I) ENGLISH

II) GERMAN

I) ENGLISH

Data exchange of patient or image data between DBSWIN and a practice administration software

1.Handing over of patient data from a practice administration software to DBSWIN

- Handing over is carried out via a text file (in the following referenced as ‚import file') which is produced by the practice administration software. Path and name of the file that is scanned for patient data can be configured arbitrarily in DBSWIN. (Filename could be 'patdata.txt', for example)
- The import file is generated by the practice administration software (or replaced, if already existing) for example by clicking a button or automatically, when a patient is selected. Consult the manufacturer of your practice administration software to learn, how it can generate such a file.
- the first line of the file must contain patient data in the following format (order is essential), *italic* entries are optional:

PATLASTNAME;PATFIRSTNAME;PATBIRTHDAY;*PATCARDNUMBER;PATTOWN;PATSTREET;PATPHONENUMBER;PATTITLE; PATSEX;PATPOSTALCODE;*

When the first letter of PATSEX is 'f' or 'F', the patient's sex is set to 'female', for 'm' or 'M' to 'male', else unknown

e.g.

Smith;Brian;15.1.1996;23;Smalltown;Smallstreet 33;01234/5678;Prof. Dr.;m;76543;

- All fields except PATBIRTHDAY are non-empty character sequences, each field MUST be terminated by semicolon (also the last one)
- *PATCARDNUMBER* is optional, but should be used to prevent data confusion
- If a data field is not available, but you desire to hand over following fields, you must nevertheless end the missing field with semicolon,

e.g . when the street is not available:

Smith;Brian;15.1.1996;23;Smalltown;;01234/5678;Prof. Dr.;male;76543;

- PATBIRTHDAY must be given in the format day.month.year
- only one patient at a time can be handed over

1.1 Settings and operating of DBSWIN

In DBSWIN, the import file must be specified as described in the following:

- configure your practice administration software to generate the import file, formated as requested above.
- Start DBSWIN and Login as administrator, select in menu ‚Options' the item 'Show Configuration'
- Select in module ‚Config.' the sheet ‚Modules' and open the symbol 'Patient' by double clicking
- Check the checkbox ‚Show button for patient data import from file' (now must be marked)
- Push the button for file selection and select the import file in the dialog which now opens, **or** enter the full path and name of the file in the edit field
- then press OK
- in DBSWIN, you see a button in the patient module that starts the data input from the import file. If a file is present, the button will be marked red.
- pushing the button reads the data into DBSWIN. The patient is selected, if already present in DBSWIN.
- If not, mode ‚New Patient' is entered.
- If a patient exists with same name, but other ‚Card Number', the ‚Edit Mode' is entered
- In both cases, you have to push ‚Save' to save to database
- Automatic mode: If you wish to automatically select a patient and switch to Viewer as soon as import file is present please edit the file 'bin\Ldbs.ini'. Enter/change in Section [Patient] the line: 'AutoActivatePatient=1'. normally set to 0)

2. Handing over of images from DBSWIN to an administration software

DBSWIN offers an interface for image export. If the administration software is capable to use the interface, it is possible to hand over images and additional image information to the administration software. To learn if this is the case, please contact your administration software dealer.

In the following is explained how the interface is to be used (information for programmers of practice administration software) .

General:
This functionality is implemented by a DLL "dbsExpW.dll" located in windows-system directory (see also section availability). It provides functions FindFirst, FindFirstEx, FindNext, FindClose, ImgToClipbrd (can be slow), ThumbnailImgToClipbrd (fast) , with which you can get images or preview (thumbnail) images of a specific patient and image related information.
For referencing a patient, the 'card number', the patients first and last name and the birthday can be used. This data is shown in the card form of DBSWIN. Search goes over all practices in DBSWIN.

----------!!IMPORTANT!! ----------
DBSWIN should be configured with patient data import from administration program *activated*, imported data should include 'card number' (see above). If only the card number is used to identify a patient (function FindFirst), you must make sure, that a patient has the same number in all practices of DBSWIN and the administration program! Else, if a 'card number' is associated with differnet patients in different DBSWIN-practices, name and birthday of the patient can be used for identification (function FindFirstEx). Then this data must be synchronized in DBSWIN and administration software.

Otherwise images related to another patient could be returned!
----------!!IMPORTANT!! ----------

Demo:
There is program demo to see the usage of the DLL and to test image export on DBWIN - CD in folder Tools\DatExp. There's also an example in Delphi and Visual basic, and a function declaration for C(++)

Functions of the DLL:
Following are the prototypes in pascal syntax and some parameter explanations.
'Pchar' are pointers to zero terminated strings, integer are 4 byte.
Calling convention: **stdcall** (This means the order of the parameters on the stack is from right to left and the stacked is cleared by the routine, not the caller).
A *procedure* corresponds in C to a function with return type *void*
The buffer for result strings must be **allocated by the caller** in the below defined sizes

