



DpuScan

Janich & Klass
Computertechnik GmbH



DpuScan 6.x

Referenzhandbuch

Copyrights

© 1997 bis 2021 Janich & Klass Computertechnik GmbH. Alle Rechte vorbehalten. Gedruckt in Deutschland. Die in dieser Dokumentation enthaltenen Informationen sind Eigentum der Janich & Klass Computertechnik GmbH. Ohne schriftliche Genehmigung der Janich & Klass Computertechnik GmbH begründen weder der Empfang noch der Besitz dieser Informationen irgendein Recht auf Reproduktion oder Veröffentlichung irgendwelcher Teile davon.

Warenzeichen

Alle Produktnamen und Logos sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Eigentümer.

Haftungsausschluss

Die Anweisungen und Beschreibungen in diesem Handbuch waren zum Druckzeitpunkt zutreffend. Wir behalten uns jedoch das Recht vor, sowohl Beschreibung als auch Produkt jederzeit ohne Benachrichtigung zu ändern. Nach dem derzeitigen Stand der Softwaretechnik ist es nicht möglich, Programme zu entwickeln, die unter allen Bedingungen in jeder Konfiguration fehlerfrei arbeiten. Die Janich & Klass Computertechnik GmbH übernimmt keinerlei Haftung für Defekte, die direkt oder indirekt durch Fehler dieses Handbuches, Weglassen von Informationen oder durch Unstimmigkeiten zwischen diesem Referenzhandbuch und dem Produkt entstanden sind.

Aktualität

Es ist möglich, dass im Internet eine neuere Version dieses Handbuches verfügbar ist. Wir empfehlen deshalb, die Version anhand des auf dieser Seite abgedruckten Datums mit der Version auf dem Internet zu vergleichen. Falls die Version im Internet neueren Datums ist, sollten Sie diese herunterladen und ggf. selbst ausdrucken.

Die aktuelle Version des DpuScan Referenzhandbuch finden Sie im Web unter:

<http://www.jkimaging.com/pdf/DpuScan-Referenzhandbuch.pdf>

Inhaltsverzeichnis

1 ODBC	4
1.1 Usage of ODC PlugIn	5
1.2 Configuration ODBC	5
1.2.1 Configuration Data Connection	8
1.2.2 Configuration Query Result Options	10
1.2.3 Configuration Broker	10
Sequence of Broker Events	12
1.3 Variable Assignment	15
1.4 Viewing Results	16
1.5 Version	16

1 ODBC

About Plugins

A Plugin is a software module that can be used in DpuScan to perform certain tasks. It therefore has a well defined interface so that it can be loaded and configured in the scan software. DpuScan can send instructions, pictures and text data to the Plugin. On the other hand DpuScan can accept the results in the form of images or text data. Before a Plugin can be used in a task profile, it must be loaded in the base profile. Some Plugins must be licensed for use, i. a suitable **key** must be specified.

The ODBC Plugin

The ODBC Plugin provides access to a data source. A data source is usually a database but can also be a comparable collection of tabulated data. The Plugin works only with text data; no pictures are processed. A typical job is the reading of an index value during scanning or indexing and then the acquisition of the corresponding data record from a database. For example, a customer number can be found by barcode and then the appropriate address can be determined by ODBC.

Access to such a source can be done in various ways.

Access via an ODBC data source:

The data source has been set up (by the administrator), the operator usually does not have to log on to the database. The interface to the data source corresponds to the requirements of the "Open Database Connectivity". For operation with the Plugin it is enough to know the table, which one wants to query. The individual columns can be assigned to %-Codes in DpuScan.

Access via OLEDB data connection:

The operator must log on to the database using a ConnectionString. After this, it can issue any SQL commands, e.g. also write data to the database. The interface to the data source uses "Object Linking and Embedding" Database objects. The Plugin allows the returned field contents to be assigned to different %-Codes in DpuScan during a read access, ie a SELECT command.

Miscellaneous

The basic prerequisite for the functioning of both methods is the existence of a suitable interface software on the database side, ie a so-called "ODBC Driver" or a "Database Provider". They are usually found on the manufacturer's pages. The access authorization must also be guaranteed by the database administrator.

