
PHP Program Guide

Version

1.0

2.0 2003-04-24

/ (KCOM Proprietary).

한국컴퓨터통신(주)

135-080

679-4

13

02-3430-1200 <http://www.unisql.com>

1.	2
1.1	PHP	2
1.1.1	PHP UNISQL	2
1.1.2	2
1.2	PHP	3
1.3	4
1.4	4
1.4.1	Easy-Manager UniCAS	4
1.4.2	Terminal UniCAS	8
2. PHP	10
2.1	DB Connection	10
2.2	11
2.3	12
2.4	13
2.5	13
3. OID	16
3.1	OID	16
3.2	OID	16
3.3	OID	16
3.4	OID	17
4. (Collection Data Type)	18
4.1	Collection Data Type set	18
4.1.1	18
4.1.2	가	19
4.1.3	19
4.2	collection type sequence	20
4.2.1	20
4.2.2	가	20
4.2.3	21
4.2.4	21
5. GLO	23
5.1	glo instance	23
5.2	glo instance	23
5.3	glo instance	24
5.4	glo data	25

1.

1.1 PHP

PHP 가 , CGI . form
data 가 , Cookie .

1.1.1 PHP UNISQL

PHP 가 가 Database . PHP
UniSQL Database
UNISQL

1.1.2

UniSQL PHP dev.unisql.com Download
so (Unix) unisql_err.msg
PHP PHP3.0.16 , PHP4.0.0, PHP4.0.1.pl2, PHP4.0.2 ,
PHP4.0.3, PHP4.0.4

PHP(server-side HTML-embedded scripting language)

PHP 가 가 Database .
UniSQL Database , PHP
UniSQL Database
, APACHE PHP UniSQL

APACHE PHP

[http:// www.apache.org](http://www.apache.org)

[http:// www.php.net](http://www.php.net)

1.2 PHP

APACHE PHP 가 PHP.init UniSQL

php가 UniSQL
 php.ini(PHP3 php3.ini extension_dir unisql.so가

extension_dir= /usr/local/lib/php

, "Dynamic Extensions" extension=unisql.so 가

< >

Dynamic Extensions PHP so

extension=unisql4.0.1pl2.so

(php 4.0.1pl2.so 가 .)

UniSQL

php.ini(PHP3 php3.ini extension_dir unisql.so가
 ,extension_dir=/usr/local/lib/php 가
 dl("unisql.so"); dl

Error Message

php.ini(PHP3 php3.ini)
 [UniSQL]
 unisql.err_path = { unisql_err.msg }

:

[UniSQL]
 unisql.err_path=/usr/local/lib/php/unisql_err.msg

1.

linux php3 php main module(CGI: "php", apache: "httpd")
 build link option(eg. LDFLAGS) "-rdynamic" link

apache configure가 , src/Makefile EXTRA_LDFLAGS "-
 rdynamic" 가 , CGI configure가 , Makefile
 LDFLAGS "-rdynamic" 가

1.3

PHP (phpinfo()) unisql ()
 가 . 가
 가 가 .

```
<html><body>
```

```
<?
```

```
phpinfo();
```

```
?>
```

```
</body></html>
```

unisql

UniSQL	Value
Version	1.2.0
Error message file path	/user/local/lib/php/unisql_err.msg

1.4

UniSQL PHP UniCAS가 . UniCAS UniSQL Pooling
 . UniSQL PHP UniCAS .
 UniCAS Easy-Manager Terminal
 • PHP
 가 .

1.4.1 Easy-Manager UniCAS

UniCAS 1 CAS Type 가 ON
 , 2 CAS Type Application Server가
 . Easy-Manager CAS Type 가

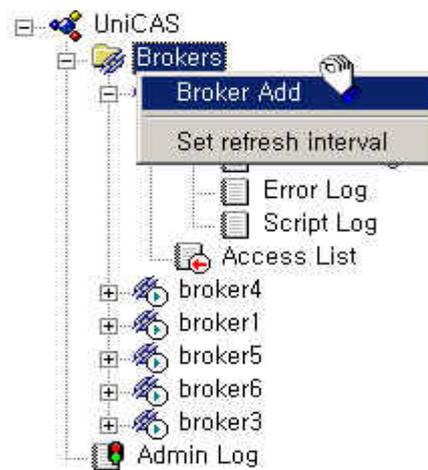
Name	Type	State	PID	Port	AS	Job Queue	Thread	CPU	Time	Reques
broker0	CAS	ON	24443	35100	5	0	7	0.06	0	0
broker1	VAS	ON	24429	35200	1	0	9	0.00	0	0
broker2	WAS	ON	24072	35300	1	0	11	0.00	0	0
broker3	ULS	ON	24075	53300	1	0	9	0.00	0	0

