

# Ingres II

# Introduction

These functions allow you to access Ingres II database servers.

Note
If you already used PHP extensions to access other database servers, note that Ingres doesn't allow concurrent queries and/or transaction over one connection, thus you won't find any result or transaction handle in this extension. The result of a query must be treated before sending another query, and a transaction must be committed or rolled back before opening another transaction (which is automatically done when sending the first query).

# Installing/Configuring

## Requirements

To compile PHP with Ingres support, you need the Ingres OpenAPI library and header files.

## Installation

This » [PECL](#) extension is not bundled with PHP. Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here:  
» <http://pecl.php.net/package/ingres>.

In order to have these functions available, you must compile PHP with Ingres support by using the `--with-ingres[=DIR]` option, where DIR is the Ingres base directory, which defaults to `/ll/ingres`. If the `II_SYSTEM` environment variable isn't correctly set you may have to use `--with-ingres=DIR` to specify your Ingres installation directory.

When using this extension with Apache, if Apache does not start and complains with "PHP Fatal error: Unable to start ingres\_ii module in Unknown on line 0" then make sure the environment variable `II_SYSTEM` is correctly set. Adding "export `II_SYSTEM="/home/ingres/II"` in the script that starts Apache, just before launching httpd, should be fine.

## Runtime Configuration

The behaviour of these functions is affected by settings in *php.ini*.

### Ingres configuration options

Name	Default	Changeable	Changelog
ingres.allow_persistent	"1"	PHP_INI_SYSTEM	Available since PHP 4.0.2. Removed in PHP 5.1.0.
ingres.max_persistent	"-1"	PHP_INI_SYSTEM	Available since PHP 4.0.2. Removed in PHP 5.1.0.
ingres.max_links	"-1"	PHP_INI_SYSTEM	Available since PHP 4.0.2. Removed in

			PHP 5.1.0.
ingres.default_database	NULL	PHP_INI_ALL	Available since PHP 4.0.2. Removed in PHP 5.1.0.
ingres.default_user	NULL	PHP_INI_ALL	Available since PHP 4.0.2. Removed in PHP 5.1.0.
ingres.default_password	NULL	PHP_INI_ALL	Available since PHP 4.0.2. Removed in PHP 5.1.0.
ingres.report_db_warnings	"1"	PHP_INI_ALL	Available since ingres 1.1.
ingres.cursor_mode	"0"	PHP_INI_ALL	Available since ingres 1.1.
ingres.blob_segment_length	"4096"	PHP_INI_ALL	Available since ingres 1.2.0.
ingres.trace_connect	"0"	PHP_INI_ALL	Available since ingres 1.2.1.
ingres.timeout	"-1"	PHP_INI_ALL	Available since ingres 1.4.0.
ingres.array_index_start	"1"	PHP_INI_ALL	Available since ingres 1.4.0.

For further details and definitions of the PHP\_INI\_\* constants, see the [php.ini directives](#).

## Resource Types

[ingres\\_connect\(\)](#) and [ingres\\_pconnect\(\)](#) return an Ingres II link identifier.

# Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

## **INGRES\_ASSOC** ( [integer](#) )

Columns are returned into the array having the fieldname as the array index.

## **INGRES\_NUM** ( [integer](#) )

Columns are returned into the array having a numerical index to the fields. This index starts with 1, the first field in the result.

## **INGRES\_BOTH** ( [integer](#) )

Columns are returned into the array having both a numerical index and the fieldname as the array index.

## **INGRES\_EXT\_VERSION** ( [string](#) )

Specifies the version of the Ingres Extension. Available since version 1.2.0 of the PECL extension.

## **INGRES\_API\_VERSION** ( [integer](#) )

Specifies the version of Ingres OpenAPI the extension was built against. Available since version 1.2.0 of the PECL extension.

## **INGRES\_CURSOR\_READONLY** ( [integer](#) )

Specifies that Ingres cursors should be opened in 'readonly' mode. Available since version 1.2.0 of the PECL extension.