When the data is transferred to the program, it is assumed that exactly one value is assigned to each %-Code. Therefore a selection can be offered for several hits.

For more complex queries the Script-Plugin can be used.

Licensing:

The ODBC plug-in is used for the QSI program variant without a key, in all other variants a key is required.

See also:

[Usage ODBC](#)

[Configuration ODBC](#)

[Variable Assignment](#)

[Viewing Results](#)

[Version](#)

1.1 Usage of ODC PlugIn

Before a PlugIn can be used, it must be **loaded** and licensed in the base profile. It can then be used at various points in the application:

Process mode

To do a query call the PlugIn in the task list as a step 'Call PlugIn for every image'. This step will be available after loading the PlugIn in the class only.

Toolbar command and Macro

In the application layout a button can be configured to call a PlugIn. This action can be part of a user macro also.

Interactive mode

In a UDD edit fields and drop-down lists can be configured, to retrieve data from a database for example when pressing the F12 key. The definition of variables assignment is part of this PlugIn.

A detailed description of these options can be found in the corresponding parts of the applications documentation.

See Overview for more information.

Broker Mode

If the loaded plug-in subprofile is appropriately set, the ODBC PlugIn can be called to certain events, e.g. already at the start of the task.

See also:

[Usage ODBC](#)

[Configuration ODBC](#)

[Variable Assignment](#)

[Viewing Results](#)

1.2 Configuration ODBC

The dialog for defining the ODBC connection shows a tree view on the left side, a table on the right side, and a few additional controls in the middle.

Left Side

The tree view shows the available data sources and data connections.

Data Sources

The view can be restricted to the actual data sources used. When you open the dialog, the existing ODBC data sources are determined first. It may happen that the corresponding ODBC drivers ask for further information about the data sources, e.g. to log in or to query the location of the underlying table. It has no influence on the further listing of sources if you skip one of these dialogues by clicking Cancel. When the query of the sources is completed, the determined data sources are displayed. Tables and fields can now be selected and variables assigned to them.

In this part of the tree these symbols are used:



ODBC data source, usually a database. A double-click displays all tables in the database. Attention, this can take some time with large databases.



Table in the database, a double click will show the columns

By dragging a table node and dropping it to the list on the right side, a new query will be generated automatically. Each column of the table will be assigned to a %-Code whose name is built from the initials of data source, table and column name. The leftmost column will be treated as index field the following ones as retrieval fields.



Field or column in this table.

Data Connections

The branch of the data connections initially contains only one placeholder for a new data connection. If one or more connections are already established, they are displayed here.

Here these symbols are used:



Data connection, defined by a ConnectionString.



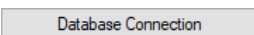
Points to a SELECT-Command, which queries data from a database



Points to an item in the SELECT clause of the command.



Points to an item in the SELECT clause of the command too, which is assigned to a program variable, a %-Code.



Under the tree view is a button for editing the selected database connection or creating a new connection.

Righth Side

The table on the right shows the fields of a table used in a database query.

The first column specifies the access type and %-Code associated with the database field. The further columns correspond to the control elements in the middle.

For Data Sources:

All fields in the table that are not explicitly deleted by the operator are displayed. The symbol before the %-Code indicates how the variable is used.



Index field: All records will be found which matches the index fields.



Retrieval-Field, the content of this field will be queried from the database.



Unknown field, which is part of the table, but is neither searched for nor returned.

With the help of the controls in the middle, the properties of the fields can be changed, even for several fields at the same time.

For Data Connections

Only the fields that are returned for a query are listed here. So there are no index fields in this list, however, any %-Code can be specified in the WHERE clause of the SQL command.



Identifier from the SELECT clause for which a %-Code is specified.



Identifier from the SELECT clause for which will not be sent to the program.


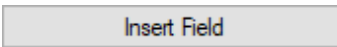
For this field types the controls in the middle will not take effect.

Further Control Elements

The controls in the middle of the dialog only work if a table is selected on the left or one or more variables are marked on the right. Their properties can be changed here.