1. Easy-Manager

AS ID	PID	C	PSIZE	STATUS	CPU	CTIME	Last Access Time
1	24444	0	4616	IDLE	0.00	0:00	2002/07/09 16:03:56
2	24445	0	4616	IDLE	0.00	0:00	2002/07/09 16:03:56
3	24446	0	4616	IDLE	0.00	0:00	2002/07/09 16:03:56
4	24447	0	4616	IDLE	0.00	0:00	2002/07/09 16:03:56
5	24448	0	4616	IDLE	0.00	0:00	2002/07/09 16:03:56

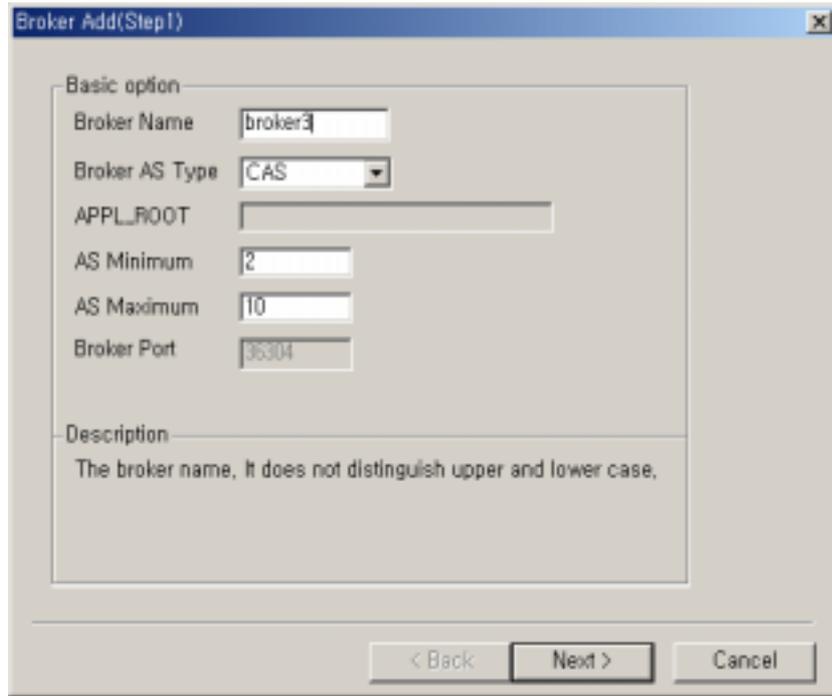
2. Easy-Manager Application Server

가 Broker Level “ Broker Add”
1 ~ 3 .



3.

.1



4. Broker Add – Step1

Broker Name :

Broker AS Type :

CAS

APPL_ROOT :

. CAS

AS Minimum :

AS Maximum :

Broker Port :

