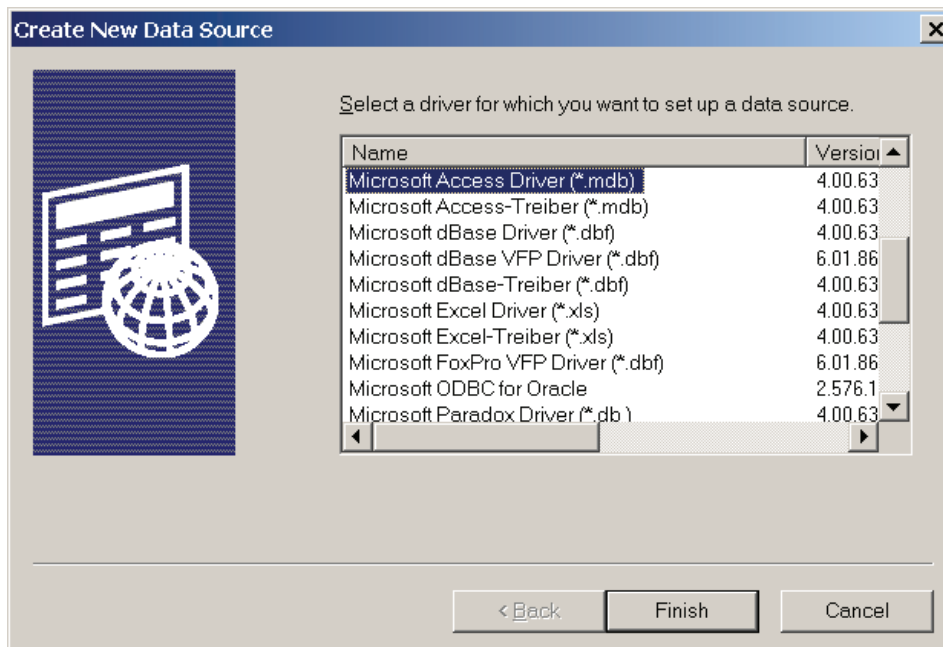


Illustration 3 – Setting Up a New Data Source

The following steps determine the details, in this case the storage place for the database:

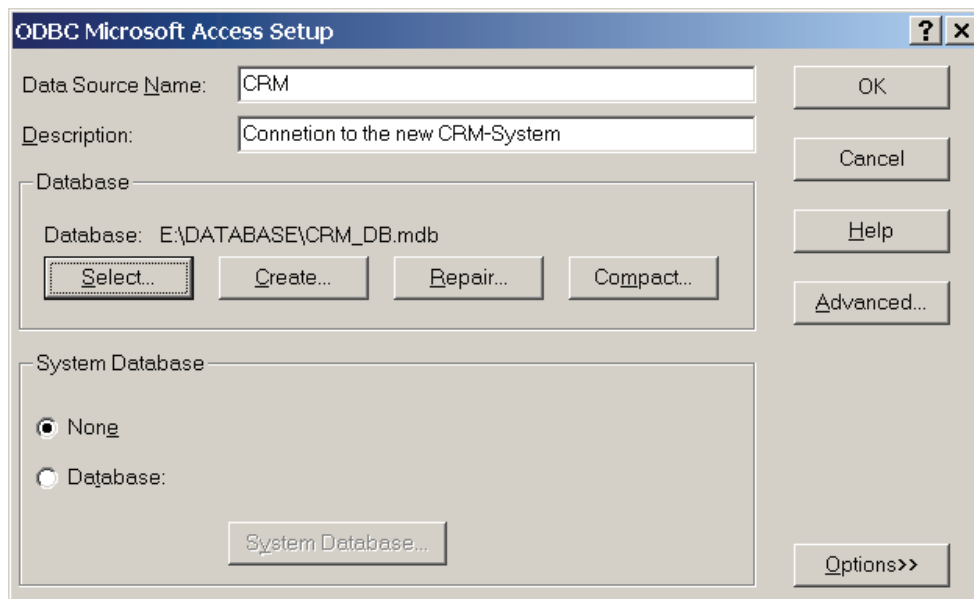


Illustration 4 – ODBC Microsoft Access Setup

Hint: During its configuration, the Plugin will query the existing data sources; a Login procedure may display, or an Open database dialog. Therefore please assure that the existing data sources are in connection to their according databases. Not required data sources should be deleted.

2 The Plugin in the Task Definition

As mentioned in Chapter [1 Overview](#) on page [4](#), the ODBC Plugin can be used in Process mode in the Task. After getting the image and eventually after some recognition, the Task step "Call plugin for every image" must be executed. This action is available only if the Plugin was loaded for the actual Class.