## **INGRES\_CURSOR\_UPDATE** ( [integer](#) )

Specifies that Ingres cursors should be opened 'for update'. Available since version 1.2.0 of the PECL extension.

## **INGRES\_DATE\_MULTINATIONAL** ( [integer](#) )

Equivalent to the II\_DATE\_FORMAT setting of MULTINATIONAL. Available since version 1.2.0 of the PECL extension.

## **INGRES\_DATE\_MULTINATIONAL4** ( [integer](#) )

Equivalent to the II\_DATE\_FORMAT setting of MULTINATIONAL4. Available since version 1.2.0 of the PECL extension.

## **INGRES\_DATE\_FINNISH** ( [integer](#) )

Equivalent to the II\_DATE\_FORMAT setting of FINNISH. Available since version 1.2.0 of the PECL extension.

## **INGRES\_DATE\_ISO** ( [integer](#) )

Equivalent to the II\_DATE\_FORMAT setting of ISO. Available since version 1.2.0 of the PECL extension.

## **INGRES\_DATE\_ISO4** ( [integer](#) )

Equivalent to the II\_DATE\_FORMAT setting of ISO4. Available since version 1.2.0 of

the PECL extension.

**INGRES\_DATE\_GERMAN ( [integer](#) )**

Equivalent to the II\_DATE\_FORMAT setting of GERMAN. Available since version 1.2.0 of the PECL extension.

**INGRES\_DATE\_MDY ( [integer](#) )**

Equivalent to the II\_DATE\_FORMAT setting of MDY. Available since version 1.2.0 of the PECL extension.

**INGRES\_DATE\_DMY ( [integer](#) )**

Equivalent to the II\_DATE\_FORMAT setting of DMY. Available since version 1.2.0 of the PECL extension.

**INGRES\_DATE\_YMD ( [integer](#) )**

Equivalent to the II\_DATE\_FORMAT setting of YMD. Available since version 1.2.0 of the PECL extension.

**INGRES\_MONEY\_LEADING ( [integer](#) )**

Specifies the currency character that should be placed at the start of a money value. Equivalent to setting II\_MONEY\_FORMAT to 'L:'. Available since version 1.2.0 of the PECL extension.

**INGRES\_MONEY\_TRAILING ( [integer](#) )**

Specifies the currency character that should be placed at the end of a money value. Equivalent to setting II\_MONEY\_FORMAT to 'T:'. Available since version 1.2.0 of the PECL extension.

**INGRES\_STRUCTURE\_BTREE ( [integer](#) )**

Specifies the default table or index structure to BTREE when used in combination with the table\_structure or index\_structure option when connecting. Available since version 1.4.0 of the PECL extension.

**INGRES\_STRUCTURE\_CBTREE ( [integer](#) )**

Specifies the default table or index structure to COMPRESSED BTREE when used in combination with the table\_structure or index\_structure option when connecting. Available since version 1.4.0 of the PECL extension.

**INGRES\_STRUCTURE\_HASH ( [integer](#) )**

Specifies the default table or index structure to HASH when used in combination with the table\_structure or index\_structure option when connecting. Available since version 1.4.0 of the PECL extension.

**INGRES\_STRUCTURE\_CHASH ( [integer](#) )**

Specifies the default table or index structure to COMPRESSED HASH when used in combination with the table\_structure or index\_structure option when connecting. Available since version 1.4.0 of the PECL extension.

**INGRES\_STRUCTURE\_HEAP ( [integer](#) )**

Specifies the default table structure to HEAP when used in combination with the table\_structure option when connecting. Available since version 1.4.0 of the PECL extension.

**INGRES\_STRUCTURE\_CHEAP** ( [integer](#) )

Specifies the default table structure to COMPRESSED HEAP when used in combination with the `table_structure` option when connecting. Available since version 1.4.0 of the PECL extension.

**INGRES\_STRUCTURE\_ISAM** ( [integer](#) )

Specifies the default table or index structure to ISAM when used in combination with the `table_structure` or `index_structure` option when connecting. Available since version 1.4.0 of the PECL extension.