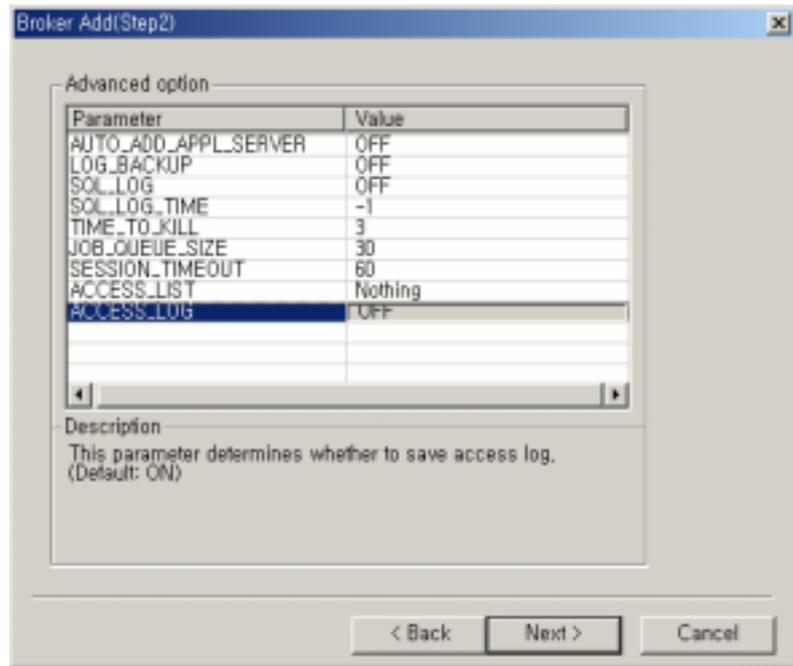
Connection

가

가 PHP

Description :

. 2

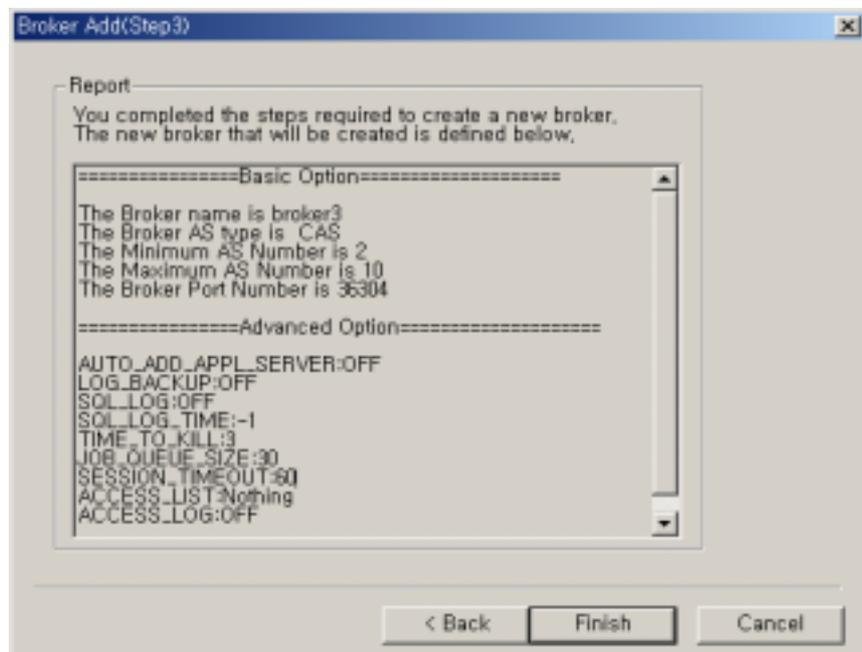


5. Broker Add – Step2

Advanced option : 1.

Description :

. 3



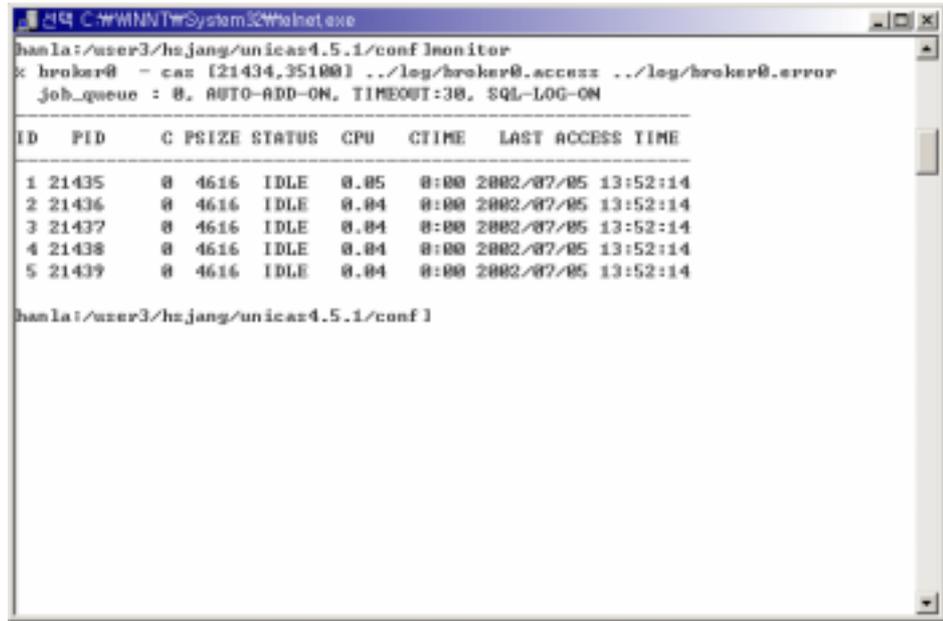
6. Broker Add – Step3

Report :

