Innovative Access Control Systems

JANUS SQLExec

User Manual

grosvenor TECHNOLOGY

Copyright Notice

Limitation of Liability

In no event will Grosvenor Technology Ltd. be responsible for direct, indirect, incidental cover, or consequential damages arising out of the use or inability to use JANUS SQLExec or its documentation, even if advised of the possibility of such damages.

Warranty

Grosvenor Technology Ltd. makes no warranty or representation, either express or implied, with respect to JANUS SQLExec or its documentation, including quality, performance, merchantability or fitness for a particular purpose.

Grosvenor Technology Ltd.

Millars Three
Southmill Road
Bishops Stortford
Herts, CM23 3DH, UK
+44 (0) 1279 838000

Website: http://www.grosvenortechnology.com

Contacting Grosvenor Technology Ltd.

Grosvenor Technology is committed to supplying premier products and services as well as providing technical support and expertise in design, implementation, and systems after care.

Millars Three, Southmill Road
Bishops Stortford, Herts, CM23 3DH

Tel: +44 (0) 1279 838000
Fax: +44 (0) 1279 504776
Email: sales@grosvenortechnology.com
support@grosvenortechnology.com
Web: http://www.grosvenortechnology.com

Services provided

Technical Training - Comprehensive training programs to meet the needs of Approved/ Authorised Installers. Courses include design, installation, troubleshooting, and the fundamentals of PC networking, and operation and administration of all JANUS systems.

Pre-Sales Services - Technical and quotation assistance including application, configuration and installation planning.

Help Desk Services - Telephone support to assist commissioning engineers with installation and system set-up. Unlimited availability during software warranty periods, or when covered by a Software Support Plan. Also available at published rates if not covered.

Remote Diagnostics - Dial-in assistance to aid system troubleshooting. Unlimited availability during software warranty periods, or when covered by a Software Support Plan. Also available at published rates if not covered**.**

Product Repairs - Available from our workshops in Bishop's Stortford. A standard fee is charged as to Grosvenor Technology published rates. Advanced warranty replacements are available upon request.

Database Conversions - Easy upgrades from one system to another where database structures would otherwise be incompatible. Fees may be charged if complicated scripts or interfacing is required. Contact Grosvenor Technology in each instance.

Services on Demand - Supplied where a Software Support Plan is not in force. Fees are charged as to Grosvenor Technology published rates for technical support and on-site services.

Grosvenor Technology reserves the right to refuse assistance to any Service Company or engineer that has not been fully trained to our required system standards. Unqualified engineers may compromise the security and efficient working of the End-User access control and/or other security systems.

Using this manual

The manual has been laid out in 3 main sections:

- General Introduction
- System requirements and installation instructions
- How to use JANUS SQLExec

This symbol represents information important to the operation of the JANUS SQLExec application.

This symbol represents a warning of potential conflict or data loss when using the application.

Table of Contents

Introduction

JANUS SQLExec from Grosvenor Technology Ltd can be used to update a JANUS for Windows system with data from an external data source e.g. an existing personnel database, or student records.

It functions by executing SQL statements (either from stored procedures, or from a text file), against a source database. It then applies the results to the JANUS for Windows database. The results of the program execution may also be stored in a log file.

JANUS SQLExec can only modify/update fields in the JANUS Card and Users tables within the JANUS for Windows database.

Key features of JANUS SQLExec

- Bulk updates of the JANUS Cards and Users records from external data sources
- Scheduled execution of queries
- Insert only, insert/update, update only and delete modification types
- Enables the use of *Variables* to reduce the amount of processing required
- Database vendor support via BDE and ODBC

SQLExec Architecture

Figure 1 below describes how JANUS SQLExec works in conjunction with BDE to query an existing database and use the results to update the JANUS for Windows database.