Access Type	Defines how to handle the data exchange for the variable:
Index	All records will be found which matches the index
Retrieval	Retrieval-Field, the content of this field will be que database
Unknown (not of interest)	Will be ignored during database access.
Exact Match	The database be queried with an exact match to the index field or with a similar match. For the non-exact match placeholders can be used:
*	A star, or asterisk, stands for several characters
?	A question mark stands for one character exactly
Data Source	DSN (Data Source Name) as defined in the system configuration. New data sources can be created with a system tool ODBC-manager.
Table	Name of the table in the database.
Field	Name of the field in the table.
Scope	This is the variable's scope, that is the set of things a variable belongs to: a batch, a folder, a file,...
Variable	Name of the variable in the program, also called %-Code, the nearby button % opens a dialog box to choose a variable.

Using the control elements in the middle these properties can be changed:

	If a table is selected in the tree view, this button copies all columns of the table to the right and automatically assigns them variable names. The name consists of the name of the data source, the table, and the field name. However, it can be changed as desired.
	If a field is selected in the tree view it will be added to the list of variables on right side.

If changes were made in the middle of the dialog, these values will be combined to create a new variable in the list to the right.

Replace	If changes were made, these will be assigned to the selected line at the right
Remove	Releases the assignment of this field to the program variable. Removes the line from the list.
Reset	Discards all changes and loads the subprofile again.
Query Result Options	<p>Opens a dialog box to define the behavior when</p> <ul style="list-style-type: none"> • there is no result or • more than one result or • exact one result in the database.
Create Index Mask	<p>Creates a UDD (user-defined dialog) that displays the name of the database field in the index mask and associates the associated input field with the variable. The index fields are automatically sorted upwards and the background colors are assigned to yellow (for unprocessed) and light yellow (for processed). The retrieval fields are attached to the background colors green and light green.</p> <p>This UDD can be changed at any time via the class settings. If the ODBC PlugIn is loaded in the base profile, the connection to the data source can also be changed there. It is therefore not necessary to create a new index mask for each change.</p>

OK, Cancel, Apply, Help These buttons will work in the well known manner.

See also:

[Usage ODBC](#)
[Configuration ODBC](#)
[Variable Assignment](#)
[Viewing Results](#)

1.2.1 Configuration Data Connection

This dialog is used to set up a connection to an SQL database. To do this, you first specify a `ConnectionString`, which connects to a database service, a so called provider. In most cases, you specify who wants to access where the which database. Please make also sure that the correct provider is installed and access to the database you are looking for is allowed. This information can be provided by the administrator of the database.

When the connection is established, SQL commands can now be specified to retrieve or modify the data in the database. The commands can contain %-Codes that are converted to the current value before execution.

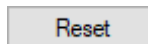
If the commands are query commands, "SELECT ... FROM ... WHERE ...", the returned values of the fields can be copied in percent codes.

You can specify how to proceed if there are no hits in the database or more than one.

These controls can be used:

Connection String Field for entering a ConnectionStrings. Successful connections are saved in the

drop-down list underneath.



Clears the field with the ConnctionString and its history.

SQL Command

Processing field for the command. %-Codes in the SQL command are replaced before execution.

List of SQL commands. The commands are executed in this order from top down.



Selects an SQL command to edit and copy it to the edit line.



Inserts the contents of the edit line as a new SQL command in the list.



Copies a modified SQL command back to the list. If possible, all returned fields are determined and suitable variables suggested.



Removes the selected SQL commands.



Removes all SQL commands.



Moves the selected command one line up or down resp.

Variables

List of assignments between the field in the database and the variable in the program (% code)



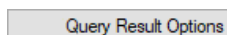
Opens the selection dialog for program variables to assign a % code to the selected field.



Solves the association between table field and %-Codes.



Solves all associations.

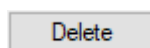


Opens a dialog to the behavior when it is at the query in the database

- no hits,
- or several hits
- or exactly one hit



Executes all SQL-Commands



Deletes the data connection and closes the dialog.

The further buttons will work in the known manner.

See also:

[Usage ODBC](#)

[Configuration ODBC](#)

[Variable Assignment](#)

[Viewing Results](#)

1.2.2 Configuration Query Result Options

When the data is transferred to the program, it is assumed that exactly one value is assigned to each %-Code.

In this dialog the behavior can be adjusted after querying the database.

