

## Re: Return value in stored procedure

---

*Source:*

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.database.ado/2008-05/msg00017.html>

---

- *From:* "Kevin Frevert" <[kevin@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:kevin@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 12 May 2008 07:56:10 -0500
- 

"Man T" <[alan\\_nospam\\_pltse@xxxxxxxxxxxxx](mailto:alan_nospam_pltse@xxxxxxxxxxxxx)> wrote in message  
[news:4828264c@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:4828264c@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

I want to return a value 0 or 1 from the stored procedure in SQL 2000.  
How do I get this return value if I use TADOPStoredProcedure? Or do I need  
to use TADOQuery?

```
CREATE PROCEDURE dbo.gt_IsAddressOnHold (@Address varchar(40), @City
varchar(40), @State varchar(15), @PostalCode varchar(5), @CtryID varchar(2))
AS
```

```
SET NOCOUNT ON /* Suppressed the " Rows affected" message */
```

```
DECLARE
@Result Integer
```

```
SET @Result = 0 -- Default value
-- Your exectuting code
```

```
Return @Result
-----
```

In Delphi (you can use TADOPStoredProc, TADOQuery, but I would use  
TADOCommand)

Using the TADOCommand component, setting the appropriate properties (at  
design time), in particular change the CommandType to cmdStoredProc (the  
CommandText property will be changed to a drop-down of stored procedures  
(selecting the stored procedure of choice)

To make sure, select the Parameters property and verify Delphi 'filled in'  
the Parameters collection with the stored procedure parameters, in  
particular RETURN\_VALUE

In Delphi code (datamodule/form method). Disclaimer, code is off the top of  
my head, there are probably syntax errors:

## Re: Return value in stored procedure

```
function TdmData.IsAddressOnHold (const Address, City, State, PostalCode,
CtryID :String) : Boolean;
begin
with gt_IsAddressOnHold do
begin { Start of with ADO command do }
try { Start of try...except block }
//Set the corresponding properties
Execute();
Result := (Parameters.ParamByName('@RETURN_VALUE').Value = 1);
except
on e: Exception do
begin
Result := False;
//Raise a more meaningful/user-friendly error, log the
error, set a property, etc
end;
end; { End of try...except block }
end; { End of with ado command do }
end;
```