Easycom For Delphi

Introduction

Introduction to Easycom For Delphi

Easycom For Delphi is a complete and powerful AS/400 development set of tools for Delphi. It is an high-level performance middleware to access all AS/400 resources and services in a native mode.

- **Easycom For Delphi** opens the access to AS/400 data through the standard Delphi components (TTable, TQuery, TDatabase) by using alias definition.
- **Easycom For Delphi** includes a full record level access over the TCP/IP protocol.
- **Easycom For Delphi** shields the developers from the complexity of designing and calling API's directly.
- Accessing data is performed in native mode for all file I/O operations.
- Record Blocking is optimized and completely handled in a transparent form.
- Access all AS/400 resources in batch mode including executing any command or program on the AS/400.
- Communicating with all preexistent COBOL, RPG, CL, and C, in a very simple manner.
- Access to stored procedures, data queues, and data areas.
- The security and data integrity of the AS/400 is not compromised in any way, offering Delphi developers a secure and reliable high capacity data server.
- **Easycom For Delphi** uses the Easycom engine that was proved over the years as one of the best integration of visual component access to AS/400 data.

Performance

Using **Easycom For Delphi**, Delphi applications can have complete real time access to AS/400 resources, without any data replication.

Easycom For Delphi database-related features include

Files can be read from, written to and updated whether they are physical, logical or 36 format. In order to provide optimum performance, **Easycom For Delphi** uses existing access paths and uses native mode access only.

- Delphi programs use logical files as indexes.
- No code is generated on the AS/400.
- Database integrity constraints are fully respected.
- Access to query files.
- Setup and query of databases (OPNQRYF, OVRDBF, etc.).
- Simultaneous access to multiple AS/400.

- File sharing and record level locking.
- Access to Commitment Control.
- Record Level access to data.

Easycom For Delphi resource-related features include

All AS/400 ressources can be accessed from within Delphi. Programs can be called from within Delphi, parameters can be passed, and the results can retrieve back in the PC environment.

- Submit a command and execute on AS/400.
- Invoke AS/400 programs and retrieve returned parameters.
- Access to data areas and data queues.
- Access to AS/400 jobs.
- Access to communication APIs.

AS/400 Communication Protocol

TCP/IP

TCP/IP is the standard in communication for AS/400. It provides better performances and is integrated in Windows, so no additional product is required.

Native Mode

All database access is done in native mode for optimum access speed and to minimize resource requirements. Native mode access yields performance, which cannot be attained using DDM (Distributed Database Management), or resource intensive SQL.

Batch Reads

By using batch reads, entire sets of data can be retrieved in a single pass.

Integration

Once the AS/400 side of **Easycom For Delphi** is installed on the AS/400, nothing else needs to be done on the AS/400.

No programs need to be compiled on the AS/400, no host programs need to be written, and resource usage is minimal.

Easycom For Delphi uses the database description from the AS/400, in real time.

There is no duplicate copy kept on the client station. Data type conversions are done automatically, and access can be in either navigational mode or through SQL.

Delphi developers access data through the standard VCL (Visual Component Library) components.

Applications can simultaneously use data on AS/400 system(s), local files and on other servers.

For example, you could easily create an application using personnel data from the AS/400, sales orders from a Microsoft SQL Server, and commission rates from an Oracle database.

How does it work ?

Easycom For Delphi is a Client/Server product. The Client running on the PC sends requests and receives results from a Server running on the AS/400.

Server Side

The EASYCOM database engine resides on the server. This proven technology provides a robust and reliable way to reach the AS/400 data.

The EASYCOM engine is capable of supporting multiple simultaneous clients, from multiple simultaneous client surfaces (i.e. : Delphi and Excel).

For further information, please see “EASYCOM Integration and Interoperability”.

Client Side

A request for data from an AS/400, is sent from client program and dispatched to the EASYCOM server running on the AS/400. Data is returned to the client program.

EASYCOM Integration and Interoperability

Easycom For Delphi is fully integrated with the EASYCOM communications engine, making all of the advanced AS/400 functionality available to Delphi developers.

This integration allows users to access data using Delphi applications, with simultaneous access by Excel or Word users who are using the same data.

Record-Oriented

Because **Easycom For Delphi** is record-oriented ; it lends itself well to applications, which require fast performance in a client/server environment.

Co-operative Systems

Because RPC is fully supported, you can achieve tremendous performance gains and reduced network traffic by having your AS/400 developers working with the Delphi developers.

Batch-oriented processing routines could be written to run on the AS/400 and be invoked by the Delphi applications. This optimum approach leverages the benefits of both environments, and maximizes the investment in hardware.

Incremental Phase-in

Because RPC allows creation of co-operative systems, you can start developing in Delphi and call existing routines written on the AS/400, which makes it possible to upgrade systems on a module by module basis.

System Requirements

System Requirements

AS/400 System Requirements

- All OS/400 version from V3R2 to V7R4. Minimum of V5R3 is recommended. Versions V7R3 and V7R4 are supported during installation. Ask to technical support for older versions.
- TCP/IP protocol.

PC System Requirements

- TCP/IP protocol.
- Operating system : all versions supported by Microsoft.
- Delphi version from XE7 to Alexandria 11.

Installation Steps

Installation

The installation procedure for **Easycom For Delphi** is performed from a PC and it has two different parts :

- The installation of the PC side.
- The installation of the server side on the AS/400.

The AS/400 server side is to be installed once only per AS/400. The following steps will be performed during each side of the installation.

On the PC side :

- Creating an ***'\ACE400'*** directory.
- Copy all required files.

Depending on the options selected during setup, "ACE400" and "EASYCOM" component tabs will be created in the tool palette.

On the AS/400 :

- Creating an library, default name is ***'EASYCOM'***
- Creating a variety of files in this library.

Requirement to install

The PC station has to be connected to the AS/400 by either an **APPC router** session (PCS, Client ACCESS, NETSOFT...), or by an access **TCP/IP** to the AS/400.

For the installation only, the user profile using the AS/400 connection has to be of type ‘QSECOFR' security officer. It is possible to set that profile during the installation steps.

Prior to installing the server side of **Easycom For Delphi**, the user has to know the type of connection to the AS/400.

- Default user profile.
- Name of the AS/400 or its IP address.

AS/400 Installation Steps

Installation using TCP/IP is performed with the FTP protocol. In this case a FTP server on the AS/400 is required (for installation process only). It is also possible to install with an APPC router and then switch to a TCP/IP connection.

If you used our FTP TCP/IP Server installation, CFGEACTCP command will be automatically launched. Otherwise, you can launch it by hand. This command will :

- Create JOBQ in EASYCOM system.
- Create JOBQ in EASYCOM library.
- Create the Class : Easycom/EACCLS (*CLS).
- Create the sub system : Easycom/EASYCOM (*SBSD).
- Start the sub system Easycom with the job EASYCOMD on it.

You can modify the objects generated by CFGEACTCP to tune your system.

The Easycom Jobs will start according to the description Easycom/EACJOBD.

If Easycom/EACJOBD does not exist, the jobs start according to the job description of the user, or according to QDFTJOBD.

By default it uses the priority class EACCLS defined in the installation library.

Notice : EASYCOM is the default library name. It could be changed during installation process.

After each IPL, you have to

- Start the Easycom sub system by the command :
  `STRSBS Easycom/EASYCOM`

This start can be included in the autostart of your system for next IPL.

You can configure your system to run the jobs in another subsystem. In that case, you have to create a JOBD named EACJOBD in the Easycom library. EACJOBD$ is given as a sample of the JOBD.

In case of troubles :

Check if the EASYCOMD job is running in the Easycom subsystem. This job is started while the EASYCOM subsystem is started.

Once the JOBQ EACJOBQ exists in the EASYCOM Library, this JOBQ must be added to an active subsystem, and the routing data and class have to be configured correctly.

Opt	S-syst/trav	Util en cours	Type	% UC	Fonction	Etat
__	EASYCOM	QSYS	SBS	0,0		DEQW
__	A168.15.41	QPGMR	BCH	0,0	PGM-EASYCOM	TIMW
__	DSEBBAN	QPGMR	BCH	0,0	PGM-EASYCOM	TIMW
__	EASYCOMD	QTCP	ASJ	0,0	PGM-EASYCOMD	SELW

Easycom For Delphi will automatically use the PC station name to identify the job on the subsystem. When the station id is not found then the IP address will be used.

To monitor the jobs executed on the AS/400, operations could be reviewing the jobs in the Easycom subsystem.

AS/400 license

To run Easycom For Delphi on your AS/400, you have to purchase a user license.

You have to register the serial number of the Easycom for Delphi product on our website.

Easycom For Delphi connector

The Easycom IDAPI Driver (with BDE)

The Easycom IDAPI Driver

The IDAPI Driver support is possible from Delphi XE7 to Alexandria 11, in any 32-bit application using the BDE.

The BDE is mandatory for this driver.

Using the IDAPI Driver is done from the TDatabase, TTable, TQuery, ... components. Those components can be found in the "Database" or "BDE" tab, depending on the Delphi version you are using.

BDE components

BDE has been removed from RAD Studio.

You are encouraged to use “native access” components.