Multiple Results Sets the behavior if there are multiple hits in the database during the query:

- Take first hit and cancel further search.
- Provide a list of the records found
-

The selection is made by double-clicking the desired line.

If the selection is aborted, the program behaves as if there were no hits.

One Result Sets the behavior if there is exactly one hit in the database:

- simply use the found values
- show and offer cancellation
-

If the selection is aborted, the program behaves as if there was no hit.

No Result Sets the behavior if there is no hit in the database during the query:

- Delete all fields that should be filled by the query (default)
- do nothing
- Delete all fields.
- Delete all fields that were compared during the query

Deleting the fields means to set the content of the %-Code to an empty string.

Allow Cancellation If this option is enabled, a cancel button will be offered when choosing a result from a list.

The further buttons will work in the known manner.

See also:

[Usage ODBC](#)

[Configuration ODBC](#)

[Variable Assignment](#)

[Viewing Results](#)

1.2.3 Configuration Broker

In this dialog, you can specify the event to run the query automatically. Please see also the [sequence of broker events](#).

All Chooses all events.

At Task Start [09] The event is sent at the beginning of task, this event always occurs before any other event.

The task start occurs when the Start Task button was pressed in the toolbar or the corresponding menu item is selected .

With Batch Creation [11] The event is sent after a batch is created. But it will *not be sent* if an already existing batch will be opened again.

Witch Batch Opening [16]	<p>The defined event is sent after a batch was opened</p> <p>This event is sent after a batch was opened. It doesn't matter whether the batch is new or does already contain images.</p>
Before Batch Export [15]	<p>This event will be sent immediately before exporting an OpenJob. This event occurs when the Export button is pressed or a corresponding command is executed in the Task or in a macro. Please see also the sequence of broker events.</p>
Before Batch Finalization [13]	<p>This event will be sent immediately before finalizing an OpenJob. This event occurs when the Finalize button was pressed. It occurs too if a corresponding command is executed in the Task or in a macro. Please see also the sequence of broker events.</p>
With Path creation [09]	<p>The event is sent when you create a new path. If a new path is created, a new file is generated inevitably and the new image is stored there.</p>
With the first image group [01]	<p>This event is sent to the broker only once during the first scan. Here, you can send title information like batch name, station name, start time, or any desired start text.</p>
With document creation [21]	<p>This event is sent when a document is created. A new document is created when</p> <ul style="list-style-type: none"> • The "Set document level" action is executed in the event rules. <p>Opening and closing is independent of the directory structure, a document can be spread over several files and folders!</p>
With File Creation [05]	<p>The defined event is sent after a new file is created and before the first image is stored in the new file.</p>
For Every Image Group [02]	<p>This event is sent to the broker after every scan for front side and rear side separately. But there is only one call for all images of one side (filtered copies, cut-outs,...). It is used to place information which are important for one side.</p>
As Event Rule [04]	<p>This event will only be sent to the Broker when an event has occurred, for that an event mark shall be sent. The related event must have been defined in the Event Rules, the action must be "Broker / Batchfile message".</p> <p>This event will be fired in DirectMode only!</p>
With File Closing [06]	<p>The defined line is sent after a closing a file, for example a Multi-TIF file. A file will be closed when</p> <ul style="list-style-type: none"> • opening a new file, • opening a new folder or • finalizing a batch.
With document closing [22]	<p>This event is sent when a document is closed. An open document will be closed if</p> <ul style="list-style-type: none"> • The "Set document level" action is executed in the event rules. • The batch is finalized <p>Opening and closing is therefore independent of the directory structure; a document can span several files and folders!</p>

After The Last Image Group [03]	This event is sent to the broker file only after the last scan. This gives the possibility to write defined end lines.
With Path Closing [08]	The event is sent if a path is closed. This occurs if <ul style="list-style-type: none"> • a new subdirectory is created • or the batch gets finalized.
With Batch Closing [17]	This event is sent after a the last path was closed.
After Batch Export [14]]	The defined event is sent after the export of an OpenJob, i.e. the batch was copied. Please see also the sequence of broker events .
Error On Batch Export [18]	The defined event will be sent to PlugIns with a Broker interface only. It will be send if the operating system signals an error at the batch export, e.g. disk full, network drive not found, insufficient rights,...
After Batch Finalization [12]	The defined event is sent after the finalization of an OpenJob. Please see also the sequence of broker events .
Error On Batch Finalization [19]	This event will be sent to PlugIns with a broker interface only. It will be send if the operating system signals an error at the batch finalization.
At Task End [10]	The defined event is sent at end of task. In DirectMode the Task End is reached when <ul style="list-style-type: none"> • the last image was written to a file • the last file was closed • the last folder was closed • the batch was closed. In OpenJob the Task End is reached when a batch was closed, exported or finalized. So here "At Task End" can occur more then once.
At Task Final [20]	This event is fired when all commands of the task list are processed. So it occurs always as the last event.