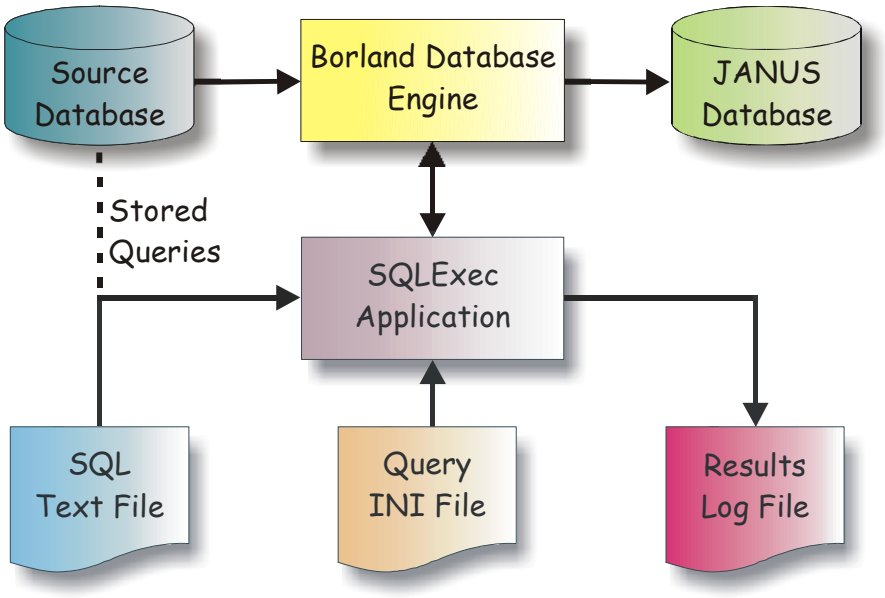

Figure 1. SQLExec Architecture

System Components

JANUS SQLExec employs a number of different files and libraries in its operation.

INI Files

Windows INI files control the operation of JANUS SQLExec. These are plain text files, which adhere to a simple format and can be read using a number of standard Windows APIs.

A minimum of two INI files are required, one to define the configuration and overall parameters for how SQLExec operates (SQLEXEC.INI), and another file that contains details on the queries that are to be executed (Query.INI). See the **SQLEXEC.INI** and **Query.INI File** sections for further details.

Database Access

All databases are accessed through the BDE. The BDE allows database access from many vendors through a common programming interface. The BDE requires the correct drivers to be installed in order to access a database.

A BDE alias is used to identify the type of database and its location. BDE aliases are configured using the BDE Administrator, which is installed in the Windows Control Panel by the JANUS SQLExec installer. See *Windows Help* for more details

Supported Databases

The BDE can connect to a number of different database types. These include Corel Paradox, dBase, Microsoft Foxpro and Microsoft Access. The BDE also allows direct connection to ODBC database connections. These are defined in the ODBC Control Panel application in Windows. SQLExec can therefore connect to all database formats that have either a BDE driver or an ODBC driver.

Microsoft Windows can have the following ODBC drivers installed:

- SQL Server
- Oracle
- dBase
- Foxpro
- Excel
- Text Driver
- Paradox
- Access

Other drivers can be obtained from third party supplier.

System Requirements

JANUS SQLExec should be installed on computers running Windows 2000, Windows NT4 and Windows XP (the use of Service Pack 6a or higher is recommended on NT4). If you have an older operating system such as Windows for Workgroups 3.11, Windows 95, Windows 98 or Windows NT 3.51, contact support@grosvenortechnology.com.

Grosvenor recommends a minimum specification of:

- 350MHz Pentium II processor
- PCI bus
- 128 Mb RAM for Windows NT, Windows XP and Windows 2000
- CD ROM drive
- SVGA colour screen
- Hard disk capacity is dependent on system size

All JANUS computers require a network card, or software equivalent to be installed to allow the correct configuration of TCP/IP. For Windows NT, Windows 2000 and Windows XP use a network card or the software MS Loopback adapter. Consult Windows Help for details.

SQLExec requires version 5 or later of the Borland Database Engine.

Licensing

Before using JANUS SQLExec for the first time, the user must contact Grosvenor Technology for a licence file to enable the application to work.