**INGRES\_STRUCTURE\_CISAM** ( [integer](#) )

Specifies the default table or index structure to COMPRESSED ISAM when used in combination with the `table_structure` or `index_structure` option when connecting. Available since version 1.4.0 of the PECL extension.

# Examples

This simple example shows how to connect, execute a query, print resulting rows and disconnect from an Ingres database.

## Example #1 - Simple Ingres Example

```
<?php
// Connecting, selecting database
$link = ingres_connect('database', 'user', 'password')
    or die('Could not connect: ' . ingres_error($link));
echo 'Connected successfully';

// Select from a table that exists in all Ingres databases
$query = 'SELECT * FROM iirelation';
$returncode = ingres_query($query,$link) or die('Query failed: ' .
    ingres_error($link));

// Print results in HTML
// relid - table name
// relowner - table owner
echo "<table>\n";
while ($iirelation = ingres_fetch_object(INGRES_BOTH, $link)) {
    echo "\t<tr>\n";
    echo "\t\t<td>" . $iirelation->relid . "</td>\n";
    echo "\t\t<td>" . $iirelation->relowner . "</td>\n";
    echo "\t</tr>\n";
}
echo "</table>\n";

// Commit transaction
ingres_commit($link);
// Closing connection
ingres_close($link);
?>
```



# Ingres II Functions

# ingres\_autocommit

ingres\_autocommit -- Switch autocommit on or off

## Description

bool **ingres\_autocommit** ( [ resource *\$link* ] )

[ingres\\_autocommit\(\)](#) is called before opening a transaction (before the first call to [ingres\\_query\(\)](#) or just after a call to [ingres\\_rollback\(\)](#) or [ingres\\_commit\(\)](#) ) to switch the "autocommit" mode of the server on or off (when the script begins the autocommit mode is off).

When the autocommit mode is on, every query is automatically committed by the server, as if [ingres\\_commit\(\)](#) was called after every call to [ingres\\_query\(\)](#).

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_rollback\(\)](#)
- [ingres\\_commit\(\)](#)

# ingres\_close

ingres\_close -- Close an Ingres II database connection

## Description

bool **ingres\_close** ( [ resource *\$link* ] )

[ingres\\_close\(\)](#) closes the connection to the Ingres server that's associated with the specified link.

[ingres\\_close\(\)](#) isn't usually necessary, as it won't close persistent connections and all non-persistent connections are automatically closed at the end of the script.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [ingres\\_connect\(\)](#)
- [ingres\\_pconnect\(\)](#)

# ingres\_commit

ingres\_commit -- Commit a transaction

## Description

bool **ingres\_commit** ( [ resource *\$link* ] )

[ingres\\_commit\(\)](#) commits the currently open transaction, making all changes made to the database permanent.

This closes the transaction. A new one can be open by sending a query with [ingres\\_query\(\)](#).

You can also have the server commit automatically after every query by calling [ingres\\_autocommit\(\)](#) before opening the transaction.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_rollback\(\)](#)
- [ingres\\_autocommit\(\)](#)

# ingres\_connect

ingres\_connect -- Open a connection to an Ingres database

## Description

```
resource ingres_connect ( [ string $database [, string $username [, string $password [,  
array $options ]]] ] )
```

[ingres\\_connect\(\)](#) opens a connection with the given Ingres *database*.

The connection is closed when the script ends or when [ingres\\_close\(\)](#) is called on this link.

All the other ingres functions use the last opened link as a default, so you need to store the returned value only if you use more than one link at a time.

## Parameters

If some parameters are missing, [ingres\\_connect\(\)](#) uses the values in *php.ini* for *ingres.default\_database*, *ingres.default\_user* and *ingres.default\_password*.

*database*

The database name. Must follows the syntax *[node\_id::]dbname[/svr\_class]*.

*username*

The Ingres user name.

*password*

The password associated with *username*.

