mqseries

Introduction

This extension aims to provide an interface for communicating with IBMs WebSphere MQ series Queue managers.

The interface mimics the C-API client interface of WebSphere MQ Series as close as posible. Using the same naming conventions and posibilities of the C-API. In order to understand the workings of this extention some level of understanding the C-API is required.

For MQ-options, MQ-structures, MQ-results etc. please read the WebSphere MQ Application Programming Guide and WebSphere MQ Application Programming Reference.

Installing/Configuring

Requirements

A working IBM WebSphere MQ installation. If building the extention the SDK for IBM WebSphere MQ is also required.

Note
Be aware that when running against a IBM WebSphere MQ Client installation some methods are not available. This is not a problem of the extention but just the way the WebSphere MQ Client Interface works.

Installation requirements on non windows platforms

Build the extention and put it in the PHP extention directory.

Installation requirements on Windows

No additional requirements.

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/mqseries.

Note
The official name of this extension is *mqseries*.

There are two ways to connecto to a queue manager. These depend on the way the extention is compiled and linked.

- First one and also the default one is using the mqic libraries. Compiling and linking the extention against these IBM WebSphere MQSeries libraries allows the extention to connect to the Queue manager using the client interface. Remote conections are

posible this way.

- The other way is to compile and link against the mqm libraries. Using these libraries it is possible to make use of the transaction management of a queue manager.

Currently selecting the libraries to use is done by changing the config.m4 file.

Runtime Configuration

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

No extra configuration parameters exsists.

Resource Types

This extension defines a connection and object_handle resources.

The mqseries_conn() and mqseries_connx() define the connectionn handles.

The mqseries_open() defines the object handle.

Predefined Constants

For each WebSphere MQ Constant there is a mqseries conterpart.

For definitions and usage see the WebSphere MQ Application Programming Guide and WebSphere MQ Application Programming Reference red books.

The name of the mqseries counterpart is made by appending the WebSphere MQ constant with MQSERIES_, for example the CompletionCode constants are:

mqseries constants

PHP Constant	**MQ Constant**
MQSERIES_MQCC_OK	MQCC_OK
MQSERIES_MQCC_WARNING	MQCC_WARNING
MQSERIES_MQCC_FAILED	MQCC_FAILED
MQSERIES_MQCC_UNKNOWN	MQCC_UNKNOWN

mqseries Functions

mqseries_back

mqseries_back -- MQSeries MQBACK

Description

mqseries_back (resource `$hconn`, resource `$compCode`, resource `$reason`)

The mqseries_back() (MQBACK) call indicates to the queue manager that all the message gets and puts that have occurred since the last syncpoint are to be backed out. Messages put as part of a unit of work are deleted; messages retrieved as part of a unit of work are reinstated on the queue.

Using mqseries_back() only works in conjunction with mqseries_begin() and only function when connecting directly to a Queueu manager. Not via the mqclient interface.

Parameters

`hConn`
: Connection handle. This handle represents the connection to the queue manager.

`compCode`
: Completion code.

`reason`
: Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #1 - mqseries_back() example

```
<?php
   mqseries_back($conn, $comp_code, $reason);

   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("CompCode:%d Reason:%d Text:%s<br>\n", $comp_code, $reason,
mqseries_strerror($reason));
   }
?>
```

Notes

Note
mqseries_back() will not function when using MQSeries Client to connect to a Queueu Manager.

See Also

- mqseries_conn()
- mqseries_connx()
- mqseries_begin()

mqseries_begin

mqseries_begin -- MQseries MQBEGIN

Description

mqseries_begin (resource `$hconn`, array `$beginOptions`, resource `$compCode`, resource `$reason`)

The mqseries_begin() (MQBEGIN) call begins a unit of work that is coordinated by the queue manager, and that may involve external resource managers.

Using mqseries_begin() starts the unit of work. Either mqseries_back() or mqseries_cmit() ends the unit of work.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #2 - mqseries_begin() example

```
<?php
   $mqbo = array();
   mqseries_begin( $conn,
                   $mqbo,
                   $comp_code,
                   $reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       /* reason code 2121 is a warning for more information see MQSeries
reference manual.*/
       if ($reason !== 2121) {
           printf("CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
```

```
        }
    }
?>
```

Notes

Note
mqseries_begin() will not function when using MQSeries Client to connect to a Queueu Manager.