If you still want to install the BDE components perform these steps:

1.- Download the external BDE installer from your “My downloads” section: https://my.embarcadero.com/#downloadsPage

and complete the installation process

2.- After this step, projects that use the BDE units will compile correctly. If you also want to see the components in the IDE you need to install the design package for the “BDE components”. From the IDE select “Component->Install packages…”

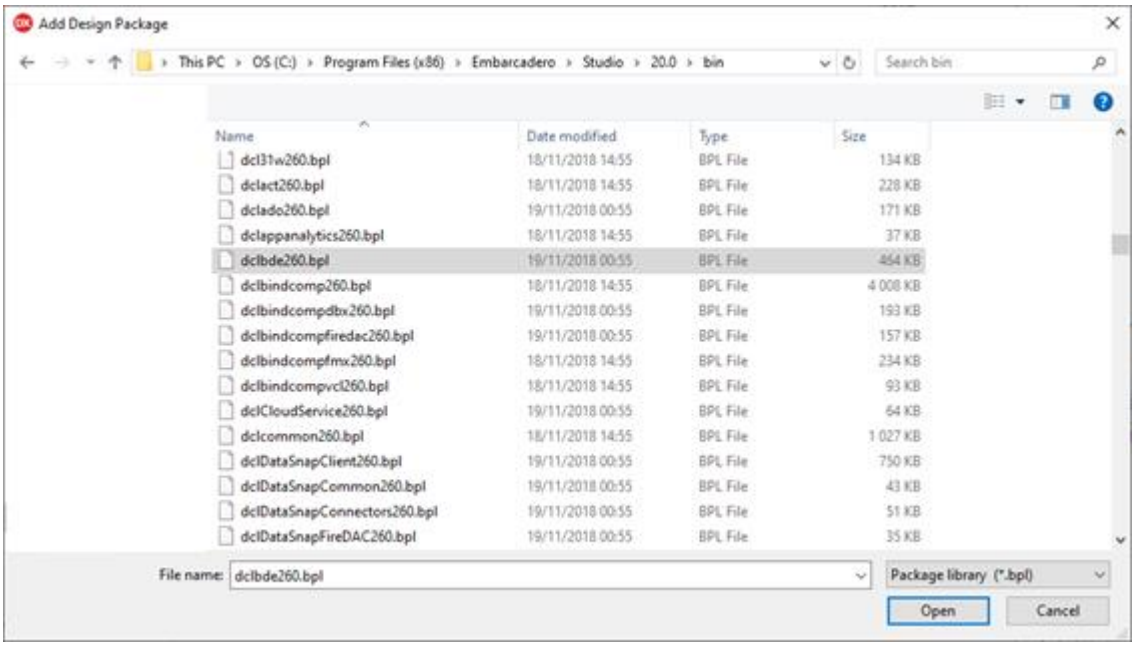
3.- There click over the button "Add..."

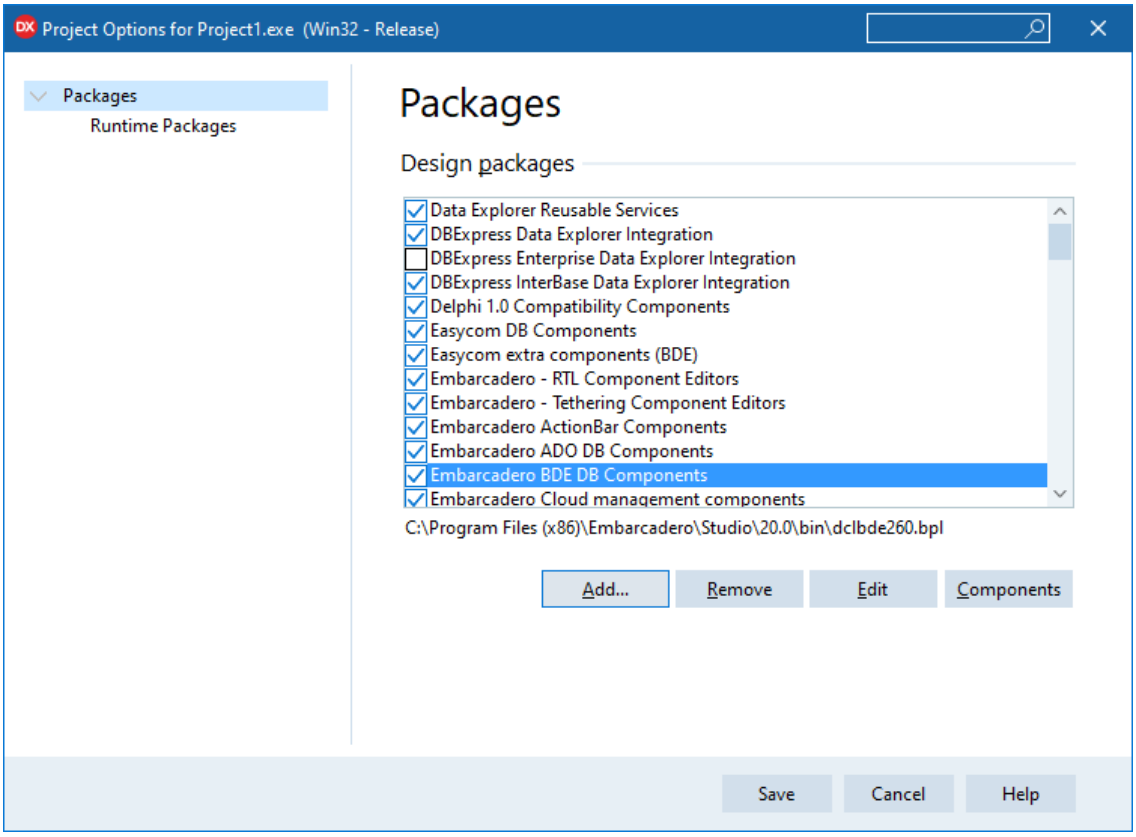
4.- Browse to the bin folder of the IDE installation (by default, C:\Program Files (x86)\Embarcadero\Studio\XX.0\bin, where XX can be 15, 16, 17…) and search for the

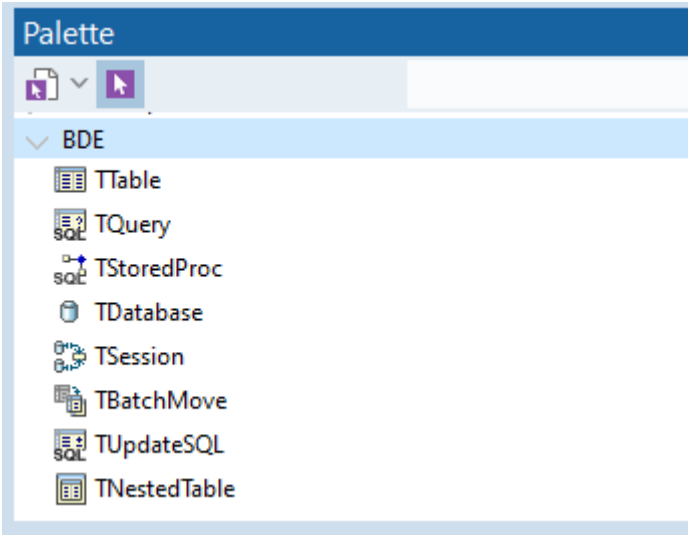
file "dclbdexxx.bpl", where “xxx” can be 210, 220, 230… depending on the version. Click “Open”

Afterwards, the “Embarcadero BDE DB components” will appear in the list of design packages. It maybe necessary to restart RAD studio to see components into e project.

Example for Delphi RIO 10.3







Alias configuration options

These are main configuration options.

These options can be changed at 3 different levels:

- At the driver level.
- At the alias level.
- In a TDatabase component of the application.

SERVER NAME

This is the name (or IP address) of your AS/400. It is possible to change it at run time, and connect to multiple AS/400 at the same time.

NETWORK

It represents the network type to use to connect to the AS/400.

If blank, the default value is taken from ACE/400 configuration.

Other possible values are:

- Tcp for TCP/IP.
- ehn for all EHNAPPC routers (like Client/Access, Netsoft, ...).
- sna for Microsoft SNA server.

LIBRARY NAME

This is the default working library. If in your applications you don't name the library, the value will be the default one.

Valid values are:

- any existing library name on your AS/400
- *LIBL
- *USRLIBL (default)
- *CURLIB

BROWSE LIBS

During design time, a special dialog box prompts allowing you to browse the files of your AS/400.

If you don’t want to see this dialog box, change this option to FALSE. The files will be the files of the library specified by ‘LIBRARY NAME’ option.

Default value is TRUE.

CODEPAGE FILE

This is the name of a codepage file, to allow custom EBCDIC-ASCII conversions.

> **In most cases, you will NOT need to fill this setting. The appropriate code page will be automatically chosen by analyzing your PC and AS/400 configuration.**

Be careful : You must specify the FULL PATH of the file.

The code page files are provided in the "international pack".

If you don’t have the codepage file and for Korean/Japan/China countries please install the International pack to have appropriate conversion pages.

To know which is the codepage file for your AS/400, make a DSPSYSVAL QCCSID on a green terminal.

C:\Program Files\EasycomINT\e285ansi.cpg

CONVERSION RULES

This parameter allow to customize field conversion rules.

It contains a list of rules separated by a semicolon.

Possible rules are :

- LONGNUM=double

By default, a number (PACKED or ZONED) that cannot fit a double is converted to characters. With this option, it is converted to a double (and can lose precision).

- LONGNUM=BCD

With this option, big numbers will be converted to Delphi BCD type. But unfortunately most visual components do not support this data type.

- LONGNUM=COMP

With this option, big integer numbers will be converted to Delphi 64-bit integer.
This option can be combined with other (for example CONVERSION RULES=
By default this option is enabled from version 5.0 of the BDE (disabled is other cases).

- LONGNUM=NOCOMP
- SMALLPACK=integer

By default, a small packed or zoned field (under 5 digits with no decimals) are converted to the Delphi type ‘smallint’. When setting this option, it makes these fields to be converted to integers.

Example of use :
CONVERSION RULES=LONGNUM=DOUBLE;SMALLPACK=INTEGER

CMTCTL

Set it to TRUE if you want to work with transactions.
You will be able to use BeginTransaction. Default is FALSE.

EACUNLOCK

This option allows advanced protection of your Client/Server programs.
See section 'Easycom For Delphi program protection' for more information.

LANGDRIVER

This is the default language driver for your PC. Default is ANSI.

LOCKTYPE

This option is used to choose the way of locking records when using physical/logical file accesses.
Two possible values are :

- PESSIMISTIC (default)

With this value, the record is locked during the whole process of editing a record.
For example, with Delphi the record is locked with the Table.Edit and release with Table.Post.
So if different users try to modify the same record, other users must wait for the record post.

- OPTIMISTIC

With this value, the record is only locked when trying to update the record.

If the record has been modified by another user since the Table.Edit, the update is canceled and a Bde exception is raised.

LOCKTYPESQL

This option is similar to LOCKTYPE, but is application to SQL queries.

With SQL, when using this option with 'OPTIMISTIC' setting, SQL statements will always be opened read/only. Updates, Delete and Insert orders will be performed by separated SQL statements. Default value is PESSIMISTIC.

PRIVATE SESSION

Set it to TRUE if you want that each connection (by a TDatabase Component) must be one different connection on the same AS/400.

Default is FALSE.

LOG LEVEL

This allows you to make a text trace of operations that are made by the server. If 0, no trace is made.

Default is 0.