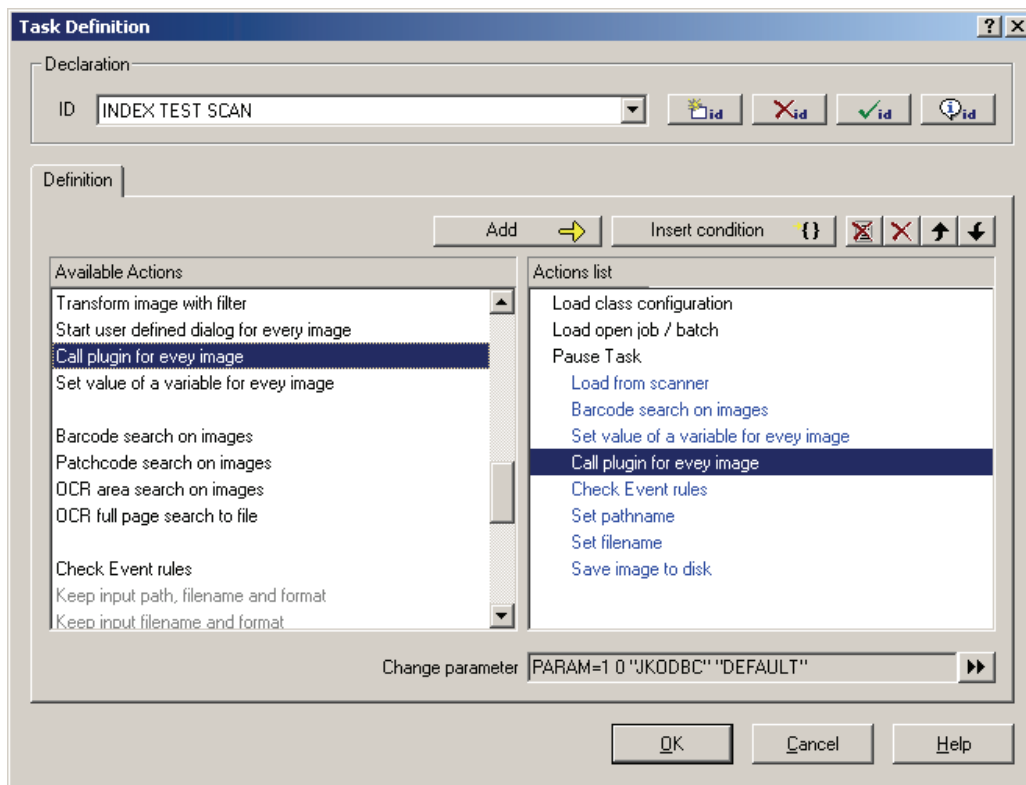


Illustration 5 – Calling the Plugin in the Task

Before the Plugin is called, first the index value is set which is used to search in the database; for a closer description please see Chapter [5.4 Presetting already during Scanning](#) on page [21](#).

For the subsequent call of the Plugin, one of the Plugin configurations can be selected by its **ID**. Furthermore, the call can be restricted to certain images.

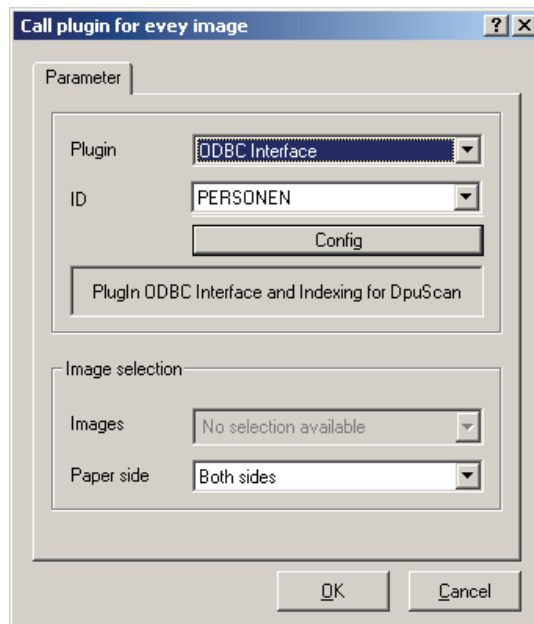


Illustration 6 – Calling the PlugIn in the Task

It is not necessary to call the PlugIn before the scan loop, as initializing is automatically executed already while loading the Class.

In such an application, an index value (a DpuScan variable) is therefore transmitted to the PlugIn and several related values are returned to DpuScan from the ODBC source in due case. These values are then assigned to DpuScan variables and are then internally available in DpuScan like any other variable.

3 The Plugin in the Class Definition

The Plugin must be loaded and configured within the Class. This is done in the **Class configuration**, on the **Process** property sheet. There, the **Plugins** button opens the dialog with the list of the actually active Plugins. Use the **Add** button to reach the selection dialog for available Plugins.

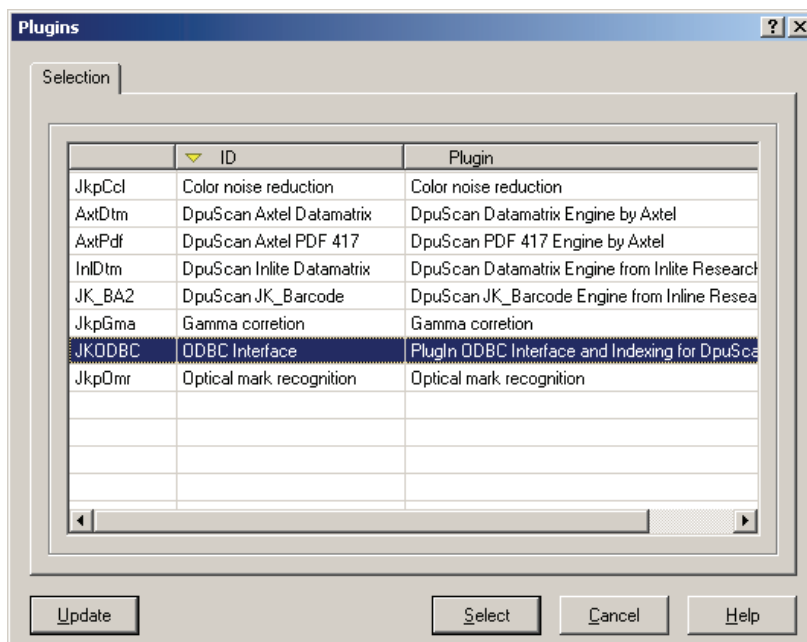


Illustration 7 – Plugin Selection

After its selection, the Plugin will display in the list of Defined Plugins.