Installing JANUS SQLExec

To install the JANUS SQLExec software from the CD:

1. Insert the CD labeled 'JANUS for Windows' into your CD-ROM drive. If Autorun is enabled on your system the CD starts up and you can select **Install JANUS** from the CD autorun page.
2. Alternatively, from the **Start** menu select **Run**. Type **F:\autorun.exe** (substitute the appropriate letter of your CD-ROM drive for F).
3. The installation program prompts you to select which components are to be installed. Select **SQLExec** and click on **Next**.
4. Follow the instructions on the screen to proceed with the installation. Once underway, a screen will display both a progress bar and an estimated time to completion.
5. When all files have been installed, click **Finish** to exit the installer.

JANUS SQLExec Components

This section identifies each of the components that are installed with SQLExec.

Table 1. Components required for JANUS SQLExec

Component	Description
SQLEXEC.EXE	Application executable. Installed by default into **C:\WINJAN**.
SQLEXEC.INI	Initialisation file for SQLEXEC.EXE installed into the **system's Windows** directory. See the **SQLEXEC.INI** section below for more details.
QUERY.INI	Contains the queries that JANUS SQLExec will execute.
BDE	Provides a standardised method for accessing databases. See **BDE Installation and Settings** for more details.

SQLEXEC.INI

The installer places a copy of this file in the system's Windows directory, which the user must modify before using the application. It contains the overall settings used by the SQLExec application.

Table 2. SQLEXEC.INI file settings

Setting	Description
QueryFile	Identifies the location and name of the INI file containing the sql queries that are to be executed.
LogFile	Identifies the location and name of the file where results may be written. The log file path can use two metadata values in the file name. **#DATE#** will be replaced with today's date in YYYYMMDD format i.e. 19990809 **#TIME#** will be replaced with the time when JANUS SQLExec starts in HHMMSS format i.e. 172635
ImgOffset	The maximum offset into the supplied file to look for a valid JPG or BMP when importing into the users table. This value can be in the range of 1 to 1024. This setting is optional.
GTFmtPic	Set to 'yes'/'no' to inform JANUS SQLExec that pictures being imported are in Grosvenor Technology format. This setting is optional with a default of 'no' assumed if it is not supplied. Note: Errors will be generated if Grosvenor formatted pictures are imported without this setting.
Verbose	Set to 'yes'/'no' to enable/disable output of live information to the console window during execution. This setting is optional. If a no setting is entered then 'no' is assumed.
DeleteLog	Set to 'yes'/'no' to enable/disable deletion of the log file on the start of each run. This setting is optional. Only a log file with the same name as the current log file will be deleted. If a no setting is entered then 'no' is assumed.

SQLEXEC.INI Example File

This section provides an example of a SQLEXEC.INI file.

```
[SQLExec]
QueryFile=C:\Winjan\SQLRun.ini
LogFile=C:\winjan\SQLExec-#DATE#-#TIME#.log
Verbose=YES
```

As the SQLEXEC.INI file is fundamental to the operation of the JANUS SQLExec application, Grosvenor Technology recommends that the file be protected. This will ensure that only authorised users can make changes to the settings.

Query.INI File

Query INI files may contain one or more queries. Each SQL query is made up of a descriptive title enclosed in square brackets and some of the following item headings. See the section on **Example Query INI Files** for more details.

Operation

This section identifies the type of operation that the query will perform. **Table 3** below describes all valid operation values.

Table 3. Valid Operation values

Value	Description
I	Insert new records. If the record already exists in the database, the insert operation is aborted.
U	Update existing records. If the record is not found an insert operation is performed. If a record is found, it will be modified. This is the equivalent of I + M.
M	Modify existing records. If the record is not found the modify operation is aborted.
D	Delete all records identified by the query.
V	Validate the records only. Will not make any changes to the JANUS database. All records that fail validation will be logged.
N	No Operation. The query will be run on the source database only. No changes will be made to the JANUS database.

Alias

This section identifies the BDE alias used by SQLExec to access the source database. This alias must exist within the BDE Administrator alias list.SQL

Identifies the file containing the SQL to be executed against the source database. The file name can be in UNC format. The query should be formatted in ANSI SQL.
Note: This item cannot appear in the same query section as a 'Proc' item.