LOG FILE

Represents the file on the AS/400 that will be created if LOG LEVEL is not zero.

OPTIONS

This is the general-purpose alias option.

The syntax of the value is a semicolon-separated list.

Example :

LOCALFILTER=TRUE;DEFAULTNULL=TRUE

Supported sub options are :

CRTTABLE SCRIPT=<path>

This option creates an SQL script for tables and index creation. With this option enabled, all file creation will be performed with SQL (otherwise we will use DDS in case of non-using of long table names).

<path> is the name of the script file to create.

LOCALFILTER=TRUE

This option allows a local processing for filters. It is a useful value if the application filters only small files. It means that all data (non filtered) will be transferred on PC side, and then filtered by the ACE400 driver.

Otherwise, the filters will be performed on the AS/400 side (by an OPNQRYF). With server-side filters (default behavior), only one filter at a time is allowed for each access path (physical/logical).

Note : you also can set-up this option for each file with the PROPERTIES options. Default value is FALSE.

DEFAULTNULL=TRUE

This option allows to have null values by default for each field. Otherwise, you will get blank values for each field (except for a Date field that will get null if the field is null-capable or 1/1/0001 if not null capable).

This option will have an effect only on Null-capable fields.

Default value of this option is FALSE.

USEALIASES=TRUE

By default, ACE/400 will use field alias names as long field names. This provides a easy migration for an SQL application to an ACE/400 application.

But for backward compatibility purposes, you might want to not use the alias names (and then use the regular 10 digits field names). You can set this option to FALSE to do this.

Default value for this option is FALSE.

CACHEDSQL

This allows to have a local SQL (or storedprocedure) result. This make multi-selection easier, and prevents for unwanted refresh.

The syntax is :

CACHEDSQL=n/ALL<,m<,MIN/MAX>>

n is the upper limit of the number of rows that we can hold in memory for a query.

ALL value means no limit.

m is the number of rows that we can read in one time. Default is 32kb of data.

MIN means that m cannot be greater.

MAX means that is m is a too small value (less than 32Kb of data), the value will be increased to have 32kb of data. Default is MIN is m is specified.

Here are sample of syntaxes :

CACHEDSQL=ALL ==> queries and stored procedure resulsets will be hold completely in memory (if the use pushes the 'LAST' button). Data will be fetch by 32kb of data.

CACHEDSQL=ALL, 100 ==> queries and stored procedure resulsets will be hold completely in memory (if the use pushes the 'LAST' button). Data will be fetch by 100 records

CACHEDSQL=ALL, 100, MAX ==> queries and stored procedure resulsets will be hold completely in memory (if the use pushes the 'LAST' button). Data will be fetch by 100 records or by 32kb of data if the record size is small

CACHEDSQL=10000, 100 ==> queries and stored procedure resulsets will be hold completely in memory (if the use pushes the 'LAST' button). Data will be fetch by 100 records. If there are more than 10000 rows in the result, the cached SQL will abort, and dynamic SQL will be used instead.

This CACHEDSQL parameter is in the 'OPTIONS' BDE parameters. So you can have :

OPTIONS=

If you want to combine Cached and nonCached queries, you need to define two TDatabase components (the connection can be shared).

If you define a 'PESSIMISTIC' request live query, the option will have no effect.

JOBNAME=

BIGPARAMTYPE=MEMO/CHAR

VARLENGTH

This option allows to customize AS/400 Variable Length fields handling.

If you put 'VARLENGTH=MEMO', it will convert it to Delphi's memo fields.

By default, it will convert it to simple Delphi's string fields.

STOREDPROC=SQL

SQLCAUSISTANTBM=TRUE

Additional treatment for better stability of DBGrid, in multi-selection mode.

Misc. configuration options

Additional configuration options are provided, for special behaviors.

Some of them are only provided for backward compatibility with LightLib/400.

These options are in easycom.ini file.

- - Oldconversion, in [general] section.
  This allows to use old conversion behavior that was in LightLib/400.
  The numeric fields (packed or zoned) will always be converted into double values or character values (if above the precision of double).
  Put Oldconversion=1 to enable this.
  Default is 0

- - ThreadAttach in [Multithread] section and NbSession in [NT] section.

By default, with multithread, the AS/400 connections are shared with the threads.

To automatically generate one connection per thread, add in easycom.ini file:

[MultiThread]

ThreadAttach=1

[NT]

NbSession=10

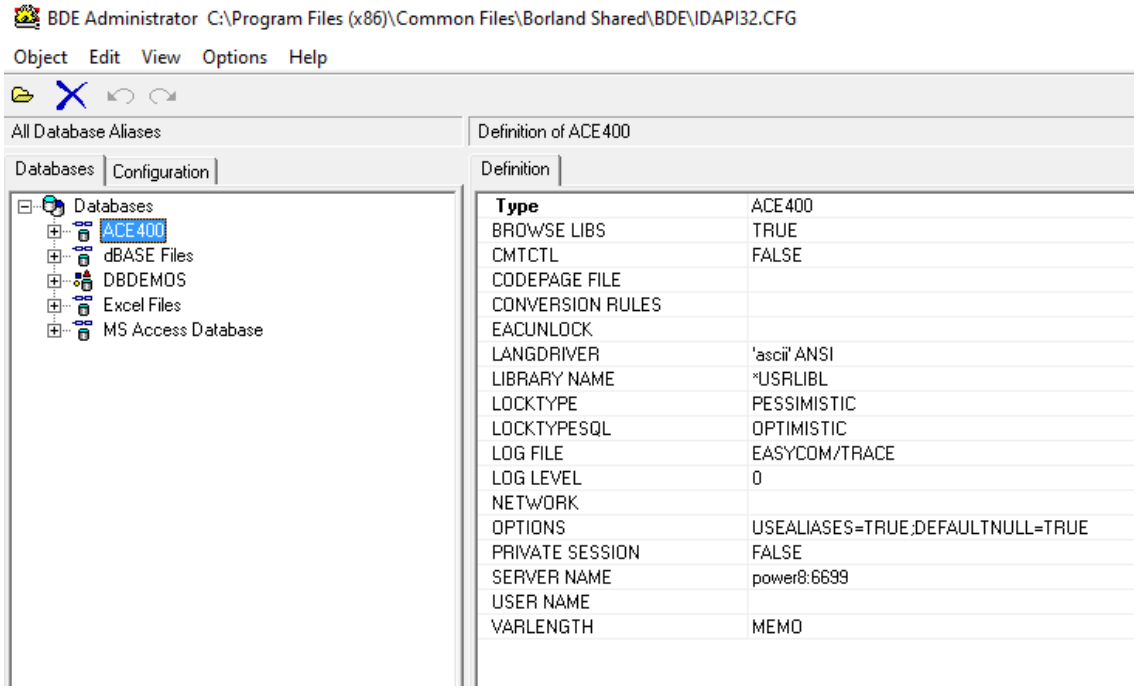
Default is 0 for ThreadAttach and 1 for NbSession.

Nbsession is the maximum number of AS/400 connections. It can be illimited (but allocates some more memory).

Because most of APPC routers are not thread-safe, Multithreading is only safe and supported on TCP/IP.

Connection

The connection to BDE is done by using the ACE400 database, defined into the BDE Administrator:



Testing Data Access

Before running the **Easycom For Delphi** samples using BDE, you must be sure your PC is properly connected to the AS/400.

BDE samples are available here :
C:\Users\Public\Documents\Easycom For Delphi samples\db

If you were able to run the EASYCOM installation and transfer files onto the AS/400, this can be taken as a good sign, but it's not a guarantee.

You could still experiment connection errors, depending on the setup of drivers and the AS/400 communications layer.

Launch Delphi. Create a New Application.

Drop a TTable component onto the main form.

Set the DatabaseName to ACE400.

Set the Table Name to SP_CUST in the EASYCOM library using the Dialog. Once you select it, a "real" connection will be established to the AS/400, which could take a while. When prompted for a Database password, press enter.

Note : The dialog fills the combo box list of the Table Name property. Once this is done, you still need to select one of the items.

Set the Active property to True.

Drop a TDataSource component and connect it to the TTable.

Drop a Grid component, connect it to the DataSource, you should see the AS/400 data in the grid.

If you have been successful with these steps, then congratulations, you have just created your first Delphi AS/400 application ! You are ready to move on and do some real work with the AS/400 and Delphi. But first, we would suggest that you first take a look at the sample applications provided with **Advanced Client Easycom/400 for Delphi**.

Please notice that the Logical files associated to a physical file are visible in the IndexName and IndexFields.

If you did not get data displayed in a grid, then there's no point in trying the samples as they will fail.

Carefully check all of the installation steps, on the Delphi side (aliases ...), then, on the AS/400 side, check if you are receiving any error messages, and check the connectivity layers.

Use the Benchmark and Serialization programs to retest communications.

Field type management

All kinds of AS/400 field types are supported. For the field types that don't exist on AS/400, they are converted to the nearest field type.

Null capability is also fully supported : you can get or set null field information with tables and queries. You can even make a key search with null values.

Depending of the AS/400 type of a field, it is converted (in read and write) to a PC type. Conversion between AS/400 types and PC types are made by the client.

You can notice that PACKED or ZONED fields are optimally mapped to a type depending on the size.

From Bde version 5 PACKED or ZONED field that have no decimal but a number of digits between 16 and 18 are mapped to 64 bits integers.

AS/400 type	Size	Dec	PC Type
CHAR	Any		ZString (pChar)
BIN2			INT16

BIN4			INT32
PACK or ZONED	1->4	0	INT16
PACK or ZONED	5->9	0	INT32
PACK or ZONED	10->15	Any	DOUBLE
PACK or ZONED	Other cases		ZString (pChar)
DOUBLE			DOUBLE
FLOAT			DOUBLE
DATE			DATE
TIME			TIME
TIMESTAMP			DATETIME

Easycom For Delphi Properties

You may want to set-up special properties for particular AS/400 files.

Having a TTable, a TDatabase or TQuery component, some additional properties can be set-up.

You will use DbiSetProp/DbiGetProp with the Handle property of your component.

For example :

```
DbiSetProp(hDBIObj(Table1.Handle), curNRECORD, 50) ;
```

Current available properties for TTable/TQuery are :

```
const
dbCONVRULES =  $0404ACE5 ;
curNRECORDS =  $22050000 ;
curLOCALFILTER =  $22050001 ;
curLOCKTYPE =   $22050002 ;
```

dbCONVRules value is used to change conversion rules.