*options*

[ingres\\_connect\(\)](#) options

Option name	Description
date_century_boundary	The threshold by which a 2 digit year is determined to be in the current century or in the next century. Equivalent to II_DATE_CENTURY_BOUNDARY.
group	Specifcifies the group ID of the user, equivalent to the '-G' flag
role	The role ID of the application. If a role password is required, the parameter value should be specified as "role/password"
effective_user	The ingres user account being

	impersonated, equivalent to the '-u' flag
dbms_password	The internal database password for the user connecting to Ingres
table_structure	<p>The default structure for new tables. Valid values for table_structure are:</p> <ul style="list-style-type: none"> <li>• INGRES_STRUCTURE_BTREE</li> <li>• INGRES_STRUCTURE_HASH</li> <li>• INGRES_STRUCTURE_HEAP</li> <li>• INGRES_STRUCTURE_ISAM</li> <li>• INGRES_STRUCTURE_CBTREE</li> <li>• INGRES_STRUCTURE_CISAM</li> <li>• INGRES_STRUCTURE_CHASH</li> <li>• INGRES_STRUCTURE_CHEAP</li> </ul>
index_structure	<p>The default structure for new secondary indexes. Valid values for index_structure are:</p> <ul style="list-style-type: none"> <li>• INGRES_STRUCTURE_CBTREE</li> <li>• INGRES_STRUCTURE_CISAM</li> <li>• INGRES_STRUCTURE_CHASH</li> <li>• INGRES_STRUCTURE_BTREE</li> <li>• INGRES_STRUCTURE_HASH</li> <li>• INGRES_STRUCTURE_ISAM</li> </ul>
login_local	Determines how the connection user ID and password are used when a VNODE is included in the target database string. If set to TRUE, the user ID and password are used to locally access the VNODE and the VNODE login information is used to establish the DBMS connection. If set to FALSE, the process user ID is used to access the VNODE and the connection user ID and password are used in place of the VNODE login information to establish the

	DBMS connection. This parameter is ignored if no VNODE is included in the target database string. The default is FALSE.
timezone	Controls the timezone of the session. If not set it will default the the value defined by II_TIMEZONE_NAME. If II_TIMEZONE_NAME is not defined the NA-PACIFIC (GMT-8 with Daylight Savings) is used.
date_format	<p>Sets the allowable input and output format for Ingres dates. Defaults to the value defined by II_DATE_FORMAT. If II_DATE_FORMAT is not set the default date format is US, e.g. mm/dd/yy. Valid values for date_format are:</p> <ul style="list-style-type: none"> <li>• INGRES_DATE_DMY</li> <li>• INGRES_DATE_FINISH</li> <li>• INGRES_DATE_GERMAN</li> <li>• INGRES_DATE_ISO</li> <li>• INGRES_DATE_ISO4</li> <li>• INGRES_DATE_MDY</li> <li>• INGRES_DATE_MULTINATIONAL</li> <li>• INGRES_DATE_MULTINATIONAL4</li> <li>• INGRES_DATE_YMD</li> <li>• INGRES_DATE_US</li> </ul>
decimal_separator	The character identifier for decimal data
money_lort	<p>Leading or trailing currency sign. Valid values for money_lort are:</p> <ul style="list-style-type: none"> <li>• INGRES_MONEY_LEADING</li> <li>• INGRES_MONEY_TRAILING</li> </ul>
money_sign	The currency symbol to be used with the MONEY datatype

money_precision	The precision of the MONEY datatype
float4_precision	Precision of the FLOAT4 datatype
float8_precision	Precision of the FLOAT8 data
blob_segment_length	The amount of data in bytes to fetch at a time when retrieving BLOB/CLOB data, defaults to 4096 bytes when not explicitly set.

## Return Values

Returns a Ingres link resource on success, or **FALSE** on failure.