Proc

Defines the stored procedure that SQLExec will execute to create the record set for the operation.
Note: This item cannot appear in the same query section as a 'SQL' section.

Username

Defines the database user identifier that is to be employed when opening the source database. The password entry (see below) must correspond to this user id.

Password

Identifies the password to be employed when opening the source database with the above alias item. This parameter is optional though if used must be valid for the user name that has been set for the above alias.

Dest

Identifies the destination for the imported records. This item can currently be *CARDS*, *USERS* or *VALUES.* These options allow records to be imported into the cards and users tables in the JANUS Database, or into a holding table (Values), which can be used to store data for use in further queries.

Variables

In the context of SQLExec, a variable is a name/value pair. This can be created using a query, stored in the *VALUES* destination described above and retrieved for use in further queries. Variables can only be used in queries, NOT in stored procedures. See **Appendix A – Using Variables** for more details on when and how to use variables.

Info

This can be any user-defined text associated with the query. This will be displayed on screen when the query is run.

Example Query INI Files

A query INI file can contain multiple queries. Two examples are shown below.

Example 1

In this example, there are two queries contained in the INI file, these will run on different source databases.

```
[Insert Query]
Alias=GuardCard
SQL=\\employee\queries\insert.sql
Operation=I
Username=sa
Password=

[Modify Query]
Alias=ManagerCard
Proc=Update_Query
Operation=U
Password=
```

Example 2

This example contains three queries in the INI file.

This first query inserts values retrieved from the source database into the *VALUES* table.

```
[Insert Values]
Alias=DummyDB
SQL=d:\devwork\sqlexec\binr\values.sql
Dest=Values
Operation=I
```

The second query inserts records into the *USERS* table for people whose surname begins with A and B. It also stores a variable called **'the surname'**. It does this by iterating through the columns of the result set of the query and stores the name of the column as the *name* part of the variable (in this case **Surname**) and the data as the *value* part of the variable (in this case, the name e.g. Brown).

```
[Insert A & B, Variable stored as 'the surname']
Alias=DummyDB
SQL=d:\devwork\sqlexec\binr\users a-b.sql
Dest=Users
Operation=I
variables=Surname=the surname
```

The final query now uses the variable created by the second query. The name of the item is passed to the query file using a meta-value, in this case #VAR#the surname#. JANUS SQLExec substitutes the value found for that name into the query before running it. In this instance, the value of the variable is Brown, as it was the last surname ending in B. Therefore, this query will insert the records for people with surnames from C onwards into the USERS table of the database.

```
[Insert c to z, Variable retrieved and stored as 'the surname']
Alias=DummyDB
SQL=d:\devwork\sqlexec\binr\users c-h.sql
Dest=Users
Operation=I
variables=Surname=the surname
```

As the Query.INI file is fundamental to the operation of the JANUS SQLExec application, Grosvenor Technology recommends that the file be protected. This will ensure that only authorised users can make changes to the settings.

BDE Installation and Settings

The BDE is installed to:
c:\Program Files\Common Files\Borland Shared\BDE

Each database that SQLExec will be required to query must have an alias created using the 'BDE Administrator'. The 'BDE Administrator' can be found in the Windows Control Panel. See *Windows Help* for more details.

By default, SQLExec also requires the JANUS for windows settings to be present. These are 'JanusData' and 'JanusSys' and are created by the installer.

If the source database is connected via ODBC, then a 'mapped' BDE alias must be attached to a genuine BDE alias.

User Names and Passwords

Some databases require the use of specific user names and passwords. The manner in which these settings are used is dictated by the BDE and the underlying database management system. For Paradox databases no user name is required, however Microsoft Access and SQL Server databases have the username set in the BDE alias settings page. The required password will be retrieved from the INI file identified by the QueryFile setting in SQLEXEC.INI. See the **SQLEXEC.INI** and **Query.INI File** sections for more details.

Using the JANUS SQLExec software

This section describes how to set up and use SQLExec. This assumes that a successful installation has already been performed. See **Installing JANUS SQLExec** for more details.

