

:SQL INTERCHANGE

Introduction

Your **IES** provides you with direct connectivity to SQL databases. Data may be retrieved from SQL databases, or updated (written) to SQL databases. INFOLAB provides you with 3 simple-to-use routines, which may be called from any INFOLAB objects, e.g. Screens, Programs, Triggers, etc.

Purpose of the Manual

The purpose of this Document is to introduce the provided connectivity routines, and how to use them.

IES remains very much a multi-dimensional database solution, but provides you with options to connect to the so-called relational world. As such, the purpose of these connectivity tools is decidedly NOT intended to encourage you to store system data in a SQL database, unless there is a real need to do so. In other words, while IES functions and stores its data in a multi-value database, you may have valid reasons to engineer some interface connections to SQL databases, for updating or retrieval of data on / from either side (e.g. integration of systems).

How to establish a Connection

This is a very easy thing to do. Typically, you require User Id, Password, Initial Catalog and Data Source to connect to a SQL database. In IES, it is only necessary to establish the 'Connection Parameters' once per Session, unless you will operate multiple such connections, in which case it will be necessary to call the connection routine before each 'get' or 'do' operation.

Please note that IES does not remain permanently connected to the SQL database for the duration of the Session. Instead, it connects when required to perform a 'get' or 'do' operation and then disconnects, but it continues to use the same Connection Parameters until they are changed.

The connection routine is called like this:

```
set.sql.connect(uid,passw,catlog,dsource,result)
```

You may call this routine from any code or object in IES.

'uid' = User Id, and is mandatory

'passw' = Password, and may be NULL

'catlog' = Initial Catalog to connect, and is mandatory

'dsource' = Data Source, e.g. in the SQL Group, and is mandatory

'result' = NULL, and returns NULL if the Connection succeeds, else the Error Message for the Connection failure is returned.

How to perform a GET action

A GET action is used whenever you wish to retrieve a Record Set FROM the SQL database.

SQL INTERCHANGE

The routine to call is: -

```
get.from.sql(params,result)
```

where

```
params = a dynamic array,  
params<1> = sql statement,  
params<2> = target_filename  
and params<3> = start_rec_number
```

The SQL Statement must be formulated by you (i.e. your program).

The Target Filename is a FileName in IES, and represents the Filename where the results will be written. Alternatively, for small record sets, you may use '*' instead of a Filename, in which case the results set will be returned in 'params', as a dynamic array, with each attribute comprising a Record (i.e. row in SQL), and each Field is a multi-value within the attribute.

Start_rec_number is the 1st Target Record Key to apply on 'Filename' (this may be zero, especially if '*' is used instead of a Filename).

The system will execute the SQL statement onto the SQL database, and will process the resulting record set, if any. If output is to a Filename, the 1st Record will be written to a 7-digit key, representing the start number. In other words, if Start_rec_number is 10, this key will be "0000010", with the next key being "0000011", and so on. If a Filename is used, then params will return the value of the last record written. For example, if the start rec = 10, and the system writes 3 records, e.g. "0000010", "0000011" and "0000012", then 'params' returns "12". If '*' is used instead of a Filename, then the results are in 'params'. In this example of 3 records, 'params<1>' = the content that would have been in record "0000010", with each Field being a multi-value.

If any error is encountered, then 'result' returns the error message, else it returns NULL.

How to perform a DO action

The DO action is used for all other operations, i.e. other than retrieving a record set. Therefore, use the DO action for updating records in SQL, deleting, etc. The entire action detail should be encapsulated in the SQL statement.

The routine to call is: -

```
do.to.sql(sql_statement,result)
```

The system will perform the SQL statement onto the SQL server, and 'result' returns NULL for a successful execution, else it contains an error message.

In Summary

Use 3 easy-to-understand IES subroutine calls (no ODBC setup is required!) to perform any required operations to a SQL database. This may be used to populate parts of a Screen, update results after a screen, or simply as a data exchange between IES and an SQL Server, or you may even write a daily

SQL INTERCHANGE

synchronizing routine to keep certain tables harmonized between **INFOLAB** and a SQL database, perhaps asking ALERTER (see ALERTER User Manuals) to do this automatically at regular intervals.

© Infolab, 2004.

This Documentation is copyrighted by Infolab (Pty) Ltd. [www.infolab.cc] All rights are reserved. Licensed INFOLAB Users are granted permission, for internal use ONLY, to reproduce the Documentation, and to include amendments dealing with specific instructions local to your installation.