## Examples

### Example #2 - [ingres\\_connect\(\)](#) example

```
<?php
$link = ingres_connect("mydb", "user", "pass")
    or die("Could not connect");
echo "Connected successfully";
ingres_close($link);
?>
```

### Example #3 - [ingres\\_connect\(\)](#) example using default link

```
<?php
ingres_connect("mydb", "user", "pass")
    or die("Could not connect");
echo "Connected successfully";
ingres_close();
?>
```

## See Also

- [ingres\\_pconnect\(\)](#)
- [ingres\\_close\(\)](#)



# ingres\_cursor

ingres\_cursor -- Gets a cursor name for a given link resource

## Description

string **ingres\_cursor** ( [ resource *\$link* ] )

Returns an string containing the active cursor name. If no cursor is active then NULL is returned.

If a *link* resource is passed to [ingres\\_cursor\(\)](#) it returns the cursor name recorded for the link. If no link is passed then [ingres\\_cursor\(\)](#) returns the cursor name associated with the default link.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns an string containing the active cursor name. If no cursor is active then NULL is returned.

## Examples

### Example #4 - [ingres\\_cursor\(\)](#) example

```
<?php
$link = ingres_connect($database, $user, $password);

ingres_prepare("select * from table", $link);

$cursor_name = ingres_cursor($link);

echo $cursor_name;

?>
```

## See Also

- `ingres_prepare()`
- `ingres_execute()`

# ingres\_errno

ingres\_errno -- Gets the last ingres error number generated

## Description

int **ingres\_errno** ( [ resource *\$link* ] )

Returns an integer containing the last error number. If no error was reported 0 is returned.

If a *link* resource is passed to [ingres\\_errno\(\)](#) it returns the last error recorded for the link. If no link is passed then [ingres\\_errno\(\)](#) returns the last error reported using the default link.

The function, [ingres\\_errno\(\)](#), should always be called after executing a database query. Calling another function before [ingres\\_errno\(\)](#) is called, will reset or change any error code from the last Ingres function call.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns an integer containing the last error number. If no error was reported 0 is returned.

## Examples

### Example #5 - [ingres\\_errno\(\)](#) example

```
<?php
$link = ingres_connect($database, $user, $password);

ingres_query("select * from table", $link);

$error_code = ingres_errno($link);

if ( $error_code != 0 ) {
    echo "An error occurred - " . $error_code;
}
?>
```

## See Also

- [ingres\\_error\(\)](#)
- [ingres\\_errsqlstate\(\)](#)

# ingres\_error

ingres\_error -- Gets a meaningful error message for the last error generated

## Description

string **ingres\_error** ( [ resource *\$link* ] )

Returns a string containing the last error, or NULL if no error has occurred.

If a *link* resource is passed to [ingres\\_error\(\)](#) it returns the last error recorded for the link. If no link is passed then [ingres\\_error\(\)](#) returns the last error reported using the default link.

The function, [ingres\\_error\(\)](#), should always be called after executing any database query. Calling another function before [ingres\\_error\(\)](#) is called will reset or change any error message from the last Ingres function call.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns a string containing the last error, or NULL if no error has occurred.

## Examples

### Example #6 - [ingres\\_error\(\)](#) example

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");

$error_text = ingres_error();

if (!is_null($error_text)) {
    echo "An error occurred - " . $error_text;
}
?>
```

## See Also

- [ingres\\_errno\(\)](#)
- [ingres\\_errsqlstate\(\)](#)

# ingres\_errsqlstate

ingres\_errsqlstate -- Gets the last SQLSTATE error code generated

## Description

string **ingres\_errsqlstate** ( [ resource *\$link* ] )

Returns a string containing the last SQLSTATE, or NULL if no error has occurred.

If a *link* resource is passed to [ingres\\_errsqlstate\(\)](#) it returns the last error recorded for the link. If no link is passed then [ingres\\_errsqlstate\(\)](#) returns the last error reported using the default link.

The function, [ingres\\_errsqlstate\(\)](#), should always be called after executing any database query. Calling another function before [ingres\\_errsqlstate\(\)](#) is called will reset or change any error message from the last Ingres function call.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns a string containing the last SQLSTATE, or NULL if no error has occurred.

## Examples

### Example #7 - [ingres\\_errsqlstate\(\)](#) example

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");

$error_sqlstate = ingres_errsqlstate();

if (!is_null($error_sqlstate)) {
    echo "An error occurred - " . $error_sqlstate;
}
?>
```

## See Also

- [ingres\\_errno\(\)](#)
- [ingres\\_error\(\)](#)



# ingres\_fetch\_array

ingres\_fetch\_array -- Fetch a row of result into an array

## Description

array **ingres\_fetch\_array** ( [ int *\$result\_type* [, resource *\$link* ] ] )

This function is an extended version of [ingres\\_fetch\\_row\(\)](#). In addition to storing the data in the numeric indices of the result array, it also stores the data in associative indices, using the field names as keys.