This as the same features as the CONVERSION RULES option of the ALIAS.

Current possible values is a combination of integers :

```
const
CONVR_LNUM_DOUBLE = 1 ;
CONVR_SMALLPACK_INTEGER = 2 ;
CONVR_USEBIG_BCD  = 4 ;
CONVR_COMP = 8 ;
```

CONVR_LUM_DOUBLE conrresponds to LONGNUM=DOUBLE

CONVR_SMALLPACK_INTEGER conrresponds to SMALLPACK=INTEGER

CONVR_USEBIG_BCD conrresponds to LONGNUM=BCD

CONVR_COMP conrresponds to LONGNUM=COMP

curNRECORDS is used to change default number of records that are used for internal cache. Default value is 30.

curLOCALFILTER corresponds to LOCALFILER=TRUE in the OPTIONS ACE/400 option alias.

curLOCKTYPE corresponds to the LOCKTYPE and LOCKTYPESQL alias options.

value 0 correponds to PESSIMISTIC (default)

value 1 correponds to OPTIMISTIC

Specific Components

Components of Easycom For Delphi

The components on the ACE/400 of the Component palette make specialized AS/400 access functionality available to your Delphi applications.

All standard database tasks are performed with the native AS/400 driver ACE400.
These components are designed for AS/400 specifics purposes only.

To use these components, it may be necessary to copy files from Easycom installation directory to the DCP directory.

For instance :

C:\Users\Public\Documents\Embarcadero\Studio\21.0\Dcp : for Delphi 10.4 Sydney.

C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp : for Delphi 11 Alexandria.

For instance for SYDNEY 10.4, copy all files from:

C:\Program Files (x86)\ACE400\comp\Delphi10.4Sydney

to:

C:\Users\Public\Documents\Embarcadero\Studio\21.0\Dcp

For instance for Alexandria 11, copy all files from:

C:\Program Files (x86)\ACE400\comp\Delphi11Alexandria

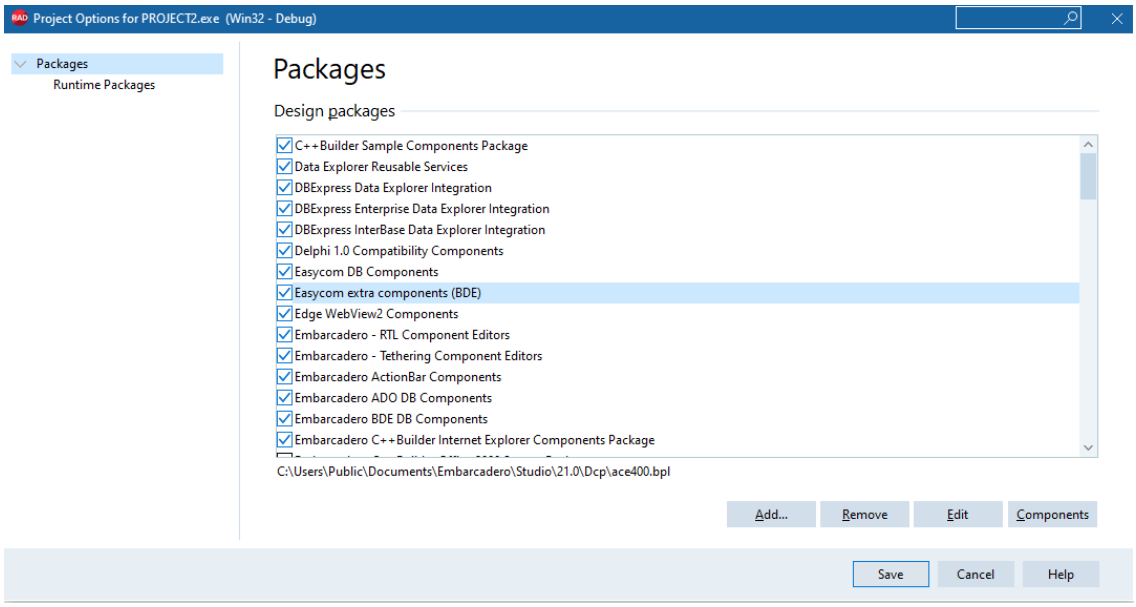
to:

C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp

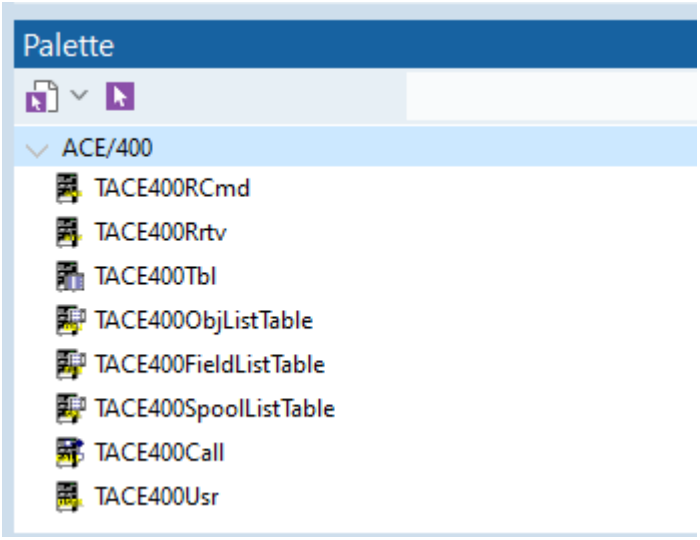
Then, for Delphi 11 Alexandria, from the IDE select “Component->Install packages…”, and select “**Easycom extra components (BDE)**”.

It corresponds to :
C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp\ace400.bpl.

Finally, click over the button "Add...".



The components are therefore visible into the IDE:



Overview of components :

TACE400Rtv	This component allows AS/400 commands that returns a value, such as Rtvsysval. From your application, you can send a command to your AS/400, and receive values from the command.
TACE400RCmd	This component allows remote command to the AS/400. It can be used to use simple commands like ADDLIBLE or program call (with CALL command).
TACE400Tbl	This component allows to update the 'DisplayLabel' properties of a TTable component to be update from the column headers.
TACE400ObjListTable	This component allows the listing of AS/400 objects. Typical application is listing files. However this component can be used to list any type of AS/400 objects.
TACE400FieldListTable	This component allows to get full information about AS/400 fields. It will retrieve full AS/400 information, such as column headings, AS/400 types, ...
TACE400SpoolListTable	This component allows the listing of spool files.
TACE400Call	This component allows to call AS/400 programs, with complete input/output features.
TACE400Usr	This component allows to change current user at **run time**.

ACE400Rtv

Description

This component allows AS/400 commands that returns a value, such as Rtvsysval.

From your application, you can send a command to your AS/400, and receive values from the command.

How to use

Step 1 : Fill the Command property. It can contains some variable declarations.
Step 2 : Run the execute function.
Step 3 : Examine the resulting variables with FieldByName method.

Example : Retrieve a Data Area value

```
JobRcvMsg.command:='dta=char(200);rtvdtaara dtaara(*lda) rtnvar(&dta)'
JobRcvMsg.execute;
MsgToSendToJob.Text:= trim(JobRcvMsg.Fieldbyname('dta').AsString);
```

Properties

Command

property Command:TStrings;

Description

This property holds the text of the command. It can also contain declaration of variable types and sizes.
Here is the general syntax :

<decl1>;<Decl2>...;<Decln>;MYCOMMAND &PARM1 (&Decl1), ...

Each item are separated by a semicolon. Declarations can have the form :

NAME=CHAR(xx)
or NAME=DEC(xx yy)
Default type of a variable is CHAR. You must specify a size of a character variable if it exceeds 20 characters.

Fields

property Fields[index:integer]:TACE400Field;

Description

This function is used to retrieve result value(s).

CommandText

property CommandText:string;

Description

This property has the same purpose as the Command Property. It is an easiest way to setup the command at run time.

For syntax of the command, see description of Command Property.

Methods

Execute

procedure Execute;

Description

This function executes the command.

If the execution fails, it generates an exception.

FieldByName

function FieldByName(const FieldName:string):TACE400Field;

Description

This function is used to retrieve result value(s).

This allow an easy access to values, like this :

Result := MyACE400Rtv.FieldByName('MYVAR').AsString;

ACE400RCmd

Description

This component allows remote command to the AS/400.

How to use

Fill ‘Commands’ property with the commands (one command each line). Then call ‘Execute’ method.

The execute function will return the number of succeeded executed commands.

Example of use

```
{ Add libraries in the LIBL with TACE400Rcmd Component }
ACECmd..Commands.Clear;
ACECmd.Commands.Add(‘ADDLIBLE MYLIB1’);
```

```
ACECmd.Commands.Add('ADDLIBLE MYLIB2');
If ACECmd.Execute <> 2 Then
begin
// error
end;
```

Properties

Commands

property Commands:TStrings;

Description

This property holds all the commands you want to send the AS/400.

Each item of this property represents a command.

NoWaitObject

property NoWaitObject:TOleControl;

Description

This property is used to link the component to an TACE5250 component.

This is useful when having an ACE5250 terminal component running in the same job than easycom.

This allow to have end notification in TACE5250 component after having used TACE400Rcmd.ExecuteNoWait.

StopOnError

property StopOnError:boolean;

Description

This property is useful only if you plan to put multiple commands in the component.

If this property is true, the component will stop processing if one command fails.

Otherwise, it will ignore all commands failing.

The number of successfully executed commands is returned by the TACE400Rcmd.Execute method.

Methods

Execute

function Execute:integer;

Description

This function executes the commands that are in Commands property. It returns the number of the commands that were successfully executed.

ExecuteNoWait

procedure ExecuteNoWait;

Description

This procedure executes the command that is in the Commands property (for this function, only one command is allowed).

It is designed to be used only in conjunction with a 5250 emulation component.

It can indeed handle interactive commands (such as WRKACTJOB).

This only can work if the 5250 terminal is working in the same job than your client/server application.

The function returns immediately without waiting command completion.

All BDE components using the TACE400Rcmd.DatabaseName are unavailable during command processing :

only the 5250 is useful there.

End notification will be sent to the 5250 emulation component by TACE5250.RcmdComplete Event.