1.4.2 Terminal

UniCAS

UniCAS monitor (7.) PHP
 CAS type broker가
 , CAS broker Application Server
 Terminal CAS type 가



7. broker and current job information

CAS \$UNICAS/conf/unicas.conf broker
 .(8.) broker SERVICE ON ,
 Application Server 1 MIN_NUM_APPL_SERVER
 1

```
C:\WINNT\System32\cmd.exe
g> MASTER_SHM_ID      35000
s> ADMIN_LOG_FILE    log/admin.log

s> BROKER0
SERVICE             ON
BROKER_PORT          35100
APPL_SERVER          CAS
MIN_NUM_APPL_SERVER  5
MAX_NUM_APPL_SERVER  7
AUTO_ADD_APPL_SERVER ON
APPL_SERVER_SHM_ID   35101
APPL_SERVER_MAX_SIZE 10
LOG_DIR              log
LOG_BACKUP           OFF
SOURCE_EMU           unisql.env
SQL_LOG              ON
#SQL_LOG_TIME        60
TIME_TO_KILL         3
SESSION_TIMEOUT      30
JOB_QUEUE_SIZE       30
ACCESS_LOG           ON

#
"unicas.conf" 31 행, 699 문자
```

8. specify the parameters of unicas.conf file

2. PHP

UniSQL UniSQL 가 PHP
 . PHP
 5 “demodb” UniSQL “demodb”
 \$UNISQLX/demo/make_unisqlx_demo

2.1 DB Connection

```
<?
$con=unisql_connect("IP", CAS Port number, "demodb"); // DB

if($con) {
  $req=unisql_execute($con, "select city,country from location"); //
  if($req) {
    while($row=unisql_fetch($req)) { // row 가
      $city=$row[city]; // row city 가
      $country=$row[country]; // row country 가
      echo "$city";
      echo "$country";
    }
    unisql_close_request($req); // request
  } unisql_disconnect($con); //
}
?>
```

- DB connection unisql_connect() . unisql_connect() IP, CAS Port , , ID, Password , ID Password public
- * DB (ID) (password)
- DB connection() DB connection (unisql_fetch(), unisql_get()) 가
- unisql_execute() : request handle : unisql_execute(conn_handle , "query " ,);

```

        UNISQL_INCLUDE_OID,      UNISQL_ASYNC      가      ,
UNISQL_INCLUDE_OID      가      OID      가      ,
unysql_current_oid()      .(      )

```

```

UNISQL_ASYNC      .

```

```

• unysql_fetch()      result set      row      .      row
array      row      ,      row      .

```

```

(12)      .

```

```

• unysql_close_request( $request_handle)      unysql_execute()      request

```

```

• unysql_disconnect()      unysql_connect()

```

2.2

```

<?
$con=unysql_connect("IP", CAS Port number, "demodb"); // DB

if($con) {
    $req=unysql_execute($con, "select city,country from location",UNISQL_INCLUDE_OID |
UNISQL_ASYNC);
    if($req) {
        while ( $row=unysql_fetch ($req ) ) {

            echo "$row[0]";           //      1
            echo "$row[country]"; //      2
        }
        unysql_close_request($req); // request
    }

    $req2=unysql_execute($con, "select * from location");
    if($req2) {
        while($row=unysql_fetch($req2,UNISQL_OBJECT)) {
            echo "$row->city";           //      4
            echo "$row->country";
        }
        unysql_close_request($req2);
    } unysql_disconnect($con); //
}
?>

```

```

• unysql_fetch()      : unysql_fetch ($req_handle [,      ])

```

```

4가 가      .      result      array      object

```

UNISQL_NUM: Numerical array (0-based)


```

} unisql_disconnect($con); //
}
?>

```

```

        unisql_execute()                                OID
unisql_put()                                           .( 16 )