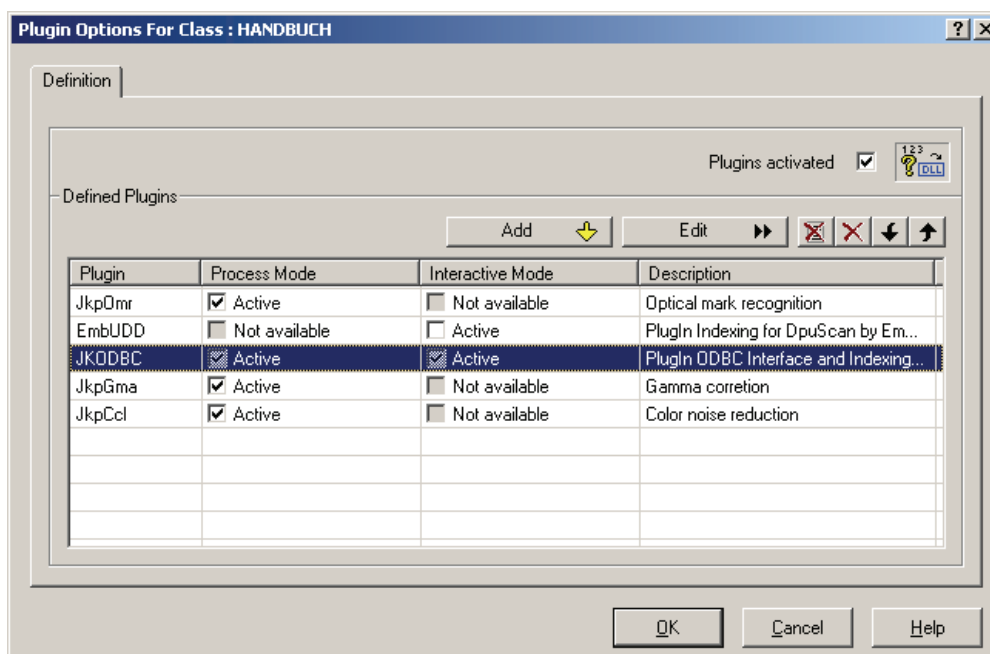


Illustration 8 – Plugin Configuration for the Class

The ODBC PlugIn is now loaded for usage with this Class.

Please remember that the "PlugIns activated" checkbox must be marked by a hook as otherwise the PlugIns will not be used.

The entry in the list in the above illustration shows activated checkboxes in both columns, **Process mode** and **Interactive mode**. This means that the PlugIn can be used while scanning, or in the Pause mode.

3.1 Configuration of the PlugIn in DpuScan

You can open the PlugIn configuration with a double-click to the list entry – see [Illustration 12 – PlugIn Configuration](#) on page 11. It offers, in its top area, the control elements for administering different PlugIn configurations.



Selects an existing configuration and assigns it to this Class.



Creates a new configuration. The settings of the actual configuration are copied and can then be modified.

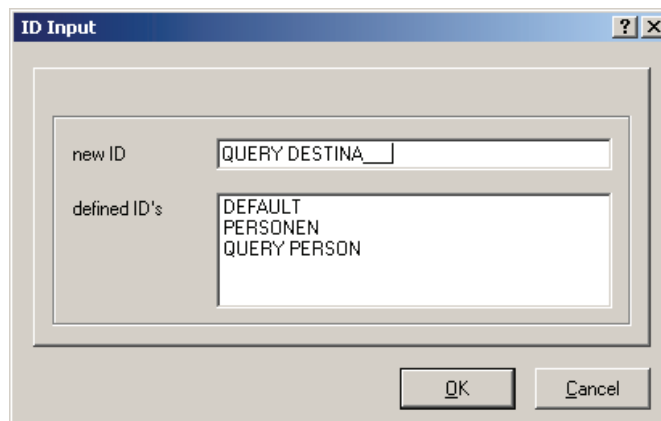


Illustration 9 – Creating a new ID



Deletes the actual configuration. The below warning message will pop up in due case:

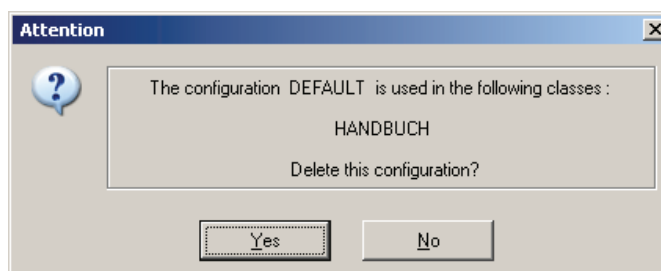


Illustration 10 – Warning before Deleting



Assigns the modifications to the actual configuration.



Opens a dialog which informs in which Classes the actual configuration is also in use.

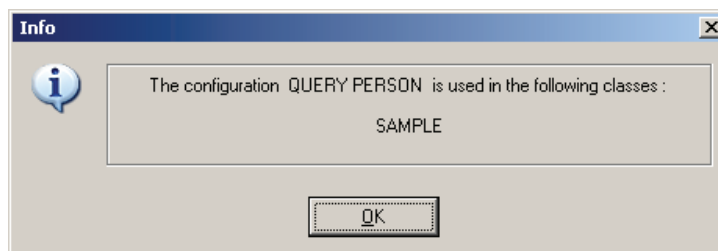


Illustration 11 – Information about Usage of the ID

In the lower area, the Property Pages **General**, **Percent Code** and **Information** are displayed.

3.1.1 Property Page: General

The **General** page gives detail information about the PlugIn, in this case about its version and about the manufacturer.

