

AdsClass++

Working with the Advantage Database Server

AdsClass++ is a object oriented binding of the ADS DBMS to a Xbase++ application. The communication is based on the ADS API and does not use the Alaska ADSDBE. All tables or SQL queries are kept in objects.

Advantages of AdsClass++:

- fast because based on the ADS API
- supports all ADS versions, also the local engine
- supports all ADS table formats:
DBF, CDX, VFP, ADT
- supports all data types
- supports Rawkey indexes
- supports all further ADS functionalities
- simplest integration of SQL
- thread save or thread spanning useable
- access solely via objects without workarea
- Developer Version with sourcecode
- English and German documentation and samples

The base idea behind this library is the encapsulation of a data table into an object. Xbase++ can define dynamic classes and can create the belonging objects at runtime.

The table structure is build in as ACCESS ASSIGN methods. For the navigation, search, filter, scope, etc. methods are available. Every Xbase++ file command is replaced by a conform method. Additional the Xbase++ commands are enhanced with productive methods.

A further advantage of the objects is the typecast. They can be deployed like all other variables, passed as parameters and evaluated, etc. Solely the table object communicates with the ADS, whereby no workarea is used. Hence this class is more useful for new developments less for converting old Clipper code.

Our experience with table objects shows that the code is easier readable, better to maintain and less prone to error regarding to table access. Especially with MDI applications the connection of a table to a dialog is relieved.

This concept is realized in **XClass++** for all Xbase++ DBEs and arrays. **AdsClass++** integrates itself seamless into **XClass++**, but can also be deployed independent from XClass++. Hence the use of table objects encapsulates an application from the data access technique.

Here the most important components as overview:

- **dsAceSession**, create and maintain a connection to the ADS server, for further connections to the same or another ADS server a new object must be created.

```
adsConnection := dsAceSession():New()
adsConnection:connect ("\\myServer")
? adsConnection:IsConnected()
AdsSetDefault ("\\myServer\myShare\myDir")

dbTable :=
OpenAceTable ("customer",adsConnection )
dbTable:close()
adsConnection:disconnect()
```

- **dsAceDD**, build connection to an ADS Datadictionary.

```
adsDD:=dsAceDD():New()
adsDD:connect ("\\myServer\myShare\myDir")
? adsDD:IsConnected()
dbTable := adsDD:OpenTable ("Kunde" )
dbTable:gotop()
```

With that class an ADD is maintained too:

- create groups and users
- administration of transactions
- maintenance of tables, indexes, views, SQ functions and stored procdures
- opening of application-global useable tables

- **Transaction**

per connection (dsAceSession or dsAceDD) a transaction can be started.

```
adsConnection:BeginnTransaction()
beginn sequence
...
adsConnection:CommitTransaction()
recover
adsConnection:RollBackTransaction()
end sequence
```

➤ **dsAceTable**, derivations of that are created table specific and represent a table. A table can be here a physical table, a SQL query, a view or the result of a stored procedure. There is no distinction between creation or use.

o table opening

```
dbTable := OpenAceTable("Customer",
    adsConnection/adsDD )
dbTable:gotop()
? dbTable:CUSTID // 4711
if dbTable:rlock()
    dbTable:CUSTID := 4712
    dbTable:unlock()
endif
? dbTable:alias // CUSTOMER_1
? dbTable:tablename() // CUSTOMER
do while !dbTable:eof()
    ...
    dbTable:skip()
enddo
dbTable:cose()
```

o indexing

```
// NTX
dbTable:ordcreate("NAME",,;
    "upper(lastname)+Upper(firstname)")
// CDX
dbTable:ordcreate("NAME",,;
    "upper(lastname)+Upper(firstname)")
// ADT, fieldtype CiCharacter, Rawkey Index
dbTable:ordcreate("NAME",,;
    "lastname;firstname")
// descend sort
dbTable:ordcreate("NAME",,;
    "lastname;firstname",,ADS_DESCENDING )
// or special collation
dbTable:ordcreate("NAME",,;
    "lastname;fisrtname",,;
    "GERMAN_VFP_CI_AS_1252" )
```

Rawkey indexes are binary and can mix all data types and are therefore extremely performant!

o and the respective search

```
// NTX, CDX
dbTable:ordsetfocus("NAME")
dbTable:seek(upper(padr("Müller",20))+
    upper(padr("Alfred",20)))
// ADT, Rawkey Index
dbTable:seek({"Müller","Alfred"})
? dbTable:found()
```

o filter and scope:

```
// is evaluated at client side
dbTable:setfilter({|db| myFilter(db)})
// is evaluated at server side
dbTable:setfilter("customer='Smith'")
// is optimized by index
dbTable:setaof("customer='Smith'")
// top and bottom scope
dbTable:Setscope("BERLIN")
```

SQL (customer can be a table or view)

```
dbTable := OpenSqlTable("select * from"+
    " customer where name like 'Smith%'";
    adsConnection )
```

➤ SQL with parameters

```
sqlConn :=;
    adsSqlTable():New(adsConnection)
sqlConn:prepare(
    "SELECT * from customer where city=:P1")
sqlConn:setParam("P1", "Berlin")
dbTable := sqlConn:Execute2Server()

sqlConn:setParam("P1", "Hamburg")
dbTable := sqlConn:Execute2Server()
sqlConn:close()
```

The table object of a SQL result can be indexed too!

➤ SQL stored procedures

If a stored procedure returns a resultset, it is used like every other table

```
sqlConn :=;
    adsSqlServer():New(adsConnection)
dbTable := sqlConn:SQL2Table(
    "execute procedure sp_MyProc()")
```

➤ dsAceMg, ADS management for display and maintenance of ADS settings and parameters

- o display of logged users
- o display of opened tables
- o display of locked records with the belonging users
- o disconnecting of users

➤ dsAdsListening, class for handling notifications received from ADS Server. Notifications which are announced previously, can be received and handled in a thread.

```
// initialize in MAIN
AdsListening(dsAceDD)
// create dialog with browse
dlgWindow := xbpDialog():New(...)
...
//announce events, browse will be refreshed
// automatically, if a record was saved.
AdsListening():CreateEvent(dlgWindow,,;
    "ARTICLE.UPDATE ",,;
    {| | dlgWindow:obrowse:RefreshAll()})
```

More informations for ADS under:

www.sybase.com/products/databasemanagement

An trialversion with full functionality, documentation and samples can be downlaoded from

www.ds-datasoft.de