Preparation

In order to use SQLExec, the following items must be created:

1. Initialise the SQLEXEC.INI file.
   Modify the default SQLEXEC.INI file created by the installer. See the **SQLEXEC.INI** section in **JANUS System Components**.

2. Create or modify the INI file identified by the QueryFile setting in the SQLEXEC.INI file.
   This file contains details on the queries that SQLExec will execute. For details on the contents and format of this file, see the **Query.INI File** section.

3. Initialise the Database Aliases.
   SQLExec retrieves a set of records from a source database in order to update the JANUS for Windows database. The source database and the JANUS for Windows database are identified to SQLExec via BDE Aliases. These must be unique and are configured in the 'BDE Administrator' located in the Windows Control Panel.

4. Setup a Password and User ID.
   The SQLExec application requires the following Password and User ID to be set up. These should be entered via the JANUS Database Update program.

   Password – SQLEXEC
   User ID – SQLEXEC
   The logon level of the JANUS SQLExec user is not important.

Before using JANUS SQLExec for the first time, the user must contact Grosvenor Technology for a licence file to enable the application to work.

Import Fields

This section provides guidelines on the fields that are used during SQLExec operation.

Validity Checks

When SQLExec works with records, it performs a number of validity checks against each record. This ensures that the JANUS database is maintained in a consistent state.

Key Validity

Any record in the database has to be unique. This is accomplished by defining one or more of the pieces of information from the record that when taken as a whole, can appear only once.

SQLExec enforces this rule by making sure that the supplied record has all of the key fields supplied. In some cases, the JANUS database allows some of the key fields to be blank. However, SQLExec will ensure that this is the correct option as it requires the fields to be defined as being blank.

Required Fields

Required fields have to be entered as a valid value. The value will normally have to be defined in an associated JANUS database table.

Other Rules

Other rules are enforced when appropriate. See **Tables 4** and **5** for more details.

Modifying Key Fields

To modify key fields, the supplied record must contain the old values for each key field and new values for each of the key fields to be changed. The field names for the new key values are the name of the key field preceded by "_NEW_KEY_VAL_" (not case sensitive). The field name for a new surname value, for example, would be "_NEW_KEY_VAL_SURNAME".

Suppose a record exists with the following key in the JANUS Card Table;

Surname=Jean
First Name=Norma
Title=Ms
Emp Number=1234
Issue=1

Suppose that SQLExec is used to change the surname to “Monroe” and the first name to “Marilyn”. The following record must be supplied;

Surname=Jean
First Name=Norma
Title=Ms
Emp Number=1234
Issue=1
_NEW_KEY_VAL_Surname=Monroe
_NEW_KEY_VAL_First Name=Marilyn

The old key fields are used to locate the record in either the “JANUS Cards” or “JANUS” Users table, the key fields are then set to the values in the “_NEW_KEY_VAL_...” fields as supplied.

Card Records

Table 4 defines the data rules that have to be met for JANUS SQLExec to work successfully with JANUS Card records.

Table 4. Data rules for card records

Field	Key	Req'd	Blank	Rules
Surname	✓	✓	×	Must not be blank
First Name	✓	✓	×	Must not be blank
Title	✓			
Emp Number	✓			
Issue	✓			
Card Number				If the card number is supplied it must either not exist in the database, or must already be used in the card record that is being updated.
Card Group		✓	×	Must contain a valid entry from the JANUS card groups table.
Expiry Date				
Expiry Time				
Card Status		✓	×	Can contain anything (up to the field's size limit) with 'Valid' making the card useable on the JANUS system. Standard entries include: Valid, Pending, Suspended, Lost, Stolen, Destroyed, Not issued.
Report Group				Must contain a valid entry from the JANUS report groups table.
APB Exempt				
Department				If present, must contain a valid entry from the JANUS department table.
Card ID				
Comments				
Start Date				
Start Time				
Global CG				For use with multiple card groups. Contains a comma-separated list of card groups. Must contain valid entries from the JANUS card group table.
Card Design				If supplied, must contain a valid entry from the JANUS Card Design table.

User Records

Table 5 defines the data rules that have to be met to successfully work with JANUS User records.