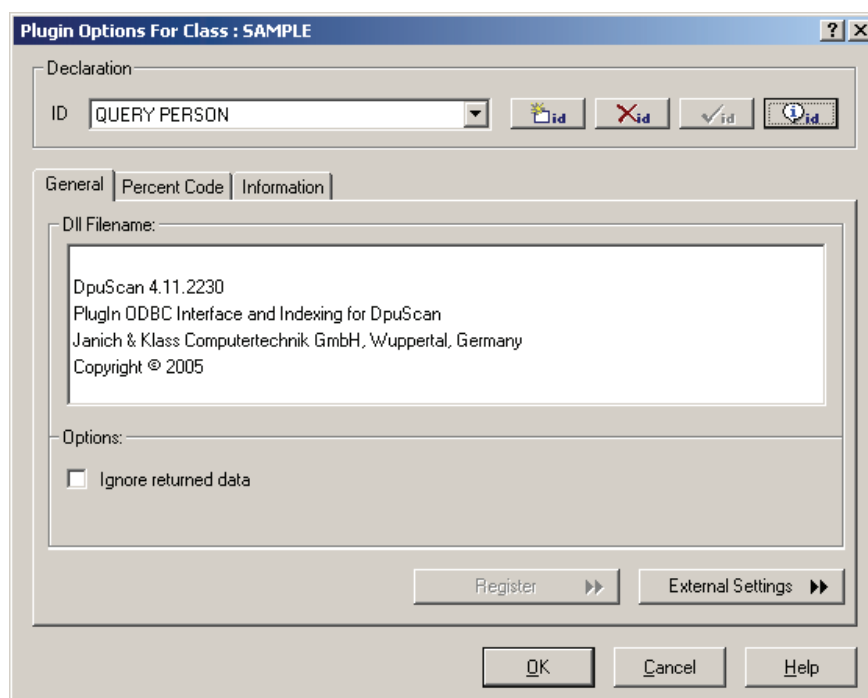


Illustration 12 – PlugIn Configuration

Ignore returned data

The variables will not be returned to DpuScan if this check box is activated.

Register

Opens the dialog for entering the registration key.

External Settings

Opens the dialog for PlugIn configuration, see Chapter [4 Configuration of the Database Query](#) on page [14](#).

Prior to its first usage, the PlugIn must once be registered. Please click the tool button Registration and enter the registration key in the next dialog, in order to unlock the OMR-PlugIn.

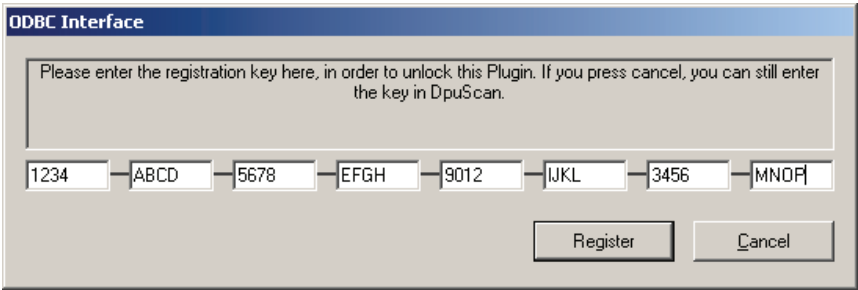


Illustration 13 – Registration Dialog

Registration for the ODBC PlugIn is void for DpuScan QSI. Here, the ODBC PlugIn belongs to the scope of delivery and is automatically decrypted by the QS -Dongle.

3.1.2 Property Page: Percent Code

The **Percent Code** page lists the variables which are used by the PlugIn or which were defined in the configuration.

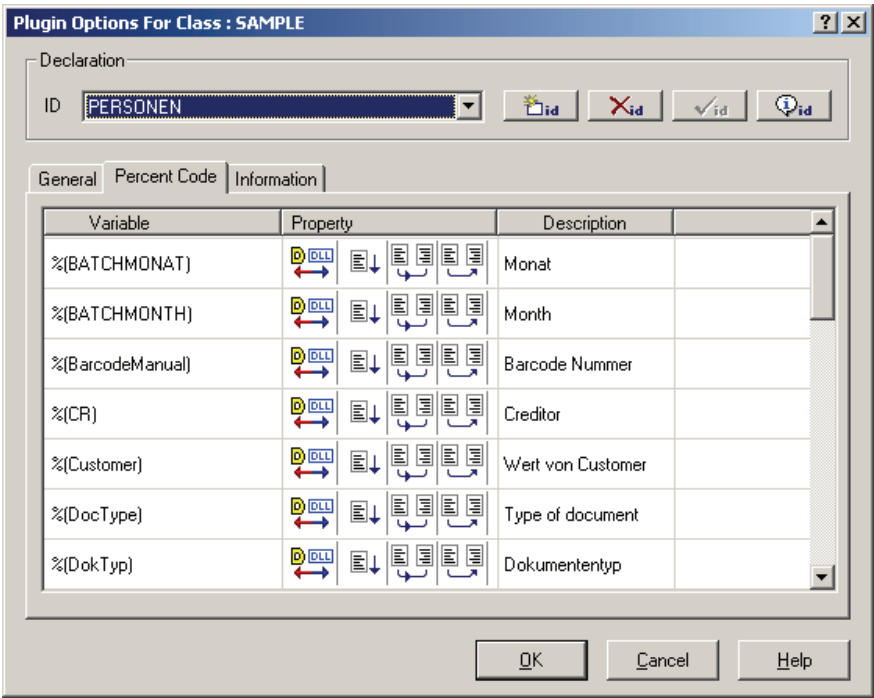


Illustration 14 – List of Variables

In the listed table you can see, next to the name of the variable, the **Property** column. Special symbols mark how the variables are processed by the system.

With the ODBC PlugIn ODBC, the following symbols display:



DpuScan send its data to the PlugIn; the modified data are returned by the PlugIn.



In the **Process mode** the data are sent as a Task step.



In the **Interactive mode** the data are exchanged **before** the changed selection is displayed (when, for example, you switch to the next image).



In the **Interactive mode** the data are exchanged **after** the selection change is executed and the new image is displayed.

The **Description** column describes the provenience of the variables.

We recommend, mainly if you use your own names for the variables, that you check them, after completed configuration, at this instance.

3.1.3 Property Page: Information

This page offers, in form of a tree view, information about the name of the PlugIn, its manufacturer and the version. The ID lists the windows, images and variables that are used by the PlugIns.