Parameter explanations.:
PatKNR : maximum 15 [chars] (+ terminating 0) This is the 'card number'.
PatFirstName, PatLastName: Firstname, lastname of a patient. Maximum 48 [chars]
Annotation: for the fields PatKNR, PatFirstName, PatLastName: Leading and ending spaces should be eliminated, because they are cut by DBSWIN if present. The fields are not case sensitive. (as a result, the card number ' 2' and '2' are treated as the same, as well as 'charles', 'Charles' and ' charles ')
PatBirthday: must be given in the format 'day.month.year'. Year should be 4 digits, if year < 100, then 1900 is added. Separator must be '.'
ImgID: 24 *[chars]*
ImgTyp: 24 *[chars]* The image type, a string that describes what the image shows, displayed in DBSWIN as 'Object type', e.g. '11' for an image of tooth 1 in quadrant 1
ImgDiag: 100 *[chars]* Diagnosis of an image
ImgCreateDate: 9 *[chars]* Date on which the image was created, in format yyyymmdd
Remark: The function **ImgInfo** provides the image creation **time** also

function FindFirst (PatKNR: PChar;
ImgID, ImgTyp, ImgDiag, ImgCreateDate: Pchar): integer; stdcall;

!!This function shouldn't be used anymore! It is replaced by FindFirstEx!

Input: PatKNR: Card number of the patient for which you want to find images.
Result: handle>0, which can be used to continue the search with FindNext or as parameter for FindClose. Handles loose their validity when the DLL is freed or when FindClose is called for a handle!
If handle <=0: an error has occured.
Output: ImgID. This is the ID or the first image found. With this ID the function ImgToClipbrd can be parametrized.
ImgTyp: Image type of this image
ImgDiag: Diagnosis
ImgCreateDate:Creation date of the image in format yyyymmdd
If no image was found: ImgID='' (empty string)
The search for images of a given PatKNR includes *all practices* defined in DBSWIN

function FindFirstEx (PatKNR, PatFirstName, PatLastName, PatBirthday : PChar;
ImgID, ImgTyp, ImgDiag, ImgCreateDate: Pchar): integer; stdcall;
Input: PatKNR, PatFirstName, PatLastName, PatBirthday: To identify the patient for which you want to find images. Some values can be left blank (empty string ''), this means 'don't care'.
It is looped over **all** DBSWIN-practices, for every practice, the first patient qualifies which matches all non-empty values. The ImgIDs of all qualified patients are returned in this and the following FindNext calls.

Result: handle>0, which can be used to continue the search with FindNext or as parameter for FindClose. Handles loose their validity when the DLL is freed or when FindClose is called for a handle!
If handle <=0: an error has occured.
Output: ImgID. This is the ID or the first image found. With this ID the function ImgToClipbrd can be parametrized.
ImgTyp: Image type of this image
ImgDiag: Diagnosis
ImgCreateDate:Creation date of the image in format yyyymmdd
If no image was found: ImgID='' (empty string)
The search for images of a given PatKNR includes *all practices* defined in DBSWIN

procedure FindNext (handle: integer;
ImgID, ImgTyp, ImgDiag, ImgCreateDate: Pchar) ; stdcall;
Input: handle. Allowed are only handles which you received by FindFirst/FindFirstEx before.
Output: ImgID. This is the ID or the first image found. With this ID the function ImgToClipbrd can be parametrized.
ImgTyp: Image type of this image
ImgDiag: Diagnosis
ImgCreateDate:Creation date of the image in format yyyymmdd
ImgID='' (empty string) if no more image was found

function ImgInfo (handle: integer;
ImgID, whichInfo: Pchar;
buflen: integer; outp: Pchar): integer; stdcall;
This function can provide additional image info as string.
Input:
handle. search handle of the currently active FindFirst/FindFirstEx/FindNext loop. Handle must still be opened!
ImgID. This is the ID of the image for which info is queried. **It must be an ID that was returned in the current FindFirst/Next search loop!**
whichinfo: Pchar. Specifies, which kind of info you want. At the moment, only 'CreateDateTime' is recognized and returns in <outp> the creation date and time of the image in the format yyyymmddhhnnss (n=minute of creation, s=second of creation)

buflen: the length of the buffer **that the caller must provide**
outp: pointer to the buffer **that the caller must provide**

Output: Up to <buflen> characters are copied to <outp>

result: if result > 0: Success
else: Failure in parametrization or imageID not found in current FindFirst/Next search

procedure FindClose (handle: integer); stdcall;
Eingabe: handle. Allowed are only handles which you received by FindFirst/FindFirstEx before.
For every FindFirst/FindFirstEx, there should be called exactly one FindClose.