```

2.4

```

<?
$con=unisql_connect("IP", CAS Port number, "demodb"); // DB

if($con) {
    $req=unisql_execute($con, "delete from person where name='John'"); //
    if($req) {
        $row_count=unisql_affected_rows($req); // row
        echo "$row_count";
        unisql_close_request($req); // request
        unisql_commit($con);
    } else {
        unisql_rollback($con);
    } unisql_disconnect($con); //
}
?>

```

- unisql_affected_rows() : row . insert,update,delete 가 . Select unisql_num_rows() . (14)

```

: unisql_affected_rows( request_handle);

```

```

        unisql_execute()                                OID
        unisql_drop()                                  .( 16 )

```

2.5

```

<html>
<body>
<?
$con=unisql_connect("IP", CAS Port Number, "demodb"); // DB

if($con) {
    $req=unisql_execute($con, "select * from location",UNISQL_INCLUDE_OID);
    if($req) {
        unisql_move_cursor($req ,2, UNISQL_CURSOR_FIRST);
        unisql_fetch($req);
        $oid=unisql_current_oid($req);
        echo "oid $oid .<br>";

        $row_count=unisql_num_rows($req);
        $col_count=unisql_num_cols($req);
    }
}
?>

```

```

echo "row  $row_count      .<br>";
echo "col  $col_count     .";

$coltypes=unysql_column_types($req);
$colnames=unysql_column_names($req);

    while( list ($key, $colname) =each ($colnames))
        echo "$key      : $colname <br>";

    while( list ($key, $coltype) =each ($coltypes))
        echo "$key      :$coltype<br>";
        unysql_close_request($req); // request
    }
    unysql_disconnect($con); //
}
?>
<body>
</html>

```

- unysql_current_oid() : oid .
, unysql_execute() UNISQL_INCLUDE_OID
- unysql_current_oid(request_handle);
- unysql_move_cursor() : .
:unysql_move_cursor(request_handle, offset [, option]);
- offset : row ,
- option: offset 가
(UNISQL_CURSOR_FIRST, UNISQL_CURSOR_CURRENT, UNISQL_CURSOR_LAST)
- unysql_move_cursor(\$req, 2,UNISQL_CUROSR_FIRST);
, 2 row 가 .
- unysql_num_rows() : row . , select ,
synchronous 가 .
: unysql_num_rows(request_handle);
- unysql_num_cols() : column . ,select
가 .
: unysql_num_cols(request_handle);

- `unysql_column_names()` : column .
:unysql_column_names(request_handle);
- `unysql_column_types()` : column type .
:unysql_column_types(request_handle);

2. OID

3.1 OID

```
<?
$con=unysql_connect("IP", , "demodb"); // DB
if($con) {
    $req=unysql_execute($con, "select location from location where country='British West Indies'");

    if($req) {
        while($row=unysql_fetch($req)) {

            $oid=$row[location]; //$oid      oid
            echo "$oid\n";
        }
        unysql_close_request($req);
    } unysql_disconnect($con);
}
?>
```

"select location from location"

OID

* 3.2 3.4 3.1 \$oid . \$oid

3.2 OID

```
<?
$con=unysql_connect("IP", CAS Port Number,"demodb");

if($con) {

    $city = unysql_get($con, $oid, city); // $oid      city      가
    echo "$city"; //< 1>

    $array = unysql_get ($con, $oid); // $oid      array      가
    echo "$array[lodging]"; //< 2> array      lodging
    unysql_disconnect($con);
}
?>
```

- unysql_get() OID

:unysql_get(conn_handle, oid, " ");

unysql_get(conn_handle, oid);

3.3 OID

```
<?
$con=unysql_connect("IP", CAS Port Number,"demodb");
```

```

if($con) {
$new_country="America";

$city=unysql_put($con, $oid, "city", "New York"); // $oid          city          "New Y
ork"          update          .
$country=unysql_put($con, $oid, "country", $new_country);// $oid          country
$new_country          update          .

unysql_commit($con);
}
unysql_disconnect($con);
}
?>

```

- unysql_put() OID

```

:unysql_get(conn_handle, oid, "          "          ");

```

3.4 OID

```

<?
$con=unysql_connect("IP",Cas Port Number, "demodb"); // DB
if ($con) {
$delete_oid=unysql_drop($con,$oid); // $oid
  If($delete_oid) {
unysql_commit($con);
} else {
unysql_rollback($con);
}
unysql_disconnect($con);
}
?>

```

- unysql_drop() OID

```

:unysql_get(conn_handle,          OID);

```

3. (Collection Data Type)

(Collection Data Type)

Collection type set sequence가 .