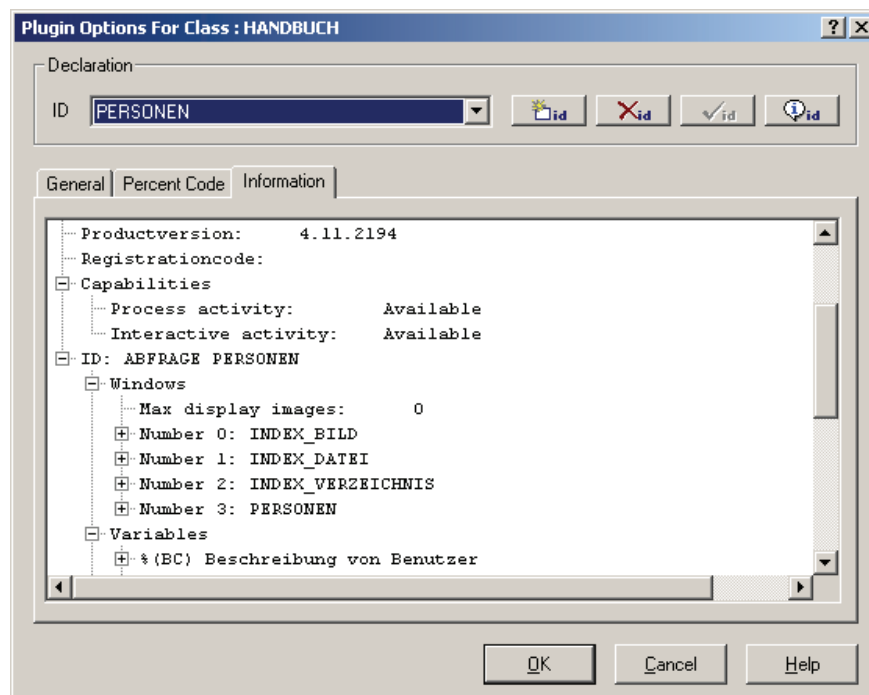


Illustration 15 –Property Page: Information

In the above example, the user-defined dialogs (UDDs) are shown which can be used for indexing. Below, the variables display which are exchanged with DpuScan.

4 Configuration of the Database Query

When you click, on the **General** page, the **External settings** button, the dialog opens for configuration of the database query.

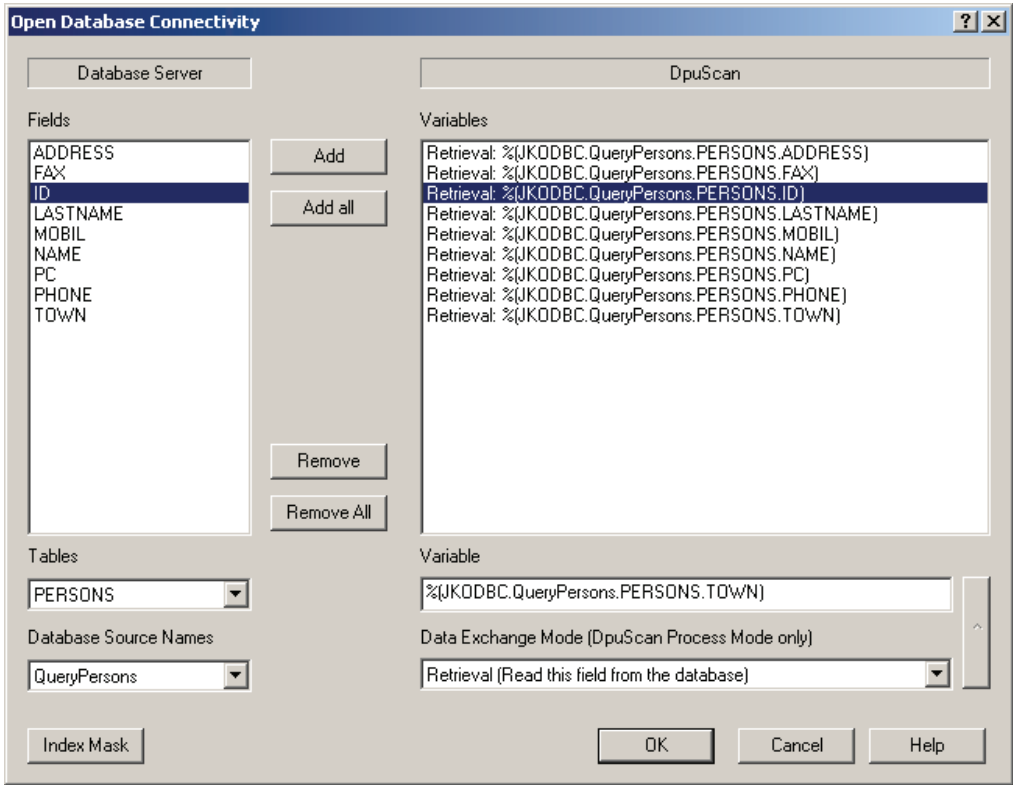


Illustration 16 – Setting Dialog for Database Query

When the dialog opens, the existing ODBC data sources are detected. Here, it may occur that the according ODBC driver asks for further information about the data sources, eventually execute a login, or query the memory space of the basing table. It will not influence the further listing of sources when you can overleap one of these dialogs only be Cancel.

Once the sources query is completed, they will display in the left bottom corner in the dropdown list called **Database Source Names**. After selecting the sources, you can select **Tables** and **Fields**, and assign them to the **Variables** in DpuScan (%-Codes).

Already existing variables can be copied into the editing line, by a double click. Then, also the assigned fields in the database will be displayed.

Fields Displays all fields in the actually selected table.

Tables Displays all available tables in the database. If possible, those tables will be omitted that only serve for the administration of the database.

Database Source Names

Here you can select an existing database source.

ODBC database sources can be set in the "Administration" of the computers. Usually, you must log in at the relative database, so that DpuScan will later not ask for the user or for a password.