function ThumbnailImgToClipbrd (ImgID: pchar): integer; stdcall;
Input: ImgID. Allowed are only handles which you received by FindFirst/FindFirstEx any time before.
The function needs not necessarily be called within a FindFirst/Next/Close sequence.
result: if result > 0: A low resolution preview (thumbnail-) image for this ImgID is in the clipboard in 'Windows Bitmap format'. Functions **works fast** and should be used to present the user an overview of all available images of a patient.
else: No data in clipboard, failure. Possible reasons: it was deleted or is only accessible in long time archive, the practice was deleted in DBSWIN or an error occured

function ImgToClipbrd (ImgID: pchar): integer; stdcall;
Input: ImgID. Allowed are only handles which you received by FindFirst/FindFirstEx any time before.
The function needs not necessarily be called within a FindFirst/Next/Close sequence.
result: if result > 0: Full image data associated with 'ImgID' is in the clipboard in 'Windows Bitmap format'.
Function is slow especially for high resolution scanner images! Should only be used, if the user wants to examine/diagnose an image.
else: No image data in clipboard. Possible reasons: it was deleted or is only accessible in long time archive, the practice was deleted in DBSWIN or an error occured

Typical usage:
You start a search with FindFirst or FindFirstEx, get a handle and, if an image was found, an ImageID and additional data. You can get additional info by passing handle and ImageID to function 'ImgInfo'. If ImgID <> '', you can pass the handle to FindNext and get next data set, else there are no more images for this search. Close the search with FindClose to free associated memory objects! You can store the received ImageIDs and pass them to 'ThumbnailImgToClipbrd' to present the user an image overview. When the user decides to examine an image (for example he double clicks the preview image), you use 'ImgToClipbrd' to get full data.
The library can be loaded with Loadlibrary and freed with Freelibrary. Please carry out Freelibrary only if your search is finished because handles loose their validity when the DLL is removed from memory.

Errors, problems, troubleshooting:
- If problems occur while executing the DLL-functions, they are shown in DBSWIN-Logger. The logger appears as icon in the task bar left of the system clock. Double click it to see a window, you can activate 'Show all' to see all messages. Messages are also stored to file automatically (directory DBSWIN\bin\log*.rtf).
- Because of system restrictions there may occur problems for some programs using the DLL.
- (on Win32 plattforms, the number of DLLs that a process can open at one time is restricted. If the number is exceeded, errors occur, see article Q193462 Microsoft Knowledge base)
- Symptoms:
- - when executing LoadLibrary a message like 'error in the initialization routine of a library' is shown **or**
- - when executing FindFirst, in DBSWIN-Logger a message is shown like 'An error occurred while attempting to initialize the Borland Database Engine (error $2519)'. **or**
- - when executing ImgToClipbrd, you get a return value of 0 and a message in DBSWIN-Logger appears like '...error 489 can't find file format library module'

- if dbsexpw.DLL is loaded, there is shown an error like "Error by load Library 1157". The cause may be that the DLL fails to load some other DLL from the DBSWIN\bin directory. (see previous)
- The dbsexpw.DLL searches the current DBSWIN Installation via **Inifile 'dbsExpW.ini'** located in the windows directory (NOT windows\system). The Inifile is setup by DBSWIN-Installation
- Format (example):
- [Setup]
- DBSWINBinPath=C:\dbswin\bin
- If this directory doesn't exist, loading fails.

- **Problem with database on Novellserver**, this occured on Novellserver V5.1, NovellClient V3.21 and Win98 workstation: if DBSWIN is running and then the interface is called or vice versa, DBSWIN and the interface keeps hanging for more than a minute (Error in DBSLog: "Timeout bei Sperren der Datei PDOXUSRS.NET" or " Network initialization failed. Lock time out. File: PDOXUSRS.NET")

Solutions:
- install Novell Client V3.30 (or newest version)
- install the newest service pack for the server
- at the server console "set client file caching enabled=off"

Availabilty:
Functionality is not available in all versions of DBSWIN. To see if your version supports it, look for the file ExpService.dll in the directory DBSWIN\bin. Sometimes the explorer has to be configurated to display DLL files.
The FindFirstEx function is available only in versions >=2.1.9 (Nov 2000).
Please make sure you are using the newest version of DBSWIN.

Performance aspects:
- Some operations are more time consuming, and may lead to a delay. (Often delays shorten when executing an operation a second time, because data is then already cached)
- To reduce delays to a minimum, access to the services should be optimized accordingly (e.g. z.B. get *all* image IDs for a patient in *one* Find-Sequence, do not request image data unnecessarily often)
- **You mustn't call full image data** for a patient every time he is activated. This is too time consuming, use preview data (function ThumbnailImgToClipbrd) instead!

Following is a listing, which operations are most time consumimg and what other operations they trigger:
- **LoadLibrary**: the DLL and other DLLs needed for image operations are loaded, the logger is started (if not active).
- **FindFirst, FindFirstEx**: On first call, the database engine is loaded and initialized, this doesn't happen again as long as the DLL isn't freed.
  All requested ImgIDs are collected. This means that following FindNexts do *not* need database accesses, therefore run faster. This advantage remains until FindClose is executed
- **ThumbnailImgToClipbrd**: Thumbnail data of the image in low resolution are fast available and put to clipboard
- **ImgToClipbrd**: The image file is searched via database, loaded from harddisk to memory and transfered to clipboard. This function is especially for high resolution scanner images very time consuming!