See also:

[Configuration ODBC](#)
[Sequence of the broker events](#).

1.2.3.1 Sequence of Broker Events

The sequence in which the events occur and their numbering have been constantly expanded in the course of the program development. For reasons of compatibility, however, the events have neither been renamed nor renumbered. Therefore, there are some special issues:

In DirectMode

At begin of the task

```
[09] At task Start
[11]     With batch creation
[16]     With batch opening
[07]         With path creation
```

```
[01]          With the first image
[22]          With document creation
[05]          With file creation
[02]          With every image group
[02]          With every image group
[04]          As event rule
[02]          With every image group
```

please keep in mind the event "As event rule" is available in **DirectMode** only!

at end of the task

```
[02]          With every image group
[02]          With every image group
[02]          With every image group
[06]          With file closing
[03]          With the last image
[08]          With path closing
[17]          With batch closing
[10] At task End
[12] After batch finalization
[20] At task final
```

In OpenJob-Mode

A regular scan task with closing the batch at the end will raise these events:

```
[09] At task Start
[11]    With batch creation
[16]    With batch opening
[17]    With batch closing
[10] At task End
[20] At task final
```

The following opening of the batch looks like this:

```
[09] At task Start
[16]    With batch opening
[17]    With batch closing
[10] At task End
[20] At task final
```

It doesn't matter whether the batch contains images or not.

The **Finalization** of a paused batch shows these events at begin:

```
[13] Before batch finalization
[07]    With path creation
[01]    With the first image
[22]    With document creation
[05]    With file creation
[02]    With every image group
[02]    With every image group
```

[02] With every image group

and at end:

[02] With every image group
[02] With every image group
[06] With file closing
[23] With document closing
[03] With the last image
[08] With path closing
[17] With batch closing
[10] At task End
[12] After batch finalization
[20] At task final

The **Export** of a paused batch raises these events:

[15] Before batch export
[17] With batch closing
[10] At task End
[14] After batch export
[20] At task final

If the combination of **Export & Finalization** will show this at the begin:

[15] Before batch export
[17] With batch closing
[10] At task End
[13] Before batch finalization
[07] With path creation
[01] With the first image
[22] With document creation
[05] With file creation
[02] With every image group
[02] With every image group
[02] With every image group

and at the end:

[02] With every image group
[02] With every image group
[02] With every image group
[06] With file closing
[23] With document closing
[03] With the last image
[08] With path closing
[17] With batch closing
[10] At task End
[12] After batch finalization
[14] After batch export
[20] At task final

So at the combination **Export & Finalization** the events "With batch closing" and " At task End" will be fired twice.

See also [Broker Settings](#).

1.3 Variable Assignment

This dialog box comes up when in a userdefined dialog UDD a variable should be assigned to a field in the database. From top it shows the following entry fields.

Variable	Name of the variable as %-Code in the program where it can be used for display, output,...						
Scope	This is the variable's scope, that is the set of things a variable belongs to: a batch, a folder, a file,...						
Access Type	Defines how to handle the data exchange for the variable: <table><tr><td>Index</td><td>All records will be found which matches the index fields.</td></tr><tr><td>Retrieval</td><td>Retrieval-Field, the content of this field will be queried from the database</td></tr><tr><td>Unknown (not of interest)</td><td>Will be ignored during database access.</td></tr></table>	Index	All records will be found which matches the index fields.	Retrieval	Retrieval-Field, the content of this field will be queried from the database	Unknown (not of interest)	Will be ignored during database access.
Index	All records will be found which matches the index fields.						
Retrieval	Retrieval-Field, the content of this field will be queried from the database						
Unknown (not of interest)	Will be ignored during database access.						
Match phrase exactly	The database be queried with an exact match to the index field or with a similar match. For the non-exact match placeholders can be used: <table><tr><td>*</td><td>A star, or asterisk, stands for several characters</td></tr><tr><td>?</td><td>A question mark stands for one character exactly.</td></tr></table>	*	A star, or asterisk, stands for several characters	?	A question mark stands for one character exactly.		
*	A star, or asterisk, stands for several characters						
?	A question mark stands for one character exactly.						
Database Source Names	DSN (Data Source Name) as defined in the system configuration. New data sources can be created with a system tool ODBC-manager.						
Table	Name of the table in the database.						
Field	Name of the field in the table.						