Variables

This list shows the arranged DpuScan percent codes and their behavior at the query.

A double click to the line copies its text into the editing field and also displays, on the left side, to which field this percent code belongs.

Variable

In this editing line, you can change the name of the percent code to your demands.

By stating an according prefix, the validity range of the code can be fixed, like for all images in a multi-page file.

Data Exchange Mode (DpuScan Process Mode only)

Here you can set how the fields must be handled during the database query.

Unknown

These fields are just skipped during the query.

Index

All data sets are searched from the database where the according field has the same value as the percent code at the output.

Retrieval

These percent codes will be filled with the results from the database.



This button assigns the modifications again to the percent code.

Add

Creates a DpuScan percent code for the selected field.

The name is composed from the name of the Plugin, the database source, the table, and the name of the field. It can, however, be modified as you like.

Add all

Creates new percent codes for all fields.

Remove

Separates the percent codes from the variable and deletes it from the list.

Remove all

Deletes all variables from the right side.

Index Mask

Creates a user-defined dialog (UDD) that displays as prompts the names of the percent codes and assigns the entered values to the percent codes. Please find a detailed description of the UDDs in the Reference Manual for DpuScan.

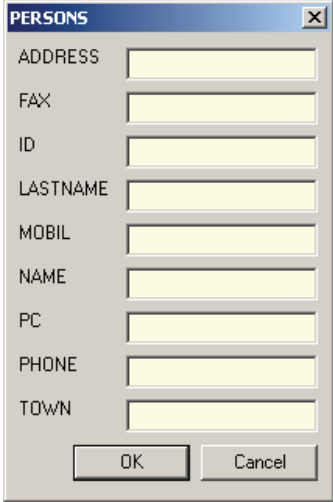


Illustration 17 – Automatically Created Indexing Dialog

The indexing fields are automatically sorted upwards, and they receive the background colors yellow (for not yet processed) and white (already filled in).

The following lines form the retrieval fields, with the colors light yellow and white.

OK, Cancel, Help

These standard buttons work in their usual manner.

5 Using the ODBC Plugin in the Interactive Mode

How the Plugin is used in the process mode directly while scanning, has been described in Chapter 2 [The Plugin in the Task Definition](#) on page 7. Here, however, we describe how to re-open the batch after scanning in another job where data can be entered manually.

Please bear in mind that the ODBC-Plugin will not be loaded in the Class if it is a mere scanning Task.

The job is opened only to enter data manually, therefore the Task definition is correspondingly short in this case:

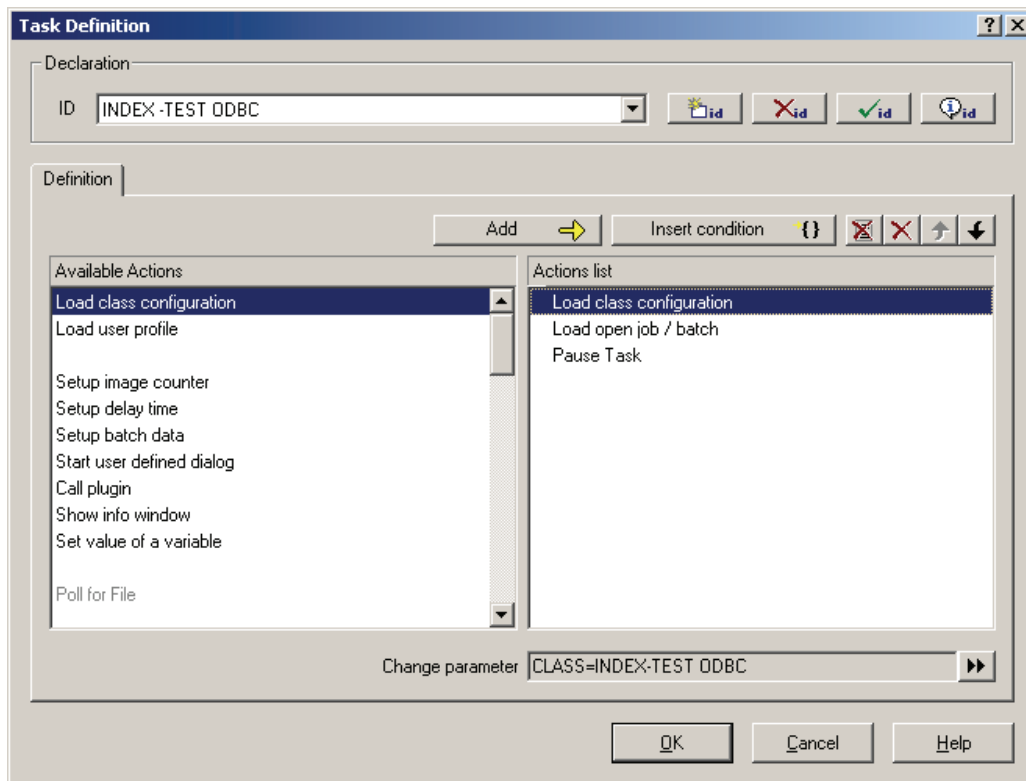


Illustration 18 – Task for Plugin in the Interactive Mode

The configuration itself is set in the **Class** during the screen layout. If Plugins that work in the interactive mode are loaded in the Class, their exported sub-windows are available for the screen layout in the **Window type** dropdown list:

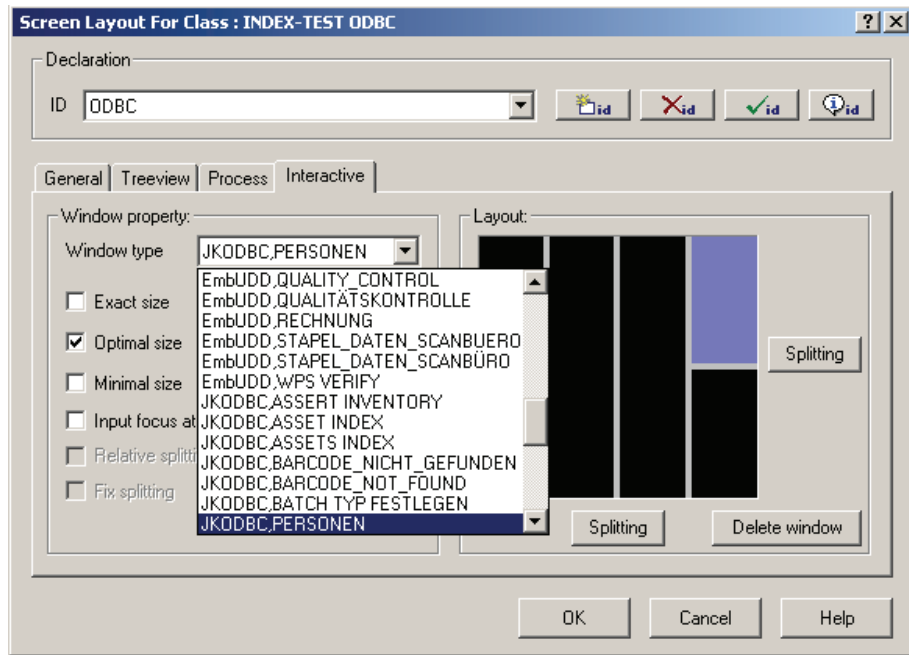


Illustration 19 – Screen Layout for PlugIn in the Interactive Mode

In this example, the Window type PERSONEN of the PlugIn is assigned to the sub-window in the upper part of the right column of the screen.

5.1 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 you 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.

You must start any new stack with a new complete document.

5.2 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 [5 Using the ODBC Plugin in the Interactive Mode](#) on page [17](#). All available UDDs are cited in the drop-down list of window types, with the prefix **EmbUDD**, refer [Illustration 19 – Screen Layout for Plugin in the Interactive Mode](#) on page [18](#).

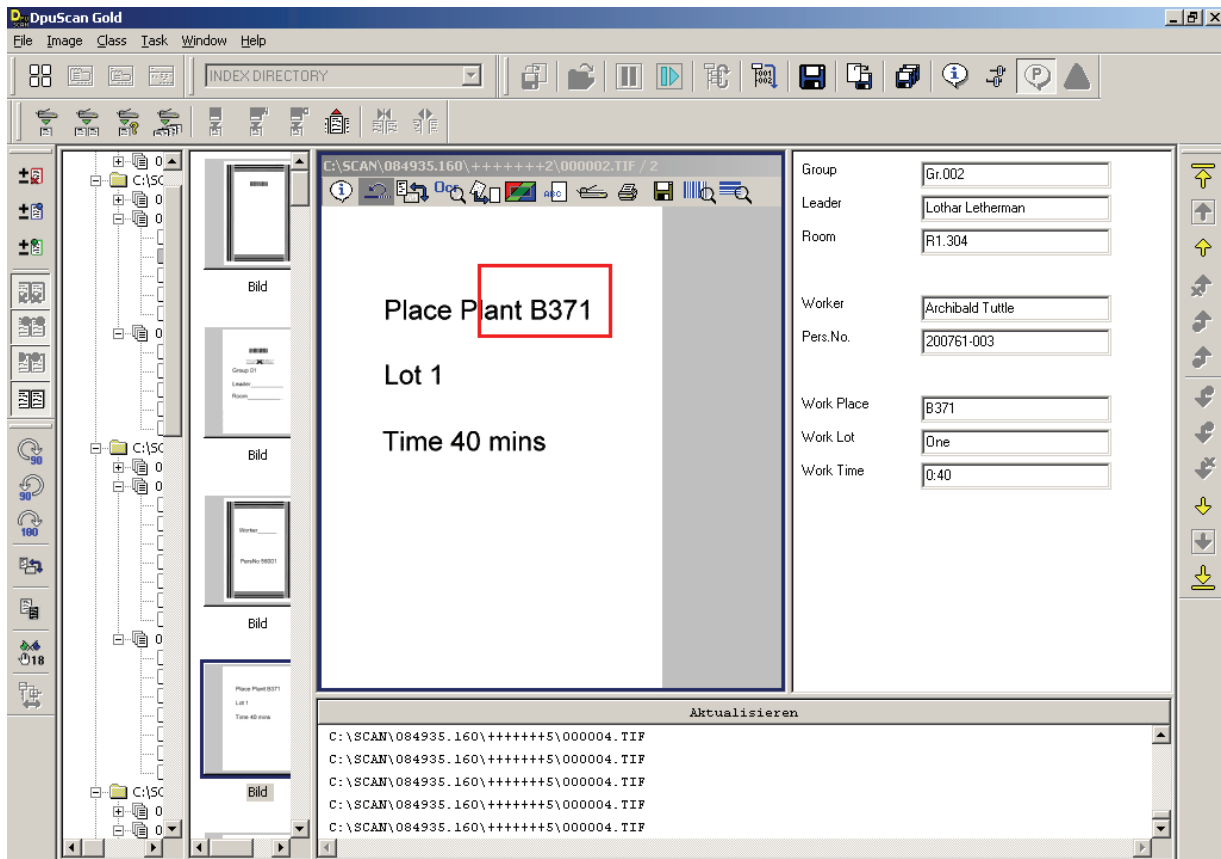


Illustration 20 – JK_ODBC Embedded Dialog

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.