If two or more columns of the result have the same field names, the last column will take precedence. To access the other column(s) of the same name, you must use the numeric index of the column or make an alias for the column.

```
<?php

ingres_query("select t1.f1 as foo t2.f1 as bar from t1, t2");
$result = ingres_fetch_array();
$foo = $result["foo"];
$bar = $result["bar"];

?>
```

Speed-wise, the function is identical to [ingres\\_fetch\\_object\(\)](#), and almost as quick as [ingres\\_fetch\\_row\(\)](#) (the difference is insignificant).

## Parameters

*result\_type*

*result\_type* can be INGRES\_NUM for enumerated array, INGRES\_ASSOC for associative array, or INGRES\_BOTH (default).

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns an array that corresponds to the fetched row, or **FALSE** if there are no more rows.

## Examples

### Example #8 - [ingres\\_fetch\\_array\(\)](#) example

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");
while ($row = ingres_fetch_array()) {
    echo $row["user_id"]; // using associative array
    echo $row["fullname"];
    echo $row[1];         // using enumerated array
    echo $row[2];
}
?>
```

### See Also

- [ingres\\_query\(\)](#)
- [ingres\\_num\\_fields\(\)](#)
- [ingres\\_field\\_name\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_fetch\_object

ingres\_fetch\_object -- Fetch a row of result into an object

## Description

object **ingres\_fetch\_object** ( [ int *\$result\_type* [, resource *\$link* ] ] )

This function is similar to [ingres\\_fetch\\_array\(\)](#), with one difference - an object is returned, instead of an array. Indirectly, that means that you can only access the data by the field names, and not by their offsets (numbers are illegal property names).

Speed-wise, the function is identical to [ingres\\_fetch\\_array\(\)](#), and almost as quick as [ingres\\_fetch\\_row\(\)](#) (the difference is insignificant).

## Parameters

*result\_type*

The optional argument *result\_type* is a constant and can take the following values: INGRES\_ASSOC, INGRES\_NUM, and INGRES\_BOTH.

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns an object that corresponds to the fetched row, or **FALSE** if there are no more rows.

## Examples

### Example #9 - [ingres\\_fetch\\_object\(\)](#) example

```
<?php
ingres_connect($database, $user, $password);
ingres_query("select * from table");
while ($row = ingres_fetch_object()) {
    echo $row->user_id;
    echo $row->fullname;
}
?>
```

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_num\\_fields\(\)](#)
- [ingres\\_field\\_name\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_fetch\_row

ingres\_fetch\_row -- Fetch a row of result into an enumerated array

## Description

array **ingres\_fetch\_row** ( [ resource *\$link* ] )

[ingres\\_fetch\\_row\(\)](#) returns an array that corresponds to the fetched row, or **FALSE** if there are no more rows. Each result column is stored in an array offset, starting at offset 1.

Subsequent call to [ingres\\_fetch\\_row\(\)](#) would return the next row in the result set, or **FALSE** if there are no more rows.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns an array that corresponds to the fetched row, or **FALSE** if there are no more rows.

## Examples

### Example #10 - [ingres\\_fetch\\_row\(\)](#) example

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");
while ($row = ingres_fetch_row()) {
    echo $row[1];
    echo $row[2];
}
?>
```

## See Also

- [ingres\\_num\\_fields\(\)](#)
- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)

- [ingres\\_fetch\\_object\(\)](#)

# ingres\_field\_length

ingres\_field\_length -- Get the length of a field

## Description

int **ingres\_field\_length** ( int *\$index* [, resource *\$link* ] )

[ingres\\_field\\_length\(\)](#) returns the length of a field. This is the number of bytes used by the server to store the field. For detailed information, see the Ingres/OpenAPI User Guide - Appendix C.