The 5250 component that will recieve notification must be linked to the TACE400Rcmd component using

TACE400Rcmd.NoWaitObject property.

Steps to follow to use this function :

- fill TACE400Rcmd.Commands with your interactive command
- point TACE400Rcmd.NoWaitObject to your TACE5250 component
- make your 5250 working in the same job (by using Easycom Protocol or by using EACHOST).
- point TACE5250.DatabaseName to the same alias than your C/S application.
- use TACE400Rcmd.ExecuteNoWait
- show your 5250 window to let the user work with it
- intercept the TACE5250.RcmdComplete Event to show again your C/S application.

ACE4Tbl

Description

This component is essentially a design-time component.

It allows to update the 'DisplayLabel' properties of a TTable component to be update from the column headers.

How to use

Put a TTable linked with ACE/400 on the Form. Connect it to an AS/400 file that has column h headers in it (you can check it with a DSPFFD on the file). Use the field editor of the TTable to add the fields you want to use with the file.

Then put a TACE400Tbl component on the form. Link it to the TTable component. Then double-click on the TACE400Tbl component, and all the display labels are updated !!!

At the end, the component can be deleted, or used to update another TTable component.

Properties

ACEDataSet

property ACEDataSet:TDBDataSet;

Description

This property points to the TTable component that will be updated.

Methods

UpdateFields

procedure UpdateFields;

Description

This function is used to update the display labels at run time.

ACE400ObjListTable

Description

This component allows the listing of AS/400 objects.

Typical application is listing files. However this component can be used to list any type of AS/400 objects.

How to use

Fill all the properties (don't forget the DatabaseName property). Then link the component to a TDataSource, and then to a data component such as a TDBGrid.

ACE400FieldListTable

Description

This component allows to get full information about AS/400 fields.
It will retrieve full AS/400 information, such as column headings, AS/400 types, ...

How to use

Fill all the properties (don't forget the DatabaseName property). Then link the component to a TDataSource, and then to a data component such as a TDBGrid.

Properties

FldFile

property FldFile:string;

Description

This property holds the AS/400 file name.

FldFormat

property FldFormat:string;

Description

This property holds the object format to examine.

Possible values are :

- a Format Name
- *FIRST for the first format.

FldLibrary

property FldLibrary:string;

Description

This property holds the AS/400 library name. This value can be *LIBL or a full qualified library name.

ACE400SpoolListTable

Description

This component allows the listing of spool files.

How to use

Fill all the properties (don't forget the DatabaseName property). Then link the component to a TDataSource, and then to a data component such as a TDBGrid.

Properties

SplUserName

property SplUsername:string;

Description

This options allows to filter the user name. This can contain a user name, or *ALL.

SplOutputQueueName

property SplOutputQueueName:string;

Description

This parameter specifies the AS/400 Output queue name. This can be for example QPRINT, *LIBL/QPRINT or any queue name.

SplFormType

property SplFormType:string;

Description

This property holds the form type selection. A value for this parameter can be for example *STD or *ALL.

SplUserSpecData

property SplUsername:string;

Description

This options allows to filter the user special Data. This can be any value or *ALL.

SplFullFormat

property SplFullFormat:boolean;

Description

The listing can have two formats. If this property is true, the result will include all the fields.

If this property is false, the result will only include the first 7 fields.

ACE4Call

Description

This component allows to call AS/400 programs. This can also be done with TACE400Rcmd or TACE400Rrtv, for simple calls.

This component is inherited from TStoredProc Component.

This component allows full-feature remote procedure call.

To call a program, proceed in 3 steps :

1. At design time, describe the procedure call.
2. At run time, prepare the call (it will bind the parameters definition).
3. Setup input parameters and call execute method.

The step 3 can be iterated as needed.

ACE400Usr

Description

This component allows to change current user at run time.

How to use ?

Step 1 : Fill the username and password

Step 2 : Put the active property to true.

Be careful: To enable this property, you must provide access rights to the objects QSYGETPH, QWTSETP, and QSYRLSPH.

By default this objects are only accessible to QSECOFR and QSYS users. Change the rights of this objects only if you need to use this component.

Properties

Active

property Active:boolean;

Description

This property triggers the user change. If it fails, it will throw an exception. If it succeeds the current right will belongs the users specified by the UserName property.

Password

property Password:string;

Description

Specifies the password of the user to change to. Be careful, if you fill this property at design time, the password will appear in clear during development.

This property can of course be used at run time (just after the Database connection).

Username

property UserName:string;

Description

Specifies the username to change to.

The Easycom Native Driver (without BDE)

The Easycom Native driver is the first, and probably easiest way to access AS/400 data and programs using Easycom.

The Easycom Native components are visible on the "EASYCOM" tab in the tool palette.

Those components are not using the BDE stack level, but have a specification very close to the BDE components: Ttable, Tquery, ...

The deployment is much more easier than with the BDE.

To use these components, it may be necessary to copy files from Easycom installation directory to the DCP directory.

For instance:

C:\Users\Public\Documents\Embarcadero\Studio\21.0\Dcp : for Delphi 10.4 Sydney.

C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp : for Delphi 11 Alexandria.

For instance for SYDNEY 10.4, copy all files from:

C:\Program Files (x86)\ACE400_SYDNEY_10.4\comp\Eac\Delphi10.4Sydney

to:

C:\Users\Public\Documents\Embarcadero\Studio\21.0\Dcp

For instance for Alexandria 11, copy all files from:

C:\Program Files (x86)\ACE400\comp\Eac\Delphi11Alexandria

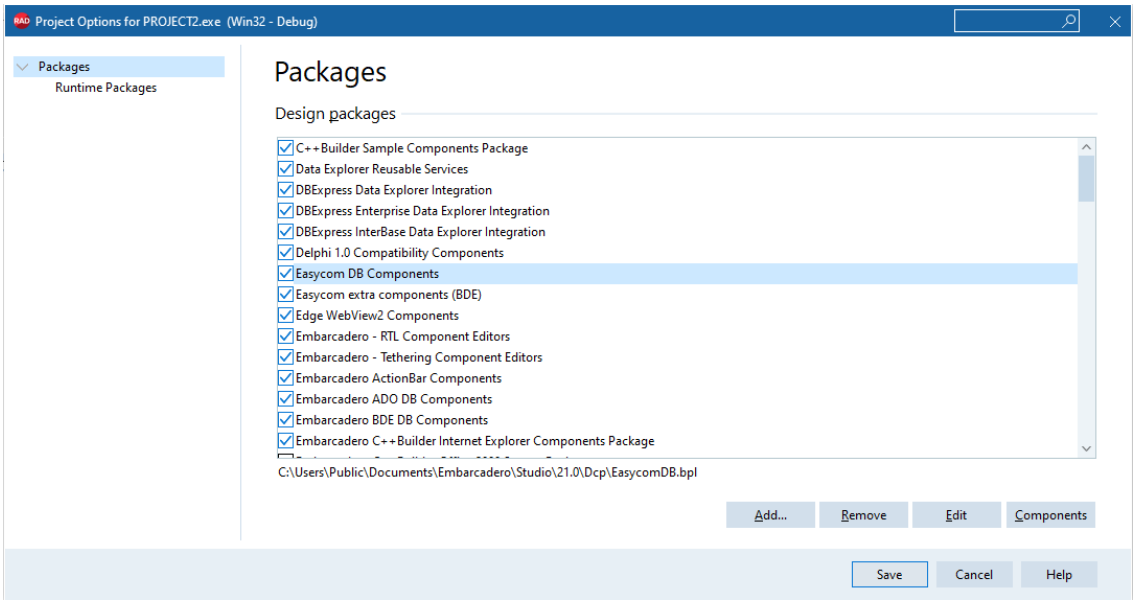
to:

C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp

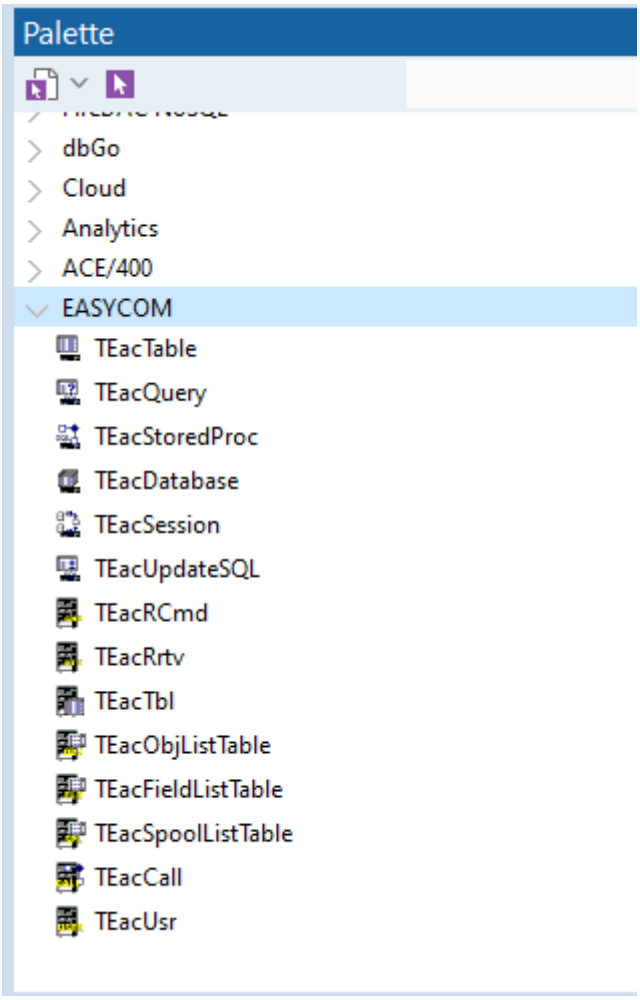
Then, for Delphi 11 Alexandria, from the IDE select “Component->Install packages…”, and select “**Easycom DB components**”.

It corresponds to : C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp\EasycomDB.bpl.

Finally, click over the button "Add...".



The components are therefore visible into the IDE:



The class names are different. When you have "TTable" in the BDE components, you will have "TEacTable" with Easycom. TACE400Rcmd becomes TEacRcmd, and so on.

However, the property and method names are identical. So porting a BDE application to those components is an easy task.