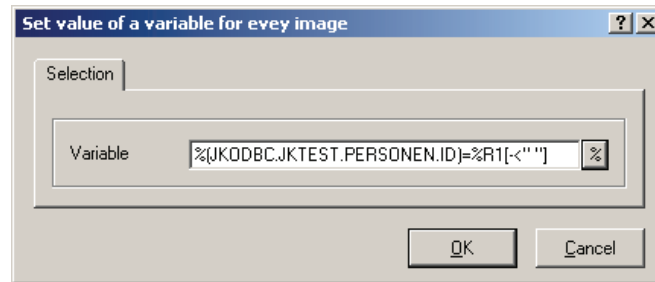


Illustration 22 – Presetting the Variable in the Task

In this example, the contents of the first barcode found are assigned to the Index variable `%(JKODBC.JKTEST.PERSONEN.ID)`; trailing blanks will be removed. Please find a detailed description about formatting the DpuScan variables in the Reference Manual for DpuScan.

5.5 How to Query Data while Indexing

When you later, during indexing, click into an image, an automatic research is started in the database whether a data set exists with this certain ID. If this set exists, the other fields in the indexing mask will be completed accordingly.

If there is more than one hit in the database, the first matching data set is returned.

The Index operator can now enter another value in the ID field. In order to restart the database query, you might also use the **F12** key. Also, the data are again queried if you moved one image forward and then return to it.

5.6 Positioning When Leafing Through the Images

JK_ODBC 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:

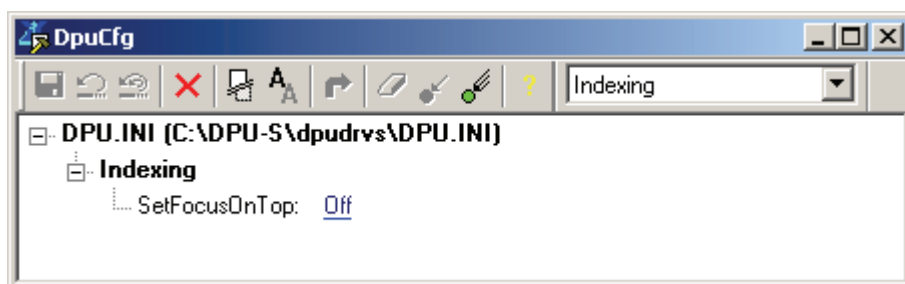


Illustration 23 – Positioning the Cursor for the Next Image

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

5.7 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:

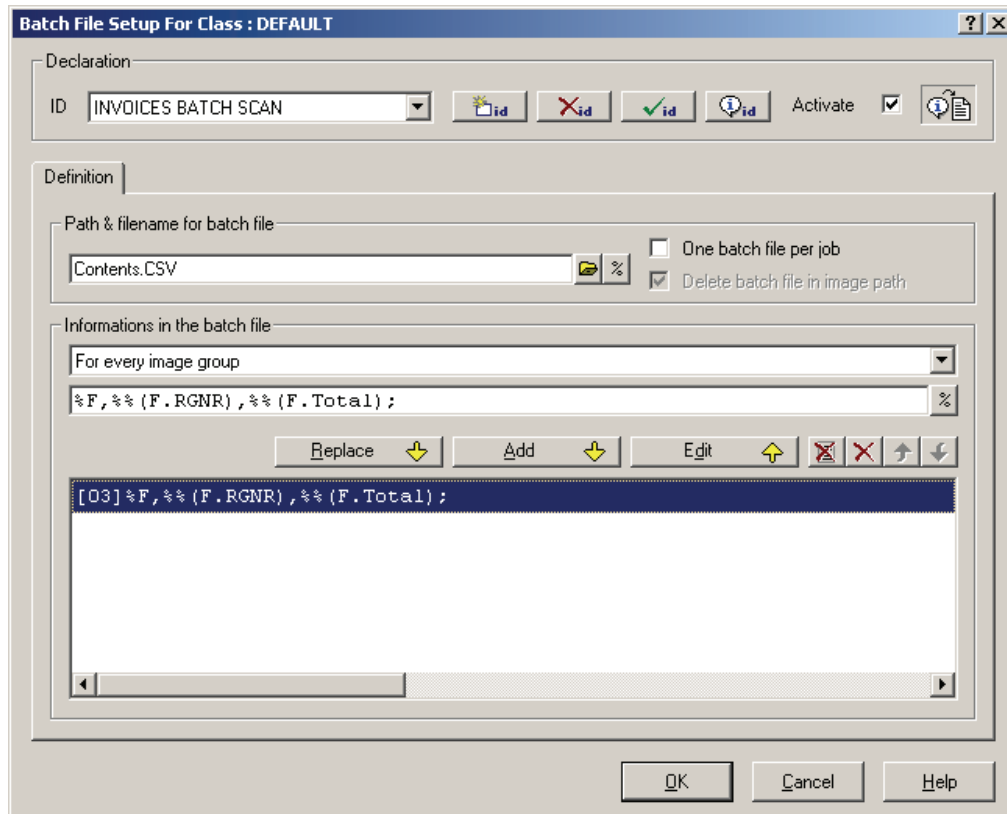


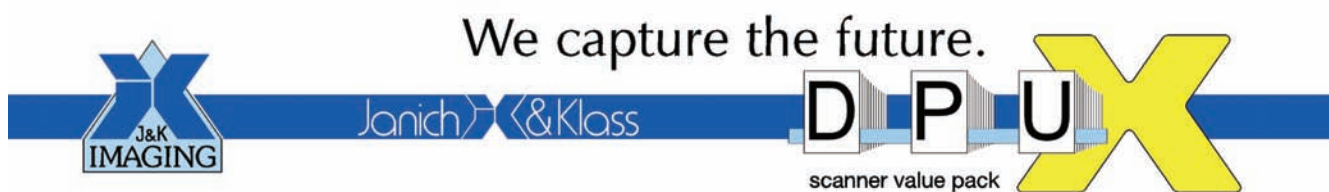
Illustration 24 – JK_ODBC Double Percent Signs for the Output

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, see Chapter 5.4.

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



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>

408.200408.001 3