Table 5. Data rules for user records

Field	Key	Req'd	Blank	Rules
Surname	✓	✓	✗	Must not be blank
First Name	✓	✓	✗	Must not be blank
Title	✓			
Emp Number	✓			
Def 2				
Def 3				
Def 4				
Def 5				
Def 6				
Address				
Picture				Contains a JPG or BMP file.
Signature				Contains a JPG or BMP file.
PicFile				Full path to a JPG or BMP file. If this field exists it will be used instead of the Picture field.
SigFile				Full path to a JPG or BMP file. If this field exists it will be used instead of the Signature field.

Starting JANUS SQLExec

The application can be run from the **Start** menu, or from the **Command** prompt. Alternatively, JANUS SQLExec can be scheduled to run at a specific time using the *Windows Task Scheduler*, which can be found in the Control Panel. See *Windows Help* for details of using this option.

1. Ensure that JANUS Comms is running and select **Start • Programs • JANUS for Windows • JANUS SQLExec**. Alternatively, select **Start • Run**. Type **cmd** then click on **OK**. The command prompt will open.

   Change directory to the location of the SQLEXEC.exe (this is usually C:\winjan). Type **sqlexec** and hit return.

2. The application will automatically start and run the queries defined in the Query.INI file named in the SQLEXEC.INI file.

3. Upon completion of all operations, JANUS SQLExec will close down. **Figure 2** shows an example of the information displayed in the command prompt window.