DEUTSCH

II) Datenaustausch von Patienten- oder Bilddaten von DBSWIN mit einer Praxisverwaltung

1.Patientendatenübergabe von der Praxisverwaltung zu DBSWIN über Textdatei

- Übergabe erfolgt in einer Textdatei, die von der Praxisverwaltungssoftware erzeugt wird. Der Pfad und Name der Datei ist von DBSWIN aus beliebig konfigurierbar.(Der Name kann z.B. 'patdata.txt' lauten)
- Die Datei wird von der Praxisverwaltungssoftware erzeugt (oder wenn sie schon besteht, dann ersetzt) z.B. durch Drücken eines bestimmten Knopfs oder automatisch beim Anwählen eines Patienten. Wenden Sie sich an den Hersteller Ihrer Praxisverwaltungssoftware, um zu erfahren, wie eine solche Datei erzeugt werden kann.
- Die erste Zeile muß die Patientendaten in folgendem Format (Reihenfolge ist wichtig) enthalten, *kursive* Einträge sind optional, d.h. können auch weggelassen werden:

PATNACHNAME;PATVORNAME;PATGEBDATUM;*PATKARTEINUMMER*;*PATWOHNORT;PATSTRASSE;PATTELEFONNUMMER;PATTITEL;PATSEX;PATPOSTLEITZAHL;*

Ist der erste Buchstabe von PATSEX 'f' oder 'F', wird das Patientengeschlecht auf 'weiblich' gestellt, falls 'm' oder 'M' auf 'männlich', sonst auf 'unbekannt'

Z.B.

- Schulz;Karl-Eugen;15.1.1996;23;Kleinstadt;Adenauerring 88;0123/232323;Dr. h.c.;m;98765;
- Alle Felder außer PATGEBDATUM sind beliebige (nichtleere) Zeichenfolgen, jedes Feld (auch das letzte) muß von einem Semikolon abgeschlossen werden.
- Obwohl *PATKARTEINUMMER* optional ist, sollte es übergeben werden (um Verwechslungen bei Namensänderungen von Patienten zu vermeiden) .
- Wenn ein Datenfeld nicht verfügbar ist, aber darauffolgende Felder übergeben werden sollen, muß trotzdem das fehlende Feld mit Semikolon abgeschlossen werden,

Z.B. eine Übergabe ohne Straße:

Schulz;Karl-Eugen;15.1.1996;23;Kleinstadt;;0123/232323;Dr. h.c.;m;98765;

- Geburtsdatum muß im deutschen Format (tag.monat.jahr) sein.
- Es kann immer nur ein Patient auf einmal übergeben werden

1.1 Einstellungen und Bedienung in DBSWIN

In DBSWIN muß der Dateipfad für die Übergabedatei eingestellt werden. Gehen Sie bitte folgendermaßen vor:

- Führen Sie in Ihrer Praxisverwaltungssoftware die notwendigen Schritte aus, um eine Übergabedatei zu erzeugen.
- In DBSWIN melden Sie sich als Administrator an und wählen im Menü 'Optionen' den Punkt 'Konfiguration anzeigen'
- Wechseln Sie (in der Registerkarte 'Konfig') auf die Registerkarte 'Module' und aktivieren das Symbol 'Patient' durch Doppelklick
- Schalten Sie das Kontrollkästchen 'Anzeige des Knopfs für Patientenimport von Datei' ein (sofern es nicht eingeschaltet ist)
- Betätigen Sie die Schaltfläche für Dateiauswahl und wählen Sie im folgenden Dialog die Übergabedatei aus **oder** tragen Sie den kompletten Pfad und Name manuell ins nebenstehende Textfeld ein
- Bestätigen Sie mit OK
- In DBSWIN sehen Sie nun in der Patientenkartei einen Knopf, der die Datenübernahme von Datei auslöst. Falls eine Übernahmedatei vorhanden ist, hat er einen roten Rahmen
- Bei Betätigung des Knopfes werden die Daten übernommen und wenn der Patient schon vorhanden ist, ausgewählt.
- Ist ein Patient mit gleichen Daten noch nicht vorhanden, wird in den 'Patient neu anlegen' Modus geschaltet.
- Falls ein Patient mit gleichem Namen aber anderer Karteinummer schon existiert, wird in den 'Patient bearbeiten' Modus geschaltet.
- In beiden Fällen müssen Sie jeweils noch die Schaltfläche 'Speichern' drücken, um die Änderungen in die Datenbank zu übernehmen
- Automatikmodus: Wenn Sie wünschen, daß bei Vorhandensein von Übergabedaten automatisch der Patient angewählt wird und auf den Lichttisch gewechselt wird, tragen Sie bitte in der Datei 'bin\Ldbs.ini' in der Sektion [Patient] folgendes ein/ändern Sie die Zeile zu: 'AutoActivatePatient=1'. (Schalter steht normalerweise auf 0)

Falls ein Patient mit gleichem Namen aber anderer Karteinummer schon besteht, wird in den 'Patient bearbeiten'

Modus geschaltet.
Sie müssen jeweils noch die Schaltfläche 'Speichern' drücken, um die Änderungen in die Datenbank zu übernehmen

2. Datenaustausch mit der VDDS-Media Schnittstelle

Möglich ist derzeit Patientendatenübernahme per VVDS und die Übergabe von Leistungsdaten (Bildinformationen, Diagnose,..) an ein Praxisverwaltung. Dies sind die VDDS-Implementierungsstufen 1+2.