At the bottom of the dialog there are some buttons:

Release Assignment	Closes the dialog and set fields DSN, table and field to an empty string and the access type to "Unknown"
OK, Cancel, Help	Will work as usual.

See also:

[Usage ODBC](#)
[Configuration ODBC](#)
[Variable Assignment](#)
[Viewing Results](#)

1.4 Viewing Results

This dialog displays the found records in a table.
Double-clicking on a row selects a record and closes the dialog.

If there is a Cancel button, the dialog is closed and the program behaves as if there was no hit in the database.

See also:

[Usage ODBC](#)

[Configuration ODBC](#)

[Variable Assignment](#)

[Viewing Results](#)

1.5 Version

Open Database Connection
Help for PlugIn JK_ODBC
JK_ODBC_09 , Version 6.11.1254

JK_ODBC_09.pdf 5/10/2021, 11:50 AM
AK, Janich & Klass Computertechnik GmbH

Wuppertal, 5/10/2021, 11:50 AM

[Updates](#)

Index

- " -

"SELECT ... FROM ... WHERE ..." 8

- % -

%-Code 5

%-Codes 4

- A -

access authorization 4

access to a data source 4

Access Type 5, 15

actual data sources 5

administrator 4

Allow Cancellation 10

- B -

base profile 4

Batch Closing 10

Batch Creation 10

Batch Export 10

Batch Finalization 10

Batch Opening 10

begin of the task 12

Behavior 5

Broker Mode 5

- C -

Cancel button 16

Configuration 5

Configuration Broker 10

Configuration Data Connection 8

Configuration Query Result Options 10

ConnectionString 5, 8

copy all columns 5

Create index mask 5

- D -

Data Connections 5

Data Source 4, 5

Data Sources 5

Database Provider 4

Database Source Names 15

defined interface 4

DirectMode 12

document closing 10

document creation 10

Double-clicking 16

- E -

end of the task 12

Error On Batch Export 10

Error On Batch Finalization 10

Event Rule 10

event to run the query automatically 10

Every Image 10

Exact Match 5

Export 12

Export & Finalization 12

- F -

Field 5, 15

Field or column in a table 5

File Closing 10

File Creation 10

Finalization 12

first image 10

- I -

Index field 5

Interactive mode 5

- L -

Last Image 10

Licensing 4

loaded 5

- M -

Macro 5

Match exactly 15

Match phrase exactly 15

Multiple Results 10

- N -

No Result 10

- O -

Object Linking and Embedding 4
 ODBC data source 4, 5
 ODBC Driver 4
 ODBC PlugIn 4
 OLEDB data connection 4
 One Result 10
 Open Database Connectivity 4
 OpenJob-Mode 12

- P -

Path Closing 10
 Path creation 10
 PlugIns 4
 prerequisite 4
 Process mode 5

- R -

Release Assignment 15
 Retrieval-Field 5

- S -

Scope 5, 15
 Script-PlugIn 4
 SELECT clause 5
 SELECT-Command 5
 Sequence of Broker Events 12
 software module 4
 SQL Command 8
 SQL database 8
 suitable interface software 4
 suitable key 4

- T -

Table 5, 15
 Table in the database 5
 Task End 10
 Task Final 10
 task profile 4

Task Start 10
 Toolbar command 5

- U -

UDD 15
 Unknown field 5
 used 5

- V -

Variable 5, 15
 Variable Assignment 15
 Variables 8
 Version 16
 Viewing Results 16