See Also

- mqseries_conn()
- mqseries_connx()
- mqseries_back()
- mqseries_cmit()

mqseries_close

mqseries_close -- MQSeries MQCLOSE

Description

mqseries_close (resource `$hconn`, resource `$hobj`, resource `$compCode`, resource `$reason`)

The mqseries_close() (MQCLOSE) call relinquishes access to an object, and is the inverse of the mqseries_open() (MQOPEN) call.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`hObj`
Object handle. This handle represents the object to be used.

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #3 - mqseries_close() example

```
<?php
   mqseries_close($conn, $obj, MQSERIES_MQCO_NONE, $comp_code, $reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("close CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
   }
?>
```

See Also

- mqseries_open()
- mqseries_conn()
- mqseries_connx()

mqseries_cmit

mqseries_cmit -- MQSeries MQCMIT

Description

mqseries_cmit (resource `$hconn`, resource `$compCode`, resource `$reason`)

The mqseries_cmit() (MQCMIT) call indicates to the queue manager that the application has reached a syncpoint, and that all of the message gets and puts that have occurred since the last syncpoint are to be made permanent. Messages put as part of a unit of work are made available to other applications; messages retrieved as part of a unit of work are deleted.

Parameters

`hConn`
: Connection handle. This handle represents the connection to the queue manager.

`compCode`
: Completion code.

`reason`
: Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #4 - mqseries_cmit() example

```
<?php
   mqseries_cmit($conn, $comp_code, $reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("cmit CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
   }
?>
```

Notes

Note
mqseries_back() will not function when using MQSeries Client to connect to a Queueu Manager.

See Also

- mqseries_begin()
- mqseries_back()
- mqseries_conn()
- mqseries_connx()

mqseries_conn

mqseries_conn -- MQSeries MQCONN

Description

mqseries_conn (string `$qManagerName`, resource `$hconn`, resource `$compCode`, resource `$reason`)

The mqseries_conn() (MQCONN) call connects an application program to a queue manager. It provides a queue manager connection handle, which is used by the application on subsequent message queuing calls.

Parameters

`qManagerName`
: Name of queue manager. Name of the queueu manager the application wishes to connect.

`hConn`
: Connection handle. This handle represents the connection to the queue manager.

`compCode`
: Completion code.

`reason`
: Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #5 - mqseries_conn() example

```
<?php
   mqseries_conn('WMQ1', $conn, $comp_code, $reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("conn CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
       exit;
   }
?>
```

See Also

- mqseries_disc()

mqseries_connx

mqseries_connx -- MQSeries MQCONNX

Description

mqseries_connx (string `$qManagerName`, array `$connOptions`, resource `$hconn`, resource `$compCode`, resource `$reason`)

The mqseries_connx() (MQCONNX) call connects an application program to a queue manager. It provides a queue manager connection handle, which is used by the application on subsequent MQ calls.

Parameters

`qManagerName`
: Name of queue manager. Name of the queueu manager the application wishes to connect.

`connOps`
: Options that control the action of function See also the MQCNO structure.

`hConn`
: Connection handle. This handle represents the connection to the queue manager.

`compCode`
: Completion code.

`reason`
: Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #6 - mqseries_connx() example

```
<?php
   $mqcno = array(
        'Version' => MQSERIES_MQCNO_VERSION_2,
        'Options' => MQSERIES_MQCNO_STANDARD_BINDING,
        'MQCD' => array('ChannelName' => 'MQNX9420.CLIENT',
        'ConnectionName' => 'localhost',
        'TransportType' => MQSERIES_MQXPT_TCP)
```

```
   );

   mqseries_connx('MQNX9420', $mqcno, $conn, $comp_code,$reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("Connx CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
       exit;
   }

?>
```

See Also

- mqseries_disc()

mqseries_disc

mqseries_disc -- MQSeries MQDISC

Description

mqseries_disc (resource `$hconn`, resource `$compCode`, resource `$reason`)

The mqseries_disc() (MQDISC) call breaks the connection between the queue manager and the application program, and is the inverse of the mqseries_conn() (MQCONN) or mqseries_connx() (MQCONNX) call.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #7 - mqseries_disc() example

```
<?php
   mqseries_disc($conn, $comp_code, $reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("disc CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
   }
?>
```

See Also

- mqseries_conn()
- mqseries_connx()

mqseries_get

mqseries_get -- MQSeries MQGET

Description

mqseries_get (resource `$hConn`, resource `$hObj`, array `$md`, array `$gmo`, int `$bufferLength`, string `&$msg`, int `&$data_length`, resource `&$compCode`, resource `$reason`)

The mqseries_get() (MQGET) call retrieves a message from a local queue that has been opened using the mqseries_open() (MQOPEN) call

Parameters

`hConn`
: Connection handle. This handle represents the connection to the queue manager.