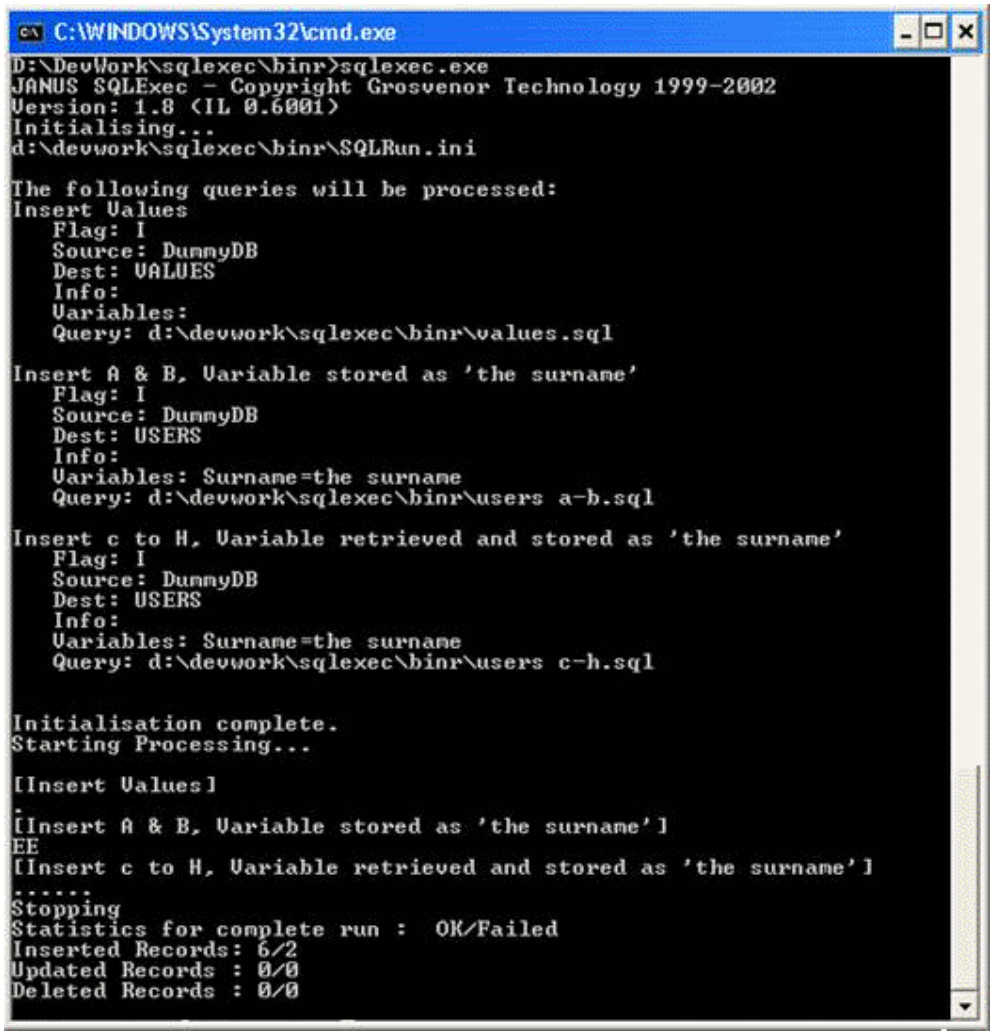

Figure 2. JANUS SQLExec runtime details

If a log file has been defined in the SQLEXEC.INI file, full details of the query results can be viewed. An example of a log file is shown in the **Log Files** section.

Log Files

A log file is generated each time JANUS SQLExec is run, only if a log file has been defined in the SQLEXEC.INI file (see the **SQLEXEC.INI** file section for more details).

- Log files are listed by the date and time at which they were generated.
- Log files are plain text files which can be opened and printed using a wide range of programs.

The example below shows that a query failed to run because either the User ID or Password listed in the Query.INI file was incorrect.

```
[SQLExec–20011129–160308]
29\11\101 16:03 - ////////////////////////
29\11\101 16:03 - SQLExec is starting
29\11\101 16:03 - ////////////////////////
29\11\101 16:03 - JImport Create Session: Username or password is
incorrect
29\11\101 16:03 - Statistics for complete run :  OK/Failed
29\11\101 16:03 - Inserted Records: 0/0
29\11\101 16:03 - Updated Records : 0/0
29\11\101 16:03 - Deleted Records : 0/0
29\11\101 16:03 - Validation Failed on 0
```

A range of error codes may be displayed in the log files, see **Table 6** for a description of each error code.

Table 6. Error codes

Error Code	Description
1	The function completed successfully.
0	An undefined error occurred. This is a catch-all error condition that is returned whenever an unexpected error occurs.
-1	The given user name and password could not be validated against the JANUS user database.
-2	The record could not be inserted as there is already a record of that key in the table.
-3	An invalid parameter was detected.
-8	The record being updated or deleted could not be found.
-9	The BDE aliases could not be found or created.
-10	Unable to find a JANUS installation.
-14	The record being entered or updated is not valid.
-15	The key provided is not valid.
-16	The record could not be inserted.
-17	The record could not be written. Note: This error can indicate that the database write has failed due to tables being corrupt.
-18	The record being updated or deleted could not be locked. The original record is unchanged.
-19	An attempt to update a card record in the JANUS database failed because the given card number does not exist. This error is returned because the card number field has to be unique.
-20	An attempt to insert a card record in the JANUS database failed because the given card number already exists.
-22	The PIN supplied was invalid.
-23	The supplied key contained duplicate entries.
-24	The supplied data record contained duplicate entries.

Appendix A – Using Variables

Variables are supported by JANUS SQLExec in order to preserve information between executions of the queries. This is most useful when the user wishes to limit the amount of processing that SQLExec undertakes.

For example, a user runs SQLExec to import new records into the JANUS database. The query needs to be run frequently as new records are added rapidly. As more records are added, the result set that SQLExec uses will increase in size. The result of which is an increase in the time it takes JANUS SQLExec to process the records.

What the system needs is a method of feeding back the key of the last record processed into the query the next time it is run. In this way, SQLExec will start processing from the next record in the series. Variables provide this facility in JANUS SQLExec.

Variables are stored as *name/value* pairs within the JANUS for Windows database. The *name* is a textual field up to a maximum of 20 characters in length. The characters allowed are subject to SQL permissible character restrictions. The *value* part of the variable is stored as a BLOB (binary large object) field, also subject to the same character restrictions.

Although it is possible to store large amounts of data within the *value* field, the only use this data can be put to is within other SQLExec queries.