Aktivierung von VDDS:
* Schalten Sie im Menü 'Optionen' die Konfiguration ein.
* In der Konfiguration wechseln sie auf 'Module' und doppelklicken Sie auf DBSWIN.
* Auf dem Reiter VDDS gibt es die Möglichkeit zum Aktivieren der VDDS-Schnittstelle
* Die Praxisverwaltung muß entsprechend den Anweisungen des Herstellers eingestellt werden

3. Bilddatenübergabe von DBSWIN an eine Praxisverwaltung

DBSWIN bietet eine Schnittstelle für den Bildexport. Sofern das verwendete Praxisverwaltungsprogramm diese nutzen kann, ist es möglich, Bilder in vollem Datenumfang (kann länger dauern) oder Vorschaubilder in geringer Auflösung (geht schnell), sowie Bildzusatzinformationen aus DBSWIN an das Praxisverwaltungsprogramm zu übertragen. Um zu erfahren, ob dies der Fall ist, fragen Sie bitte den Hersteller Ihres Praxisverwaltungsprogramms.

Im folgenden wird erklärt, wie man diese Schnittstelle in ein Programm einbinden kann (Information für Hersteller von Praxisverwaltungsprogrammen).

Allgemeines:
Realisierung erfolgt durch die DLL "dbsExpW.dll" im Windows-Systemverzeichnis (siehe auch Abschnitt 'Verfügbarkeit'). Diese stellt unten näher beschriebene Funktionen FindFirst, FindFirstEx, FindNext, FindClose, ImgToClipbrd, ThumbnailImgToClipbrd zur Verfügung, mit denen man patientenbezogen Bilder und Bildzusatzinformationen erhalten kann.
Die Funktion ThumbnailImgToClipbrd eignet sich gut, um dem Benutzer **schnell** einen Überblick über die verfügbaren Bilder zu präsentieren. Erst, wenn der Benutzer ein Bild betrachten will, sollte man dieses in voller Auflösung abfragen.
Zur Referenzierung von Patienten kann die externe Karteinummer verwendet werden sowie der Vor-, Nachname und das Geburtsdatum des Patienten. Diese Daten werden bei DBSWIN in der Patienteninformationsmaske angezeigt.
Die Suche nach den Bildern eines Patienten erstreckt sich über **alle Praxen** von DBSWIN.

----------!!WICHTIG!! ----------
DBSWIN sollte so konfiguriert werden, daß die Patientendatenübernahme vom Praxisverwaltungsprogramm mit Karteinummer aktiviert ist (siehe oben). Wenn nur die Karteinummer zur Patientenidentifikation verwendet wird (Funktion FindFirst, s.u.), muß sichergestellt werden, daß ein Patient im Praxisverwaltungsprogramm und in DBSWIN dieselbe Karteinummer hat! Ein Patient muß dann in allen Praxen von DBSWIN unter derselben Karteinummer abgelegt sein.
Kann das nicht gewährleistet werden, muß die Patientenidentifikation anhand von Vor-, Nachnamen und Geburtsdatum erfolgen (Funktion FindFirstEx, s.u.). Zur korrekten Funktion müssen dann diese Daten zwischen Praxisverwaltungsprogramm und DBSWIN synchronisiert werden.

Andernfalls kann es zu Bildverwechslungen kommen! (d.h. Bilder eines anderen Patienten werden zurückgeliefert)
----------!!WICHTIG!! ----------

Demo:
Für die Nutzung der DLL gibt es ein Testprogramm auf der DBWIN - CD im Ordner Tools\DatExp. Dort gibt es außerdem ein Delphi- und einVisualbasic Beispiel, sowie die Schnittstellendeklaration für C(++)

Funktionen der DLL:
Prototypen in Pascal Syntax.
Pchar sind Zeiger auf nullterminierte Zeichenketten, integer sind 4 Byte groß.

Aufrufkonvention: **stdcall** (Reihenfolge der Parameterübergabe auf dem Stack von rechts nach links, Stack wird von der Routine aufgeräumt).
Eine *procedure* entspricht einer Funktion in C mit Rückgabe *void*
Der **Puffer** für die Ergebniszeichenketten muß **vom Aufrufer** in der unten angegebenen Größe zur Verfügung gestellt werden