- Set : element () , 가 .

- Sequence: 가 .

4.1 Collection Data Type set

```
create class student (  
  name char(15),  
  friend set of char(20)  
)  
  
insert into student(name, friend) values('Jery' , {'James','Michael'});
```

4.1.1

```
<?  
$con=unysql_connect("IP",cas port number, "demodb"); // DB  
  
if($con) {  
  
  $req=unysql_execute($con,"select student from student where name='Jery'");  
  
  if ($req) {  
  
    $row=unysql_fetch($req);  
    $oid=$row[student];  
    $name=unysql_get($con,$oid,name);  
    unysql_close_request($req);  
  }  
  $elem_count=unysql_col_size($con, $oid, "friend");  
  echo "$name friend $elem_count \n";  
  $elem_array=unysql_col_get($con,$oid,"friend");  
  while(list ($key, $val) =each ($elem_array)) {  
    echo "$val\n";  
  }  
  
  unysql_disconnect($con);  
}  
?>
```

- `mysqli_col_size()` : collection data type . collection type set
multiset, sequence 가

`mysqli_col_size(conn_handle, oid, "str_name");`

- `mysqli_col_get()` : collection data type . array (0 –
bases numerical) collection type set multiset, sequence
가 .

`mysqli_col_get(conn_handle, oid, "str_name");`

4.1.2 가

```
<?
$con=mysqli_connect("IP",cas port number, "demodb"); // DB

if($con) {

$req=mysqli_execute($con,"select student from student where name='Jery'");

if ($req) {

    $row=mysqli_fetch($req);
    $oid=$row[student];
    mysqli_close_request($req);
}
mysqli_set_add($con, $oid,"friend","Tomas");//element 가
mysqli_set_add($con, $oid,"friend","Susan");

$elem_array=mysqli_col_get($con,$oid,"friend");

while(list ($key, $val) =each ($elem_array)) {
echo "$val\n";
}
mysqli_commit($con);
mysqli_disconnect($con); //
}
?>
```

- `mysqli_set_add()` : collection data type set , element 가 .

`mysqli_set_add (conn_handle , oid , "attr_name", " 가 ")`;

4.1.3

```
<?
$con=mysqli_connect("IP",cas port number, "demodb"); // DB

if($con) {

$req=mysqli_execute($con,"select student from student where name='Jery'");

if ($req) {

    $row=mysqli_fetch($req);
    $oid=$row[student];
    mysqli_close_request($req);
}
}
```

```

    unisql_set_drop($con, $oid,"friend","Tomas"); //element
    unisql_set_drop($con, $oid,"friend","Susan");

    $elem_array=unisql_col_get($con,$oid,"friend");
    while(list ($key, $val) =each ($elem_array)) {
    echo "$val\n";
    }

    unisql_commit($con);
    unisql_disconnect($con);    //
}
?>

```

- unisql_set_drop() : collection data type set , element .
: unisql_set_drop(conn_handle, oid, "attr_name", " ");

4.2 collection type sequence

```

create class student (
name char(15),
friend sequence of char(20)
)

insert into student(name, friend) values('Jery' , {'James','Michael'});

```

4.2.1

set . (4.1.1)

4.2.2 가

```

<?
$con=unisql_connect("IP",cas port number, "demodb"); // DB
if($con) {
    $req=unisql_execute($con,"select student from student where name='Jery'");
    if ($req) {
        $row=unisql_fetch($req);
        $oid=$row[student];
        unisql_close_request($req);
    }
    unisql_seq_insert($con, $oid,"friend",1,"Tomas"); //element 가
    unisql_seq_insert($con, $oid,"friend",2,"Susan");

    $elem_array=unisql_col_get($con,$oid,"friend");
    while(list ($key, $val) =each ($elem_array)) {
    echo "$val\n";
    }
}

```

```

    unisql_commit($con);
    unisql_disconnect($con); //
}
?>

```

- unisql_seq_insert() : collection data type sequence , element
가 .
: unisql_seq_insert (conn_handle, oid, "attr_name", index, " 가 ");
sequence 가 index 가 ,
가 가 .