## Parameters

*index*

*index* is the number of the field and must be between 1 and the value given by [ingres\\_num\\_fields\(\)](#).

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns the length of a field.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_field\_name

ingres\_field\_name -- Get the name of a field in a query result

## Description

string **ingres\_field\_name** ( int *\$index* [, resource *\$link* ] )

[ingres\\_field\\_name\(\)](#) returns the name of a field in a query result.

## Parameters

*index*

*index* is the number of the field and must be between 1 and the value given by [ingres\\_num\\_fields\(\)](#).

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns the name of a field in a query result, or **FALSE** on failure.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)



# ingres\_field\_nullable

ingres\_field\_nullable -- Test if a field is nullable

## Description

bool **ingres\_field\_nullable** ( int *\$index* [, resource *\$link* ] )

Test if a field is nullable.

## Parameters

*index*

*index* is the number of the field and must be between 1 and the value given by [ingres\\_num\\_fields\(\)](#).

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

[ingres\\_field\\_nullable\(\)](#) returns **TRUE** if the field can be set to the **NULL** value and **FALSE** if it can't.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_field\_precision

ingres\_field\_precision -- Get the precision of a field

## Description

int **ingres\_field\_precision** ( int *\$index* [, resource *\$link* ] )

[ingres\\_field\\_precision\(\)](#) returns the precision of a field. This value is used only for decimal, float and money SQL data types. For detailed information, see the Ingres/OpenAPI User Guide - Appendix C.

## Parameters

*index*

*index* is the number of the field and must be between 1 and the value given by [ingres\\_num\\_fields\(\)](#).

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns the field precision as an integer.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_field\_scale

ingres\_field\_scale -- Get the scale of a field

## Description

int **ingres\_field\_scale** ( int *\$index* [, resource *\$link* ] )

[ingres\\_field\\_scale\(\)](#) returns the scale of a field. This value is used only for the decimal SQL data type. For detailed information, see the Ingres/OpenAPI User Guide - Appendix C.

## Parameters

*index*

*index* is the number of the field and must be between 1 and the value given by [ingres\\_num\\_fields\(\)](#).

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns the scale of the field, as an integer.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_field\_type

ingres\_field\_type -- Get the type of a field in a query result

## Description

string **ingres\_field\_type** ( int *\$index* [, resource *\$link* ] )

Get the type of a field in a query result.

## Parameters

*index*

*index* is the number of the field and must be between 1 and the value given by [ingres\\_num\\_fields\(\)](#).

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

[ingres\\_field\\_type\(\)](#) returns the type of a field in a query result, or **FALSE** on failure. Examples of types returned are "IIAPI\_BYTE\_TYPE", "IIAPI\_CHA\_TYPE", "IIAPI\_DTE\_TYPE", "IIAPI\_FLT\_TYPE", "IIAPI\_INT\_TYPE", "IIAPI\_VCH\_TYPE". Some of these types can map to more than one SQL type depending on the length of the field (see [ingres\\_field\\_length\(\)](#) ). For example "IIAPI\_FLT\_TYPE" can be a float4 or a float8. For detailed information, see the Ingres/OpenAPI User Guide - Appendix C.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_num\_fields

ingres\_num\_fields -- Get the number of fields returned by the last query

## Description

int **ingres\_num\_fields** ( [ resource *\$link* ] )

[ingres\\_num\\_fields\(\)](#) returns the number of fields in the results returned by the Ingres server after a call to [ingres\\_query\(\)](#)

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns the number of fields.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_num\_rows

ingres\_num\_rows -- Get the number of rows affected or returned by the last query

## Description

int **ingres\_num\_rows** ( [ resource \$link ] )

This function is mainly meant to get the number of rows modified in the database. If this function is called before using [ingres\\_fetch\\_array\(\)](#), [ingres\\_fetch\\_object\(\)](#) or [ingres\\_fetch\\_row\(\)](#) the server will delete the result's data and the script won't be able to get them.