Informationen zu den Datenfeldern:
PatKNR : höchtens 15 *[Zeichen]* (+ Abschließende 0) Hierbei handelt es sich um die Karteinummer.
PatVName, PatNName: Patienten Vor/Nachname. Höchtens 48 *[Zeichen]*
Anmerkung: Für **PatKNR**, **PatVName, PatNName** gilt: Führende und abschließende Leerzeichen sollten nicht vorhanden sein, sie werden von DBSWIN nicht gespeichert/berücksichtigt und falls vorhanden, abgeschnitten. Groß/Kleinschreibung wird nicht beachtet. (als Folge wird von DBSWIN z.B. ' 3' und '3' sowie 'Karl' und 'karl' und ' karl ' als *gleich* behandelt).
PatGebDat: Geburtsdatum des Patienten im deutschen Format (tag.monat.jahr). Jahr sollte vierstellig sein. Bei Jahren kleiner 100 wird 1900 dazugezählt. Trenner muß '.' sein
ImgID: 24 *[Zeichen]*
ImgTyp: 24 *[Zeichen]* Beschreibung, was für ein Objekt das Bild zeigt, in DBSWIN angezeigt unter 'Objekttyp'. Z.B. '11' bei einem Bild von Zahn 11
ImgDiag: 100 *[Zeichen]* Kommentar, der zu einem Bild gegeben wurde
ImgCreateDate: 9 *[Zeichen]* Datum, an dem das Bild aufgenommen wurde, im Format yyyymmdd.
Anmerkung: Mit der Funktin **ImgInfo** ist auch die Bilderstellungs**zeit** verfügbar

function FindFirst (PatKNR: PChar;
ImgID, ImgTyp, ImgDiag, ImgCreateDate: Pchar): integer; stdcall;

!!Achtung Diese Funktion sollte nicht mehr verwendet werden! Sie wird ersetzt durch FindFirstEx!!

Eingabe: PatKNR: Karteinummer des Patienten, für den man Bilder suchen will.
Rückgabewert: handle>0, den man zur Fortsetzung der Suche mit Findnext oder für FindClose verwenden kann. Handles behalten nur solange ihre Gültigkeit, wie die DLL im Speicher verbleibt oder man Findclose ausführt!
Falls handle <=0 ist: Fehler aufgetreten
Ausgabe: ImgID. Die BildID des ersten gefundenen Bilds, mit der man ImgToClipbrd parametrieren kann, um die Bilddaten abzuholen.
ImgTyp:Bildobjektbezeichung
ImgDiag:Kommentar/Diagnose
ImgCreateDate:Erstelldatum des Bilds im Format yyyymmdd
Falls kein Bild gefunden wurde, ist ImgID = '' (der Leerstring)
Die Suche nach Bildern erstreckt sich *über alle* in DBSWIN definierten *Praxen*!

function FindFirst Ex(PatKNR, PatVName, PatNName, PatGebDat: PChar;
ImgID, ImgTyp, ImgDiag, ImgCreateDate: Pchar): integer; stdcall;
Eingabe: PatKNR, PatVName, PatNName, PatGebDat. Diese Daten identifizieren den Patienten, für den man Bilder suchen will. Eingabefelder können auch leer gelassen werden, eine leere Zeichenkette ('') bedeutet 'Wert ist egal'.
Für **jede DBSWIN-Praxis** qualifiziert sich der erste Patient, bei dem alle von '' verschiedenen Kriterien übereinstimmen (oder auch keiner). Die ImgIDs aller qualifizierten Patienten werden in diesem und den folgenden FindNext-Aufrufen zurückgeliefert.

Rückgabewert: handle>0, den man zur Fortsetzung der Suche mit Findnext oder für FindClose verwenden kann. Handles behalten nur solange ihre Gültigkeit, wie die DLL im Speicher verbleibt oder man Findclose ausführt!
Falls handle <=0 ist: Fehler aufgetreten
Ausgabe: ImgID. Die BildID des ersten gefundenen Bilds, mit der man ImgToClipbrd parametrieren kann, um die Bilddaten abzuholen.
ImgTyp:Bildobjektbezeichung

ImgDiag:Kommentar/Diagnose
ImgCreateDate:Erstelldatum des Bilds im Format yyyymmdd
Falls kein Bild gefunden wurde, ist ImgID = '' (der Leerstring)
Die Suche nach Bildern erstreckt sich *über alle* in DBSWIN definierten *Praxen*!

procedure FindNext (handle: integer;
ImgID, ImgTyp, ImgDiag, ImgCreateDate: Pchar) ; stdcall;
Eingabe: handle. Es sollten nur handles übergeben werden, die man vorher mit Findfirst oder FindFirstEx erhalten hat.
Ausgabe: ImgID. Eine BildID, mit der man ImgToClipbrd parametrieren kann um die Bilddaten abzuholen.
ImgTyp: Bildobjektbezeichung
ImgDiag: Kommentar/Diagnose
ImgCreateDate: Erstelldatum des Bilds im Format yyyymmdd
Falls kein weiteres Bild gefunden wurde, ist ImgID = '' (der Leerstring)