4.2.3

```

<?
$con=unisql_connect("IP",cas port number, "demodb"); // DB
if($con) {
    $req=unisql_execute($con,"select student from student where name='Jery'");
    if ($req) {
        $row=unisql_fetch($req);
        $oid=$row[student];
        unisql_close_request($req);
    }
    unisql_seq_put($con, $oid,"friend",1,"Jack");//element
    unisql_seq_put($con, $oid,"friend",2,"Mery");//element

    $elem_array=unisql_col_get($con,$oid,"friend");
    while(list ($key, $val) =each ($elem_array)) {
        echo "$val\n";
    }

    unisql_commit($con);
    unisql_disconnect($con); //
}
?>

```

- unisql_seq_put() : collection data type sequence , element
.
: unisql_seq_put(conn_handle, oid, "attr_name", index, "
Tomas Jack , Susan Merry
.

4.2.4

```

<?
$con=unisql_connect("210.216.33.136",43300, "demodb"); // DB
    $row=unisql_fetch($req);
    $oid=$row[student];
    unisql_close_request($req);
}

```

```
$elem_array=unysql_col_get($con,$oid,"friend");
```

```
while(list ($key, $val) =each ($elem_array)) {
```

```
if($key=="0") {
```

```
echo "$val .";
```

```
unysql_seq_drop($con, $oid, "friend",1); // 1 .
```

```
}
```

```
}
```

```
unysql_commit($con);
```

```
unysql_disconnect($con); //
```

```
}
```

```
?>
```

- unysql_seq_drop() : collection data type sequence , element

: unysql_seq_drop (conn_handle, oid, "attr_name", index, " ");

가 .

5. GLO

UniSQL/X

GLO(Generalized Large Object)

```
create class person (
  name string,
  photo glo
)
```

5.1 glo instance

```
<?
$con=unysql_connect("IP", Cas Port Number, "demodb");
if ($con) {
  $oid=unysql_new_glo($con,"glo","input.gif"); //glo

  if($oid) {
    $req=unysql_execute($con, "insert into person(name,photo) values('input.gif',$oid)");
    if($req) {

      echo "Image inserted successfully";

      unysql_close_request($req);
      unysql_commit($con);
    }
  }
  unysql_disconnect($con);
}
?>
```

- unysql_new_glo() : glo data , LO
OID
glo data glo (holder_obj,unit_size , header_size)

Unit_size: 8 , header_size: 0

: unysql_new_glo(conn_handle, class_name, "file_name");

glo OID , OID glo

5.2 glo instance

```
<?
$con=unysql_connect("IP",CAS Port Number, "demodb");
if ($con) {
```

```

$req=unysql_execute($con, "select photo from person where name='input.gif"); //glo
OID
if($req) {

list ($oid)=unysql_fetch($req);

unysql_close_request($req);

$res=unysql_save_to_glo($con,$oid, "input.gif"); // glo

if ($res) {

echo "image saved successfully";

unysql_commit($con);
}
}
unysql_disconnect($con);
}
?>

```

```

unysql_save_to_glo() : glo

:unysql_save_to_glo(conn_handle, oid, file_name);

glo OID

```

5.3 glo instance

```

<?
$con=unysql_connect("IP", CAS Port Number, "demodb");
if ($con) {

$req=unysql_execute($con, "select photo from person where name='input.gif");
if($req) {
list ($oid)=unysql_fetch($req);
unysql_close_request($req);
}
$res=unysql_load_from_glo ($con, $oid, "/tmp/output.gif");
if ($res) {
echo "Image loading Successfully";
}
unysql_disconnect($con);
}
?>

```

```

unysql_load_from_glo() : glo (LO )
(FBO )
.( )

: unysql_load_from_glo(conn_handle, oid, file_name);

```

glo OID

5.4 glo data

```
<?
$con=unysql_connect("IP", CAS Port Number, "demodb");
if ($con) {

$req=unysql_execute($con, "select photo from person where name='input.gif'");
if($req) {
list ($oid)=unysql_fetch($req);
unysql_close_request($req);
}
Header ("Content-type: image/jpeg");
unysql_send_glo($con, $oid); // glo                  Standard Output
if ($res) {
echo "Image loading Successfully";
}
unysql_disconnect($con);
}
?>
```

unysql_send_glo() : glo PHP standard ouput

: unysql_send_glo(conn_handle, oid);