`hObj`
: Object handle. This handle represents the object to be used.

`md`
: Message descriptor (MQMD).

`gmo`
: Get message options (MQGMO).

`bufferLength`
: Expected length of the result buffer

`msg`
: Buffer holding the message that was retreived from the object.

`data_length`
: Actual buffer length

`compCode`
: Completion code.

`reason`
: Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #8 - mqseries_get() example

```
<?php
// open connection to the queue manager
   mqseries_conn('WMQ1', $conn, $comp_code, $reason);
// $conn now hold the reference to the connection to the queue manager.

// open the connectio to the testq queueu
   mqseries_open(
               $conn,
               array('ObjectName' => 'TESTQ'),
               MQSERIES_MQOO_INPUT_AS_Q_DEF |
MQSERIES_MQOO_FAIL_IF_QUIESCING | MQSERIES_MQOO_OUTPUT,
               $obj,
               $comp_code,
               $reason);
// $obj now holds the reference to the object (TESTQ)

// setup empty message descriptor.
   mdg = array();
// setup get message options
   $gmo = array('Options' => MQSERIES_MQGMO_FAIL_IF_QUIESCING |
MQSERIES_MQGMO_WAIT, 'WaitInterval' => 3000);

// get the message from the queueu
   mqseries_get($conn, $obj, $mdg, $gmo, 255, $msg, $data_length,
$comp_code, $reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("GET CompCode:%d Reason:%d Text:%s<br>", $comp_code, $reason,
mqseries_strerror($reason));
   }

// open connection to the queue manager
   mqseries_conn('WMQ1', $conn, $comp_code, $reason);
// $conn now hold the reference to the connection to the queue manager.

// open the connectio to the testq queueu
   mqseries_open(
               $conn,
               array('ObjectName' => 'TESTQ'),
               MQSERIES_MQOO_INPUT_AS_Q_DEF |
MQSERIES_MQOO_FAIL_IF_QUIESCING | MQSERIES_MQOO_OUTPUT,
               $obj,
               $comp_code,
               $reason);
// $obj now holds the reference to the object (TESTQ)

?>
```

See Also

- mqseries_conn()
- mqseries_connx()

- mqseries_open()
- mqseries_put()

mqseries_inq

mqseries_inq -- MQSeries MQINQ

Description

mqseries_inq (resource `$hconn`, resource `$hobj`, int `$selectorCount`, array `$selectors`, int `$intAttrCount`, resource `$intAttr`, int `$charAttrLength`, resource `$charAttr`, resource `$compCode`, resource `$reason`)

The mqseries_inq() (MQINQ) call returns an array of integers and a set of character strings containing the attributes of an object.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`hObj`
Object handle. This handle represents the object to be used.

`selectorCount`
Count of selectors.

`selectors`
Array of attribute selectors.

`intAttrLength`
Count of integer attributes.

`intAttr`
Array of integer attributes.

`charAttrLength`
Length of character attributes buffer.

`charAttr`
Character attributes.

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #9 - mqseries_inq() example

```
<?php
   $int_attr = array();
   $char_attr = "";

   mqseries_inq($conn, $obj, 1, array(MQSERIES_MQCA_Q_MGR_NAME), 0,
$int_attr, 48, $char_attr, $comp_code, $reason);

   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("INQ CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
   } else {
       echo "INQ QManager name result ".$char_attr."<br>\n";
   }
?>
```

See Also

- mqseries_conn()
- mqseries_connx()
- mqseries_open()

mqseries_open

mqseries_open -- MQSeries MQOPEN

Description

mqseries_open (resource `$hconn`, array `$objDesc`, int `$option`, resource `$hobj`, resource `$compCode`, resource `$reason`)

The mqseries_open() (MQOPEN) call establishes access to an object.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`objDesc`
Object descriptor. (MQOD)

`options`
Options that control the action of the function.

`hObj`
Object handle. This handle represents the object to be used.