You should instead retrieve the result's data using one of these fetch functions in a loop until it returns **FALSE**, indicating that no more results are available.

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

For delete, insert or update queries, [ingres\\_num\\_rows\(\)](#) returns the number of rows affected by the query. For other queries, [ingres\\_num\\_rows\(\)](#) returns the number of rows in the query's result.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)

# ingres\_pconnect

ingres\_pconnect -- Open a persistent connection to an Ingres II database

## Description

```
resource ingres_pconnect ( [ string $database [, string $username [, string $password ] ] ] )
```

Open a persistent connection to an Ingres II database.

There are only 2 differences between this function and [ingres\\_connect\(\)](#): First, when connecting, the function will first try to find a (persistent) link that's already opened with the same parameters. If one is found, an identifier for it will be returned instead of opening a new connection. Second, the connection to the Ingres server will not be closed when the execution of the script ends. Instead, the link will remain open for future use ( [ingres\\_close\(\)](#) will not close links established by [ingres\\_pconnect\(\)](#) ). This type of link is therefore called 'persistent'.

## Parameters

*database*

The database name. Must follow the syntax *[node\_id::]dbname[/svr\_class]*.

*username*

The Ingres user name.

*password*

The password associated with *username*.

## Return Values

Returns a Ingres II link resource on success, or **FALSE** on failure.

## See Also

- [ingres\\_connect\(\)](#)
- [ingres\\_close\(\)](#)

# ingres\_query

ingres\_query -- Send a SQL query to Ingres II

## Description

bool **ingres\_query** ( string *\$query* [, resource *\$link* ] )

[ingres\\_query\(\)](#) sends the given *query* to the Ingres server.

The query becomes part of the currently open transaction. If there is no open transaction, [ingres\\_query\(\)](#) opens a new transaction. To close the transaction, you can either call [ingres\\_commit\(\)](#) to commit the changes made to the database or [ingres\\_rollback\(\)](#) to cancel these changes. When the script ends, any open transaction is rolled back (by calling [ingres\\_rollback\(\)](#) ). You can also use [ingres\\_autocommit\(\)](#) before opening a new transaction to have every SQL query immediately committed.

## Parameters

*query*

A valid SQL query (see the Ingres SQL reference guide). Some types of SQL queries can't be sent with this function:

- close (see [ingres\\_close\(\)](#) )
- commit (see [ingres\\_commit\(\)](#) )
- connect (see [ingres\\_connect\(\)](#) )
- disconnect (see [ingres\\_close\(\)](#) )
- get dbevent
- prepare to commit
- rollback (see [ingres\\_rollback\(\)](#) )
- savepoint
- set autocommit (see [ingres\\_autocommit\(\)](#) )
- all cursor related queries are unsupported

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.



## Examples

### Example #11 - [ingres\\_query\(\)](#) example

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");
while ($row = ingres_fetch_row()) {
    echo $row[1];
    echo $row[2];
}
?>
```

## See Also

- [ingres\\_fetch\\_array\(\)](#)
- [ingres\\_fetch\\_object\(\)](#)
- [ingres\\_fetch\\_row\(\)](#)
- [ingres\\_commit\(\)](#)
- [ingres\\_rollback\(\)](#)
- [ingres\\_autocommit\(\)](#)

# ingres\_rollback

ingres\_rollback -- Roll back a transaction

## Description

bool **ingres\_rollback** ( [ resource *\$link* ] )

[ingres\\_rollback\(\)](#) rolls back the currently open transaction, actually canceling all changes made to the database during the transaction.

This closes the transaction. A new one can be open by sending a query with [ingres\\_query\(\)](#).

## Parameters

*link*

The connection link identifier. If not specified, the last opened link is used.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## See Also

- [ingres\\_query\(\)](#)
- [ingres\\_commit\(\)](#)
- [ingres\\_autocommit\(\)](#)