function ImgInfo (handle: integer;
ImgID, whichInfo: Pchar;
buflen: integer; outp: Pchar): integer; stdcall;
Die Funktion kann innherhalb einer FindFirst/Next Suchschleife zusätzliche Bildinfo abrufen
Eingabe:
handle. Der handle für die momentan aktive FindFirst/FindFirstEx/FindNext - Suche. Der Handle muß geöffnet sein!
ImgID. ID des Bilds, für das Information abgefragt werden soll. **Es muß eine ID sein, die man im gerade aktiven FindFirst/Next Suchlauf erhalten hat!**
whichinfo: Kennzeichnungsstring, welche Information man erhalten will. Im Moment wird nur 'CreateDateTime' unterstützt, dann wird in <outp> der Erzeugungstag und - uhrzeit kopiert, im format yyyymmddhhnnss (n=minuten , s=Sekunden)
buflen: Pufferlänge des Puffers, **den der Aufrufer zur Verügung stellt.**
outp: *Zeiger auf den Puffer*, **den der Aufrufer zur Verügung stellt**

Ausgabe: höchstens <buflen> Zeichen werden nach <outp> kopiert

Ergebnis: falls > 0: Erfolg
sonst: Parameterfehler, oder imageID wurde nicht in der durch <handle> gegeben FindFirst/Next Suche gefunden.

procedure FindClose (handle: integer); stdcall;
Eingabe: handle. Es sollten nur handles übergeben werden, die man vorher mit Findfirst oder FindfirstEx erhalten hat. Für jedes FindFirst/FindfirstEx sollte man genau ein FindClose ausführen.

function ThumbnailImgToClipbrd (ImgID: pchar): integer; stdcall;
Eingabe: ImgID. Es sind nur ImgIDs erlaubt, die man von der DLL irgendwann mit Findfirst/FindFirstEx oder FindNext erhalten hat. Die Funktion kann auch außerhalb einer First/nextClose Sequenz aufgerufen werden, sie liefert ein Vorschaubild in geringer Auflösung (bis zu 140x140 Pixel). Die Funktion arbeitet schnell und sollte verwendet werden, um dem Benutzer eine Übersicht aller verfügbaren Aufnahmen zu präsentieren.
Rückgabewert:
falls > 0: Ein Vorschaubild in geringer Auflösung befindet sich im Windows Bitmapformat im Clipboard.
sonst: Das Vorschaubild ist nicht im Clipboard vorhanden. Mögliche Ursachen: es wurde gelöscht, das Bild ist nur noch auf einem Langzeitarchiv verfügbar, die Praxis wurde in DBSWIN gelöscht oder ein Fehler ist aufgetreten

function ImgToClipbrd (ImgID: pchar): integer; stdcall;
Eingabe: ImgID. Es sind nur ImgIDs erlaubt, die man von der DLL irgendwann mit Findfirst/FindFirstEx oder FindNext erhalten hat. Die Funktion kann auch außerhalb einer First/nextClose Sequenz aufgerufen werden. Die Funktion ist besonders bei hochauflösenden Scannerbildern aufgrund der Datenmenge langsam und sollte nur aufgerufen werden, wenn der Benutzer ein Bild genauer ansehen will.

Rückgabewert:

falls > 0: Das mit ImgID assoziierte Bild befindet sich im Windows Bitmapformat im Clipboard.

sonst: Das Bild ist nicht im Clipboard vorhanden. Mögliche Ursachen: es wurde gelöscht, das Bild ist nur noch auf einem Langzeitarchiv verfügbar, die Praxis wurde in DBSWIN gelöscht oder ein Fehler ist aufgetreten

Ablauf:

Mit FindFirst oder FindFirstEx wird eine Suche begonnen, man erhält einen Handle, ggf. eine BildID und weitere Bildinformationen zurück. Man mit dem Handle und der BildID kann man zusätzliche Bildinformation abfragen. Wenn die erhaltene BildID <> '' ist, kann man den Handle an Findnext übergeben und erhält dann den nächsten Datensatz, andernfalls gibt es keine weiteren Bilder für diese Suche. Mit FindClose muß der Suchvorgang abgeschlossen werden, um damit assozierte Speicherobjekte freizugeben. Man kann sich die erhaltenen BildIDs merken und die Funktion ThumbnailImgToClipbrd benutzen, um dem Benutzer eine Bildübersicht zu präsentieren. Wenn der Benutzer ein Bild der Übersicht doppelklickt, kann man ImgToClipbrd nutzen, um die Daten in vollem Umfang anzufordern.

Die Bibliothek kann mit Loadlibrary geladen und Freelibrary freigegeben werden. Freelibrary darf erst erfolgen, wenn ein Suchvorgang komplett abgeschlossen ist, da die Handles ungültig werden, wenn die DLL aus dem Speicher entfernt wird

Fehler, Probleme und Lösungen:

- Falls bei der Ausführung der DLL-Funktionen Probleme auftreten, erscheinen die Meldungen im DBSWIN-Logger. Der Logger erscheint als Icon neben der Systemuhr in der Taskleiste, ein Doppelklick zeigt ein Meldungsfenster an, der Knopf 'Show all' darin bewirkt detailliertere Anzeige. Die Meldungen werden im Verzeichnis DBSWIN\bin auch als Dateien gespeichert (log*.rtf).
- Aufgrund von Systembeschränkungen können bei manchen aufrufenden Programmen Probleme auftreten. (bei Win32 Plattformen können nur beschränkt viele DLLs pro Prozeß auf einmal geöffnet sein, wird diese Zahl überschritten, treten Fehler auf, siehe Artikel Q193462 Microsoft Knowledge base)
  Symptome:
  - beim Ausführen von LoadLibrary eine Fehlermeldung wie 'Fehler bei der Initialisierungsroutine einer Bibliothek' **oder**
  - beim Ausführen von FindFirst eine Meldung im DBSWIN-Logger wie ': An error occurred while attempting to initialize the Borland Database Engine (error $2519)'. oder
  - beim Ausführen von ImgToClipbrd erhält man Rückgabewert 0 und es erscheint im DBSWIN-Logger eine Meldung wie '...error 489 can't find file format library module'

- Wenn die dbsexpw.DLL geladen wird, erscheint eine Fenster mit der Meldung: "Error by load Library 1157". Dies kann daran liegen, daß der Versuch, eine weitere DLL aus dem DBSWIN\bin Verzeichnis nachzuladen, scheitert (siehe auch voriger Punkt).
- Die dbsexpw.DLL sucht die aktuelle DBSWIN Installation per Inidatei 'dbsExpW.ini', die im Windowsverzeichnis liegt (NICHT windows\system) Die Datei wird beim Installieren von DBSWIN gesetzt. Existiert das angezeigte Verzeichnis nicht mehr, funktioniert das Nachladen nicht.
- Format (Beispiel):
- [Setup]
- DBSWINBinPath=C:\dbswin\bin

- **Problem bei Datenbank auf Novellserver**, dies wurde beobachtet bei NovellServer V5.1, NovellClient V3.21 und Win98 Arbeitsplatz: wenn DBSWIN schon läuft und dann die Schnittstelle aufgerufen wird oder umgekehrt, bleibt DBSWIN und das Programm, das die Schnittstelle verwendet mindestenes eine Minute hängen (Fehler im DBSLog: Timeout bei Sperren der Datei PDOXUSRS.NET).

Maßnahmen:

- Novell Client V3.30 verwenden (oder neueste Version)
- Das neueste ServicePack für den Server installieren
- an der Serverconsole "set client file caching enabled=off" einstellen

Verfügbarkeit:
Die Funktionalität ist nicht in allen Versionen von DBSWIN vorhanden. Ob Ihre Version dies bereits unterstützt, erkennen Sie daran, daß im Unterverzeichnis 'bin' des DBSWIN-Installationsverzeichnisses die Datei ExpService.dll vorhanden ist. Evtl. müssen Sie Ihren Explorer erst so einstellen, daß DLL-Dateien angezeigt werden. Die FindFirstEx Funktion ist erst in Versionen >= V2.1.9 (ab Nov. 2000) verfügbar.
Sorgen Sie bitte dafür, daß Sie die aktuellste Version von DBSWIN verwenden!

Performance Aspekte:
- Manche Operationen sind zeitintensiv, so daß sie sich als Verzögerung bemerkbar machen. (Oft wird die Verzögerung geringer, wenn beim selben Programmlauf eine Operation ein weiteres mal ausgeführt wird, weil sich die Dateien dann schon im Cache befinden)
- Um die Verzögerungen möglichst gering zu halten, sollte man den Zugriff auf die Dienste entsprechend gestalten (z.B. in *einer* Find-Sequenz *alle* BildIDs zum aktuellen Patient holen, Bilddaten nicht unnötig oft anfordern)
- Man muß **unbedingt darauf verzichten**, bei jedem Aufruf eines Patienten die vollen Daten von allen seinen Bildern abzufragen, anstatt dessen sollte man **nur die Vorschaubilder abfragen**!

Es folgt eine Auflistung, welches die zeitaufwendigeren Operationen sind und was sie für Folgeaktionen nach sich ziehen:
- **LoadLibrary**: die DLL und die für Bildverarbeitung notwendigen DLLs werden geladen, der Logger wird gestartet (falls noch nicht aktiv).
- **FindFirst, FindFirstEx**: Beim ersten Mal wird die Datenbankengine geladen und initialisiert. Dies geschieht nicht mehr, solange man nicht mit FreeLibrary die DLL wieder entfernt.
  Es werden *alle* gesuchten BildIDs herausgesucht. Das bedeutet, daß nachfolgende FindNexts keine Datenbankzugriffe mehr benötigen, also schneller ablaufen. Der Zeitvorteil bleibt bis zum FindClose erhalten
- **ThumbnailImgToClipbrd**: Die Vorschaudaten sind schnell verfügbar und werden ins Clipboard transferiert

ImgToClipbrd: Mit Hilfe der Datenbank wird die Bilddatei festgestellt, von der Festplatte in den Speicher geladen und die Daten ins Clipboard transferiert. Die Funktion ist insbesondere bei hochauflösenden Scannerbildern sehr zeitaufwendig!