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #10 - mqseries_open() example

```
<?php
   mqods = array('ObjectName' => 'TESTQ');
   mqseries_open(
               $conn,
               $mqods,
               MQSERIES_MQOO_INPUT_AS_Q_DEF |
```

```
MQSERIES_MQOO_FAIL_IF_QUIESCING | MQSERIES_MQOO_OUTPUT,
               $obj,
               $comp_code,
               $reason);
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("open CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
       exit;
   }
?>
```

See Also

- mqseries_close()

mqseries_put1

mqseries_put1 -- MQSeries MQPUT1

Description

mqseries_put1 (resource `$hconn`, resource `$objDesc`, resource `$msgDesc`, resource `$pmo`, string `$buffer`, resource `$compCode`, resource `$reason`)

The mqseries_put1() (MQPUT1) call puts one message on a queue. The queue need not be open.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`objDesc`
Object descriptor. (MQOD) This is a structure which identifies the queue to which the message is added.

`msgDesc`
Message descriptor (MQMD).

`pmo`
Put message options (MQPMO).

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #11 - mqseries_put1() example

```
<?php
TODO:
?>
```

See Also

- mqseries_conn()
- mqseries_connx()
- mqseries_open()
- mqseries_get()

mqseries_put

mqseries_put -- MQSeries MQPUT

Description

mqseries_put (resource `$hConn`, resource `$hObj`, array `$md`, array `$pmo`, string `$message` , resource `$compCode`, resource `$reason`)

The mqseries_put() (MQPUT) call puts a message on a queue or distribution list. The queue or distribution list must already be open.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`hObj`
Object handle. This handle represents the object to be used.

`md`
Message descriptor (MQMD).

`pmo`
Put message options (MQPMO).

`message`
The actual message to put onto the queue.

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #12 - mqseries_put() example

```
<?php
// open connection to the queue manager
```

```
   mqseries_conn('WMQ1', $conn, $comp_code, $reason);
// $conn now hold the reference to the connection to the queue manager.

// open the connectio to the testq queueu
   mqseries_open(
               $conn,
               array('ObjectName' => 'TESTQ'),
               MQSERIES_MQOO_INPUT_AS_Q_DEF |
MQSERIES_MQOO_FAIL_IF_QUIESCING | MQSERIES_MQOO_OUTPUT,
               $obj,
               $comp_code,
               $reason);
// $obj now holds the reference to the object (TESTQ)

// setup the message descriptor array. Check MQSeries reference manuals.
   $md =  array(
               'Version' => MQSERIES_MQMD_VERSION_1,
               'Expiry' => MQSERIES_MQEI_UNLIMITED,
               'Report' => MQSERIES_MQRO_NONE,
               'MsgType' => MQSERIES_MQMT_DATAGRAM,
               'Format' => MQSERIES_MQFMT_STRING,
               'Priority' => 1,
               'Persistence' => MQSERIES_MQPER_PERSISTENT);

// setup the put message options.
   $pmo = array('Options' =>
MQSERIES_MQPMO_NEW_MSG_ID|MQSERIES_MQPMO_SYNCPOINT);

// put the message 'Ping' on the queueu.
   mqseries_put($conn, $obj, $md, $pmo, 'Ping', $comp_code, $reason);

   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("put CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
   }

// close the object reference $obj
   mqseries_close($conn, $obj, MQSERIES_MQCO_NONE, $comp_code, $reason);

// disconnect from the queue manager.
   mqseries_disc($conn, $comp_code, $reason);

?>
```

See Also

- mqseries_conn()
- mqseries_connx()
- mqseries_open()
- mqseries_get()

mqseries_set

mqseries_set -- MQSeries MQSET

Description

mqseries_set (resource `$hconn`, resource `$compCode`, resource `$reason`)

The mqseries_set() (MQSET) call is used to change the attributes of an object represented by a handle. The object must be a queue.

Parameters

`hConn`
Connection handle. This handle represents the connection to the queue manager.

`compCode`
Completion code.

`reason`
Reason code qualifying the compCode.

Return Values

No value is returned.

Examples

Example #13 - mqseries_set() example

```
<?php
TODO:
?>
```

See Also

- mqseries_inq()

mqseries_strerror

mqseries_strerror -- Returns the error message corresponding to a result code (MQRC).

Description

string **mqseries_strerror** (int `$reason`)

mqseries_strerror() returns the message that correspond to the reason result code.

Parameters

`reason`
Reason code qualifying the compCode.

Return Values

string representation of the reason code message.

Examples

Example #14 - mqseries_strerror() example

```
<?php
   if ($comp_code !== MQSERIES_MQCC_OK) {
       printf("open CompCode:%d Reason:%d Text:%s<br>\n", $comp_code,
$reason, mqseries_strerror($reason));
       exit;
   }
?>
```

The above example will output:

```
Connx CompCode:2 Reason:2059 Text:Queue manager not available for
connection.
```