The BDE administrator is replaced by a simple, not mandatory, .ini file. This .ini file will contain the same information that were in the BDE aliases. Here are the default .ini file contents :

;

; EasycomAlias.ini

;

[Aliases]

Alias1=EasycomAlias

[EasycomAlias]

OPTIONS= LOCALFILTER=FALSE;DEFAULTNULL=TRUE;USEALIASES=TRUE

SERVER NAME=power8

So you need to fill an "AliasXX" entry in the [Aliases] section, and fill all the properties you want.

To have the documentation for TEacTable, TeacQuery, TeacStoredProc, TeacSession, TEacDatabase, TEacUpdateSQL, please refer to the Borland documentation.

64-bit application and the Easycom Native Driver (without BDE)

Since **Delphi 11 Alexandria (and only this version)**, it is possible to use Easycom Native components into a **64-bit** application.

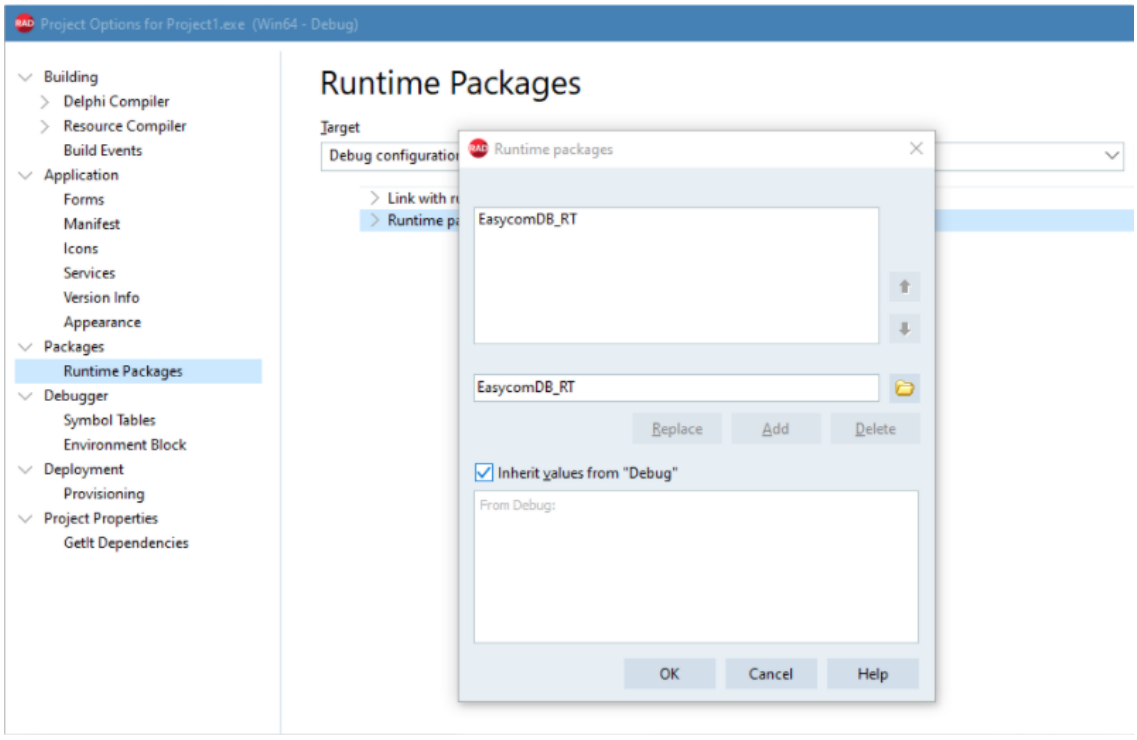
64-bit Runtime components are available here into the installation directory :
C:\Program Files (x86)\ACE400\comp\Eac\Delphi11Alexandria\RT_64bits

To use these components, it may be necessary to copy files from the Easycom installation directory to the following DCP directory :
C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp\Win64.

For **Delphi 11 Alexandria**, copy all files from:
C:\Program Files (x86)\ACE400\comp\Eac\Delphi11Alexandria\RT_64bits
to:
C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp\Win64

With the 64-bit configuration of your project, open the options and go to Packages> Runtime Packages, and add the following Easycom 64-bit Runtime package:

C:\Users\Public\Documents\Embarcadero\Studio\22.0\Dcp\Win64\EasycomDB_RT.dcp



After this step, save your project, do a clean and test your project.

If you encounter the following error:

[dcc64 Fatal Error] Unit1.pas (6): F2613 Unit 'Messages' not found.

Add Winapi:

uses

Winapi.Messages

32-bit Applications deployment

You will need to deploy the following DLL (available here : C:\Windows\SysWOW64) with the Delphi application:

In all cases:

Easyco32.dll

Easyco32.fr (for French support of Easycom dialog boxes).

Eac32slf.dll

Eac32prc.dll

If you use Easycom Native Driver without BDE:

EasycomIdapi.dll

If you use Easycom IDAPI Driver with BDE:

Ace40032.dll

The BDE has to be operational on PC where application is deployed.

For the deployment you can avoid the usage of the .ini file (this is recommended), and use the "DriverName" property of a TEacDatabase component.

You also might want to deploy the easycom ini file (EasycomAlias.ini and Easycom.ini).

64-bit Applications deployment

64-bit applications have to use only the Easycom Native components (without BDE).

You will need to deploy the following DLL (available here : C:\Windows\System32) with the Delphi application:

Easyco32.dll

EasycomIdapi.dll

AS/400 Functions

Stored procedure configuration

AS/400 programs do not have any external description.

The program expects data in parameters, and returns results through the same parameters.

Only the program itself knows about the type and size of those parameters.

You need to describe your program, **because Easycom For Delphi** needs to know how to translate the information from AS/400 into PC format.

If your program has a variable structure depending on the value of a parameter during the CALL, you can provide multiple descriptions for the same program.

Each description will have its own procedure name.

The EASYCOM Stored Procedure Configuration tool allows you to describe any program on the AS/400 by giving the type and size of each parameter it expects to receive.

You can describe any AS/400 program written in any AS/400 language (RPG, COBOL, CL, C, etc.).

The only restriction is that the program must not do any screen or keyboard I/O, because it will be treated as a stored procedure, and will be an extension of the Easycom server.

RPC - DTAQ Configuration tool

You can start the RPC/DTAQ Configuration tool from the shortcut in the program group “ACE400”.

See : Describing AS/400 native programs

Program description

1. Start the « EASYCOM Stored Proc Configuration » , in the program group « ACE400 ».

2. Create a new procedure.

3. Complete the dialog box with the Procedure description :
   - **Procedure Name** : This name will be the only one used on the client side. Your application will open and call this procedure name.
   - **Description** : Enter any text to describe your procedure.
   - **AS/400 Program Name :** The complete AS/400 qualified program name. LIBRARY/PROGRAM
   - **Program/DataQueue :** Leave this Combo Box in Program state.

4. Describe the parameters expected by your program.

Because on AS/400, a parameter can be a Data Structure, you need to describe each Field (item of the Data Structure), and to put the type of the field (parameter or first field of a Data Structure or not).

You can Add, Delete or Modify field descriptions by clicking the "+", "-" or "M" buttons.

Complete the next Dialog box "Parameter field edition" :

- **Field Name :** The field to be used.
- **Field usage :** The field is sent to the program (IN), or it is the return result of the program (OUT) or both. Usually the parameters are IN/OUT.
- **This field is a parameter or the first field of a Data Structure :** This box must be unchecked only if the field is in the second or greater position inside a parameter that is a Data Structure.
- **AS/400 format :** Describe exactly which data format your program expects : Data Type, Number of characters or digits for a decimal value, and the number of decimals.
- **PC format :** Describe how you want the data to be converted by Easycom.

You can use this process to describe several programs, also you can provide multiple descriptions for the same program, if needed.

See StProc sample provided for a complete example.

DataQueues description

You can also create procedures for all type of known AS/400 Data Queues : simple file (FIFO) or keyed file (KEYED).

“DTAQ_FIFO” is an example based on simple dataqueue while “DTAQ_KEY” is an example with keyed procedure

1. Start the “ Easycom Stored Proc Configuration ” in the program group “ ACE400 ”.
2. In the “ Stored Procedure configuration ” window, select “ New Procedure ”.
3. Complete the dialog box "Procedure definition" :
   - **Procedure Name** : This name will be the only one used on the client side.
   - **Description** : Enter any text to describe your DataQueue,
   - **AS/400 Progam Name** : Path (<Library>/ <Program>) to the DataQueue,
   - **Program/DataQueue** : "DataQueue".
4. Describe the fields of your Data Queue. Using “ **+** ”, “ **–** ” or “ **M** ” buttons, you can add, delete or modify fields.

In the description of the DataQueue format, you have to create three fields used by **Advanced Client Easycom/400 for Delphi**.

- TIMEOUT : ‘char’ type , size 6 Time-out reading delay,
- FILER : ‘char’ type , size 2 Future use,
- ORDER : ‘char’ type, size 4 Kind of key operation (DataQueue KEYED),

In case of Keyed Data Queue, fourth field is the key

Complete the next Dialog box "Parameter field edition" :

- **Field Name** : Virtual name of the field,
- **Field usage** : No meaning for the DataQueue.
- **This field is a parameter or the 1st of a Data Structure** : No meaning for the DataQueue,
- **AS/400 format** : Describe exactly which data format your program expects : Data Type, Number of characters or digits for a decimal value, and the number of decimals,
- **PC format** : Describe how you want the data to be converted by Easycom.

You can use this process to describe several Data Queues, and you can provide multiple descriptions for the same Data Queue, if needed.

Examples

Some sample Delphi programs have been installed here:

C:\Users\Public\Documents\Easycom For Delphi samples

Advanced Security Features

Advanced Security Features with TCP/IP

By default the EASYCOM Job is launched by the daemon program EASYCOMD running in the EASYCOM subsystem.

EASYCOMD searches for a program named EACTCP003 that allows to customize connection check. This program EACTCP003 can be in the library list or in the Easycom library.

So, it is possible to customize security by simply creating a EACTCP003 program.