Using variables

To make use of a variable, a meta-command must be added to a query. For example, the following query uses a variable named **Index** in order to limit the number of records that will be processed.

```
Select *
from
  MyTable
where
  ID > #VAR#Index#
```

SQLExec parses the query before it is passed to the database layer in order to substitute the #VAR#Index# with the value of the variable named Index. If the variable does not exist, the meta-command will be removed and nothing used to replace it.

Variables may only be used within query files, NOT in stored procedures.

Storing variables

To store a value into a variable (and to also initialise it), settings must be added to the **Variables** section of the Query.INI file.

The *Variables* setting is a comma-separated list of column names, with optional aliases. For example,

```
Variables = MyColumn:ColumnAlias, Column2
```

- MyColumn is the name of the column in the results set which is processed.
- ColumnAlias is the name to use when storing in the variables.

When the variable is to be used in a query, it must be referred to by the alias. If an alias is NOT given, then the name of the column is used as the name for the variable.

Initialising variables

It can often be useful to initialise variables in a *one-off* manner. For example, creating a query that sets up a variable with a value, without using the *real* query. This is performed using the *VALUES* destination in the Query.INI file.

When used with a query, JANUS SQLExec will process the results set returned by the query and create a variable for each column that occurs in that results set. In the following example a query is processed and the results set shown in **Table 7** is returned.

Table7. Example results set

Column Name	Data
Index	12
ExpDate	92/07/31
Name	Jones

This leads to the creation of the three variables shown in **Table 8**.

Table 8. Variable details

Variable Name	Variable Value
Index	12
ExpDate	92/07/31
Name	Jones

If a results set is returned containing more than one row, then the behaviour of JANUS SQLExec is undefined.

Using and updating variables concurrently

It may be necessary while using a variable in a query, to update its value while processing the results set.

In the following example, a variable is initialised to limit the number of records processed. The value for the variable is saved as the last record number processed. Since the last run of JANUS SQLExec however, more records have been entered, and only those records should be processed when SQLExec is next run, with the last record processed again stored in the variable.

The **Variables** setting in the Query.INI File would contain an entry like the one shown below:

```
Variables = Emp_No:Last Processed
```

This will store the values form the Emp_No column in the results set in a variable called **Last Processed**. The query should look like the following:

```
Select *
from
   Cards
where
   Emp_No > #VAR#Last Processed#
```

In this way, after each row in the results set has been processed, the value of Emp_No from the results set is stored in the variable **Last Processed**.

Glossary

API - Application Programming Interface.

BDE - Borland Database Engine.

Cards - Everyone in the system is allocated a card. Each card has a unique number, which is contained in the card record within the system database along with the cardholder's details and the Access Choices that have been allocated. This information determines through which doors and at what times the cardholder is allowed.

Database - A store of information on the MAIN JANUS PC. The information defines cards, access choices, doors, etc., and so determines how the access control system operates. The database is edited and saved using the JANUS Database Update program.

Field - Each part of the record in which you can enter data is called a field. A record is made up of a number of fields. Certain fields contain text such as a name, others contain numbers such as the card number.

JANUS Comms - The main program used by JANUS to communicate with between elements of the system, including doors, readers and other JANUS programs.

JANUS Database Update - Part of the JANUS for Windows suite of applications, this program is used to edit card and user records.

JANUS for Windows - JANUS for Windows is an on-line access control system with software options from 2 doors up to 1000+. JANUS provides access control, graphical alarm monitoring, video imaging, card production, asset protection, plant control and integration with safety and building management systems such as intruder, fire, CCTV, T&A, cashless vending and building management.

Password - This is a security measure used to restrict access to the JANUS for Windows system and resources. It is a unique string of characters that must be provided before a User ID is authorised.

Record - Each group of information within the database, for example, about a card, a time zone, or an area is called a record.

SQL - Structured Query Language.

User ID - This defines the logon level of a JANUS User and so determines the degree of access to the JANUS for Windows system resources. It is used in conjunction with a password.

Index

Millars Three, South mill Road, Bishops Stortford, Herts. CM23 3DH, UK
Tel: +44 (0)1279 838000, email: sales@grosvenortechnology.com, web: www.grosvenortechnology.com