EACTCP003 program interface (CL)

```
PGM
   PARM(&TPPGM &TPLIB &USER &EAC_PARM1 + &EAC_PARM2
   &RMT_ADR &JOBNAME)

DCL  VAR(&TPPGM) TYPE(*CHAR) LEN(10)
DCL  VAR(&TPLIB) TYPE(*CHAR) LEN(10)
DCL  VAR(&USER) TYPE(*CHAR) LEN(10)
DCL  VAR(&EAC_PARM1) TYPE(*CHAR) LEN(30)
DCL  VAR(&EAC_PARM2) TYPE(*CHAR) LEN(30)
DCL  VAR(&RMT_ADR) TYPE(*CHAR) LEN(50)
DCL  VAR(&JOBNAME) TYPE(*CHAR) LEN(10)
```

Short description of each parameter

- **TPPGM** : Target Program Name

Name for the server program that will be launched. Default is Easycom.

- **TPLIB** : Library where TPPGM is

Location of the server program.

- **USER** : User Name

  User Name for the new connecting client (can be used to allow only a set of users)

- **EAC_PARM1** : Parm 1 to send to the TPPGM

Parameter to give if you make the SBMJOB yourself.

- **EAC_PARM2** : Parm 2 to send to the TPPGM

Parameter to give if you make the SBMJOB yourself.

- **RMT_ADR** : TCP/IP adrress of the caller

  TCP/IP Address of the caller (can be used to allow only a set of workstations)

- **JOBNAME** : Job Name for the SBMJOB

  Name for the job that will be in the Easycom subsystem. Default is workstation name.

  If this JOBNAME is changed to *NO, the EASYCOM job will not be started.

  If this JOBNAME is changed to *YES, the EASYCOM job will be accepted, but not the EASYCOM job is supposed to be started by EACTCP003 program.

Example

EACTCP003 can check the authorization of the user and his TCP/IP address, and then it can do one of the following:

- Give authorization to run:

EACTCP003 leaves the variable &JOBNAME as it is, or change it to another job name, and then exit normally.

EASYCOMD will launch the target program with this new name.

- Denny the execution:

EACTCP003 changes the value of &JOBNAME to '*NO', and then exit normally.

```
CHGVAR VAR(&JOBNAME)  VALUE('*NO')
```

- Launch itself the target program:

EACTCP003 launches the target program by a SBMJOB and changes the value of &JOBNAME to '*YES'.

```
CHGVAR VAR(&JOBNAME) VALUE('*YES')
SBMJOB CMD ( CALL  PGM(&TPLIB/&TPPGM)+
                   PARM (&EAC_PARM1&EAC_PARM2))
           JOBD(&TPLIB/EACJOBD) +
           USER(&USER) RTGDTA(*JOBD)
```

Easycom For Delphi program protection

In some situations, protection is not made on AS/400 files, but on AS/400 programs (some users will be only able to use some programs).

This will allow to restrict the access of the AS/400 to a specific selection of PC programs. This restriction is configured on the AS/400 side.

In other words : by giving the right password(s) to the right developers, you can avoid unauthorised developers to write unauthorised programs.

So, you can set-up a controlling program that will receive the EAC_UNLOCK value in a parameter. So, application that do not know the appropriate value, will not be able to use ACE/400.

Here are details on how to use this feature.

You can do the following steps :

Step1 : Configure EASYCOM server in a LOCKED mode

It is done with the CFGEAC command on AS/400.

Do :

```
CHGCURLIB EASYCOM
CFGEAC
```

It will show you the following screen :

```
Easycom server library name  EASYCOM
Easycom server job priority  0
TCP/IP Keep Alive frequency  120
Delay before asking again pwd  0
Delay before automatic SIGNOFF  0
Easycom Log File level   0
Print the clock in Log File  *NO
Automatic Keep Alive start  *YES
Detailed Job Log   *NO
Lock Easycom host   *YES
```

If the option ‘Lock Easycom host’ is set to *YES, by default no files can be accessed and no AS/400 programs or commands can be executed. The Easycom server can be unlocked remotely by a password sent by the PC application.

Step 2 : Make a validation program

When the password is sent from the PC, the Easycom server will call a program named EACP003 in *LIBL.

Here is a sample on how to make the validation program.

```
PGM PARM(&PASSW &RESULT)
DCL VAR(&PASSW) TYPE(*CHAR) LEN(100)
DCL VAR(&RESULT) TYPE(*CHAR) LEN(10)
…
/* IF PASSW HAS THE RIGHT VALUE */
CHGVAR VAR(&RESULT) VALUES('*YES')
…
/* IF PASSW DOES NOT HAVE THE RIGHT VALUE */
CHGVAR VAR(&RESULT) VALUES('*NO')
```

Step 3 : Put the password in your Delphi Application

Use the EAC_UNLOCK option to put the right unlock password. You can set-up this option using the TDatabase component in your application.

Because EACP003 is called by EASYCOM, you also can control the current user (by a RTVJOBA in EACP003). You also can decide to assign an unlock password for a set of users.

You also can decide to use a different password for each PC application. Doing this you can decide in the EACP003 program if a particular user can use a particular program (because the password will identify the PC program).

Important note :

Be aware of TBatchMove component limitations.

TBatchMove does not create indexes for the destination table, and field types are automatically matched with source field types, which can lead to problems trying to create non-supported field types in the AS/400 or transferring empty date fields.

You will probably use native AS/400 tools to create and manage AS/400 tables, but we wanted to provide you with maximum compatibility with existing Delphi components.

EASYCOM Server

See Easycom server documentation.

Appendices

Common Problems

Common Delphi problems

The Delphi VCL directory must be in your path, if you exceed the path limit you'll need to map a drive or consolidate VCLs in a separate directory to ensure they are all found.

Using Language Conversion Tables

In some countries (i.e. Israel), you would need to use translation table. This is the case when the AS/400 or the PC does not produce US/LATIN EBCDIC or ANSI characters. This will allow Delphi DataAware controls to work properly.

To use the conversion table, modify the “CODEPAGE FILE” in the alias configuration as follow :

CODEPAGEFILE …MyDirectory\MyPath\MyConversionTableName

Easycom For Delphi comes with a set of conversion tables. They all follow the naming convention :

Table name convention :

EnnnAmmm.tbl

Where

nnn is the EBCDIC code page.

mmm is Ascii code page. Ammm can be replaced by ANSI.

Ex : C:\ACE400\CodePage\e297ansi.cpg

The contain of a table is :

Bytes 0 to 255 = EBCDIC to ASCII Conversion.

Bytes 256 to 511 = ASCII to EBCDIC Conversion.

The value of the character will be used as an index in the table.

Example:

To translate character '0':

'0' = 0xF0 (240) in EBCDIC

'0' = 0x30 (48) in ASCII

Offset 240 of the file contains value 48.

Offset 560 (512+48) contains value 240.

If the alias is modified in such a way that the translation may not generate an ANSI character set. So, the 'Langdriver' in the alias may have to be set properly, using Borland's tables (use BDE administrator). Langdriver is commonly set to “‘ascii’ ANSI”.

As an example, use the table ExxxAyyy.TBL. If your PC is using a Windows code page yyy and your AS/400 is using an EBCDIC code page xxx.

In such a case you will need to set the 'LangDriver' accordingly.

Known Limitations

Using the Database Desktop to access AS/400 table will result in an error.

This is due to an undocumented IDAPI function called by the Database Desktop and we don't expect this to be resolved in a future release.

Note that it does not make any sense to modify the structures of the AS/400 table using the Database Desktop instead of the AS/400 native tools as most of the AS/400 tables features are not directly mapped to PC equivalent.

TBatchMove has been successfully tested in a lot of situations (see MoveFile sample).

However, if the file already exists in the AS/400 you will get an execution error.

This is not the standard behavior of the TbatchMove component as it should override the existing table, however it is a pretty safe protection against wrong manipulations.

FAQ

Installation FAQ

With TCP/IP, installation claims `Connection refused'

TCP/IP installation is using the FTP protocol. This error means that no FTP server is running on the AS/400. You can start the FTP server using the command 'STRTCPSVR SERVER(*FTP)'.

My alias and drivers were not created by setup

Before running the installation, all Delphi or other IDAPI applications must be shut down. Try to close all your applications and re-execute the setup.

Limits FAQ

Is Easycom compatible with MIDAS ?

Yes.

Is Easycom multithread compliant ?

Yes, but it is guaranteed with native TCP/IP connection only (because many routers are not thread-safe). Please consult manual to know how to configure it.

What is the limit of the number of fields in an index ?

Unfortunately, Borland IDAPI has a limit of 16 fields for an index. Some AS/400 files can have more. With Easycom For Delphi this index will appear like a regular index but with only the 16 first fields.

Common Problems FAQ

My Index list is not complete, or was not updated from recent changes !

The index list will only contain 'true' indexes, so all logicals will be shown except special logicals like :

- multiformat logicals.
- logicals with select/omit clauses.

To use these logicals, use the 'TableName' property of your TTable component.

Easycom maintain a cache of the index list, because the BDE often asks for this index list. This improve the time for opening a file during runtime. At runtime or with deployment, this cache is never cleared or updated.

So, to have the correct list, ask for your Index list within Delphi or C++ Builder. The index list will be updated in these situations :

- with a TTable component WHEN ACTIVE is FALSE.
- with Database Explorer UNDER DELPHI (opened with the Database/Explore menu).

If you are still having problems, you can clear EASYCOM index list cache. It can be done executing 'DELETE FROM EASYCOM/YIDXR0100' from the database explorer (where EASYCOM is the AS/400 library that contains the Easycom server).

My application is too slow, how to tune it up ?

Here are some rules that can help to speed up an Easycom application :

1. TTable <-> TQuery

TTable is faster because it uses Record-level accesses. It uses the key fields in native mode.

So TQuery should only be used when TTable usage is impossible. You even can create additional logicals (with select/omit for example) to avoid Tquery.

TQuery and TTable are allways faster when used in Read/Only mode. So if you don't need to update your data, always select the Readonly mode to your components (ReadOnly:=true for TTable, and Requestlive:=false for TQuery).

2. Setrange <-> Filters <-> OnFilterRecord

This is important to know how these different filtering features are working :

- SetRange is a record-level action (it relay on Key fields defined in IndexFieldNames).
- Filter generates an OPNQRYF each time the filter expression is changed.
- OnFilterRecord is a local filtering. It means that all records will be read.

So, always prefer a SetRange (or SetRangeStart ...) to filters.

The filter is powerful when you only have to filter with always the same expression. But if you want to change often the filter expression it can be a little bit slow. It also can be slow if you often deactivate/activate the filter.

<u>How to solve this situation ?</u>

You can combine a SetRange with an OnFilterRecord. The setrange clause will eliminate most records, and OnFilterRecord will only eliminate some others.

Of course in some cases, filter property will be very useful. Just be careful to not have a filter on each TTable component !

3. DBLookupComboBox

To tune your DBLookupComboBox, be careful to have a keyField for the ListSource that is the keyfield in your TTable. If your list source is a 'Detail TTable', best is to have a key that involves the masterfields and the key of your list source.

Example :

You have 3 files :

- COMPANY (COMP_ID, COMP_NAME, ...)
- PEOPLE (COMP_ID, PEOPL_ID, PEOPL_NAME, ...)
- COUNTRY (COUNT_ID, COUNT_NAME, ...)

Imagine you want to have a DBLookUpCombo for choosing somebody (POEPLE file) of the COMPANY, and another to choose a COUNTRY. Of course COMPANY is a MasterSource of PEOPLE on the COMP_ID field.

You will have best results if you choose the correct IndexFieldNames for PEOPLE and COUNTRY files :

PEOPLE : COMP_ID;PEOPL_ID

COUNTRY : COUNT_ID

However, just select PEOPL_ID in the DblookupComboBox component.

So COMP_ID will be used for the Master/Detail relationship (as a partial key), and PEOPL_ID will be used for the DBLoopUpCombo.

NB : if you don't select the right IndexFieldNames (for example COMP_ID instead of COMP_ID;PEOPL_ID), Delphi will generate filters to mach the record, and it will be more slow (the result will be the same, but slower).

Decision cube omits sometimes some columns. How to fix that ?

Decision cube components generates an SQL query.

You can see it by clicking on 'SQL' property of your TDecisionCube component.

This query often like :

```
SELECT x, y, SUM(z)
GROUP BY x, y
```

And the Decision cube component will assume that x and y values will come in the right order.

But the AS/400 SQL engine never sorts the result if you don't ask it so.

So, to be sure to have correct results, just modidy the SQL query like this :

```
SELECT x, y, SUM(z)
GROUP BY x, y
ORDER BY x, y
```

AS/400 specifics FAQ

How to execute AS/400 commands ?

Use the Easycom For Delphi special component 'ACE400RCmd'.

How to call AS/400 programs ?

Use the TstoredProc component to call programs with input/output parameters. The procedure must be created before using the PC program 'EASYCOM Stored Procedures configuration'. Please see our manual for more details.

How to access Data queues ?

Dataqueue access is also very easy to use (for data queues with and without key). This is done by using regular Ttable components !

How to have the source of the AS/400 files made by EASYCOM ?

When using CreateTable, TBatchmove, or the Datapump tool, EASYCOM creates AS/400 files.

You might want to change the file descriptions.

With EASYCOM, files can be created with DDS or with SQL.

1. When creating table using DDS

DDS is used if the AS/400 file name fit 10 characters and CRTTABLE SCRIPT setting of CONVERSION RULES is not used.

The DDS source is created in QDDSSRC in the library you created the file.

You will have the original field name in COLHDG if it is longer than 10 characters.

You will also have the ALIAS clause to have long names with your DDS.

All fields will be null capable by detault.

So you can change the source, and simply recompile it !

2. When creating table using SQL

SQL is used when file names are longer that 10 characters (AS/400 long file names) or when CRTTABLE SCRIPT option in CONVERSION RULES option is used.

You can have the SQL statements generated in a PC file.

Change CONVERSION RULES like this:

CONVERSION RULES=CRTTABLE SCRIPT=

You will have all the SQL table creation statements generated in that file. You will just have to transfer it to your AS/400 and use RUNSQLSTM to execute it after modifications.

This file will also contain the index creation statements.

How to access join files ?

Join files are used like regular files. Simply fill TableName with the name of the join file.

How to make a case insensitive key search or use a different sort display ?

For each AS/400 logical/physical, you can define a sorting sequence (diffrent from the current sorting sequence).

Check the QSRTSEQ system value to see what is your system's default value. You also can check your job attributes (with option 2 of dspjob). The dedault system's value is *HEX.

To provide a case insensitive key search, you can create a logical file with ALTSEQ option, giving a file like QSYS/QSYSTRNTBL. Use 'go CMDTBL' to display, create or modify the tables.

How to make Delayed updates working ?

Delayed updates cannot work with AS/400. The AS/400 does not permit multiple locks on a file, and Delphi tries to do this. So, it is not advisable to use Delayed updates.

How to access file members ?

For a file (using Ttable), simply use syntax LIBRARY/FILE(MEMBER) in the TableName property.

For queries, use OVRDBF AS/400 command to select the member.

How to use 36 files ?

If you don't have any description created on the AS/400, create and empty Physical file in AS/400 mode (DDS) that has the description of the F36 record. Then fill-up the 'TableName' property as follows :

How to call an interactive program from Easycom For Delphi ?

It is not possible to do this directly. But it is possible to call a non-interactive program and communicate with it using a Dataqueue.

How to call a Delphi program from an AS/400 menu ?

This can be done with our product Launcher/400.

See our web site http://www.easycom-aura.com

Is it possible to retrieve jobs properties such as user profile, user library list, and so

This is easily done with the component 'ACE400Rrtv'. This components allows to easily use 'RTV' commands like RTVJOBA or RTVSYSVAL. Install the components and try the sample 'Delphi\RCmd'.

Is it possible to work with multiple AS/400 at a time and how ?

Simply create multiple Tdatabase components or multiple aliases and use them.

Does the product support journalized files and how?

With Easycom For Delphi it is used exactly the same way it is with any SQL database (using Tdatabase components methods).

Does Easycom For Delphi Support files with Select/Omit ?

This is fully supported. But because of not being true indexes they do not appear in the 'IndexName' property of a Ttable Component. As they are not true indexes, this is safe to use it like a physical file (referencing it in the 'TableName' property).

How to contact Technical Support ?

For support requests you must create a ticket via the support website: http://support.easycom-aura.com.

You must register the first time to receive a password.

Assistance contract

AURA Equipements offers you several levels of technical support.

Contact us, we will send you the best commercial offer to meet your needs.

For general or commercial information : info@easycom-aura.com.

User License

Ownership of the Software

1. The Easycom For Delphi software programs (« Software ») and accompanying written materials (the « Documentation ») are owned by AURA Equipements (« Licensors ») and are protected by International copyright laws and by international treaties. The Easycom engine is a product that is the sole propriety of AURA Equipements.

Definitions

2. End User : The End user is either the natural person or the legal subject that bought the license or that uses an Evaluation Version.

3. Evaluation Version : Your use of the software is for the purpose of evaluating whether to purchase an ongoing license to the Software. You use an Evaluation Version when you install the Software without the appropriate serialization code (which takes the form of a serial number) to be install system. When you use an Evaluation Version, as soon as the DLL files in the Software are called upon, will be prompted before a dialog box appears explaining that you are running an Evaluation Version of the Software. You will then receive about twenty minutes of time to use the product by a similar dialog box every minute after one hour of use. The total number of time you can use the DLL is limited to 100. Anyhow the case the evaluation period for use is limited to 15 days.

4. Software : Software is the right to use the software-product "EASYCOM Server" and the Easycom For Delphi a direct level access for a PC Side for one development tool.

(a) Installation : The "Easycom For Delphi" software, includes two parts : the client part and the server part. The client part is to be installed on the PC which one is linked to the AS/400. The server part is to be installed on the AS/400.

(b) You can install the server part only once at a time one a specific AS/400 machine. (The software activation key is calculated by using the AS/400 serial number). In case, the software is to access to several AS/400, you have to purchase a license for each AS/400.

(c) The client part can be installed on as many PC's you want. But only as many PC's as the number of users of the license will be able to access the AS/400 at a time. (The activation key is calculated by using the number of users too)

(d) The evaluation version includes only one server part, one client part with one single connection.

4.1 The "Easycom For Delphi developer" gives to the developer all the functions requested for the development of an application.

4.2 The "Easycom For Delphi runtime" gives to the user the possibility to execute the application, which has been developed with "Easycom For Delphi developer". You can not develop any application with "Easycom For Delphi runtime".

4.3 The evaluation version is "Easycom For Delphi developer" limited for a period of time.

5. Maintenance release : shall mean corrections of errors and minor additions or improvements of functions in the software that are released to the public by Licensor from time to time at its sole discretion. A new maintenance release is identified by a modification of the product numbering to the right of a decimal point (e.g., a released labeled « 4.1 » would be a maintenance release of version 4 or 4.0 of the software).

6. New Version : shall mean important additions of improvements of functions or modules in the software that are released to the public by Licensor from time to time at its sole discretion. A New Version is identified by a modification of the product numbering to the left of decimal point (e.g., a move from « 4.1 » to « 5.0 »). Licensor maintains full discretion as to whether and when to issue a new release of the Software and whether to classify a new release as a « Maintenance Release » or « New Version ».

Grant of License

7. License for Evaluation : You may use the Evaluation Version of the Software and Documentation in accordance with the terms of this Agreement (except that you will not benefit from the rights set forth in sections 9,18,22 and 23 below) and solely for the purpose of deciding whether to purchase a full License to the Software. You may also reproduce and distribute, including by means of posting on an Internet forum, the un-installed Evaluation Version of the Software known as the « install disks », provided you do not modify or delete this agreement, any copyright or trademark notices or any portion of the Software. You shall be solely responsible for any costs or liabilities arising from any such reproduction or distribution.

8. Acquisition of Full License : If you have an Evaluation Version of the Software and wish to benefit from the rights set forth in sections 9,18,22 and 23 below, you must obtain a serialization code, which will disable the evaluation dialog prompt cycle described in section 3 above. A Serialization code may be obtained, in exchange for the applicable Licensee fee, simply by contacting Licensors or any authorized AURA Equipements distributors, the coordinates of which are set forth below.

9. License : In consideration for the payment of the License fee and your agreement to abide by the terms of this agreement, Licensor grants non exclusive License to use the Software and the documentation depending on (i) number of Client Licenses bought (ii) the price paid Software Development and/or Software Run Time, connected to a single server.

Governing Law : This agreement shall be governed by the laws of the United States and all the signature of the international copyright treaties.

Reference : DP-USER$032005

AURA EQUIPEMENTS

129 Rue de l'Abbé Groult, 75015 PARIS, France

Tel: 33 (0)1 53 76 86 35

Web: www.easycom-aura.com